AN INTELLIGENCE INFORMATION SYSTEM BASED ON SERVICE-ORIENTED ARCHITECTURE: A SURVEY OF SECURITY ISSUES

Jugoslav ACHKOSKI, Vladimir TRAJKOVIK
and Metodija DOJCHINOVSKI

Abstract: Security is an important requirement for a service-oriented architecture (SOA), since SOA in principle considers services spread widely on different locations and diverse operational platforms. The main challenge for SOA security still drifts around 'clouds' and there is still a lack of suitable frameworks for security models based on consistent and convenient methods. In this paper, we propose security solutions for an Intelligence Information System completely based on SOA. Contemporary security architectures and security protocols are still evolving. SOA-based systems are characterized with differences in security implementation as encryption, access control, security monitoring, security management through disparate domains etc. Domains have services as endpoints in the information systems, which usually form composite services. The workflow which is established through composite services is extending on different endpoints in different domains. The paper's main aim is to provide a contribution in developing suitable security solutions to Intelligence Information Systems using web service security standards in order to reach appropriate level of information security considering authentication, authorization, privacy, integrity, trust, federated identities, confidentiality and more. The paper reflects an approach in which useful information provided by the services is sent out directly from the creators of information to the consumers of information. We introduce security and logging system that can be used as verification and validation middleware.

Keywords: SOA, authentication, authorization, auditing, access control, model, security solution, XML encryption, security assertion markup language, signature, federated identity confidentiality, integrity.

1. Introduction

Intelligence as a service has a great significance for any country. An information system for support of intelligence activities should be used on daily basis and has a great influence on senior decision making processes. The use of modern information tech-

nology contributes greatly for improvement of the process (activities) supporting intelligence cycles (planning, collecting, analyzing and dissemination). Although there is constant improvement as a result of the progress in the area of information technology, significant difference in the quality of work in the field of intelligence has not taken place in the last ten years.

The implementation of Service Oriented Architecture (SOA), i.e. the usage of SOA provides new opportunities in a wider range of solutions for designing intelligence information systems, regarding the more efficient management of information, as well as their use by the end users for whom they are intended. In order to keep up with the pace of organisational and technological development, planning on short, medium and long term is needed for development of information systems in support of intelligence, as it relates to IT development. The SOA approach in information systems is a logical solution, not only for a temporary and short term usage but as a perspective solution for general strategy in companies and governmental institutions. To achieve such SOA based systems, one should consider security requirements and goals at the certain level already in the process of their planning and creation.

This paper is organized further as follows. Section 2 presents security solutions for protecting SOA based information systems. The goals of SOA security are elaborated in Section 3. Section 4 provides a description of frequently used security protocols for web services such as XML, XML encryption, XML signature, SAML, SOAP and other standards within the WS-Security family. A model of security solution for an intelligence information system based on SOA and its implementation are presented in Section 5.

2. Related Work

Available studies project that 90 percent of external attacks on applications will be related to security vulnerabilities and mis-configured systems.[1] Because it is not possible to develop 100% secure applications, practical approaches prescribe to analyse threats, vulnerabilities, risks and then implement security mechanisms for SOA based systems. Solutions seek to improve security through the entire system, thus contributing to the decrease of incident response costs, application outage costs, costs of fixing a malfunctioning system, reputation damage costs, etc. Towards this purpose the paper outlines directions for implementing security integration and access control in SOA and Web service-oriented architecture (WSOA) initiatives, addressing the following topics: access control models, a meta-model for WSOA, goals of SOA Security, SOA security implementation models, industry standards for SOA security and Service-Oriented Information Integration (SOII).

Securing service-oriented systems is a specific challenge, since security services are equally distributed as a workflow services in SOA based systems.[2] Establishing security only on endpoints is not an adequate security solution for SOA systems. On the other hand, implementing security services on each endpoint would be an expensive solution. Currently, there is little work done to separate security from endpoints of services. As one solution the model of Security As A Service (SAAS) is indicated which, however, increases the security burden on the endpoints by using shared security services within the security domain. Security services are composed of integrated components based on Service Component Architecture (SCA) models. In this paper the SAAS paradigm is used; it is implemented in securing the SECTISSIMO platform. A referent security architecture for protecting critical SOA systems based on the SAAS paradigm is also presented.

In 2005, Hutchison, Hinton and Hondo from IBM introduced the SAAS approach and proposed a Security Decision Service (SDS), which provides service-based PDP to multiple enforcement points.[3] A more comprehensive treatment addresses implementation aspects of authentication, trust and secure conversation as separate services to solve security manageability and interoperability problems.[4] Another study takes a security-specific view on SOS architectures.[5]

3. Security and Access Control

Considering the big picture for SOA security, it is important to understand different aspects, such as the role of AAA—Authentication, Authorization and Auditing—in SOA security and their implementation as industry standards. Special attention needs to be given to web service security because web services are used widely in implementing SOA paradigm.[6]

Since the usage of SOA is increasing, the limitations of using services are decreasing and it becomes a comprehensive venue for using myriads of applications. In order to reach real reusability of services, organizations should give access to the services to third parties, partners and end-users through unsecured networks such as Internet. Services are organizational property and without proper security measurements the level of risk for the organization increases, e.g. as a result of possible unauthorized access, misuse of services, overuse of services and hacker vulnerability.[7]

Security systems should afford business application to fulfil necessary users' requirements in order to reach security goals: authentication, authorization, federated identity, privacy, integrity, accessibility, non-repudiation in terms of sending and receiving messages to users.

