Planning for an open-source entrant in the PKI interoperability trials

Andrew Findlay

9th June 2001

Skills 1st Ltd
2 Cedar Chase
Taplow
Maidenhead
SL6 0EU

andrew.findlay@skills-1st.co.uk
01628 782565

http://www.skills-1st.co.uk/

netproject
124 Middleton Road
Morden
Surrey
SM4 6RW

info@netproject.com
020 8715 0072

http://www.netproject.com/

netproject/pki/docs/planning.aw

Planning for an open-source entrant in the PKI interoperability trials

Andrew Findlay

9th June 2001

andrew.findlay@skills-1st.co.uk

1: Background

CESG and the Office of the E-envoy conducted a PKI and secure messaging interoperability demonstration during February 2001. The aim was to demonstrate to Government that PKI products now interoperate well enough for individual departments to procure them on the basis of functionality and value for money without worrying about compatibility issues.

A second round of tests and demonstrations is planned for late 2001, with an expanded range of functions to be tested.

Netproject proposed the idea of including one or more open-source entrants in the next round of tests, and was awarded a contract to do an initial study on how this might be achieved. Several benefits are expected to flow from the participation of open-source entrants, including the provision of a non-proprietary 'neutral ground' where vendors will be able to work directly with the code at both ends of a communication.

2: Requirements

The precise definition of the tests to be undertaken has not yet been fixed, but is expected to include basic CA operations (certificate generation, signing, checking, revocation) and S/MIME messaging including both signing and encryption. An open-source entrant will need to implement all these functions.

3: Components needed

There is no complete and packaged open-source 'PKI solution', so some integration work will be required. The major building blocks will be:

Cryptography	The crypto library must implement at least: RSA, DSA, 3DES, SHA-1. Either the library or the Mail Client must implement a Personal Security Environment (PSE) for the storage of keys and certificates. Similarly, support for PKCS and S/MIME coding and decoding is required.
CA	The Certification Authority must be able to work as a root authority or be certified by another CA. It must be able to sign and manage X509v3 certificates and CRLs. It must be able to place certificates in a directory using LDAP. Other communication mechanisms may be required - e.g. for downloading certificates to client systems, and for receiving Certificate Signing Requests.
Directory	LDAP client library will be required by CA and by mail clients. LDAP server will be required for testing and may be required as part of the trials.
Mail client	Must be able to generate, send, receive, and display messages in S/MIME format. Must be able to verify signatures. Must be able to decrypt encrypted messages. May need to retrieve certificates from directory using LDAP. May need to check certificate status using OCSP.
MTA	A Mail Transfer Agent will be needed, to handle SMTP mail routing.
Message Store	A message store will be needed to hold messages awaiting collection by the mail client.
Operating System	To support all the other bits

4: Candidates

A thorough search of open-source resources on the Web was undertaken to locate candidate components. Most effort was devoted to finding crypto libraries, S/MIME libraries, CAs, LDAP libraries, and mail user interfaces. The more likely candidates are listed in Appendices 1 to 4. For completeness, some open-source MTAs, Message Stores, and Operating Systems are listed in Appendices 5 to 7, though the choice of these components is much less critical to this project. Other useful resources and sources of information are listed in Appendix 8.

Crypto and S/MIME

A surprising number of crypto libraries were found, with many being very comprehensive. The most suitable for this project appear to be *OpenSSL* and *Mozilla NSS/PSM*. *Crypt++* with the *Getronics freeware libraries* could also be considered, but these do not appear to have been used in open-source implementations of CAs, MUAs, etc so the integration work would be greater.

The three crypto libraries listed above also include S/MIME handlers to generate and parse PKCS and other S/MIME formats.

Certification Authorities (CAs)

The OpenSSL package comes with a basic CA implementation, but it is not a complete solution. Several packages exist that build on OpenSSL to provide more usable CA functions. *pyCA* and *OpenCA* seem to be the most developed of these. Both provide a web interface. OpenCA is capable of being segmented so that different functions (CA, RA, etc) can be run on separate machines. It also has some support for the Online Certificate Status Protocol. OpenCA's documentation is rather thin at present, and work would be needed on this aspect.

