Journal of Information Technology Management

ISSN #1042-1319

A Publication of the Association of Management

MANAGING WEB SERVICES SECURITY

KENNY KHOO
UNIVERSITY OF MARYLAND, BALTIMORE COUNTY
Kkhoo1@umbc.edu

LINA ZHOU
UNIVERSITY OF MARYLAND, BALTIMORE COUNTY
zhoul@umbc.edu

ABSTRACT

The promising features of Web services also make them vulnerable to new types of security threats. Web service providers must assure their clients' confidentiality, integrity and availability over a trusted relationship that may be asynchronous and that may involve multiple business partners. Despite the continued significance of the traditional approaches to securing content, transmission and connection in a Web-based business environment, including Secure Socket Layer, Virtual Private Networks, Internet Protocol Security, and so on, they are not able to address the new challenges posed by Web services. This paper aims to provide insight into the management of Web services security. We first introduce key concepts and reviews state-of-the-art standards for Web services security. Then, by aligning the Web services security standards with security threats, we provide guidance for the practical implementation of Web services security. Finally, we point out some limitations in the current practice and suggest future directions of securing Web services.

Keywords: XML, SOAP, WS-Security, Web Services, Security.

INTRODUCTION

Web services provide a new channel for conducting business [8]. There is a momentum for businesses to adopt web services as part of business-to-business transactions and to develop new business models [4]. According to a survey of senior-level IT executives, engineers and project managers at global 2000 organizations [15], 75% of the respondents planned to roll out Web services in the next 12 months using a number of security standards. In addition, more than 40% plan to use four or more standards with a majority emphasizing WS-Security [9] and Security Assertion Mark-up Language (SAML) [16]. XML Encryption [17], XML Signature [19], Kerberos, and X.509 were among the other standards that the respondents plan to use. However, more than half of the respondents in the same survey won't use Web services outside the corporate firewall until they are sure that transactions are protected against cyber-attacks. This raises serious concerns over the security issues associated with adopting Web services.

Web services extend the current client-server model with the concept of "loose coupling," which allow services to be discoverable, platform independent, and expressible with a self-describing interface [20]. It provides a means of communication and interoperability among different software applications that may run on different platforms. However, these applications bring risks to business providers by potentially exposing internal business processes and confidential data to

distributed clients. An example of such an application is stock trading [11]. The business of stock trading is highly time-sensitive and dependent upon the information-intensive transactions. Traders expect to have access to trading processes even while traveling (either via wireless or fixed lines), international brokers have the need to trade simultaneously in multi-nation markets, and currency traders typically diversify their portfolio by trading over international exchanges. Therefore, large volumes of information are being exchanged at any time in the stock market. On the one hand, most of the information, such as account numbers, personal information, and gain/loss from completed transactions should be kept confidential from the third parties. On the other hand, the stock trading business strives to ensure that traders are really who they are in conducting the transactions. All these security issues pose challenges to successfully deploying Web services. Nevertheless, companies have successfully implemented web-services in the finance industry. For example, a large German joint-use center enhanced the integration capabilities of its core banking system with 237 individual savings banks by using web services [22]. Currently, all security requirements are fully addressed on the network layer, on the transport layer, and on the application layer.

The safety of information exchange can be assured by addressing five security requirements: authentication, authorization, non-repudiation, confidentiality and Integrity [14]. Client-to-server and server-to-server authentication should be seamless. If a Web service provider receives a request for service, it must be able to verify the requester's identity and privileges. Conversely, the requester who receives any information or service can verify that the information is coming from a trusted source. Based on a requester's authentication, authorization may be granted such that information meant for a particular client can only be accessed specifically by that client. Because of the multiple hops in a Web service environment, it is possible that the information requestor and the service provider do not share the same authentication or authorization infrastructure. SAML, which is an XML based infrastructure, may be implemented to handle this challenge. In non-repudiation, clients should not be able to reject transactions that were initiated by them. Confidentiality ensures that the exchanged information is protected against interception. Integrity provides the assurance that the message was not modified deliberately or accidentally during transit.

