A Security Model and its Strategies for Web Services

Mehdi Sabbari
Sama technical and vocational training college,
Islamic Azad University, Borujerd Branch
Borujerd, Iran

Hadiseh Seyyed Alipour
Qazvin Islamic Azad University
Qazvin, Iran

ABSTRACT
Service Oriented Architecture (SOA) establishes an architectural model that aims to enhance the efficiency, agility, and productivity of an enterprise by positioning services as the primary means through which solution logic is represented in support of the realization of strategic goals associated with service-oriented computing. Web services seem to become the preferred implementation technology for realizing the SOA promise of maximum service sharing, reuse, and interoperability. However, one of the big stumbling blocks in executing SOA is security. This article addresses security in SOA with understand which security requirements and proposed conceptual security reference model for SOA with techniques and industry standards are right for web services.

General Terms
Service Oriented Architecture, Distributed System, Software, Web Application, Security, Web Services, et. al.

Keywords
Service Oriented Architecture (SOA), Security requirements, Reference model, Web Services, Security Standards.

1. INTRODUCTION
Nowadays, computing systems cannot be viewed anymore as isolated hosts offering computational functionality to human users. Rather, modern computing systems are loosely coupled components distributed over a network and communicating with each other: they are heterogeneous, distributed, and inter-connected. For one, it is evident that a system which is connected to other systems is exposed to a considerable amount of additional security threats.

SOA supplies various services to process business by joining various specific components together in a loose coupled manner with uniform interfaces [1][2][5]. SOA and the corresponding Service-Oriented Computing (SOC) have received significant attention recently as major computer and software companies, such as IBM, Intel, Microsoft, HP, SAP, and Sun Microsystems, as well as government agencies, such as U.S. Department of Defense (DoD), have embraced SOA/SOC [2].

Web Service supports the communication between applications developed on different platform by different programming languages with different technological standards. The ultimate goal of Web Service is to realize the integration and interaction between various systems in Internet/Intranet environment, just similar to the function of components [2][4]. Web services technology platform is comprised of the following core open technologies and specifications: Web Services Description Language (WSDL), XML Schema Definition Language (XSD), SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery, and Integration), and the WS-I Basic Profile [1]. A SOA implementation through the web services represents an inherently decentralized computing concept. Hence, an appropriate understanding of the concept of security needs to take into account the system, its context and dependencies between both.

We present a definition of security in the environment of Service Oriented Architecture in section (2) of the article. Here we also review some of the models which have been presented in the field of Service Oriented Architecture. Then in section (3) we proposed a conceptual reference model including all the security requirements in a SOA environment. Further, all the available layers and security requirements in the model are described. In section (4), we also deliver a classification of security standards and guidelines which can support the already-mentioned requirements of the model and the importance degree of security principles in each layer is shown by spider graphs. The classification helps the programmers and developers – not to be confused and worried during the establishing and development of security in SOA environment – to determine the security demands they expect from the proposed model, then based on its offered solutions they implement the security requirements in web services.

2. SECURITY IN SOA ENVIRONMENT
Security is an important issue that must be well-defined in SOA environment so that it could be used in implementing the web services. In general and according to [6], security in the environment can be defined in this way:

"the sum of all techniques, methods, procedures and activities employed to maintain an ideal state specified through a set of rules of what is authorized and what is not in a heterogeneous, decentralized, and inter-connected computing system".

Security requirements provide a categorization of the most basic security needs of an asset. Define in [6]:

"a statement of intent to counter identified threats and/or satisfy identified organization security policies and assumptions".

So far, several models for the security of Service Oriented Architecture have been presented. Such as the model that is presented by National Security Telecommunications and Information Systems Security (NSTISS) committee in 1994. The NSTISS committee defines "a comprehensive model for the security of information systems which also functions as an assessment, system development and evaluation tool" [21][7]. This three-dimensional model illustrated in fig.1 attempts to address all security issues in an information system. The three layers of security measures can be utilized to minimize vulnerabilities based on the threats to an information asset.

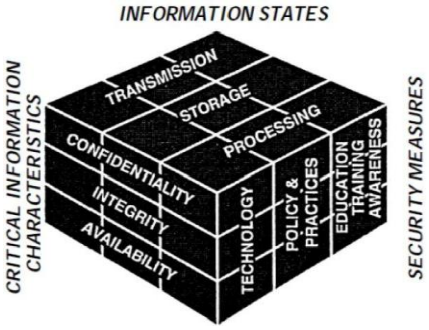

Fig 1: Comprehensive Model for securing Information Systems

The model is too basic and explains security requirements of information systems in a general format. It doesn't refer to any implementing tool and method so it cannot be a complete basis for security of Service Oriented Architecture systems.

