A Survey on Quantitative Evaluation of Web Service Security

Bo Zhou, Qi Shi, Po Yang
Department of Computer Science
Liverpool John Moores University
Liverpool, UK
{B.Zhou, Q.Shi, P.Yang}@ljmu.ac.uk

Abstract -**The number of web services available on the Internet has grown rapidly. Service consumers face a hard decision over which service to choose among the available ones. Security holds a key after various vulnerabilities have been exploited by attackers on number of notable web services. This paper carries out a survey on how security has been expressed and promised for web services, through both the Web Service Description Language and Service Level Agreements. It reviews existing technologies used for comparing individual web services, as well as for service compositions. Taking security into account further complicates the already difficult process of choosing the right service. The paper reveals that despite existing efforts, a quantitative solution needs to be established urgently in order to help service consumers to choose the most secure service for them to use.**

***Index Terms*—Web Service Security, Service Composition, Service Level Agreement, Quantitative Security Attribute.**

I. INTRODUCTION

In the digital world, a service is defined as a software unit that provides certain functionalities. A web service is a service that is made remotely available to other entities through networks. By using standard communication protocols and languages, web services provide the necessary interfaces so that any system can invoke them remotely. Moreover, through careful design, web services can work together in an ad-hoc manner in order to provide new applications that formed by one or more web services. The Service-Oriented Architecture (SOA) concept provides designs and frameworks to offer services as self-contained units [1]. One can invoke a web service, as long as the input satisfies the interface specifications, and let the output of the web service to be an input for another service. In this way new service compositions are formed by letting the web services communicate with each other. The most commonly used communication protocol for exchanging information between web services is Simple Object Access Protocol (SOAP) [2]. SOA platforms provide a foundation for modelling new service compositions, which involves planning, searching for, connecting, and invoking web services.

One of the issues faced by a service consumer is to choose a right service from potentially a very large service pool. Services provided by different providers may offer the similar functionality, but they could be very different in terms of cost, quality, or security. Therefore the service consumer faces the dilemma of picking up the most suitable services for his/her purpose. In this paper we review the existing methods that have been used to quantify and compare web services. In particular, we consider security as our top priority since most of the works to date focus on the quality of the services. There are two folds of research: the first is to compare individual web services; the second is to evaluate the service compositions.

The rest of the paper is organised as follows. The next section explains how web service security has been expressed in the past and the challenges service consumers are facing. Section III reviews existing techniques used for describing security properties for individual web services and the methods used for their comparisons. The differences service composition has brought to this issue are analysed in Section IV. Finally the paper concludes with a brief analysis of future directions in Section V.

II. WEB SERVICE SECURITY

Like any other entity exposed to the Internet, security is a big threat to web services. There are many incidents happened to a number of rather high-profile web service providers such as eBay and Apple in recent years. Therefore service consumers have to seek assurance from the service providers for the security and privacy of their data, before using the services. This is particularly paramount to organizations using web services, as the security is not just vital to their businesses, but also regulated by authorities. As we mentioned earlier, since many web services offered in the market have similar functionality, to some extent security could play the key role in helping service consumer to make the decision on which service to use. Therefore it is important that the security solutions employed by the service provider are both measurable and auditable by the service consumer. In this section, we first explain how the security attributes are expressed for web services.

A. Security requirements and SLAs

Web services are normally made available together with a service-level agreement (SLA) [3]. A SLA is a formal guarantee that has to be accepted by service consumers before the service being used. A SLA can specify the properties of a service across different levels. For example, on business level it can describe what kind of functionality the service is offering and how the users will be charged (cost); on technical level it may describe the number of shutdowns the service might experience each year, i.e. the Quality of Service (QoS).

Security can also be promised as part of the SLA. However its coverage is rather poor to date due to the lack of well defined semantics. The SLAs traditionally focus on the QoS metrics such as a bandwidth guarantee and backup strategy. Even when the security being mentioned, in practice it tends to be written in a natural language with fuzzy terms such as 'High', 'Good', and etc. Therefore it is very difficult for the service consumer to really understand the situation and compare the web services from the security perspective. Nevertheless some works have been done in order to address this issue as we will explain in the next section. Here we just show an example schema from the paper [12], which extends the standard WS-Agreement [31] schema to cover the security requirements.