Security measures in Web services environment should be implemented to ensure that data can only be accessed by authorized users, or to provide a certain level of assurance on the identity of service processes when a client is about to pass sensitive information. The rest of paper is organized as follows. We first review Web services standards. Then, we examine security challenges specific to Web services and Web services security standards. Based on our investigation, we provide guidelines for matching security standards to business needs. Finally, we discuss the limitations in the current practice of Web services security and suggest future directions.

WEB SERVICES STANDARDS

The current web environment is "human-centric" in that a typical user would interact with a merchant web-site which lacks integration with the entire supply chain. Web services offer the capability of integrating the processes and becoming more "application-centric". The "application-centric" approach will allow various business applications from different service providers to share data using XML standards. Web services typically consist of four main components: Web service consumer, Web service provider, business agreement, and registry, as shown in Figure 1.

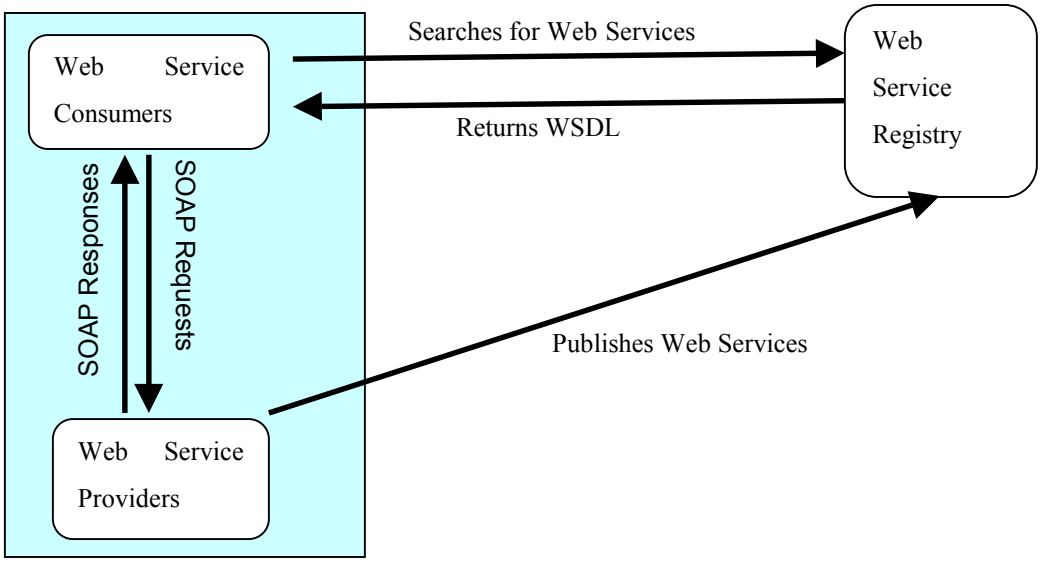

Figure 1: Web Services Architecture

The Web service consumer or client program makes a request to, and gets a response from one or several Web service providers. The Web service provider is a company or organization hosting a program in Web service that can receive requests from and send response to Web services consumers. A business agreement or document defines the functionality of a Web service and how that functionality can be accessed by consumers. The fourth component is a registry where Web service providers publish their services and Web consumers search for a web service.

Web services must be published to the Internet community by the Web service provider in order to make them available to potential consumers. Web Services Description Language (WSDL) is a standard XML based vocabulary used to describe Web services, which are then published using the Universal Description, Discovery and Integration (UDDI) protocol, another part of the XML vocabulary. The purpose of WSDL is to provide a file that a client can use to invoke the Web service and its functions. A WSDL document simply consists of a set of definitions. The UDDI protocol is one of the major building blocks required for successful Web services [18]. The UDDI enables businesses to quickly, easily, and dynamically discover and interact with one another independent of the platform, by using applications that they are familiar with. In addition, the UDDI registry allows virtual business functions to be integrated regardless of the internal processes of each business entity as long as they meet the Web services standards. Simple Object Access Protocol (SOAP) is a lightweight, XML based messaging protocol framework for building and exchanging distributed, structured information in a decentralized and distributed environment. There are three parts in a soap message -an envelope, a set of encoding rules, and a convention for representing remote procedure calls and responses.

