SECURE DYNAMIC WEB SERVICES COMPOSITION IN THE CONTEXT OF CONSTRUCTION E-PURCHASING

U. M. Mbanaso, G. S. Cooper, Y. Rezgui, M. Wetherill, S. C. Boddy

ABSTRACT: Service-Oriented Architectures based on Web Services are promising to revolutionize the implementation of open and dynamic transactions in many industries, including construction. However, the application of the technology is raising new security and privacy challenges. One aspect to be addressed in dealing with the security issues is user authorization. Traditionally, authorization systems tend to be unilateral in the sense that the service provider assigns the access rights and makes the authorization decision, and there is no negotiation between the client and the service provider. Trust negotiation builds on this through the gradual release of remotely issued credentials to service providers. However, this is not sufficient where strict privacy governance is a requirement, particularly where the communicating parties have no pre-existing direct trust relationship. This paper addresses some of the security issues in Web Services composition in the context of construction e-purchasing. The framework presented in this paper allows Service Providers and Service clients to dynamically exchange security requirements and capabilities to determine how they can share their e-resources. We describe some applications of these concepts and show how they can be integrated into a Web Services environment for construction epurchasing.

1 INTRODUCTION

Authorization is the process of assigning privileges to users and then determining whether an authenticated entity can be granted access to a requested resource under the given circumstances. The latter stage is termed access control and is the process of constraining access to protected resources to only those users who have valid privileges. However, the use of personal, sensitive information, such as privileges, to gain access to a resource in a Web Services (WS) environment raises an interesting paradox. On the one hand, in order to make the services and resources accessible to legitimate users the authorization infrastructure requires valid and verifiable service clients' attributes or privileges. On the other hand, the service clients may not be prepared to disclose their privileges or attributes to a remote Service Provider (SP) without determining in advance whether the service provider can be trusted to adhere to their confidentiality and privacy. Thus, confidentiality and privacy are critical considerations for distributed authorization systems such as secure Web Services environments. Trust negotiation[1-5] has partly solved this problem, through the gradual negotiated release of privileges to recipients who trusted third parties have vouched for. However, it is not a complete solution to the problem and a standardized framework has yet to emerge.

Privacy is often considered from the users' perspective, just as authorization is considered from the SP's standpoint, resulting in unilateral, asymmetric approaches. However the SP may also have sensitive properties such as membership certificates of consortia, or trust relationships with trusted third parties (TTPs), or policies of various kinds that service clients may demand to see before releasing their PII. Furthermore, both parties in WS transactions may require the exchange of service level agreements (SLA), business level agreements (BLA) or other contractual documents on the fly. Enabling the runtime exchange of these business components requires a bilateral, symmetric negotiation to allow the communicating parties to indicate their willingness to accept constraints being imposed on their use of information provided by the other party, before the other party is prepared to reveal their sensitive information. Since sensitive information may change from time to time due to business and/or security expectations, the dynamic negotiation of participating parties' requirements and capabilities is important. We believe that the characteristic of preserving the confidentiality of service outputs and the confidentiality of service meta information (identities, attributes, policies etc.) is getting blurred and the requirements for protecting both types of information are similar and symmetric. Both types of resource may be seen simply as sensitive information resources.

1.1 *Related research*

The Shibboleth federated identity management[6, 7] is a SAML[8] based distributed authentication and authorization system designed to exchange attributes across security realms. It provides a platform for a secure transfer of attributes of a web-browsing user from the user's origin site, Identity Provider (IdP), to a resource provider site usually called the target site, or Service Provider (SP).

