OASIS for Access Management: Emerging Standards Offer Healthcare Help with Security

by **Hal Lockhart**

Technology standards not only facilitate the exchange of electronic health information, they also provide for its security and confidentiality by promoting mechanisms that have been reviewed by experts. If electronic data exchange cannot ensure privacy and security, it will never realize its promises of improved care and reduced costs.

One organization developing security standards is the Organization for the Advancement of Structured Information Standards. OASIS standards have been recognized for application in healthcare, and the organization is currently profiling specifications that will be available for protecting electronic health information.

Selected for Standards Harmonization

The Healthcare Information Technology Standards Panel is a public-private collaborative harmonizing and integrating standards for sharing health information. HITSP's security and privacy technical committee produced two specifications addressing the technical means of providing privacy and security for electronic healthcare records. Most of the standards referenced by these two documents were developed at OASIS, including the standards described below.

HITSP Security and Privacy Technical Note (HITSP/TN900) includes an overall road map, specific security and privacy constructs, applicable standards, gaps and outages, a glossary of terms, and useful information on risk management. HITSP Access Control Transaction Package (HITSP/TP20) focuses more narrowly on the specifics of making access control decisions relating to electronic healthcare records. It also calls out a number of technical standards to be used for this purpose.

OASIS is now working to produce technical profiles of three standards—XACML, SAML, and WS-Trust—which detail the mechanisms that provide the capabilities described in HITSP/TN900 and HITSP/TP20.

XACML

The eXtensible Access Control Markup Language (XACML) is an XML-based language for specifying access control policies. XACML policies are designed to be evaluated by software whenever an access control decision is required, potentially using any available information to determine if access should be permitted or denied.

The XACML architecture separates the task of making access control decisions from that of enforcing those decisions. A policy decision point is responsible for evaluating policies, while a policy enforcement point enforces the decision.

XACML policies can consider information in any of four categories: subject, resource, action, and environment. The policies can compare these values to each other or to constants using standard operations such as equal, greater, or less than, depending on the type of data. Results can be combined using standard logical operators such as "and," "or," and "not." Because XACML allows any kind of name to be used, its policies can control access to any resource from a component in a personal digital assistant, a data store on the Web, or the rooms of a building.

XACML is different from access control systems such as file permissions or access control lists. Multiple policies may apply in a given situation. If a policy evaluates to "true," then its effect (permit or deny) is noted. Otherwise it is ignored. Since two or more policies may be applicable in a given situation, a combining algorithm is applied to obtain a single decision. The usual combining algorithm is deny overrides, which means all applicable policies must permit the request.

XACML policies may also return one or more obligation values with either a permit or deny decision. An obligation is merely an associated value of which the policy enforcement policy knows the meaning. For example, an obligation might indicate that an audit record must be generated, that the data provided must be destroyed after 30 days, or that a particular quality of service should be provided.

SAML

Security Assertion Markup Language (SAML) conveys information relating to security from one system to another. The fundamental construct in SAML is an XML document called an assertion. All assertions have a header that contains information about who issued it, the subject it refers to, and conditions of its use, such as the time period during which it is valid. A SAML assertion can be signed by its issuer and can contain a subject key, just like an X.509 certificate.

A SAML assertion contains one or more statements. There are two types of statements in use: authentication statements and attribute statements.

An authentication statement reports that the subject successfully authenticated using a particular method at a particular date and time. This is generally used to implement single sign-on protocols, particularly for Web portals. An attribute statement carries attributes of the subject, including the attribute name, value, and data type. Typical attributes include group, role, job title, department, and organization.

SAML defines a large number of distinct protocols for obtaining assertions to meet the requirements of different use scenarios. SAML can operate anywhere from low-security environments using passwords and Web cookies to high-security environments using public key certificates for digital signatures and encryption.

SAML enables identity federation, which means that a given user's attributes may be maintained and published by distinct organizations. Ideally the organization with the most authoritative knowledge of an attribute is its keeper.