Authentication is the validation of checking subject identity. 'Subject' can be a user, web service, computer or application.[8] Authentication is the first step in access con-

trol. In order to provide access control on certain level it is necessary that the system identifies the subject and an appropriate level of trust is provided for authentication of the subject identity. The mutual authentication represents two-way authentication and allows proving the identity of both parties involved in communication.

There are different mechanisms for authentication, but most used are following:

- user name and password;
- digital certificate;
- biometrical devices.

Common protocols for encryption are SSL (Secure Sockets Layer)/TLS (Transport Layer Security). These protocols have embedded mechanisms for authentication as checking user name and password and checking validation of digital certificate. Whatever the level of security, it can be used for authentication only for two subjects simultaneously. Most importantly, the last component of the processes is to accept identity of the initial component which means that SOA based systems can use SSL protocols, but it should be upgraded with other security mechanisms.

Figure 1 provides an example of a user who authenticates on a portal and sends a request for service. In this example the method sender-vouch is used, which means that the portal guarantees for user's identity. This is one of the methods for identity propagation that is used in Web Service Security (WS-S). The use of Single Sign-On

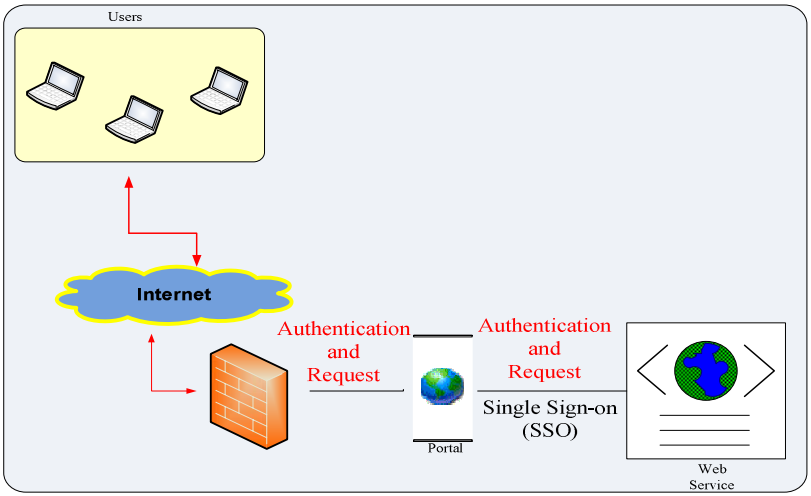

Figure 1: Simple example of identity propagation.[9]

(SSO) method affords identity propagation from application to the services. The portal should provide high level of guarantee in order to establish reliance between services and users. That means that services have to trust the authentication of the portal and its accuracy because it gives a guarantee. These two types of ensuring trust should be considered separately.[10]

Authorization is determining the right of access and permission to the users. After authentication and identity checking, the system should determine user access rights and resources which can be used by the user.[11] Authorization can apply two types of access control:

- Discretionary Access Control (DAC);
- Mandatory Access Control (MAC).

DAC forbids access control based on permission, roles, attributes and group where subject belongs. Permissions to the system resources should be centralized when a subject requests access and PDP decides for access rights. This type of access control is generally used in commercial SOA systems. MAC is commonly used in government SOA systems and is used to control access rights related to security permissions and security symbols of resources. It means that data have to have security symbols and PEP points must determine user's access rights *vis-à-vis* requested resources by comparisons between security symbols and approved access rights for subjects.

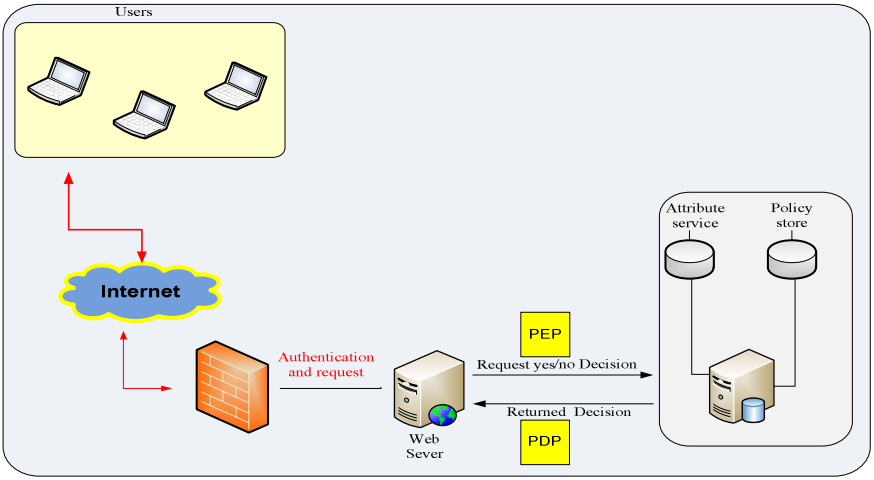

Figure 2: Using PEP/ PDP in the process of authorization.[12]

