Experiences using Bridge CAs for Grids*

Jim Jokl[a], Jim Basney[b], and Marty Humphrey[a]

[a]University of Virginia, Charlottesville, VA, USA
[b]NCSA/University of Illinois, Urbana-Champaign, IL, USA

Abstract

Public-Key Infrastructures (PKIs) are widely used for authentication in Grids, due in large part to the success of the Globus toolkit, despite the challenges and difficulties both for PKI administrators and users. The Bridge Certificate Authority (CA) is a compromise between a strictly hierarchical PKI and a mesh PKI and achieves many of the benefits of the hierarchical PKI and mesh PKI but has been untested for use with Grid software. This paper reports on our use of a Bridge CA with the Globus Toolkit v2 and with WSRF.NET. We find that neither software package immediately supports a Bridge CA, and we propose modifications for each software package. With these modifications in place, we believe that Bridges can become an important approach for Grid authentication.

1 Introduction

As more self-contained Grids are created, users will need to cross Grid boundaries. For example, Mary may be authorized to use a particular high-energy physics Grid, and she may also be authorized to use her own local campus Grid. Eventually, there will come a time at which Mary will want to, for example, use her authorized cycles on the high-energy physics Grid and store the resulting data seamlessly on her local campus Grid. The problem is that, it is likely in this scenario that Mary's credential for the high energy physics Grid means nothing to the local campus Grid and vice-versa. There are three main reasons why this could be true. First, each Grid could be independently managed such that the authentication authority in one doesn't even know that the other Grid exists! Second, the high-energy physics Grid might know of but not *trust* the operations of the campus Grid (or vice versa) to the extent sufficient to allow the campus Grid's credential to be allowable proof for authentication to the high-energy physics Grid. Third, the high-energy physics Grid might in theory trust a particular authentication authority for a Campus Grid (and vice-versa) but either does not have the necessary personnel to establish and monitor this trust relationship or cannot make this trust relationship operational via the appropriate legal documents and/or software infrastructures. Given this situation, Mary could conceivably manage her credentials by hand and present them to the appropriate resources by hand, but this is tedious, error-prone, and does not scale.

We are currently developing two different support mechanisms for this use-case. The first approach, which is on-going and not presented in this paper, extends MyProxy [1][2] to handle multiple credentials, so that MyProxy determines which credential is most appropriate for the target service. This approach is particularly valuable in that it is the most general case; however,

* This work was supported in part by the National Science Foundation under grants ACI-0203960 (Next Generation Software program), ANI-0222571 (NSF Middleware Initiative), the National Partnership for Advanced Computational Infrastructure (NPACI), SCI-0123937 (through a subcontract to SURA), and Microsoft Research.

it is also the most difficult to develop because it requires potentially complex integration with the entire client-side Grid environment.

This paper gives a status report on the second approach, which is to make cross-Grid (institutional) trust operational via Bridge Certificate Authorities (BCAs). This work is in part based on our previous work contributing to the Internet2/EDUCAUSE HEPKI-TAG group [3] and the EDUCAUSE effort to build and deploy a Higher Education Bridge CA (HEBCA) [4] in the US. The contribution of this paper is a description/endorsement of the Bridge as an alternative to hierarchical PKIs, a report on observations made while attempting to use the Bridge CA via Globus, and a report on observations made while attempting to use the Bridge CA in our implementation of the Web Services Resource Framework (WSRF [5]) on .NET (WSRF.NET [6]).

.

2 Bridge Certificate Authorities (CAs)

Authentication in Grids is usually accomplished via Public Key Infrastructures (PKIs), which are generally categorized as either a single CA, a hierarchical arrangement, or some kind of mesh structure (see the excellent [7] for more details). With each option, the critical issues are: how to add/remove/revoke a certificate, how to perform path validation, and what is the effect of a compromised private key (anywhere in the system).

