Accepted for publication in Communications of the ACM, To appear in 2006.

An Integrated Approach to Federated Identity and Privilege Management in Open Systems

Rafae Bhatti
School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN
rafae@purdue.edu

Elisa Bertino
CERIAS and Department of Computer Sciences, Purdue University, West Lafayette, IN
bertino@cs.purdue.edu

Arif Ghafoor
School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN
ghafoor@purdue.edu

1. Introduction

Web-based collaboration in the highly-networked enterprise environment is essential to maintaining strategic partnerships on the Internet. The access management to enterprise resources in such collaborative environments is absolutely critical for their security. The major industrial players in security also opine that "today's collaborative and interconnected e-business landscape requires a secure and effective way for enterprises to share trusted user identities"[1] and entitlements. The ability to federate identity across organizations while maintaining access rights and privileges poses a major challenge [5]. The solution is federated identity and privilege management, which now stands as the key to seamless and secure enterprise integration and collaboration on the Web. However, almost all well-known such schemes have their drawbacks. Additionally, the development of Web-based federated identity solutions has advanced more rapidly as compared to the Web-based privilege management mechanisms, resulting in a wide gap in integrating privilege management with existing federated identity mechanisms to provide a comprehensive access management solution. This disparity is quite alarming, and the increasing trend of migrating enterprise operations to the Internet demands a significant evolution of the traditional access management mechanisms in order to secure the inherently dynamic Web-based resources [5]. Simply put, both federated identity and privilege management are cornerstones of an access management framework; the strength of each is critical to the effectiveness of the overall mechanism. In this paper, we discuss these challenges, namely the shortcomings of federated identity mechanisms, and their integration with privilege management mechanisms. In response, we present an integrated approach to federated identity and privilege management specifically designed for Web-based platforms.

[1] Federated Identity white paper, RSA Security Inc.

At the very onset, we would outline the requirements that we believe an integrated federated identity and privilege management mechanism should satisfy.

(i) Single sign on (SSO): SSO essentially implies persistence of user identity and entitlement across multiple enterprise domains. Although many SSO solutions exist, the widening gap between identity and privilege management leads to many challenges with regards to granting single-sign-on access to collections of resources that might have contradictory access-protection rules [5].

(ii) Effective access control: The access management solution relies on the strength of the access control model, and should support an effective and fine-grained access control model that can manage access to dynamically evolving enterprise resources. This requirement is particularly challenging to meet in a Web-based environment.

(iii) Decentralized model: This implies that the system should not rely on a centralized or single point for accessing user authentication and authorization information. This requirement is motivated by the market demand for B2B scenarios, where it is desired to have a decentralized model for federating user identities and entitlements and thereby, as Pfitzmann et. al put it, avoiding a scenario where "one enterprise essentially authenticates the world population".

(iv) Authentication for strangers: In the widely distributed Internet environment, it is no longer a workable business model for a service provider to assume in advance the knowledge of the identities or capabilities of all users. The use of identity and capability-based credential in most existing systems is a major bottleneck to achieving this objective.

(v) Trust, Anonymity and Privacy: Privacy protection is becoming an increasingly significant issue, more so from social and legal perspective, and it is a challenge to provide sufficient level of anonymity and privacy without compromising on security. The paradox here is clear:

while avoiding name-binding appears viable for preserving privacy, it complicates the accountability in trust establishment.

(vi) Standardized Approach: With numerous schemes in several stages of adoption, it is only prudent to take an incremental or "integrate"-able approach: design new solutions that complement existing accepted standards. We have therefore carefully evaluated the existing technologies and attempted to address only the open issues; for other functionality, we provide hooks within our specification where existing standards can be tied into.

