NISTIR 7313
ISBN 1-886843-39-2

5th Annual PKI R&D Workshop
"Making PKI Easy to Use"
Proceedings

William T. Polk
Nelson E. Hastings
Kent Seamons

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7313
ISBN 1-886843-39-2

5th Annual PKI R&D Workshop
"Making PKI Easy to Use"
Proceedings

William T. Polk
Nelson E. Hastings
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology

Kent Seamons
Brigham Young University

July 2006

U.S. DEPARTMENT OF COMMERCE
Carlos M. Gutierrez, Secretary
TECHNOLOGY ADMINISTRATION
Robert Cresanti, Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
William Jeffrey, Director

Foreward

NIST hosted the fifth annual Public Key Infrastructure (PKI) R&D Workshop on April 4-6, 2006. The two and a half day event brought together PKI experts from academia, industry, and government to explore the current state of public key technology and emerging trust mechanisms, share lessons learned, and discuss complementary topics such as usability. The workshop also served as a forum to review continuing progress in focus areas from previous workshops. In addition to the seven refereed papers, this proceedings captures the essence of the workshop activities including the keynote address, four invited talks, five panels, the work-in-progress session and, new to the workshop this year, an informal rump session.

This workshop began with a variation on a familiar theme: usability. Angela Sasse presented the keynote, “Has Jonny Learnt to Encrypt By Now?”, revisiting Alma Whitten’s keynote from the 2003 workshop. Sasse’s approach emphasizes “value-based design”: by understanding the users’ goals, and designing around them, we can build a more usable system. Features and complexities not essential to the user experience should be hidden by simplifying systems and hiding complexity. Usability was also addressed in a paper session on “Easy-to-Use Deployment Architectures” and panels on digital signatures and browser security interfaces.

Improving the security of infrastructure and applications was another recurring theme throughout the workshop. A presentation on trust infrastructures and DNSSEC by Allison Mankin was given on the first day of the workshop. Although attacking DNS is straightforward, there are few incentives for attackers so DNS poisoning is relatively rare. The low threat level may be one reason that DNSSEC deployment has been slow. A panel on Domain Keys Identified Mail (DKIM), which leverages the DNS for key distribution, was held on the second day of the workshop. DKIM would seem to provide the incentive for attacking the DNS, so perhaps DNSSEC deployment will become a more urgent requirement. Phillip Hallam-Baker’s presentation on the DKIM panel, “Achieving Email Security Luxury” proposed leveraging DKIM, XKMS, and the PKIX logotype extension to create a comprehensive and compelling solution for securing applications and the infrastructure.

Another theme of the workshop was the convergence of PKI and other technologies. Jeffrey Altman’s presentation highlighted progress in the convergence of PKI and Kerberos. A decade’s efforts have produced PK-INIT, PK-CROSS, and PK-APP, forming a comprehensive suite of standards. PK-INIT and PK-APP allow users to leverage PKI certificates to obtain Kerberos credentials, and vice versa. PK-CROSS supports the establishment of Kerberos cross realm relationships with PKI credentials. The “Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy” presentation described the integration of the Grid PKIs, Security Assertion Markup Language (SAML), Kerberos, and one time passwords to support authorization decisions for Grid computing.

Identifying and resolving revocation issues continues to be a topic of critical interest. This year’s workshop featured two presentations at very different levels of abstraction. Kelvin Yiu’s invited talk focused on challenges that had to be faced and compromises required to make revocation usable for consumers in the forthcoming Vista operating system. Santosh Chokhani explored some of the more arcane nuances of the X.509

standard, and their implications for real PKI deployments. A less than cautious approach to CA key rollover or PKI architecture design can introduce circularities in trust paths when validating CRLs or OCSP responses.

The first two days of the workshop also included the ever-popular Works In Progress session. This session allowed presenters to obtain early feedback on ongoing work or projects that are in the early conceptual stages. Major WIP presentations addressed interoperability results for the Suite B cipher suites, progress in the Global Grid, and experiences with securing the DNS. In the rump session, brief presentations questioned old paradigms (e.g., are offline CAs really more secure?) and proposed novel applications of current technology (such as mobile phones as secure containers).

The workshop closed with a half day devoted to PKI deployment issues. The panel on "PKI in Higher Education" had an international flavor, featuring a presentation on the Australian CAUDIT PKI Federation. This was followed by a snapshot of U.S. government PKI deployment activities in the "Federal PKI Update" panel. The workshop ended with a look at leading edge deployment activities in the "Bridge to Bridge Interoperations" panel. Bridge-to-bridge cross certification will create policy and technology challenges, however the panel concluded that these challenges are not insurmountable.

The 150 attendees represented a cross-section of the global PKI community, with presenters from the USA, United Kingdom, Britain, Israel, Australia, Norway, Sweden, Germany and Canada. Due to the success of this event, a sixth workshop is planned for Spring 2007.

William T. Polk and Nelson E. Hastings
National Institute of Standards and Technology
Gaithersburg, MD USA

2006 PKI R&D Workshop
Making PKI Easy to Use

Gaithersburg, Maryland USA

April 4-6, 2006

http://middleware.internet2.edu/pki06/

(Pre-proceedings were distributed at the workshop)

WORKSHOP SUMMARY 1

Provided by Ben Chinowsky, Internet2

REFERRED PAPERS

How Trust Had a Hole Blown In It. The Case of X.509 Name Constraints 13

David Chadwick — *University of Kent, England*

Navigating Revocation through Eternal Loops and Land Mines 31

Santosh Chokhani — *Orion Security Solutions, Inc.*
Carl Wallace — *Orion Security Solutions, Inc.*

Simplifying Public Key Credential Management through Online Certificate Authorities and PAM 46

Stephen Chan — *NERSC/Lawrence Berkeley National Lab*
Matthew Andrews — *NERSC/Lawrence Berkeley National Lab*

Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy 54

Tom Barton — *University of Chicago*
Jim Basney — *NCSA/University of Illinois*
Tim Freeman — *University of Chicago*
Tom Scavo — *NCSA/University of Illinois*
Frank Siebenlist — *University of Chicago & MCSD, Argonne National Lab*
Von Welch — *NCSA/University of Illinois*
Rachana Ananthakrishnan — *MCSD/Argonne National Lab*
Bill Baker — *NCSA/University of Illinois*
Monte Goode — *Lawrence Berkeley National Lab*
Kate Keahey — *University of Chicago & MCSD/Argonne National Lab*

PKI Interoperability by an Independent, Trusted Validation Authority 68

Jon Ølnes — *DNV Research; Norway*

Achieving Email Security Usability 79

Phillip Hallam-Baker — *VeriSign Inc.*

CAUDIT PKI Federation - A Higher Education Sector Wide Approach **92**

Rodney McDuff — *The University of Queensland*
Viviani Paz — *Australian Computer Emergency Response Team*

LIST OF ACRONYMS **105**

Organizers

General Chair: Ken Klingenstein, University of Colorado
Program Chair: Kent Seamons, Brigham Young University
Steering Committee Chair: Neal McBurnett, Internet2
Local Arrangements Chair: Nelson Hastings, NIST
Scribe: Ben Chinowsky, Internet2

Program Committee

Kent Seamons, *Brigham Young Univ.* (chair)
Peter Alterman, *National Institutes of Health*
Stefan Brands*, Credentica and McGill Univ.*
Bill Burr, *NIST*
David Chadwick*, University of Kent*
Yassir Elley, *Forum Systems*
Carl Ellison, *Microsoft*
Stephen Farrell, *Trinity College Dublin*
Richard Guida, *Johnson & Johnson*
Jason Holt, *Brigham Young Univ.*
Russ Housley, *Vigil Security, LLC*
Ken Klingenstein, *Internet2*
Neal McBurnett, *Internet2*
Clifford Neuman, *USC-ISI*
Eric Norman, *University of Wisconsin*
Tim Polk, *NIST*
Ravi Sandhu, *GMU and TriCipher*
Krishna Sankar, *Cisco Systems*
Frank Siebenlist, *Argonne Nat'l Laboratory*
Sean Smith, *Dartmouth College*
Von Welch, *NCSA*
Stephen Whitlock*, Boeing*
Michael Wiener, *Cryptographic Clarity*
William Winsborough, *Univ. of Texas at San Antonio*

Archival Sites

PKI 2006:	http://middleware.internet2.edu/pki06
PKI 2005:	http://middleware.internet2.edu/pki05
PKI 2004:	http://middleware.internet2.edu/pki04
PKI 2003:	http://middleware.internet2.edu/pki03
PKI 2002:	http://www.cs.dartmouth.edu/~pki02

This page has been left intentionally blank.

5th Annual PKI R&D Workshop Summary

Ben Chinowsky, *Internet2*

Note: this summary is organized topically rather than chronologically. See http://middleware.internet2.edu/pki06/proceedings/ for the workshop program, with links to papers and presentations.

