Security framework for dynamic service-oriented IT systems

Grzegorz Kolaczekea and Jolanta Mizera-Pietraszkob

Faculty of Computer Science and Management, Wroclaw University of Science and Technology, Wroclaw, Poland; Faculty of Mathematics, Physics and Computer Science, Opole University, Opole, Poland

ABSTRACT
The paper proposes a framework for dynamic service-oriented IT systems security. We review the context of service-oriented architecture (SOA), which constitutes a paradigm of dynamic system configuration including security constraints at the system module development stage, supporting with the domain-driven resources, carrying out routine SOA maintenance and implementing XML-compatible parsing technologies in order to improve the system performance. Likewise, we discuss the fundamental differences between security management systems with traditional centralized and monolithic architecture and service-oriented IT systems from the perspective of security-related issues. Web services security becomes fairly crucial, in particular, when it relates to distributed system environments. Our multi-layered reference framework for service-oriented systems is aimed at principal objectives predominantly related to IT systems security working in dynamic environments. Furthermore, we carry out an in-depth security analysis of a multi-agent system design dedicated to work in the service-oriented environments. Finally, we conclude briefly with the findings of our study on IT security requirements and performance on the comparison basis of correlation between the observations at the low and at high layers of our reference security model. The paper is an extended version of INISTA 2017 paper (Kolaczeck, G., & Mizera-Pietraszko, J. (2017) and presents more detailed related works overview, explanation of the subjective logic application in the process of security level evaluation and extensive discussion of the obtained results and their role in SOA security level modelling.

ARTICLE HISTORY
Received 29 November 2017
Accepted 20 May 2018

KEYWORDS
Service-oriented systems; security level; security incidents

Introduction
The ability to provide security in computer systems is one of the most principal non-functional requirements in software engineering and it is also the quality attribute discernible at the system runtime. While at the early stages of preliminary considerations of IT systems security, the issues were marginalized (Harris & Hunt, 1999), over the recent years, with the dramatic increase of accessibility to information owning to its ubiquitous nature, security aspects have become the key element for making the decision on the further system development (Dlamini, et.al., 2009). Moving from monolithic architectures and centralized...
systems to distributed and service-oriented systems makes the yet reliable conventional methods of security assessment not sufficient nowadays and thus, while building infrastructure in the area, they need a strong support with the new aspects such as: a high level of the resources heterogeneity, their diverse granularity and a high dynamic diversity (Chneider, 2012), (Conti et al., 2011). Service-oriented systems encounter many security problems that cannot be resolved simply by implementing the patterns and solutions applicable to the systems whose architectures are of a different that service-oriented nature (e.g. centralized systems, monolithic) (Chen, Paxson, & Katz, 2010). An example illustrating the specificity of the problems in service-oriented systems may be a task of data protection to prevent security code breaches. In monolithic systems, separation of resources, is a standard architecture, as the data processed by the system remain within it and therefore an access to the particular data set implies the necessity to have an appropriate permissions granted in the context of this system and its specific resources. Consequently, in order to protect the confidentiality and integrity of the system data, it is enough to provide a full strong access control. On the other hand, data processing in service-oriented systems such as ‘software as a service’ (SaaS) platforms, can be easily disclosed and modified in an unauthorized manner even despite a reliable massive access control. One of the reasons is that confidentiality is susceptible to loss due to some errors in the system separation mechanisms where the data is stored while the physical location of the data is shared by some other services at the same time (Rahman & Choo, 2014). In order to make it global, security requirements for Service-Oriented Architecture (SOA) include identification of message-level security provided by multiple handlers that exchange information in real-time. Web Service Description Language (WSDL) allows the Web Service Provider to provide an encrypted key information to the client while operating the service (Shah & Patel, 2008).

Figure 1. SOA layered architecture.


It security

**Generic security problems in IT systems**

Specificity of IT security problems stems from the diversity of information system architecture. Security issues related to centralized systems, in which processing is performed in the highlighted central node, are a subset of problems arising in the systems having distributed architecture. Centralized systems enable simplified processing control procedure and provide unrestricted access to the resources. Naturally, the central system component is to be protected and monitored in real-time. Distributed systems allow data processing and storage in different places of the system environment, quite often geographically located very far from each other. Such a quality of distributed systems implies that not only the same security level should be ensured at all the locations of the system runtime environment, but the whole diversity of the local system components such as diverse computing power, different levels of interaction with the users, variation of requirements arising from the locally applicable legal system, etc., should comply with the central server. For distributed systems, the critical factor is also the requirement to ensure a secure communication between all the system components. Complexity level of selection of adequate methods and mechanisms for security management system grow proportionally to the number and diversity of the distributed system components being a subject to protection (Benson, Akyildiz, & Appelbe, 1990).

Pragmatically, heterogeneity of the solutions in designing computer system architecture also impedes the process of providing a reliable security level evaluation. Nowadays different solutions for utilizing physical system resources and the software layer are researched. This creates the need for the IT security management of the technical specifications of every system characteristic (e.g. the availability of mechanisms which support hardware access control, programming errors detection, etc.) to guarantee the secure

| Table 1. The security evaluation requirements for each of the SOA functional layers (selection) (Kolaczek, 2013). |
|------------------|------------------|
| **SOA layer**    | **Evaluate/verify/test** |
| Business processes | • policy consistency,  |
|                   | • policy completeness, |
|                   | • trust management,   |
|                   | • identity management,|
|                   | • integrity of business process, |
| Composite service | • identification of the services,   |
|                   | • authentication of the services,   |
|                   | • management of security of the composite services, |
| Atomic service    | • description completeness,   |
|                   | • availability,             |
|                   | • protection from attacks,   |
| Communication     | • confidentiality of data transferred, |
|                   | • authentication,           |
|                   | • norms compliance,        |
| Computation       | • availability of computational resources, |
|                   | • protection from attacks,  |
|                   | • integrity of data processed. |
interaction between the system components working on different hardware and software types (Foster, Kesselman, Tsudik, & Tuecke, 1998).

A similar complexity growth of security management process is noticeable when it comes to moving from monolithic systems to those based on service-oriented SOA.

