Web Services Security Issues in a Justice Environment

Prepared by the Global Security Working Group
August 2003

The opinions, findings, and conclusions or recommendations expressed in this publication are those of the Global Justice Information Sharing Initiative and do not necessarily reflect the views of the U.S. Department of Justice.

This project was supported by Award No. 2000-LD-BX-0003, awarded by the Office of Justice Programs.

Acknowledgements

The *Web Services Security Issues in a Justice Environment* was developed through a cooperative effort under the direction of the Global Justice Information Sharing Initiative (Global). Global's mission—the efficient sharing of data among justice entities—is at the very heart of modern public safety and law enforcement, and the efforts of Global have direct impact on the work of more than 1.2 million justice professionals. Global is the foremost voice for justice information sharing.

Global advises the nation's highest-ranking law enforcement officer, the U.S. Attorney General. Global aids its member organizations and the people they serve through a series of important information sharing initiatives. These include the development of technology standards such as the Justice XML Data Dictionary; creation of reports for recommending policies on data sharing issues, such as the National Criminal Intelligence Sharing Plan; and the facilitation of justice information sharing through the Global Web site.

The Global Security Working Group (GSWG) is an example of the various Global working groups. Its focus is on the trusted and secure information exchange among justice agencies. This document is the product of the GSWG and its membership of justice practitioners and industry professionals who volunteer their time and resources to ensure effective information exchange throughout the law enforcement and public safety communities.

Therefore, a special indebtedness is offered to Alan Harbitter, Ph.D., for being the principal author of this document, as well as a special expression of thanks to the following developers for their commitment and expertise.

Lieutenant John Aerts
Project Manager
Los Angeles County Sheriff's Department
Norwalk, California

Mr. Steve Correll
Executive Director
National Law Enforcement Telecommunication System
Phoenix, Arizona

Mr. Fred Cotton
Training Services Director
SEARCH, The National Consortium for Justice Information and Statistics
Sacramento, California

Mr. Ken Gill
Technology Advisor
Bureau of Justice Assistance
Office of Justice Programs
U.S. Department of Justice
Washington, DC

Alan Harbitter, Ph.D.
Chief Technology Officer
PEC Solutions, Inc.
Fairfax, Virginia

Mr. Jim Jolley
Computer Training Specialist
SEARCH, The National Consortium for Justice Information and Statistics
Sacramento, California

Mr. James Pritchett
Executive Director
Southwest Alabama Integrated Criminal Justice System
Foley, Alabama

Mr. Bob Slaski
Product Development Vice President
Advanced Technology Systems, Inc.
McLean, Virginia

Ms. Monique Schmidt
Research Associate
Institute for Intergovernmental Research
Tallahassee, Florida

Table of Contents

Scope

This document raises information security issues that should be considered by justice and public safety managers who are deploying justice XML-based systems for the exchange of justice and public safety information. These concerns are not meant to discourage continued development of these standards, but rather to assist justice managers, technologists, and practitioners in understanding and managing risk.

The first sections provide background by presenting a working definition for Web services and an overview of standards. This discussion is followed by a summary of security concerns and approaches to mitigate those concerns. Finally, we describe the security practices employed by the Southwest Alabama Integrated Criminal-Justice System (SAICS) to provide an operational example.

What are Web Services?

"Web services" is a frequently used and commonly misunderstood term. What do we mean when we use the term Web services? It all starts with XML.

While not all Web services applications use SOAP, WSDL, and UDDI, for the purposes of our definition, we will say that if you are going to call it "Web services," XML and computer-to-computer communications have to be in the mix.

While HTML[1] is the universal language for computers to present multimedia information to people over the World Wide Web, XML is the new universal language for computers to exchange information. A host of new protocols has been introduced to ride on top of XML and enhance the ability of two or more computer programs to cooperate in exchanging information.