The same goes for the model which is presented by Pajevski in 2004 shown in fig.2. According [20], SOA security issues can be resolved by mitigating risks caused by the increased exposure of services by using a two-fold approach. Firstly, to use a proxy service to insulate services from consumers and to split the service registry into public and private areas. By using this technique we can debilitate and also to an extent eliminate direct attacks. Secondly, by using access control techniques which limits what the users can do, whilst also limits the harm they can cause.

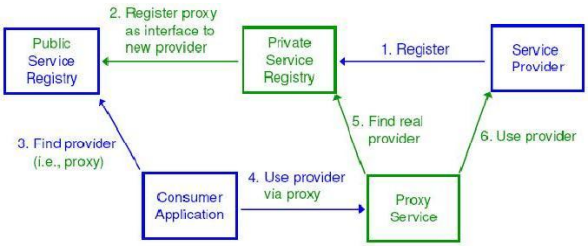

Fig 2: A Security Enhanced SOA Interaction Model

It mostly focuses on the security of a service provider while in a Service Oriented Environment, other security principles must be regarded such as transmission of messages, packaging the messages, identities management, interaction of services, etc. and Lack of some factors is also felt like explaining techniques, tools and methods of implementation.

Regarding reference [7] too, a model was presented in 2009 based on the model presented by NSTISS committee in which some security principles including authentication, authorization, identity, auditing & compliance and security policies were available. But there also were some defects in the model e.g. characteristics and dimensions of the model have not been explained clearly, there weren't some of the security needs needed for Service Oriented Architecture (Non-Repudiation, Federation, Accountability, policies management, Enforcement the policies, etc.), the model was completely conceptual and tools and techniques for meeting the needs in Service Oriented Architecture environments like web services have not been mentioned.

Since the previous models have referred to some security requirements in this field, and the models' structures were mostly conceptual and the presented methods also were sporadic, brief and out of an overall framework and classification, so in this article we have tried to discuss all security needs of a Service Oriented Architecture environment, and followed by classified explanation of implementing methods of security principles in web services.

3. A SECURITY MODEL FOR SOA ENVIRONMENT

Our proposed model shown in Fig.3 is a security reference model in SOA environment. The right layer on the model shows those areas of SOA environment in which the security must be certainly established. The middle layer shows all the security requirements in SOA; the requirements are (security principles, security policy, and physical infrastructure). The left layer shows the rule of SOA which is responsible for supervising the methods and tools, guidelines and defined roles in the entire SOA environment.

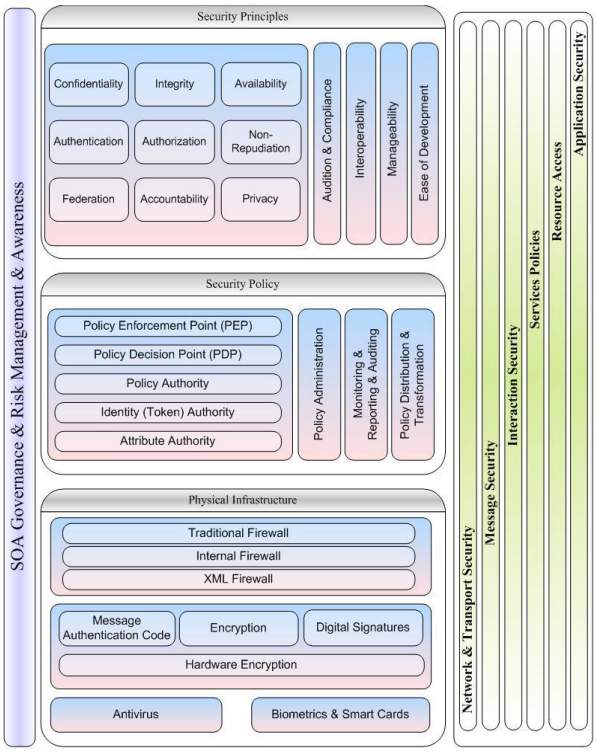

Fig 3: A security model for SOA environment

3.1 Security Principles Layer

We proposed, security principles are divided into two general categories: functional aspects of security and non-functional aspects of security that the non-functional aspects of security covers all functional requirement of security. Those cases belong to functional aspects of security group include (confidentiality, integrity, availability, authentication, authorization, non-repudiation, federation, accountability and privacy). Those cases belong to non-functional aspects of security include (auditing & compliance, interoperability, manageability and ease of development). The subclasses which

belong to manageability group and are not shown in the model include configuration, provisioning, throttling & hack-proof and maintenance.