Table 1 SLA Schema for data retention requirement

```
<xs:simpleType name= ”DataRetention”>
<xs:annotation>
<xs:documentation>
       The DataRetention condition specifies how
       long data will be retained by the service
       provider.
</xs:documentation>
</xs:annotation>
       <xs: restriction base= ”xs: string”>
              <xs:enumeration
       value= ”no-retention”/>
              <xs:enumeration
       value= ”stated-purpose”/>
              <xs:enumeration
       value= ”legal-requirement”/>
              <xs:enumeration
       value= ”business-practices ”/>
              <xs:enumeration
       value= ”indefinitely ”/>
       </xs: restriction>
</xs:simpleType>
```

Table 1 shows how we could specify the security requirement on data retention in an SLA. There are five possible outcomes:

- no–retention: The consumer's data will not be retained by the service.
- stated–purpose: Data will be discarded at the earliest time possible.
- legal–requirement: Data retained as required by law or liability under applicable law.
- business–practices: Data retained according to the service provider's business practices, with an explicit destruction time table.
- indefinitely: literally no guarantee about data retention and discard.

Table 2 WSDL and WS-SecurityPolicy example

```
<wsdl:definitions
targetNamespace="http://ws.sosnoski.com/library/w
sdl"...>

  <wsp:Policy wsu:Id="UsernameToken"
xmlns:wsp="http://www.w3.org/ns/ws-policy"...>
    <sp:SupportingTokens>
      <wsp:Policy>
        <sp:UsernameToken
sp:IncludeToken=".../IncludeToken/AlwaysToRecipie
nt">
          <wsp:Policy>
            <sp:HashPassword/>
          </wsp:Policy>
        </sp:UsernameToken>
      </wsp:Policy>
    </sp:SupportingTokens>
  </wsp:Policy>

  <wsp:Policy wsu:Id="SymmEncr"
xmlns:wsp="http://www.w3.org/ns/ws-policy"...>
    <sp:SymmetricBinding>
      <wsp:Policy>
        <sp:ProtectionToken>
          <wsp:Policy>
            <sp:X509Token
sp:IncludeToken=".../IncludeToken/Never">
              ...
            </sp:X509Token>
          </wsp:Policy>
        </sp:ProtectionToken>
        ...
      </wsp:Policy>
    </sp:SymmetricBinding>
    ...
  </wsp:Policy>
   ...
  <wsdl:binding name="LibrarySoapBinding"
type="wns:Library">
    <wsp:PolicyReference
xmlns:wsp="http://www.w3.org/ns/ws-policy"
        URI="#UsernameToken"/>
    ...
    <wsdl:operation name="getBook">
      <wsp:PolicyReference
xmlns:wsp="http://www.w3.org/ns/ws-policy"
          URI="#SymmEncr"/>
      <wsdlsoap:operation
soapAction="urn:getBook"/>
      <wsdl:input name="getBookRequest">
        <wsdlsoap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="getBookResponse">
        <wsdlsoap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  ...
</wsdl:definitions>
```

B. WSDL and WS-SecurityPolicy

Apart from the SLAs, a web service also describes its interfaces through a Web Service Description Language (WSDL). A WSDL file specifies how to invoke the service, i.e. the input parameters in order to communicate with the service and the expected output for each of the operations provided by the service. The file can be generated automatically by programs from the web service. Based on the WSDL specification files, a service consumer can design his/her service composition accordingly and use SOAP to call the operations listed in the WSDL files.

Although WSDL is mostly used to specify the functional aspects of a service, it is possible to attach non-functional properties such as security to the WSDL. WS-SecurityPolicy [32] is an extension to WSDL to secure SOAP messages. It utilises standards like SAML [28], XML Signature [29] and XML Encryption [30] to achieve the goal of secure communications with web services. WS-SecurityPolicy is different from the SSL (Secure Socket Layer) protocol as the WS-SecurityPolicy only encrypts the content of a SOAP message while SSL can encrypt the entire communication channel. Comparing to SSL, WS-SecurityPolicy is more flexible as it can choose which part of the SOAP message to be encrypted by using which cryptographic algorithm. WS-SecurityPoliy is attached to the WSDL by declaring itself in the WSDL file. An example of WS-Security and its application in a WSDL file is presented in Table 2 [22].