Role-Based Access Control (RBAC) is a regular method used in DAC. Security roles are defined and given to the subjects and user's rights are defined with security roles. PDP decides taking in account user's rights and roles of subjects. *Attribute-Based Access Control* (ABAC) is a similar method, but it uses security attributes instead of security roles. In the systems like this, authorization rights are roles, groups, security approvals and they are presented as attributes. ABAC is used in DAC and MAC methods for access control. *Predetermined Authorization Decision-Based Access Control* (PADBAC) is a strategy for access control and it uses DAC and MAC methods for access control which means that user's rights are presented as cards and they will be used for accessing the resources.

Federated identity. The purpose of federated identity is to enable users access to several security domains using unique identification.[13] Security domains are components in different companies with their own resources. The sum of partner companies is called a federation and users can access different domains through the SSO method. Also, the method of access in federated identity is determined through assessment of user validation and authorization methods. Accordingly, companies should rely on external identity; they do not need to create own local identity for each user separately who has accessed company resources as an external user. This type of approach decreases companies' costs for managing security system without possible consequences.

The successful implementation of a federated identity depends of the capabilities for dividing shared sum of request between companies in the federation. The idea of federated identity evolves and spreads its influence to other information besides identity. There are two protocols for access in federated environment:

- Browser-based SSO, used in browser client and web application through HTTP (Hypertext Transfer Protocol);
- Service-based SSO, used between two services.

Here, dynamic relations as features in a federation are important. Even though federated identity is a long-term solution, requirements for flexible models in dynamic businesses are more important at current. All companies developing architectures should establish strong coordination between business processes. Besides, protection of privacy and confidential information place further requirements.

In 2005, the Organization for the Advancement of Structured Information Standards (OASIS) published the Security Assertion Markup Language (SAML 2.0) standard, fastening together Shibboleth and ID-FF (Liberty Alliance Identity Federation Framework) standards which refer to the federated identity. Also, the Liberty Alli-

ance published ID-WSF (Identity Web Services Framework) which is based on SAML 2.0. These two standards are most used in modelling federated identity.

Confidentiality. Each communication established between authorized stakeholders should be protected. 'Privacy of information' means that information flows through a secure environment.[14] Privacy is achieved with encryption on sensitive information. In the process of encryption regular text messages are encrypted using cryptographic algorithms; an encrypted message is received as result. There are different cryptographic algorithms and they can be symmetric algorithms using secret keys and asymmetric algorithms using public keys.

In business SOA environment can provide for flows of classified information implementing an appropriate level of encryption.[15] Various protocols make bulk encryption between two points possible. Also, there are SSL/TLS protocols. However, the use of these protocols is limited because some SOA systems do not afford bulk encryption. SSL/TLS protocols afford protected communication only between two points, but each point on the communication path between sender and message recipient can use that information.

The following issues have to be considered in searching for solutions to confidentiality requirements:

- key management – ways of distributing keys;
- ciphers to be used;
- cryptographic protocols that provide these services;
- the amount of encryption necessary to meet enterprise security requirements.

Using secret keys for encrypting information is challenging. Cryptography with public keys is used for key negotiation – the establishment of secret keys used for data encryption.

Standards such as SSL provide opportunities for establishing keys which are used in long HTTP sessions. Messages are exchanged between two parties trough an equivalent key which provides message confidentiality for the duration of the session.

Standards such as WS-Security SOAP messaging [16] do not support the session concept and it is a challenge when it is necessary to establish long message exchanges. As a consequence, each message should be encrypted with public key for determining reliability of secret key. As a result, a combination between WS-Security SOAP messaging and the SSL standard is used for communication between two points. In 2007, OASIS published a standard called WS-Secure Conversation which is used for establishing long sessions using the WS-trust model.[17]

Integrity. In communication between parties, especially in service transaction, it is necessary to establish control mechanisms in order to check whether data has been changed or interfered. A number of techniques can be used for validating data integrity, i.e. to check whether data have been altered by unauthorized subject. Because of possibility for interference, messages which flow through TCP/IP networks should use digital signatures, Message Authentication Codes, or hash algorithms to validate the integrity of the data.

Additional mechanisms are needed to provide validation of message integrity between each service consumer and service provider in the SOA environment. Message end-user has to have high level of confidence that the message is not altered by the other user in communication. Besides, message end-user should have trust that message has valid integrity and it is not replicated from a third party which can interfere. Therefore, security protocols for messages use security mechanisms for integrity to combine messages with a time stamp and message identifier. In that manner time and data, identity and the message itself are guaranteed. If the integrity of time identifier is not valid or it has expired and message identity is not valid or message is changed the 'consumer' automatically rejects that message. Mechanisms such as WS-Security SOAP messaging specification can provide for message integrity.