Service-oriented systems provide some user-defined functionalities. Service is defined as a software component that can work independently of the others with an interface for using the implemented functionalities (Papazoglou, Traverso, Dustdar, & Leymann, 2007). SOA is mainly related to the non-functional properties such as the reuse of software components, encapsulation of the system functionality, precisely defined interfaces and flexibility of applications aimed to create the framework for the service composition.

The life cycle of SOA application consists of four phases: modelling, composition, execution and management. These phases apply to all software built with accordance to the SOA paradigm and they are specified by assorted degrees of complexity and peculiar problems, including IT security. Reference model for service-oriented systems defines five layers dedicated to the fundamental system functions, ranging from low-level features associated with the provisioning of communication among the services to the description of the high-level dependencies between the services resulting from the business processes (Figure 1) (Portier, 2007).

As the SOA system can be defined by its five functional layers (Figure 1) such as Business Process, Composite Service, Atomic Service, Communication and Computation,
the corresponding definition of SOA security requirements for the security evaluation process, should address the specific security problems related to each of these layers. Some key issues from the set defining security requirements for the SOA layers have been presented in Table 1.

**Security management in dynamic environments of service-oriented systems**

Complexity of the security-related problems in service-oriented systems is also determined by their parallel persistence; that is those related to the development of distributed heterogeneous systems and confronted with some security issues such as during the processes of modelling, composing, adjusting and service management. Moreover, both protection and evaluation of service-oriented systems security level consider specification of the reference model (Figure 1). Particularly, such an impact on security management process in service-oriented systems can be described by the following requirements (Security, 2008):

- identity management – a service may be executed in different contexts and by disparate users’ groups; the service should always be carried out with all the permissions appropriate for a given execution context (security of business processes)
- proper security controls management – it is necessary to provide all the appropriate mechanisms for security management, both at the level of individual services as well as for composite services (security services)
- security management sovereignty – it is necessary to guarantee the security management mechanisms independent of the underlying technology hardware and software (security service description)
- seamless connection to other organizations on a real-time basis – services can be offered by different service providers; the final functionality delivered to the user is

![Figure 3. The opinion triangle (Josang, 1999).](image-url)
supplied with the resources (services) originating from those different service providers, however, it should not be noticeable to the user (security protocols services).

- protection of data in transit and at rest – execution of composite services may be related to the data flow in systems managed by different entities and using distinguishable security mechanisms; regardless of the place where the data are processed and to what systems they are allocated, it should be possible to manage data security accurately (transport safety).

The opportunity to provide adequate solutions for evaluation of the security level and the trust level affect the following SOA system features:

- a high level of heterogeneity (in both hardware and software),
- enormous dissipation of resources,
- intensity and diversity of communication channels,
- varied granularity of tasks and resources,
- resource sharing,
- competing for resources,
- dynamics of changes and the volatility of resources,
- high-level requirements resulting directly from the specific business processes.

**Related works**

Research carried out in the field of security of heterogeneous service systems referring to the current standards and guidelines defined e.g. by SAML (Lewis & Lewis, 2009) and WS-Security (Satoh & Yamaguchi, 2007) demonstrate the significance of problems resulting from granularity, reuse, autonomy or loose connections between components. Correct implementation of security rules, e.g. in the field of federated identity management in an environment composed of various types of services, is problematic (Kohler, Labitzke, Simon, Nussbaumer, & Hartenstein, 2012). The complexity of the IT system architecture causes that with each additional element new attack vectors appear, which can be a source of various types of fraud, regardless of the application of the recommended security standards. An illustration of this type of problems may be, for example, the use of Universal Discovery Description and Integration (UDDI) registers, or service description files in the WSDL to obtain information enabling a malicious entity to carry out a Denial of Service (DoS) attack (Sidharth & Liu, 2007).

The basic directions of research in the field of IT systems security are works aimed at strengthening preventive elements of security in the system architecture, through the proposal of new security protocols (Pei & Chen, 2011), (Security, 2008), (Nordbotten, 2009), trust management mechanisms (Artz & Gil, 2007), (Aljazzaf, Perry, & Capretz, 2010), and access control mechanisms (Bertino, Martino, Paci, & Squicciarini, 2010). The measures taken so far in this area have proved to be insufficient and have not led to the development of mechanisms that ensure the full protection of the system. Above all, there is still no comprehensive solution that would take into account all the elements of the complex structure of modern IT systems with SOA (Gruschka & Iacono, 2009).
There are no mechanisms that take into account high-level dependencies in the security analysis, which are defined at the business process layer in the SOA architecture reference model. An example of this type of dependency is among others social relations between system users (Carminati, Ferrari, & Perego, 2009), which are reflected in the way the system is used (El-Ramly & Stroulia, 2004), and thus also affect the ability to provide protection and detect security problems.

Another aspect that has been highlighted in research area related to the security of information systems is the dynamics of the structure (Manikrao & Prabhakar, 2005) of modern information systems and the important role of cooperative activities, especially in the context of systems with a SOA. The task of providing security for dynamic service system consists of developing methods for updating and adapting the level of protection to the current state and context of the system’s functioning, and remains largely unsolved (Weerawardhana & Jayatilleke, 2011).

The analysis of system state changes is also an important trend in the field of methods and tools for detecting security problems of the IT system (Oppliger & Rytz, 2005). This problem is related to the task of detecting attacks (Hansman & Hunt, 2005) and fraud (Erbacher, Walker, & Frincke, 2002). For this purpose, the methods of classifying attacks (Halfond, Viegas, & Orso, 2008), defining signatures (Khamphakdee, Benjamas, & Salyod, 2014) and attack patterns (Ning, Cui, & Reeves, 2002) as well as detecting undesirable (Aydin, Zaim, & Ceylan, 2009) and untypical states (Lee & Xiang, 2001) are defined. Published works in this field are dedicated to the analysis of network traffic (Gu, Zhang, & Lee, 2008) and software diagnostics (Payne, 2002). The use of the results of these works, without taking into account additional elements characterizing information systems with a variable structure related to loose coupling between components of the system is increases the risk of incorrect diagnosis of the security level (Fernandez, Washizaki, Yoshioka, & VanHilst, 2010).