The workshop addressed its theme of "making PKI easy to use" from three angles: how much to expect from the user, and how to design accordingly; PKI and the DNS (DKIM and DNSSEC in particular); and deployment experiences. There were also some additional talks not directly related to the workshop theme.

What's reasonable to expect of the users? How to design around what it's not reasonable to expect of them?

Angela Sasse keynoted with a talk titled **Has Johnny Learnt To Encrypt By Now?** The short answer is "no", for reasons that haven't changed since Alma Whitten posed the question at PKI03: security is complex and unlike anything else users have to deal with, and people aren't properly motivated to use it. Much of Sasse's talk counterposed her approach to solving these problems to Whitten's. The overarching difference in approach to solution is Sasse's skepticism that users can learn all they'd need to in order for Whitten's approach to be successful. Sasse cited Eric Norman's "Top 10" (actually more than that) list of things that users would need to learn to use a typical PKI implementation. Whitten's own research suggests users would need a day and a half of training to get started; for many organizations this is too long.

Sasse's approach to these problems overlaps with Whitten's, but with marked differences of emphasis. Sasse favors:

- designing a "socio-technical system", not just a user interface. In particular, Sasse advocates "design to secure things people care about", citing Felten & Friedman's work on "value-sensitive" design.
- more emphasis on simplifying systems, and less emphasis on teaching users to understand complex systems.
- automating security, rather than keeping it visible.

One example of this approach is to find better names for things. Sasse laid great stress on the need to find better words for the concepts users will still need to learn; for example, the meanings of "key", "public", and "private" in PKI are completely different from their meanings in everyday life. Sasse also cited Garfinkel & Miller's work on Key Continuity Management, which makes heavy use of colorcoding (see

http://groups.csail.mit.edu/uid/projects/secure-email/), and approvingly cited Bruce Schneier's work for its focus on "business and social constraints".

In the discussion following this session, the group greatly extended the analogy between driving and computer security that Eric Norman had used to introduce the "Top 10" list cited by Sasse. Is requiring users to understand the basic concepts of public key cryptography more like requiring them to know how the engine works (avoidable and bad) or more like requiring them to know the rules of the road (unavoidable and good)? Sasse suggested propounding "simple but strong" rules, like "never externalize your password in any way". She also suggested that Whitten's "safe staging" idea has some promise. Sasse strongly advocates risk analysis, in particular to see where security measures shift risks. For example, similarly to the way that car alarms lead to carjackings (instead of being able to hot-wire the vehicle, the attacker now needs to get the keys), biometrics have led to attackers chopping off fingers. Sasse also agreed with David Wasley's comment that the user needs to know at least a little in order to cope when things go wrong — like the driver knowing what the symptoms of underinflated tires are.

Usability Panel Discussions

There were two usability panels, one on digital signatures and the other on browsers. In the **digital signatures panel,** Ron DiNapoli asked if the Kerberos KClient common interface could serve as a model. He argued that a unified interface makes things much simpler, and from this standpoint gave an optimistic assessment of PDF signing and encryption support. Anders Rundgren discussed **webform signing,** which is already used by millions in Europe, largely for citizen-to-government transactions. However, the systems used are proprietary and non-interoperable, so Rundgren is launching the WASP (Web Activated Signature Protocol) standards proposal in cooperation with five groups in Europe. The WASP use cases all stem from efforts to increase usage of e-government. Sandhu discussed prospects for **transaction signatures,** as vs. document signatures — addressing the many potential applications in which there are many transactions requiring only a modest level of assurance, instead of a few transactions requiring high assurance. One key difference is that where document signatures are generally human-verified, transaction signatures are verified by a computer, "with possibly human audit and recourse forensics". Both Rundgren and Sandhu noted the Outlook Express "Security Warning" black screen as a particularly egregious example of how not to design a user interface for email security.

In the discussion, Rich Guida stressed the importance of asking "Is it better than the way we do it now?" Guida suggested that even with their imperfections, any of the signing mechanisms presented in the panel would be better than paper-based signature processes like signing every line of a form. Guida noted that SAFE (http://www.safe-biopharma.org) is working on a universal signing interface. One of the project contractors has developed an approach to verifying historical digital signatures, based on retrieving historical CRLs. This sparked controversy about record-retention issues more generally. David Chadwick argued that efforts to develop trusted timestamping standards for verifying digital signatures are "a complete waste of time", with the exception of one-party signing situations, like a will. Otherwise, the two parties can always put time fields in the signed documents, and the recipient can use this information as part of the process of deciding if the signature is good. Chadwick said that to expect a relying party to trust you to (for example) pay an invoice for goods received, but not trust you to be able to tell the time correctly, seems like a rather strange trust model. Peter Hesse noted signing of lab notebooks to back patent claims as another example of one-party signing. Sandhu argued that record retention will clearly not be a killer app for digital signatures, and expressed surprise that it had dominated the discussion; he stressed the need to look at the application requirements and let that drive the discussion. Hesse brought this back around to "is it better than paper?", which can't prove when it was signed and doesn't need to; he also suggested that "are we overengineering?" is a valid question here.

Amir Herzberg, Frank Hecker, Sean Smith, George Staikos, and Kelvin Yiu gave a joint presentation on **browser security user interfaces,** moderated by Jason Holt. Particularly noteworthy in their slides was a good assortment of bad examples. Holt noted that a common element of these is that the user doesn't know what they need to know in order to quantify the risk involved. Herzberg made two suggestions for improvement: a mechanism that would let you choose a certificate validation service that you trust, like you choose antivirus software; and "public-protest-period certificates", for which the certificate request would be published for a time before the certificate is issued, in order to give the targets of misleading certificate requests an opportunity to object. Herzberg also argued that security indicators should always go in the graphical elements of the browser itself (the browser "chrome"), not in the page content.

The discussion centered around the need for browser and web site designers to get guidance on how to handle the naive user. Holt noted that there doesn't seem to be any documentation of best practices for secure web site developers, and suggested that the PKI community might be well suited to produce such documentation. Hecker noted that the Mozilla Foundation may have grant funds available for the development of best practices documents.

Sean Smith noted a recent paper titled "Why Phishing Works"; see http://people.deas.harvard.edu/~rachna/. Herzberg suggested that the long-term solution for the naive user will be a "secure browsing mode". James Fisher suggested that developers need guidelines for naive users similar to those developed for sight-impaired users; David Wasley suggested "a UL Labs for software," offering certification that user interfaces are no more complex than necessary. Sean Geddis argued that security should be built into the operating system, and the applications should be forced to acquire the appropriate credentials. There was general agreement that while this is true in principle, the amount of cooperation it requires from application developers is not forthcoming, so it's not going to happen. There was also a short demonstration of the security user interface in Internet Explorer 7, which uses red-yellow-green colorcoding. Holt summed up the discussion by stressing the need to compile best practices to guide development of secure browsers and web sites.

Easy-to-Use Deployment Architectures

Stephen Chan described work at NERSC on **Simplifying Credential Management through Online Certificate Authorities and PAM.** The paper and presentation include a useful list of PKI "de-motivators" and the ways in which they are addressed by using short-lived certificates and having users authenticate with PAM (Pluggable Authentication Modules). Chan noted that most of the code from this project is freely available upon request.

Von Welch provided an overview of **the Globus Toolkit, Shibboleth, GridShib, and MyProxy.** The Globus Toolkit (http://www.globus.org/toolkit/) is Globus' core Grid software; Shibboleth (http://shibboleth.internet2.edu) is the Internet2 Middleware Initiative's flagship federating software. GridShib (http://gridshib.globus.org) adds Globus Toolkit and Shibboleth plugins to enable Shibboleth Identity Provider data to be used for Grid access control decisions. MyProxy (http://grid.ncsa.uiuc.edu/myproxy/) is a credential repository and CA that greatly reduces the pain involved in acquiring credentials to run Grid jobs. Work on integrating GridShib and MyProxy is ongoing.

Jon Olnes discussed **PKI Interoperability by an Independent, Trusted Validation Authority.** This approach aims to lessen the complexity faced by relying parties. A Validation Authority (VA) is "an independent trust anchor" — CAs do not delegate trust to a VA; rather the VA offers validation services directly to the relying parties. Olnes's employer, DNV, describes itself as "a leading international provider of services for managing risk", among other things certifying the seaworthiness of ships and the management processes of corporations. Offering VA services is how DNV plans to expand this role into the area of "digital value chains". The idea of a VA was well received by the

group; one attendee described it as "perhaps the most important solution the PKI community has been missing". A deployment is planned for this summer.

