Security in XML-based financial reporting services on the Internet

J. Efrim Boritz *, Won G. No

School of Accountancy, University of Waterloo, Waterloo, ON, Canada N2L 3G1

Abstract

Many companies are attempting to leverage the power of financial information by creating corporate websites to provide such information to employees, investors, and financial analysts. Extensible Business Reporting Language (XBRL) was developed to provide users with an efficient and effective means of preparing and exchanging financial information over the Internet. Extensible Assurance Reporting Language (XARL) was designed to enable assurance providers to report on the integrity of information distributed over the Internet and help users and companies place warranted reliance on such information.

The XBRL and XARL services are Internet-based message exchange methods. The Internet is insecure in nature. Without good security, XBRL and XARL services will not reach their full potential. Today’s security approaches consist of a combination of user IDs and passwords and point-to-point, transport-level security for data transmissions over the Internet such as SSL/TLS, S-HTTP, and VPN. Access control techniques based on user IDs and passwords can protect files or data from unauthorized access but cannot guarantee the integrity of the information. Transport-level, point-to-point security is not sufficient for securing information that travels between several intermediaries or for encrypting only selected portions of an information set. Thus, alternative security approaches are needed to compensate for these limitations.

* Corresponding author. Tel.: +1 519 888 4567x5774; fax: +1 519 888 7662.
E-mail address: jeboritz@watarts.uwaterloo.ca (J.E. Boritz).
This paper addresses security in financial reporting services. First, it describes Web services and conceptualizes financial reporting services such as XBRL and XARL as Web Services. Next, it discusses security threats and limitations of current security technologies. Then, it identifies security requirements that should be considered to ensure reliable, trustworthy XBRL and XARL services. Finally, the paper explains several proposed security standards and suggests Web Services Security Architecture as a suitable security mechanism for financial reporting services.

© 2004 Elsevier Inc. All rights reserved.

Keywords: XBRL; XARL; Security; Information integrity; Web services

1. Introduction

Rapid advances in computer and communication technology have revolutionized the way business is conducted and the way in which financial information is disseminated. Many organizations are currently attempting to provide financial information over the Internet, especially the World Wide Web (hereafter, the Web), to increase users’ access to information as well as to allow the timely and efficient exchange of information between preparers and various stakeholders. However, this often requires tedious, costly and error-prone cutting and pasting or transcription due to the lack of common, generally accepted formats for presenting business information.

Extensible Business Reporting Language (XBRL) was developed to overcome this limitation. XBRL provides financial information users with a standardized method to prepare, publish, and exchange financial information. It also offers technology independence, full interoperability, efficient preparation of financial statements, and efficient extraction of financial information for analysis purposes (Hoffman and Strand, 2001; Boritz and No, 2003a). Extensible Assurance Reporting Language (XARL), an extension of XBRL, was developed to enable assurance providers to report on the integrity of XBRL-tagged information distributed over the Internet and help users and companies place warranted reliance on such financial information (Boritz and No, 2003b).

Despite the enthusiasm about XBRL, there is a growing concern over the security of financial information on the Internet. The Internet is insecure. Without proper security, financial information on the Internet, especially the Web, can be intercepted, spoofed and altered. Therefore, fundamental security issues must be considered to ensure reliable, trustworthy electronic transmission of financial information on the Internet.

In this paper, we address security in XML-based financial reporting services on the Internet. The paper makes three contributions to the growing financial and accounting literatures on security and financial reporting
over the Internet. First, it conceptualizes XBRL and XARL as financial reporting services within a Web services framework. Second, it identifies security threats and requirements, which can help financial professions and financial service developers to consider issues for achieving an appropriate security protection for financial reporting services. Finally, the paper suggests a security solution for financial reporting services based on evolving standards.

The paper is organized into five sections. Following this brief introduction, Section 2 briefly describes XBRL and XARL. Section 3 outlines XBRL and XARL financial reporting Web services. Then, the main topic of this paper, security in financial reporting Web services is discussed in Section 4. Finally, conclusions and some issues for future consideration are presented in Section 5.

2. A new paradigm for financial reporting on the Internet: XBRL and XARL


The advent of the Web enables information providers to easily and cheaply distribute electronic documents to audiences on the Internet. Today it is generally accepted that the Web is a main channel for the global delivery of information from person to person, from business to customer, and from business to business. Extensible Markup Language (XML) was established by the World Wide Web Consortium (W3C) in 1996 to extend Web technology (Bosak and Bray, 1999) to enable the efficient exchange of information over the Web. The basic structure of XML is similar to HTML in many respects. XML documents consist of XML elements, which are a start tag, an end tag, and the content between two tags. However, unlike HTML, XML contains contextual information. For example, recipients of `<CompanyName>Waterloo Inc.</CompanyName>` can understand, decode, and use it for their own purposes since a `<CompanyName>` tag indicates what each item of data is. Content therefore becomes more readily accessible through XML.

XML enhances the power and flexibility of Web applications and other business software packages (W3C-XML in 10 points; Halfhill, 1999; Elliotte, 2001). First, it facilitates effective and efficient data exchanging because XML-coded data is self-describing so that data can be exchanged and processed without modification. XML also offers more meaningful searches. Since XML provides context information by coding information with special tags that describe content and structure, searches can generate more accurate and relevant results. Furthermore, a user can view data in various ways by using
structural information in XML data. Finally, XML is an open standard and is system and platform independent, creating a universal way for both formatting and presenting data.

