An OGSI CredentialManager Service

Jim Basney[a], Shiva Shankar Chetan[a], Feng Qin[a], Sumin Song[a], Xiao Tu[a], and Marty Humphrey[b]

[a]National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign
[b]Department of Computer Science, University of Virginia

Abstract

We present an OGSI CredentialManager service that allows users to obtain proxy credentials over the network using pass phrase authentication and provides a credential refresh service for long-running jobs. Users can store credentials with the CredentialManager as an alternative to manually managing private key and certificate files. The CredentialManager provides functionality similar to the widely used MyProxy Online Credential Repository, leveraging the standard services provided by OGSI.

1 Introduction

Grid users and their jobs need credentials to access secure grid services but the management of these credentials presents significant security and usability challenges. Initiating grid sessions from multiple locations can require manually copying private key and certificate files, which may open keys to compromise by transferring keys over the network, setting incorrect file permissions on key files, or leaving key copies on vulnerable systems. Additionally, ensuring that jobs will have valid credentials can require delegating long-lived credentials to the jobs, due to difficulties in predicting job lifetime, resulting in an increased time window of vulnerability for the credentials. Also, users must often manage multiple credentials from different credential authorities that give them different privileges on the grid.

The MyProxy Online Credential Repository [1] addresses these concerns by allowing users and security administrators to store grid credentials on a secure server, encrypted with user-chosen pass phrases. User's private keys never leave the server; instead, users retrieve short-term proxy credentials, signed by their private keys, from the server for their grid sessions. MyProxy provides an alternative to user-managed private keys by protecting private keys in one well-secured location and allowing convenient access to proxy credentials from different locations over the network.

The Open Grid Services Infrastructure (OGSI) [2], standardized in the Global Grid Forum in June 2003, defines a new paradigm for grid services. In this paper, we present our implementation of MyProxy functionality for OGSI using the Globus Toolkit.

2 Credential Access

Our OGSI CredentialManager service allows users to store proxy credentials, encrypted by a pass phrase, for later retrieval over the network. As with MyProxy, credentials are stored and retrieved via the proxy delegation protocol [4], in which the delegatee generates a new key pair and the delegator signs a proxy certificate containing the delegatee's public key, so that no private keys are transferred over the network.

1.1 Grid Service Implementation

The OGSI CredentialManager service follows the OGSI factory/instance pattern. To store a credential, the client sends a request to the CredentialManagerFactoryService to create a CredentialManager service instance to which the client then delegates a proxy credential. The client's request to the factory includes

the pass phrase and a name for the credential, which are passed as initial parameters to the newly created service instance. The instance generates a key pair, completes the delegation protocol, and immediately encrypts the private key with the user's chosen pass phrase. The delegated credential (private key and certificate chain), along with other parameters of the service instance, is stored as a persistent service property so the service's state can be recovered after failures or restarts. The lifetime of the instance is set equal to the lifetime of the delegated credential, so instances containing expired credentials will themselves expire and be reclaimed by the OGSI hosting environment. Users can also destroy their service instance(s) at any time (analogous to removing a credential from a MyProxy repository).

The CredentialManager service is implemented for the Globus Toolkit OGSI container using the operation provider model [5]. Initial service parameters are passed in the Factory createService method using extensible CreationParameters, and the standard GridService destroy method is used to destroy the service. The CredentialManager provides only one method in addition to the standard GridService methods: the getProxy method for proxy retrieval. Credential information (name and lifetime) is published via OGSI serviceData. (The XML definitions for the CredentialManager portType and serviceData are included in appendices at the end of this paper.) Leveraging the capabilities provided by the OGSI hosting environment allowed us to implement the CredentialManager service in about 500 lines of code.

1.2 Using the IndexService

When the client creates the service instance, it obtains the Grid Service Handle (GSH) for the new instance. The GSH is a (long) URI that can be used to later locate the service instance to retrieve a proxy credential. It is not convenient for users to remember this GSH, however, so we allow users to choose names for their CredentialManager instances when they are created. Each instance advertises its name, along with other parameters, to the Globus Toolkit IndexService, and CredentialManager clients query the IndexService to obtain the GSH for an instance with a particular name. This results in a user interface very similar to MyProxy, whereby users can retrieve a proxy credential by name and pass phrase, without needing to know the internal details of GSHs and service instances. CredentialManager service instances re-register with the IndexService periodically in case the IndexService is restarted. If multiple credentials are registered with the same name, the CredentialManager clients display an error message and require the user to choose one of the matching instances by GSH.