PKI and the DNS

IETF DKIM Working Group co-chair Barry Leiba moderated a **panel discussion on Domain Keys Identified Mail (DKIM).** After asking for a show of hands that revealed that few in the room were familiar with the technology, Jim Fenton gave an **Introduction to DKIM.** DKIM is a way for an email domain to take responsibility for sending an email message. The central goal of DKIM is to stop email spoofing; its central concepts are 1) key distribution via DNS ("a useful pseudo-PKI for DKIM"), 2) using raw keys, with 3) signatures representing the domain, not the author. Tim Polk discussed **DKIM Seen Through a PKIX-Focused Lens;** he noted that "DNS poisoning is not that difficult, it just isn't that interesting in most cases. DKIM makes it interesting." Nonetheless, Polk argued that from a spam-mitigation standpoint DKIM is much better than nothing, and that the incentive it provides to attack the DNS may in turn drive DNSSEC deployment. Polk also noted that DKIM is extensible to other key-fetching services, and suggested that these services include one based on X.509.

In the discussion, there was strong approval of the concept of DKIM as a good foundation to build on, rather than a complete solution. Leiba noted that DKIM is good for whitelisting, not blacklisting. Neal McBurnett suggested that the semantics of a DKIM signature are basically "I [the domain] am willing to be punished if this is bad"; Leiba said that it's more like "I acknowledge that I put this on the Internet". Different signers will have different interpretations of exactly what that means; some people want more clarity in the interpretation, and that complicates things. Phillip Hallam-Baker expects the DKIM standard to provide a flag to say "all messages from this domain should be signed"; in his view, giving potential signers confidence that signing will make a message more likely to get through — in particular that it will be less likely to get flagged as spam — will be key to DKIM uptake. Also, in response to questions from Chadwick, Hallam-Baker agreed that DKIM is just as susceptible to bad client design as S/MIME, and relies just as strongly as any PKI on CAs not permitting lookalike domains. There was strong general agreement that widespread DKIM deployment would mean that a lot more would be riding on the success or failure of attempts to secure the DNS. More on DKIM is at http://mipassoc.org/dkim/.

Noting the need to raise our sights from the goal of mere "usability", Phillip Hallam-Baker offered an approach to **Achieving Email Security Luxury,** relying centrally on DKIM. Hallam-Baker wants to have a security interface as compelling as a video game — if we aim high, maybe we'll hit higher than we would by aiming lower. First among his requirements is to avoid the assumption that users want to become computer experts. Some development of expertise among the users will nonetheless be needed; here Hallam-Baker stressed the importance of providing education ("empowerment"), and not just training ("mere instruction"). Hallam-Baker's software solution relies centrally on the power of branding. This solution uses DKIM and the PKIX LogoType extension to implement "Secure Internet Letterhead" — verified mail will display the logo of the sender and (upon request) the logo of the verifier, in the "chrome" of the email client. The use of DNS to distribute keys improves the chances of rapid deployment. Other than DKIM, all components of this solution have been standardized; DKIM is currently being standardized in IETF (see http://www.ietf.org/html.charters/dkim-charter.html). A prominent theme in Hallam-Baker's talk (as well as Welch's and Chan's Grid presentations) was that most of the things we need to architect an easy-to-use PKI are already available — it's largely a matter of putting existing components together in new ways.

Allison Mankin presented an update on **Trust Infrastructure and DNSSEC Deployment.** Attacks on the DNS are usually not well publicized; http://www.dnssec-deployment.org has details on recent attacks. Mankin noted that the major costs of DNSSEC deployment are in training, operation, and key management, not computing and network resources. More cost-benefit analysis is needed. Operating system, firewall, and application support for DNSSEC still needs work, and an extension to prevent zone-walking is still in development, but Mankin strongly advocates deploying pieces as soon as they're ready. She was seconded in this view by Hallam-Baker, who pointed out that SSL — the only implementation of public-key cryptography to deploy widely — had serious flaws when deployment first got under way.

Deployments

In his opening remarks for the workshop, Ken Klingenstein observed that the PKI community is currently engaged in working from the bottom up, building "pockets" of functioning infrastructure. One new pocket is the **CAUDIT PKI** for higher education in Australia; Viviani Paz provided an overview. Four levels of assurance are offered, depending on the strength of the proofs of identity provided by a prospective certificate holder. Of particular note is the points system the CAUDIT PKI uses for identity proofing (e.g., a passport is worth 70 points, a driver's license only 40 points); this system is based on the

laws governing financial transaction reporting in Australia. CAUDIT is taking a phased approach to deployment; the pilot phase has concluded and the pre-production phase is underway.

One of the largest existing pockets of deployment is the **US Federal PKI.** Peter Alterman gave an update and moderated a panel on developments in this area. Thirteen Federal entities are currently cross-certified; further information is available at http://www.cio.gov/fpkipa/. David Cooper discussed developments in the **Path Discovery and Verification Working Group** of the FBCA (see http://www.cio.gov/fbca/pdvalwg.htm). A path discovery test suite is under development. Judy Spencer explored **The Role of Federal PKI in compliance with Homeland Security Presidential Directive 12.** HSPD-12 is titled "Policy for a Common Identification Standard for Federal Employees and Contractors". PKI and smartcards are central to the implementation, as are new processes for personal identity verification; one major change will be requiring government contractors to pass the same background checks as government employees. See http://csrc.nist.gov/piv-project/ and http://www.cio.gov/ficc/.

There were also reports on steady though incremental progress in building corridors among these and other pockets. Alterman moderated a **panel on Bridge-to-Bridge Interoperability;** he observed that cross-certification among bridges has the potential to greatly expand the reach of PKI. Debb Blanchard provided an overview of the **Bridge-to-Bridge Working Group.** The BBWG was launched to address issues around the FBCA cross-certifying with other bridges such as HEBCA, but has since broadened its scope to BCAs more generally. A fundamental principle for the BBWG is that no transitive trust is allowed across bridges. This point was also stressed by Santosh Chokhani in his talk on **Technical Considerations for Bridge-to-Bridge Interoperability:** trust is bilateral like business relationships; it cannot be transitive across bridges. Finally, Scott Rea updated the group on PKI in higher education and progress toward HEBCA deployment. The key uses he sees for PKI in higher education are S/MIME, paperless workflow, Shibboleth, federated Grid, and e-grants. Because higher education gets so much federal funding, FBCA is the primary target for HEBCA cross-certification. A prototype is operational, and from a purely technical standpoint, HEBCA has been ready to launch for several months; watch http://www.educause.edu/hebca/.

Snags in the standards process can prevent us from getting as far as we might have in building and interconnecting pockets of PKI. David Chadwick explored **How Trust Had a Hole Blown In It: The Case of X.509 Name Constraints.** For ten years ISO/ITU-T and IETF PKIX have failed to bring their interpretations of name constraints into alignment. Chadwick argued

that imprecision in the base standard led to misunderstanding of the original intentions behind name constraints, and that both sides have been slow to rectify these misunderstandings. His talk was followed by a spirited discussion which included several of the individuals involved in the history recounted by Chadwick, disagreeing with his account of that history, the current seriousness of the problem, and the best way to fix it.

Other topics

Bill Burr presented a comprehensive **NIST Cryptographic Standards Status Report.** NIST's current focus is getting Federal users off of 80-bit equivalent cryptography (e.g. 1024-bit RSA & DSA) by 2010. There are complex patent issues with elliptic-curve cryptography (ECC); Burr was asked whether ECC provides enough performance improvement at real-world keylengths to make it worth the uncertainty around patents. Burr responded that as a part of the Department of Commerce, which also includes the Patent and Trademark Office, NIST cannot discriminate against technologies based on patent status; he also expects Windows Vista to make ECC more widely available. Burr said that he is now 98% sure that there will be a NIST competition for a replacement for SHA.

Jeffrey Altman gave an overview of the state of the art in **Integrating PKI and Kerberos.** PK-INIT, a means of using a certificate to get a Kerberos ticket, is the most well-established project, but there are also PK-APP (KX.509 — using Kerberos to get a cert) and PK-CROSS (using certs for inter-domain Kerberos). Altman recommends that deployment efforts focus on reducing the number of credentials that users have to worry about.

There were two presentations on revocation. Santosh Chokhani presented Marine Corps-funded work on **Navigating Revocation through Eternal Loops.** Chokhani presented various options for dealing with the problem of the circular dependencies in revocation that can be created by self-issued certificates. Chokhani noted that he's not advocating any of these options over the others, rather saying "if you pick your poison, here's your antidote."

Kelvin Yiu, lead Program Manager for Microsoft Windows security, discussed **Enabling Revocation for Billions of Consumers,** with a focus on revocation in Windows Vista. Internet Explorer 7 in Vista will enable revocation checking by default. Yiu explored various lessons learned and tradeoffs between usability and getting the large downloads required. Yiu's slides include a list of best-practice recommendations to the industry, headed by "Use HTTP, not LDAP".

There were four short work-in-progress presentations.

