Grid Delegation Protocol

Mehran Ahsant[a], Jim Basney[b] and Olle Mulmo[a]

[a]Center for Parallel Computers,Royal Institute of Technology, Stockholm
{ mehrana,mumlo@pdc.kth.se }

[b]National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, USA
{ jbasney@ncsa.uiuc.edu }

Abstract

We propose a delegation protocol based on the WS-Trust specification, which is applicable for a wide range of Grid applications. The protocol is independent of underlying security mechanisms and is therefore applicable to all security mechanisms of common use in Grid environments, such as X.509 proxy certificates, Kerberos based delegation, and SAML assertions. We emphasize that this is work in progress. In this paper, we document our thoughts and current strategy, and we solicit comments and feedback on our approach.

1 Introduction

Delegation is a common requirement for a wide range of Grid applications. Delegation is also supported by various existing security mechanisms in different protocols (Welch, Foster, Kesselman, Mulmo, Pearlman, Tuecke, Gawor, Meder and Siebenlist, 2004). Considering the fact that current in-place security mechanisms will continue to be used, specifying a standard delegation method that can be used uniformly and independently of underlying security mechanisms is an essential effort. By describing delegation as a standalone Web Services portType, and by providing ready-to-use library implementations of this portType, service implementers do not need to deal with the details of the delegation mechanisms, and can factor this functionality out of the internal application logic. Furthermore, delegation can be made part of the functionality provided by the container that hosts the service. In this paper, we describe a delegation protocol, which can be leveraged by diverse Grid applications and environment scenarios. The protocol is compliant with the WS-Trust specification, a technology that can be used to express the specifications required to define a delegation protocol (WS-Trust, 2004). We emphasize that this is work in progress, and with this document we hope to solicit comments from the Grid security community on our proposed approach.

2 Terminology

In this section, we provide the basic definitions and terminologies used by this document for a Grid Delegation Protocol as follows:

Delegation is the act of transferring rights and privileges to another party (the Delegatee).

Delegatee is the delegation target. It is the entity that the Delegation Credential is delegated to.

Delegator is the entity that delegates the abilities and/or rights to the Delegatee.

Delegation Credential is the desired result of delegation protocol. It is a message conveying the abilities and rights from the Delegator to the Delegatee. While the actual syntax and contents of Delegation Credentials are out of scope for this paper, we note that they are typically integrity protected and digitally signed.

3 Usage Scenarios

A wide range of Grid applications make use of delegation. In this section, we describe some usage scenarios in Grid environments that all have delegation mechanisms as a common requirement.

3.1 Delegation of privilages

When a Grid user makes use of a remote resource to e.g. execute a job, that resource may in turn need to access third-party resources (e.g. data repositories) on behalf of the user in order to complete the task. Such access may possibly span across multiple security domains. In this scenario, delegation can be used to delegate (parts of) the user's rights to the remote resource such that it in turn can access the necessary third parties.

3.2 Renewal of delegation credentials

Credentials in general have a validity or lifetime. In case of delegated Grid credentials, their lifetimes are typically short (less than a day) and thus need to be renewed periodically for long-running tasks such as complex scientific applications or ongoing experiments. The need for delegation renewal is essential for Grids (Welch, Siebenlist, Foster, Bresnahan, Czajkowski, Gawor, Kesselman, Meder, Pearlman and Tuecke, 2003). The responsibility of initiating a renewal process may be put either on the Delegatee, or on some external party such as e.g. the Delegator or the originating user (human) that submitted a long-running job. In the case of external parties, they need a mechanism to trigger the Delegatee to request a new delegation credential.

3.3 Online CA

Online certification authorities (CA) tied to the Kerberos (KCA/Kx509) and PKI infrastructure (CACL) can simplify the process of certificate management. Online CAs provide a simple way of PKI enrollment and credential management. Users authenticated to the online CA can request credentials on demand or in advance. Online CAs, which are managed and operated under tight and restricted security policies, can issue long-term or short-term certificates in response to the user's request. Issuing long-term certificates mandates users to manage their credentials in a traditional way of credential management. On the other hand upon on-demand credential request, online CA issues a short-term credential or it might even delegate a short time credential to the user (Basney, Yurcik, Bonilla and Slagell, 2003).