Security can be viewed as a three part process which involves protection, detection and response. Functional aspect of security requirements are protection schemes. We need a requirement which helps us to detect security vulnerabilities and respond with suitable measures which involves auditing. Audition & compliance service itself is viewed as one of the main non-functional aspects of security since the operations in SOA environment have crucial importance and must be concerned more.

1. *Confidentiality*. This is the characteristic or assurance that information is being shared only among authorized persons, entities and processes at authorized times and in an authorized manner [7].
2. *Integrity*. This is the characteristic or assurance that a given piece of information is timely, accurate, authentic and complete. Integrity acts as a primary indicator of security in information systems [2][7][8].
3. *Availability*. This is means that resources are accessible to authorized entities within a reasonable time. Resources should also be accessible in a convenient format [13]. The opposite of availability, denial of service (DOS), is also a common concept in security, since DOS attacks can cripple the availability of a computer system [8].
4. *Authentication*. This is the act of proofing that a claimed identity is true [7]. Therefore, authentication also involves identification. Identification involves the act of claiming an identity. Identification is usually the first step in the authentication and authorization process [8].
5. *Authorization*. This is a security concept where access to resources is allowed to only those who are permitted to use them. This is termed as access control and is usually determined by finding out if a person is a member of a particular privileged group [8].
6. *Non-Repudiation*. Communication security also has to make sure that an entity sending or receiving information cannot later deny having sent or received the information [17].
7. *Federation*. When a service requires authentication against another external system, federation is used. Federation is an extension of authentication that helps the service provider to establish trust between the provider's security domain and an external domain. So the external provider trusts the request and considers it authenticated without expecting an additional credential [13].
8. *Accountability*. This security principle is to keep an account for provided determinant in communications taken place. Therefore people who share the communications can be responsible for their activities based on the account [13].
9. *Privacy*. This is to keep the different informative parts of a message away from being seen by the entities not related to these parts. So the XML record must be parted and it also must have the ability to protect other parts despite service provider's interest [9].
10. *Audition & compliance*. A system should be configured to track messages between services and generate usage logs during specific periods of time. This audit serves as an important record of what has happened that can be used to investigate problems and diagnose potential security weaknesses. Compliance is the state of being in accordance with established guidelines, specifications and legislation. It's not just archiving, but the ability to access information that is essential for achieving compliance [7].
11. *Interoperability*. This is unique concept for SOA and tells a different security solution should not violate the services are compatible [10].
12. *Manageability*. This means that, need security solutions to protect the various services. A good security architecture should be easy to manage [10].
13. *Ease of Development*. This is aspect of security solutions for all common. However the complexity of developing security solutions is higher. The possibility of adapting the architecture would be less relevant [10].

3.2 Security Policy Layer

We proposed, in general the security policy layer can be divided in two categories: accomplishment security policies and management security policies in which the management security policies are covers them all accomplishment security policies elements. Accomplishment security policies include Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Authority, Identity (Token) Authority and Attribute Authority which are the basic principles of SOA security policy and must be considered in the model. Management security policies include Policy Administration (PA), policy distribution and transformation, monitoring and reporting and auditing which must be considered in the model.

1. *Policy Enforcement Point (PEP)*. PEP is actually the interface of the whole environment to the outside world. It receives the access requests and evaluates them with the help of the other actors and permits or denies the access to the resource [14].
2. *Policy Decision Point (PDP)*. PDP is the main decision point for the access requests. It collects all the necessary information from other actors and concludes a decision [14].
3. *Policy Authority*. The storage keeps the rules used to decide to set access permissions to resources. If a entities is authorized to access an object according to policies defined in a policy repository This storage can also be a database [11].
4. *Identity (Token) Authority*. Maintains identification information and issues security tokens. e.g., a PKI service interface or a Kerberos Key Distribution Center (KDC).

