

Simplifying Public Key Credential Management Through Online Certificate Authorities and PAM

Stephen Chan <sychan@lbl.gov> Matthew Andrews <mnandrews@lbl.gov>

Abstract

The secure management of X509 certificates in heterogeneous computing environments has proven to be problematic for users and administrators working with Grid deployments. We present an architecture based on short lived X509 credentials issued by a MyProxy server functioning as an Online Certificate Authority, on the basis of initial user authentication via PAM (Pluggable Authentication Modules). The use of PAM on the MyProxy server allows credential security to be tied to external authentication mechanisms such as One Time Password (OTP) systems, conventional LDAP directories, or federated authentication services such as Eduroam. Furthermore, by also leveraging PAM at the authenticating client, X509 certificates are transparently issued as part of the normal system login process. When combined with OTP authentication, both OTP and PKI become more manageable and secure. When combined with federated authentication services such as Eduroam, large, distributed user populations can have instant access to X509 credentials that provide transparent single sign-on across virtual communities that span sites, countries and continents.

Motivations

The usability and security issues of X509 certificates have been a concern for users and administrators of Grid computing for the past several years. Beckles, Welch and Basney[1] summarized the observations made in the community, as well as directions for future development. Whitten and Tygar[2] described the broad security issues with PKI and the usability issues of another PKI tool, PGP. We believe that many of the usability issues identified by Whitten and Tygar also apply to openssl, the tool generally used to manipulate X509 certificates as part of Grid certificate management practices. In fact, Whitten and Tygar evaluate a graphical user interface to PGP, which is arguably simpler for end users than a complex and overloaded command-line interface such as openssl.

Summarizing the usability and security issues from these two papers we have the following:

1. Users are sometimes unaware of, or unmotivated by, the necessity for strong passphrases to secure their private keys, and there are no administrative controls to enforce passphrase quality. It is widely observed that in the absence of strong password/passphrase enforcement mechanisms, low quality (or even null) passphrases are often chosen by users.
2. Users are not always aware of the necessary filesystem permission settings on private keys to maintain security.
3. Credentials may be stored on shared network filesystems that are vulnerable to sniffing or authentication compromise (as well as exposure due to inadequate permissions settings).
4. Certificate revocation is not uniformly deployed by certificate authorities, nor is it uniformly checked by relying parties.
5. If a user's passphrase is lost or forgotten, the only recourse is revocation and re-issuance of the certificate.
6. The "barn door" property: it is futile to lock the barn door after the horse is gone. Once a secret has been left unprotected, even for a short time, there is no way to be sure that it has not already been read by an attacker – given the problems with securing private keys listed above, it is hard to be confident of the integrity of a certificate. The problem is made worse by the long lifetimes (typically 1 year) of a certificate and the difficulty of ensuring that revocations are effective.
7. Users need to have copies of their certificate and private key at every location where they will use the certificate for authentication. This magnifies the key management issues already described.

8. Tools for manipulating PKI credentials (such as PGP and openssl) have usability issues. Acquiring a Grid credential sometimes requires either generating a keypair and certificate signing request with an openssl based tool, or else exporting the certificate and key from a browser, and using openssl to translate the certificate into a different encoding scheme[3]. Changing passphrases on private key generally requires use of openssl.

In addition, keylogging has become more common in exploits and malware - until such time as secure virtual machines that are somehow keylogger-proof[4] are deployed, the security of any secret protected by a static password/passphrase is in question.

In response to the proliferation of keyloggers, One Time Passwords (OTP) have been evaluated[5] and deployed at many sites. One Time Passwords bring their own usability issues:

9. Sites typically have their own OTP systems, and cross vendor, cross realm compatibility is often lacking. Consequently, users may be forced to have an individual OTP token per site where they have an account.
10. Asking users to authenticate with a different password every time they log into the same system may prove onerous, especially in environments where Single Sign-On authentication (Kerberos, Globus GSI, etc...) is the norm.
11. OTP mechanisms are not compatible with batch job schedulers, or many unattended distributed systems platforms.