Despite web services standards, custom implementations by independent vendors has resulted in interoperability problems. The Web-Services Interoperability Organization (WS-I) announced the first set of basic interoperability specifications in August 2003. WS-I Basic Profile 1.0 is a set of guidelines and recommendations covering the above-mentioned core web service standards. This is a crucial step toward achieving interoperable web services [10].

Most of the Web services transactions are messaging based. SOAP has been extensively used as the messaging protocol. In the next section, we will primarily focus on the security issues involved in the SOAP messaging between Web service consumers and Web service providers, as highlighted in Figure 1.

WEB SERVICES SECURITY CHALLENGES

In a Web services environment, multiple trading partners are accessing applications across disparate corporate firewalls using multiple devices and involving multi-step processes [3]. Compared with regular business transactions on the Web, Web services pose new challenges to security management, which are selectively listed as follows:

- A security method should support single sign-on schemes. Without such mechanisms in place, each trading partner has to maintain its own authentication and authorization, which may greatly compromise the convenience of Web services.
- A security method needs to consider the security implications of supporting multiple devices (e.g. Personal Digital Assistants, 3G cell phones). For example, wireless standards such as GSM and WAP do not offer end-to-end security [5].
- A security method should ensure confidentiality and integrity of the transactions in a multi-step process.

The above security challenges specific to Web services are not addressed by the current point-to-point content security technologies such as SSL, VPNs, and IPSec. Among a number of security implications resulting from Web services, we select information integrity and trust management to illustrate the limitations of traditional security standards.

Information integrity provides the assurance that messages were not modified deliberately or accidentally during transit. A traditional environment may involve only two partners. SSL supports transport layer security between two SSL-enabled parties. It is based on point-to-point connection sessions, and thus each SSL session is unique. When the data are not "in transit" between the two parties, it is not encrypted and thus not secure. The data is also vulnerable to attacks when it is between two SSL sessions in a multi-party or multi-step operation. Therefore, SSL is not sufficient for ensuring integrity in a multi-party or multi-step Web service transaction. Another drawback of SSL is that it is designed to encrypt the entire document before transmitting it. However, in a Web services environment, there may be a need to only secure part of a document. For example, a merchant should only be allowed to read information in a message that is pertinent to them. Other point-to-point technologies such as VPN and IPSec suffer from the same inherent weaknesses.

Web services across enterprises are dealing with potentially un-trusted clients. Trust management in a multiple-partner transaction must be more dynamic in order for distributed networks to scale [12]. By definition, Web services are loosely coupled. As such, authorization policies are more difficult to implement by individual merchants. Nonetheless, there is still a need to identify if a caller is authorized to request for a Web service. While the commonly used trusted third party approach on the Web [3] allows end-to-end agreement with security policies, it does not manage versioning for a long-duration operation. It has been recently proposed that Web services themselves may be used to provide trust services [2].

In sum, Web services require a finer grained security protocol beyond the traditional security standards. In the next section, we will discuss Web services security standards that are developed to resolve some of the security challenges.

WEB SERVICES SECURITY STANDARDS

Web services are situated at the application layer within the network protocol stack. Therefore, it is important to secure the perimeter, network, and host to reduce threats. Since Web services are XML based, in principle, existing security mechanism should supplement XML security standards to ensure that Web services are secured. We summarize some of the existing key standards for Web services security from the XML based framework in Figure 2. As Web services rely on the Internet to operate, the non-XML frameworks, which are situated underneath the application layer, can still be used to secure the communication of Web services. Thus, we also include some standards from the non-XML framework in Figure 2. Next, we elaborate on the key features of each of the XML framework based security standards.