5. *Attribute Authority*. In this storage, there are some information on ability to identify entities features. It could be used as a directory service (e.g. LDAP) or database (e.g. Oracle).
6. *Policy Administration (PA).* Addresses policy life-cycle management including creation, maintenance, change, and deletion. This should allow for business policies to get refined to service specific policies like security, performance indicators and metrics, and trust policies [14].
7. *Policy Distribution and Transformation*. Policies may be published that declare the requirement for a requestor runtime to ensure message confidentiality and provide some evidence of the identity of the requesting user.
8. *Monitoring & Reporting & Auditing*. The system must provide the completed log mechanism to record all the events and the related identity as the audit clue. Regular security audit is useful to find out the security flaws, violating security behavior, cheating and other behaviors that attempt to bypassed safety measures. Besides, the system should apply for the prevention measures such as load balancing, virus testing, packet filtering and fault switching or backup to prevent service denial of attack or other potential security attack.

3.3 Physical Infrastructure Layer

In addition to the security cases already discussed, there are some more cases mainly hardware, which are related to SOA underneath layers. These cases are usually about hardware system security, messages, transferring them in the network and other fields including: hardware encryption, smart cards, antivirus, Biometrics authentication, firewall (traditional firewall, internal firewall, XML firewall).

3.4 SOA Governance & Risk Management & Awareness

In fact it is the governance of tools for defining the organizational roles and allowing people to guiding and implementing the roles. There are different technical tools in the market to help automating the certain, defined aspects of governance process. The other element in the layer is risk management. The risk could be defined as a combination of probability of an accident with its consequences. Since there would be many risks in SOA environment because of a wide range of different interactions and places, it is necessary to have a risk management esp. On the other hand, the issue of informing and educating the whole security requirements and concepts in SOA environment must be fulfilled so that the developers, users, managers and others are able to use it correctly when it is needed; this aim is achieved by awareness.

3.5 SOA Security Precincts Layer

We proposed, the most important SOA security precincts in this part of the comprehensive model including network & transport layer, message layer, interaction layer, services policies, resource access and application security. In fact for making all of the SOA environment's scopes secure, the concepts of all three layers (physical infrastructure, security policy, security principles) have been used in the scopes; it means that the concepts, principles and policies could be used as guides and paths to gain the appropriate security in SOA security precincts. Then, after finding out the security needs of different scopes, the need in SOA environment could be met using the presented technology.

3.6 Advantages of Presented Model

The advantages of the presented model are: 1) it is layered and separated, security requirements of each layer are explained separately. 2) Being functionality or non-functionality of the security requirements is specified. 3) In addition, we tried to import all security requirements in the SOA environment and to show all dimensions which need to use these principles to secure. 4) Another benefit is that this model is not limited to technology and is independent of implementing form but provides the needed concepts and the new technologies can be used to establish the concepts. Further in the article, available technologies for implementing SOA through web services will be mentioned.

4. THE AVAILABLE STANDARDS & TECHNIQUES

Several organizations, including OASIS, W3C, the Liberty Alliance, and various members of industry have put together numerous security standards and techniques for securing Web services. For the most part, these standards and techniques all complement or extend one another, but there are some conflicting or competing standards [11]. We deliver a review and a classification of different standards here in this part; we also have determined which security requirement is needed in every area of the SOA environment and which techniques and guidelines are used in its implementation through the web services.

Also Based on our research in the field of SOA security We've reached the conclusion that the SOA environment should be divided into (transport & network, message, interaction between services, services policies, access control and applications) domains, and Security should be respected in any field. Therefore we did polling among thirty Experts in computer and information technology, and we obtained useful results. Based on the obtained results, we have shown the significance degree of security needs in each field by spider graphs (minimum (0) – maximum (5)). By this, producers of web services systems have more focus on significant factors of each field.

4.1 Transport/Network Layer Security

The HTTPS protocol is defined as HTTP over SSL/TLS. SSL/TLS provide socket-layer security, encrypting all communication over a particular TCP connection immediately granting an insecure application-layer protocol security without altering it. Through SSL/TLS, HTTPS supports authentication, confidentiality, and integrity of data sent between the endpoints [11]. SSL/TLS enables point-to-point secure sessions (Fig. 4). IPSec like SSL/TLS is another network layer standard for transport security that may become important for Web services.

XML firewalls provide packet level inspection of all inbound and outbound traffic to the back-end web application server. Internal traffic that is not seen by the firewall cannot be filtered; as a result, internal users can mount attacks on other users and

networks without the firewall being able to intervene. internal firewalls have to be deployed in the internal network. Security standards in this area as shown in table 1.