2. Background, Motivation and Related Work

The concept behind federated identity and privilege management mechanisms derives its motivation from the classical authentication and authorization protocols, as we shall now discuss. A seminal work in authentication protocols [12], implemented as Kerberos (http://web.mit.edu/kerberos/www/), uses identity-oriented name-bound credentials issued by a centralized server. Such schemes have scalability problems in distributed systems. The X.509 Internet standard for credential format defined in RFC 2511 is also identity-oriented, and its name binding tends to be long-lived, making it ill-suited to expressing distributed authorizations. Alternatively, various schemes emerged for distributed authorization using capability-based credentials. Notable amongst them are [4], and [8], which are based on the Public Key Infrastructure (PKI). The PKI-based approach to distributed access control is traditionally known as Trust Management (TM). We shall henceforth refer to the credentials used in TM schemes as TM credentials. In the above mentioned schemes, the TM credentials used have their drawbacks. Although the use of key-centric capability-based TM credentials in [4, 8] removes the dependency on names, the binding of capabilities encoded in the credential with the key blurs the distinction between authentication and authorization, thereby tightly coupling the two. Such an approach limits the expressiveness (and hence effectiveness) of the access control mechanism, since not all system-specific capabilities may be known in advance in a distributed environment. This is especially the case if SSO is to be supported, because the intention there is to prevent having multiple authorization mechanisms for access to multiple resources.

The next generation of distributed authorization models has attempted to alleviate this drawback by designing effective and more expressive access control schemes, and many have employed the Role Based Access Control (RBAC) as a solution to privilege management There, however, remain shortcomings. The X.509 based PMI and its reference implementations such as PERMIS [6], adopts a name-binding approach. Another emerging specification is the XML-Based Access Control Markup Language (XACML- http://xml.coverpages.org/xacml.html). XACML doesn't directly support role-based access control, but there exists an XACML profile for RBAC. The most current version (02) of this profile does not capture all essential features of RBAC, such as separation of duties, and session-based authorization management. X-GTRBAC and OASIS [2, 1] are similarly expressive models using RBAC to define dynamic fine-grained access control in an enterprise environment. However, all of the above schemes use either name-bound or capability-based credentials and are not scalable to the case of role assignment for unknown users on the Internet. Another scheme [9] uses Trust Policy Language (TPL) to map holders of public key certificates to roles. The Role based Trust management (RT) framework [11] merges features from TM and RBAC and uses a more expressive policy language compared to TPL. The TM credentials used in [9, 11] are examples of property-based credentials because they allow user authentication and subsequent authorization based on certain properties thereof. These can be used to authenticate unknown users into known roles, since pre-defined identities and capabilities are not assumed. Both schemes, however, have shortcomings. Firstly, they do not support an elaborate access control scheme beyond the basic permission-to-role assignment mechanism in RBAC. Additionally, [9] in its present implementation uses X.509-based PKI, and hence adopts a name-binding approach.

None of the above schemes has support to satisfy the requirement of SSO, a fundamental component of federated identity. The most prominent Web-based SSO system in use today - Microsoft Passport - is based on a centralized server model, and is much like a Kerberos counterpart for the Web. However, on an Internet scale, the centralized approach is not without its due share of risks- amongst them are compromise of the central repository and subjugation to denial of service attacks. A centralized model, in fact, is antithetical to the distributed nature of the Internet [10]. Two prominent SSO mechanisms, Shibboleth and Liberty Alliance, are based on decentralized approach. They, however, are limited to providing support for distributed authentication and do not provide support for specifying and enforcing access control policies.

3. Proposed Solution

We address the problem of providing improved identity and privilege management solution through an interoperable and modular design of underlying authentication and authorization mechanisms. In particular, we integrate decentralized SSO mechanism within an authorization model by adapting it to use property-based TM credentials and incorporating support for credential management.

An initial requirement the authorization model needs to satisfy is suitability to Web-based applications. Based on the original system requirements and the discussion in Section 2, we believe that X-GTRBAC [2] is one candidate. The X-GTRBAC model through its XML-based specification enables