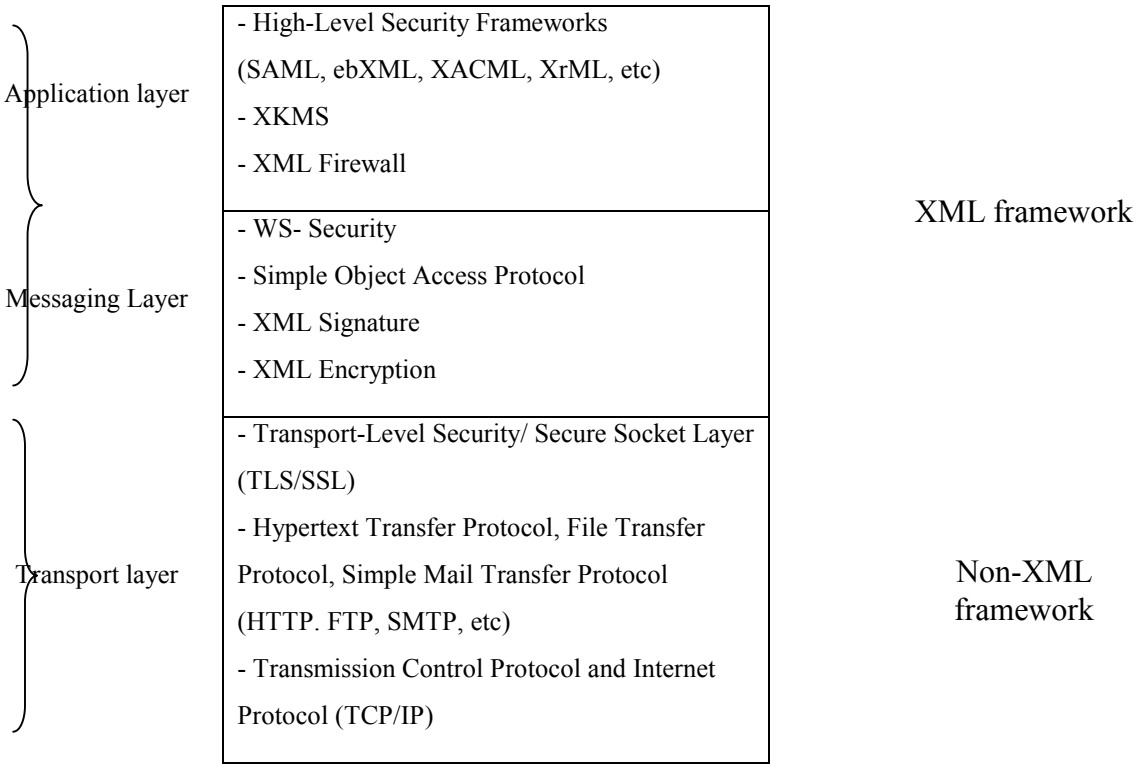

Figure 2: Web Services Security Standards

Security Assertion Markup Language (SAML) defines an XML-based protocol by which clients can request assertions from SAML authorities and receive responses from them. SAML assertions can be digitally signed. The assertions can convey information about whether accesses to resources are allowed. It allows disparate security systems to interoperate with each other. However, SAML assumes trust between the participants and it defers this responsibility to XML Encryption and XML Digital Signature. There are three kinds of SAML assertion statements: authentication, attribute, and authorization decision. Electronic Business XML (ebXML) is an initiative between OASIS and UN/CEFACT (the United Nations Center for Trade Facilitation and Electronic Business). ebXML specifies a framework for the exchange of electronic business data. Extensible Access Control Markup Language (XACML) is an XML specification for expressing policies for information access over the Internet. It provides a fine-grained access control to XML documents and other documents shared on the internet. XACML differs from SAML in that SAML assertions are formulated at runtime. Extensible Rights Markup Language (XrML) provides a method for specifying and managing the rights and policies associated with digital content and services. Compared with XACML, XrML is easier to use but at the cost of expressivity and flexibility.