Table 1. Comparing Credential Management Operations

MyProxy	OGSI CredentialManager
Store proxy in repository	Create persistent service instance containing proxy
Remove proxy from repository	Destroy service instance
Retrieve proxy by name and pass phrase	Locate instance by name using IndexService then retrieve proxy with pass phrase

1.3 Authorization

The CredentialManager service leverages the authorization methods provided by the Globus Toolkit OGSI hosting environment. To control who may store credentials with the CredentialManager, the system administrator can set an access control list (using "gridmap" authorization) for the CredentialManagerFactory service in the hosting environment configuration file. Only authorized users will be able to access the factory to create service instances to hold credentials. This is analogous to the MyProxy "accepted_credentials" policy.

To control who may retrieve credentials, the system administrator can set an access control list on the CredentialManager instances by setting the "instance-authorization" parameter in the hosting environment configuration file. For example, the administrator may allow only trusted grid portals to retrieve credentials, much like how the MyProxy "authorized_retrievers" policy is used today. By default, the CredentialManager allows any client presenting the correct pass phrase to retrieve a credential, including so-called "anonymous" clients that do not have an existing proxy credential. This allows users to obtain their initial proxy from the CredentialManager when needed, without needing to manage long-term certificates and private keys in files on their workstations.

Unlike the current MyProxy service, it is not currently possible to set access policies on individual credentials (i.e., individual CredentialManager service instances) because the capability to dynamically manage service authorization is not provided by the Globus Toolkit OGSI hosting environment. Also, the

current MyProxy service allows regular expressions in its access control policies but the Globus Toolkit gridmap authorization does not.

Table 2. Comparing Access Control Policies

MyProxy	OGSI CredentialManager
accepted_credentials	<service> <parameter name="authorization" value="gridmap"/> <parameter name="gridmap" value="grid-mapfile.cmfs"/> </service>
authorized_retrievers	<service> <parameter name="instance-authorization" value="gridmap"/> <parameter name="instance-gridmap" value="grid-mapfile.cmi"/> </service>
Per-credential authorization	Not provided

3 Credential Renewal

The CredentialManager service can also be used to automatically refresh the credentials of long-running jobs submitted to the Globus Toolkit ManagedJobService. Credential refresh is useful in cases when it is difficult to predict the run-time of submitted jobs and the user doesn't want to delegate a long-lived credential to the job service for security reasons. In that case, the user instead creates a new CredentialManager service instance, initialized with the GSH for the ManagedJobFactoryService associated with the user's job, and delegates a long-lived credential to the new CredentialManager service instance. The CredentialManager service periodically queries the service data of the ManagedJobFactoryService to find all ManagedJobService instances, then queries the service data of the instances to find jobs owned by the user with credentials nearing expiration. Implementing this functionality required modifying the ManagedJobService to publish this service data. As the user submits new jobs with short-lived credentials, the CredentialManager service will discover them and renew their credentials as needed.

The underlying mechanism for credential refresh is provided by Globus Toolkit OGSI security. A client renews a service's credential by calling any method of the service using WS-SecureConversation and delegating the new credential as part of the WS-SecureConversation handshake. Delegating a proxy credential to the ManagedJobService makes it available to the running job so it can access secure grid services during its run.

The CredentialManager must be able to renew credentials automatically without user intervention, so it is not feasible to encrypt the proxy credential with a pass phrase provided by the user. The CredentialManager currently does not have a reliable long-term encryption key—it typically runs with a Grid Resource Identity Mapper (GRIM) credential that is periodically re-generated with a new key-pair by the Globus Toolkit OGSI container [3]. Thus, the CredentialManager currently must store the user's proxy unencrypted to support credential refresh. It would be convenient if OGSI hosting environments provided secure storage for a service's persistent data.

4 Conclusions and Ongoing Work

The OGSI CredentialManager service leverages OGSI facilities to implement a credential repository service similar to MyProxy. The CredentialManager follows the OGSI factory/instance pattern for stateful services by creating a service instance for storing each credential, mapping the credential store operation to instance creation, and mapping the credential remove operation to instance destruction. The CredentialManager uses OGSI service data to publish information about stored credentials and OGSI lifetime management to automatically discard expired credentials. The Globus Toolkit IndexService provides the facility to map credential names to CredentialManager instance handles, and OGSI security provides the capability to refresh service credentials to support credential refresh for jobs. The OGSI CredentialManager service is available for download from http://myproxy.ncsa.uiuc.edu/ogsa/.