When the user first attempts to gain access to a "shibbolized" site, she is redirected to a service called Where Are You From (WAYF) that enables her to pick her identity provider (IdP) to authenticate with. Upon the authentication, the IdP service generates a one-time session parameter known as a handle for this authenticated individual and passes it on to the target site. The purpose of this handle (a temporary reference) is to enable the SP to ask the IdP for desirable attributes that can satisfy its access control requirements, before access is granted to its protected resources. Whilst the Shibboleth model provides a robust authentication mechanism, its authorization component is not fine grained and doesn't support access negotiations. The Platform for Privacy Preferences (P3P)[9] is one approach that attempts to address privacy in commercial websites. Whilst it has provided some degree of privacy awareness, it has not sufficiently addressed privacy concerns particularly in distributed authorization systems. P3P has also not satisfactorily resolved the requirements for bilateral privacy negotiation. Anonymity schemes [10] have attempted to address confidentiality and privacy problems in some cases. Though anonymity may be the only failsafe option in certain situations, in many cases it is not a tenable option since parties must disclose one or more identifying attributes in order to obtain services. In this paper we are particularly interested in those cases requiring the confidentiality of information at the consuming endpoints. In particular, can the receiving party be trusted to keep the items confidential based on the sender's security preferences? Recent research efforts in the field of trust negotiation[5, 11, 12] favour dynamic access to services, through the gradual negotiated release of personal identifying and service provider attributes, so that trust can be incrementally increased until the user is satisfied that the SP is trustworthy enough to be sent all their confidential attributes. Trust is built upon the assertions of trusted third parties, rather than on the communicating parties themselves, and neither party provides an enforceable commitment or obligation to the other party. Our work builds on that of trust negotiation, by supplementing it with obligations provided by the communicating parties themselves.

The rest of the paper is structured as follows: Section 2 gives an overview of the problem space. In section 3, we describe in detail WS-XACML platform for enabling authorization framework for secure WS transactions. Section 4 presents a general discussion on the proposed solutions and section 5 concludes the paper.

2 PRIVACY AND TRUST CHALLENGES IN WEB SERVICES ENVIRONMENTS

Authorization in WS systems presents a number of significant challenges. First, the discovery of services and the information needed to access them may have to be handled dynamically in the sense that services and players can change regularly without notification. Thus services can be added or withdrawn, as well as the requirements for gaining access to them. The degree of trust may vary from one participant to another making negotiation of services desirable. Second, the service providers, service users and Trusted Third Parties (TTP) are unlikely to belong to the same security domain in all scenarios. In some cases service users will not have pre-existing relationships with the service providers. Third, all the credentials (signed privilege assertions) are unlikely to be issued by one TTP, thus a collection of different TTPs may be involved in the negotiation and authorization operations.

Enforcing constraints and obligations at a remote service provider or service client end point is obviously difficult. Furthermore, current authorization systems make an assumption that service requesters have prior knowledge of access control requirements; in open systems with diverse and unbounded communicating parties, this may not be the case. To solve this problem, users may be made to submit more credentials than are necessary, which potentially exposes them to unnecessary privacy risks, or they will need to participate in a trust negotiation protocol. Since electronic resources may change over time, along with their privacy needs, the dynamic disclosure of access requirements and capabilities, and the appropriate credentials is desirable. At request time, both parties can make known their access requirements and capabilities, and they can check whether they can meet those requirements. Again, all parties need to evaluate the risk of giving out their information, and determine the degree to which they are prepared to trust the remote parties. They will also need to identify any constraints and obligations they may wish to place on the other parties. Trust negotiation partially solves the problem via the gradual releasing of confidential information to the other party, but it does not address the problem of issuing constraints and obligations. Furthermore the TTPs upon which trust negotiation is based, are usually trusted to assign privileges or attributes to the negotiating parties, rather than to make statements about their privacy policies. It is worthwhile mentioning that P3P only facilitates the communication of privacy statements and has no enforcement mechanisms. It neither sets standards for privacy compliance nor monitors whether sites can adhere to their own statements, but it does describe "disputes" and "redress" mechanisms in the event of violations. Thus, a comprehensive privacy infrastructure should devise a way to ensure that service providers act according to their policies. This requires automated enforcement machinery and an electronic legal discovery system in the event of disputes but these topics are out of the scope of this paper.