WS-Security [9] describes the enhancements that can be used to accommodate a variety of security models and encryption technologies. The goal is achieved by providing quality of protection of SOAP messaging. Web services have typically relied on traditional security methods such as SSL and firewalls. However, SOAP messages can easily pass through firewalls and therefore requires additional security solutions. This can be achieved with WS-security. The WS-Security specification proposes a set of SOAP extensions that can be used when building secure Web services to ensure integrity and confidentiality. The purpose of WS-Security is to allow SOAP messages to be constructed securely. It also supports a variety of security models including Public Key Infrastructure (PKI), Kerberos, and SSL. Among others, security token propagation, message integrity, and

message confidentiality are three mechanisms provided by this specification.

Data can be digitally signed. An XML Signature is defined within the <Signature> element. XML Signatures provide integrity, message authentication, and/or signer authentication services for data of any type, either located within the XML (including the signature) or elsewhere. It also provides XML-compliant syntax for representing the signature for Web resources and portions of protocol messages and procedures for computing and verifying such signatures. XML signatures provide proof for non-repudiation of who created it [13]. XML Encryption is a process of encrypting data and representing the result with XML. The main element is <Encrypted Data>. XML Encryption defines how digital content is encrypted and decrypted, how the encryption key is passed to a recipient and how encrypted XML data and non-XML data are identified. The result is an XML encryption element that contains or references the cipher data. Both XML Encryption and XML Signature can be applied in any sequence. If XML encryption is applied after XML signature, the XML signature will not be immediately verifiable but becomes verifiable after it is decrypted. The XML Signature specification includes information to identify documents that are encrypted either after signing or before signature. XML Encryption and XML Signature also provide the ability to encrypt only selected portions of a document. This is an enhancement over SSL which encrypts the entire document before it is transmitted.

Current corporate firewalls can only filter at the packet level but not at the content level. They typically work by blocking TCP ports except for port 80 (HTTP traffic), port 25 (email traffic), and port 443 (HTTPS traffic). This raises a serious security issue as many Web services are designed to pass through port 80. XML firewalls are designed to examine XML content of the incoming traffic. XML firewalls may be hardware or software based. XML Key Management Specification (XKMS) defines protocols for distributing and registering public keys which may be used in conjunction with XML Signature and XML Encryption standards. It also describes how a client may receive key information from a Web service. Other researchers have suggested incorporating a semantics-aware firewall at the SOAP level and at the lower network-based layers [6].

A Web Services Security Scenario

In this section, we use a scenario (Figure 3) to illustrate how the Web services security standards can be applied in a typical Web services environment.

Joe Shopper (either a person or an agent), a Web services consumer, is interested in purchasing products A and B from MyShopping.com. Joe Shopper submits an order of products A and B at MyShopping.com's purchasing portal using an HTML form. The portal authenticates the user and creates two SOAP messages, a billing SOAP message to MyBilling.com and a shipping SOAP message to MyShipping.com. MyShopping.com can also be considered as a consumer of the services provided by MyShipping.com and MyBilling.com. The SOAP message to the billing Web service includes a WS-Security header enclosing Joe Shopper's name and the password digest. The SOAP envelope includes the payment information encoded with XML Encryption to hide his credit card number. If the billing is completed successfully, MyBilling.com will return a SOAP message to MyShopping.com with the billing status, which in turn submits shipping SOAP message to MyShipping.com. The SOAP message for the shipping Web service includes a WS-Security header enclosing a SAML assertion, which may be signed using XML Signature. The SOAP envelope includes the list of items to be shipped to the consumer. MyShipping.com returns a SOAP message in response to MyShopping.com once the shipping process is completed.