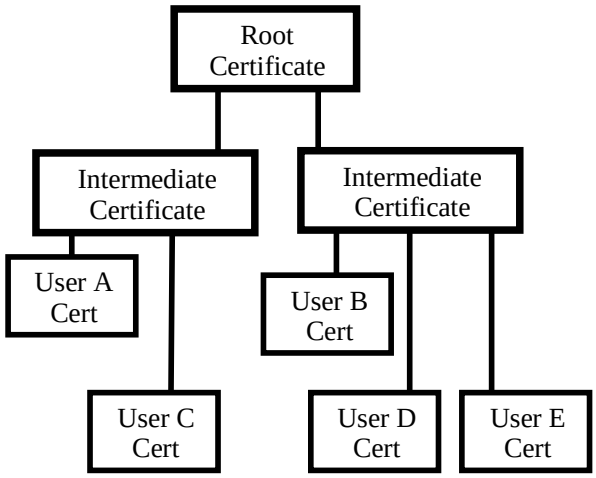

Figure 1: Typical Hierarchical PKI

A typical hierarchical PKI is shown in Figure 1. The "root certificate" is usually a self-signed certificate, which is used to anchor the trust chain. That is, when a certificate is presented to a "relying party", the relying party determines trust in the certificate by validating all of the certificates starting from the user's cert up to a root that is trusted. In Figure 1, the software of the relying party is usually configured (e.g., in Globus) to inherently trust the root certificate. In addition, to make path validation faster for those certificates within the home enterprise, often, the home enterprise CA is often explicitly trusted (e.g., by configuring the software to explicitly trust the appropriate "Intermediate Certificate" of Figure 1). The major advantages of this architecture are that it is scalable, and that there is a unidirectional, deterministic, relatively short path validation. The disadvantages are a single trust point (thus the entire PKI may have to be reconstituted if the root is compromised); the single root may be politically impossible; and legacy PKIs can be very difficult to incorporate.

A variation on the singly-rooted hierarchical CA is for a single Grid to inherently trust multiple CAs by installing multiple CAs in the trusted certificates directory. This is the approach of many Grids, such as the TeraGrid [8]. The problem with this approach is when a

new CA is added, all client installations must be manually updated to reflect the new trust relationship. Similarly, if one of these CAs is compromised, *all* client installations must be updated.

While the hierarchical design imposes a superior-subordinate relationship on existing or new PKIs, it is also possible to connect existing PKIs in peer relationships (this is particularly valuable in those situations in which politics *preclude* a superior-subordinate relationship). In contrast to the hierarchical CA, which have a unidirectional issuing of certificates, CAs form a "web of trust" by issuing certificates to *each other*. Therefore, it does *not* require every CA to agree on a common set of policies and practices. Another value of this architecture is to remove the single point of failure; however, one of the most significant downsides is that path validation may be arbitrarily long and can be extremely complex when taking into account the various policy constraints and extensions defined in X.509.

The Bridge CA is the compromise of the hierarchical CA and the mesh CA, and, arguably, achieves the benefits of each approach. Figure 2 illustrates an example Bridge PKI (the "hub-and-spoke" PKI). In general, the Bridge CA is used to bridge multiple hierarchical CAs. In contrast to the mesh CA in the "web of trust", which may require in the worst case, $O(n^2)$ cross-certificates will be issued, with the Bridge, each CA just cross-certifies with the Bridge. The major advantage is that there is no need to reconstitute the whole PKI if a single CA is compromised. The disadvantage is that it generally requires more infrastructure than just the cross-certificate pairs (as we saw in our experiments reported in Section 3 of this paper).