2.2. XBRL: A standard way of preparing and exchanging financial information

XBRL is the financial profession’s adaptation of XML for financial reporting. All financial data are attached to tags that distinguish them as asset, liability, capital, profit, and so forth. Therefore, users can easily find the data with the tags (e.g., cash), extract or transform the data, and analyze the data with analytical applications. A more extensive discussion of XBRL can be found in Hoffman and Strand (2001) and Boritz and No (2003a).

2.3. XARL: Assurance for XBRL documents

The integrity of financial information contained in an XBRL document depends on the reliability of the processes used to create the document, the nature and extent of assurance procedures performed on that information, and the security measures taken to protect the integrity of the information. Extensible Assurance Reporting Language (XARL), an XML-based extension of XBRL, was designed by the authors to enable assurance providers to report on the integrity of XBRL-tagged information distributed over the Internet and help users and companies place warranted reliance on such financial information. With an appropriate infrastructure, XARL can provide a method for preparing and providing assurance about the integrity of financial information contained in XBRL documents.

Similar to XBRL, XARL defines elements to represent assurance information and supporting components to integrate with XBRL in order to allow users to make an informed judgment about the degree of trust to give each data item received. In other words, an XARL document includes tags that indicate assurance type, assurance date, auditor’s digital signature, system reliability, and so forth. A more extensive discussion of XARL can be found in Boritz and No (2003b and 2004a).

2.4. How XBRL and XARL work

Suppose a public company, Toronto Inc., wishes to provide financial statements to creditors, investors, and analysts, and engages an assurance services company, Waterloo Assurance Co. to provide assurance on that information. Fig. 1 depicts how XBRL and XARL would be used.

After Toronto Inc. prepares its financial information using its internal accounting system, an XBRL document is created by mapping the financial
information to XBRL taxonomy elements. The XBRL taxonomy is obtained from the XBRL organization (XBRL.ORG) over the Internet. Verification software is used to check that the document is a valid XBRL document. Then, the verified XBRL document is placed on the company’s Web site or FTP server ( ). Also, the document is sent to the assurance provider, Waterloo Assurance Co., over the Internet with appropriate security ( ).

When users need the information contained in the XBRL document for their analysis, they obtain it from Toronto Inc. over the Internet and use the XBRL document for their analysis ( and ). If they want to transform the document into HTML, a spreadsheet, or database, they can do so with appropriate style sheets developed by them or by outside software developers ( ).

Although users can obtain XBRL documents from the website of Toronto Inc. or another source such as Edgar Online, there is a question about the

---

1 An XBRL taxonomy is a dictionary of the financial terms used in preparing financial statements or other business reports and the corresponding XBRL tags (Engel et al., 2003). It defines tags corresponding to concepts that can be referenced in XBRL documents; for example, the tag with the name “CashCashEquivalentsShortTermInvestments” represents such a concept.
integrity of such XBRL documents, since XBRL tags can be misapplied and modified without proper authorization. This is particularly an issue if the XBRL document purports to have been audited by an independent accountant. Therefore, to obtain reliable financial information, users should obtain XBRL documents that are assured by Waterloo Assurance Co. It is important to note here that the assurance referred to here goes beyond the assurance provided on Toronto Inc.’s financial statements, to encompass the XBRL-coded version of those financial statements. To provide such assurance, Waterloo Assurance Co. (WAC) performs assurance procedures such as confirming the reliability of the processes that produced the information, performing analytic procedures on the data, and checking the validity of the XBRL tags. WAC then creates an XARL document by mapping assurance information related to the business reporting elements in the XBRL document to agreed-upon XARL taxonomy elements. The XARL taxonomy is obtained from the XARL organization over the Internet. Users who need reliable, assured financial information can request XBRL and XARL documents from WAC. This can be done through an on-line or off-line process.

Upon receiving a request for information, the Authentication System at WAC sends the XBRL and XARL documents to users, if the users are authorized to receive the information. The XARL document is digitally signed by WAC using its private key and, if appropriate, also encrypted with the user’s public key. Then, the encrypted XBRL and XARL documents are sent to the user.

Using its private key, and the assurer’s public key, the user decrypts the XBRL and XARL documents and uses them for its purposes. Also, the user can authenticate the XBRL and XARL documents by verifying the digital signature of the assurer. Since the digital signature is an unforgeable data element, the digital signature authenticates the integrity of that document as well as the identity of the sender who attached his or her signature to the document. If users want to translate the document into HTML, a spreadsheet or database, they can do so with appropriate style sheets developed by them or by outside software developers.

In summary, XBRL can provide a standardized method to prepare, publish, and exchange financial information. XARL provides evidence not only that the XBRL-coded information was audited by a public accountant but that it has not been tampered with. Therefore, uncertainty about the integrity of the financial information is significantly reduced.

---

2 An XARL taxonomy is the template that defines the assurance tags and describes how assurance data are interrelated. The XARL taxonomy is designed not only to deliver a framework for consistent creation of XARL documents, but also to provide financial information users with a standardized method to ensure the reliability of the information provided in XBRL-coded documents.
3. Financial reporting as a Web service

3.1. Service-oriented architecture (SOA) and Web services

The competitive nature of e-commerce is forcing all companies to cut costs and enhance interoperability among computer systems through adoption of Service-Oriented Architectures (SOA). A Service-Oriented Architecture (SOA) is a way of designing a software system to provide services to either end-user applications or other services. The goal of a SOA is to achieve loose coupling among interacting software applications (He, 2003). It is the movement from a technology-oriented solution to a service-oriented solution (Stevens, 2004).