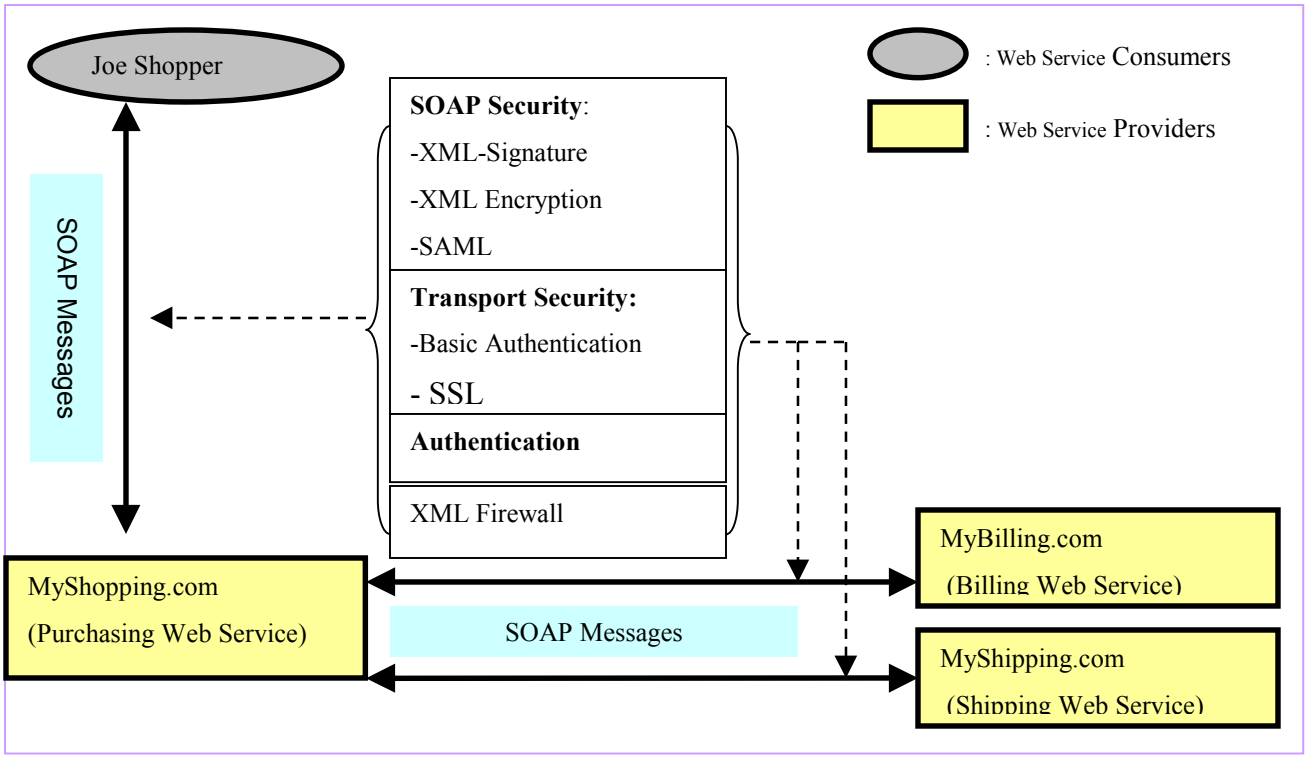

Figure 3: A Web Services Scenario Enhanced with Web Services Security

MATCHING SECURITY STANDARDS TO BUSINESS NEEDS

The implications of Web Services innovation for general adopters are significant [21]. Web services can lead to e-service interactions within and across organizations. In view of the unique environment and strategy of individual service providers, security considerations should be customized to meet business goals. There is a wide variety of security threats, including un-authenticated sender, un-authorized receiver, denial of service, dictionary attack, replay attack, token substitution, message modification, man-in-the-middle attack, domain name server attacks, Trojan horse, and so on. In security management, it is practically impossible to prepare for countering against each and every type of security threat. A cost-effective strategy is to align security-enhancing solutions with business needs. The focus should be placed on reducing the exposure and spread of the risk. However, web services security still rely on traditional security solutions. Some attacks such as Denial of Service may be addressed by traditional mechanisms. To provide some guidance for the selection of XML standards in Web services applications, we match various XML security standards to business security goals in Table 1.