Other packages worthy of note are IBM's *Jonah*, and Leeds University's *LURCIS*. However, it appears that both would require more effort to port to a completely open-source platform than the web-based packages mentioned above.

Directory

There is really only one choice here: *OpenLDAP* provides both an LDAP server and LDAP client-side libraries and utilities. In the context of the formal test, it may be decided that all parties will use one directory (Novell provided it in the first round of tests) but for initial testing a directory service will certainly be required.

Mail Client

In some ways this is the most critical (and difficult) component. It is critical because the MUA is the only part of the system that end-users come into contact with on a day-to-day basis, and it is difficult because it must handle the choice, application, and verification of certificates.

This has proved to be the most difficult component to source. There are many open-source MUAs (Appendix 4 lists just eight but a search of the Web reveals more than 30) but few of them have any support for secure e-mail. Of those that do support security, almost all use PGP or one of its derivatives rather than S/MIME.

The only open-source MUA with any sort of S/MIME support today is *Mutt*, and even that is done through a patch which is not part of the official distribution. Mutt is a very powerful character-mode mail interface which will easily be recognised by users of the earlier *Elm* and *Pine* packages though it does not use any code from them. In its current form, the Mutt S/MIME patch is usable but a bit clumsy - particularly with respect to certificate management. It does not have any LDAP lookup facilities, and the use of CRLs requires too much manual intervention. The S/MIME patch does prove that Mutt has a suitable architecture for secure e-mail support, and the remaining problems could be fixed with a few weeks programming

effort. The work should be directed mostly towards completion of the PSE functions - preferably as a separate module that can be used with other applications.

Of the GUI mailers, the most promising one is the Gnome project's *Evolution*. This has recently been released in Beta-test form, and it already has support for PGP security as well as encrypted connections to MTAs and mailbox servers. On contacting the developers directly it was found that S/MIME support is already well under way, with usable code expected by the end of July 2001. Evolution will use the Mozilla NSS/PSM, which is written specifically to be a portable embeddable component. This should allow it to share certificates and keys transparently with other applications.

A third class of mailer is the web-gateway, typified by *Hotmail* and many similar free-email services. The great benefit of this format is that service can be provided to anyone who has a web browser without requiring any software installation or configuration on their machine. True end-to-end security is not possible, but it can be approached closely by handling S/MIME at the gateway and using SSL connections from there to the browser. At present there do not seem to be any open-source gateways that implement S/MIME, but the facility could be added to *IMP* or *WING* with a bit of effort.

A note on Netscape and Mozilla: Netscape Communicator 4.x has good S/MIME support, and has indeed been used for testing in other trials. Much of the core Netscape code has been released as open source through the Mozilla project. Unfortunately, the S/MIME code was not released - partly because of the U.S. regulatory environment at the time, and partly due to the inclusion of modules owned by RSA Inc. As a result, the current Mozilla release and Netscape 6 which is derived from it do not support S/MIME. Some of the code has now been released, but there is no firm timescale for S/MIME support in Mozilla. Netscape 4.x will be very useful for testing, but it is not open source so we cannot consider it a component for this project.

MTA

The Mail Transport Agent should not be involved in S/MIME transactions, as its job is simply to route and deliver messages. The choice of MTA for the project is therefore not critical provided the package chosen does not modify the body of messages in transit in any way.

Message Store

In a practical situation, mailboxes will be accessed using the IMAP4 or POP3 protocols. A server package will therefore be needed - preferably one that allows both protocols to access the same mailboxes. As with MTAs, the choice is not critical provided a sufficient subset of the protocols are supported.

Operating System

The two major open-source operating systems today are Linux and FreeBSD. Linux is better-known because it has more focus on the desktop environment, and is thus seen by more people. FreeBSD is used mostly in server environments, and is particularly popular with Internet Service Providers and other organisations running mail services. Well-written software should build without difficulty for either environment.

It is also possible to run open-source software on top of closed-source operating systems, but the combination does not seem attractive to developers and there is much less written for this environment.