Web services are built around a set of well-accepted Internet protocols. One important protocol, SOAP,[2] defines specific fields in an XML message that enable multiple programs ("software objects") to communicate over the Web. SOAP messages can be exchanged by software objects through the standard Web communications protocol, HTTP. HTTP is the same protocol that Web sites use to send HTML pages to Web browsers. Software objects are computer programs packaged into "black boxes." Other programmers or programs can use a software object's services ("methods," in object-speak) without knowing anything about how the object is designed or written. Certainly this is a useful feature when you have programs that want to exchange information but are operating on different computers at far reaches of the Internet.

WSDL standard, also XML-based, is used to describe the types of services that an online business (or justice organization) might offer. WSDL works in conjunction with the UDDI standard that defines an XML-based registry of services listed in WSDL format. While not all Web services applications use SOAP, WSDL, and UDDI, for the

[1] http://www.w3.org/MarkUp/
[2] http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

purposes of this document, XML and computer-to-computer communications will be considered a part of Web services.

Here is a simple generic example of how these protocols and standards might be used. Consider a fictitious Web site, www.theweather.com, that provides current weather and forecast information. If you were writing a Web services program that needed to know about the weather in Washington, DC, in real time, your software object could look in the www.theweather.com Web site's UDDI directory for the WSDL that describes weather services offered by the site. Then your software object could use SOAP to retrieve the information you need. This Web site would return the weather information in XML. XML labels each field in such a way that a software program can read and understand text fields that describe weather characteristics.

How might justice organizations use Web services? Many own dozens of disparate computer systems and automated databases. These databases may contain investigative, court, or corrections records. Web services offer a standardized way for a database system to share this information by posting the information that it can provide and responding to programmatic requests for that information. These database systems usually reside on an intranet or some other closed network.

As is the case with many new technologies, there are some challenges associated with Web services. Information security is the main challenge. To date, no one has really produced standards-based, packaged products that provide comprehensive security for a Web services application. Basic security services, such as identifying who is asking for information, protecting the integrity of information, and guarding against unauthorized intrusion is just starting to be addressed by standards committees.

Standards Overview

Engineering security services into Web services is a challenge that requires involvement from a variety of stakeholders, including both product manufacturers and standards bodies. A major step forward came on April 5, 2002, when Microsoft®, IBM®, and VeriSign released the WS-Security specification, which provides a foundation for adding security features to Web services. In addition to the WS-Security[3] specification, Microsoft and IBM published the first version of *Security in a Web Services World: A Proposed Architecture and Roadmap*[4] ("Roadmap") that looks to the future and outlines the standards and development work that will have to be performed on the three- to five-year horizon.

The WS-Security Specification—A Step in the Right Direction

The WS-Security specification describes enhancements to SOAP messaging to provide message integrity, confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies. WS-Security also provides a general-purpose mechanism for associating "security tokens" with messages. A security token is generally used to maintain the security procedures and "state" in a multiparty information interchange. No specific type of security token is required by WS-Security. It is designed to be extensible (e.g., support multiple security token formats).

The specification describes how to encode binary security tokens—specifically, how to encode digital certificates

[3] Specification: Web Services Security (WS-Security), Version 1.0, April 5, 2002, http://www-106.ibm.com/developerworks/library/ws-secure/

[4] *Security in a Web Services World: A Proposed Architecture and Roadmap*, IBM and Microsoft Corporation, April 7, 2002, http://www-106.ibm.com/developerworks/webservices/library/ws-secmap/

and security authorization tickets—as well as how to include cryptographic key information. It also includes extensibility mechanisms that can be used to further describe the characteristics of the credentials that are included with a message. The specification defines the use of XML Signature and XML Encryption in SOAP headers. As one of the building blocks for securing SOAP messages, it is intended to be used in conjunction with other security techniques.