A P3P statement has no strong binding to the service providing party - it is the service user that "relies" on it - so this is particularly vulnerable to exploitation; for example, in a site scripting attack, the attacker can fool the browser by embedding the right P3P policy. Hence, we can say that P3P is not presented in a tamper-resistant way. The origin is not verifiable and cannot be validated, except for server authentication based on third party signed sever certificate. This must limit its use in a high risk business environment where non-repudiation is important.

3 WS PRIVACY AWARE AUTHORIZATION SCHEME

In network systems, authentication verifies an identity claim made by an entity, but does little to say or predict the capabilities and/or intentions of that entity. Thus authentication is not sufficient to predict the behaviour of any entity without obtaining other properties of that entity. In Shibboleth [13], user authentication and the user's privilege attributes are provided by the user's Identity Provider (IdP) (the Authentication Service (AS) and Attribute Authority (AA) components respectively), but a user's identifying attribute is not released (unless it is one of the attributes provided by the AA). Privacy is ensured by the use of a signed one-time opaque authentication handle which hides the true identity of the user from the target site. This is fine as long as the SP doesn't require any identifying attributes to complete the service, but this is unlikely to be the case in most transaction scenarios. In this case the AA will need to provide these attributes to the SP, so the IdP has an Attribute Release Policy (ARP) to say which SP can receive which user attributes. Since there is no way to guarantee, or even specify, the remote enforcement of privacy obligations, the receiving SP is at liberty to distribute, use or correlate information about the user without being subject to any liabilities. Again, the SP doesn't convey to the service clients the capabilities it tends to offer, so an impostor SP can use this weakness to extract users' personal identifying information. Imperatively, a mechanism for the dynamic composition of requirements and capabilities between the service provider and client is very desirable. The combination of XACML authorization model[14, 15] and XACML SAML profile model[16] bring flexibility in the simultaneous handling of confidentiality and privacy in open systems, and have addressed a number of use cases.

The Web Services Profile of XACML (WS-XACML) which builds on the XACML and SAML standards is suitable for securing services such as construction e-purchasing using the WS platform. Doing this requires the building of technical trust, which may be facilitated by means of public key encryption technology[17]. The essence of trust is to establish confidence among communicating participants. Finding ways to assure each party that their information will be used in accordance with their wishes will increase the level of trust and confidence between the communicating parties and may even reduce the liabilities of regulated organizations such as construction industry. In the construction industry where WS is becoming popular, adding components that can enhance trust and confidence will be beneficial to all parties.

3.1 *WS trust models*

Figure 1 depicts one approach for brokering trust and the issuance of security tokens as defined in [wstrust]. The first part of the figure (i) illustrates the trust relationships that might exist between the various participants of the flow. The second part of the diagram shows the flow of the message. Here, (1) the WSclient obtains security tokens from its trust realm Identity Provider / Security Token Service (IP/STS), (2) these tokens are then presented to the WS –SP trust realm (IP/STS) (3) which can certify the presented tokens or issue fresh tokens to access the services provided by the WS-SP. Precisely, a token from one STS is exchanged for another at the second STS. The claims and tokens can be verified and validated based on the trust relationships shown in the first part of the figure. The initial process of authentication is omitted for simplicity.