5: The next stage

Work is in progress to secure funding for the integration work needed to create a complete PKI package. Effort will focus on these components:

Cryptography	OpenSSL
CA	OpenCA
Directory	OpenLDAP
Mail Client	Evolution with Mozilla NSS/PSM

The target is to have a complete package ready for testing by September 2001 so that informal tests can be conducted with other participants before the final intensive test week later in the year.

Appendices

Each appendix tabulates available packages for a particular function. A web source is shown, from which the package can be obtained. *Italicised text* is taken directly from the online description of the package.

Appendix 1: Cryptography and message parsing components

OpenSSL

http://www.openssl.org/
OpenSSL is derived from SSLeay, and is a very active project.

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The OpenSSL toolkit is licensed under a Apache-style licence which basically means that you are free to get and use it for commercial and non-commercial purposes.

OpenSSL implements all the encryption and hash algorithms required for the trial. It also includes X509 certificate handling and S/MIME encoding/decoding. This one package could fulfil several of the component requirements. All the tools supplied as part of OpenSSL are command-line based and require many options when used, so they are more suitable for scripting than for direct use.

Several OpenSSL developers (including most of the core team) are based in the UK.

Cryptlib

http://www.cs.auckland.ac.nz/~pgut001/cryptlib/index.html
Largely the work of Peter Gutmann at the University of Auckland.

cryptlib is a powerful security toolkit which allows even inexperienced crypto programmers to easily add encryption and authentication services to their software. The high-level interface provides anyone with the ability to add strong security capabilities to an application in as little as half an hour, without needing to know any of the low-level details which make the encryption or authentication work. Because of this, cryptlib dramatically reduces the cost involved in adding security to new or existing applications

Cryptlib appears to be purely a library, without any directly-usable tools. The source code is openly available, but commercial use requires a licence from the author.

Mozilla NSS and PSM

http://www.mozilla.org/projects/security/pki/nss/
Network Security Services (NSS) - part of the Mozilla open-source project.

Network Security Services (NSS) is a set of libraries designed to support cross-platform development of security-enabled server applications. Applications built with NSS can support SSL v2 and v3, TLS, PKCS #5, PKCS #7, PKCS #11, PKCS #12, S/MIME, X.509 v3 certificates, and other security standards.

http://www.mozilla.org/projects/security/pki/psm/
Personal Security Manager (PSM)

Personal Security Manager (PSM) is a client-independent desktop security module. It performs PKI operations on behalf of desktop client applications, including certificate and key management, SSL, S/MIME, cryptographic token support, and centralized administration.

Most of this code started life as part of the Netscape Communicator browser. Interestingly, the current Mozilla release has no provision for e-mail signing or encryption.

Catacomb

http://www.excessus.demon.co.uk/misc-hacks/#catacomb

A library of cryptographic primitives. Currently, there are a few block ciphers and hash functions, together with generic modes for building more interesting constructions from them. There's also what used to be a simple key management system, a multiprecision maths library, some public key algorithms and various useful tools.

Cryptix

http://www.cryptix.org/products/cryptix31/index.html

Cryptix 3 is a cleanroom implementation of Sun's Java Cryptography Extensions (JCE) version 1.1. In addition to that it contains the Cryptix Provider which delivers a wide range of algorithms and support for PGP 2.x. Cryptix 3 runs on JDK 1.1, JDK 1.2 (Java 2) and JDK 1.3.

This is one of the few toolkits to include Rijndahl (though we do not need that for the current project)

GnuPG

http://www.gnupg.org/
The Gnu Privacy Guard

GnuPG is a complete and free replacement for PGP. Because it does not use the patented IDEA algorithm, it can be used without any restrictions. GnuPG is a RFC2440 (OpenPGP) compliant application.

OpenPGP technology is a competitor to S/MIME, but is based on the same cryptographic principles. The crypto libraries from GnuPG could therefore be used in S/MIME and PKI applications.

BeeCrypt http://www.virtualunlimited.com/products/beecrypt/faq.html

BeeCrypt is an open source cryptography library released under the GNU LGPL license. It contains implementations of well-known algorithms including Blowfish, SHA-1, Diffie-Hellman, and ElGamal.Two versions of BeeCrypt are available: one written in C and assembler and another written in pure Java.