- Sam Sun presented **Experiences Securing DNS through the Handle System.** Plans for the software include an open-source release, deployment in the .cn TLD registry, and using it to support ENUM service.
- Michael Helm presented an overview of the **International Grid Trust Federation.** The IGTF is composed of three Policy Management Authorities covering the Americas, Europe, and the Asia/Pacific region; see http://www.gridpma.org. Helm noted the January 2006 launch of the European Commission's E-Infrastructure Shared Between Europe and Latin America (EELA) project to support Grid development in Latin America. The Large Hadron Collider (LHC) is a major driver for the IGTF.
- Doug Olson discussed **PKI in the Open Science Grid.** OSG (http://www.opensciencegrid.org) is also heavily focused on the LHC, as well as virtual-organization support. OSG uses the NSF Middleware Initiative distribution as its core software. Both Helm and Olson cited making PKI more usable for less technical users as a major issue.
- Robert Relyea and Kelvin Yiu presented **Suite B Enablement in TLS: A Report on Interoperability Testing Between Sun, RedHat, and Microsoft.** Suite B is an NSA standard for elliptic-curve cryptography (ECC); see http://www.nsa.gov/ia/industry/crypto_suite_b.cfm. Bill Burr noted that while NIST is not mandating ECC, they are advocating it. Burr also remarked that if you want to use ECC anywhere, you want to use it on smartcards.

The WIP session concluded with a "rump session" in which presenters were given three minutes each for impromptu presentations. Ron DiNapoli explained the motivation for, and gave a very short demonstration of, his work on **Integrating PKCS-11 with Apple Keychain Services.** Chris Masone, a student of Sean Smith, set out the early stages of his work on **Attribute Based, Usefully Secure Email (ABUSE),** using shortlived credentials. Anders Rundgren outlined his work on **WS-Mobile,** a scheme for using cell phones to replace smartcards. Finally, David Cooper of NIST posed the question, **Are Offline Root CAs worth it?** — not offering an answer but providing a useful rundown on the pros and cons.

Conclusion

PKI06 further solidified the consensus from PKI04 and PKI05: "Understanding and educating users is centrally important" and "The specifics of any particular PKI deployment should be driven by real needs, and should be only as heavyweight as necessary." PKI06 also filled out this consensus with further examples and experiences. With respect to experiences, there was strong interest in expanding the work-in-progress and rump-session components of future workshops. There was also increased interest in documenting best practices for industry to use in implementing the PKI0x consensus.

PKI06 was well attended, setting an all-time attendance record for the workshop series. Program Committee Chair Kent Seamons pointed out that although the number of technical paper submissions was quite low this year, the peer review process was rigorous and the acceptance rate was comparable to that in previous years. As had been recommended by attendees at previous workshops, this year's program had many more invited talks and panel discussions; this change was well received at PKI06. The organizers will make a concerted effort to increase the number of technical paper submissions in the future.

PKI07 will focus on applications. Please join us at NIST, April 17-19, 2007.

REFEREED PAPERS

How Trust Had a Hole Blown In It

The Case of X.509 Name Constraints

David Chadwick, University of Kent. d.w.chadwick@kent.ac.uk

Abstract

A different interpretation of the Name Constraints extension to that intended by ISO/ITU-T in its 1997 edition of X.509, was made by the IETF PKIX group in its certificate profile (RFC 2459). This has led to conflicting implementations and misalignment of the standard and its profile. This paper reviews the history of the Name Constraints extension, and how it has evolved to the present day from an original concept first described in Privacy Enhanced Mail. The paper concludes by suggesting possible ways forward to resolve this unfortunate conflict.

1. Introduction

The name constraints extension in X.509 was first introduced in the 1997 edition of X.509 [2]. But its history goes back further than that, back in fact to the early 1990's and Privacy Enhanced Mail (PEM) [1]. The extension has evolved over time since its first introduction, and, due to lack of precision in the original X.509 definition, varying interpretations of its meaning have evolved. This has now led to a divergence between the Internet PKIX group's profile of X.509 [3] and the latest edition of the X.509 standard [4, 8], which is about to be merged and published as X.509 (2005). This matters, because some certificates accepted as valid by one interpretation, will be treated as invalid by the other, and vice versa.

This paper tries to untangle the confusion surrounding the name constraints extension, and understand how we have got into the situation we are in today, where the X.509 standard and the RFC 3280 profile [5] disagree about both the syntax and the semantics of this extension. This paper then poses the question, "Where do we go from here?" This is still an unanswered question, but some possibilities are suggested in the final section of this paper. This will no doubt provoke some further discussion of the problem both within the standards settings groups and with implementers, and this might help to draw this misalignment to a successful conclusion.

This paper has been written mostly from the documents (standards and draft standards) published during the last 12 years, but also partly from the memories of those working in this area at the time [9]. It therefore could contain errors in the interpretation of what was actually published. However it is a best efforts attempt at trying to understand how the current problem has arisen. It also provides an interesting historical case study of the standardisation process which shows how original intentions evolve with time, but due to imprecise specifications, and a lack of dialogue, different conclusions about these intentions are reached by different groups of people. The contents of this paper are as follows. Section 2 describes a motivating example to show how and when name constraints can be useful. Subsequent sections refer to this to show

how it can (or cannot) be supported with the various flavours of name constraints as it has evolved with time. Section 3 provides a history of the early developments of the name constraints extension, up until 2000. Section 4 provides a more recent history of the extension, from 2001 to the current date. Section 5 then concludes and suggests answers to the question "Where do we go from here?" This might help to guide subsequent discussions on this topic.

2. A Motivating Example (or two)

Suppose organisations X, Y and Z all operate CAs, with respective DNs: {cn=CA, o=X, c=GB}, {ou=admin, o=Y} and {o=Z, c=US}. Assume each CA issues certificates to its employees, who all have DNs under their respective organisational arcs {o=X, c=GB}. {o=Y} and {o=Z, c=US}. Some of the CAs may also issue certificates to other people, e.g. contractors, subsidiaries, business partners etc. We assume that these are named under different arcs to those of their employees.

Scenario 1. Suppose that any two of these three organisations wish to cross certify each other, and constrain the certificates they wish to trust to only those issued to their employees. This is easily achieved by placing a name constraints extension in each cross certificate issued to X, Y or Z indicating that only certificates starting with a DN of {o=X, c=GB}, {o=Y} or {o=Z, c=US} respectively will be trusted. Any other certificates issued to contractors, business partners etc. will not be trusted, provided their DNs are not in the employee's name space.

Scenario 2. Suppose one of the organisations only wishes to trust a subset of the certificates issued to the employees of another of the CAs, for example, to employees within the marketing department. This can be achieved by using a name constraints DN of {OU=marketing, o=X, c=GB}, {OU=marketing, o=Y} or {OU=marketing, o=Z, c=US} respectively.

Scenario 3. Suppose a Bridge (or some other) CA exists that has cross certified all of the CAs in North America, including CA_Y and CA_Z, and a European Bridge CA exists that has cross certified all the CAs in Europe, including CA_X, and that these two bridge CAs have cross certified each other. Now each of the three organisational CAs will trust all the certificates issued by all of the CAs cross certified with the bridges. Suppose however that one of these organisational CAs wants to limit the certificates that are deemed to be trustworthy via the Bridge CAs e.g. CA_X only wants to trust certificates issued by CA_Y to its employees and not any certificates issued by CA_Z. In this case, CA_X issues a cross certificate to the European Bridge CA that has a name constraints of {o=Y}, with a parameter to indicate that the first certificate in the chain (that of the North American Bridge CA) is not to be bound by the name constraints rule[1].

[1] Santosh Chokhani has pointed out that this has a potential weakness since the relying party could take another path and trust some other cross certified CA (one CA removed) to issue CA_Y's certificate, since it is not specified which CA is to be skipped. The simplest way around this is to add a second name constraint containing the name of the North American Bridge CA, so that another CA may not be substituted for it.

3. An Early History of Name Constraints (93-2000)

"Name constraints" was originally introduced as a concept to limit the X.509 certificates that could be issued to support Privacy Enhancements for Internet Electronic Mail (PEM). As RFC 1422 states below, the rationale was to try to ensure that each CA only issued certificates containing globally unique distinguished names, since this was a fundamental requirement of the X.500 standard, of which X.509 was an integral part.

RFC 1422 [1] states:
To complete the strategy for ensuring uniqueness of DNs, there is a DN subordination requirement levied on CAs. In general, CAs are expected to sign certificates only if the subject DN in the certificate is subordinate to the issuer (CA) DN. This ensures that certificates issued by a CA are syntactically constrained to refer to subordinate entities in the X.500 directory information tree (DIT), and this further limits the possibility of duplicate DN registration.

There was much debate during this period about how globally unique distinguished names could be formed. Questions included: who would be the global naming authorities; who would manage the root of the Directory Information Tree (DIT); and what would be the contents of distinguished names, in terms of the allowed attribute types and values? There were no conclusive answers to this debate when the PEM RFCs were published, and so PEM neatly sidestepped this issue for user certificates, by saying that they would be named subordinate to the names of the CAs, assuming that each CA would have a globally unique name. This mindset tended to continue in the PKIX working group in subsequent years, and still continues in some quarters today, where some experts believe that a subject DN can only be regarded as globally unique if it is assumed to be subordinate to, or used in conjunction with, the name of the issuing CA. This name subordination was never an assumption of X.500, which instead, required that each user DN would be globally unique in its own right.