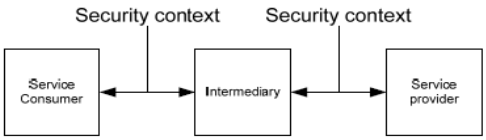

Fig 4: Point-to-point configuration

Table 1. Standards & strategies for transport/network layer security

Area	Requirements	Standards
Transport/Network	Authentication Confidentiality Integrity	SSL/TLS
		HTTPS
	Firewall	IPSec
		Internal Firewall
		XML Firewall

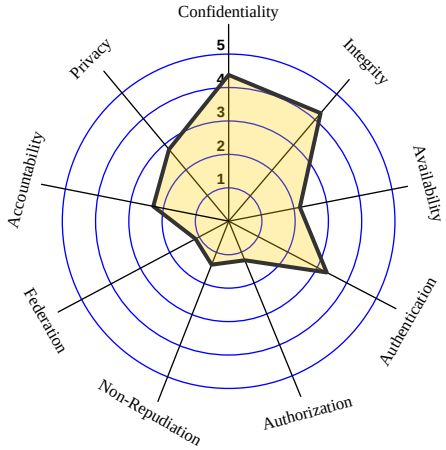

Fig 5: The importance degree of security principles in network /transport layer

4.2 Message Layer Security

Message level security ensures a message is protected throughout communication of the message. This means that confidentiality and integrity of the message is protected all the way from message sender to message receiver (Fig. 6) [8].

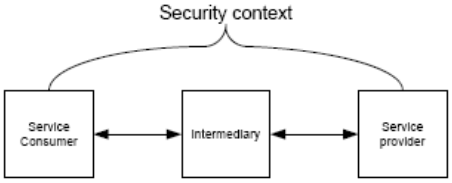

Fig 6: End-to-End configuration

For integrity, XML Signature defines digital signatures and related cryptographic integrity and authentication measures for XML. For confidentiality, XML Encryption supports encryption using a variety of both symmetric and asymmetric cryptographic algorithms. For key management, XKMS addresses PKI and key management network services in XML.

WS-Security (Web Service Security) standard was designed to use the XML Encryption and XML Signature specifications for message-layer confidentiality and integrity [11][19]. WS-Security also includes profiles that specify how to insert different types of binary and XML security tokens in WS-Security headers for authentication and authorization purposes: Username with optional Password digest, X.509 Certificate (a signed data structure designed to send a public key to a receiving party), Kerberos ticket (an authentication and session token), REL document (license tokens inserted in WS-Security headers are used for authorization), XCBF document (defines how to use the XML Common Biometric Format language for authentication with the WS-Security specification) [15].

WS-Addressing provides an XML framework for identifying web services endpoints and for securing end-to-end endpoint identification in messages [15].

The WS-Reliability and WS-ReliableMessaging standards provide some level of QoS. Both standards support guaranteed message delivery and message ordering. The standard considers other QoS parameters, such as rate of failure or average latency, as out of scope because they are usually dealt with by lower layer protocols. Security standards and strategies in this area as shown in table 2.

Table 2. Standards & strategies for message layer security

Area	Requirements	Standards
Message	Confidentiality Integrity Privacy	XML encryption
		XML signature
		XKMS
	Authentication	WS-Security
		WS-Addressing
		X.509 Certificate
		Kerberos ticket
		REL
		XCBF
	Availability	WS-ReliableMessaging
		WS-Reliability

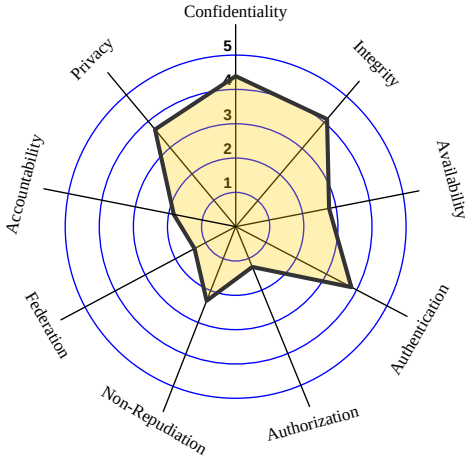

Fig 7: The importance degree of security principles in message layer

4.3 Interaction Layer Security

Trust in distributed computing environments is usually verified using PKI (Public Key Infrastructure) certificates signed by a certificate authority or by passing custom tokens generated by a TTP (Trusted Third Party), as is done in a Kerberos environment. In a SOA, Web services from multiple organizations should be able trust one another without requiring extensive restructuring of the trust environment. To this end, trust federation frameworks can be configured to use an organization's pre-existing authentication mechanisms.