The interest in research related to the use of semantic knowledge about the components of the information system environment in order to increase the level of security also gains growing interest (Liu, Xie, Li, Zhang, & Chunming, 2009). A semantic description of the implemented IT system services enables characterization and definition of security requirements for each element of the system (Kim, Luo, & Kang, 2007). However, only a few studies present practical methods of using a semantic representation of knowledge for the analysis and improvement of system security. An interesting approach that provides a good illustration of the potential range of applications for this field of research is the work (Vorobiev & Han, 2006) in which the authors presented the ontology for a selected class of attacks on web services. However, this proposal is limited only to modelling threats to web services (e.g. XML attacks, SOAP attacks).

Incomplete or lack of knowledge about the runtime environment and system users causes that an important task for the protection system is the ability to reflect the level of trust in relation to the selected system elements or users. Most of the work related to trust modelling presents solutions for a formal representation of trust, taking into account the structural dependencies between services (Skogsrud, Motahari-Nezhad, Benatallah, & Casati, 2009), (Zhengping, xiaoli, Guoqing, Min, & Fan, 2007), but without taking into account the factor that is the dynamics of the environment (Blaze 2009). To the best authors knowledge, there are no proposals dedicated to the analysis of the trust management related to the relationship between the business processes defined in the system and the services that implement them. The works appearing in this context focus on
business processes and trust management at the level of system users, not services. An example of an interesting approach in this context may be the work (Skopik, Schall, & Dustdar, 2010) in which the authors proposed a hybrid trust modelling environment for users and services. Nevertheless, the presented approach has been limited mainly to the concept of trust between users (social trust) built in the opinion of user experience, without considering service-level interactions.

**Evaluation of security and trust level**

The ability to determine the level of security is of a great importance throughout the whole process of security management, hence it is monitored at every stage of the system life cycle from the design, through dimensioning, implementation, to the management of the final software product. At the design stage, identified are all the user’s requirements, also from the perspective of the expected security level. Appropriate selection and implementation of protection methods determines the extent to which these requirements will be met or satisfied. At the runtime, the security level is constantly monitored to allow the user a rapid intervention when it comes to a security incident.

Assessment of the trust level in the service-oriented systems is equally important as the ability to estimate the security level. Final security level in service-oriented systems is influenced by the series of security events, not only those related to the direct user-system interaction but also to the interactions between the services, in particular. Consequently, evaluation of the trust level between the services and the users enables to keep control over the access to the system resources such that it minimizes the risk of fraud and security breaches. Furthermore, the providers of the service-oriented systems whose services are delivered from afar to the users, regard the inference about the security level based on the incomplete and often uncertain knowledge. In this context, evaluation of the trust level becomes the fundamental task of the security management system. The trust level is the representation of subjective beliefs about the security level, such that its value can be estimated from the unreliable and incomplete data processed at the specific system state. The security level and the trust level in the computer system are two integral quality aspects of the security management process (Brotby, 2009).

The level of system security is a value defined in the security metric space. A set of elements of the security metric space is determined by the characteristic system states in the context of the specific security requirements.

In the process of security management, there are two separate tasks: providing the required security level and assessment of the current security level.

The task of providing the required level of a computer system security is aimed to reduce the likelihood of the identified risk factors below the predetermined security level system value. It is decomposed into subtasks dedicated to the process of protecting stored, processed and transmitted data by ensuring the integrity, confidentiality and accessibility (Pipkin, 2000).

The goal of assessing the security level is to estimate the current value of the risk associated with the concern of losing the integrity, confidentiality or availability of the system resources despite the applied methods of their protection.

A trust level is defined as an abstract value of the trust, which is reflected in the subject’s belief that the system will operate in accordance with the objectives (Gambetta, 2000).
The subject may be a standalone software component (autonomous agent), or the system’s user. The trust level in the context analysis of the computer security has been introduced to enable the automation of decision-making processes relating to security, e.g. in tasks of an access control, the management rights. The role of estimating the trust level is to define a function on the set of system components (trustee) with associated values of the set trust levels.

The trust level allows the formal representation of the subjective beliefs about the security level of a part, or of the whole system. It is different from the security level in the way that it is defined with reference to the relationship between two entities (a user – a system, a software agent – a system, a service – a service, etc.). The trust level by definition includes subjectivity of the assessment resulting from the limited knowledge or limited analytical capabilities. In contrast, the security level is calculated assuming that the knowledge about the possible threats and the strength of the protection mechanisms applied is complete and reliable (Choo, 2011).

Providing that in the process of assessing the level of the system security, the only available source of knowledge is incomplete and/or uncertain, applicable are usually two possible (potential) scenarios. In the first scenario, the security level is calculated assuming that the incompleteness of knowledge, or its uncertainty, is not affected seriously enough to assess the level of security. In the second scenario, when the first trust level is determined, then the security level is calculated as the value of some function defined over the set of trust levels and the codomain being a set of the security levels.

In the cases of the established system security level and on the basis of the unique requirements of the entities, each of them has associated its own trust level. For example, encryption with an AES algorithm and the encryption key of 128 bits is objectively less secure (lower security level) than an encryption using the same algorithm, but a 256 bits key. However, for two different users with different security requirements and depending on the data type to be protected (e.g. personal data, project documents, administration, private correspondence, photographs of landscapes, etc.), the subjective trust level assigned by the first user using encryption algorithm with the 128 bits long key, may be the same as for the other user using 256 bits encryption key.

The current methods for the security level and the trust level suitable for monolithic systems focus on:

- analysis of the current system configuration based on a fixed set of requirements and metrics, including the guidelines of the applicable standards (e.g. ISO/IEC 15408, ISO/IEC 27000-series, PCI DSS)
- analysis of information about the system states changes, including security-related events, fraud and anomaly detection.

Analysis of the current system configuration starts from defining a set of threats and a set of security countermeasures. Then, defined is a mapping of the security countermeasures in relation to the set of the recognized threats. The final step in the process of the security level evaluation is to conduct a formal verification of the correctness of the implementation of the methods used in order to protect the system.