In this example, there are two security elements: UsernameToken and SymmEncr (Symmetric Encryption). When they are referred in the WSDL, the UsernameToken is required at the <wsdl:binding> level, and the SymmEnrc is only required by the getBook operation. This shows how WS-SecurityPolicy can be enforced at different levels in a WSDL file. More details about the WS-SecurityPolicy can be found in [23].

C. Challenges

Despite of the introductions of SLA and WSDL, security remains a big challenge for web services. One service developed with good faith in its security may not be necessarily good enough for another to use. The dilemmas faced by a service consumer are in three folds.

- Firstly, security is a broad concept that includes many aspects such as confidentiality and privacy. One service may be stronger than another in terms of confidentiality; while it is also possible that the very same service has weaker protection of privacy. It is a typical multi-criteria issue, which service consumers are not always in the position to resolve due to the lack of expertise.
- Secondly, WS-SecurityPolicy was proposed to secure the SOAP messages. It is well equipped for - but also limited to – the security of communication with web services. Security requirements at higher levels, such as the data retention issue mentioned in Table 1, are hard to express by using WS-SecuirtyPolicy. In contrast, security descriptions in SLAs are more open and inclusive but not always precise. It lacks a widely accepted standard to help understand the real strength of the security solution that a service is offering. In some cases security attributes are expressed by fuzzy languages such as "High", "Good", "Fair" and etc, which makes the comparison of different security attributes very difficult. Moreover, the situation gets even more complicated when more than one SLA languages are used by different service providers.
- Finally, even though some security modelling and verification techniques allow the service consumer to specify certain security properties that the service has to comply with before the service being used [4], in practice the number of services that satisfy the security requirements could still be very large. Therefore the service consumer still needs help to make the right choice from a potentially very large pool of services.

In summary, WS-SecurityPolicy is not inclusive enough to express security requirements other than communication needs. SLAs, on the other hand, are flexible but need a carefully designed schema and ontology to express security in a clear and precise fashion. Even if this can be addressed, service consumers still need a straightforward solution to compare and pick up the right services to maximise the security. In the next two sections we will look at existing efforts in attempting to solve these problems for individual web services and service compositions respectively.

III. Comparing individual web services

A. Expressing Security in SLAs

As we already explained the SLAs are more flexible in terms of defining complex security requirements. The problem with SLAs is the lack of a common ground for the definition and interpretation of security. Crucially this makes it very difficult to make the SLAs machine readable. Some works have been done in the past in order to express the security attirbutes of web services in the SLAs and help the consumers to compare the web services in an automatic way.

Paper [7] was among the first works trying to address the quantifiable security issue in SLAs. Basically it tries to express and measure the security of a service by associating it with performance related metrics. For example, a security requirement of "restore backed up data" is measured by the quantifiable metric of "data restored 95% of time within response time". The way the security has been expressed is rather subjective, depending on the scenario of each enterprise, where the research was targeting. Therefore the process cannot be implemented automatically. Instead, it requires a close study of the enterprise's configurations by security specialists.

SecAg is another framework proposed to express security metrics in SLAs [9][10]. Similar to the method used in Table 1, SecAg also extends the standard WS-Agreement to provide necessary semantics for specifying security properties. For example, with the extensions it can specify which service level objective (SLO) is auditable and assign an access control list to the SLO. Based on the extensions, the author also proposed a risk-based approach for service matchmaking. Each SLO is assigned a weight *w* representing the risk that the SLO is not fulfilled. By calculating the weighted Euclidean distance of each SLA to the security requirements, using techniques such as a text similarity analyser, the SLA that is closest to the security requirements will be selected as the risk is at the minimum.

For cloud consumers, before employing any cloud service they have to make sure that the service is compliant with their security requirements. In addition, business users seek for assurance that the cloud service they use comply with both industrial standards and government legislations. Unfortunately, SLAs often are not rich enough or directly linked with such legislations or standards, in order to support an automatic compliance check. Paper [13] tried to solve the issue by proposing a compliance vocabulary to embed security controls in the SLAs of cloud services. This vocabulary is associated with the security controls from governance documents. Therefore the SLAs become more transparent to the consumers in terms of the level of security being offered. Maintenance of such vocabulary is a challenging task though.