A Roadmap to Security-Conscious Web Services Implementations

The Roadmap document proposes a technical strategy whereby the industry can produce and implement a standards-based architecture that is comprehensive, yet flexible enough to meet the Web services security needs of real businesses. It is a proposed plan for developing a set of specifications that address how to provide protection for message exchange and an overall strategy for security within a Web services environment. It defines a comprehensive WS-security model that supports, integrates, and unifies several popular security models, mechanisms, and technologies (including both symmetric and public key technologies), in a way that enables a variety of systems to securely interoperate in a platform- and language-neutral manner. It also describes a set of specifications and scenarios that show how these specifications might be used together.

The Roadmap presents a broad set of specifications that covers security services, including authentication, authorization, privacy, trust, integrity, confidentiality, secure communications channels, federation, delegation, and auditing across a wide spectrum of application and business topologies. These specifications provide a framework that is extensible and flexible and maximizes existing investments in security infrastructure.

The Standard Bodies

Several organizations are working on Web services security standards, including the W3C[5] and OASIS.[6]

W3C: W3C was created in October 1994 to develop common Web protocols that promote its evolution and interoperability. W3C has approximately 450 member organizations from all over the world. Before WS-Security, W3C developed standards for XML digital signature and encryption to provide message confidentiality and integrity. These cryptographic capabilities provide fundamental XML security features. W3C also developed standards for XKMS—the protocols for distributing and registering public keys in conjunction with the XML Signature and XML Encryption. WS-Security builds on the W3C encryption and digital signature specifications by tailoring them to SOAP.

OASIS: Members of the OASIS consortium have formed a technical committee to facilitate distributed systems management over the Internet. The goal of the new OASIS Management Protocol Technical Committee is to enable businesses to manage their own Web services and oversee their interaction with services offered by other companies. The OASIS Management Protocol will be designed to manage desktops, services, and networks across an enterprise or Internet environment. OASIS is reviewing a number of Web services standards and operations for use in the Management Protocol, including XML, SOAP, OMI, and the Web Services Distributed Management Technical Committee's CIM. The Management Protocol joins several Web services standards currently being developed within OASIS. Other specifications include UDDI for discovery, ebXML[7]

[5] http://www.w3.org
[6] http://www.oasis-open.org/home/index.php
[7] http://www.ebxml.org

for electronic business commerce, WS-Security for secure Web services, WSIA for interactive Web applications, WSRP for remote portals, and others. In particular, VeriSign, IBM, and Microsoft submitted WS-Security specifications to OASIS in June 2002. At the XML Web Services One Conference in Boston, Massachusetts, OASIS and W3C held an all-day forum to determine where to pool their resources and integrate security standards efforts. Despite extensive efforts to come to agreement on Web security standards, the two leading standards bodies can say that at least a start on moving toward a common set of standards has been made.

An Editorial View of Web Services Standards Efforts

Without a comprehensive set of standards and complying products, engineers will, by necessity, end up developing solutions that use temporary workarounds and have proprietary aspects.

There is rapid and substantial progress being made in developing standards that will uniformly specify the protocols used to secure Web services. However, the effort is complicated by a number of factors, including:

- The number of parties involved. While it will take close collaboration by a wide variety of industry and standards organizations to define enduring standards, this process, requiring consensus among many parties, takes more time.

- The desire to provide flexibility within the standards. This point is illustrated by the WS-Security standard. There is great flexibility for the product developers to implement security features and still comply with the standard. The downside of this flexibility is that two products that comply with the standards will not necessarily be able to interoperate.

- The scope of the task. Providing a full range of security services will involve attacking complex problems, such as the standardization of federated identity.

Currently, the accepted standards cover fundamental security services within the XML and SOAP. While there are tools available to implement these fundamental security services, there are no products that provide a complete, standards-based security solution. This places the responsibility on justice information system software engineers to develop secure Web services applications.