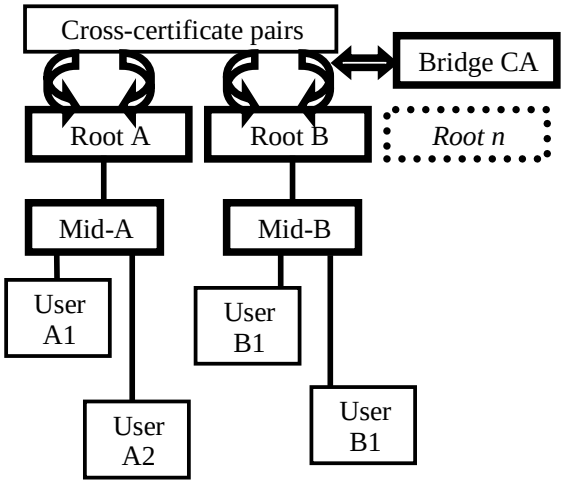

Figure 2: Typical Bridge PKI

One of the important areas in which the hierarchical CA is different than the Bridge CA is path validation, as seen through an example in which Fred from Campus U. attempts to validate *machine1* from Org1. Figure 3 shows the path validation for the Hierarchical PKI, and Figure 4 shows the path validation for a Bridge CA. The hierarchical path validation is fairly straightforward in that all validations are rooted in the same cert, which is the self-signed *rootCA* cert. In contrast, in the Bridge CA, the trust chain is always rooted in the home organization of the entity that is performing the path validation. For example, in Figure 4, because Fred is performing the path validation, the path validation is rooted in Fred's home organization, which is Campus U.

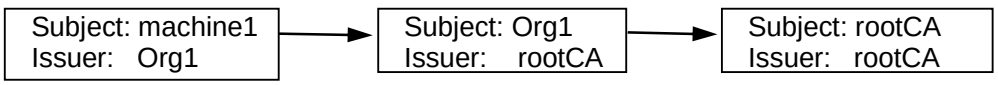

Figure 3: Path Validation for Hierarchical CA

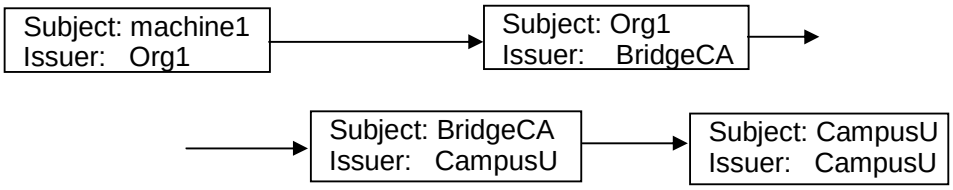

Figure 4: Path Validation for Bridge CA

Note that in general in PKI, when a client attempts to authenticate a service, the service will provide a "candidate" certificate chain, and if the client trusts the root of the candidate chain, path validation is easy. In other words, the chain of Figure 3 is usually provided by the service, and it is *not* built by the client. In a Bridge CA, note that the service will generally *not* provide the certificate chain in Figure 4. Instead, the service will provide a chain of length 2, where the first certificate is the same as the first one shown in Figure 4 and the second is the self-signed certificate from Org1. Since Fred from Campus U. does not trust Org1, it is up to the client to build the certificate chain shown in Figure 4. As we saw in our experiments with Globus/OpenSSL, building this path can be difficult and is not guaranteed.

3 Experiences using Bridge CAs in Grid Software

We conducted investigations using two different Grid software infrastructures with Bridge CAs. In both cases, we set up a Bridge CA at the University of Virginia. Our initial work utilized a simple test bridge that we created as part of the HEPKI-TAG effort to understand the native bridge path validation capabilities in Windows XP and Windows 2000. This demonstration bridge is on-line at *http://pkidev.internet2.edu/* . This Bridge PKI consisted of three separate PKIs (*OrgCA, CorpCA,* and *RootCA).* Each of these PKIs had a subordinate CA.

In the work reported in this paper, we updated the CA certificates in this test environment and issued user certificates from the various hierarchical CAs in the bridge. The cross-certificates were installed on the nodes of a test cluster and each node received only the root certificate corresponding to its hierarchical CA in the bridge. Each node simulated a campus in the PKI-Bridged Grid. After this initial experimentation, we then created a new semi-production bridge CA using an off-line dedicated computer and used this new bridge to cross-certify the campus Certification Authorities at the University of Virginia, the University of Alabama Birmingham (UAB), the University of Southern California (USC), and TACC at the University of Texas. The tests that report regarding Globus V2 used UAB and the University of Virginia in particular.

3.1. Globus V2

Version 2 of the Globus Toolkit implements its PKI operations using OpenSSL and so inherits some of the limitations of the OpenSSL software.