We have begun investigating how our experience with the OGSI CredentialManager service can be applied to the WS-Resource Framework (WSRF) announced in January 2004, and we have done some prototyping of credential managing capabilities using WSRF.NET. We are also developing additional

credential management services based on WS-Trust and the IETF SACRED proposed standard protocols.

5 Acknowledgements

The authors wish to thank Von Welch for his input on the CredentialManager service design.

6 References

[1] Novotny, J., Tuecke, S., and Welch, V. An Online Credential Repository for the Grid: MyProxy. In Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[2] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Snelling, D., and Vanderbilt, P. Open Grid Services Infrastructure (OGSI) Version 1.0. Global Grid Forum Proposed Recommendation, June 2003.

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

[4] Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S., and Siebenlist, F. X.509 Proxy Certificates for Dynamic Delegation. In Proceedings of the 3rd Annual PKI R&D Workshop, 2004.

[5] Sandholm, T. and Gawor, J. Globus Toolkit 3 Core—A Grid Service Container Framework, July 2003.

Appendix A. CredentialManager Port Type

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="CredentialManagerService"
  targetNamespace="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager"
  xmlns:tns="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager"
  xmlns:credential-state="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager/state"
  xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
  xmlns:gwsdl="http://www.gridforum.org/namespaces/2003/03/gridWSDLExtensions"
  xmlns:sd="http://www.gridforum.org/namespaces/2003/03/serviceData"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
<import location="../ogsi/ogsi.gwsdl"
  namespace="http://www.gridforum.org/namespaces/2003/03/OGSI"/>
<import location="credential_state.xsd"
  namespace="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager/state"/>
<types>
<xsd:schema
  targetNamespace="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  xmlns="http://www.w3.org/2001/XMLSchema">
<xsd:element name="getProxy">
      <xsd:complexType>
            <xsd:sequence>
                  <xsd:element name="certreq" type="xsd:base64Binary"/>
                  <xsd:element name="pass" type="xsd:string"/>
                  <xsd:element name="lifetimeInHours" type="xsd:int"/>
            </xsd:sequence>
         </xsd:complexType>
   </xsd:element>
   <xsd:element name="getProxyResponse">
      <xsd:complexType>
         <xsd:sequence>
```

```
            <xsd:element name="valueReturn" type="xsd:base64Binary"/>
          </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
</types>
<message name="GetProxyInputMessage">
    <part name="parameters" element="tns:getProxy"/>
</message>
<message name="GetProxyOutputMessage">
    <part name="parameters" element="tns:getProxyResponse"/>
</message>
<gwsdl:portType name="CredentialManagerPortType"
  extends="ogsi:GridService ogsi:NotificationSource">
    <operation name="getProxy">
        <input message="tns:GetProxyInputMessage"/>
        <output message="tns:GetProxyOutputMessage"/>
        <fault name="Fault" message="ogsi:FaultMessage"/>
    </operation>
<sd:serviceData name="CredentialState" type="credential-state:CredentialStateType"
  minOccurs="1" maxOccurs="1" mutability="mutable" modifiable="false" nillable="false">
            <documentation> Credential Status as SDE </documentation>
 </sd:serviceData>
</gwsdl:portType>
</definitions>
```

Appendix B. CredentialManager Service Data

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="CredentialStatus"
  targetNamespace="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager/state"
  xmlns:tns="http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager/state"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
<schema
  targetNamespaces=http://www.ncsa.uiuc.edu/namespaces/2003/07/CredentialManager/state
  attributeFormDefault="qualified" elementFormDefault="qualified"
  xmlns="http://www.w3.org/2001/XMLSchema">
    <complexType name="CredentialStateType">
        <sequence>
            <element name="CertName" type="string"/>
            <element name="TerminationTime" type="dateTime"/>
            <element name="Destroyed" type="string"/>
        </sequence>
    </complexType>
    <complexType name="RegistryEntryType">
        <sequence>
            <element name="handle" type="ogsi:HandleType"/>
            <element name="credential" type="tns:CredentialStateType"/>
        </sequence>
    </complexType>
</schema>
</wsdl:types>
</wsdl:definitions>
```