Considerations

The World Wide Web was developed to provide information sharing on a grand scale. The communications protocols behind the Web are geared to maximize information accessibility and exchange. Web services share this underlying philosophy. Freewheeling and widespread information accessibility are the antithesis of rigorous information security. This is the fundamental reason that it will take considerable time on behalf of industry, practitioners, and standards organizations to engineer comprehensive information security into Web services.

In order to understand, at a high level, where the Web services security liabilities are and what mechanisms must be added to mitigate these liabilities, we can look at the three fundamental information security areas: confidentiality, integrity, and availability, as well as identification and

authentication, which are important security functions.

Confidentiality

Confidentiality services support the policies governing access to information and are designed to ensure that information is not exposed to unauthorized parties. Currently, a common practice providing confidentiality service on the Internet is the use of the standard end-to-end encryption protocol, Secure Socket Layer or "SSL" (also more accurately referred to as Transport Level Security or "TLS"). A key problem with using SSL to provide confidentiality in Web services applications is granularity. SSL encrypts the entire session between a user and a Web server or, in the case of Web services, between two computers. More sophisticated applications of Web services may call for encrypting select fields of an XML message. For example, maybe the XML message includes medical information fields that must be encrypted to comply with HIPAA, but all other fields, for the purposes of wide-scale use, must be unencrypted. The WS-Security standard provides this kind of granularity, and there are tool-kits now available that allow developers to encrypt specific XML fields. However, we consider WS-Security-compliant applications to be an emerging technology that requires considerable expertise and complex programming to implement.

Integrity

Integrity services maintain the accuracy of information products to prevent unauthorized parties from modifying or compromising the integrity of information. One way to provide integrity in the XML message is to digitally sign selected fields or embedded documents. This feature is also supported by the WS-Security standard, and as with the confidentiality features, granular integrity in Web services is considered to be an emerging technology.

In addition to digital signature, there are data integrity issues that arise in applying Web services to implement complex transactions such as those that perform multiple updates on a distributed database or set f databases. Most sophisticated transaction systems, such as CICS or Tuxedo, include integrity features to make sure data is not corrupted by failed transactions or other anomalies. There are no comparable standardized mechanisms that are widely implemented in Web services. As a result, data integrity in a transactional setting is generally implemented through proprietary means.

Availability

Availability services provide confidence that information systems will be on the job when needed. A significant threat to the availability of computer systems is a security attack called "distributed denial of service"—one of the most difficult attacks to prevent.

While Web services do not introduce substantial new risks with respect to availability, all of the existing availability risks associated with Web sites and Internet protocols remain. If Web services are being considered for a production-level justice application, the exposure to the kinds of availability risks that are present in all Internet-based applications must be considered and mitigated.

Identification and Authentication

Identification and Authentication (I&A) are the first line of defense in many information systems. I&A mechanisms provide a basic security function: they ensure that those wishing to gain access to information resources are indeed who they represent

themselves to be. Traditional means of I&A, such as user ID and password or other challenge-response techniques, can be applied to Web services. The SSL protocol, frequently used to secure Web information exchanges, includes the ability to mutually identify and authenticate two parties. It is also a candidate for I&A in a Web services application.

In a Web services setting, further complexity is introduced into the I&A process by the concept of "federated identity." Web services offer the possibility of implementing complex transactions involving not just two parties, but multiple computers spread across an enterprise network. In order for a secure transaction to take place, each participant in the transaction must be uniquely identified (and subsequently authenticated) to the others. An enterprise-wide, unique identification scheme is referred to as federated identity. The incorporation of federated identity into Web services security is an emerging technology. As a result, current implementation will likely use proprietary mechanisms to conduct I&A across the federation.

The new protocol to augment I&A and authorization capabilities of Web services is called SAML or Security Assertion Markup Language. SAML is also based on XML and includes features that identify how a subject is authenticated, what characteristics that subject possesses (i.e., which group memberships or roles are associated with the subject), and what specific resources that the subject is or is not allowed to access. SAML is an emerging standard.