Table 1. Salient Features of X-GTRBAC

Element Type	Element Name	Purpose
RBAC Element	*XML User Sheet (XUS)*	Declares the users and their authorization credentials
	XML Role Sheet (XRS)	Declares the roles, their attributes, role hierarchy, and any separation of duty and temporal constraints associated with roles
	XML Permission Sheet (XPS)	Declares the available permissions
RBAC Assignments	*XML User-to-Role Assignment Sheet (XURAS)*	Defines the rules for assignment of users to roles; these assignments may have associated temporal constraints
	XML Permission-to-Role Assignment Sheet (XPRAS)	Defines the rules for assignment of permissions to roles; these assignments may have associated temporal constraints
RBAC Constraints	*XML Separation Of Duty Definition Sheet (XSoDDef)*	Defines the separation of duty constraints on roles
GTRBAC Constraints	*XML Temporal Constraint Definition Sheet (XTempConstDef)*	Defines the temporal constraints on role enabling and activation; also defines temporal constraints for user-to-role and permission-to-role assignments
	XML Trigger Definition Sheet (XTrigDef)	Defines context-based triggers for invocation of periodic events subject to associated constraint evaluation
Authenticating Credentials	*XML Credential Type Definition Sheet (XCredTypeDef)*	Defines the available credential types

effective Web-based access control capabilities [3]. It has therefore been adopted as the authorization model in our system. For the benefit of the reader, we tabulate the salient features of the model in Table 1. The complete grammar specification and a detailed discussion of the access control mechanism are presented elsewhere [2, 3]. The central idea is that the X-GTRBAC system uses credentials supplied by users to assign them to roles (i.e. authentication) subject to any assignment constraints. The users can subsequently access resources according to their role memberships (i.e. authorization) subject to any access constraints. Hence, X-GTRBAC supports fine-grained attribute-based access control with modular authentication and authorization mechanism. To adapt the model for Web-based SSO, we outline the configuration shown in Figure 1. In the remainder of this section, we discuss the X-GTRBAC configuration. The next section explores the software architecture of the system, and discusses a prototype implementation of our model with the help of an execution scenario described later in this section.

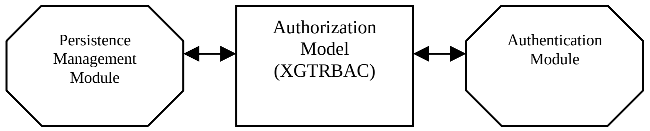

Figure 1: Design methodology of the proposed solution

The interface to the system has been designed so that it should support, and not duplicate, the functionalities available in existing standards. Although many specifications are in the works, the Security Assertion Markup Language (SAML - http://xml.coverpages.org/saml.html) is currently hailed as the enabling technology for SSO. SAML provides a message exchange protocol between autonomous business entities, and can be used to encode security attributes and decisions called “assertions”. However, SAML is not a self-sufficient mechanism to ensure SSO as it does not provide any authentication or authorization support; it does the important task of allowing the communicating entities to exchange security information in a decentralized manner but does not establish, check or revoke any information on its own. Therefore, a mechanism is needed that SAML can tie in to. Our specification provides one such mechanism, and is designed so as to accept SAML-encoded assertions as an acceptable form of credential. However, that alone is not sufficient for our purposes- SAML assertions are inherently subject to the same name-binding problem that exists in the protocols it is designed to work with, such as

1. User `Bob` needs to access a library resource, say `“CACM_Vol8_No2”`, through his local library login.
2. The local library (`LibBob`) is part of a digital library federation (`FedDigLib`).
3. `“CACM_Vol8_No2”` is not available locally but at another library (`LibElse`) which is part of `FedDigLib`.
4. `LibElse` categorizes all resources. `“CACM_Vol8_No2”` is categorized as `“LibResourceLevel2”`.
5. `LibElse` is not aware of any user `Bob`, but has a resource access policy that is not based on user identity. Instead, it is based on attributes that a user must satisfy depending on the resource category.
6. The access policy for category `“LibResourceLevel2”` requires a user to provide attributes that include a date of birth (to establish age) and a valid driver’s license. It also restricts the resource access to 2 days.
7. `LibElse` publishes the resource metadata that includes the attributes required for access together with a list of attribute authorities `LibElse` trusts. The metadata is available at a well-known URL.

Figure 2: Example of a Web-SSO request motivating use of property-based credentials

Kerberos and X.509. Therefore, we have designed a specification that works with property-based TM credentials. In particular, we have created a SAML profile for X-GTRBAC involving the feature set from latest SAML specification (v2.0). The use of SAML profile in X-GTRBAC system requires a translation from SAML encoding to X-GTRBAC format, and vice versa, using XSLT.