The SSL/TLS protocol provides integrity between two points and it combines them with mechanisms for authentication and confidentiality. However, this protocol does not provide appropriate level of security in SOA systems. The XML Signature standard can be used in additional to previously mentioned standards. A unique request is digitally signed and transmitted, thus providing for message integrity. If data is changed in any moment of transmission, the security clearance for data integrity will be negative. Also, XML signature is used to prevent resending the same message.

As a conclusion for SOA security related to integrity it can be concluded that in planning process to build SOA enterprise architects should take into consideration all scenarios which refer to usage of services.

Non-repudiation. Non-repudiation emerges as a consequence of digital message signature. It presents approval that a subject digitally signed message. In cryptography, a digital signature joints signer identity and the content of the signed message. Using cryptography with public keys allows non-repudiation of the sender of a message who signed the message.

Using the XML Signature standard for signing a message, or certain elements of a message, affords to prove integrity and non-repudiation. This standard can be used for services based on REST and WS-Security SOAP Messaging standards.

Although digital signature gives high level of security it is important that implementation of solutions for sending messages within SOA systems are used carefully due to associated potential security gaps and vulnerabilities. If an XML element is approved without fulfilling certain conditions for its usage in SOA systems then the level of security decreases. Limitations and conditions for using signed data need to be underlined. When identity identification is necessary, a cryptographic connection should be provided, and under certain conditions upgraded with digital signature. That means that signature, besides data should include content for their usage and limitations related to date and time. Also, usage of digital signature in SOA systems leads to decreasing security because due to a variety of threats. Security systems which are based on SOAP standards provide an environment that alleviates security threats to information systems.

4. Security Standards and Specification for Web Services

Web services and Web services security are based on several standards briefly presented here in order to explain the selection of an appropriate solution for security in information systems based on service-oriented architecture.

eXtensible Markup Language. XML is a basis for web services and web service security standards used for their construction. This is an unclassified standard approved by World Wide Web Consortium (W3C) as a method for exchanging data in text-based format. Providing opportunities for implementation in heterogeneous systems, it is conveniently used for creating web services and SOA where users use different platform.

Simple Object Access Protocol. SOAP is an important protocol for transmitting messages while it forms a basis for web services protocols. SOAP messages [18] are designed to be independent from transport protocols. They are transmitted through HTTP or HTTPS. SOAP message are not tied to the HTTP protocol; on the other hand, they can be used in a line of message sent through e-mail or another transport mechanism.

The SOAP standard is based on XML. It defines a structure of the message transmitted through the systems. Messages defined in SOAP have an envelope, a header and a body. The SOAP header allows to embed cryptographic elements as a digital signature within message. Although there are no limitations on embedding security elements in the message header, they are used in WS-S standards to transmit security information in the message (Figure 3).

SOAP uses two basic modes for message transmission: document mode and remote procedure call (RPC) mode. The document mode is convenient for one-way message transmission, when a user sends a message and do not expect a response. The RPC

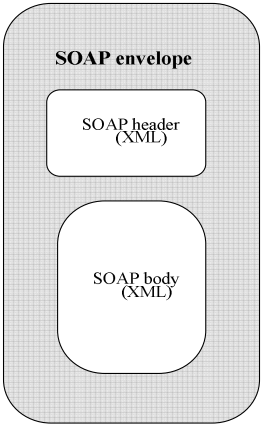

Figure 3: An SOAP message.[19]

mode is commonly used; it is based on a request-response model where the user sends a SOAP message and expects a SOAP response.

XML Signature. XML Signature provides integrity and authentication for XML data using digital signature; it can be used in any digital content. Basic usage of XML Signature within Web service security is to provide integrity for digital signature on XML message and to prove signer's identity (Figure 4).

```
<Signature ID?
<SignedInfo>
<CanonicalizationMethod/>
<SignatureMethod/>
(<Reference URI? >
 (<Transforms>)?
  <DigestMethod>
  <DigestValue>
</Reference>)+
</SignedInfo>
<SignatureValue>
(<KeyInfo>)?
(<Object ID?>)*
</Signature>

  ?  =  Zero or More Occurrence
  +  =  One or More Occurrences
  *  =  Zero or More Occurrences
```

Figure 4: Informal XML signature syntax.[20]

XML Signature is presenting itself as a XML. XML Signature consists of the following elements (see Figure 5 for an example):