Workarounds

Despite the immaturity of native security for Web services, justice information systems designers and owners are finding that Web services' benefits outweigh the risks. As a result, "workarounds" to security limitations are surfacing as Web services pilot implementations become more common. Some of these workarounds are summarized in the following paragraphs. We believe that by using these techniques, it is possible to deploy a Web services-based justice information system with an acceptable level of security risk.

These concerns are not meant to discourage continued development of these standards but rather to assist justice managers, technologists, and practitioners in understanding and managing risk.

Use of SSL

The lack of "granularity" in using SSL to help provide confidentially for a Web services information exchange was pointed out in the *Considerations* section of this document. SSL will encrypt all of the data exchanged between two communicating hosts. It will not selectively encrypt specific fields or documents within a given session. However, in some applications, this limitation is acceptable.

SSL can also perform two-way authentication—allowing both the information provider and receiver to identify each other. Using SSL for authentication generally requires that both parties exchange public key certificates that are part of a common, or at least mutually compatible, Public Key Infrastructure (PKI).

Application Level Security

Web services designers will, in some cases, augment their application to fill in security gaps. For example, consider a system in which a user issues a query that is resolved by collecting information

from multiple disparate databases residing on multiple computer systems. This type of query can be implemented using Web services. The query application on the originating user's personal computer might prompt the user for an ID and password. The Web services software application might then package up the ID and password into an XML message that is transferred to the appropriate Web services server in fulfilling the query. The database application at the destination computer system knows where to look in the XML to extract the ID and password. The authorization profile of the Web services transaction then adopts the privileges of the originating user. Note that this XML exchange should probably be encrypted with SSL in order to protect privacy of the ID and password.

The example provided in this workaround implements I&A. However, other security services can be implemented at the application level as well. The limitation, in contrast, to a standards-compliant approach is lack of interoperability. Every computer system that participates in the Web services application must be aware of how each security function has been encoded and must run software that is capable of implementing the customized function.

Operational Restrictions on Data or Environment

There are several ways to compensate for the lack of comprehensive security services by placing operational restrictions on how Web services are used.

- Limit the data shared through Web services to non-sensitive data. There is a considerable amount of justice information that is freely available to the public. Limiting the access of Web services to this type of data reduces the need to implement confidentiality controls. In fact, in some jurisdictions state laws may place restrictions on the type and amount of data that can be shared through a technology such as Web services. In states such as California, Web services applications designers must consider how privacy laws may impact decisions to share data. In some cases, while individual sources of information may not be sensitive, the accumulation and correlation of data from multiple sources may change the level of sensitivity. The correlated data may carry a higher sensitivity level than any one of the individual data sources. Even with restrictions on the data shared, data integrity will still need to be addressed. The Web services applications developers must consider these potential confidentiality and integrity ramifications and ensure that the proper security precautions are taken.

- Physically control access to computers through which Web services can be obtained. Controlling physical access is a traditional "low-tech" approach to controlling security risk. Of course, with Web services, additional care must be taken to assure that physical constraints on accessibility are not compromised by the lack of electronic constraints. In other words, while it may be difficult for an unauthorized individual to gain physical access to a computer that participates in a Web services application, it may be easier to gain electronic access through network connectivity.

- Locate the Web services enterprise on the "private side" of the firewall (i.e., on an intranet). In fact, most of the Web services applications that are currently being deployed in a justice setting reside on an intranet or perhaps an extranet (an intranet that is extended to additional user

communities through mechanisms such as virtual private networks or "VPNs"). While limiting access is not in the spirit of Web services (i.e., to provide wide-scale information access), it reduces security risk by controlling the potential user community. Of course, if this type of workaround is to be relied upon, the justice organization must be confident that their intranet is adequately protected against intrusion or other security violations.

Tightening Up Internet Security Policies and Practices

Traditionally, the information placed on Web servers in an enterprise is not highly protected. Many organizations put their Web server in the "demilitarized zone" established by a firewall. Most system managers view Web servers as computer systems that will be frequently accessed by the public. As a result, there is a tendency to isolate them from production servers and implement a more lax access security policy through the firewall.