B. Ranking the Web Services

After expressing the security in the SLAs, it is still necessary to compare and rank the web services based on the consumer's requirements. In the past the focus was on raking web services based on just their QoS metrics and trying to find the best match. We analyse each of them and consider how to fit security into the existing ranking strategies.

Paper [19] proposed a Web Service Relevancy Function (WsRF) to measure the relevancy ranking of a particular web service based on the user's preferences (weights) and the QoS metrics such as Response Time, Throughput, Availability and Cost. It uses a simple mathematical matrix to normalize the QoS metrics of web services. The method is suitable for QoS metrics that have real numbers. However as security is often described by fuzzy terms, the application of this method is limited. Similarly, paper [20] uses a Singular Value Decomposition (SVD) based technique, and a user assisted weighting system to find higher order correlations among web services. This enables the selection of web services without an exact match of required QoS attributes.

Paper [21] ranks web services under multi-criteria matching. It targets at accurate web service selection and assigns a dominance score to each advertised web service. Security unfortunately was not the research focus. Other similar works include paper [18], which defines a business-focused ontology to enable semantic matchmaking in open cloud markets.

Paper [17] proposed the concept of Quality of Security Service. It treats the security as part of QoS requirements. The author argues that security requirements such as the strength of a cryptographic algorithm, the length of a cryptographic keys, security functions, confidence of policy-enforcement and the robustness of an authentication mechanism would all be specified and measured as the quality of security services. However no explicit example was given in the paper.

Paper [11] proposed an AHP (Analytic Hierarchy Process) based framework for web service quality evaluation. It uses a quality meta-model to format SLAs and assigns weights to different quality characteristics based their importance. The web services are measured by a satisfaction function, which covers both measurable and non-measurable characteristics. For example, the property of *Confidentiality* is measured by combining the encryption algorithm, key length and key protection used. The web service that has the greatest value in satisfaction function will be chosen.

Paper [15] proposes a method for finding semantically equal SLA elements from different SLAs by utilizing several machine learning algorithms. The user requirements are specified in a SLA template, which are compared to different SLAs offered by various service providers. The offered SLAs can be specified in different languages. This method tries to map the elements in different SLAs and generates an equivalence probability score. The cloud service that has the highest score will be selected for the users automatically.

IV. Comparing Service Compositions

A. Service Composition and SLAs

An important feature of SOA is that the web services can be composed together to form new applications. A service composition is also a web service, but consists more than one individual services. The structure of the composition decides its functionality and internal process. For example, a *Travel Planner* service composition includes four individual services of *GPS*, *Map, Route* and *Payment* as shown in Figure 1. With this service composition, a traveler can specify point of interest (POI) through his/her mobile phones. The *Travel Planner* service will provide *Route* service, after invoking *GPS* and *Map* services respectively. At the end of the process the *Travel Planner* could also provide a *Payment* service for purchasing tickets for transportation or visiting the POI. Here the ⊕ symbol means the *GPS* service and *Map* service are executed in parallel.

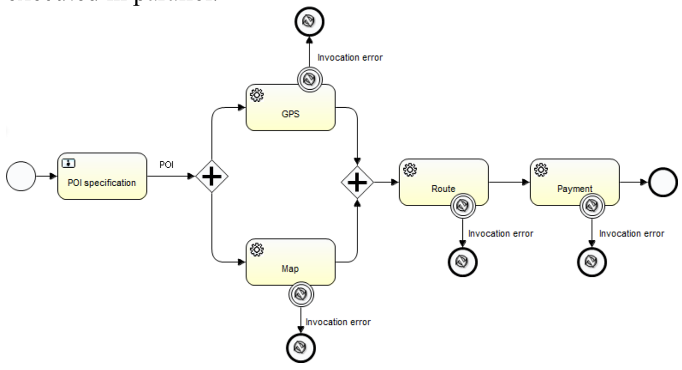


Figure 1 Travel Planner service composition

In contrast to individual web services, a service composition may not have a dedicated WSDL as the interfaces to invoke the composition are provided separately by the individual services, which serve as entry points to the composition. When it comes to the SLAs, it is possible that a service composition may provide a SLA to the service consumer on request, but the real questions is what should be included in the SLA and how to assure it. The situation is more complex simply because in many cases the service compositions are maintained in an ad-hoc manner, i.e. individual services join and leave a composition at runtime, due to various reasons. For example, a service may encounter an error and be replaced by another service to maintain the functionality of the entire composition. Therefore the properties of the composition can be changed at runtime. It then becomes a question that how the SLA can be honoured or updated at runtime to reflect the change of composition. In respect to security, it is very hard to guarantee that the new service composition will always comply with the consumer's security requirements.