The security level evaluation which is based on an analysis of the system states changes uses this mapping technique, being the current state of the system as the one of the predefined security levels. The final quality of the methods for the security level evaluation is
dependent on the quality of matching of the selected metric space elements to the specific system states being under the assessment.

So far, the methods used to assess the security level and trust level do not allow to consider such features of service-oriented systems like (Almorsy, Grundy, & Müller, 2016), (Dwivedi & Rath, 2015):

- dynamics of dependencies between the services – the dynamics of service-oriented systems due to the SOA design principles with a loose coupling between the services, which is a fundamental difference from the traditional, predefined methods of data exchange between the systems,
- heterogeneity of runtime environment – services may be provided by different suppliers, so the final level of security and trust level result from quite different security levels of infrastructure and development environment used to implement and maintain the service component,
- diversity of policies – the level of security and trust level of composite service is influenced not only by the hardware and software layers but also by the high-level security policy of individual service providers,
- missing, or incomplete knowledge about the implementation details of services – services in SOA are defined only by their interface, all the other implementation details remain generally unknown to the service recipient.

**Security management in service-oriented systems**

Synthesis of the methods dedicated to solve the problems specifically related to the above security aspects of the service-oriented systems, enabled not only the knowledge integration about the evaluation of both the current security level and the emerging threats, but it also provided some substantive solutions for protecting all the system layers. The method for security management proposed explores an agent-based architecture resulting with more precise evaluation of quality and security of services than so-called classical approaches. The greatest advantage of it, is an ability to analyse both security level and the trust level in the systems-oriented services, also in the cases of missing or incomplete knowledge about the implementation details of the services.

Methods of modelling the trust level aimed at its analysis and evaluation in the service-oriented systems which base on subjective logic give rather subjective assessment results of the security evaluation. The subjectivity of the trust assessment has been modelled by subjective logic opinions of autonomous software agents. These opinions can be processed and thus the trust level is evaluated by modelling the graph structure, the communication links in the system and the standard subjective logic operators.

**Multi-agent system for security level evaluation**

The research objective of this work is to develop a detailed architecture of multi-agent system for assessing security level based on some specific elements that are characteristic for the systems with SOA (Figure 2).

The main components of the architecture proposed for service-oriented systems security level evaluation, are the three sets of software agents. The agents at the lowest level of
the architecture \(\{a_1, a_2, \ldots, a_n\}\) are responsible for data acquisition and for monitoring the execution of the services (e.g. the time of the service completion, the amount of the data transmitted and received by the service, etc.). Middleware agents (transport layer, communication protocols layer, \(\ldots\), business processes layer) are responsible for aggregation and processing data transmitted by the agents from the lowest layer. As a result, we can assess the security level for a corresponding layer from the SOA reference model. At the highest level of the multi-agent architecture developed by us, proposed is the managing agent responsible for providing information about the security level to the external systems working in cooperation with this one. Furthermore, this agent also provides a numerical value of the monitored system overall security level computed on the basis of information about each of the SOA layers security level. In addition to the architecture of multi-agent system, an algorithm for the security level evaluation of the individual system layers in accordance with a multi-layered reference model for service-oriented systems has been proposed.

**Subjective logic as a framework for formal representation of security level**

Subjective logic, which was originally introduced by Josang, was intended to serve as a comprehensive model for reasoning about trust propagation in secure information systems. It is strongly related to Dempster–Shafer’s theory of evidence (Josang, 2001) and binary logic (Jøsang & Grandison, 2003). Subjective logic uses standard logic operators and additionally it introduces two special operators for combining beliefs – consensus and recommendation. The basic definitions of subjective logic given in this section come from (Josang, 1999 and 2001).

Subjective logic is used to express so-called opinions (see below) about the facts regardless of the knowledge on how they were grounded or inferred. In this way, it has a straightforward connection to the intuitive understanding of the trust relationship. As one may also have an opinion about some subject (source of information), Subjective Logic formalism can be applied within a trust-level evaluation architecture.

In a similar way, following the fuzzy logic, when expressing belief about a statement (predicate), either truth or falsehood is assumed with some uncertainty, – one is only able to have an opinion about it (because of our imperfect knowledge). Let’s denote belief, disbelief and uncertainty as \(b\), \(d\) and \(u\), respectively.

**Definition 1.** A tuple \(\omega = \langle b, d, u \rangle\) where \(\langle b, d, u \rangle \in [0,1]^3\) and \(b + d + u = 1\) is called an opinion.

From this definition, the three atomic components of an opinion (belief, disbelief and uncertainty) are dependent upon each other and one of them is redundant. In addition, it is useful to use all the three of them, because this way, we may obtain relatively simple expressions when defining subjective logic operators. When uncertainty equals 0, then the opinion is called dogmatic opinion. Opinions with either a belief value that is equal to zero \((b = 0)\) or disbelief \((d = 0)\) are referred to as absolute opinions, and correspond to TRUE and FALSE propositions of binary logic.

Based on Definition 1 we may also graphically express an opinion as a point belonging to so-called opinion triangle (Figure 3. – dot \(\omega\) is marked as opinion \(\langle 0.8, 0.1, 0.1 \rangle\)).

Opinions have always assigned membership meaning they are expressed by certain software agents. As a result, the opinions are not inherent qualities of objects, but of
judgments about them. For any opinions \( \omega_p = \langle b_p, d_p, u_p \rangle \) and \( \omega_q = \langle b_q, d_q, u_q \rangle \) about the logical value of predicates \( p \) and \( q \) the following operators (equivalent to well-known operators of binary logic) may be defined (Josang, 2001):

**Definition 2.** (Conjunction)

\[
\omega_{p \land q} = \langle b_p b_q, d_p + d_q - d_p d_q, b_p u_q + u_p b_q + u_p u_q \rangle
\] (1)

**Definition 3.** (Disjunction)

\[
\omega_{p \lor q} = \langle b_p + b_q - b_p b_q, d_p d_q, d_p u_q + u_p d_q + u_p u_q \rangle
\] (2)

**Definition 4.** (Negation)

\[
\omega_{\neg p} = \langle b_p, d_p, u_p \rangle
\] (3)

For absolute opinions, both conjunction and disjunction operators give the same results as AND and OR of binary logic. For dogmatic opinions, they return the results of product and co-product of probabilities, respectively.