The rest of this article focuses on precise policy configuration semantics of our proposed specification. To keep the discussion focused, we use the Web-based SSO request shown in Figure 2 as a running example. Table 2 provides the credential configuration using SAML profile for X-GTRBAC in the context of this example. It uses features from SAML standard v2.0 which allows this credential configuration to be adopted by all entities that are already using SAML-compliant protocols. The credential is represented by a SAML assertion. We have only included the attributes and elements relevant for this discussion, and also omitted the namespace prefixes for compactness. The mapping rules used to translate a SAML assertion to X-GTRBAC format have been provided in the table. The X-GTRBAC credential is represented as an XUS document in our system (see Table 1).

We now discuss the noteworthy features of the credential configuration:

Property Based Credential: Of particular interest is the configuration of TM credentials in property-based mode which allows authentication for unknown users since identity is not assumed to be known. It can be observed that if a user name is not provided in the SAML credential, the corresponding credential in X-GTRBAC is constructed using the reserved word `“any”` which represents anonymous

Table 2: Credential Configuration in SAML profile for X-GTRBAC

SAML Credential	X-GTRBAC Instance	Mapping Rules
<Assertion id=“XXX-MAA-001“> <Issuer format=“entity“> www.my-attribute-authority.com </Issuer> <AuthnStatement>…</AuthnStatement> <AttributeStatement> <Subject> <NameID format=“persistent“> *Bob’s public key* </NameID> </Subject> <Conditions> <NotBefore> 2005:01:30</NotBefore> <NotOnOrAfter> 2006:12:31</NotOnOrAfter> </Conditions> <Attribute name =“DOB“> <AttributeValue> 1978:05:21 </AttributeValue> </Attribute> <Attribute name =“DLN“> <AttributeValue> 0991-09-0991 </AttributeValue> </Attribute> </AttributeStatement> <ds:Signature>…</ds:Signature> </Assertion>	<XUS xus_id=“LibElseXUS”> <User user_id =“any“> <UserName/> <CredType cred_type_id=“LibElseResL2SAML“ cred_type_name= “LibElseResL2SAML“> <Header> <Issuer> www.my-attribute-authority.com </Issuer> <Principal format=“persistent“> *Bob’s public key* </Principal> <Validity> <NotBefore> 2005:01:30</NotBefore> <NotOnOrAfter> 2006:12:31</NotOnOrAfter> </Validity> <DSig>…</DSig> </Header> <CredExpr > <Attribute name=“DOB“ value=“1978:05:21“ /> <Attribute name=“DLN“ value=“0991-09-0991“/> </CredExpr> </CredType> </User> </XUS>	- User@user_id = auto generated (“any“ if NameID@format=“persistent”) - NameID->UserName (empty if NameID@format = “persistent“) -CredType@cred_type_id = auto generated -CredType@cred_type_name = auto generated - Issuer -> Issuer - NameID-> Principal - NameID@format -> Principal@format - NotBefore->NotBefore - NotOnOrAfter->NotOnOrAfter - ds:Signature -> DSig - Attribute@name-> Attribute@name - AttributeValue -> Attribute@value

users. In this case, the credential used is non-name-bound, and defines the identity of the subject in terms of a public key (or hash of it). This kind of binding is indicated by the value of “persistent” for the format attribute of NameID element in SAML assertion. Persistent is a format for NameIDs in SAML standard that allows opaque values (such as random hashes) to be used in place of subject names in support of anonymity and privacy. Note that name-binding credentials can still be used if desired, which will be indicated by the appropriate value of the format attribute of NameID element as per the SAML standard (for e.g. X.509 Subject Name or Kerberos Principal Name).

Authenticating Attributes: The AuthnStatement element in the SAML assertion contains the authentication context used to generate the authenticator (i.e. credential) for the subject. The attribute information contained in the credential is not necessarily owned by a centralized entity, and can be collected from multiple attribute authorities. The authentication statement for a subject can in practice be obtained by invoking the SAML Authentication Request protocol on an identity provider. The latter responds with the authentication statement, and optionally also including attribute statements. This protocol includes the specification of a metadata repository from where required resource attributes may be learnt, and subsequently obtained using the attribute authorities indicated in the resource metadata. We maintain that our focus is not on attribute collection and credential generation. Instead, our specification is designed to work with SAML assertions that already include such credentials generated through prior means.