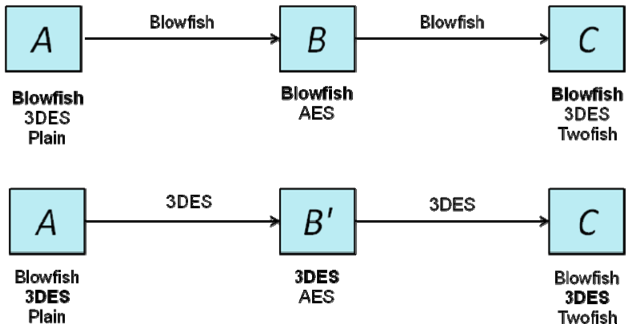


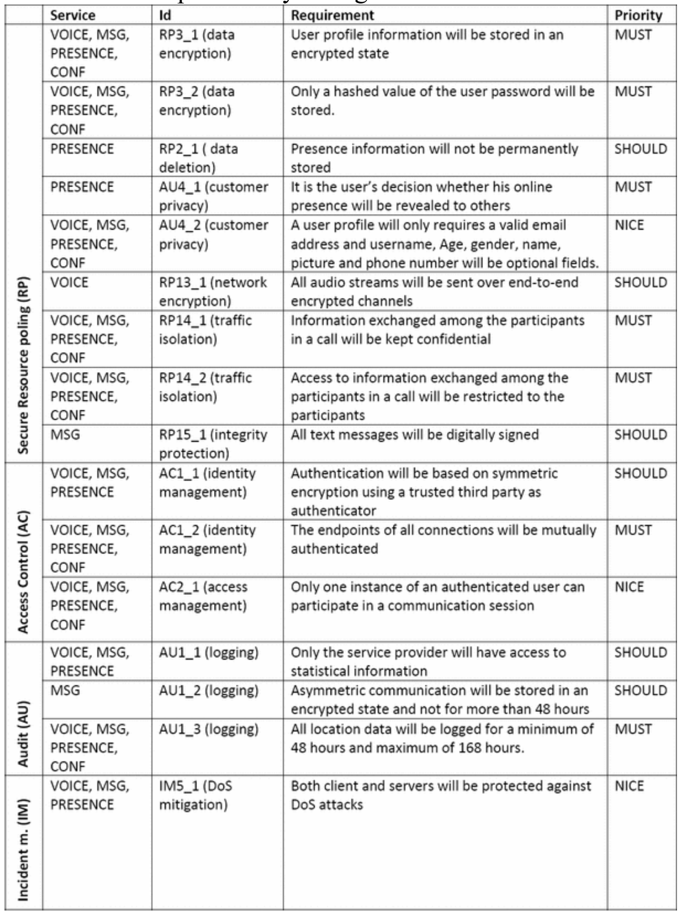
Figure 2 Composition of three services

To further illustrate the problem, taking the example in Figure 2, which demonstrates a scenario where services A, B, and C are composed in a sequence order. The three services support different cryptographic algorithms and have to use the commonly supported Blowfish [24] for secure communications with each other in the composition. Suppose that later on vulnerability has been found with service B and it has to be replaced by another service B'. However, it transpires that the only common algorithm supported by B', A and C is 3DES [25]. Thus the security properties offered by the entire composition is consequently changed from Blowfish to 3DES, and the situation could be much worse if no common algorithm is found at all in this case.

To solve the SLAs maintenance issue, paper [16] argues that a service broker's role is necessary. The broker analyses the needs of the service consumers and helps to select and compose the individual services into new compositions required. As shown in Figure 3, it gives a very good example about what needs to be covered in the SLAs for a composed online meeting service, which has four individual services: *VOICE*, *MSG*, *PRESENCE*, and *CONF*. Furthermore, paper [12] proposed a scheme to let the service broker bridge the security requirements from service consumers and the security guarantees promised by service providers. The broker is in charge of compositing web services and maintaining a mutually agreed contract. As we already mentioned in Table 1, this work uses a schema that is based on the WS-Agreement.