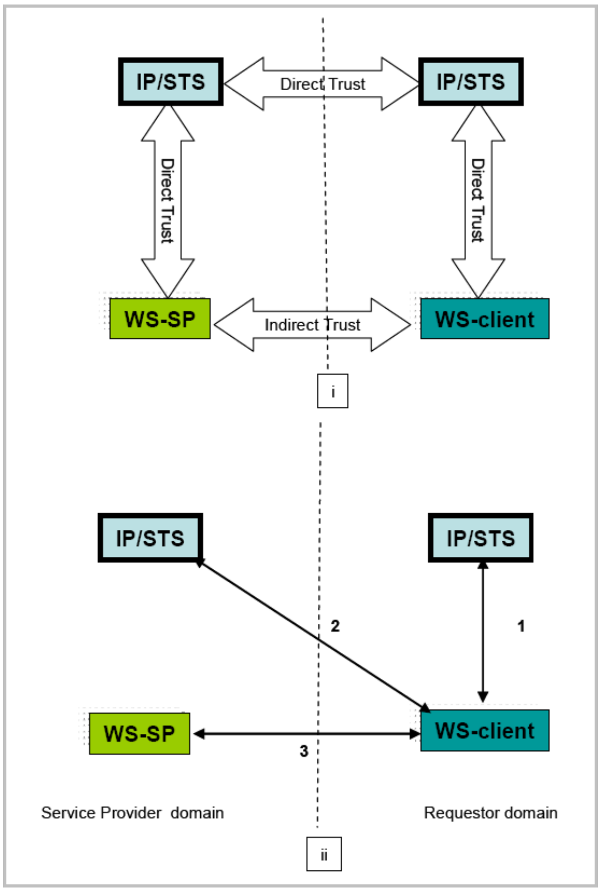


Figure 1. Direct Brokered Trust.

Figure 2 illustrates a second approach; the assumption is that direct trust among all participants is impractical so that an intermediary STS is necessary to broker federated trust among trust realms. However, the intermediary STS must be trusted by the participants for this approach to work. The figure below depicts the basic principles. In part (ii), the WS-client obtains security tokens from its trust realm (IP/STS), (2) these tokens are then presented to the intermediary STS trust realm (3) which can certify the presented tokens or issue fresh tokens, the WS-client presents the new or certified tokens to the WS-SP trust realm to access the services provided by the WS-SP (4). Precisely, a token from the client's STS is exchanged for another at the intermediary STS and another for the WS-SP realm. This federating framework provides the mechanism for building trust relationships among participating parties in our secure e-purchasing model which simultaneously facilitates the protection of resource in relation to confidentiality and privacy. Figure 3 shows the typical trust relationships among the participants. This shows that trust between two entities that is indirect at one level may be direct at the higher layer. In this case, indirect trust between SP and the client is enabled by trust between the Identity Provider / Security Token Service (IdP/STS) and the client.

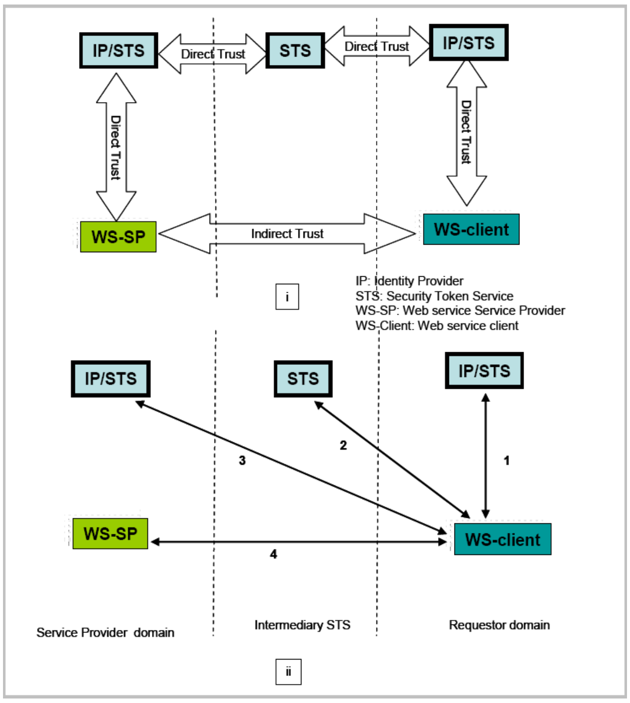


Figure 2. Indirect Brokered Trust.