Unfortunately no use for the current purpose as it does not implement 3DES (though this would be easy to add).

Crypto++ http://www.eskimo.com/%7Eweidai/cryptlib.html
Crypto++ 4.1

Crypto++ is a free C++ class library of cryptographic schemes. It includes all the encryption and hash functions needed for PKI and S/MIME.

Code from many sources, collected and wrapped in C++ by Wei Dai. Crypto++ is required by the S/MIME Freeware Library

Getronics SFL http://www.getronicsgov.com/hot/sfl_home.htm
S/MIME Freeware Library (SFL)

The S/MIME Freeware Library (SFL) implements the IETF S/MIME v3 RFC 2630 Cryptographic Message Syntax (CMS) and RFC 2634 Enhanced Security Services (ESS) specifications. It also implements portions of the RFC 2633 Message Specification and RFC 2632 Certificate Handling document. When used in conjunction with the Crypto++ freeware library, the SFL implements the RFC 2631 Diffie-Hellman (D-H) Key Agreement Method specification.

See also CML, Crypto++, SNACC, ACL

Getronics CML http://www.getronicsgov.com/hot/cml_release.htm
Certificate Management Library

The Certificate Management Library (CML) is described in the CML Application Programming Interface (API) document. It implements the 2000 X.509 certification path processing rules and SDN.706. It meets the majority of the IETF PKIX RFC 2459 Certificate/CRL Profile requirements. The accompanying Storage and Retrieval Library (SRL) (optionally) provides local certificate and CRL storage management functions. The SRL (optionally) provides remote directory retrieval capabilities using the Lightweight Directory Access Protocol (LDAP). It uses the v2.0 Certificate Path Development Library (CPDL) developed by CygnaCom Solutions, an Entrust Technologies company, to provide robust certification path building capabilities such as using cross certificates.

Getronics ACL http://www.getronicsgov.com/hot/acl_home.htm
Access Control Library

The Access Control Library (ACL) provides an Access Control Decision Function (ACDF) that determines if a subject's authorizations (contained in an X.501 Clearance attribute) allow the subject to access data labeled with specific sensitivity values (included in a security label).

Getronics SNACC http://www.getronicsgov.com/hot/snacc_home.htm
SNACC Library

SNACC implements the majority of ASN.1 encoding/decoding rules. SNACC does not support all of the latest ASN.1 features, but there are work-arounds that allow SNACC to be used to produce ASN.1 hex encodings that are identical to those produced by ASN.1 libraries that do support the latest ASN.1 features. Also note that many of the PKIX specs, such as RFC 2459, include 1988--compliant ASN.1 syntax modules which can be directly compiled using SNACC.

Appendix 2: Certification Authority components

OpenSSL http://www.openssl.org/

OpenSSL has certificate signing and management facilities. Tools are command-line only so will need wrappers. OpenSSL is the basis for most other open-source CA kits.

See entry under *Cryptography* for further information.

pyCA http://www.pyca.de/
pyCA - X.509 CA by Michael Ströder <michael@stroeder.com>

pyCA tries to make it easier for people to set up and run a organizational certificate authority which fulfills the need for a fairly secure certification processing. The package also tries to reduce administrative tasks and user's frustration by providing a comfortable web interface to users contacting the certificate authority.

Written in Python. Uses Apache, mod_ssl, OpenSSL. The main developer is based in Germany. Does not appear to separate the CA and RA functions.

OpenCA http://openca.sourceforge.net/

The OpenCA Project is a collaborative effort to develop a robust, full-featured and Open Source out-of-the-box Certification Authority implementing the most used protocols with full-strength cryptography world-wide. OpenCA is based on many Open-Source Projects. Among the supported software is OpenLDAP, OpenSSL, Apache Project, Apache mod_ssl. The project development is divided in two main tasks: studying and refining the security scheme that guarantees the best model to be used in a CA and developing software to easily setup and manage a Certification Authority.