B. Ranking Service Compositions

To make sure that a service composition always satisfies its consumer's requirements, constant analysis and monitoring of the composition are essential, which necessitates quantitative calculations and comparisons of the security attributes of service compositions. Because the service compositions are offered at runtime, its security properties need to be determined in real time as well. Ranking service compositions is different from ranking individual services as it must consider the structure of the service compositions and make recommendations to the consumers with the maximum security. Some previous works tried to calculate the properties of a service composition by taking its structure into account.



	Service	Id	Requirement	Priority
Secure Resource poling (RP)	VOICE, MSG, PRESENCE, CONF	RP3_1 (data encryption)	User profile information will be stored in an encrypted state	MUST
	VOICE, MSG, PRESENCE, CONF	RP3_2 (data encryption)	Only a hashed value of the user password will be stored.	MUST
	PRESENCE	RP2_1 (data deletion)	Presence information will not be permanently stored	SHOULD
	PRESENCE	AU4_1 (customer privacy)	It is the user's decision whether his online presence will be revealed to others	MUST
	VOICE, MSG, PRESENCE, CONF	AU4_2 (customer privacy)	A user profile will only requires a valid email address and username, Age, gender, name, picture and phone number will be optional fields.	NICE
	VOICE	RP13_1 (network encryption)	All audio streams will be sent over end-to-end encrypted channels	SHOULD
	VOICE, MSG, PRESENCE, CONF	RP14_1 (traffic isolation)	Information exchanged among the participants in a call will be kept confidential	MUST
	VOICE, MSG, PRESENCE, CONF	RP14_2 (traffic isolation)	Access to information exchanged among the participants in a call will be restricted to the participants	MUST
	MSG	RP15_1 (integrity protection)	All text messages will be digitally signed	SHOULD
Access Control (AC)	VOICE, MSG, PRESENCE	AC1_1 (identity management)	Authentication will be based on symmetric encryption using a trusted third party as authenticator	SHOULD
	VOICE, MSG, PRESENCE, CONF	AC1_2 (identity management)	The endpoints of all connections will be mutually authenticated	MUST
	VOICE, MSG, PRESENCE, CONF	AC2_1 (access management)	Only one instance of an authenticated user can participate in a communication session	NICE
Audit (AU)	VOICE, MSG, PRESENCE	AU1_1 (logging)	Only the service provider will have access to statistical information	SHOULD
	MSG	AU1_2 (logging)	Asymmetric communication will be stored in an encrypted state and not for more than 48 hours	SHOULD
	VOICE, MSG, PRESENCE, CONF	AU1_3 (logging)	All location data will be logged for a minimum of 48 hours and maximum of 168 hours.	MUST
Incident m. (IM)	VOICE, MSG, PRESENCE	IM5_1 (DoS mitigation)	Both client and servers will be protected against DoS attacks	NICE

Figure 3 SLA for an Online Meeting Service Composition

Paper [14] targets a business process with service composition in mind. It proposed the idea of Quality of Protection (QoP), in contrast to the traditional QoS. The QoP emphasises on properties that relate to security, such as incident recovery time, backup frequency etc. It does not cover security requirements beyond the basic level nor mention how to aggregate QoP properties when services are composed together.

Paper [6] focuses on the QoS values of service composition. It determines the QoS values based on the structure of the services, i.e. looking at their relationships. Based on the process sequence such as *loop*, *and*, *or*, etc, the QoS values are calculated based on predefined rules. For example, in a sequential process, the execution time of a service composition is calculated as the sum of each individual service's execution time; while in the case of a parallel process, it would be the maximum execution time of all the individual services involved. This work is very useful but it does not consider the security attributes of service compositions.

Paper [26] uses a similar method but the focus is on the trustworthiness of service composition. It also uses the structure of service composition to decide its property's value. The authors argue that the trustworthiness value of a service composition is a combined result of relevant properties such as reputation, reliability and availability etc. These properties are one step closer towards the general security properties targeted in this paper.