In addition to TM credential configuration as specified by SAML profile for X-GTRBAC, there are additional requirements on the use of credentials within the X-GTRBAC system to allow the access control capabilities of X-GTRBAC system to be integrated with Web-based SSO features of SAML.

Table 3: Constraint Specification in X-GTRBAC*

#	Constraint	X-GTRBAC Instance	Meaning

1.	Role Assignment	`<XURAS xuras_id="LibElseXURAS">`
		`<URA ura_id="uraBorrowerL2"`
		`role_name="BorrowerL2">`
		`<AssignUser user_id="any">`
		`<AssignConstraint>`
		`<AssignCondition cred_type_id=`
		`"LibElseResL2SAML"`
		`d_expr_id="TwoDays">`
		`<LogicalExpr>`
		`<Predicate>`
		`<Operator>neq</Operator>`
		`<FuncName>hasValue</FuncName>`
		`<ParamName>DLN</ParamName>`
		`<RetValue>null</RetValue>`
		`</Predicate>`
		`<Predicate>`
		`<Operator>neq</Operator>`
		`<FuncName>hasValue</FuncName>`
		`<ParamName>DOB</ParamName>`
		`<RetValue>null</RetValue>`
		`</Predicate>`
		`</LogicalExpr>`
		`</AssignCondition>`
		`</AssignConstraint>`
		`</AssignUser>`
		`</URA>`
		`<XURAS>`
			The role `BorrowerL2` can only be assigned to a user who possesses the credential `LibElseResL2SAML`. This refers to the credential defined in XUS document in Table 2. The assignment condition includes rules on credential attributes. It asserts the existence of the `DLN` and `DOB` attributes. The assignment condition also refers to a duration expression which implements the restriction that the resource can be borrowed only for 2 days. The duration expression is defined in XTempConstDef document in our system (see Table 1).

2.	Role Delegation	`<XRS xrs_id="xrsBorrowL2">`
		`<Role role_id="rBorrowerL2"`
		`role_name="BorrowerL2"`
		`<Junior>BorrowerL1</Junior>`
		`<DelegationConstraint>`
		`<DelegationCondition`
		`d_expr_id="OneWeek"/>`
		`</DelegationConstraint>`
		`</Role>`
		`</XRS>`
			The role `BorrowerL2` can only be delegated if the delegation constraint is satisfied. The delegation condition on the role refers to a duration expression which imposes a restriction on the duration of the delegation. The duration expression is defined in XTempConstDef document in our system (see Table 1).

* This represents only a subset of access constraints in X-GTRBAC. For complete specification, see [2].

Role Assignment: The property-based credentials from Table 2 are used by X-GTRBAC for attribute-based role assignment for unknown users. An appropriate X-GTRBAC policy configuration (see Table 3) allows `Bob` to access `CACM_Vol8_No2` at a federated site (`LibElse`) using only his certified attributes. The assignment policy is represented as an XURAS document (see Table 1).

Delegation: The requirement for delegation of authority is the key to decentralization, and is captured elegantly through the use of role hierarchy in our RBAC mechanism: a junior role inherits all privileges of a senior role. At present, we only support delegation within the role hierarchy (i.e. delegation always occurs from a senior role to a junior role). An optional Delegation Constraint may be used in the role definition (See Table 3) to limit the extent of delegation (in terms of time and associated privileges); unrestricted delegation is otherwise assumed. The role definition is given in an XRS document (see Table 1).

Digital Signatures: An effective SSO solution depends on the persistence of the authentication and authorization assertions across enterprise domains. Toward this end, the Header element of an X-GTRBAC credential includes support for digital signatures. The support for digital signatures in SAML allows signed assertions to be exchanged between all SAML-compliant entities.