The Liberty Alliance provides both Web application and Web service federation using SAML to perform the trust brokering.

WS-Federation allows different security realms to federate by defining trust brokers, who will validate security tokens used between Web services using WS-Trust [11].

WS-Trust is used to exchange trust tokens between Web services. WS-Trust is an extension to WS-Security that provides methods for issuing, renewing, and validating security tokens as well as methods for establishing and brokering trust relationships between Web services [19].

SAML (Security Assertion Markup Language) defines an XML vocabulary for sharing security assertions that specify whether and how an entity was authenticated, information about an entity's attributes or whether an entity is authorized to perform a particular action. These assertions enable identity federation and distributed authorization within a SOA [11][19]. SAML provides a standard way to transfer cookies across multiple Internet domains. SAML provides a standard protocol to implement SSO within a single domain or across multiple domains. SAML enables identity management (a user can have several identities on the Internet) and SAML provides a standard security token (a SAML assertion) that can be used with the WS-Security framework [15].

WS-SecureConversation leverages WS-Security and WS-Trust. WS-SecureConversation defines the creation and sharing of security contexts between communicating parties.

Security contexts mitigate the overhead involved in multiple-message exchanges. WS-SecureConversation defines a Security Context Token (SCT) element to support the requirements of security contexts. An SCT involves a shared secret used to sign and/or encrypt messages. Security standards and strategies in this area as shown in table 3.

Table 3. Standards & strategies for interaction layer security

Area	Requirements	Standards
Interaction	Non-Repudiation	XML signature
		XML Encryption
	Federation	Liberty Alliance
		WS-Federation
		WS-Trust
		Kerberos ticket
		SAML
	Authentication	WS-SecureConversation

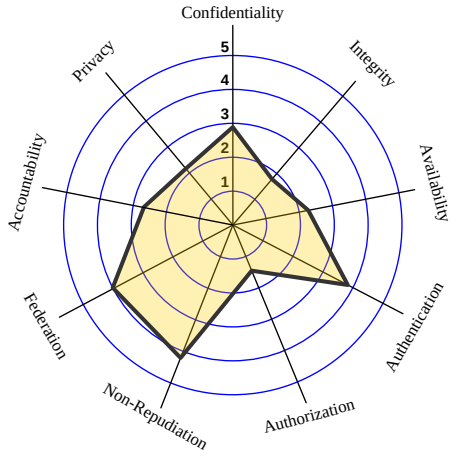

Fig 8: The importance degree of security principles in interaction layer

4.4 Services Policies Layer Security

A web service provider may define conditions (or policies) under which a service is to be provided.

WS-Policy will be fully extensible and will not place limits on the types of requirements and capabilities that may be described; however, the specification will likely identify several basic service attributes including privacy attributes, encoding formats, security token requirements, and supported algorithms. For example, a policy assertion may stipulate that a request to a web service be encrypted. Likewise, a policy assertion can define the maximum message size that a web service can accept.

WS-SecurityPolicy defines assertions to specify integrity, confidentiality, and information about security tokens [15].

WS-MetadataExchange specification defines an encapsulation format for Web service metadata (such as WS-Policy expressions), a mechanism for metadata-driven message exchange, and relies on the WS-Transfer specification to provide a Web service endpoint from which requesters can retrieve the metadata.

The WS-PolicyAttachment specification defines how to reference policies from WSDL definitions, how to associate policies with deployed endpoints, and how to associate policies with UDDI entries. Security standards and strategies in this area as shown in table 4.