3.4 Online credential repository

Online credential repositories, such as MyProxy, provide secure storage and management services for long-term credentials. Online credential repositories recognize the fact that users are mobile and typically do not properly protect any kind of software-based credentials with a long validity. Instead, users request short-term credentials derived (delegated) from the long-term credentials, maintained by the repository. Besides providing direct access to the end users to retrieve time-limited credentials from the online credential repository in a restricted-access manner, the online credential repository can also act in an "indirect" mode and delegate credentials to a third party

on behalf of the user. Online credential repositories also play a significant role in delegating credentials to web portals when the native protocols, such as HTTPS, do not support credential delegation. Moreover, online credential repositories can be leveraged by Grid services to renew the credentials issued for long lasting tasks before expiration (Novotny, Tuecke and Welch, 2001).

4 WS-Trust

WS-Trust is a specification defined by IBM, Microsoft, RSA and VeriSign that uses the secure messaging mechanisms of WS-Security and aimed to define additional primitives and extensions for security token exchange (WS-Trust, 2004). WS-Trust does not describe or mandate using explicit fixed security protocols: instead, it provides a common SOAP messaging structure that can be used by a flexible set of security mechanisms. The specification provides a simple request/response protocol for issuing, exchanging and validating security tokens and establishing trust relationships between different security realms.

5 GrDP and WS-Trust

We propose a Grid Delegation Protocol, GrDP, building on the WS-Trust specification. The protocol is explicitly and intentionally aimed to be compliant with the WS-Trust specification. The protocol proposed by this document might be implemented for different WS-Trust delegation profiles such as GSI proxy delegation, Kerberos delegation or SAML. Each profile would describe how to use the WS-Trust specification for a particular GrDP implementation. Our current implementation of GrDP, profiling the existing GSI proxy delegation in terms of WS-Trust specification, has essentially the same steps as the current GSI proxy delegation protocol (Welch et al., 2004).

5.1 GrDP requirements

Below, we identify and describe some general requirements of GrDP, and how WS-Trust can be used to meet those requirements.

Transparency The protocol should reduce the burden of understanding the format of security tokens and it should also ease the management of trust relationships with external entities. Thus developer and administrators should not be anymore worried about security formats and mechanisms which are used and known by target domains (WS-Trust, 2004).

Generality GrDP should be used by a wide range of Grid applications in diverse scenarios. Thus it should be general in design and be able to support the most possible scenarios in the Grid environment. WS-Trust provides different operations such as issuance, exchange and validation with the same protocol design and different protocol implementations that helps the GrDP cover various scenarios of delegation in Grid environment (WS-Trust, 2004).

Interoperability GrDP should be interoperable between different security realms which support different kinds of security tokens. It should be able to translate or exchange security tokens into different formats. It should also be able to map the identities and interpret the name-space and attributes of security tokens into the proper format. The protocol should address the issue of trust interoperability as well. Having a security token which is comprehendible by its syntax in other security realms does not mean that other parties can trust the issuer of the security token. GrDP relies on WS-Trust which enables clients and services to interoperate without sharing a common security token format or establishing a direct trust relationship (Madsen, 2003).

Modularity Apparently no protocol can provide a complete security solution for Grid services. Thus GrDP as other Grid-based protocols should have the property of being a building block for the whole infrastructure. WS-Trust has the capability of being used in conjunction with other Web service specifications and a variety of security models and protocols. Thus a protocol based on WS-Trust would be modular (WS-Trust, 2004).

Flexibility GrDP should be able to use different types and models of security tokens. The exchanged credentials by GrDP might be any kind of security tokens. This protocol uses the element "tokenType" specified by WS-Security to pass security tokens inside SOAP messages. The WS-Trust specification allows for describing the type and encoding model of security tokens as well (WS-Trust, 2004).

6 Grid Delegation Protocol

GrDP provides for delegation of rights and abilities in terms of Delegation Credentials between a Delegatee and a Delegator. In this section, we give a general description of this security mechanism agnostic protocol for performing delegation in Grid environments.

6.1 Protocol overview

GrDP is based on a WS-Trust Request/Response message pair from a Delegatee to a Delegator. The Delegatee issues a delegation request and sends that to the Delegator, who verifies the validity of the request, performs the requested operation denoted by the request, and sends the response containing the delegation credential back to the Delegatee (see Figure 1). The UML diagrams below depict different senarios more clearly: in particular, an example where the initiating party is different from the Delegator will be shown.