A Web service is a form of SOA, 3 that is intended to enable developers to create services (or applications) that can be assembled and deployed in a distributed environment (Cardellini et al., 2002; IBM developerWorks). According to W3C, a Web Service is a software system that enables applications to communicate with each other in a platform- and programming-language-independent manner. It is a technology that accepts requests from different systems over the network through the application of Web technology standards including XML (Bray et al., 2004), SOAP (Gudgin et al., 2003), WSDL (Chinnici et al., 2003), and UDDI (Januszewski, 2001; Seely, 2001).

In general, Web services consist of three main operations (Booth et al., 2004). First, Web services expose their functionality (service) to users through a standard messaging protocol known as Simple Object Access Protocol (SOAP). 4 Second, Web services provide a mechanism to describe the services a company offers and a way for users to access those services. The description of the service is usually provided in a XML document called a Web Services Description Language (WSDL) document. 5 Finally, Web services

---

3 A Web service is a set of code implementing the XML interface that describes a collection of operations that can be accessed over the Internet through standardized XML messaging. Therefore, a group of Web services interacting together in this manner can be defined as a Service-Oriented Architecture.

4 Simple Object Access Protocol (SOAP) is an XML-based communication protocol that defines a format for sending messages. It was developed to facilitate communications between applications over the Internet. Thus, it provides the mechanisms for information exchange among applications running under different operating systems in a network by using XML and HTTP.

5 Web Services Description Language (WSDL) is an XML document that describes a set of SOAP messages and how the messages are exchanged. It specifies what a request message must contain and what the response message will look like.
are registered so that potential users can find them easily. This is done with Universal Discovery Description and Integration (UDDI). In summary, Web services are developed to bridge communications gaps between software written in different programming languages, developed by different vendors and running on different systems. This interoperability leads to several benefits to organizations: enhancing the flexibility in software development, increasing agility and productivity, saving developer time and cost, and leveraging existing IT investments.

3.2. XBRL and XARL as financial reporting services

The primary goal of XBRL and XARL is to provide a standardized mechanism to prepare, publish, and exchange reliable financial information over the Internet between different software applications running on a variety of platforms and frameworks. Thus, XBRL and XARL can be conceived of as XML-based message exchange application services. These services not only can perform activities on their own, but also can engage other services such as FpML and FIX in order to complete higher-order transactions.

3.3. How XBRL and XARL Web services work

Fig. 2 depicts how financial reporting services work.

Suppose a financial analyst, Harry, wishes to analyze three companies’ financial statements (Ford, GM, and Chrysler) for his investment decision by importing current financial information such as accounting ratios and current stock prices into an Excel spreadsheet. Assume that an assurance provider (Waterloo Assurance Co.) provides an XARL service and that Kitchener Service Co. (hereafter, KSC) is an entity that provides a UDDI service.

After WAC obtains each company’s XBRL documents, it generates XARL documents and stores them in its database. Using a provider agent (software
application) via a UDDI interface, WAC publishes a SOAP-encoded XARL service description to KSC, a discovery registry (1 and 2). Once the XARL service description is published, it will be used by Harry, the service requester, to access the XARL service. A XARL service description and service semantics (WSDL file) define the functionality of the XARL service, the XARL service semantics and the XARL service interface (i.e., how to interact with the XARL service).

When Harry wants to get the financial statements of Ford, GM, and Chrysler for his analysis, he can request an XARL service by submitting his request to a requester agent (3). Then, the requester agent communicates with KSC (the UDDI service provider) via the UDDI interface and transmits the service request.

Next, if the discovery agent at KSC accepts the request from the service requester agent, it searches the discovery registry for an appropriate XARL

Fig. 2. Financial reporting services.

11 Service semantics is the contract between the service provider and the service requester that expresses the requirements pertaining to the use of a service and its effects (i.e., what the service does and how to access it).
service. Once the discovery agent finds WAC’s XARL service, the discovery agent retrieves WAC’s XARL service description and service semantics from the discovery registry, and sends them to the requester agent (4). Using the information obtained from KSC, the requester agent sends a SOAP-encoded XARL request message to WAC to directly bind to the service and invoke it. Then, WAC provides Harry with the XARL service based on the service semantics and the service description. That is, the service agent at WAC sends the financial statements of Ford, GM, and Chrysler to the requester agent (5).

Finally, Harry’s application (Excel) loads these financial statements and calculates the financial ratios of the three companies. Once the XARL service binding between the requester agent and the provider agent is established, the financial information in Excel can be updated automatically. Through similar steps, Harry can load the current stock price of each company into his Excel spreadsheet by requesting a stock price quoting service from other Web service providers. In other words, the service requester agent at Harry performs this function automatically and simultaneously by engaging the stock price quoting service based on his service request while he is getting the XARL service from the WAC.