- Element that identify digital signature;
- SignedInfo consists of references for data; it determines whether data are digitally signed;
- CanonicalizationMethod refers to manner which element SignedInfo is prepared before signature is calculated. The reason for that is that different platforms can interpreted data in a different way (e.g., carriage returns <CR> versus carriage return/line feeds <CRLF>); that can cause signature to be coded differently in different platform;
- SignatureMethod refers to algorithms which are used for creating or validating signature as a *dsa-sha1* and it is used for the DSA algorithm and SHA-1 function for hashing;
- Reference element is complex but the most important is that it refers to data which should be signed; it is embedded in XML data or uniform resource identifier (URI) which refers to external data as document, web site or other digital content. In addition, Reference element determines transformation which will have influence to the content of hash function (via DigestMethod). As a result, there is a hash value stored as DigestValue;
- SignatureValue is a genuine computed value of signature. Rather than digitally signing content, signature is computed with element SignedInfo so that all references, algorithms and resultant values are digitally shared signed thus providing integrity of signed data;
- KeyInfo allows a recipient to receive key to approve signature if that is necessary. The structure of this function is very complex;
- Object element consists of illogical XML data which can be referenced within method SignedInfo. It can include Manifest element which provides varied list of references, where integrity of list is validated itself and integrity of the actual items will not validate the signature. The purpose of this list is to include the items which should be in relation to Manifest element. Also it defines SignatureProperties element where other properties of signature are stored, e.g. time and date when signature is created.

The standard XML Signature defines three types of digital signatures: *enveloped, enveloping,* and *detached. Enveloped* signature refers to signature of XML data where Signature element is in the XML body. *Enveloping* signature consists of XML content which is signed where Object element is used for signing data. Detached signature signs content that is external for XML signature defined by the URL.

```
<Signature Id-"MySignature"
xmlns="http://www.w3.org/2000/0 9/xmldsig#">
<SignedInfo>
<CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-d4n-
20010315"/>
 <SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-shal"/>
<Reference URI="http://www.company.ccm/file.doc">
<Transforms>
<Transform
 Algorithm="http://www.w3.org/TR/2001/REC-xml-cl4n-20
010315"/;
 </Transforms>
<DigestMethod
Algorithm="http://www.w3.org/2  0 0 0/0 9/xmldsig#shal"/>
<DigestValue>j90j2fnkfew3...</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>GFh8fw3greU...</SignatureValue>
 <KeyInfo>
 <KeyValue>
<DSAKeyValue>
<P>. . .</PXQ>. . .</QXG>. . .</GXY>. . .</Y>
</DSAKeyValue>
</KeyValue>
</KeyInfo>
 </Signature>
```

Figure 5: XML signature – an example.[21]

XML Signature allows each type of digital content to be signed and used together with Web services security standards.

XML Encryption. According to its design, XML has simple text format without embedded security. XML Encryption provides confidentiality of data through mechanisms for encrypting XML content where a symmetric encryption key is used. Techniques for exchanging keys are based on cryptography for exchanging public keys which provides secrecy for the key. Typically, a symmetric key is embedded within the XML message in cryptographic form, URI or it is considered through key exchanged data. Because exchange of public keys is very slow, a symmetric key can be used to encrypt data for performance reasons.

Also, XML encryption is presented as a XML. The structure of XML encryption considers following elements:

- EncryptedData is element which identifies that it is encrypted data;
- EncryptionMethod defines encryption algorithm which is used for encrypting data as a Triple-DES (3DES). This is an optional element and if it is not present then the recipient must know which algorithm is to be used to decrypt data;
- ds:KeyInfo has information for encrypted key which was used for message encryption, so that the actual key is embedded in encrypted form or there is information which affords the key to be derived or located;
- EncryptedKey has encrypted form of key which should be shared with others. As previously mentioned, this type of key will be encrypted using public key cryptography. It is possible that there are more recipients for the key, but for each of them there is an encrypted key element;
- AgreementMethod is an alternative way for sharing keys using Diffie-Hellman method. This method does not to allow keys to be embedded or shared in EncryptedKey element;

```
<EncryptedData Id? Type? MimeType? Encoding?>
<EncryptionMethod/>?
<ds:KeyInfo>
<EncryptedKey>?
<AgreementMethod>?
<ds:KeyName>?
<ds:RetrievalMethod>?
<ds:*>?
</ds:KeyInfo>?
<CipherData>
<CipherValue>?
 <CipherReference URI?>?
</CipherData>
<EncryptionProperties>?
</EncryptedData>

? = Zero or One Occurrence
+ = One or More Occurrences
* = Zero or More Occurrences
```

Figure 6: Informal XML encryption syntax.[22]

- ds:KeyName provides an additional way for sharing encryption keys according to their name;
- ds:RetrievalMethod is a method for retrieving encryption key form fromURI reference, whether it is in XML or external to it;
- ds:* refers to other information for keys which are emerging, such as X.509v3 keys, PGP keys, and SPKI keys;
- CipherData has encrypted data where CypherValue consists of data encrypted with base64 text or it uses CypherReference that refers to location of encrypted data in XML;
- EncryptionProperties includes additional properties as date and time for encryption.

Figure 7 presents an example of an XML encrypted message. Encrypted data are located in the CipherValue element. Together, XML Signature and XML Encryption standards form the basis on which WS-S standards rely.