We have worked to address the usability and security issues around X509 certificates and One Time Passwords in our design, however the solution is not tied to One Time Passwords and is compatible with many legacy and future authentication systems.

Deploying a MyProxy based Online Credential Authority

MyProxy[6] has been used as an online credential repository in the Grid Community for several years and has been undergoing constant development. Historically, Grid Authentication has been done with proxy certificates, which are short lived certificates signed either by the user's end entity certificate or by another proxy[7]. Because proxies are short lived, the consequences of compromise are limited in time. Therefore, it is

considered an acceptable risk to store the proxy certificate credentials unencrypted, but protected with secure file permissions. With an unencrypted proxy, the user no longer needs to enter a passphrase to decrypt the private key at each authentication. Assuming the relying party trusts the certificate authority that signed the user's certificate, the certificate chain from the proxy to the CA can be used to authenticate the user.

Proxy certificates vastly simplify the authentication process, allowing Grid users to have single sign-on across physically and administratively distributed systems. Systems in different administrative domains can decide independently if they will accept an individual certificate, and map the certificate into a local account. This provides for single sign-on across a collection of loosely coupled systems.

Normally users need a copy of their personal certificate credentials at every location where they may want to generate a proxy - for users with many accounts across many machines, this often means copying the credentials to each working account on the different machines. This creates security and logistical issues because all credential copies must be managed properly: file permissions, passphrases and revocation/renewal must be applied to each certificate at each location. As the problem gets larger, the temptation to take shortcuts and the likelihood of errors inevitably becomes greater.

The MyProxy service addresses these issues by allowing the user to store a set of longer lived proxy credentials on a central server. After authenticating to the MyProxy service, a client can then locally generate a new key-pair, and request that the stored proxy credentials sign a short-lived proxy certificate for those local credentials. In this way, users can generate a signed proxy from any location that has network access to the MyProxy server, without needing to manage multiple copies of their personal certificate credentials.

In response to the threat posed by keystroke loggers, a roadmap for integration of MyProxy with OTP was described by Basney, Welch and Siebenlist in 2004[8]. Since then, development on MyProxy has progressed along the roadmap:

- NCSA has added support for OTP using PAM[9]
- Code from Monte Goode and Mary Thompson of Lawrence Berkeley Lab was included in the MyProxy 3.0 release that supported online Certificate Authority (CA) functionality[10]. The Online CA serves as a certificate authority that returns a signed short lived end entity certificate to the client instead of a short lived proxy certificate. So

long as the relying parties trust the certificate used by the MyProxy online CA to sign the certificate request, this certificate is valid for Grid authentication, or any other X509-based authentication. By using an online CA with short lived certificates, we avoid the key management problems of having large numbers of long lived certificates that need to be managed by either the end user, or the MyProxy administrators.

Our efforts at NERSC/LBL have been to work with Goode and Thompson to specify and test the online CA functionality, and to integrate the MyProxy online CA into existing and future authentication systems (PAM, OTP and Kerberos). We have developed PAM modules that make the process of acquiring certificates from MyProxy and mapping them to Kerberos credentials transparent to end users.