Table 1: XML Security Standards and Their Goals

XML Standards	Security Goals
WS-Security	Authentication, Confidentiality
SAML	Authentication
XML Digital Signature	Authentication, Integrity, Non-repudiation, Audit, Trust
XML Encryption	Confidentiality, Integrity
XKMS	Confidentiality, Non-repudiation, Audit, Integrity, Trust
XACML	Authorization
XrML	Authorization

Businesses or organizations should take advantage of various technologies available to meet the

security requirements as they plan for the launch of Web services. Some standards such as XKMS address broader security goals while others such as SAML are more goal-oriented. It is not uncommon for managers to implement more than one of the above standards to ensure the Web services security.

DISCUSSION

Security is a major concern of potential adopters of Web services. Organizations using the traditional HTTPS or TLS/SSL should be aware that their security measures need to be updated to meet the needs of Web services. Moreover, emerging standards for Web services security should be selected cautiously to match them to business security goals. IT managers need to be careful that their security goals are not unnecessarily complicated by coding for all possible security threats. Due to the limitations of the individual standards for Web services security, it is necessary to combine multiple standards in implementing Web services security. For example, SAML assumes trust between trading partners, which should be supplemented by XML Signature and XML Encryption. Also, XrML and XAML manage authorization but do not address authentication, which can be complemented with encryption and digital signature protocols. Finally, Web services security should be integrated with the enterprise environments [14]. The recipients of SOAP messages with the accompanying security information should provide an environment where the messages can be processed and acted upon.

We have focused on the implementation of Web service security in this paper. To achieve business success with Web services, an organization should also take into account a number of other factors as follows:

- Financial considerations. Web services will provide a new channel for developing new markets. Gartner predicts that in 2004, sales in the Web services market is expected to grow to $28 billion [7]. However, Web services are geographically independent. It is important that critical business data and proprietary processes are secured from unauthorized access. A breach in corporate data integrity will result in serious financial consequences.
- Legislative Compliance. Government legislation increasingly requires that consumer data are not revealed without the permission of their owners. Insurance companies, hospitals and organizations working with patient data need to ensure compliance with The Health Insurance Portability and Accountability Act of 1996, known as HIPAA. HIPAA is expected to cost the healthcare industry at least $3.8 billion between 2003 and 2008 [1].
- Privacy. Using SOAP messages, data are being exposed increasingly as it moves over the insecure Internet. Any breach of data privacy may result in the loss of trust from consumers and business partners.

CONCLUSION

While traditional security infrastructures can still be used to support Web services, they cannot address the special challenges posed by Web services, which is to allow multiple parties to access multi-step processes across disparate corporate firewalls using multiple devices. Therefore, developing and adopting XML-based web services security standards is important to leverage the capabilities of web services. Given a variety of emerging Web services security standards, a service provider should not choose security solutions randomly. The framework for matching the security standards to the business security goals, as proposed in this paper, provides some guidance to assist service providers in choosing security solutions strategically.

Web services are already being implemented and are becoming a normal part of daily electronic transactions. Widespread implementation of web services and the generation of new revenue streams rely on ensuring that Web services security requirements are met satisfactorily. Moreover, it is important to emphasize that security of any business application must be integrated into the overall security plan of the firm. Standards for Web service security are still evolving. With the integration of web services security into enterprise security, we can envision the roll-out of novel Web services in the future.

REFERENCES

[1] Beaver, K. and Herold, R. "HIPAA Cost Considerations", *The Practical Guide to HIPAA Privacy and Security Compliance*, Auerbach Publications, 2003, Chapter 3, pp. 23-33.

[2] Baldwin, A., Shiu, S. and Mont C, M. "Trust Services: A framework for service-based solutions," *Proceedings of the 26th Annual International Computer Software and Applications Conference* (COMPSAC'02), 2002, pp. 507-513.