Written in Perl. Uses Apache, mod_ssl, OpenSSL Implements the CA/RA separation. Development team spread around the world, one in UK. Some support for OCSP.

Oscar

http://oscar.dstc.qut.edu.au/

The Oscar Public Key Infrastructure Project at Queensland

Oscar is a project of the Distributed Systems Technology Center of the Queensland University of Technology to create a public-key infrastructure. It implements the cryptographic functions using directly the GMP library, unlike other implementations that use OpenSSL. The programming language chosen is C++. The licensing of Oscar allows the non-commercial usage of the software. To use it commercially, one needs to obtain a license.

Written in C++. No longer being actively developed: apparently the authors are now working on a commercial PKI called µPKI.

Jonah

http://www.foobar.com/jonah/
Jonah PKI

Jonah PKIX is a freeware reference implementation of several standards being worked on in the PKIX Working Group of the IETF. It is written by IBM (and its subsidiaries Lotus and Iris). Jonah makes use of the Foundation Suites cryptographic toolkit made available by Cylink.

There is some doubt about building Jonah on open-source platforms like Linux due to the use of certain commercial toolkits in the original implementation.

http://www.foobar.com/papers/usenix-jonah.html
Jonah: Experience Implementing PKIX Reference Freeware

http://www−4.ibm.com/software/security/keyworks/library/whitepapers/pkix.html
Public Key Infrastructure: The PKIX Reference Implementation Project (aka Jonah)

IDX-PKI

http://idx-pki.idealx.org/
IDX PKI

IDX-PKI is an Open Source implementation of a Public Key Infrastructure which aims to be IETF compliant for PKIX recommandations.

Uses PHP and Postgres. Code is Alpha-level.

LURCIS

http://www.personal.leeds.ac.uk/~ecldh/lurcis/
LURCIS - Leeds User Registration & Certificate Issuing System

CA written in Java. Currently based on Win32 environment. Supports certificate loading into browsers without the use of JavaScript.

Appendix 3: Directory components

OpenLDAP

http://www.openldap.org/
OpenLDAP

The OpenLDAP Project is a collaborative effort to develop a robust, commercial-grade, fully featured, and open source LDAP suite of applications and development tools.

Written in C. Included in most common Linux distributions.

UMich SLAPD

http://www.umich.edu/~dirsvcs/ldap/
UMich SLAPD

The original native LDAP server. This code was the base for the Netscape/iPlanet LDAP server, OpenLDAP, and several others.

No longer in active development.

ISODE Quipu

ftp://ftp.funet.fi/pub/unix/osi/

Open-source implementation of X.500(1984). Includes LDAP access server. This is where all the LDAP developments started. Also includes ASN.1 tools.
No longer in active development.

Appendix 4: Mail Clients

Mutt http://www.mutt.org/

Mutt is a small but very powerful text-based mail client for Unix operating systems. The current public release version is 1.2.5.

Mutt includes support for PGP/GPG security. It is probably the most configurable and powerful mailer available today.

http://elmy.myip.org/mutt/smime.html
S/MIME for Mutt

This patch adds support for S/MIME Version 2 messages according to RFC2311. As with the current support for PGP, this was done by means of an external program. OpenSSL actually provides full support for handling of MIME entities, so it was the natural choice. Unfortunately OpenSSL does not provide a keypair managing system similar to PGP's keyrings, a problem that is also addressed by this patch. Additionally, a small perl script is available, allowing to manage your certificates (add, remove, verify, ..)

Pine http://www.washington.edu/pine/

Pine® - a Program for Internet News & Email - is a tool for reading, sending, and managing electronic messages. Pine was designed by the Office of Computing & Communications at the University of Washington specifically with novice computer users in mind, but it can be tailored to accommodate the needs of "power users" as well. Versions are available for various flavors of Unix as well as for personal computers running a Microsoft operating system.

Like Mutt, Pine has its roots in the Elm mailer and is basically character-oriented though some versions now have GUI features.

Pine has suffered a number of security problems over the years, so may not be a credible platform for this sort of work.