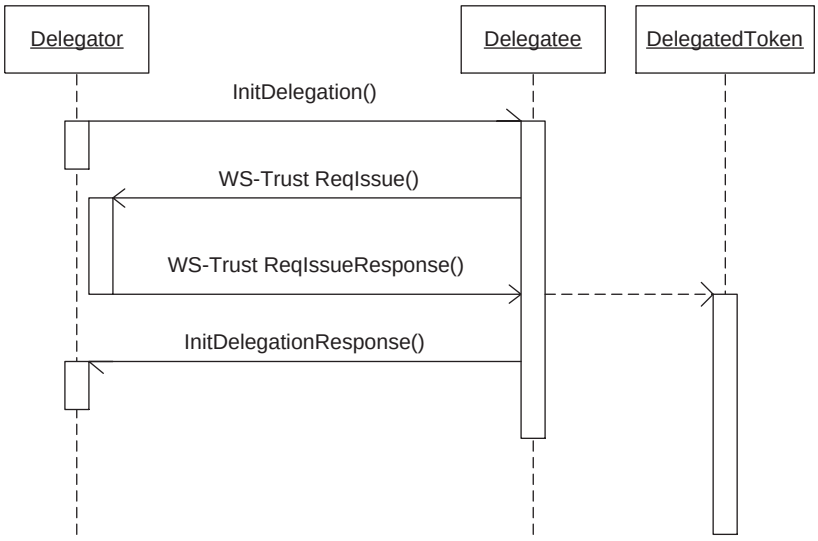

Figure 1: GrDP Request/Response Messaging

We note that this protocol solves a "pull" model (see Figure 3) for obtaining a delegation credential. However, in some use cases, a "push" model is more favorable, for example in case of

a renewal of a previously delegated credential. In those cases, the initiating party (which may be different from the Delegator) makes use of an Initiate message, to trigger the Delegatee to make a delegation request to the Delegator. Finally, the Delegatee may report back to the initiating party on the success status of the delegation operation, as an InitiateResponse message (see Figure 2).

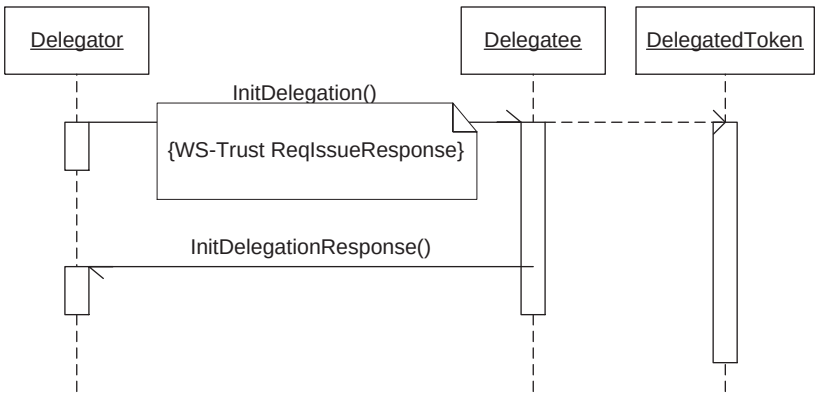

Figure 2: GrDP Push Model

In order to keep the number of roundtrips to a minimum, and in order to provide for the (common) use case where the Delegator already has enough information to do a delegation without the Delegatee needing to get involved, the Initiate message may contain an optional Response message payload with the new Delegation credential.

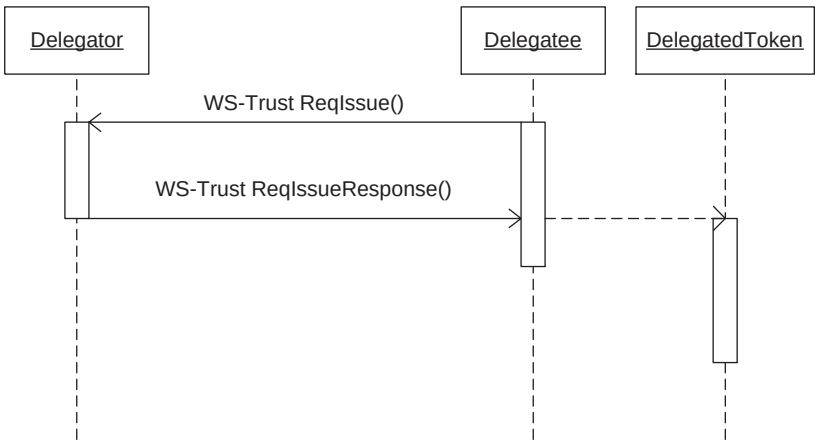

Figure 3: GrDP Pull Model

6.2 Protocol operations

GrDP supports several operations that can be requested by the Delegatee to be performed by the Delegator. We divide the operations of the Delegation Protocol into two main categories: Core delegation operations and Extended delegation operations.

6.2.1 Core operations

Three main operations are supported by the delegation protocol: to issue, renew and exchange delegation credentials.