At the time the PEM standard was being released, in 1993, the second edition of X.509 was also being released. Unfortunately X.509(93) did not contain any technical mechanism to indicate any sort of constraints on the subject names that a CA could place in the V2 certificates that it issued. A CA could issue a certificate with any valid subject DN. Thus the PEM standard had to ensure this constraint on subject names through procedural means that were placed on the CA (by the above wording in the PEM standard) and by a technical requirement to check name subordination during certificate path validation. Whilst these mechanisms are sufficient to enforce name subordination, they are very inflexible, since they can only cater for Scenario 1 above (and not for 2 and 3) since there is no information in the X.509 certificate to indicate how and when name subordination rules should be applied (or not). Consequently, as soon as X.509 (93) was released, work started on defining the policy rules that could be placed inside certificates, in order to allow much more flexibility in determining which certificates should be trusted. This work culminated in edition three of X.509, published in 1997 [2]. The primary work on edition three of X.509 was the technical definition of the

protocol elements inside certificates that would support the policies and procedures of a CA. This was achieved by adding extensions to the X.509 V2 certificate format, to produce the V3 certificate format that we all use today. (Since V3 certificates are infinitely extensible there has never been a requirement since 1997 to define a V4 certificate format.)

During the four years that it took to produce the 1997 edition of X.509, several working drafts were produced. The name constraints extension was there from the outset, and its syntax and semantics remained constant until 1996. Annex 1 shows the name constraints definition in the output produced by the Orlando meeting in December 1994 [6] and the Ottawa meeting in 1995 [7]. The only difference, shown by the underlined text, was some more explanation of the meanings of the various fields added in 1995. One can see that a primary requirement was to satisfy PEM's concerns to constrain which names a CA could issue to its subjects, but also to add greater flexibility in order to cater for all three scenarios described above (and more!). There are three notable features of this definition.

- Firstly, the only name form that was supported was the X.500 distinguished name (DN) and the way that a name space was constrained was via the subtreeSpecification directly imported from the X.501(93) standard. The subtreeSpecification allows ***any*** arbitrary DIT subtree to be defined, including chopped subtrees which define branches of the top level subtree that are to be chopped off. (Note that X.501 allows filtered (disjoint) subtrees as well, but X.509 stated that filtered subtrees should not be permitted in name constraints). The subtreeSpecification allows us to easily cater for Scenarios 1 and 2 above.

- Secondly, there were no loopholes. Any user certificate that did not fall within the scope of a specified name constraint, should not to be regarded as valid. The semantics of the extension could therefore be stated as "**every name that is not explicitly trusted is untrusted**" i.e. the name constraint specifies a white list of trusted subtrees. Since all constrained names were based on distinguished names, there was no possibility that a constrained certificate could contain other than a name in X.500 DN format. This feature ensured that certificates issued to sub-contractors, business partners etc. who had different DNs would not be trusted inadvertently.

- Thirdly, not all certificates issued by subordinate CAs need be constrained. Two control mechanisms were provided for the certifying CA to specify which certificates did not fall within the scope of the name constraints extension. The certifying CA could either specify a set of certificate policies to which this constraint applied, or could specify how many CAs in the chain should be skipped before the constraint applied. This skipping mechanism allows us to cater for Scenario 3 above.

The net result of this extension was that the issuing (superior) CA could tightly control which (subject names in) certificates issued by cross certified (subordinate) CAs should be trusted. Any relying party (RP) using the superior CA

as its root of trust could be sure that certificate path validation software would not trust any certificate falling outside these name constraints. We thus had a watertight trust model.

Another extension was also being defined during this period, entitled the subject alternative name field. This extension defined "*one or more alternative names, using any of a variety of name forms, for the entity that is bound by the CA to the certified public key*". Several possible alternative name forms for the certificate subject were specified, including a DNS name, an RFC822 email address and an X.400 OR address. This extension underwent some growth during this period, starting out with just four alternative name forms and eventually ending up with nine. Its intention was to allow a certificate subject to have a variety of names in different formats, because it was recognized in the mid 1990s that there was not going to be a global X.500 directory service. If the X.509 standard could not cater for subjects with other name forms besides X.500 ones, then this would significantly limit is scope and applicability. Thus X.509 should support alternative name forms. In order to make the extension fully extensible and able to cater for future name forms that currently do not exist, the alternative name can also be an *other* name form, which is identified by a globally unique object identifier. Thus it is likely that a relying party might encounter a subject alternative name form that it is not able to recognize. In order to cater for this, the definition of this extension included the text "*a certificate-using system is permitted to ignore any name with an unrecognized or unsupported name form*". The implicit assumption was however, that this was an alternative name for the subject, not a replacement name, and the subject would always have an X.500 distinguished name, even if it did not have an entry in an X.500 directory service. We shall see later that this ability to ignore unrecognized name forms probably indirectly led to the erosion of the trust model built into name constraints.

Yet another certificate extension that was being defined through this period was the one that eventually became known as the basic constraints extension. This had something of a Jekyll and Hyde life. Initially known in the PDAM [6] as the *CA or end entity indicator*, it had virtually the same syntax and semantics as the basic constraints extension used today. It then grew in significance in the DAM [7], when it changed its name to basic constraints and added a simplified name constraints capability to its syntax, specifically, the ability to specify the set of permitted subtrees in which all subsequent certificate subject names should fall.

Dramatic changes to the X.509 draft standard occurred in April 1996 at the Geneva meeting, precipitated by amongst other things the Canadian national ballot comment. The Canadian ballot comment proposed three things:

- to introduce the syntactic construct GeneralName, in order to group together into one super-type all the name forms in the subject alternative name field
- to add further capability to basic constraints in two ways, firstly by allowing denied subtrees as well as permitted subtrees to be specified; and secondly to replace the X.500 distinguished name type with the GeneralName super-type.

- to remove the name constraints extension since it was no longer needed, as it main purpose was now usurped by the enhanced basic constraints extension being proposed in this ballot comment.

The outcome of the resolution of the Canadian and other national ballot comments is well documented; it is the 1997 edition of X.509 (see Annex 2). Precisely what technical discussions were had in order to get there have now largely been forgotten with time, but several things are clear. The Canadian introduction of the GeneralName super-type was accepted, and this was used to specify the subject alternative name extension. The changes to basic constraints were rejected, and this extension reverted to its original 1994 definition. However, the intention of the ballot comment was accepted in principle, by modifying the name constraints extension to match the proposed basic constraints extension. In other words, name constraints was modified by replacing the X.500 distinguished name type with the GeneralName super-type, deleting the policy and skip certs controls that limited when the name constraints should apply, and adding denied (or excluded) subtrees. The intention of name constraints was still very clear, as stated in the first sentence of the description "*indicates a name space within which all subject names in subsequent certificates in a certification path must be located*". It can be seen that its purpose was to tightly constrain the names that the subordinate or cross certified CA could put into the subject field of the certificates that it issued, and more than that, to constrain all additional subordinate CAs further along the certification path. Whereas the original name constraints allowed certain groups of certificates to be specifically excluded, via the skipCerts and policySet fields, the new definition did not. The semantics were very definitely "**every name that is not explicitly trusted is untrusted, with no exceptions**". In other words, the original trust model still held true, but was even tighter than before, because Scenario 3 can no longer be easily supported (although it can be supported in a more complex way by adding a second permitted subtree containing the name of the North American Bridge CA and an excluded subtree of all names subordinate to this). This tight trust model is further shown by the Certificate Path Processing Procedure in Section 12.4.3 of the 97 standard, which states:

The following checks are applied to a certificate:
…..
e) Check that the subject name is within the name-space given by the value of permitted-subtrees and is not within the name-space given by the value of excluded-subtrees.

If any of the above checks fails, the procedure terminates, returning a failure indication and an appropriate reason code.