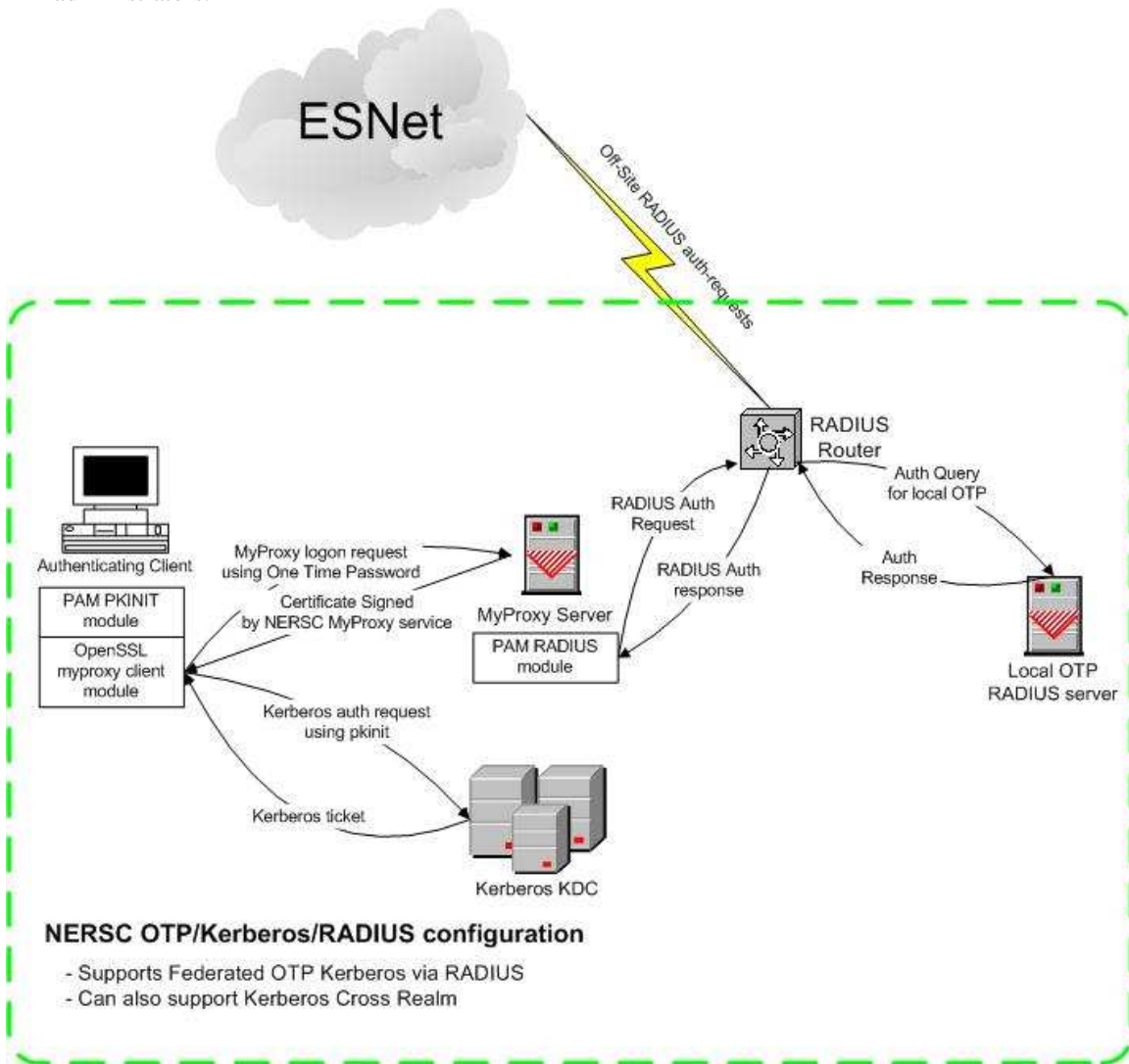


Figure 1: Logical Diagram of NERSC OTP/MyProxy environment

Figure 1 is a logical diagram of the environment being developed and tested at NERSC. It implements the roadmap described by Basney, Welch and Siebenlist as well as introducing a PAM module on the client that transparently acquires a short lived credential from the MyProxy service and uses it to acquire a Kerberos credential. For sites that do not

require Kerberos, we will release a PAM module that implements only the MyProxy credential functionality. The components of the environment are:

- **MyProxy 3+** - configured as an Online Certificate Authority and using a RADIUS PAM module to contact a Radius router

- **Radius Router (FreeRadius)** – configured with a module that queries a local OTP service over an SSL connection. The Radius server is capable of supporting a Radius Authentication Fabric[11] such as Eduroam[12] for authentication federations.
- **Kerberos** – our environment uses Heimdal Kerberos because it has the most mature support for pkinit, allowing X509 certificates to be used to acquire Kerberos credentials.
- **PAM** – We are using a set of patches by Doug Engert to the standard Kerberos5 PAM module[13]. In the current design pkinit calls an openssl engine module to transparently (from the user’s point of view) acquire a certificate from the MyProxy server. Future work will include a standalone PAM module that acquires a certificate from MyProxy without any connection to Kerberos.
- **One Time Password Server** – we use an OTP service developed within the Department of Energy that supports authentication tokens from Cryptocard™. This particular OTP server can be replaced with a different OTP service, or with a static authentication system such as LDAP. An open source FreeRadius module that supports Ansi X9.9 authentication tokens[14] is also available.