Table 4. Standards & strategies for services policies layer security

Area	Requirements	Standards
Services Policies	Authorization	WS-Policy
		WS-SecurityPolicy
		WS-MetadataExchange
		WS-PolicyAttachment

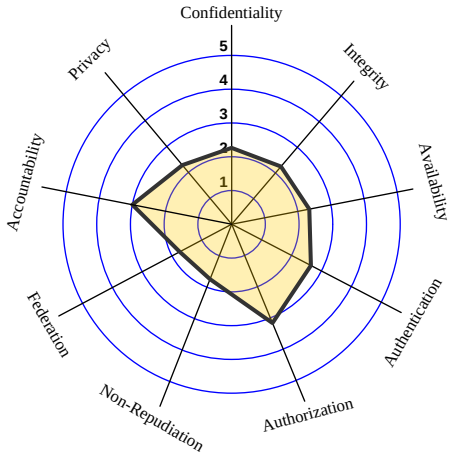

Fig 9: The importance degree of security principles in services policies layer

4.5 Resource Access Layer Security

Given the distributed nature of Web services architectures, managing authorization and access control credentials for users in a SOA environment can be challenging.

In IBAC, permissions to access a resource is directly associated with a subject's identifier (e.g., a user name). Access to the resource is only granted when such an association exists. An example of IBAC is the use of Access Control Lists (ACL), commonly found in operation systems and network security services [12].

Role-based access control (RBAC) is an authorization mechanism that associates a set of access privileges with a particular role, often corresponding to a job function [16].

ABAC policy rules are generated as Boolean functions of S, R, and E attributes and dictate whether a subject S can access a resource R in a particular environment E - as loosely indicated in Formula (1). ABAC clearly provides an advantage over traditional RBAC when extended into SOA environments, which can be extremely dynamic in nature [3][12].

$$Rule\ X : can\_access(s, r, e) \leftarrow f(ATTR(s), ATTR(r), ATTR(e)). \quad (1)$$

Risk adaptive access control (RAdAC) is another variation on traditional access control methods. As opposed to RBAC, ABAC, and PBAC, however, RAdAC makes access control decisions on the basis of a relative risk profile of the subject and not necessarily strictly on the basis of a predefined policy rule.

XACML (eXtensible Access Control Markup Language) is an OASIS standard that describes both a policy language implemented in XML and an access control decision request/response language implemented in XML [11][19]. The policy language details general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service) [18].

EPAL (Enterprise Privacy Authorization Language) is a formal language for writing enterprise privacy policies to govern data handling practices in IT systems according to fine-grained positive and negative authorization rights.

Logging operation for recording the events and accidents meet the accountability needs [13]. Security standards and strategies in this area as shown in table 5.

Table 5. Standards & strategies for resource access layer security

Area	Requirements	Standards
Resource Access	Authentication	X.509 Certificate
		Kerberos ticket
		WS-Security Token
		REL
		XCBF
	Authorization	IBAC/RBAC / ABAC /RAdAC
		SAML
		XACML
	Privacy	EPAL
	Accountability	Logging

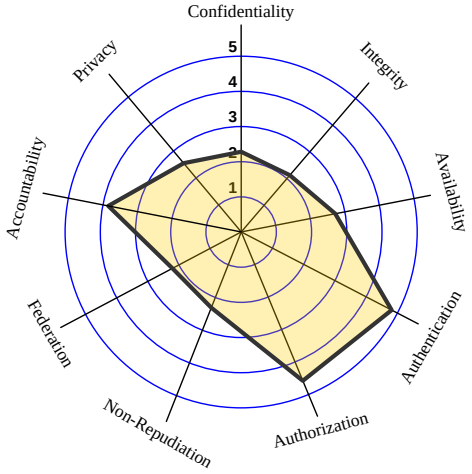

Fig 10: The importance degree of security principles in resource access layer

4.6 Application Layer Security

Ultimately, as for applications have a reasonable security Is required to use some physical infrastructures to avoid such applications from some risks like viruses, Unauthorized entries, uncontrolled access, copy, Transfer of sensitive information. These infrastructures as shown in table 6.

Table 6. Standards & strategies for application layer security

Area	Requirements	Standards
Application	Hardware Encryption	Encryption Ciphers
	Biometrics	Fingerprint Identification
		Retina Scan
		Voice Analysis
	Smart Cards	microchip technology
	Authentication	Username & Password
	Antivirus	Signature Based Detection
		Heuristic-Based Detection