WS-Security, WS-Trust

Web Services Security (WS-Security) is a set of specifications that enable portions of simple object access protocol messages to be digitally signed or encrypted, enabling authentication, message integrity, and message confidentiality. WS-Security uses a token or an information object that contains claims (such as attributes of a subject), which may contain a key and may be digitally signed by its issuer. Common types of tokens include X.509 certificates, SAML assertions, Kerberos tickets, and the Username token defined by WS-Security.

In a distributed environment when it is necessary to obtain access to a Web service outside of the local domain, it may not be possible to use the same token that is used for access to local service. Web Services Trust Language (WS-Trust) defines the concept of a security token service (STS). An entity that requires different credentials for some access contacts an STS, authenticates itself using the credentials it already has and requests a token suitable for some other purpose. If policy permits, the STS issues an appropriate token. It may also return a protected copy of any secret key associated with the token.

WS-Trust can also be used to cancel or renew tokens as well as support key distribution, key rollover, and various key agreement schemes. STS can also be used to validate a token presented by another party. It supports both two- and three-party interactions.

Applying the Standards: Three Examples

Within healthcare, XACML may be used to define policies that implement the required access control policies. Healthcare access control is complex, embodying a number of distinct models, including role-based access control, consent directives, permissions, purpose, confidentiality codes, dissenting roles and individuals, and locality of access. Because of the flexibility of the XACML language, all of these requirements can be met. In addition there are requirements for various post conditions that can be met by the use of obligations.

However, an XACML policy decision point cannot make an access control decision without the necessary input data. This is where SAML comes in. SAML attribute assertions are used to convey the detailed properties of the parties involved in

requesting access to electronic healthcare records. Depending on the exact scenario, either SAML or WS-Trust protocols may be used to issue the necessary tokens and keys and potentially validate them.

A few examples will make this clearer. The basic mechanism for controlling access to medical records is the Health Level Seven permission codes. They specify a particular type of access to a particular type of record. For example, PRD-010 is defined as “review patient medications.” Patient records are marked with the permissions required for certain types of access. Users are assigned the permissions they require to do their jobs.

In the first example, Dr. Jones wishes to review the medical records of a new patient, Mary Smith. First an SAML assertion is generated that represents the relevant information about Dr. Jones. It would include his name, role (physician), location (metro hospital), purpose (healthcare treatment), and permissions he has been granted.

When Dr. Jones requests access to Mary’s records, his computer system passes the SAML assertion along with the request. Next the medical record system retrieves the list of permissions that are required to access the requested records. Now the policy is evaluated to determine if Dr. Jones has the necessary permissions. (Other information would also be checked by policy, such as whether Dr. Jones has the proper hospital affiliation.) In this example, Dr. Jones has the proper permissions and access is granted.

In another example, Dr. Adams, a radiologist, tries to access the records of Peter Brown. Peter has created a patient directive (as is his privacy right) that prohibits radiologists from accessing his records. This directive is associated with his records and managed by the medical records system. Although Dr. Adams has the permissions necessary to access the information, the XACML policies check for patient directives that apply to this request and refuse Dr. Adams access.

In the final example, Mary Smith is injured in an auto accident and brought to Country Clinic. The ER physician, Dr. Miller, attempts to access Mary’s records. Although the clinic computers are connected by network to the records kept at Metro Hospital, clinic staff are normally not permitted access to patient records there. Dr. Miller declares an emergency condition for the request.

This information is contained in the SAML assertion sent with the request. The emergency condition causes a different set of XACML policies to apply and access is allowed. The policies also specify an obligation to create an audit log record of all relevant details. This makes it possible to later review emergency requests to ensure that emergency access is being used appropriately.

Hal Lockhart (hal.lockhart@oracle.com) represents Oracle in the development of information security standards at standards bodies including OASIS, W3C, and WS-I. He is a cochair of both the OASIS SAML and XACML technical committees and a member of the OASIS technical advisory board.

Article citation:
Lockhart, Hal. "OASIS for Access Management: Emerging Standards Offer Healthcare Help with Security" *Journal of AHIMA* 80, no.3 (March 2009): 52-53.