4. Security in financial reporting services

4.1. Security threats in financial reporting services and limitations of current security technologies

Security issues related to the Internet are important due to the fact that the Internet is an insecure and unreliable public network. The insecure nature of the Internet attracts intruders and hackers. However, XML did not originally address security issues, and SOAP was also not designed with security in mind. Thus, financial reporting services based on these components are inherently insecure. Some of the major security treats in financial reporting services are message alteration, message disclosure (confidentiality), message substitution (man-in-the-middle attack), IP spoofing, denial of service, packet sniffing, and virus attacks (Bosworth and Kabay, 2002). Table 1 summarizes these threats.

Financial reporting services must provide financial information to a large number of users by interconnecting heterogeneous and distributed systems. However, security requirements should ensure that such services allow only authorized recipients to obtain the financial information and, where necessary, determine the identity of the user. Financial reporting service users, in turn, must be able to verify the integrity of the information being provided, to receive a message confidentially, to validate the identity of the service providers,
Table 1
Security threats in financial reporting services

<table>
<thead>
<tr>
<th>Security threats</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message alteration</td>
<td>Message alteration is an attack that modifies XML-coded message content (i.e., XBRL or XARL documents) in order to disrupt the information flow implied by the message. For instance, an attacker may modify part of an XML message, or insert extra information into an XML message, or delete part of an XML message.</td>
</tr>
<tr>
<td>Message disclosure</td>
<td>Message disclosure is an attack whereby an unauthorized user obtains access to information within a message or part of a message.</td>
</tr>
<tr>
<td>Message substitution (Man-in-the-middle attack)</td>
<td>Man-in-the-middle attack, also known as a bucket-brigade attack, involves intercepting the messages and masquerading as the parties concerned. For example, an attacker intercepts a XARL document that the provider agent sent to the requester agent, and replaces it with an alternate document. Thus, both agents will think that they are communicating with each other, and they may never know that the document is being intercepted and modified.</td>
</tr>
<tr>
<td>IP spoofing</td>
<td>IP spoofing is a technique that is used to gain unauthorized access to a system, whereby an attacker sends an XML message to the system with an IP address indicating that the message is coming from a trusted domain. For example, by sending an XARL service request message with an IP address of a trusted service requester, an attacker may obtain XARL documents from the XARL service provider.</td>
</tr>
<tr>
<td>Denial of service</td>
<td>Denial of service (DoS) is an attack design to prevent legitimate users from using a service, or to render a computer or network incapable of providing normal services. A DoS attack is a type of security breach to a system that does not usually result in theft or change of information. However, this attack can result in the use of non-current information if that is the default and can cause the target users or organizations a great deal of time, money, and reputation.</td>
</tr>
<tr>
<td>Packet sniffing</td>
<td>Packet sniffing is the act of intercepting and reading any or all network traffic that is being transmitted across a shared network communication channel. By using a packet sniffing tool, an attacker can capture a user ID and password when the service agents transmit them in an unencrypted XML message.</td>
</tr>
<tr>
<td>Computer virus</td>
<td>A computer virus is a malicious code (program) that can invade computer systems and perform a variety of functions such as deleting files, destroying the system, and opening up the system to hackers. A computer virus is called a virus because it shares some of the traits of biological viruses, passing the malicious code from computer to computer.</td>
</tr>
</tbody>
</table>

\(^a\) Transmission Control Protocol/Internet Protocol (TCP/IP) is the basic protocol suite used on the Internet. TCP/IP consists of a two-layer program. The higher layer, Transmission Control Protocol (TCP), manages the assembling and reassembling of a message (packet) that is transmitted over the Internet. TCP offers reliable, flow-controlled delivery. The lower layer, Internet Protocol (IP), defines both the format of packet and the mechanism for routing a packet to its destination. It handles the address part of each packet so that it gets to the right destination.
and to determine whether the service provider is authorized to provide the financial reporting services.

Today’s security approaches consist of a combination of user IDs and passwords and point-to-point, transport-level security for data transmissions over the Internet such as SSL/TLS, S-HTTP, and VPN (Bosworth and Kabay, 2002).

Access controls that rely on user IDs and passwords are not sufficient for financial reporting services because such controls only ensure that authorized users are able to access the services, but cannot provide the other assurances that providers and users require. Secure Sockets Layer (SSL)/Transport Layer Security (TLS) can be used to encrypt and decrypt messages exchanged among various parties on public networks such as the Internet, to protect messages from unauthorized access while they are in transit. In addition, with digital certificates, such as the X.509 certificate, as part of the authentication process, SSL/TLS can verify the authenticity of the sender and receiver. Another protocol for transmitting data securely over the Web is Secure HTTP (S-HTTP). S-HTTP is a secure message-oriented communications protocol designed for use in conjunction with HTTP. SSL/TLS and S-HTTP are complementary rather than competing technologies since they do not work on the same level. While S-HTTP makes it possible to transmit individual messages securely, SSL allows safe connection among various parties, so that any amount of data can be sent securely. A Virtual Private Network (VPN) enables an entity to have a private network, but use a public network as a carrier. VPN restricts traffic so that messages can only travel among specified hosts on the private network. These three techniques have proven to be robust and effective.

Although SSL/TLS, S-HTTP, and VPN can provide secure communication for XBRL and XARL transmissions if the appropriate infrastructure is in place, they are not sufficient for providing end-to-end security (MSDN, 2003a). For instance, SSL/TLS and VPN operate between communication endpoints, not between applications. That is, they are designed to provide transport-level, point-to-point security. In financial reporting services, a XML message may travel between various intermediaries before it reaches its destination. Therefore, it is difficult to create point-to-point security among those

---

12 The Secure Sockets Layer (SSL) developed by Netscape Communications Corporation to provide secured communication over the Internet. SSL has been succeeded by Transport Layer Security (TLS) which is standardized by the Internet Engineering Task Force (IETF).