The introduction of Web services applications redefines the role of the venerable Web server from a generic information-posting repository to a productive, mission-essential enterprise system. As a result, more rigorous and restrictive policies and protection mechanisms are warranted. Further, as a mission-essential application provider, the Web server's capacity to handle workload becomes a more important issue. Web services application designers should conduct the appropriate analyses and workload tests to confirm that the capacity of the impacted servers and communications facilities is sufficient to meet the anticipated traffic.

A new technology direction in Web services security is to create "application-aware" firewalls that understand the content of Web services messages and can enforce initial access policies. However, these types of firewalls are still an emerging technology. Until the technology matures, organizations should consider workarounds, such as establishing dedicated computer systems for Web services applications and protecting these computer systems as they would any other production information system. Production strength protection may include placement on the private side of the firewall with a more restrictive access policy and the use of intrusion detection monitoring.

General Proprietary Solutions

Many of the workarounds that have been described in this section can be categorized as "proprietary." In other words, they are not standards-based (at least not industry standards-based) and will be unique to the organizations that implement them. In addition to the workarounds already discussed, there are commercially available products that provide Web services security by using standardized approaches, where available, and plugging in vendor-proprietary approaches, where necessary, to fill the gaps.

Any organization that uses proprietary approaches to provide more secure Web services should realize that eventually the security standards and products that implement them will mature. At that point in time, the proprietary approach will rapidly become obsolete and be a hindrance to wide-scale interoperability. Organizations that adopt proprietary security approaches would be well-advised to plan to migrate to standards-based approaches as they mature.

A further limitation of proprietary solutions is the inherent level of assurance. Generally, standards-based solutions have had the benefit of scrutiny and critique by academic and industry experts. From a security

standpoint, the quality of a proprietary solution is solely dependent upon the skills of the small team that implemented the solution. Historically, standards-based products can provide a generally higher level of assurance than proprietary products.

Best Practices Case Study

The SAICS project is the realization of the vision of Baldwin County District Attorney David Whetstone. District Attorney Whetstone spearheaded the multiagency initiative in southwest Alabama with the help of former U.S. Representative Sonny Callahan. Representative Callahan secured a $10.4 million direct appropriation for the Baldwin, Clarke, Choctaw, Escambia, Mobile, Monroe, and Washington counties, which comprise the first congressional district.

The primary goal of the SAICS project is to create and maintain an accessible and appropriately secured information system on individuals and events for criminal justice users, law enforcement, and homeland security needs, which supports effective administration of these programs, as well as public policy decisions, in a cost-effective manner throughout the southwest Alabama region. As it currently exists, SAICS uses Web services to provide information retrieval capabilities that surpass existing state criminal justice information resources by providing rapid access to multiple records databases across many different state agencies. The project plans to continue expanding the number of databases available to law enforcement and to link jails, judges, and district attorneys' offices in southwest Alabama to this information. Databases in other regions of Alabama, other states, and other federal agencies will be linked to SAICS as well.

Phase one of the new project went online January 7, 2003, and now links a multitude of law enforcement agencies in eight southwest Alabama counties to millions of records, including current and historic information such as driver's records, felony warrants, court protection orders, and state prison records, from a single query. Law enforcement officers with Internet access and a Web-based browser can now access these state databases to retrieve information vital to investigation of criminal suspects within seconds through the SAICS Web services.