Unfortunately, when the GeneralName syntax replaced the X.500 DN syntax in the name constraints extension, it was not as straightforward as simply replacing one syntax with another. The text describing the name constraints extension should have been significantly enhanced, because new possibilities now existed that did not before. Enhancements were needed in a number of ways. Firstly, how was the name constraints extension to

handle general names that were not hierarchically structured, such as IP addresses. How could one specify permitted and excluded subtrees for non-hierarchical names? The answer was to exclude these name forms from being applicable to this extension, as is indicated by the text "*only those name forms that have a well-defined hierarchical structure may be used in these fields*". Secondly, what was a relying party to do if there was a mismatch between the various subject alternative name forms in a certificate, and those in the name constraints extension in the issuing CA's certificate? Furthermore what is the default constraint on a name form that is not included in the name constraints extension? Several new possibilities now exist: **(i)** the subject's alternative names are a subset of the name forms listed in the CA's name constraints; **(ii)** the subject's alternative names are a superset of the name forms listed in the CA's name constraints; **(iii)** the subject's alternative names intersect with the name forms listed in the CA's name constraints; **(iv)** the subject's alternative names do not overlap with the name forms listed in the CA's name constraints; and **(v)** the subject's alternative names are identical to the name forms listed in the CA's name constraints. Unfortunately the standard is strangely quiet on this aspect. This is clearly a bug. The fact that appropriate wording was not included to reflect the change of syntax can be seen from the first sentence of the definition, which continued to state "*indicates a name space within which all subject names in subsequent certificates*". In fact, with the introduction of General Names, it does not indicate a single name space any longer, but possibly many different name spaces. How a relying party should behave when all these new possibilities present themselves can be resolved in one of two ways, either conjunctively or disjunctively. Logical conjunction requires all the name forms in the certificate to match the constraints in the extension, whereas logical disjunction requires just one subject name in the certificate to obey one of the name constraints. When this issue was recently debated on the X.500 mailing list, the X.500 rapporteur stated "*I considered (subject) alt names to be truly alternate forms of the subject name in the certificate. That subject name had to be within the scope of any name constraints, if specified. If the subject name was in scope, the alternative name would be considered within scope. I don't think we, meaning the X.509 group, ever considered what to do for any other conditions*".

As soon as X.509 (97) was published, the IETF PKIX group started to work on their profile for X.509 public key certificates. The first version of this was published in 1999 as RFC 2459 [3]. In an attempt to guide implementers in their coding, it had to work out what the intended X.509 semantics were when there was a mismatch between the name forms in a subject's certificate and those in the name constraints extension of the issuing CA. Therefore RFC 2459 added the following two critical sentences to its specification "*Restrictions apply only when the specified name form is present. If no name of the type is in the certificate, the certificate is acceptable*." Precisely why these sentences were added is not known. It might have been a best efforts interpretation of how the subject alternative names logic (which stated that unknown name forms could be safely ignored) applied to name constraints. On

the other hand it might have been a poor attempt at resolving mismatches between name forms in subject names and name constraints.

Unfortunately, and perhaps without realizing it, the RFC 2459 wording was also flawed in two ways. Firstly it does not explicitly cover all the five cases listed above. Specifically what rule should apply when the certificate simultaneously has no name of the type specified in name constraints but also has a name of the type specified in name constraints (cases (ii) and (iii) above). Should it be trusted or not? But more importantly, it has introduced a potentially massive security hole in the trust relationship between the superior CA issuing the certificate with a name constraints extension and the subordinate (or cross certified) CA receiving it. In fact, it has completely reversed the X.509 trust model into one of "**every name form that is not explicitly untrusted is trusted**" i.e. name constraints now become black lists rather than white lists. For example, referring to Scenario 1 above, where organization X cross certifies organization Y, suppose that unknown to organization X, organization Y's CA is somewhat untrustworthy, or it simply changes its rules, and decides it will issue certificates with other name forms as well as or instead of X.500 DNs, for example RFC822 names. A user, Freddie Fraudster (who may or may not be employed by Y), with the email address nice.guy@cheap.goods.com wants to obtain a certificate that will be trusted by organization X's CA, so it asks organization Y's CA to issue him with a certificate containing only his email address. Using the RFC 2459 semantics of "trust all except", the certificate will be trusted by relying parties who have a root of trust in organization X's CA. However, using the X.509 "untrust all except" semantics, the certificate will not be trusted. This reversal of semantics has now blown an unblockable hole in the trust relationship between the two CAs. The reason is that the number of subject alternative name forms is infinite, through using the *other* name form variant. Since it is impossible to list an infinite number of name forms, it is impossible to list all the name forms that are trusted (according to RFC 2459) or untrusted (according to X.509 (97)). Thus it is much safer for name constraints to contain white lists rather than black lists. As Marcus Raunum states in [10], the dumbest idea of all in computer security is to have a default permit policy, but this is precisely what RFC 2459 has done.

3. A Recent History of Name Constraints(2001-05)

Despite its publication in January 1999, the RFC 2459 trust hole and reversal of the X.509 trust semantics, went largely unnoticed by the X.509 standards body for several years. So much so that the third edition of X.509 was published in 2001 [4] with almost exactly the same wording for the name constraints extension as the 1997 edition. This lack of awareness is perhaps not that unusual, since RFC 2459 was only a profile of X.509, designed to give implementers recommendations on which options of X.509 to implement and which not to. It was not meant to be redefining the logic of X.509, and certainly not reversing it, although it might serve to further explain the intended logic to implementers. Consequently the two critical sentences of RFC 2459 were not added to the X.509 standard. Whilst many companies had implemented the X.509 semantics, including Entrust, some companies had

implemented the RFC 2459 reversed semantics. In essence, the market place was in chaos. An attempt at reconciliation was attempted in late 2001 by the X.509 editor. This entailed a change of syntax and semantics to the X.509 standard, so that it could capture both the "trust all except" (black list) and "untrust all except" (white list) semantics. The expectation (at least in some quarters) was that the proposed update of RFC 2459 would adopt the new X.509 syntax and semantics. The change to X.509 was published in October 2001 as a technical corrigendum [8]. This is shown in Annex 3. The update to RFC 2459 was published in April 2002 as RFC 3280 [5]. Perhaps surprisingly, RFC 3280 contained exactly the same text as RFC 2459 and made no attempt at profiling the revised version of X.509 which had attempted to resolve the conflict.

The important things to note about the revised X.509 (2001) version are,

- a new object identifier was allocated to the revised extension, so that the original name constraints extension was no longer part of the X.509 standard,
- in an attempt to align with the reversed RFC semantics, the original syntax had the new "trust all except" semantics applied to it, whilst the new syntax had the original "untrust all except" semantics applied to it,
- the new syntax added a "required name forms" field, with the semantics that each subsequent certificate in the chain "*must include a subject name of at least one of the required name forms*". Thus was to stop certificates with no names in the constrained name forms from being accepted, as they are with the RFC semantics.
- it still does not easily cater for Scenario 3 without specifically permitting the names of the intermediary CAs, since there is no way of skipping one or more certificates in a certificate chain before the name constraints take effect.

In summary, the various editions of X.509 and their RFC profiles have remained out of synchronisation over name constraints for all of their lifetimes, with the latest version of X.509 (the 2001 corrigendum) and RFC 3280 being out of synchronisation for the last 4 years. The situation has recently been brought to the attention of the X.509 standards community again, through the issuing of defect report 314 by the RFC 3280bis design team. This recommends that X.509 reverts to the original 1997 and 2001 syntax but keeps the new "trust all except" (black list) semantics instead of its original "untrust all except" (white list) semantics, and, in addition, X.509 should define a new certificate extension that will capture the original "untrust all except" (white list) semantics.

4. Conclusions and Way Forward

This is clearly a sorry tale of continually changing syntaxes and semantics, misunderstandings between two standards creating bodies, the IETF and ISO/ITU-T, a lack of communication and perhaps even lethargy at dealing with issues in a timely manner. The obvious question to ask now is "where do we go from here". Clearly there are several possibilities. This paper lists some of them, primarily from a technical perspective without considering the commercial or political implications of any one of them. The other considerations that will also need to

be taken into account when coming to a resolution of the problem, are trust and usability, and how relying parties should behave or adapt when they are presented with either of the trust paradigms "trust all except" and "untrust all except". Different user communities may prefer different trust paradigms.

Some of the different technical possibilities envisaged by the author are:
1. The ITU-T/ISO X.509 group could accede to the RFC 3280bis design team's request, and revert the X.509 name constraints syntax to that of 1997 and 2001, whilst keeping the new "trust all except" (black list) semantics, even though this is really dumb thing to do [10]. A new extension would then need to be defined that encapsulated the original "untrust all except" (white list) semantics, perhaps along with the original exclusion control mechanisms from the 94/95 drafts i.e. of specifying policy sets and certificate path skipping that control which sets of certificates the constraint applies to. In this case the IETF would need to do nothing to its profile. Implementers who conform to the IETF semantics would not need to do anything unless and until the new "white list" extension is defined and they decide to add it to their implementations.

2. The ITU-T/ISO X.509 group could revert to the 1997 and 2001 syntax and original "untrust all except" (white list) semantics, and add additional clarifying text to make it clear that unspecified name forms are fully constrained (i.e. untrusted) and that logical conjunction is used to evaluate all the subject names. This would be in the spirit of the original extension, although it would not allow certificates that contained additional enterprise specific names (for internal domain use only) to be used in an external cross certified domain. Further, it would not cater for those implementations that support the IETF semantics, in which case the IETF would need to take this change of semantics into account when revising RFC 3280 e.g. by deleting the two critical sentences that they added in RFC 2459. ISO/ITU-T should then consider enhancing the extension, or creating a new one, so that it can cater for enterprise specific names.