Balsa http://www.balsa.net/

Balsa is a GNOME email client. It supports mbox, maildir, and mh local mailboxes, and IMAP4 and POP3 remote mailboxes. You can send mail via sendmail or SMTP. Optional multithreading support allows for non-instrusive retrieval and sending of mail. A finished GUI similar to that of Eudora lets you view images inline, save message parts, view headers, add attachments, move messages, print messages, and just about anything you would expect in a robust mail client.

Very active development group. S/MIME not currently supported. Some indication that PGP/GPG has been attempted, but not well integrated yet.

http://download.sourceforge.net/janitor/ might provide an interface to a crypto module.

Evolution http://www.gnome.org/gnome-office/evolution.shtml

GNOME Office - Evolution

Evolution's tightly integrated mail, calendar, and addressbook brings your GNOME desktop the ultimate tool for personal and workgroup information management.

A good-looking mail client, well integrated with the other Gnome desktop tools. Just recently out of the alpha-test phase and progressing fast.

Direct contact with developer revealed current work to integrate S/MIME functions using the Mozilla NSS/PSM. This is expected to show fruit by the end of July 2001.

IMP http://www.horde.org/imp/

IMP Web-based mailer

There are currently two versions of IMP. 2.2 is the recommended branch for new users installing IMP. It uses PHPlib for session management, handles MIME attachments much more gracefully than 2.0 (it passes the "MIME Torture Test" that UW makes available), and is generally faster, more optimized, and nicer to the IMAP server than 2.0. It works with both PHP3 and PHP4, which is useful for sites in transition.

Early versions of IMP were very heavy on the backend IMAP server. This one claims to be better. No obvious support for S/MIME but the Web-mail model does not rule it out.

WING http://users.ox.ac.uk/~mbeattie/wing/

Web-IMAP-NNTP-Gateway

WING is an Open Source Apache/mod_perl based system which allows users to access email held on an IMAP server via any web browser. The latest public release is version 0.9.

Very efficient gateway system. Not ‘pretty’ like some other web-based gateways, but that could be fixed. No obvious support for S/MIME but the Web-mail model does not rule it out.

Exmh http://sourceforge.net/projects/exmh

EXMH - Extensible MH Email Interface

exmh is a user interface to the MH (a.k.a. nmh) email system. It is extensible and its users have exploited this to integrate PGP, Glimpse, ispell, faces/picons, MIME, and much more.

Kmail http://www.kde.org/

Kmail - the KDE mailer

KMail is a fully-featured e-mail client. It features support for filters, PGP privacy, inline attachments, multiple POP accounts and drag and drop support of messages and attachments.

Appendix 5: MTAs

Sendmail	http://www.sendmail.org/ Very common mail system, distributed with almost all open-source operating systems and other Unix variants. Can be complex to configure, and has suffered many security problems in the past.
Exim	http://www.exim.org/ Newer MTA, popular with ISPs.

Appendix 6: Message Stores

UWash	ftp://ftp.cac.washington.edu/imap/imap.tar.Z The University of Washington IMAP server.
Cyrus	http://asg.web.cmu.edu/cyrus/download/ The Cyrus IMAP server from CMU.
Others	http://www.imap.org/ IMAP resources and list of packages (at Washington) - includes mail clients

Appendix 7: Operating Systems

Red Hat Linux	http://www.redhat.com/ Probably the most widely-used Linux distribution. Based in USA.
SuSe Linux	http://www.suse.co.uk/ Very popular and extremely comprehensive Linux distribution. Based in Germany.
FreeBSD	http://www.freebsd.org/ Advanced OS based on 4.4Bsd. Commonly used for servers, and very popular with ISPs.

Appendix 8: Other resources

Cloud Cover

http://www.cesg.gov.uk/cloudcover/

CLOUD COVER was a CESG project which aimed to ensure that government departments have access to the widest possible range of secure, interoperable and cost effective PKI solutions

Open-source PKI book

http://ospkibook.sourceforge.net/

The Open-source PKI book

This project tries to collect the necessary information to create a document that describes Public-Key Infrastructures, current PKI standards, explains practical PKI functionality and gives an overview of available open-source PKI implementations. Its goal is to foster the creation of a hign quality open-source PKI.