The system described here is in development and testing at NERSC/LBL. The MyProxy, Radius, Kerberos and OTP components are in limited deployment to staff members. The pkinit/myproxy integration is in testing, which will provide seamless integration of One Time Passwords, X509 certificates and Kerberos.

The Login Process

In order to demonstrate how this system works in practice we will walk through the steps involved in authenticating a user who is attempting to log into a workstation that uses this system for its authentication service:

1. The Workstation’s login program uses the system’s PAM library to request authentication of the user.
2. The system’s PAM library passes on the authentication request to a pam_krb5 module.
3. The pam_krb5 module has been configured to attempt to authenticate the user via the pkinit extension to the krb5 authentication

- protocol which allows the user to prove his identity using x509 credentials rather than the traditional Kerberos shared secret(password).
4. The system’s krb5.conf specifies the use of an openssl engine module called myproxy_engine to acquire the x509 credentials.
 5. The myproxy_engine module prompts the user for his password using a prompter function which has been passed by reference all the way down the call stack from the original PAM aware application(in this case login.)
 6. The myproxy_engine module generates a public/private keypair, and a certificate request.
 7. The certificate request is then sent to the myproxy server along with the users username, and password as part of a myproxy protocol get request. The myproxy protocol uses the SSL/TLS protocol both to verify the authenticity of the myproxy server,(you don’t want to send a valid password to the wrong server) and to ensure the privacy of the exchange.
 8. Upon receiving the get command, the myproxy server uses the pam libraries on it’s system to attempt to authenticate the user.
 9. The pam libraries on the myproxy system pass the authentication request on to a pam_radius module which uses the RADIUS protocol to a locally trusted RADIUS server. This RADIUS server may verify the validity of the password locally, or forward the request on to a federated system such as Eduroam.
 10. If the RADIUS server confirms the validity of the user’s password, the myproxy server then creates a short lived certificate for that user, and signs it using locally accessible CA credentials(possible stored on a smart card or similar crypto system.)
 11. The myproxy server now returns the new certificate as part of the success reply to the get command, and the myproxy_engine module returns the certificate and keypair to the krb5 library, and stores them in a local file for use by the user if the login succeeds.
 12. At this point the krb5 library uses the certificate to perform a krb5 authentication exchange using the pkinit protocol extension.
 13. When the krb5 Key Distribution Center(KDC) receives the authentication

request, it checks that there is a valid certificate chain linking the certificate used in the request to a CA trusted by the KDC. If the request passes this check, then the KDC checks a local file which provides a mapping of x509 DN's to Kerberos 5 principal names to determine if the entity described in the cert maps to the principal specified in the authentication request. If this check succeeds, then the KDC sends a success reply along with a Kerberos ticket back to the krb5 library on the workstation.

14. The krb5 library finally returns successfully to the pam_krb5 module which stores the Kerberos ticket in a new credential cache, and returns success to the system PAM library, which in turn returns success to the login program.
15. The user is allowed to log into the workstation, and has access to his Kerberos, and x509 credentials which can then be used to access additional services without

additional password entry for a limited amount of time.