3. A more dramatic solution might be to add an optional parameter (e.g. integer) to the 1997 syntax with the semantics "don't check (n) CA certificates", in order to easily cater for Scenario 3. This would be similar to the skipCerts integer that was present in the 94/95 draft standard. Part of the rationale given to the author for the current RFC semantics, is so that end entities and CAs can have different name forms, and then only the end entity name forms are constrained by the name constraints. The addition of a specific parameter which indicates that this is what is required, is semantically better than the current RFC method of reversing the trust semantics to "trust all except" which is too loose in its control capability, and open to abuse. However, this is not recommended, since as indicated earlier the same effect as skipping can be achieved by specifically permitting the names of the skipped CAs.

4. Either of the standards bodies could create a completely new certificate extension with a more sophisticated ASN.1 data type that could precisely specify which names are to be trusted and which are not, and when in the certificate chain the constraint should comes into effect. For example, the extension could contain a sequence of permitted,

excluded, and required name forms and their name spaces. This is the clean sheet approach of taking the requirements and starting from scratch.

5. Finally, the resolution could simply be to do nothing to the latest X.509 syntax and semantics, since this allows both "trust all except" and "untrust all except" semantics to be specified. The IETF PKIX group can then decide to either profile the original X.509 syntax, as they currently do, and keep their existing syntax and semantics, or migrate to profiling the latest version of X.509. Since the IETF has been out of synchronisation with the X.509 name constraints extension ever since their first RFC was published in 1999, being out of synchronisation for another few years should not pose any significant problems to them or to implementers. However, their current approach to solving Scenario 3 type use cases is less than optimal.

In summary, what lessons have we learnt from this development? Clearly writing IT standards is hard, and perhaps writing security standards is even harder. Even though the editors try hard to remove ambiguities and incomplete specifications from standards, nevertheless they still exist. Standards have bugs in them just like software, and just like software, you don't know what bugs are there until someone finds them. Making very significant changes to the final draft version of a standard is not a good idea since there is insufficient time to find any bugs that might have crept in. Cross fertilisation of experts between base standards writers and profile writers will clearly help identify poor specifications, but this is not always practical given the constituencies of the two communities. Finally, given that we are human, errors will always occur. The real test of human ingenuity and adaptability is not that we never generate errors, but rather that we can resolve them effectively when they do occur. Sadly in this case we appear to have failed the test so far.

References

[1] S.Kent. "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management. RFC 1422. February 1993.

[2] ISO 9594-8/ITU-T Rec. X.509 (1997) The Directory: Authentication framework

[3] R. Housley, W. Ford, W. Polk, D. Solo. "Internet X.509 Public Key Infrastructure Certificate and CRL Profile". RFC 2459. January 1999.

[4] ISO 9594-8/ITU-T Rec. X.509 (2001) The Directory: Public-key and attribute certificate frameworks

[5] R. Housley, W. Polk, W. Ford, D. Solo "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile". RFC 3280, April 2002

[6] ISO/IEC JTC 1/SC 21 N9214 "Proposed Draft Amendments PDAM 4 to ISO/IEC 9594-2, PDAM 2 to ISO/IEC 9594-6, PDAM 1 to ISO/IEC 9594-7, and PDAM 1 to ISO/IEC 9594-8", Orlando, USA, Dec 1994

[7] ISO/IEC JTC 1/SC 21/WG 4 and ITU-T Q15/7 Collaborative Editing Meeting on the Directory "Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions", Ottawa, Canada, July 1995

[8] ITU-T. "Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks Technical Corrigendum 1". Oct 2001.

[9] Private communications with Hoyt Kesterson (X.500 rapporteur), Warwick Ford (national delegate), and Steve Kent (PKIX chair).
[10] Marcus Raunum. “The six dumbest things in computer security”. Available from http://www.ranum.com/security/computer_security/editorials/dumb/

Acknowledgements

The author would like to thank Hoyt Kesterson for providing the historical ISO/ITU-T documents on which this paper is based.

Annex 1. The original PDAM Definition of Name Constraints

12.5.2.2 Name constraints field

This field specifies a set of constraints with respect to the names for which subsequent CAs in a certification path may issue certificates. The following ASN.1 type defines this field:

```
nameConstraints EXTENSION ::= {
      SYNTAX              NameConstraintsSyntax
      IDENTIFIED BY { id-ce 11 } }

NameConstraintsSyntax ::= SEQUENCE OF SEQUENCE {
      policySet           [0] CertPolicySet OPTIONAL,
                  -- If policySet is omitted, the constraints
                  -- apply to all policies for which the
                  -- certificate is applicable
      nameSpaceConstraint     [1] NameSpaceConstraint OPTIONAL,
      nameSubordConstraint    [2] NameSubordConstraint OPTIONAL }

NameSpaceConstraint ::= SEQUENCE OF SubtreeSpecification

      (CONSTRAINED BY { -- specificationFilter is not permitted -- })

NameSubordConstraint ::= SEQUENCE {
      subordType ENUMERATED {
                  subordinateToCA             (0),
                  subordinateToCAsSuperior    (1) }
                        DEFAULT subordinateToCAsSuperior,
      skipCerts   INTEGER DEFAULT 0 }
```

This extension is always critical. The fields are interpreted as follows:

— policySet**: This indicates those certificate policies to which the constraints apply. If this component is omitted, the constraints apply regardless of policy.**

— **nameSpaceConstraint**: If this constraint is present, a certificate issued by the subject CA of this certificate should only be considered valid if for a subject within one of the specified subtrees. Any subtree class specification may contain a chop specification; if there is no chop specification, a subtree is considered to extend to the leaves of the DIT.

— **nameSubordConstraint**: This constraint is associated with a nominated CA in the certification path, being either the subject CA of this certificate or a CA which is the subject of a subsequent certificate in the certification path. If the value **subordinateToCA** is specified then, in all certificates in the certification path starting from a certificate issued by the nominated CA, the subject name must be subordinate to the issuer name of the same certificate. If the value **subordinateToCAsSuperior** is specified then, in all certificates in the certification path starting from a certificate issued by the nominated CA, the subject name must be subordinate to the name of the immediately superior DIT node of the issuer of the same certificate. The value of **skipCerts** indicates the number of certificates in the certification path to skip before the name subordination constraint takes effect; if value 0, the constraint starts to apply with certificates issued by the subject CA of this certificate.

Notes

1 The name constraint capability provided through the **subtreesConstraint** field in the basic constraints extension may be adequate for many applications. The name constraints extension is an alternative which offers a

more powerful range of constraining options, including the ability to fully reflect Internet Privacy Enhanced Mail [RFC 1422] rules.

2 The **subordinateToCA** alternative is provided only for compatibility with the Internet Privacy Enhanced Mail [RFC 1422] conventions. The **subordinateToCAsSuperior** rule is more powerful and its use is recommended in new infrastructures.

Imported from X.501(93)

```
SubtreeSpecification ::= SEQUENCE {
        base                    [0]     LocalName DEFAULT { },
                                        COMPONENTS OF ChopSpecification,
        specificationFilter     [4]     Refinement OPTIONAL }
            -- empty sequence specifies whole administrative area

ChopSpecification ::= SEQUENCE {
        specificExclusions      [1]     SET OF CHOICE {
                                            chopBefore  [0] LocalName,
                                            chopAfter   [1] LocalName } OPTIONAL,
        minimum                 [2]     BaseDistance DEFAULT 0,
        maximum                 [3]     BaseDistance OPTIONAL}
```