Security Assertion Markup Language.[23] Together, SAML, XML Signature and XML Encryption are used to achieve integrity, confidentiality, authentication and SAML assertion in SOA based information systems. SAML uses XML for establishing communication between organizations or entities in separate security domains within user's identity, user's properties and user's attributes.

```
<EncryptedData
xmlns='http://www.w3.org/2001/04/xmlenc#'
Type='http://www.w3.org/2001/04/xmlenc#Element'/>
<EncryptionMethod
Algorithm='http://www.w3.org/2001/04/xmlenc#tripledes-
cbc'/>
<ds:KeyInfo
xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
<ds:KeyName>John Doe</ds:KeyName>
</ds:KeyInfo>
<CipherData><CipherValue>F59E7F12</CipherValue></Cip
herData>
</EncryptedData>
```

Figure 7: Example of an XML-encrypted message.[24]

SAML allows an entity or an organization to guarantee for user identity through SAML assertion. SAML assertion can be presented as a proof for identity of another entity to establish relations of trust. This is very important for SOA, because services are located in different companies and security domains. The previously mentioned concept provides a basis for federated identity, which protects the organization and improves security management in order to accomplish authentication and identity to other organizations.

SAML provides an approach to solving several problems:

- Web single Sign-on – an user can access a certain web site and to continue on the following web site using the same SAML assertion from previous web site;
- Delegated identity – a user's security clearance can be used in end-point service or web site from initial service or web site;
- Brokered Sign-on – mediating security service controls user's authentication. An attribute gained from mediating security services can be used in accessing different web sites;
- Authentication-based authorization – user attributes are embedded in SAML assertion.

Within SAML assertion information for user identity can be embedded such as e-mail, X.509, name of subject, employer's ID or other attributes. For privacy purposes, SAML 2.0 introduces the concept of pseudonyms or identification by pseudonyms which can be used instead of another identification type in order to hide personal information. SAML provides two methods for confirming subject identity. The first method is 'holder key' where message sender (subject) holds the key which used for digitally signing the message. Another method for confirming subject identity is 'sender-vouches,' which means that the digital signature has been created by a third party security service.

The SAML description was intended to support usage in SOA. Increasing confidentiality between service providers, SAML provides loose coupling and services independent with respect to user identity. As a security token, SAML is connected to the WS-S standards.

Web Services Security Standards.[25] The illustration on Figure 8 facilitates the comprehensive understanding of Web Services Security protocols and how they are shaped. The diagram shows that XML Signature, XML Encryption and SOAP form the basis for Web services Security standards.

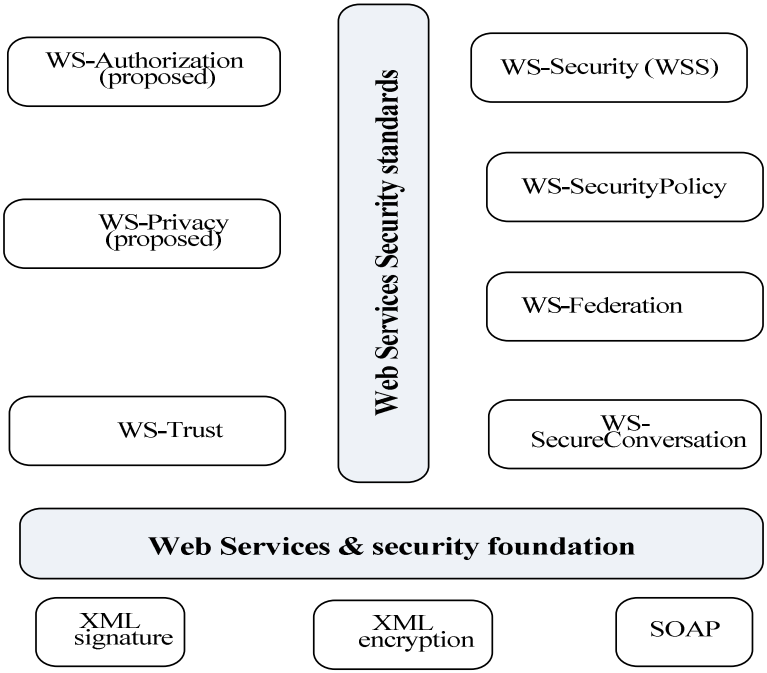

Figure 8: WS-S standards.

5. Model of Security Solution for Intelligence Information System-Based on SOA

Figure 9 presents two flows as follows: a control flow (blue direction) and data flow (red direction). The XML Signature standard is used in control flow to validate security policies. XML Encryption is used in data flow to validate security policies as well.

Our solution for service security enables application of standards described in section 3. In order to simplify the description, in the proposed model we do not explain how the policy for digital certificate is incorporated – in a sum, the respective aspects are placed in the Intelligence Information System (IIS) Centre.