4. Software Architecture

In this section, we present the software architecture of our federated identity and privilege management solution. It is depicted in Figure 3. The authentication module is responsible for generating the attribute and authentication statements included in the SAML assertion. The use of standardized protocols allows us to leverage existing mechanisms for these tasks. The SAML Authentication Request protocol discussed earlier is now implemented by stand-alone SAML-aware Web server software (e.g.: http://www.pingidentity.com/products/pingfederate.html), and may be deployed by SAML authorities to create and exchange SAML-compliant attribute and authentication statements. The persistence management module is responsible for creation of digitally signed authorization credentials. We outsource the credential management to the well-known XML Key Management Specification (XKMS - http://www.w3.org/TR/xkms/). XKMS is a Web-based service that can be invoked from a client application, and supports PKI-based key generation, registration, revocation, and verification. SOAP binding is used for message exchange. XML Encryption and XML Digital Signature standards are used to provide message confidentiality and authenticity, respectively. The end-to-end communication is assumed to be secured using mechanisms such as SSL/TLS.

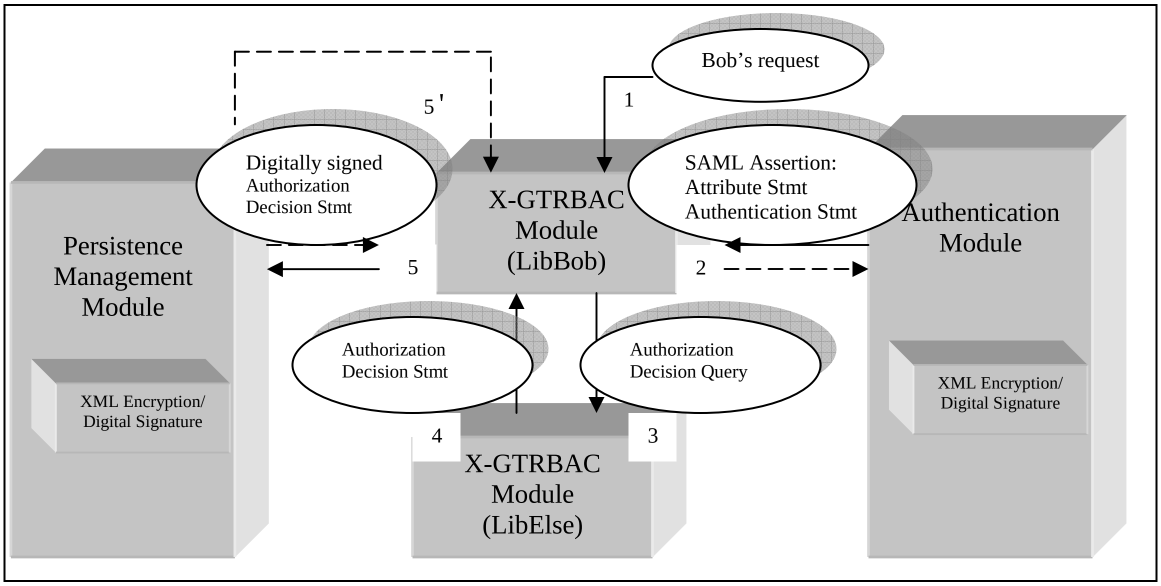
Bob's request
1
5'
Digitally signed Authorization Decision Stmt
X-GTRBAC Module (LibBob)
SAML Assertion: Attribute Stmt Authentication Stmt
Authentication Module
Persistence Management Module
5
2
XML Encryption/ Digital Signature
Authorization Decision Stmt
Authorization Decision Query
XML Encryption/ Digital Signature
4
3
X-GTRBAC Module (LibElse)

Figure 3: The software architecture for a federated identity and privilege management solution

The following execution scenario highlights the salient features of the system architecture:

Step 1: `Bob` logs into `LibBob` account and requests access to `CACM_Vol8_No2`.

Step 2: `LibBob` contacts the authentication module using SAML to obtain the necessary attribute and authentication statements. The authentication module evaluates the information in the SAML request (using either XKMS or the local server) and issues a SAML assertion including the required statements.

Step 3: `LibBob` packages the SAML assertion into the `Evidence` element in a SAML Authorization Decision Query. It then submits the query to `LibElse` on behalf of `Bob`.