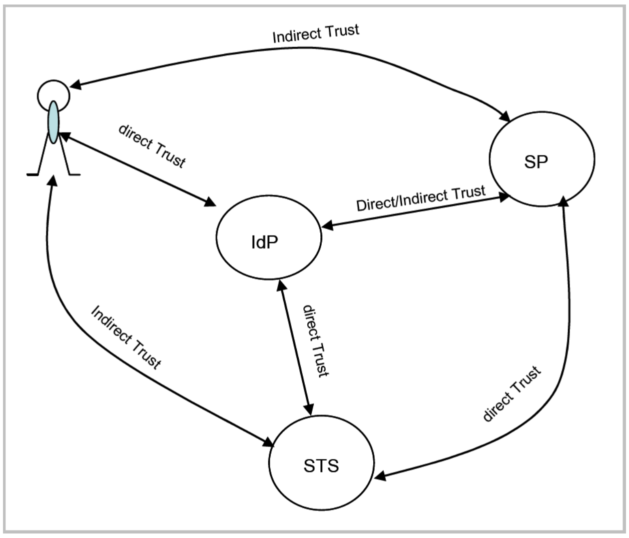


Figure 3. Typical Trust Relationships.

Similarly, indirect trust between the IdP and SP is enabled by trust between the IdP and STS on one hand, and SP and STS on the other hand. The direct trust is enabled by the certificate chain sharing a common trust anchor [17]. The security services are exposed as Web Services which brings flexibility in the handling of authentication and access control of protected resources in the context of construction e-purchasing. Trust here represents the public key infrastructure (PKI) relationship that binds the participating entities including TTPs. Whilst some of the relationships are predetermined offline before the application request, others can simply be composed dynamically; our model attempts to support both. In general, the trust realms should themselves have the infrastructure to assert and verify the claims of entities and providing this mechanism as a web services has a number of advantages. The underlying security framework is governed by security tokens and policies controlled by the different trust realms in the form of capabilities and requirements that need to be matched as depicted in figure 4.

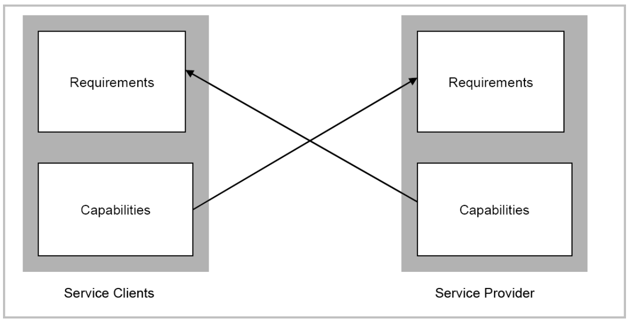


Figure 4. The Security Policy Architecture: Capability and Requirement.

3.2 *WS-XACML authorization model for secure e-purchasing*

The draft Web Services Profile of XACML (WS-XACML)[18] defines an assertion that may be used to convey both requirements and capabilities related to authorization, access control, and privacy for Web Service clients and for the Services themselves. The "XACMLAssertionAbstractType" defined in [12] allows constraints to be specified on a policy by expressing them as XACML <Apply> functions. In a privacy policy, these constraints can be used to describe a user's acceptable or a server's supported P3P policy contents. In [18], an XACML Assertion contains two sets of constraints, namely: Requirements and Capabilities. Figure 4 illustrates the WS-XACML profile architecture. The first set, called "Requirements", describes the information and/or behavior that the policy owner requires from the other party. The second set, called "Capabilities", describes the information and/or behavior that the policy owner is willing and able to provide to the other party. Requirements are logically connected by AND: the policy owner requires the other party to satisfy all of the constraints listed in the Requirements section. Capabilities on the other hand are logically connected by a non-exclusive OR: the policy owner is willing and able to provide any subset of the capabilities described by these constraints.