Three phases can be distinguished both in the control and in the data flow:

1. Request phase – identifying information requester and registering the request with the purpose of establishing security mechanism described in Section 3;

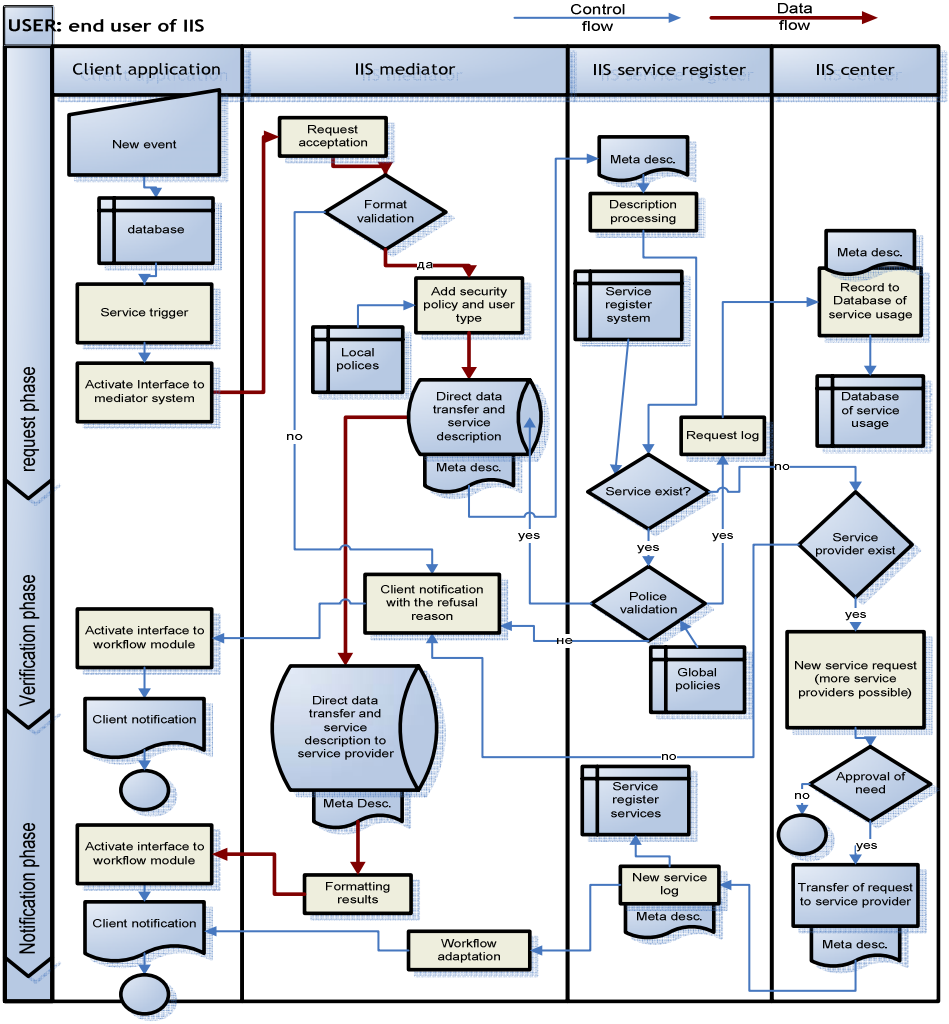

Figure 9: Model of Security Solution for Intelligence Information System.

2. Verification phase – identifying requester and whether its security mechanisms are appropriate for gaining response according to mandated information security policies;
3. Notification phase – according to security mechanisms and policies, the subject requesting information is notified for access to use information from

services or that access is denied sending forward an explanation in terms of security policy.

Requested information flows from source to the information requesters via mediation components which serve for connecting and formatting security systems in institutions. A mediation component is important for IIS, because it can be connected to more information systems which are embedded in heterogonous environment. The requested information is encrypted according to the unique security policy. Although IIS Centre and System registry are not involved, they play important roles in the control flow.

The control flow performs three functions:

1. Recording information requester in terms of date and time, location and type of user who requests the information;
2. Validation to security policy of user type called requester of information and security policies attached to the information;
3. Recording each request which is not followed with information at the moment of the request. This third function is of interest to information system designers working on future services.

The suggested model of security solution for an intelligence information system is highly structured. It provides not only a security mechanism, but also the following features related to IIS performance:

- Effective data transmission endorsing data encryption and data formatting at the appropriate level;
- Recording each request no matter whether it is inserted in a database or not, whether it is appropriate in view of the security policy or not. This feature provides for recording of possible disruptions of security policy;
- Flexible scalable mechanisms and mechanisms for extending services which are located in IIS registries.

Conclusion

This paper presented a Model of Security Solution for an Intelligence Information System based on the SOA paradigm that provides for secure data flow through information systems without any negative consequences to the security policies in terms of authentication, integrity, authorization, confidentiality and non-repudiation.

The proposed model affords appropriate recording of all requests and disruptions of security policies. The respective solution is well structured solution, easy to adopt, and in line with all contemporary security policies and protocols.

Notes:

[1] See for example Torry Harris Business Solutions, *Migration and Security in SOA*, White Paper (Leeds: Distributed Systems & Services Group, University of Leeds, March 2009), <www.thbs.com/pdfs/Migration_and_Security_in_SOA.pdf>.

[2] Mukhtiar Memon, Michael Hafner, and Ruth Breu, "Security as a Service - A Reference Architecture for SOA Security," in *Proceedings of the 7th International Workshop on Security in Information Systems, WOSIS 2009*, Milan, 6-7 May 2009 (INSTICC Press, 2009), 79-89.