For example for two agents, \( A \) and \( B \), where \( A \) has an opinion about \( B \), an opinion expressed by agent \( A \) about agent \( B \) is interpreted as an opinion about proposition ‘\( B \)’s opinion is reliable’. An opinion expressed by agent \( B \) about given predicate \( p \), and agent’s \( A \) opinion about \( B \), will be denoted as \( \omega_p^B \) and \( \omega_p^A \), respectively. Assuming that \( \omega_p^B \) and \( \omega_p^A \) are known, the opinion of agent \( A \) about \( p \) is given by discounting operator (also known as reputation operator):

**Definition 5.** (Recommendation, denoted by \( \otimes \))

\[
\omega_p^{A \otimes B} = \omega_p^A \otimes \omega_p^B = \langle b_p^A b_p^B, b_p^A d_p^B, d_p^A u_p^B + u_p^A b_p^B \rangle
\] (4)

It may be proved that recommendation operator is associative, but not commutative. This implies that the order of opinions in recommendation chains is significant. The joint opinion of two agents \( A \) and \( B \) about a given predicate is computed by a consensus operator:

**Definition 6.** (Consensus, denoted by \( \oplus \))

\[
\omega_p^{A \oplus B} = \omega_p^A \oplus \omega_p^B = \langle (b_p^A u_p^B + b_p^B u_p^A)/k, (d_p^A u_p^B + d_p^B u_p^A)/k, u_p^A u_p^B /k \rangle
\] (5)

where \( k = u_p^A + u_p^B - u_p^A u_p^B \)

If a consensus operator is commutative and associative, thus it allows to combine more opinions. It is important that consensus (operator \( \oplus \)) is undefined for dogmatic opinions. This propriety corresponds to real-world observation that there is no joint opinion, if agents are certain about a given fact. However, in this case, some limits exist and may be computed for all the three components of the resulting opinion. Recommendation and Consensus require also assumption that elementary opinions are independent on each other which means that for example, an opinion given by any of the agents cannot be taken into account more than once.
Opinions about binary events can also be expressed as probability expectation $E(\omega_p)$ value for a given opinion. In such a way, the dimensions of the problem can be reduced to a 1-dimensional probability space.

**Definition 7.** (Probability expectation)

$$E(\omega_p) = E((b, d, u)) = b + \frac{u}{2}$$

Another consequence of opinion transformation to the probability expectation value (6) is that probability removes information contained in an opinion – infinitely many opinions with the same values of probability expectation functions $E$ can be expressed.

The opinions may also be ordered. For any two opinions $\omega_1$ and $\omega_2$ about predicate $p$, $\omega_1$ is greater than $\omega_2$ (which can be denoted by $\omega_1 > \omega_2$) if and only if $\omega_1$ expresses stronger belief that $p$ is true. Thus, when ordering such opinions, the following rules (listed by priority) apply:

1. The opinion with the greatest probability expectation $E$ is the greatest.
2. The opinion with the smallest uncertainty is the greatest.

For instance: $\langle 0.5, 0, 0.5 \rangle > \langle 0.4, 0.2, 0.4 \rangle > \langle 0.2, 0, 0.8 \rangle$.

**Evaluating security level in service-oriented systems with monitoring agents and subjective logic**

This section presents briefly the main idea of the proposed solution which can be used to model in a formal way security level using Subjective Logic. Referring to the layered model of SOA architecture (Figure 1), the subjective logic opinion about Business Processes layer can be expressed as the conjunction of following opinions (Table 1):

$$\omega_{pbp} = \omega_{pcon} \odot \omega_{pcom} \odot \omega_{trm} \odot \omega_{idm} \odot \omega_{int}$$

where:

- $\omega_{pcon}$ – subjective logic opinion about policy consistency,
- $\omega_{pcom}$ – subjective logic opinion about policy completeness,
- $\omega_{trm}$ – subjective logic opinion about trust management,
- $\omega_{idm}$ – subjective logic opinion about identity management,
- $\omega_{int}$ – subjective logic opinion about business process integrity.

These opinions can be evaluated in the following way:

$$\omega_{pcon} = \langle b_{pcon}, d_{pcon}, u_{pcon} \rangle$$

where:

- $b_{pcon} = \text{verified}$
- $u_{pcon} = 1 - \text{verified}$
• \( \text{verified} = \begin{cases} 0 & \text{iff the policy consistency has not been verified} \\ 1 & \text{iff the policy consistency has been verified} \end{cases} \)

\[
\omega_{pcom} = \langle b_{pcom}, d_{pcom}, u_{pcom} \rangle
\]

where:

• \( b_{pcom} = \text{verified} \)
• \( u_{pcom} = 1 - \text{verified} \)

• \( \text{verified} = \begin{cases} 0 & \text{iff the policy completeness has not been verified} \\ 1 & \text{iff the policy completeness has been verified} \end{cases} \)

\[
\omega_{trm} = \langle b_{trm}, d_{trm}, u_{trm} \rangle
\]

where:

• \( b_{trm} = \text{exist} \cdot \text{satisfaction level} \)
• \( u_{trm} = 1 - \text{exist} \)

• \( \text{exist} = \begin{cases} 0 & \text{iff the trust management has not been applied at this SOA layer} \\ 1 & \text{iff the trust management has been applied at this SOA layer} \end{cases} \)

• \( \text{satisfaction} \subseteq \langle 0, 1 \rangle \) - represents the user satisfaction level regarding the trust management method.

\[
\omega_{idm} = \langle b_{idm}, d_{idm}, u_{idm} \rangle
\]

where:

• \( b_{idm} = \text{exist} \cdot (\text{user_control} + \text{minimal_disclosure} + \text{directed_identity} + \text{decentralized} + \text{bidirectional} + \text{single_logout} + \text{scalability}) / 7 \)
• \( u_{idm} = 1 - \text{exist} \)