Although we cross-certified with UVa, UAB, USC, and TACC, we decided to focus on only a single institution, UAB, when we tested the ability of Globus V2 to correctly process the Bridge CA structure. The first result we found was that Campus CA integration is complicated by the Globus interface. Whereby Campus CA and OS-exported certificates are generally in PKCS-12 format, Globus by default expects PEM files (although we understand that Globus can be made to utilize PKCS-12 files through some user invocations of the openssl routines).

The second result we found using the test bridge CA is that Globus does indeed work with Bridge CAs. Unfortunately, there is no directory-based discovery for cross-certificates as in many bridge environments. Therefore, all of the cross-certificates and certificates for the intermediate CAs must be preloaded into the */etc/grid-security/certificates* directory. A better approach for Globus might be to utilize the *Authority Information Access (AIA)* field in the X.509 certificates to dynamically find the needed certificates. It has also been proposed to

include intermediate CA certificates in the user's proxy chain, so they are delivered to the relying party in the SSL handshake.

Testing the Globus Toolkit using the semi-production Bridge CA yielded an unfortunate surprise: *Globus, through its use of OpenSSL, cannot validate through the Bridge CA unless the bridge occurs at the root*. Our End Entity certificates in our tests were issued by the UAB CA, which is the root of the UAB PKI. OpenSSL was able to build the necessary certificate path through the appropriate Bridge certificates in order to validate these End Entity certificates. However, the host certificate was not issued directly by the UVa root CA--instead by an intermediate CA at UVa. When OpenSSL within UAB attempted to verify the UVa Host Certificate, OpenSSL could not construct a valid path (although by inspection we could see that a valid path existed). This is unfortunate, but not a surprise--inspection of the OpenSSL source code confirmed our belief that OpenSSL isn't bridge aware in the sense of XP or other similar validators in that it doesn't try to compute all possible paths and then see if any of them work. We are currently investigating other options to solve this problem. For example, it appears that we might be able to configure OpenSSL on the UVa side to not send an entire candidate certificate path (similar to the way *mod_ssl* does for Apache). If we do this, OpenSSL on the UAB might be able to find the correct path. Currently, we are not aware that Globus offers any such configuration option.

3.2. WSRF.NET

One of the projects that we are developing at the University of Virginia is support for the WS Resource Framework on the .NET Framework [6]. In this section, we report on the testing methodology and results regarding how Windows XP interacts with the Bridge CA. We believe that this will be a critical piece for WSRF.NET. The architecture of WSRF.NET is that it directly uses the Microsoft Web Services Enhancements (WSE), which then uses the native Windows XP treatment of certificates. Because .NET inherits many of its behaviors from the underlying operations of the platform (such as Windows XP), of particular interest is how Windows XP uses the URIs in the Authority Information Access (AIA) field of the certificates in the chain being validated. Does it simply accept a single certificate for the immediately superior Certification Authority (CA) for each AIA URI found in a certificate? Will XP accept an object containing multiple certificates? Can more information be provided using LDAP URLs instead of HTTP?

As mentioned in the introduction to this section, as part of our HEPKI-TAG effort, we tested Windows support for a Bridge CA by setting up three hypothetical organizations called *OrgCA, CorpCA, and RootCA,* and bridged them via a *BridgeCA.* The organizational structure can be found on-line at *http://pkidev.internet2.edu/bridge/Bridge3_files/Bridge3_frames.htm* . We then simulated the users of these three organizations using their certificates by applying digital signatures on Microsoft Word documents. Our tests consisted of having a real user simulate a particular user in one of the three organizations by installing the cert/key of that particular user, and then attempting to read the document that was signed by a user in the other organizations. The challenge to Windows XP was to recognize the certificate Bridging structure by downloading the appropriate certificates as contained in the AIA fields. This experiment is detailed at *http://pkidev.internet2.edu/bridge/* (In fact, as discussed below, we are looking for more individuals to perform this test themselves and report to us their findings).