Two XACMLAssertions match if, for each assertion, each constraint in the Requirements section is satisfied by at least one constraint in the Capabilities section of the other assertion. WS-XACML specifies efficient generic algorithms for determining that one constraint "satisfies" another. We can use this mechanism to evaluate an XACML-P3P policy against an XACML privacy profile (or Shibboleth ARP), providing we have matching semantics between the variant policies. For example, Figure 5 illustrates how a WS-XACML constraint can indicate that a P3P policy either must contain (if in the Requirements section) or can provide (if in the Capabilities section) the "non-identifying information" value for its "ACCESS" section. A typical physical purchasing protocol (for the purchase of an air handling unit (AHU)) is illustrated as follows:

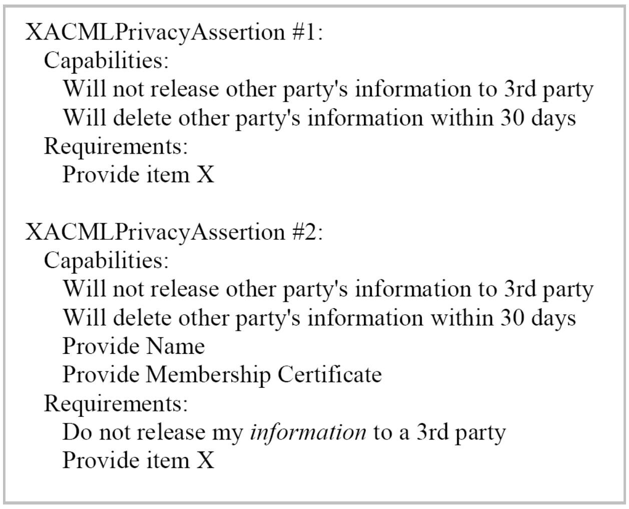
1. Client provides AHU specification to approved suppliers
2. Suppliers use their internal product data to pre-select AHUs which meet client specification

3. Supplier reviews product selections and selects AHU to meet client spec
4. Supplier submits AHU selection to client
5. Client reviews supplier submissions
6. Client selects an AHU from submitted selections (or loop back to 1)
7. Client instructs purchasing department to arrange purchase and delivery of AHU

```
<Apply FunctionId="urn:oasis:names:tc:xacml:3.0:function:must-be-present">
   <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">//P3P10/POLICIES/POLICY/
ACCESS/nonident</AttributeValue>
</Apply>
```

Figure 5. Examples of WS-XACML constraints on P3P Access.

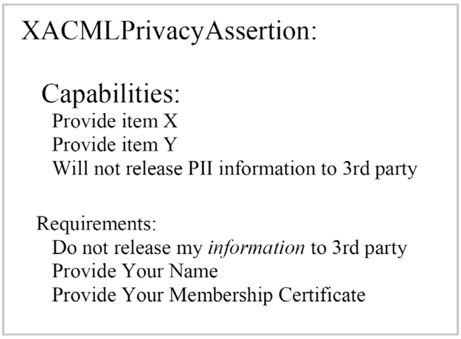
To automate this process using WS, the client and the supplier have to represent their specs and= product information in the form of policies. Using WS-XACML we can model these into requirements and capabilities using the model in figure 4 whilst figure 6 and 7 show the service client and provider policy assertions respectively, so that each party can match their capabilities against the other's requirements. Where the matches succeed they can supply each other with the values in the capabilities sections. To add security to the protocol, we make use of the concepts already presented in this paper involving a trusted third party.



XACMLPrivacyAssertion #1:
  Capabilities:
    Will not release other party's information to 3rd party
    Will delete other party's information within 30 days
  Requirements:
    Provide item X

XACMLPrivacyAssertion #2:
  Capabilities:
    Will not release other party's information to 3rd party
    Will delete other party's information within 30 days
    Provide Name
    Provide Membership Certificate
  Requirements:
    Do not release my *information* to a 3rd party
    Provide item X

Figure 6. Client's Internal XACMLPrivacyAssertion for Item X.