• \( \text{exist} = \begin{cases} 0 & \text{iff the identity management has not been applied} \\ 1 & \text{iff the identity management has been applied} \end{cases} \)

All the other variables used in \( b_{idm} \) evaluation have the values taken from the binary set \([0,1]\) where value 1 means that the identity management implements the particular functionality (e.g. directed identity management, user control, etc.).

\[
\omega_{idt} = \langle b_{idt}, d_{idt}, u_{idt} \rangle
\]

where:

• \( b_{idt} = \text{normalized}(\text{Moran} - I) \)
• \( u_{idt} = \text{normalized}(\text{Variance}_\text{Moran} - I) \)

The evaluation of the subjective logic opinion for all the subsequent SOA layers is performed in a similar way.
Security analysis at business processes level

In the previous section, some of the proposed methods for evaluation of the security level were introduced which considers characteristics of each SOA reference model layer (Shah & Patel, 2008), (Kolaczek, 2012). One of our methods allows to evaluate the security level based on the data obtained from the lower layers (transportation, communication protocol in Figure 1) and high-level dependencies between the services, including those in the business processes layer. The data illustrating the low-level characteristic of the SOA system may be referred to a network traffic generated during the information exchange between the services. The metadata, which describes high-level dependencies, define the existing links between the services – composite service structure (Figure 4).

We show that for a given composite service, there is a significant mutual correlation between the spatial data dependencies (incoming and outgoing data for some individual service components).

The results presented in Table 2. indicate a high spatial correlation in the ring-based networks (which here reflect type of data flow in an exemplary business process) while the random communication topology destroys this type of correlation Table 3.

Additionally, in the ring-type network, the most significant results (the greatest spatial correlation) have been obtained for the nodes with the highest degree (with additional links) while in the random network, the highest Local Moran test values go to the nodes with the lowest degree.

The values of Moran-I test range from $-1$ to $+1$, where the values significantly below $-1/(N-1)$ indicate negative spatial autocorrelation while the values significantly above $-1/(N-1)$ indicate positive spatial autocorrelation. As a result, the Moran-I test could indicate the presented trust level evaluation framework as the value expressing the level of belief in the business process integrity. The Moran-I value will be normalized furthermore, to fit the domain of subjective logic values to $<0,1>$.

Also, we indicate that any disturbances in the business process or the implementation of the composite service, such as repeated calling of the selected service, disturbed sequence of the service execution, etc., significantly impede the spatial autocorrelation parameter value (Figure 5).

Our study highlights the importance of the measurement of spatial autocorrelation for the security level evaluation of the composite services in service-oriented systems. More
generally, the spatial autocorrelation empowers the system to fraud detection by implementation of the business process logic (Kołaczek et al., 2015).

Our original contribution to this work is development and an experimental evaluation of the approach to estimation of the SOA systems security level. In order to evaluate the security level, we apply statistical analysis of spatial data. In this approach, we draw an analogy between the services that implement a business process and any entities that have a fixed spatial structure with the specific values assigned to it, such as e.g. the distance between the selected pair of elements.

Discussion

The proposed approach is unique as far as it gives the advantage of formal and precise analysis of the service-oriented systems security level. The formal structure provided by Subjective Logic allows for precise expression of the security level of the given element of the SOA system. In this context ‘precise expression’ means that it reflects both firm and well-defined security dependencies (e.g. length of the cryptographic key vs. secrecy of the message) and some less precisely defined requirements (e.g. the value of the protected information). Finally, the proposed method also allows for expression our lack of the knowledge about the security of the system and its components (services).

In addition, it allows drawing inferences about the changeability in the security level of the composite services depending on the component services security sublevels. The subjective logic, which is the basis for processing knowledge about the security levels, also, it gives advantages from the knowledge integration coming from different resources.

In Service-Oriented systems, this multi-perspective is a crucial part characterizing the process of security level evaluation. The services are defined and executed using different languages, different platforms, and resources. There are also many various sources of information describing their static and dynamic proprieties (log files, RDF files, etc.). This means, that to be able to evaluate the final security level all these sources of information should be taken into account. This is a possibility is provided by the proposed method.

Table 2. Moran-I test results for examined topologies.

| Network topology | Ring | Ring + 1 link | Ring + 2 links | Random |
|------------------|------|--------------|----------------|--------|
| Moran I statistic| 0.621917 | 0.705009 | 0.266890 | −0.06560 |
| p-value          | 0.000787 | 9.222e−05 | 0.05888 | 0.6412 |
| Expectation      | −0.052631 | −0.052631 | −0.052631 | −0.03448 |
| Variance         | 0.045547 | 0.041050 | 0.041725 | 0.018990 |

Table 3. Local Moran-I test results summary.

| Network topology | Ring | Ring + 1 link | Ring + 2 links | Random |
|------------------|------|--------------|----------------|--------|
| Median(Ii)       | 0.26489 | 0.4809227 | 0.19972 | −0.05212 |
| Mean(Ii)         | 0.62192 | 0.7050096 | 0.26689 | −0.06561 |
| Mean(Pr(z > 0))  | 2.870e−01 | 2.276e−01 | 0.33687 | 0.5209676 |
Figure 5. Correlation of spatial data communication traffic generated for the four selected structures of complex service. (a) ring network correlogram, (b) ring +1 network correlogram, (c) ring +2 network correlogram and (d) random network correlogram.
This type of reasoning about security level resolves some conflicts arising to possible inconsistent opinions provided by the agents on the security level of the same entity. Contradictory information about the security level can also appear while processing data from independent information sources (e.g. software agents). Uncertainty about the value of the data acquired is an important aspect of the proposed method for assessing the security level of service-oriented systems, because SOA environment usually deals with a number of independent service providers and independent information sources about the security status that may be either not to be trusted or unreliable, or imprecise (Kolaczez & Juszczyszyn, 2010).

Innovative features of our approach are multi-agent system architecture for assessment of the service-oriented systems security level, the method for estimating security level in the system individual layers, the method for integration of the security level values from different agents and the method of estimating the security level of the composite services.