Issue Creates a new credential according to the needs of the requestor, if the requestor is so authorized. The result is a Delegation Credential that, in a limited and restricted manner, delegates the rights and abilities from the Delegator to the Delegatee.

Renew Renews a Delegation Credential. Renewing only can be performed for those delegation credentials that are marked as "Renewable" for a requested period of time (as discussed later in this paper.)

Exchange The Initiator asks to exchange a Delegation Credential in a different format or even for a different trusted domain. As an example, the Initiator asks to exchange a proxy certificate into a Kerberos ticket. For this operation to perform there might be a need for other translation services to be contacted and used in the back end.

6.2.2 Extended operations

The protocol's extended operations support storing and revoking stored credentials and updating access policies attached to each credential stored in the repository.

Store credential Storing a long-term credential in a safe repository to be used to create short-lived, restricted credentials. As an example, this operation stores a long-term X.509 certificate into the repository to be used by the Delegator to issue short-lived, restricted proxy certificates.

Update access policies For each stored credential in the repository there might be several associated attributes and policies which govern how to access and use the credentials. For example, access policies can specify for which domains and scope delegation is allowed to be performed or the valid formats that credentials can be translated into. The update operation is used to specify and update those attributes.

Revoke credential The Delegator asks to revoke a credential. If the credentials are compromised, the credentials must be revoked. In such case, by revocation, the Delegation Service can be prevented to perform delegation on behalf of a Delegator.

7 GrDP messaging structure

The messaging model of GrDP, outlined in Protocol Overview, consists of three messages: Initiate, Request and Response. In the following we describe each message, including a sample XML schema which depicts the attributes and elements used to construct the message.

Initiate Message: This message may be used by the Delegator to initiate the delegation process by prompting the Delegatee to send its delegation request. It may also contain a Response

message payload, in case no information is needed from the Delegatee before a delegation credential can be created. In the following we illustrate a sample XML Initiate message (see Figure 4) for initiating a GSI delegation process. This message is sent by the Delegator to prompt the Delegatee to request for a new X.509 proxy certificate. This message implies that the Delegatee should provide the Delegator with a PKC#10 certificate request:

```
</S:Header>
    <S:Body wsu:Id="init">
        <RequestSecurityToken>
        <TokenType>grdp:PKCS#10</TokenType>
        <RequestType>wsse:ReqIssue</RequestType>
        ...
</RequestSecurityToken>
    </S:Body>
</S:Envelope>
```

Figure 4: Sample “Initiate” XML schema

Request message: This message is issued by the Delegatee to ask the Delegator for a Delegation credential. The request message specifies both the type of operation, as described in Protocol Operation, and delegation attributes as are specified below:

- *Lifetime:* specifies the lifetime of the delegated credential by specifying the “Create time” and “Expiry time”. A “Create time” dated in the future can be interpreted as a request for a post-dated credential.
- *DelegateTo:* asks the Delegatee for a proxy certificate delegated to a third party.
- *Proxiable:* specifies if the credential delegation can be used to prove a claim for another delegation. In other words, it specifies if the proxy certificate can be used to sign another issued proxy certificate.
- *Delegatable:* specifies that an issued credential delegation can be delegated further. The credential delegation bearer might use a delegatable credential to further delegate obtained rights to another entity.
- *Renewable:* describes if the requested delegation credential would be renewable. Thus it identifies if the issued delegation credential is acceptable for renewing upon exceeding the limit of validity.

The XML schema depicted in Figure 5 shows the Delegatee’s request for issuing a proxy certificate based on a provided PKC#10 proxy certificate request. Figure 6 shows a request for Exchanging (renewing) a proxy certificate for a new time interval.

Response Message: This message containing a delegation credential, which is sent from the Delegator to the Delegatee in response to a delegation request. Figure 7 shows a sample XML schema for Response message.

8 Security Considerations

All GrDP messages must be properly authenticated and integrity protected using standard techniques and protocols. Encryption may also be necessary, depending on the security mechanism used.

```
...
  <Security>
        <BinarySecurityToken
            wsu:Id="ProxyCertToken"
            ValueType="grdp:PKC#10"
            EncodingType="wsse:Base64Binary">
            MIIEZzCCA9CgAwIBAgIQEmtJZc0...
          </BinarySecurityToken>
          <ds:Signature xmlns:ds="...">
              ...
              ...
          </ds:Signature>
      </Security>
      ...
   </S:Header>
   <S:Body>
      <RequestSecurityToken>
         <TokenType>wsse:X509v3</TokenType>
          <RequestType>wsse:ReqIssue</RequestType>
          <Base>
             <Reference URI="#ProxyCertToken"/>
          </Base>
      </RequestSecurityToken>
   </S:Body>
...
```