The philosophy of the SAICS project has always been to make as much information available to the appropriate law enforcement personnel as possible, when they need it. In order to facilitate this and to adhere to recognized security practices, SAICS uses a combination of applications-level security workarounds and SSL-protected information transfers. In particular:

- Users are authenticated centrally upon accessing any part of SAICS. In order to grant access to potentially sensitive information, SAICS has designed a central repository of users (similar to a domain) that serves as the master user index for all SAICS functions. User permissions are set by local SAICS administrators but are not centrally authenticated unless required by agreement with the owners of other data systems that are being accessed. In that case, SAICS has the capability to pass user data off to the data owner of the specific system. That data owner then sets permissions and access to the data at the table or element level. This gives control to the local chief, sheriff, or district attorney. Access permissions are set individually, not globally. The central SAICS administrator authenticates agency administrators who, in turn, set permission levels for their personnel. Declarations concerning the appropriate state and federal

laws applicable to accessing law enforcement information systems are a part of the documentation. Agency chiefs must download a PDF file, sign, date, and then mail or fax the information. The central SAICS administrator then verifies that information, and a verification phone call is made. If this is successful, then access is granted to the local agency head or his/her designee. They then assume the responsibility for vetting and adding local users.

- Individual user access is configurable down to the data element level on each database being indexed. Local data owners set the access guidelines for their data. For example, the sheriff of Mobile County has agreed to share his jail management system data with the SAICS system users. He has reserved the right to only allow official law enforcement personnel from the immediate surrounding counties to access certain portions of the data. From the central repository, a flag is set that allows only those agencies that meet the criteria to see the information.

- There is a five-tier level of access from full administrative rights down to no access. At the local agency level, individual user access to data is further configurable by the local administrator.

- The use of 128-bit encryption under SSL is required for all access to the system. The confidentiality of the XML messages is protected at the session level by the SSL protocol.

- Additionally, some elements of access require further levels of encryption. This is handled via a VPN at the client level or at the router level, as appropriate.

- The use of keyboard fingerprint scanners at the user level, in addition to user ID and password, is being piloted to protect the system from fraudulent access.

These security mechanisms are added to the SAICS Web services to provide identification, authentication, access control, and confidentiality. The SAICS application does not depend upon Web services-specific protocols or standards to implement security.

Conclusion

This places the responsibility for developing secure Web services applications on justice information system software engineers. Without a comprehensive set of standards and complying products, engineers will, by necessity, end up developing solutions that use temporary workarounds and have proprietary aspects.

Web services involve a fundamental shift in how justice agencies will manage, access, and share information. Within the Web services architecture, security is key in justice implementations involving sensitive but unclassified information. While addressing Web services security is the first step, deciding the best way to implement security is obviously more complex. While there is substantial progress being made in developing standards, the effort is complicated by a number of factors, such as organizational structure and policies between two justice agencies who wish to share information, compatibility of standards implementations among justice organizations, and the necessity to make Web services as flexible as possible. Web services workarounds are a necessary step until WS-Security standards mature over the next few years.

Glossary

Authentication	A process used to verify the identity of a user, often as a prerequisite to allowing access to resources in a system. Authentication methods can include passwords, hardware tokens, software tokens, Smartcards, software Smartcards, and biometrics devices.
Authorization	The granting or denying of appropriate access rights to a user, program, or process.
Common Information Model (CIM)	A standard for extensible, object-oriented schema for managing information collected from computers, networking devices, protocols, and applications.
Customer Information Control System (CICS)	An online transaction processing (OLTP) program from IBM that, together with the COBOL programming language, has formed over the past several decades the most common set of tools for building customer transaction applications in the world of large enterprise mainframe computing.
Data Integrity	Proof that a file or communication has been changed only by authorized parties.
Denial of Service	A hacker attack designed to shut down or overwhelm a critical system, such as an authentication server or Web server.
DES	Data Encryption Standard.
Digital Certificate	A data structure used in a public key infrastructure to bind a particular individual to a particular public key.
Digital Signature	The result of a cryptographic transformation of data that, when properly implemented, provides a code that is attached to the message or document that acts as a signature; the signature guarantees the source and integrity of the message.
ebXML	Electronic business XML.
Encryption	The process of cryptographically converting plain text electronic data to a form unintelligible to anyone except the intended recipient.
Extensible Markup Language (XML)	The universal language for computers to exchange information with other computers over the World Wide Web.
Firewall	Any system or device that strives to allow safe network traffic to pass while restricting or denying unsafe traffic.