[3] Chang, S., Chen, Q. and Hsu, M. "Managing Security Policy in a Large Distributed Web Services Environment," *Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03),* 2003, pp. 617-622.
[4] Chen, M. "An Analysis of the Driving Forces for the Adoption of Web Services," *e-biz Web Workshop*, Seattle, WA, Dec 13-14, 2003, pp. 173-184.
[5] Claessens, J., Preneel, B, and Vandewalle, J. "Combining World Wide Web and Wireless Security," *Informatica* 26, 2001, pp. 123-132.
[6] Cremonini, M. and Damiani, E. "An XML-based Approach to Combine Firewalls and Web Services Security Specifications," *ACM Workshop on XML Security*, October 31, 2003.
[7] Gartner, "Gartner Says Web Services Will Dominate Deployment of New Application Solutions for Fortune 2000 Companies by 2004," *Gartner*, January 14, 2002.
[8] Hanna, J. "Web Services," http://hbsworkingknowledge.hbs.edu/pubitem.jhtml?id=3285&sid=-1&t=special_reports_cyber2003, Feb3, 2003.
[9] Khaler, C. "WS-Security," http://www-106.ibm.com/developerworks/webservices/library/ws-secure/, 2002.
[10] Kumar, K. M. S., Das, A. S., and Padmanabhuni, S. "WS-I Basic Profile: A Practitioner's View," *Proceedings of the IEEE International Conference on Web Services* (ICWS'04), 2004, pp. 17-24.
[11] Long, J., Yuan, M. and Whinston, A. "Securing a New Era of Financial Services," *IT Pro*, July | August 2003, pp. 15-21.
[12] Morioka, M., Yonemoto, Y., Suzuki, T. and Etoh, M. "Scalable Security Description Framework for Mobile Web Services," *IEEE International Conference on Communications,* 2003, pp. 804-808.
[13] Naedele, M. "Standards for XML and Web Services Security," *Computer*, Volume 36, Number 4, 2003, pp. 96-98.
[14] Nakamur, Y., Hada, S. and Neyama, R. "Towards the integration of Web Services Security on Enterprise Environments," *Proceedings of the 2002 Symposium on Applications and the Internet (SAINT'02w),* 2002, pp. 166-175.
[15] Netegrity, "Netegrity Web Services Survey Result," http://www.netegrity.com/txmindersurvey/TxMSurveyAnalysis.html, Dec 08, 2003.
[16] OASIS "SAML Version 1.1.", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, 2003.
[17] Reagle, J. "XML Encryption Requirements," W3C, http://www.w3.org/TR/xml-encryption-req, 2002.
[18] UDDI, "UDDI Executive White Paper", http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf, November 14, 2001.
[19] W3Ca, "XML Signature Syntax and Processing," http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, February 2002.
[20] W3Cb, "Web Services Activity," http://www.w3.org/2002/ws/, January 2002.
[21] Xu, H., Seltsikas, P. and O'Keefe, B. "The Implications of Web Services Innovation for General Adopters: Findings and Recommendations," *Proceedings of the Second Workshop on e-Business (Web),* Seattle, WA., Dec. 13 – 14, 2003, pp. 160-172.
[22] Zimmermann, O. and Craes, M. "Second Generation Web Services-Oriented Architecture in Production in the Finance Industry," *Conference on Object Oriented Program Systems Languages and Application*, 2004, pp. 283-289.

AUTHORS' BIOGRAPHIES

Kenny Khoo is a Ph.D. student of Information System at UMBC. His research interest is in areas of database development and knowledge management particularly in the area of biomedical databases. He is also interested in areas of security and trust on the internet. He received his MBA from the University of Maryland College Park, and his Bachelors of Science from the National University of Singapore.

Lina Zhou is an Assistant Professor of Information Systems at UMBC. Her research interest is in text mining, deception detection, ontology and Semantic Web, and Web services. Her work has been published in journals such *as Journal of Management Information Systems*, *Communications of the ACM, IEEE Transactions on System, Man, and Cybernetics, Decision Support Systems*, among others.