XACMLPrivacyAssertion:

Capabilities:
  Provide item X
  Provide item Y
  Will not release PII information to 3rd party

Requirements:
  Do not release my *information* to 3rd party
  Provide Your Name
  Provide Your Membership Certificate

Figure 7. Supplier's Internal XACMLPrivacyAssertion for Items X and Y.

4 DISCUSSION

In figure 8, we illustrate how the service client and service provider can interact using the framework described on this paper. The service client, in an attempt to request for the available services (items for sale that can meet its spec), is directed to an authentication service in step 2. The service client authenticates with its own IdP in step 3[1]. The IdP issues an authentication token to the STS which asserts the fact that the service client has been authenticated properly and can be given a ticket to approach the service provider (WS Portal). The STS issues an authorization ticket with a reference pointer to the authentication token and passes it to the service provider endpoint. The WS Authorization (WS-Authz) engine, acting on behalf of the service provider, can now convey the service provider's XACMLAssertions, which contain the Requirements and Capabilities of the SP, to the client's WXA , using the authentication reference pointer as the client's identifier. The client's WXA has the client's XACMLPrivacyAssertion also containing Requirements and Capabilities of the client. The client's WXA performs a matching process and if satisfied sends the client's XACMLPrivacyAssertion to the service provider's WS-Authz. The service provider's WS-Authz also performs a matching process and authorization which results to "PERMIT", the service provider can now give item X to the client. At the conclusion of the protocol, both parties have provided capabilities to meet all the requirements of the other party.

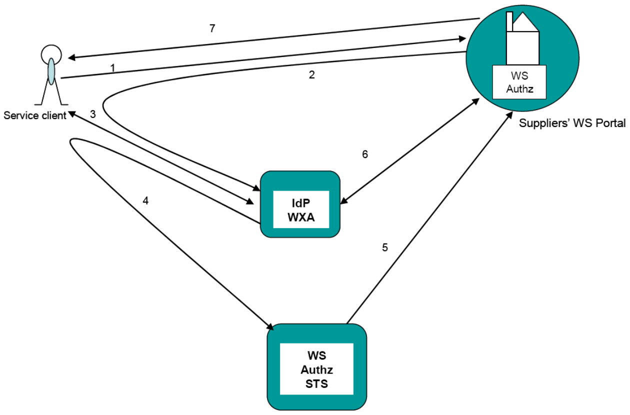


Figure 8. Secure E-Purchasing Platform Protocol.

In some transactions it will be the case that a user's capabilities are insufficient to match a SP's requirements. In this case the user might indicate to the software that if a request for an item is received that contains requirements not covered by any of their sets of capabilities, then the user should be able to view the request and possibly extend their capabilities. As an example, say Service A has agreed not to reveal the user's PII to 3rd parties, but later a new partner Service B offers very generous compensation to any of Service A's account holders willing to sign up for B's new services. In this case, Service A could send the user a new XACMLAssertion containing a Requirement to allow release of information to Service B, along with the Capability to provide compensation. The user

[1] Double arrow depicts several round of communication is possible in both directions

does not have a Capability to match this new Requirement, so the user's client software displays the new Requirement, along with the Capability of offering compensation, to the user. If the user dynamically chooses to accept this compensation, a new XACMLPrivacyAssertion is added to the user's set, for this and future use. This mechanism might be especially useful where the user has used the "namedRecipients" Attribute in the user's Capabilities – it allows the user to increment the list of namedRecipients on a case-by-case basis. Of course some users do not want to be bothered with having to decide about each new potential recipient, so it would need to be controlled by a software configuration option.

5 CONCLUSION

We have presented secure construction e-purchasing platform which enables the bilateral exchange of security requirements and the capabilities to satisfy them. Also, we have given examples of the formal usage using WS-XACML which provides a framework for the dynamic exchange of requirements and capabilities. Our solution demonstrates significant improvement in distributed authorization and the provision of resource control where privacy and trust services are essential.