Step 4: Based on SAML assertion contained in the query, the X-GTRBAC module at `LibElse` assigns a role membership to `Bob` (not identified as such by `LibElse`) according to the available information. An underlying assumption here is that the system administrator has already defined a credential type associated with SAML Assertions so that the X-GTRBAC module can appropriately translate from the

SAML Assertion format to X-GTRBAC XUS format. (In our example, this is the `LibElseResL2SAML` credential.) Once the role assignment has been made local to the X-GTRBAC system, the authorization for `Bob` is determined by the permission assignment policy for the role, and the authorization decision is issued as a SAML Authorization Decision Statement. The decision statement includes as evidence of authorization a SAML Assertion issued by `LibElse` indicating the role membership of `Bob`. This evidence can be used to allow single sign on in future.

Step 5: To facilitate SSO, the X-GTRBAC module communicates the authorization credential (i.e. SAML Authorization Decision Statement) to the persistence management module, which digitally signs it. This credential can subsequently be used by `Bob` at a federation site that accepts `LibElse`-issued credentials without going through an authentication process (step `5'`).

We have implemented a preliminary prototype of our proposed architecture, which is available for evaluation at http://web.ics.purdue.edu/~bhattir/project/sso.

Conclusion

Our framework is a novel attempt to address the issues discussed in the introduction to this paper. Our approach integrates two security standards, namely RBAC and SAML, toward designing an access management framework for open systems. It complements other efforts in this direction aimed at allowing interoperable access management using standardized protocols [7]. Overall, our grammar specification provides support for federated identity and privilege management while meeting the outlined requirements. Among future challenges are integration with existing directory schemes to support property-based credentials, supporting trust negotiation protocols for incremental attribute collection, and maintaining state information for anonymous users to ensure proper accountability.

References

[1] J. Bacon, K. Moody, and W. Yao, “Access control and trust in the use of widely distributed services”, In Middleware 2001, volume LNCS 2218, pages 300{315. Springer-Verlag, November 2001.

[2] R. Bhatti, J. B. D. Joshi, E. Bertino, A. Ghafoor, "X-GTRBAC: An XML-based Policy Specification Framework and Architecture for Enterprise-Wide Access Control”, ACM Transactions on Information and System Security (TISSEC), Vol. 8, No. 2.

[3] R. Bhatti, E. Bertino, A. Ghafoor, “A Policy Framework for Access Management in Federated Information Sharing”, In proceedings of 2005 IFIP TC-11 WG 11.1 & WG 11.5 Joint Working Conference on Security Management, Integrity, and Internal Control in Information Systems, December 2005. *(To appear)*

[4] M. Blaze, J. Feigenbaum, and A. D. Keromytis, “KeyNote: Trust management for public-key infrastructures," in Security Protocols International Workshop, Springer LNCS, no. 1550, pp. 59-63, 1998.

[5] D. Buell, R. Sandhu, “Guest Editors' Introduction: Identity Management”, IEEE Internet Computing, Nov/Dec2003.

[6] D. Chadwick, A. Otenko, “The PERMIS X.509 role based privilege management infrastructure”, In Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, June 2002.

[7] D. Chadwick, S. Otenko, V. Welch, “Using SAML to link the GLOBUS toolkit to the PERMIS authorization infrastructure”, In Proceedings of the Eighth Annual IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Windermere, UK, September 15-18, 2004

[8] C. M. Ellison, “SPKI requirements," RFC 2692, Internet Engineering Task Force Draft IETF, Sept. 1999. See http://www.ietf.org/rfc/rfc2692.txt.

[9] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, “Access control meets public key infrastructure, or: Assigning roles to strangers”, In Proceedings of the 2000 IEEE Symposium on Security and Privacy, pp. 2–14, 2000. IEEE Press.

[10] David P. Kormann and Aviel D. Rubin, “Risks of the Passport Single Signon Protocol”, Computer Networks, Elsevier Science Press, volume 33, pages 51-58, 2000.

[11] Ninghui Li, John C. Mitchell, and William H. Winsborough, “Design of a role-based trust management framework”, In Proceedings of the 2002 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, May 2002.

[12] R. M. Needham and M. D. Schroeder, “Using encryption for authentication in large networks of computers," Communications of the ACM, vol. 21, no. 12, pp. 993-999, 1978.