[3] Beth Hutchison, Heather Hinton, and Maryann Hondo, *Security Patterns within a Service-Oriented Architecture*, White Paper (IBM, 2005), <www.ebizq.net/topics/woa/features/6535.html>.

[4] Ramarao Kanneganti and Prasad A. Chodavarapu, *SOA Security in Action* (Greenwich, CT: Manning Publications, 2008).

[5] Gunnar Peterson, "Service Oriented Security Architecture," *Information Security Bulletin* 10 (November 2005): 325-30, <www.arctecgroup.net/ISB1009GP.pdf>.

[6] Torry Harris Business Solutions, *Migration and Security in SOA*.

[7] Ruth Breu, Michael Hafner, Frank Innerhofer-Oberperfler, and Florian Wozak, "Model-Driven Security Engineering of Service Oriented Systems," in *Information Systems and e-Business Technologies*, Lecture Notes in Business Information Processing, Vol. 5, ed. Roland Kaschek, Christian Kop, Claudia Steinberger and Günther Fliedl (Berlin, Springer, 2008), 59–71.

[8] Michael Rosen, Boris Lublinsky, Kevin T. Smith and Marc J. Balcer, *Applied SOA: Service-Oriented Architecture and Design Strategies* (Indianapolis, IN: Wiley Publishing, 2008).

[9] Rosen, Lublinsky, Smith and Balcer, *Applied SOA*.

[10] Rosen, Lublinsky, Smith and Balcer, *Applied SOA*.

[11] Rosen, et al., *Applied SOA*; Javier Lopez, Jose A. Montenegro, Jose L. Vivas, Eiji Okamoto and Ed Dawson, "Specification and Design of Advanced Authentication Authorization Services," *Computer Standards and Interfaces* 27:5 (2005): 467–78.

[12] Rosen, Lublinsky, Smith and Balcer, *Applied SOA*.

[13] Rosen, et al., *Applied SOA*.

[14] Rosen, et al., *Applied SOA*.

[15] Fumiko Satoh, Yuichi Nakamura, Nirmal K. Mukhi, Michiaki Tatsubori, and Kouichi Ono. "Methodology and Tools for End-to-End SOA Security Configurations," in *SERVICES '08*, Honolulu, HI, 6-11 July 2008, pp. 307–14.

[16] Rosen, et al., *Applied SOA*; Michael Hafner, *SECTET: A Domain Architecture for Model Driven Security*, PhD Thesis (November 2006).

[17] OASIS, WS-SecurityPolicy, 2007, <http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html>.

[18] Harold F. Tipton and Micki Krause, *Information Security Management Handbook*, Sixth Edition (Auerbach, 2008).

[19] Tipton and Micki Krause, *Information Security Management Handbook*.

[20] Tipton and Micki Krause, *Information Security Management Handbook*.

[21] Tipton and Micki Krause, *Information Security Management Handbook*.

[22] Tipton and Micki Krause, *Information Security Management Handbook*.

[23] OASIS Security Services Technical Committee. *Security Assertion Markup Language (SAML)* (OASIS, 2005), <www.oasis-open.org>.

[24] Tipton and Micki Krause, *Information Security Management Handbook*.

[25] *Service-Oriented Security: An Application-Centric Look at Identity Management* (Oracle, 2008), <www.oracle.com>.

JUGOSLAV ACHKOSKI is a Ph.D. candidate, and teaching assistant at Military Academy "General Mihailo Apostolski" – Skopje, associate member of "Goce Delchev" University – Shtip, Macedonia. He holds master degree in Computer Science. His field of research focuses on developing IS using contemporary ICT. He has published conference and journal research papers. *E-mail*: jugoslav_ackoski@yahoo.com.

VLADIMIR TRAJKOVIK received Ph.D. degrees 2003. He joined the Ss. Cyril and Methodious University, Skopje, R .Macedonia, in December 1997. His current position is the Associate Professor and the Vice Dean for Science at the Faculty of Computer Science and Engineering. He is currently responsible for several courses at undergraduate level, and "Mobile and Web Services", "Collaborative Systems" and "Innovative Technologies" at postgraduate level. He realized multiple research stays with several European Universities as visiting scientist within the scope of different EU and international projects. He is an author of more than 80 journal and conference papers. Dr. Trajkovik has participated (as researcher or project leader) in 15 international projects sponsored by European Commission in the framework of TEMPUS, PHARE and FP programs. *E-mail*: trvlado@feit.ukim.edu.mk.

METODIJA DOJCINOVSKI is a Head for Department for Security, Crises management, Rescue and Protection at Military academy "General Mihailo Apostolski" – Skopje, associate member of "Goce Delchev" University – Shtip, Macedonia. As a associate professor, his field of interest are group of subjects in security and defense area. His research field focuses on: National security, Intelligence and security systems, crises management and security planning processes. *E-mail*: m_dojcinovski@yahoo.com.