The WS-XACML framework provides a suitable mechanism in which two or more communicating partners can publicize their requirements and capabilities, and can determine whether their requirements are met by the other parties. Thus WS-XACML provides a standard way for enabling a secure e-purchasing platform so that two parties can propose specific sets of values that satisfy the other’s requirements and when digitally signed ends in an exchange of non-repudiable obligations which satisfy them.

Our protocol has a couple of limitations. Firstly it assumes that the other party exists as a legal entity that can be sued if violations occur. This requires a robust PKI system to exist that will only issue public key certificates to bona-fide organizations and will put meaningful identifying information in the issued certificate. Secondly, it is open to probing attacks. A malicious party can probe another party by providing bogus capabilities in order to gather the other party’s requirements and then terminate the connection before any actual data is transferred. In [5], we address how XACML can be used to minimize the risk associated with the probing attack by trust negotiation i.e. the gradual and incremental exchange of information.

6 REFERENCES

[1] E. Bertino, E.Ferrari, and A. Squicciarini, "Trust Negotiations: Concepts, Systems and Languages," IEEE Computer, pp. 27-34, 2004.

[2] K. E. Seamons, T. Ryutov, L. Zhou, C. Neuman, and T. Leithead, "Adaptive Trust Negotiation and Access Control," presented at 10th ACM Symposium on Access Control Models and Technologies, ,Stockholm, Sweden, 2005.

[3] W. H. Winsborough and N. Li, "Towards Practical Automated Trust Negotiation," presented at Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (Policy 2002), 2002.

[4] T.Barlow, A.Hess, and K.E.Seamons, "Trust Negotiation in Electronic Markets.," presented at Eighth Research Symposium in Emerging Electronic Markets, Maastrict Netherlands, 2001.

[5] U. Mbanaso, G. Cooper, D. Chadwick, and S. Proctor, "Privacy Preserving Trust Authorization using XACML," presented at Second International Workshop on Trust, Security and Privacy for Ubiquitous Computing (TSPUC 2006) Niagara-Falls, Buffalo-NY, 2006.

[6] S. Cantor, "Shibboleth Architecture," Internet2 Middleware http://shibboleth.internet2.edu/shibboleth-documents.html 2005.

[7] S. W. S. Sidharth Nazareth, "Using SPKI/SDSI for Distributed Maintenance of Attribute Release Policies in Shibooleth," Computer Technical Report TR2004-485, 2004.

[8] S. Cantor, J. Kemp, R. Philpott, and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) Version 2.0," vol. 2005: OASIS, 2005.

[9] W3C, "Platform for Privacy Preferences (P3P)," 2004.

[10] S. Fisher-Hubner, Lecture Notes in Computer Science: IT-Security and Privacy, vol. 1958: Springer, 2001.

[11] K. E. Seamons, M. Winslett, and T. Yu, " Limiting the Disclosure of Access Control Policies During Automated Trust Negotiation," presented at Network and Distributed System Security Symposium, San Diego, CA, 2001.

[12] J.Holt and K.E.Seamons, "Interoperable Strategies in Automated Trust Negotiation," presented at 8th ACM Conference on Computer and Communications Security, Philadelphia Pennsylvania, 2001.

[13] T. Scavo and S. Cantor, "Shibboleth Architecture," Shibboleth, 2005.

[14] T. Moses, "eXtensible Access Control Markup Language (XACML) Version 2.0," vol. 2005: OASIS, 2005.

[15] Y. Demchenko, "Using XACML and SAML for Authorisation messaging and assertions:," vol. 2005, 2005.

[16] A. Anderson and H. Lockhart, "SAML 2.0 profile of XACML v2.0," OASIS February 2005.

[17] C. Adams and S. Lloyd, Understanding PKI, 2 ed: Addison_Wesley, 1999.

[18] A. Anderson, "Web Services Profile of XACML (WS-XACML) Version 1.0, WD 8," OASIS XACML Technical Committee 12 December 2006.