14 For further information, visit IETF’s RFC 2660 Internet draft that describes S-HTTP (http://www.ietf.org/rfc/rfc2660.txt).

15 For further information, visit the Virtual Private Network Consortium website (http://www.vpnc.org/).
intermediaries, and is also expensive. In addition, since SSL/TLS, S-HTTP, and VPN provide transport-level security, they do not provide a method for only encrypting selected parts of a document. Thus, if a financial service provider wants to only encrypt a certain part of a document (e.g., the cash element in the balance sheet), it would be awkward/difficult to encrypt just that part with SSL/TLS, S-HTTP, and VPN.

In summary, message-level security, as opposed to protocol dependent transport-level, point-to-point security, is required to ensure the security of financial reporting services.

4.2. Security requirements in financial reporting services

As we described earlier, financial reporting services (i.e., XBRL and XARL services) involve an exchange of SOAP-encoded XML messages. Thus, it is important to provide a mechanism that satisfies basic message-level security requirements; for example, the recipient of a message should be able to ensure the integrity of the message, to receive a message confidentially, and to determine the identity of the sender. Several security requirements must be met to satisfy reliable, trustworthy electronic transmission of SOAP-encoded XML messages (Greenstein and Vasarhelyi, 2002). Table 2 summarizes these requirements.

Overall, to secure financial reporting services, these main security issues should be carefully considered, and a range of XML-based security mechanisms is needed to address these issues.

4.3. XML-based security standards

Several attempts are being made to ensure security over the Internet, especially for Web services. These approaches are XML-based, message-level security solutions, and can also be used for financial reporting services based on Web services. Table 3 summarizes these XML security standards.

Although these security standards may ensure security in financial reporting services, it might be difficult to integrate and implement all the security solutions due to the complexity of the business environment and infrastructure requirements. Furthermore, the presence of one vulnerable link in the security mesh can undermine the security of the remaining infrastructure.

4.4. A security infrastructure for financial reporting services

A Web Services Security Architecture (hereafter, WSSA) consists of a series of proposed standards from an industry group that includes IBM, Microsoft, and VeriSign to address security issues when data is exchanged as part of a Web service (MSDN, 2003b). WSSA enhances SOAP message exchange by
providing confidentiality, integrity, authentication, authorization, and trust (MSDN, 2002). Since WSSA provides an end-to-end security solution and deals with broad security issues in Web services, we believe that it is a suitable mechanism to achieve the security requirements for financial reporting services described previously.

WSSA includes several specifications: WS-Security, WS-Policy, WS-Trust, WS-Privacy, WS-SecureConversation, WS-Federation, and WS-Authorization.16 Fig. 3 illustrates the WSSA specification stacks.17 Also, Table 4 sum-

<table>
<thead>
<tr>
<th>Security requirements</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>When a sender transmits XBRL and XARL documents to a recipient through the Internet, the documents remain confidential. That is, only the sender and intended recipient can read the message.</td>
</tr>
<tr>
<td>Integrity</td>
<td>When a sender transmits XBRL and XARL documents to a recipient through the Internet, the documents have not been changed. In other words, XBRL and XARL documents received by the intended recipient are exactly the same as the documents transmitted by the sender.</td>
</tr>
<tr>
<td>Authentication</td>
<td>When XBRL and XARL documents are received by a user or system, the sender and receiver are who they claim to be.</td>
</tr>
<tr>
<td>Non-repudiation</td>
<td>When XBRL and XARL documents are sent to a receiver, the sender cannot later deny having sent the documents, and vice versa, the recipient cannot deny having received the documents.</td>
</tr>
<tr>
<td>Authorization (Access control)</td>
<td>Only authorized users are able to access the XBRL and XARL documents.</td>
</tr>
<tr>
<td>Key management</td>
<td>Encryption is used to maintain confidentiality of information transmitted over the Internet. Encryption involves the use of private and/or public cryptographic “keys” to encipher transmissions. It is important to ensure proper creation, storage, use, and destruction of each cryptographic key. Audit trails are also needed to trace user accesses and actions. They also can be used to ensure system integrity through verification.</td>
</tr>
<tr>
<td>Security enforcement mechanism</td>
<td>Financial service providers can define a security policy with varying privileges and enforce it across various platforms.</td>
</tr>
<tr>
<td>Audit trails</td>
<td>Audits trails are a series of records of system events such as user accesses and user activities. Audit trails can enhance user accountability by tracing the user’s activities, to reconstruct system events after a problem has occurred, to monitor problems, and to detect system intrusion.</td>
</tr>
</tbody>
</table>

17 The diagram is adapted from “Security in a Web Services World: A Proposed Architecture and Roadmap. (Microsoft and IBM)”.

Table 2 Security requirements for financial reporting services
Confidentiality, integrity, and authentication are the minimum requirements for controlling the security of a XARL service. WS-Security, WS-Policy, and WS-Trust provide a means for ensuring confidentiality, integrity, and authentication of financial reporting services while allowing them to operate with other Web services.