Evaluating the Design

We feel that the most important aspects of this approach are:

- Simplifying the process of acquiring and managing X509 certificates for end user by using PAM modules and short lived certificates
- Potential integration with Federated authentication systems such as Eduroam.
- The use of One Time Passwords to avoid the dangers posed by keyloggers

The following table shows the issues identified earlier and how they are addressed. In some cases the issue is totally resolved, in others it mitigates, but does not solve the problem.

| Usability/Security Issue | Response |
|---|---|
| <i>Users are sometimes unaware of, or unmotivated by, the necessity for strong passphrases.</i> | Passwords are in backend authentication system. Centralized password strength checking at backend. |
| <i>Users are not always aware of the necessary filesystem permission settings on private keys to maintain security</i> | PAM module handles short term certificates and keys on behalf of user. Long term certificates eliminated, avoiding those private keys entirely. |
| <i>Credentials may be stored on shared network filesystems that are vulnerable to sniffing or authentication compromise</i> | PAM module handles certificates – can be administratively configured to store creds in filesystem, memory, kernel keyring, HSM, etc. |
| <i>Certificate revocation is not uniformly deployed by certificate authorities, nor is it uniformly checked by relying parties</i> | Short lived (hours to days) certificates mitigate revocation issues. Configurable CA interface allows attributes such as OCSP URL to be added to certs. |
| <i>If a user's passphrase is lost or forgotten, the only recourse is revocation and reissuance of the certificate.</i> | Passphrase/password is in external authentication service (via PAM) and can be changed as appropriate. |
| <i>The "barn door" property: it is futile to lock the barn door after the horse is gone. Once a secret has been left unprotected, there is no way to be sure that it has not already been read by an attacker</i> | Mitigated by short certificate lifetimes and the potential to embed OCSP URL attribute in certificate, enabling realtime revocation, without proving onerous to user. |
| <i>Users need to have copies of their certificate and private key at every location where they will use the certificate for authentication.</i> | MyProxy credential store is originally designed to mitigate this problem. Proposed solution builds on existing benefits. |
| <i>Tools for manipulating PKI credentials (such as PGP and openssl) have usability issues.</i> | Use of PAM module merges certificate acquisition and management into normal login process. No longer necessary for user to be exposed to openssl command line. |
| <i>Sites typically have their own OTP systems, and cross vendor, cross realm compatibility is often lacking</i> | Support for RADIUS fabric allows cross platform, cross site OTP authentication. |
| <i>Asking users to authenticate with a different password every time they log into the same system may prove onerous in environments where Single</i> | Certificate (or Kerberos ticket) provides persistent authentication token. |

| | |
|--|------------|
| <i>Sign-On authentication (Kerberos, Globus GSI, etc...) is already in place.</i> | |
| <i>OTP systems are not compatible with batch job schedulers, or many distributed systems platforms</i> | See above. |

One of the benefits of this design is that it is fully backward compatible with existing systems that either use Kerberos tickets or Grid authentication: the changes only effect how a certificate and/or a Kerberos ticket are acquired. The caveat is that X509 relying parties must include the MyProxy Online CA's certificate in their collection of trusted certificates.

The system also allows any site to issue X509 certificates based on existing username/password based authentication schemes: so long as their system has a PAM interface, it can be plugged into the MyProxy server for user authentication. In an era where passwords and passphrases are vulnerable to keystroke logging, and malware installed by hackers and vendors alike, the value of centrally managed access to certificates should not be underestimated.

Because this approach only effects the initial acquisition of the certificate and Kerberos ticket, there is no performance penalty on any of the subsequent authentication using these credentials. The lifetime of the credentials determines how often new ones have to be acquired – typically sites will have a lifetime of between 1 or 2 working days. On our local systems, it takes a total of under 1.5 secs for the entire process of authenticating against an OTP service, acquiring a X509 certificate and using pkinit to acquire a Kerberos credential. This is a small fraction of the amount of time it takes a user to look up and type in a one time password. We believe that much of the 1.5 secs is due to latencies introduced by communicating with multiple services over the network, and not due to computational overhead.

Because of the infrequent need to acquire new credentials and the brief time it takes to perform the task, we do not believe that performance is an issue with this approach. Additional instances of the server would be desirable to support redundancy, not for performance.

Comparison to Similar Designs

The integration of Kerberos and X509 certificates has been successfully developed and released as part of the kx509 and KCA projects at University of Michigan[15]. OTP and Kerberos integration has been described by Hornstein, et al[16]. FermiLab has successfully integrated these

two efforts into a production service that uses OTP tokens to acquire Keberos credentials, and KCA to translate the Kerberos credentials into x509 certificate[17].