To date, we have found the following results:

1) **Authority Information Access (AIA) URI lookups**
   Windows XP reads a single object using HTTP URLs in the AIA field of the certificate. XP will try all of the URLs in the AIA field in order but will stop after it reads and caches the first entry found. For example, you can not simply add a URL for each object needed and expect Windows to read and cache all of the certificates. It will stop after downloading the first one. Windows appears to assume that the multiple URLs in the certificate all point to

the same information so further lookups after the first successful download are not needed. Windows 2000 reads objects using AIA pointers the same way as Windows XP. The caching strategy appears to be a little different but the overall functionality is similar.

2) **Download of simple certificates**
When referenced via an HTTP URL in the AIA field in a certificate, Windows XP will download and cache a single certificate. The certificate should be in DER format.

3) **Download of PKCS-7 objects**
When referenced via an HTTP URL in the AIA field of a certificate, Windows XP will accept a PKCS-7 object containing multiple certificates. XP does load all of the certificates in the PKCS-7 object into its cache and uses these certificates for path construction and validation. Thus, it is possible to enable bridge functionality by simply adding AIA pointers into EE certificates and populating the referenced PKCS-7 objects with the needed cross-certificates.

4) **LDAP AIA URL Testing**
Windows XP also supports LDAP URLs in the AIA field of EE certs. Hopefully it will also support AIA LDAP URLs at higher levels in the bridge, although we have not tested this to date. Testing has been less successful with LDAP than it was using HTTP and PKCS-7 objects in that we haven't yet been able to get Word XP to find the cross-certs via LDAP when verifying a signature. However, if you download just the EE cert that was used to sign the document and view the cert in the XP cert viewer (in Control Panel/Inet Options), then XP will correctly go to the LDAP server, download all of the cross certs, and create and validate the correct trust path through the bridge. Word XP does not even generate an access attempt entry in the LDAP server log. So, early indications are that not 100% of the PKI functionality in Microsoft applications comes directly from operating system libraries. Testing bridge functionality will need to be done with each application. We are hoping that a few other people will try to reproduce some of these results.

5) **CRL Checking**
If CRL checking is enabled in the operating system, Office XP appears to want to check CRLs while performing document signature verification even if there are not any CRL distribution points present in the certificates.

From these results, we believe that we will be able to support Bridge CAs in WSRF.NET. However, in the short term, we have been unsuccessful using WSE 2.0. While WSE inherits much of its crypto routines and behavior via operating system libraries, it appears that WSE and/or IIS does not actually chase down the information in the AIA fields. This is contrary to the behavior specified in item 1, above. We are currently investigating this further to determine if this is a behavior that we can overload and/or configure, or if WSE is just limited in this respect.

To complete this work, we need to further investigate a number of issues, including testing Windows XP behavior using LDAP URLs in the certificate AIA field, testing with CRLs in the certificates, and adding Outlook and Outlook Express (S/MIME) applications to the testing.

4 Conclusion

We believe that the time is right for increased experimentation with the Bridge as a concrete way in which to “connect” Grids. We have created the Bridge infrastructure and, in this paper, reported on its application to Grids both in the context of Globus V2 and a newly-emerging WSRF implementation (particularly, WSRF on .NET). Our continuing work is to confirm the operations of the existing Bridge CA, extend it by cross-certifying with more institutions, and create a “best practices” recommendation for those institutions wishing to download and install our Bridge CA software.

References

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

[2] MyProxy Online Credential Repository. http://grid.ncsa.uiuc.edu/myproxy/

[3] Higher Education PKI Technical Activities Group (HEPKI-TAG). http://middleware.internet2.edu/hepki-tag/

[4] Higher Education Bridge Certificate Authority (HEBCA). http://www.educause.edu/hebca/

[5] Web Services Resource Framework. http://www-106.ibm.com/developerworks/webservices/library/ws-resource/

[6] WSRF.NET: Web Services Resource Framework on .NET. http://www.cs.virginia.edu/~gsw2c/wsrf.net.html

[7] T. Polk and N. Hastings. Bridge certification authorities: Connecting B2B public key infrastructures. In PKI Forum Meeting Proceedings, June 2000

[8] TeraGrid. http://www.teragrid.org