Table 3
XML Security Standards

<table>
<thead>
<tr>
<th>Security Standard</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Encryption</td>
<td>XML Encryption was developed to provide a method for transforming a XML document so that it is readable only by the intended recipients. The XML Encryption standard/specification defines a format, data model and encryption syntax for encrypting content, along with the information an intended recipient would be required to provide for decryption. With XML Encryption, users can encrypt not only the whole XML document but certain parts of a document as well. Furthermore, this specification supports XML Signature's selective signing, and interoperates with XML Schemas.</td>
</tr>
<tr>
<td>XML Signature</td>
<td>XML Signature is a specification designed to meet the special requirements for using digital signatures with XML documents. XML Signature defines the syntax required to sign only specific elements in a document, to add more than one signature to a single document, and to deal with the transformation of signed data.</td>
</tr>
<tr>
<td>Security Assertion Markup Language (SAML)</td>
<td>Security Assertion Markup Language (SAML) is a XML-based mechanism to exchange authentication and authorization information. SAML provides a common method for the sharing of security services among various parties, and it also allows various parties to securely exchange authentication, authorization, and profile information regardless of which security systems they use.</td>
</tr>
<tr>
<td>XML Key Management Specification ( XKMS )</td>
<td>XML Key Management Specification ( XKMS ) defines the protocols for registering and managing public keys used in public-key cryptography. XKMS is designed to simplify the integration of PKI and digital certificates which are used for securing Internet transactions.</td>
</tr>
<tr>
<td>Extensible Access Control Markup Language (XACML)</td>
<td>eXtensible Access Control Markup Language (XACML) is a specification that is used in conjunction with SAML. It provides a method for standardizing access control policies to be created and applied against XML documents. XACML defines a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are identified in XML.</td>
</tr>
</tbody>
</table>

*a For more information, visit the XML Encryption website (http://www.w3.org/tr/xmlenc-core/).

*b For more information, visit the XML Signature website (http://www.w3.org/tr/xmldsig-core/).

*c For more information, visit the OASIS website (http://www.oasis-open.org).

*d For further information, visit the XKMS website (http://www.w3c.org/2001/xkms/).

*e For further information, visit the OASIS website (http://www.oasis-open.org).
4.4.1. WS-Security in financial reporting services

The main objective of WS-Security is to provide end-to-end message-level security for SOAP messages. To achieve message integrity and message confidentiality, WS-Security leverages XML Encryption and XML Signature in conjunction with security tokens such as a username/password combination, a Kerberos ticket \(^{18}\), or an X.509 certificate. \(^{19,20}\)

4.4.2. WS-Policy in financial reporting services

WS-Policy describes the specific security requirements for financial reporting services. For example, it defines digital signature algorithms and the type of security tokens that the service accepts; i.e., Kerberos ticket or X.509 certificate. \(^{21}\)

---

\(^{18}\) Kerberos is a network authentication protocol for authenticating a request for a service and was developed in the Athena Project at the Massachusetts Institute of Technology (MIT). Under Kerberos, when users want to use a service, they have to get an encrypted ‘Kerberos ticket.’ Users request an encrypted ticket from an authentication process. Then, they use the ticket to request a particular service from a server. For further information, visit the Kerberos website (http://web.mit.edu/kerberos/).

\(^{19}\) A Public Key Infrastructure (PKI) is a model to securely exchange data through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. PKI consists of three main function parts: Certificate Authority (CA) that issues and verifies digital certificates, Registration Authority (RA) that is trusted by the CA to register or attest to the identity of CA users, and Certificate Repository (CA) that is a publicly accessible electronic database that holds information such as the certificates. The X.509 certificate is an industry standard for a digital certificate. The X.509 standard defines what information can go into a certificate, and describes the data format. For further information, visit the IETF website (http://www.ietf.org/rfc/rfc2459.txt).

\(^{20}\) An example of a SOAP message that contains a digital signature and that carries both encrypted data and the encrypted symmetric key needed to read that data is available from the authors upon request.

\(^{21}\) An example of an SOAP message that requires a Kerberos ticket for authentication is available from the authors upon request.
<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
<th>Security requirements addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-Security</td>
<td>WS-Security is the foundation for all other specifications. It defines extensions to SOAP messaging that provide message integrity and message confidentiality. These security requirements are provided by leveraging XML Encryption and XML Signature in conjunction with security tokens to ensure that messages are confidentially transmitted and that messages are transmitted without modifications.</td>
<td>• Confidentiality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Non-repudiation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Integrity</td>
</tr>
<tr>
<td>WS-Policy</td>
<td>WS-Policy specifies how senders and receivers can define their preferences and requirements. This specification defines a generic SOAP format, which can support security policies, as well as how to attach or associate service policies to SOAP messages.</td>
<td>• Security enforcement mechanism</td>
</tr>
<tr>
<td>WS-Trust</td>
<td>WS-Trust specifies how to establish both direct and brokered trust relationships including third parties and intermediaries. WS-Trust specifies how to acquire security tokens to establish the trust relationships and how existing trust mechanisms can be used in conjunction with WS-Security specifications.</td>
<td>• Key management</td>
</tr>
<tr>
<td>WS-Privacy</td>
<td>WS-Privacy specifies how to state privacy preferences and organizational privacy practice statements by using a combination of WS-Policy, WS-Security, and WS-Trust. It defines how to embed a privacy language into WS-Policy, and how to use WS-Security and WS-Trust for associating privacy claims with a message.</td>
<td>• Security enforcement mechanism</td>
</tr>
<tr>
<td>WS-Secure Conversation</td>
<td>WS-SecureConversation specifies how to manage and authenticate message exchanges between parties and how to establish mutually authenticated security contexts. The specification describes how to operate WS-SecureConversation at the SOAP message layer in conjunction with WS-Security, and how to establish session keys, derived keys, and per-message keys.</td>
<td>• Authentication</td>
</tr>
<tr>
<td>Specification</td>
<td>Description</td>
<td>Security requirements addressed</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>WS-Federation</td>
<td>WS-Federation specifies how to construct federated trust relationships in a heterogeneous environment using the WS-Security, WS-Policy, WS-Trust, and WS-SecureConversation specifications. It also defines how to federate Kerberos and PKI infrastructures, and how to manage the trust relationships.</td>
<td>• Trust</td>
</tr>
<tr>
<td>WS-Authorization</td>
<td>WS-Authorization specifies how to manage authorization data and authorization policies. It describes how claims within security tokens are specified and interpreted at the endpoint.</td>
<td>• Authorization</td>
</tr>
</tbody>
</table>
WS-Trust in financial reporting services