A technical evaluation of the current Kerberos and OTP authentication scheme revealed that the Kerberos server needed to have privileged access to an OTP server, to encrypt the Kerberos ticket with the one time password. This would not be an acceptable design for a federated authentication scheme, where a Kerberos server would need privileged access to a remote OTP service to authenticate a user with a remote site's token.

We investigated approaches that used Radius to authenticate against remote authentication services, and then encrypt the Kerberos ticket using the password. Because the password is the encryption key for the Kerberos ticket, additional layers of encryption and security would be needed to ensure that the password not be exposed to sniffing and decryption. This is especially relevant given the known shortcomings of Radius crypto[18]. In a MyProxy based approach, the private key is locally generated by the MyProxy client, and it never goes over the network. The MyProxy transaction is SSL encrypted, so the password has reasonable encryption – if the PAM module on the MyProxy server is configured to use hashes instead of cleartext passwords for authentication, the user's password need never go over the network in the clear. Along with the fact that the private key does not travel over the network, this approach is significantly more secure when federated authentication is desired.

There are also commercial solutions that integrate Kerberos and One Time Passwords. In our investigations, we found no evidence that these off the shelf solutions would be interoperable among the different OTP vendors. We were also concerned about being locked into a single vendor's solution and not having access to source code, as well the cost for initial deployment and ongoing license fees. Our approach uses open source and/or standards compliant tools where ever possible. In addition, this design is vendor neutral with regards to OTP – so long as an OTP service supports RADIUS, it can operate in the framework.

Lessons Learned

The use of the openssl engine interface to get x509 certs from myproxy was chosen so that existing krb5 applications such as kinit would be able

to work without modification, however this approach has proven to have several problems.

- The engine API provides no standard way to pass a username into the engine so the Kerberos libraries needed to be modified to pass this via a generic engine control interface.
- If authentication fails later in the authentication process, there is no mechanism to go back and clean up the x509 creds stored in the local filesystem.

For this reason it is our intent to move to a system which uses a series of PAM modules, one of which performs the myproxy authentication, and another which performs the krb5 pkinit authentication using the x509 creds acquired, and stored by the first.

Future Work

In an earlier section, we described the goal of developing decoupled PAM modules for MyProxy authentication (without also acquiring Kerberos tickets). We also feel it would also be desirable to add attributes to the X509 certificates and the Kerberos tickets that designate them as having been acquired with a One Time Password. This allows relying parties to enforce policies related to password strength.

In addition to concerns about password strength, relying parties may also want to real-time revocation information about credentials. OCSP is one approach which supports this functionality. Additional attributes in the MyProxy signed certs that point to an OCSP responder is therefore another goal for future work.

Conclusion

The experience of the Grid community with deploying PKI has made clear the usability and security issues around managing certificates. One approach to simplifying the management of certificates is to entirely eliminate long term certificates, and use tools like PAM to embed short term certificates within the existing authentication processes. This is the overall approach we have taken and we believe that the improvements in usability and security are significant. While our approach is Kerberos based, we intend to decouple the MyProxy client code from pkinit, and release the source to a PAM module that uses myproxy directly to acquire a certificate from the MyProxy server, without any Kerberos requirements.

The other usability issue we have tried to address is the adoption of One Time Passwords. By

tying OTP into a single sign-on system, and providing a route for federating authentication domains over Radius, we simultaneously address the usability issues of OTP at a single site, as well as OTP across multiple sites. We believe that this approach has the potential to scale across sites, nations and continents – Eduroam is one of the first examples of a Radius authentication fabric. At the time of writing, Eduroam spans 20 nations[19] and there is interest in expanding further.

Because our approach is vendor and platform agnostic, open source, standards compliant and does not require tight administrative or technical coupling, we feel that it is a good technical starting point for developing scalable, usable and secure authentication infrastructures. Despite the potential for scalability, it is also reasonably easy for a small site to deploy such a system for internal use, and interface it into their legacy authentication scheme.