Figure 5: Sample “Issue” XML schema

```
...
  <Security>
        <BinarySecurityToken
            wsu:Id="ProxyCertToken"
            ValueType="wsse:X509v3"
            EncodingType="wsse:Base64Binary">
            MIIEZzCCA9CgAwIBAgIQEmtJZc0...
          </BinarySecurityToken>
          <ds:Signature xmlns:ds="...">
              ...
              ...
          </ds:Signature>
      </Security>
      ...
   </S:Header>
   <S:Body>
      <RequestSecurityToken>
          <TokenType>wsse:X509v3</TokenType>
          <RequestType>wsse:ReqExchange</RequestType>
          <Base>
              <Reference URI="#ProxyCertToken"/>
          </Base>
       <Lifetime>
           <wsu:Created>...</wsu:Created>
           <wsu:Expires>...</wsu:Expires>
       </Lifetime>
      </RequestSecurityToken>
  </S:Body>
```

Figure 6: Sample “Exchange” XML schema

```
....
   <RequestSecurityTokenResponse>
        <RequestedSecurityToken>
            <BinarySecurityToken ValueType="x:ProxCertToken"
                 EncodingType="wsse:Base64Binary"
                 xmlns:x="...">
                 MIIEZzCCA9CgAwIBAgIQEmtJZc0...
            </BinarySecurityToken>
        </RequestedSecurityToken>
        <RequestedProofToken>
            <xenc:EncryptedKey Id="DelegatorProof">
                ...
            </xenc:EncryptedKey>
        </RequestedProofToken>
    </RequestSecurityTokenResponse>
....
```

Figure 7: Sample "Response" XML schema

9 Current Status and Future Work

This document is a work in progress. We have already defined a WS-Trust profile for X.509 proxy certificate delegation, which binds the GrDP to the WS-Trust specification. The implementation of WS-Trust-based GrDP for GSI proxy delegation is currently underway. Specifying other profiles, such as a Kerberos to GSI translation service (e.g. KCA) and a password to GSI service (e.g. MyProxy) are work for future.

10 Related Work

In this section we mention some other projects which are defined around federation-related activities. TrustBridge, a new technology supported by Microsoft, enables different organizations running Windows operating systems to share their user identity information (TrustBridge, 2002). By leveraging Industry standard XML Web Services, TrustBridge allows participants to exchange user identities and interoperate in a heterogeneous environment. The Liberty Alliance project, by creating the Liberty specification, describes the way of federating network identities in order to enables single sign-on and attribute sharing (Liberty, 2003).

11 Conclusion

We note the significant importance of delegation in a wide range of Grid applications and use cases. The lack of standard delegation mechanisms necessitates the design of a standard and interoperable protocol for delegation mechanism in Grids. We proposed a Grid Delegation Protocol (GrDP), aimed at solving the delegation requirements in Grid environments. The protocol is based on the WS-Trust specification, which operates across security and organizational realms as it is independent of the actual security mechanism used. We re-emphasize that this is work in progress, and that one of our aims with this paper is to solicit comments and feedback from the community.

Acknowledgment: We would like to thank Von Welch for his valuable feedback and comments on this work.

References

Basney, J., Yurcik, W., Bonilla, R. and Slagell, A. (2003). The credential wallet: A classification of credential repositories highlighting myproxy, *Proceedings of the 31st Research Conference on Communication, Information and Internet Policy (TPRC 2003)*, Arlington, Virginia.

Liberty (2003). Liberty alliance project, (Online).
`https://www.projectliberty.org/index.html`

Madsen, P. (2003). Interoperable security for web services.
`http://webservices.xml.com/pub/a/ws/2003/06/24/ws-trust.html`

Novotny, J., Tuecke, S. and Welch, V. (2001). An online credential repository for the grid: Myproxy, *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press.

TrustBridge (2002). Trustbridge project, (Online).
`http://www.microsoft.com/presspass/press/2002/Jun02/06-06TrustbridgePR.asp`

Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S. and Siebenlist, F. (2004). X.509 proxy certificate for dynamic delegation, *Proceedings of the 3rd Annual PKI R&D Workshop*.

Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L. and Tuecke, S. (2003). Security for grid services, *Proceedings of the Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, IEEE Press.

WS-Trust (2004). Web service trust language, (Online).
`http://www-106.ibm.com/developerworks/webservices/library/specification/ws-trust/`