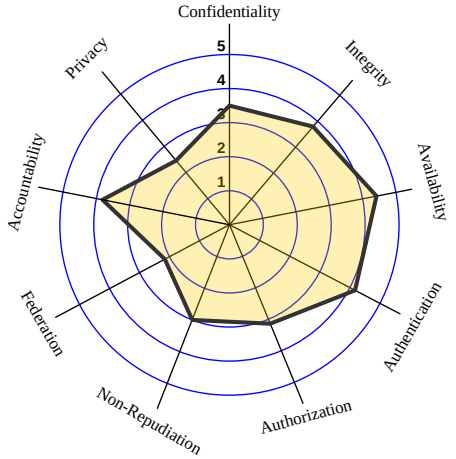

Fig 11: The importance degree of security principles in application layer

5. CONCLUSION

In this paper SOA is introduced as a new style of software architecture and its basic elements has been expressed. Because of SOA has distributed and heterogeneous nature, and uses from Combination of different service to create programs, and also has different users in various areas of in organization and out organization, Security issue is raised as a challenge in this area. In this paper security needs are expressed and accordingly a conceptual model for security architecture is presented. This model is a mapping between different areas of SOA environment and the requirements that should be respected in each area. For the implementation of SOA is typically through Web services, the strategies and standards about implementing security in Web services are classified in detail.

6. REFERENCES

[1] Erl, T. 2007 SOA: Principles of Service Design. Prentice Hall/Pearson PTR.

[2] Wang, j., Yu, A., Zhang, X. and Qu, L. 2009 A Dynamic Data Integration Model Based on SOA. In: 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, pp. 196-199,In IEEE.

[3] Karp, A. H. and Li, j. 2010 Solving the Transitive Access Problem for the Services Oriented Architecture. IEEE International Conference on Availability, Reliability and Security, DOI 10.1109/ARES.

[4] Papazoglou, M. P. and Van Den Heuvel, W. 2007 Service oriented architectures: approaches, technologies and research issues. Springer-Verlag, pp. 389-415.

[5] Eckert, J., Bachhuber, M., Miede, A., Pasageorgiou, A. and Steinmetz, R. 2010 Service-oriented Architectures in the German Banking Industry-A Multi-Participant Case Study. In: 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010).

[6] Hafner, M., and Breu, R. 2009 Security Engineering for Service-Oriented Architectures. Springer.

[7] Jonnaganti, V. 2009 An Integrated Security Model for the Management of SOA- Improving the attractiveness of SOA Environments through a strong Architectural Integrity. Master Thesis, University of Gothenburg Department of Applied Information Technology Gothenburg, Sweden.

[8] Fiere, J. 2007 SOA Security. Master Thesis, Faculty of Science Vrije Universiteit Amsterdam.

[9] Jana, D., Chaudhuri, A. and Bhaumik, B. 2009 Privacy and Anonymity Protection in Computational Grid Services. International Journal of Computer Science and Applications, Vol, 6, No, 1, pp. 98-107.

[10] Kanneganti, R. and Chodavarapu, P. A. 2008 SOA Security. Manning.

[11] Singhal, A., Winograd, T. and Scarfone, K. 2007 Guide to Secure Web Services. NIST Special Publication.

[12] Yuan, E. and Tong, J. 2005 Attributed Based Access Control (ABAC) for Web Services. IEEE International Conference on Web Services (ICWS'05).

[13] Janssen, J. 2008 Identity management within an organization. Master Thesis, Radbound University Nijmegen.

[14] Hung Le, X., Lee, S., Lee, Y., Lee, H., Khalid, M. and Sankar, R. 2010 Activity-oriented access control to ubiquitous hospital information and services. Elsevier, pp. 2979-2990.

[15] Chanliau, M. 2006 Web Services Security: What's Required To Secure A Service-Oriented Architecture. An Oracle White Paper.

[16] S.Sandhu, R. and et al. 1996 Role-Based Access Control Models. IEEE Computer, pp. 38-47.

[17] Salomon, D. 2006 Foundations of Computer Security. Springer-Verlag London Limited.

[18] Moses, T. and et al. 2005 eXtensible Access Control Markup Language(XACML) Version 2.0. OASIS Standard.

[19] Al-Kofahi, M., Chang, A. and E.Daniels, T. 2008 SCWIM An Integrity for SOA Networks. IEEE International Conference on Web Services, pp. 675-682.

[20] J.Pajevski, M. 2004 A Security Model for Service Oriented Architectures. Distributed Systems Technologies Group, Retrieved from NASA Web site: http://www.oasisopen.org/committees/download.php/17573/06-04-00008.000.pdf.

[21] Mcconnell, J.M. 1994 NSTISS : National Security Telecommunications and Information Systems Security. Rep. No 4011. Retrieved from www.cnss.gov/Assets/pdf/nstissi_4011.pdf.