We have confidence in this overall approach because it builds on the collective experience and collaborative efforts of the DOE Grids and Globus communities. Our design is one example of a new generation of PKI tools for Grid computing which is starting to appear, that builds on the experience of the past several years. This work builds on and has been deeply dependent on the efforts of Monte Goode, Mary Thompson, Jim Basney, Von Welch, Mike Helm, Eli Dart, Steve Lau, William Kramer, Buddy Bland, Scott Studham, Remy Evard, Tom Barron, Dane Skow, Craig Goranson, Gene Rackow, Tony Genovese, Dhiva Muruganatham, Suzanne Willoughby, Anne Hutton, Howard Walter, Frank Siebenlist, Ken Hornstein, Doug Engert, Love Hörnquist Åstrand and the many others who have worked on pkinit.

References

- [1] Beckles, B., Welch, V., Basney, J., “Mechanisms for increasing the usability of grid security”, International Journal of Human Computer Studies, July 2005, vol 63, pg 74-79
- [2] Whitten, A., Tygar, D., “Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0”, Proceedings of 8th USENIX Security Symposium, August 1999, pg 169-183
- [3] “How to request certificates from the DOEGrids CA”, <http://www.doegrids.org/pages/cert-request.html>
- [4] Sinclair, S., Smith, S., “The TIPPI Point: Towards Trustworthy Interfaces”, IEEE Security and Privacy, July 2005, pg 71
- [5] Chan, S., Lau, S., Srinivasan, J., Wong, A., “One Time Password for Open High Performance

- Computing Environments”,
<http://www.es.net/raf/OTP-final.pdf>
- [6] Novotny, J., Tuecke, S., Welch, V., “An Online Credential Repository for the Grid: MyProxy”, Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing, 2001, pg 104-114
- [7]
<http://www.globus.org/toolkit/docs/4.0/security/key-index.html>
- [8] Basney, J., Welch, V., Siebenlist, F., “A Roadmap for Integration of Grid Security with One Time Passwords”, May 2004,
<http://www.nersc.gov/projects/otp/GridLogon.pdf>
- [9] Basney, J., “Using the MyProxy Online Credential Repository”, presented at GlobusWorld 2005,
[http://www.globusworld.org/2005Slides/Session%204b\(2\).pdf](http://www.globusworld.org/2005Slides/Session%204b(2).pdf) pg 15
- [10] “The MyProxy Certificate Authority”
<http://grid.ncsa.uiuc.edu/myproxy/ca/>
- [11] Helm, M., Genovese, T., Morelli, R., Muruganantham, D., Webster, J., Chan, S., Dart, E., Barron, T., Menor, E., Zindel, A., “The RADIUS Authentication Fabric: Solving the Authentication Delivery Problem”, 2005, <http://www.es.net/raf/OTP-final.pdf>
- [12] Florio, L., Wierenga, K., “Eduroam: Providing mobility for roaming users”,
<http://www.eduroam.org/docs/eduroam-eunis05-lf.pdf>
- [13] Engert, D., “Use of PKINIT from PAM”, Heimdal Discuss Mailing list Archives, April 28, 2005, <http://www.stacken.kth.se/lists/heimdal-discuss/2005-04/msg00101.html>
- [14] Cusack, F., “Documentation for pam_x99_auth and rlm_x99_token”, Google, 2002,
http://www.freeradius.org/radiusd/doc/rlm_x99_token
- [15] Doster, W., Watts, M., Hyde, D., “The KX.509 Protocol”, CITI Technical Reports Series, 2001, 01-02,
<http://www.citi.umich.edu/techreports/reports/citi-tr-01-2.pdf>
- [16] Hornstein, K., Renard, K., Newman, C., Zorn, G., “Integrating Single-use Authentication Mechanisms for Kerberos”, IETF Internet Drafts Kerberos Working Group, 2004,
http://www1.ietf.org/proceedings_new/04nov/IDs/draft-ietf-krb-wg-kerberos-sam-03.txt
- [17] Private correspondences and discussions in Grid PKI working groups
- [18] Hassell, J., “The Security of RADIUS”, RADIUS, O’Reilly & Associates, 2002, pg 131-138
- [19] Eduroam web site, <http://www.eduroam.org/>