Aniketos is an EU-funded FP7 project which solves security issues in service composition [27]. It adopts the concept of a service broker and uses Business Process Model and Notation (BPMN) [33] to construct service composition, as the example shown in Figure 1. The ranking of service compositions in Aniketos is based on verifiable security properties, such as Separation of Duty and Binding of Duty in compliance with the Role Based Access Control (RBAC). For example, Separation of Duty means that two services in a composition cannot be executed by the same role. Aniketos also checks for harmful functionalities by using byte code verification, as well as the security specifications embedded into the WSDL file of each individual service. The security properties verified in Aniketos are mostly binaries, i.e. the verification module returns a result of either True or False. It works well if we only want to know if a service composition satisfies the a consumer's security requirements. However, when the service pool is getting larger, it would be useful if the services can be compared and ranked in a more fine-grained manner.

V. Conclusions and future work

The Internet becomes a world full of web services. There are already some service repositories that can store and offer a wide range of web services to service consumers, who can pick up a service or form a service composition straightaway under the SOA framework.

However, comparing and choosing the most appropriate services is not easy, especially when the security of the services is considered in the process. First of all, neither WSDL nor SLA describes security properties of web services in a consistent and precise manner due to the lack of presence of security in these files in the past. Secondly, comparing security properties is a multi-criteria problem that requires more efforts to resolve. Finally, determining the security properties of a service composition is difficult as the quantitative evaluation has to be carried out in real time and take the structure of the composition into account.

In this paper, we conducted a brief survey of the existing ranking techniques used to compare both individual services and service compositions, and concluded that a quantitative evaluation method is urgently needed in order to solve the issue. As a future work we plan to target the comparison of individual services first. It will be based on linguistic evaluation of those fuzzy terms already exist in SLAs in order to make immediate impact on real world scenarios.

Acknowledgment

The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no. 257930.

References

[1] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, 2005.

[2] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H. Nielsen, A. Karmarkar, and Y. Lafon, Simple Object Access Protocol (SOAP) 1.2, Tech. rep., World Wide Web Consortium (W3C), 2007.

[3] H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck, Web Service Level Agreement (WSLA) Language Specification, version 1.0, Tech. rep., IBM Corporation, 2003.

[4] A. Brucker, F. Malmignati, M. Merabti, Q. Shi, and B. Zhou. The Aniketos Service Composition Framework: Analysing and ranking of secure services, Secure and Trustworthy Service Composition, Springer International Publishing 978-3-319-13517-5, 8900, pp. 121-135.

[5] B. Zhou, D. Llewellyn-Jones, Q. Shi, M. Asim, M. Merabti, and D. Lamb. Secure Service Composition Adaptation Based on Simulated Annealing. Proceedings of the 6th Layered Assurance Workshop, Annual Computer Security Applications Conference (ACSAC 2012) , Orlando, Florida, USA, Dec. 2012, pp. 49-55.

[6] M. Jaeger, G. Rojec-Goldmann, and G. Muhl. QoS Aggregation in Web Service Compositions. Proceedings of IEEE International Conference on e-Technology e-Commerce and e-Service, Hong Kong, March/April 2005, pp. 181-185.

[7] R. Henning, Security Service Level Agreements: Quantifiable Security for the Enterprise? ACM Proceedings of the 1999 Workshop on New Security Paradigm, Sep. 1999, pp. 54-60.

[8] C. E. Irvine, and T. Levin. Toward Quality of Security Service in a Resource Management System Benefit Function. Proceedings of the Heterogeneous Computing Workshop (HCW2000), Cancum, Mexioc, May 2000, pp.133-139.

[9] M. Hale, and R. Gamble, Risk Propagation of Security SLAs in the Cloud, Proceeding of the workshop on Management and Security technologies for Cloud Computing, 2012 IEEE GLOBECOM Workshops, Anaheim, CA, USA, Dec. 2012, pp.730-735.

[10] M. Hale, and R. Gamble, SecAgreement: Advancing Security Risk Calculations in Cloud Services, 2012 IEEE Eighth World Congress on Services, Honolulu, HI, US, Jun. 2012, pp.133-140.