WS-Trust specifies a way to use SOAP to communicate with an organization that creates security tokens. It provides a mechanism that requests security tokens from security token services such as a Kerberos Key Distribution Center (KDC) or a Certification Authority (CA), and that responds to the request. Assume that a user wants to use an XARL service that requires a digital signature and matching X.509 certificate. First, the user makes a SOAP request to a CA requesting a X.509 certificate. The CA sends back a SOAP response that includes the X.509 certificate. Then, the user requests the XARL service passing the newly required X.509 certificate and the matching digital signature in the SOAP request message. 22

---

22 An example of a SOAP request and response message that includes WS-Trust codes is available from the authors upon request.
4.5. Incorporating security into financial reporting services infrastructure

There are several architectural approaches that can be taken to implement XARL and XBRL services. Boritz and No (2004b) address the infrastructure requirements to enable financial reporting services to be implemented. Fig. 4 depicts an example of a XARL implementation architecture that incorporates WSSA specifications.

Suppose a public company, Toronto Inc., is an entity that provides XBRL service; Waterloo Assurance Co. (WAC) is an entity that provides an XARL service; and Kitchener Service Co. (KSC) is an entity that provides a UDDI service. Also, assume that a user, Mickey, wishes to analyze Toronto Inc.’s financial statements for his investment decision.

Toronto Inc., an XBRL service provider, prepares its financial information using its internal accounting system and generates an XBRL document by mapping the financial information to company specific taxonomies and XBRL taxonomy elements. The taxonomy is obtained from the XBRL organization over the Internet. The XBRL taxonomy is transmitted as a SOAP-encoded XML message and is digitally signed by the XBRL organization using security tokens such as Kerberos ticket or X.509 certificate (WS-Security), so that if Toronto Inc. wishes to verify the legitimacy of the XBRL organization, it can do so through the third party authentication organization. The created XBRL document is automatically checked whether it is proper XBRL. Then, the validated XBRL document is placed on the company’s database.

The XARL assurance provider, WAC, obtains the XBRL document from Toronto Inc. When the service provider agent at Toronto Inc. receives a request from the service requester agent at WAC for an XBRL service, the agent at Toronto Inc. authenticates WAC by verifying the legitimacy of WAC through the third party authentication organization. The service request is a SOAP-encoded XML message that is prepared based on Toronto Inc.’s XBRL service description (WSDL document). The request is encrypted using a security token. Then, the agent at Toronto Inc. communicates with the agent at WAC, if the WAC is authorized to receive the information by sending the XBRL document with proper style sheets to the WAC. The XBRL document is digitally signed by Toronto Inc. using a security token. For example, the SOAP-encoded XBRL document is digitally signed by Toronto Inc. using its private key and encrypted with the WAC’s public key. Using its private key and Toronto Inc.’s public key, WAC decrypts the XBRL document. If WAC wishes to verify the legitimacy of Toronto Inc., it can verify Toronto Inc. through the third party authentication organization. Then, WAC performs assurance procedures on the information and creates a XARL document by mapping assurance information related to the business reporting elements in the XBRL document to agreed-upon XARL taxonomy elements. The XARL taxonomy is obtained from the XARL organization over the Internet or
another suitable source. The taxonomy is transmitted as a SOAP-encoded XML message and is digitally signed by the XARL organization using security tokens. Therefore, if WAC wishes to verify the legitimacy of the XARL organization, it can do so through the third party authentication organization. The created XARL document is automatically checked to ensure it is proper XARL. Then, the validated XARL document is placed on WAC’s database.

Once the XARL document is created, WAC, using a provider agent (software application) via the UDDI interface, publishes an SOAP-encoded XARL service description (i.e., WSDL document) to KSC, a discovery registry. Again, the SOAP-encoded WSDL document is encrypted using a security token and sent to the agent at KSC.

When Mickey needs the information contained in the XBRL document for his analysis, the requester agent communicates with the discovery agent at KSC and transmits the service request that is SOAP-encoded and encrypted using a security token. If the discovery agent at KSC accepts the request from the requester agent, it searches the discovery registry for an appropriate XBRL service. Once the service is found, the discovery agent retrieves Toronto Inc.’s WSDL document and sends it to the requester agent. Based on the information in the WSDL document, the XBRL service requester agent at Mickey sends a SOAP request to the XBRL service provider agent at Toronto Inc. Then, the agent at Toronto Inc. sends the SOAP-encoded XBRL document. Although Mickey can obtain the XBRL document directly from Toronto Inc., doing so exposes him to the risk of getting unreliable information.