HIPAA	The Health Insurance Portability & Accountability Act of 1996 (August 21), Public Law 104-191, which amends the Internal Revenue Service Code of 1986. Also known as the Kennedy-Kassebaum Act.
HyperText Markup Language (HTML)	The universal language for computers to present multimedia information to people over the Web.
HyperText Transfer Protocol (HTTP)	The set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the Web.
I&A	Identification and Authentication.
Key	The secret used to encrypt or decrypt cipher text; the security of encryption depends on keeping the key secret.
OASIS	Organization for the Advancement of Structured Information Standards.
OMI	Open Model Interface.
Protocol	In information technology, a protocol is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols exist at several levels in a telecommunication connection. There are hardware telephone protocols, and there are protocols between each of several functional layers and each corresponding layer at the other end of a communication. Both end points must recognize and observe a protocol.
Public Key	One of two keys used in an asymmetric encryption system. The public key is made public, to be used in conjunction with a corresponding private key.
Public Key Infrastructure (PKI)	A collection of people, processes, and computers which are used to bind public keys to entities, enable other entities to verify public key bindings, revoke such bindings, and provide other services critical to managing public keys.
SAICS	Southwest Alabama Integrated Criminal-Justice System.
Security Assertion Markup Language (SAML)	An XML security standard for exchanging authentication and authorization information.
Security Token	A device issued to authorized individuals that generates a code used to provide proof of their identity in a two-factor authentication system; can be a hardware or software token. Also called an authenticator.
Simple Object Access Protocol (SOAP)	A Web protocol that defines specific fields in an XML message that enables multiple programs to communicate over the Web.

Symmetric Encryption	An approach that uses the same algorithm and key to both encrypt and decrypt information.
TCP	Transmission Control Protocol. The main protocol of the Internet.
Transport Level Security (TLS)	A protocol that ensures privacy between communicating applications and their users on the Internet. When a server and client communicate, TLS ensures that no third party may eavesdrop or tamper with any message. TLS is the successor to the Secure Socket Layer (SSL).
Tuxedo	Tuxedo (which stands for *Transactions for Unix, Enhanced for distributed Operation*) is a middleware product that uses a message-based communications system to distribute applications across various operating system platforms and databases.
Universal Description, Discovery, and Integration (UDDI)	An XML-based registry of services listed in Web services description language format.
Virtual Private Network (VPN)	A collection of technologies that creates secure connections over a public network such as the Internet.
W3C	World Wide Web Consortium.
Web Services Description Language (WSDL)	An XML-based standard that is used to describe the types of services that an online business (or justice organization) might offer. WDSL works in conjunction with UDDI.
Web Services for Interactive Applications (WSIA)	OASIS Technical Committee that is working on specifications for Web services for interactive applications.
Web Services Remote Portal (WSRP)	OASIS Technical Committee that is working on specifications for Web services remote portals.
WS-Security	Web Services Security is a proposed information technology industry standard that addresses security when data is exchanged as part of a Web service.
XML Key Management Specification (XKMS)	A proposed XML security standard that defines trust issues beyond the XML Signature specification.

Resources

Information on Glossary definitions: www.whatis.com

Information on XML digital signature: http://www.w3.org/Signature

Information on XML Encryption: http://www.w3.org/Encryption

Information on XKMS: http://www.w3.org/TR/xkms

Information on SAML: http://oasis-open.org/committees/security

Information on XACML: http://www.oasis-open.org/committees/xacml

Information on WS-Security: http://www.oasis-open.org/committees/wss

Information on Web Services Security (WS-Security) Specification:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-security.asp

"Security in a Web Services World: A Proposed Architecture" (see footnote 4).