APKI

http://www.opengroup.org/pubs/catalog/g801.htm
Architecture for Public-Key Infrastructure (APKI)
The Open Group

This document describes the requirements for a Public-Key Infrastructure (PKI). A high-level structure is presented which groups the PKI Architecture's components into broad functional categories. The functionality of each component is described, together with existing specifications which could serve as candidates for each component's interface and protocols. It is assumed that these candidate interface and protocol specifications will serve as base documents for open standardization processes.

NIST

http://csrc.nist.gov/pki/rootca/
NIST PKI Interoperability Testbed

The PKI Interoperability Testbed project is designed to test the interoperability and overall functionality attained using current PKI technology. The project includes PKI components for in-house testing and configuration into different PKI architectures. The current phase of the project is focused upon the Bridge CA concept and involves implementation of a test plan using the pilot federal Bridge CA to link NIST's PKI components to PKIs in other agencies. The resulting PKI will include twelve certification authorities and four X.500 directory servers when complete. This project will provide a sanity check for performance and scalability measures, and serve as a live testbed for X.509 certification path building and validation.

Open Group http://www.opengroup.org/security/sso/sso_scope.htm
Open Group SSO work:

The scope of the Single Sign-On Standard (code-named XSSO at the present), is to define services in support of: the development of applications to provide a common, single end-user sign-on interface for an enterprise, and the development of applications for the co-ordinated management of multiple user account management information bases maintained by an enterprise

The PKI page http://www.pki-page.org/

This page contains links to various sites and documents which are related to Public Key Infrastructure (PKI) stuff, especially links to all Certification Authorities (CAs) I'm aware of.

Stefan Kelm / Secorvo Security Consulting

SACRED http://www.ietf.org/html.charters/sacred-charter.html
IETF Securely Available Credentials (sacred) working group

The credentials used in a public key infrastructure (PKI) typically consist of a public/private key pair, a corresponding certificate or certificate chain and some trust or root certification authority information. They are usually stored on a desktop or laptop system as part of an application specific store. Currently, support for credential export/import is uneven and end users need to get too involved with the mechanics of creating and maintaining their PKI credentials.

This work is at a very early stage, but will be important.

RSA http://www.rsasecurity.com/standards/smime/interop_center.html
RSA's S/MIME Interoperability Centre

S/MIME Interoperability Master Matrix Test Design: Participating vendors will test against Worldtalk's WorldSecure Client, which is the designated reference implementation for S/MIME conformance testing. Vendors will test certification and exchange of messages, as per the Test Guidelines.

Internet2 http://www.internet2.edu/middleware/
Internet2 Middleware page

Pointers to many projects including PKI work.

EEMA	http://www.eema.org/pki-challenge/ EEMA PKI Challenge *"...a two year project funded by the European Commission led by EEMA as part of a 14-strong consortium team, involving PKI technology vendors and other interested parties with the aim of solving interoperability problems between PKI/PKA technologies..."*
S/MIME Examples	http://www.imc.org/draft-ietf-smime-examples Examples of S/MIME Messages *This document gives examples of message bodies formatted using S/MIME. Specifically, it has examples of Cryptographic Message Syntax (CMS) objects, S/MIME messages (including the MIME formatting), and Enhanced Security Services for S/MIME (ESS). It includes examples of most or all common CMS and ESS formats; in addition, it gives examples that show common pitfalls in implementing CMS. The purpose of this document is to help increase interoperability for S/MIME and other protocols that rely on CMS.*
Internet Mail Consortium	http://www.imc.org/ietf-smime/ IMC S/MIME Working Group
PKIX	http://www.imc.org/ietf-pkix/ IETF PKIX Working Group Home page for the IETF group. References to drafts and RFCs. Reference to dumpasn1 program and OID list.
Primes	http://www.utm.edu/research/primes/ The Prime Pages Prime numbers are important to much crypto. Here is some of the theory behind them - and very readable too!
X.509 test suite	http://csrc.nist.gov/pki/testing/x509paths.html test suite of X.509 certificate paths