In order to obtain the XARL document, the agent at Mickey needs to find the XARL service description. The requester agent at Mickey communicates with an agent at KSC via the UDDI interface and transmits the service request. Again, the request is a SOAP-encoded XML message that is encrypted using a security token. Next, if the discovery agent accepts the SOAP request from the requester agent, it searches the UDDI registry for an appropriate XARL service. Then, the discovery agent transmits the XARL service description to the agent at Mickey.

Using the information obtained from KSC, the requester agent at Mickey sends a SOAP-encoded XARL request message to WAC to directly bind to the XARL service and invoke it. Then, WAC provides Mickey with the XARL service based on the service semantics and the service description.

Some users could be asked to provide a public key which is obtained from a third party authentication organization. Upon receiving a request for information, the agent at WAC authenticates the user by verifying the legitimacy of the user through the third party authentication organization. Then, the agent sends the SOAP-encoded XARL document with proper style sheets to the user agent, if the user is authorized to receive the information. The SOAP-encoded XARL document is digitally signed using a symmetric key. Security tokens such as Kerberos ticket or X.509 certificate are used to encrypt the symmetric key.
For instance, the XARL document is first encrypted using a symmetric key. In addition, the message is encrypted with WAC’s private key and Mickey’s public key.

Finally, using its private key, and WAC’s public key, Mickey decrypts the symmetric key. The symmetric key is then used to decrypt the XARL document. If Mickey wishes to verify the legitimacy of the assurance company, he can verify it through the third party authentication organization. Then, Mickey uses the financial statement for his analysis. For instance, Mickey loads these financial statements into Excel and calculates the financial ratios. Also, if Mickey wants to transform the document into HTML, a spreadsheet or database, he can do so with appropriate style sheets developed by him or by outside software developers.

Once the XARL service binding between the requester agent and the service agent is established, the financial information in Excel can update it continuously (WS-Trust). Also, with WS-Policy description, Mickey can verify the conformity to stated privacy policies. Furthermore, through the steps discussed earlier, Mickey can use financial reporting services and other Web services simultaneously such as stock price quoting and stock investment services. Again, once the binding is established, Mickey can use the XARL service and the other Web services automatically and continuously. Using WSSA specifications such as WS-Trust, WS-Authentication, WS-Federation, and WS-SecureConversation, federated trust relations with other Web services can be created.

5. Concluding remarks

Currently, many companies are attempting to leverage the power of financial information by using the Internet. They have set up Intranets, connected the Intranets to the Internet, and have created corporate websites to provide stakeholders with financial information. Financial information users can more easily and on a more timely basis obtain the information they need. However, data must be re-entered or cut and pasted by users seeking to analyze it because there are no common, generally accepted formats for describing business reporting data. Extensible Business Reporting Language (XBRL) was developed to overcome this limitation, providing users with an efficient and effective means of preparing and exchanging financial information over the Internet. Extensible Assurance Reporting Language (XARL) was developed to enable assurance providers to report on the integrity of such information and help users and companies place warranted reliance on it. Thus, XBRL and XARL together can provide a standardized method to prepare, publish, and exchange financial information, and also can ensure the integrity of information distributed over the Internet.
XBRL and XARL are XML-based message exchange application services that accept requests from different systems across the Internet through the application of Web technology, and respond to the requests. This paper summarized security threats facing such financial reporting services and identifies limitations of current security technologies. Then, it discussed security requirements to ensure reliable, trustworthy financial reporting services. Finally, the paper explained several proposed security standards in addition to some point-to-point, transport-level security techniques. In particular, we illustrated a Web Services Security Architecture (WSSA) as a suitable security mechanism for financial reporting services because it provides an end-to-end security solution for Web services.

Although this paper describes several security issues that need to be carefully considered, and a range of XML-based security mechanisms needed to solve these issues, there are a number of additional issues to be addressed in future research:

- **Key management, PKI infrastructure and Certification Authority Issues.** This paper assumes the availability of certificates, but managing certificates effectively and securely is difficult. A Certification Authority (CA) must maintain a Certification Revocation List (CRL) and must fully protect its private key. Trust depends on the integrity and security of a CA’s practices and procedures.

- **Reliable message delivery in Web services.** IBM and Microsoft have proposed a set of specifications to provide reliable message delivery for Web services. The specifications include WS-ReliableMessaging, WS-Addressing, WS-MetadataExchanging, WS-Transactions, WS-Coordination, WS-EndpointResolution, and WS-TransmissionControl. Harmonizing WSSA specifications with traditional security technologies is a key issue.

In conclusion, it is clear that while financial reporting services can provide several benefits for organizations and financial information users, they require effective countermeasures to neutralize new security threats.

**Acknowledgment**

The authors gratefully acknowledge the financial support provided by the University of Waterloo Centre for Information Systems Assurance, sponsored

---

by the Canadian Institute of Chartered Accountants and the Information Systems Audit and Control Association.

References


Boritz, J.E., No, W.G., 2004b. Infrastructure requirements for assurance reporting with XARL (Extensible Assurance Reporting Language), Manuscript. University of Waterloo, ON.