Conclusions

Evaluation of the security level and the trust level in service-oriented systems require not only adaptation of the methods used for monolithic systems, but also the new methods which consider different characteristics of service-oriented systems. In particular, observed is a strong demand for the methods of analysis and identification of unexpected events, problem analysis of the traffic related to service execution and finally, the working standards for typical threats detection, such as an attack on the access to the peculiar service (called Denial of Service). These newly developed methods count on the specific nature of the risk and security requirements referred to different layers of the reference model for service-oriented systems.

These new solutions allow to carry out a comprehensive and critical evaluation of the security level and the trust level based on the information about the mutual relations between the security events observed at a low-level description of the system’s state (e.g. the traffic generated by the service) in relation to the events that represent high-level functions provided by the system (e.g. the implementation of business processes).

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Grzegorz Kołaczez is an Professor at the Department of Computer Science, Wroclaw University of Technology. He received Ph.D. degree in Computer Science in 2001 and D.Sc. in 2016 from the Faculty of Computer Science and Management at Wroclaw University of Science and Technology. His major research interests are security and reliability of computer systems, multiagent systems, formal logic, semantic and social network security related problems, information and knowledge management. He is currently working on the problem of security related issues of Service Oriented Architecture and Service Oriented Knowledge Utility. His teaching includes computer and network security and computer architecture. He is an author and co-author of more than 90 scientific publications (including journal articles, book chapters and conference papers). Grzegorz took part in several research projects related to network analysis and service oriented systems and he is also
an evaluator of many national research and infrastructural project and a member of program com-
ities of several international scientific conferences.

Jolanta Mizera-Pietraszko holds a position of an Assistant Professor at the Faculty of Mathematics,
Physics and Computer Science, Opole University, Poland. Her research interests are mainly
focused on Information Technologies, Data Mining, Computational Linguistics, Natural Language
Processing, Web Information Retrieval, Machine Translation, Artificial Intelligence, e-Learning, Elec-
tronic Textbooks, Multilingual Search Technologies, Parallel Languages, Bi-text processing, Multilin-
gual Question-Answering Systems, and Multilingual Digital Libraries. She has given tutorials at
conferences in Finland, Cyprus and the UK, London where she was honored with Best Tutorial
Award for her tutorial entitled Language Parallels aimed at construction of a query profile. Also,
she gave an invited lecture at Ionian University, Corfu, Greece. She has recently been invited to
serve on international program committees of conferences in the UK, Czech Republic, India, and
Poland. Her projects have received recognition from the university, the European Union, and scien-
tific institutions abroad.

References

Aljazzaf, Z. M., Perry, M., & Capretz, M. A. M. (2010). Trust in web services. In 2010 6th World Congress
on Services (pp. 189–190). IEEE.

Almorsy, M., Grundy, J., & Müller, I. (2016). An analysis of the cloud computing security problem. arXiv
preprint arXiv:1609.01107.

Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the semantic web. Web Semantics:
Science, Services and Agents on the World Wide Web, 5(2), 58–71.

Aydin, M. A., Zaim, a. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for com-
puter network security. Computers and Electrical Engineering, 35(3), 517–526.

Benson, G. S., Akyildiz, I. F., & Appelbe, W. F. (1990). A formal protection model of security in centra-
lized, parallel, and distributed systems. ACM Transactions on Computer Systems, 8(3), 183–213.

Bertino, E., Martino, L. D., Paci, F., & Squicciarini, A. C. (2010). Security for web services and service-
oriented architectures. Security for Web Services and Service-Oriented Architectures, 54(2), 45–77.

Blaze, M., Kannan, S., Lee, I., Sokolsky, O., Smith, J. M., Keromytis, A. D., & Lee, W. (2009). Dynamic trust
management. Computer, 42(2), 44–52.

Brotby, K. (2009). Information Security Governance.

Carminati, B., Ferrari, E., & Perego, A. (2009). Enforcing access control in web-based social networks.
ACM Transactions on Information and System Security, 13(1), 1–38.

Chen, Y., Paxson, V., & Katz, R. H. (2010). What’s new about cloud computing security? University of
California, Berkeley Report No. UCB/EECS-2010–5 January, 20(2010), 1–8.

Chneider, D. (2012). The state of network security. Network Security, 2012(2), 14–20.

Choo, K.-K. R. (2011). The cyber threat landscape: Challenges and future research directions.
Computers & Security, 30(8), 719–731.

Conti, M., Chong, S., Fdida, S., Jia, W., Karl, H., Lin, Y. D., & Zukerman, M. (2011). Research challenges
towards the future internet. Computer Communications, 34(18), 2115–2134.

Dlamini, M. T., Eloff, J. H. P., & Eloff, M. M. (2009). Information security: The moving target. Computers &
Security, 28(3–4), 189–198.

Dwiuedi, A. K., & Rath, S. K. (2015). Incorporating security features in service-oriented architecture
using security patterns. ACM SIGSOFT Software Engineering Notes, 40(1), 1–6.

El-Ramly, M., & Stroulias, E. (2004). Mining system-user interaction logs for interaction patterns. Msr, 1–
5.

Erbacher, R. F., Walker, K. L., & Frincke, D. a. (2002). Intrusion and misuse detection in large-scale
systems. IEEE Computer Graphics and Applications, 22(1), 38–47.

Fernandez, E. B., Washizaki, H., Yoshioka, N., & VanHilst, M. (2010). Measuring the level of security
introduced by security patterns. ARES 2010 – 5th International Conference on Availability, Reliability,
and Security, 565–568.
Foster, I., Kesselman, C., Tsudik, G., & Tuecke, S. (1998). A security architecture for computational grids. In Proceedings of the 5th ACM conference on Computer and communications security – CCS ’98 (pp. 83–92). New York, New York, USA: ACM Press.

Gambetta, D. (2000). Can we trust trust? In D. Gambetta (Ed.), Trust: Making and breaking cooperative relations (pp. 213–237). Oxford: University of Oxford.

Gruschka, N., & Iacono, L. L. (2009). Vulnerable cloud: SOAP message security validation revisited. 2009 IEEE International Conference on Web Services, ICWS 2009, 625–631.