[11] V. Casola, A. Fasolino, N. Mazzocca, and P. Tramontana. An AHP-based Framework for Quality and Security Evaluation, Proceedings of the 12th IEEE international conference on computational science and engineering, CSE 2009, vol. 3, 2009, pp. 405-411.

[12] P. H. Meland, K. Bernsmed, M. G. Jaatun, H. N. Castejon, and A. Undheim, Expressing cloud security requirements for SLAs in deontic contract languages for cloud brokers, International Journal of Cloud Computing, 2014, Vol.3, No.1, pp.69 - 93.

[13] M. L. Hale, and R. F. Gamble, Building a Compliance Vocabulary to Embed Security Controls in Cloud SLAs. 2013 IEEE Ninth World Congress on Services, Santa Clara, CA, USA, Jun./Jul. 2013, pp.118-125.

[14] G. Frankova, and A. Yautsiukhin, Service and protection level agreements for business processes, Proceedings of the IEEE Second European Young Researchers Workshop on Service Oriented Computing, 2007, page 38.

[15] C. Redl, I. Breskovic, I. Brandic, and S. Dustdar. Automatic SLA Matching and Provider selection in grid and cloud

computing markets, Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing (GRID), pp. 85-94.

[16] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, Security SLAs for Federated Cloud Services, Proceedings of the Sixth International Conference on Availability, Reliability and Security (ARES), Vienna, Aug. 2011, pp. 202-209.

[17] C. Irvine, and T. Levin, Quality of security service, Proceedings of 2000 Workshop on New Security Paradigms, 2001, pp. 91-99.

[18] G. D. Modica, G. Petralia, and O. Tomarchio, A Business Ontology to Enable Semantic Matchmaking in Open Cloud markets, Proceedings of the 8th International Conference on Semantics, Knowledge, and Grids, 2012, pp. 96-103.

[19] E. Al-Masri, and Q.H. Mahmoud, QoS-based discovery and ranking of web services, Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, Aug. 2007, pp. 529-534.

[20] H. Chan, T. Chieu, and T. Kwok, Autonomic Ranking and Selection of Web Services by Using Single Value Decomposition Technique, Proceedings of IEEE International Conference on Web Services (ICWS 08), Beijing, China, Sept. 2008, pp. 661-666.

[21] D. Skoutas, D. Sacharidis, A. Simitsis, V. Kantere, and T. Sellis, Top-k Dominant Web Services under Multi-Criteria Matching, Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Mar. 2009, pp. 898 -909.

[22] D. Sosnoski, Java web services: Understanding WS-Policy: Learn the details of WS-Policy, and see which parts work with three open source Java web services stacks, developerWorks, IBM, Nov. 2010.

[23] J. Rosenberg, and D. Remy, Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption. Pearson Higher Education, 2004.

[24] B. Schneier, Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). Proceedings of the Second International Workshop on Fast Software Encryption, Leuven, Belgium, Dec. 1994, pp. 191-204.

[25] W.C. Barker, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Revised 19 May 2008, NIST Special Publication 800-67, Version 1.1, 2008.

[26] H. Elshaafi, J. McGibney, and D. Botvich, Trustworthiness monitoring and prediction of composite services. Proceedings of 2012 IEEE Symposium on Computers and Communications (ISCC), Cappadocia, Turkey, Jul. 2012, pp. 580–587.

[27] B. Zhou, Final prototype of secure service composition. Seventh Framework Programme: Aniketos – Secure and Trustworthy Composite Services, Technical Report, Oct. 2013.

[28] Security Assertion Markup Language (SAML) v2.0, OASIS Security Services TC, Mar. 2005.

[29] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon, XML Signature Syntax and Processing (Second Edition), W3C Recommendation, 10th Jun. 2008.

[30] T. Imamura, B. Dillaway, and E, Simon, XML Encryption Syntax and Processing, W3C Recommendation, 10th Dec. 2002.

[31] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke and M. Xu, Web Services Agreement Specification (WS-Agreement), 14th Mar. 2007.

[32] A. Nadalin, M. Goodner, M. Gudgin, and A. Barbir, and H. Granqvist, WS-SecurityPolicy v1.2, OASIS Standard, 1st Jul. 2007.

[33] Business Process Model and Notation (BPMN) v2.0, Object Management Group, Inc. (OMG), Jan. 2011.