Gu, G., Zhang, J., & Lee, W. (2008). BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. Proceedings of the 15th Annual Network and Distributed System Security Symposium, 53(1), 1–13.

Halfond, W. G. J., Viegas, J., & Orso, A. (2008). A classification of SQL injection attacks and countermeasures. Preventing SQL Code Injection By Combining Static and Runtime Analysis, 1, 13–15.

Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. Computers and Security, 24(1), 31–43.

Harris, B., & Hunt, R. (1999). TCP/IP security threats and attack methods. Computer Communications, 22(10), 885–897.

Josang, A. (1999). Trust-based decision making for electronic transactions. Proceedings of the Fourth Nordic Workshop on Secure …. Retrieved from http://folk.uio.no/josang/papers/Jos1999-NordSec.pdfTrust-based decision making for electronic transactions. Proceedings of the Fourth Nordic Workshop on Secure

Josang, A. (2001). A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. doi:10.1142/S0218488501000831

Josang, A., & Grandison, T. (2003). Conditional inference in subjective logic. In Proceedings of the 6th International Conference on Information Fusion, FUSION 2003. doi:10.1109/ICIF.2003.177484

Khamphakdee, N., Benjamas, N., & Saiyod, S. (2014). Improving intrusion detection system based on snort rules for network probe attack detection. 2014 2nd International Conference on Information and Communication Technology (ICoICT), 69–74.

Kim, A., Luo, J., & Kang, M. (2007). Security ontology to facilitate web service description and discovery. Security, 167–195.

Kohler, J., Labitzke, S., Simon, M., Nussbaumer, M., & Hartenstein, H. (2012). FACIUS: An Easy-to-Deploy SAML-based Approach to Federate Non Web-Based Services. In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 557–564).

Kolaczek, G. (2012). Spatial analysis based method for detection of data traffic problems in computer networks. In Uncertainty Modeling in Knowledge Engineering and Decision Making (pp. 919–924).

Kolaczek, G. (2013). Multi-agent platform for security level evaluation of information and communication services. In Advanced Methods for Computational Collective Intelligence (pp. 107–116). Springer Berlin Heidelberg.

Kolaczek, G., & Juszczyszyn, K. (2010). Smart security assessment of composed Web services. Cybernetics and Systems: An International Journal, 41(1), 46–61.

Kolaczek, G., Juszczyszyn, K., Świątek, P., Grzech, A., Schauer, P., Stelmach, P., & Falas, Ł. (2015). Trust-based security-level evaluation method for dynamic service-oriented environments. Concurrency and Computation: Practice and Experience, 27(18), 5700–5718.

Kolaczek, G., & Mizera-Pietraszko, J. (2017). Analysis of dynamic service oriented systems for security related problems detection. In INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on (pp. 472–477). IEEE. Gdanski.

Lee, W., & Xiang, D. (2001). Information-theoretic measures for anomaly detection. Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001, 130–143.

Lewis, K. D., & Lewis, J. E. (2009). Web single sign-on authentication using SAML. Journal of Computer Science, 2, 41–48.

Liu, M., Xie, D., Li, P., Zhang, X., & Chunming, T. (2009). Semantic access control for web services. In Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC ‘09. International Conference on (Vol. 2, pp. 55–58).
Manikrao, U. S., & Prabhakar, T. V. (2005). Dynamic selection of web services with recommendation system. Proceedings – International Conference on Next Generation Web Services Practices, NWeSP 2005, 2005, 117–121.

Ning, P., Cui, Y., & Reeves, D. S. (2002). Constructing attack scenarios through correlation of intrusion alerts. Proceedings of the 9th ACM Conference on Computer and Communications Security, pp, 10.

Nordbotten, N. (2009). XML and web services security standards. IEEE Communications Surveys & Tutorials, 11(3), 4–21.

Oppliger, R., & Rytz, R. (2005). Does trusted computing remedy computer security problems? IEEE Security and Privacy, 3, 16–19.

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. Computer, 40(11), 38–45.

Payne, C. (2002). On the security of open source software. Information Systems Journal, 12(1), 61–78.

Pei, S., & Chen, D. (2011). Research of SOAP message security model on Web services. In S. Lin & X. Huang (Eds.), Advanced research on computer education, simulation and modeling, Pt I (Vol. 175, pp. 98–104). Berlin: Springer-Verlag.

Pipkin, D. L. (2000). Information security: protecting the global enterprise.

Portier, B. (2007). SOA terminology overview.

Rahman, N. H. A., & Choo, K.-K. R. (2014). A survey of information security incident handling in the cloud. Computers & Security, 49, 45–69.

Satoh, F., & Yamaguchi, Y. (2007). Generic security policy transformation framework for WS-Security. IEEE International Conference on Web Services (ICWS 2007).

Security, S. O. A. (2008). SOA Security. Information Sciences.

Shah, D., & Patel, D. (2008). Dynamic ubiquitous security architecture for global SOA. Proceedings of The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Valencia, Spain, 483–487.

Sidharth, N., & Liu, J. (2007). A Framework for enhancing web services security. In Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International (Vol. 1, pp. 23–30).

Skogsrud, H., Motahari-Nezhad, H., Benatallah, B., & Casati, F. (2009). Modeling trust negotiation for web services. Computer, 42(2), 54–61.

Skopik, F., Schall, D., & Dustdar, S. (2010). Modeling and mining of dynamic trust in complex service-oriented systems. Information Systems, 35(7), 735–757.

Vorobiev, A., & Han, J. (2006). Security attack ontology for Web services. In 2006 2nd International Conference on Semantics Knowledge and Grid, SKG (pp. 42–47). IEEE Computer Society.

Weerawardhana, S. S., & Jayatilleke, G. B. (2011). Web service based model for inter-agent communication in multi-agent systems: A case study. In Hybrid Intelligent Systems (HIS), 2011 11th International Conference on (pp. 698–703).

Zhengping, L., xiaoli, L., Guoqing, W., Min, Y., & Fan, Z. (2007). A formal framework for trust management of service-oriented Systems. In IEEE International Conference on Service-Oriented Computing and Applications, 2007. SOCA ’07 (pp. 241–248).