Annex 2. The X.509 (1997) Standard Definition of Name Constraints

12.4.2.2 Name constraints field

This field, which shall be used only in a CA-certificate, indicates a name space within which all subject names in subsequent certificates in a certification path must be located. This field is defined as follows:

```
nameConstraints EXTENSION ::= {
        SYNTAX      NameConstraintsSyntax
        IDENTIFIED BY id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
        permittedSubtrees [0]       GeneralSubtrees OPTIONAL,
        excludedSubtrees  [1]       GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
        base            GeneralName,
        minimum             [0]     BaseDistance DEFAULT 0,
        maximum             [1]     BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

If present, the **permittedSubtrees** and **excludedSubtrees** components each specify one or more naming subtrees, each defined by the name of the root of the subtree and, optionally, within that subtree, an area that is bounded by upper and/or lower levels. If **permittedSubtrees** is present, of all the certificates issued by the subject CA and subsequent CAs in the certification path, only those certificates with subject names within these subtrees are acceptable. If **excludedSubtrees** is present, any certificate issued by the subject

CA or subsequent CAs in the certification path that has a subject name within these subtrees is unacceptable. If both **permittedSubtrees** and **excludedSubtrees** are present and the name spaces overlap, the exclusion statement takes precedence.
Of the name forms available through the **GeneralName** type, only those name forms that have a well-defined hierarchical structure may be used in these fields. The **directoryName** name form satisfies this requirement; when using this name form a naming subtree corresponds to a DIT subtree. Conformant implementations are not required to recognize all possible name forms. If the extension is flagged critical and a certificate-using implementation does not recognize a name form used in any **base** component, the certificate shall be handled as if an unrecognized critical extension had been encountered. If the extension is flagged non-critical and a certificate-using implementation does not recognize a name form used in any **base** component, then that subtree specification may be ignored. When a certificate subject has multiple names of the same name form (including, in the case of the **directoryName** name form, the name in the subject field of the certificate if non-null) then all such names shall be tested for consistency with a name constraint of that name form.

> NOTE — When testing certificate subject names for consistency with a name constraint, names in non-critical subject alternative name extensions should be processed, not ignored.

The **minimum** field specifies the upper bound of the area within the subtree. All names whose final name component is above the level specified are not contained within the area. A value of **minimum** equal to zero (the default) corresponds to the base, i.e. the top node of the subtree. For example, if **minimum** is set to one, then the naming subtree excludes the base node but includes subordinate nodes.
The **maximum** field specifies the lower bound of the area within the subtree. All names whose last component is below the level specified are not contained within the area. A value of **maximum** of zero corresponds to the base, i.e. the top of the subtree. An absent **maximum** component indicates that no lower limit should be imposed on the area within the subtree. For example, if **maximum** is set to one, then the naming subtree excludes all nodes except the subtree base and its immediate subordinates.
This extension may, at the option of the certificate issuer, be either critical or non-critical. It is recommended that it be flagged critical, otherwise a certificate user may not check that subsequent certificates in a certification path are located in the name space intended by the issuing CA.
If this extension is present and is flagged critical then a certificate-using system shall check that the certification path being processed is consistent with the value in this extension.

From Section 12.3.2.1

```
GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]  INSTANCE OF OTHER-NAME,
    rfc822Name                  [1]  IA5String,
    dNSName                     [2]  IA5String,
    x400Address                 [3]  ORAddress,
    directoryName               [4]  Name,
    ediPartyName                [5]  EDIPartyName,
    uniformResourceIdentifier   [6]  IA5String,
    iPAddress                   [7]  OCTET STRING,
    registeredID                [8]  OBJECT IDENTIFIER }
```

```
OTHER-NAME ::= TYPE-IDENTIFIER
```

Annex 3. The 2001 Corrigendum Definition of Name Constraints

8.4.2.2Name constraints extension

This field, which shall be used only in a CA-certificate, indicates a name space within which all subject names in subsequent certificates in a certification path must be located. This field is defined as follows:

```
nameConstraints EXTENSION ::= {
        SYNTAX              NameConstraintsSyntax
        IDENTIFIED BY       id-ce-nameConstraint }


NameConstraintsSyntax ::= SEQUENCE {
        permittedSubtrees           [0]     GeneralSubtrees OPTIONAL,
        excludedSubtrees            [1]     GeneralSubtrees OPTIONAL,
        requiredNameForms           [2]     NameForms OPTIONAL  }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
        base                GeneralName,
        minimum     [0]     BaseDistance DEFAULT 0,
        maximum     [1]     BaseDistance OPTIONAL }


BaseDistance ::= INTEGER (0..MAX)


NameForms  ::= SEQUENCE {
    basicNameForms      [0]     BasicNameForms OPTIONAL,
    otherNameForms      [1]     SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ -- none; i.e.: at least one component shall be present -- }))

BasicNameForms  ::= BIT STRING {
        rfc822Name                  (0),
        dNSName                     (1),
        x400Address                 (2),
        directoryName               (3),
        ediPartyName                (4),
        uniformResourceIdentifier   (5),
        iPAddress                   (6),
        registeredID                (7) }  (SIZE (1..MAX))
```

If present, the **permittedSubtrees** and **excludedSubtrees** components each specify one or more naming subtrees, each defined by the name of the root of the subtree and optionally, within that subtree, an area that is bounded by upper and/or lower levels. If **permittedSubtrees** is present, subject names within these subtrees are acceptable. If **excludedSubtrees** is present, any certificate issued by the subject CA or subsequent CAs in the certification path that has a subject name within these subtrees is unacceptable. If both **permittedSubtrees** and **excludedSubtrees** are present and the name spaces overlap, the exclusion statement takes precedence for names within that overlap. If neither permitted nor excluded subtrees are specified for a name form, then any name within that name

form is acceptable. If **requiredNameForms** is present, all subsequent certificates in the certification path must include a name of at least one of the required name forms.

If **permittedSubtrees** is present, the following applies to all subsequent certificates in the path. If any certificate contains a subject name (in the **subject** field or **subjectAltNames** extension) of a name form for which permitted subtrees are specified, the name must fall within at least one of the specified subtrees. If any certificate contains only subject names of name forms other than those for which permitted subtrees are specified, the subject names are not required to fall within any of the specified subtrees. For example, assume that two permitted subtrees are specified, one for the DN name form and one for the rfc822 name form, no excluded subtrees are specified, but **requiredNameForms** is specified with the **directoryName** bit and **rfc822Name** bit present. A certificate that contained only names other than a directory name or rfc822 name would be unacceptable. If **requiredNameForms** were not specified, however, such a certificate would be acceptable. For example, assume that two permitted subtrees are specified, one for the DN name form and one for the rfc822 name form, no excluded subtrees are specified, and **requiredNameForms** is not present. A certificate that only contained a DN and where the DN is within the specified permitted subtree would be acceptable. A certificate that contained both a DN and an rfc822 name and where only one of them is within its specified permitted subtree would be unacceptable. A certificate that contained only names other than a DN or rfc822 name would also be acceptable.

If **excludedSubtrees** is present, any certificate issued by the subject CA or subsequent CAs in the certification path that has a subject name (in the **subject** field or **subjectAltNames** extension) within these subtrees is unacceptable. For example, assume that two excluded subtrees are specified, one for the DN name form and one for the rfc822 name form. A certificate that only contained a DN and where the DN is within the specified excluded subtree would be unacceptable. A certificate that contained both a DN and an rfc822 name and where at least one of them is within its specified excluded subtree would be unacceptable.

When a certificate subject has multiple names of the same name form (including, in the case of the **directoryName** name form, the name in the subject field of the certificate if non-null), then all such names shall be tested for consistency with a name constraint of that name form.

If **requiredNameForms** is present, all subsequent certificates in the certification path must include a subject name of at least one of the required name forms.

Of the name forms available through the **GeneralName** type, only those name forms that have a well-defined hierarchical structure may be used in the **permittedSubtrees** and **excludedSubtrees** fields. The **directoryName** name form satisfies this requirement; when using this name form a naming subtree corresponds to a DIT subtree.

The **minimum** field specifies the upper bound of the area within the subtree. All names whose final name component is above the level specified are not contained within the area. A value of **minimum** equal to zero (the default) corresponds to the base, i.e. the top node of the subtree. For example, if **minimum** is set to one, then the naming subtree excludes the base node but includes subordinate nodes.

The **maximum** field specifies the lower bound of the area within the subtree. All names whose last component is below the level specified are not contained within the area. A value of **maximum** of zero corresponds to the base, i.e. the top of the subtree. An absent **maximum** component indicates that no lower limit should be imposed on the area within the subtree. For example, if **maximum** is set to one, then the naming subtree excludes all nodes except the subtree base and its immediate subordinates.

This extension may, at the option of the certificate issuer, be either critical or non-critical. It is recommended that it be flagged critical, otherwise a certificate user may not check that subsequent certificates in a certification path are located in the name space intended by the issuing CA.

Conformant implementations are not required to recognize all possible name forms.

If the extension is present and is flagged critical, a certificate-using implementation must recognize and process all name forms for which there is both a subtree specification (permitted or excluded) in the extension and a corresponding value in the **subject** field or **subjectAltNames** extension of any subsequent certificate in the certification path. If an unrecognized name form appears in both a subtree specification and a subsequent certificate, that certificate shall be handled as if an unrecognized critical extension was encountered. If any subject name in the certificate falls within an excluded subtree, the certificate is unacceptable. If a subtree is specified for a name form that is not contained in any subsequent certificate, that subtree can be ignored. If the **requiredNameForms** component specifies only unrecognized name forms, that certificate shall be handled as if an unrecognized critical extension was encountered. Otherwise, at least one of the recognized name forms must appear in all subsequent certificates in the path.

If the extension is present and is flagged non-critical and a certificate-using implementation does not recognize a name form used in any **base** component, then that subtree specification may be ignored. If the extension is flagged non-critical and any of the name forms specified in the **requiredNameForms** component are not recognized by the certificate-using implementation, then the certificate shall be treated as if the **requiredNameForms** component was absent.

NAVIGATING REVOCATION THROUGH ETERNAL LOOPS AND LAND MINES

Santosh Chokhani, Orion Security Solutions, Inc.
Carl Wallace, Orion Security Solutions, Inc.

ABSTRACT

Public Key Infrastructure (PKI) trust architectures can lead to certification paths that can not be validated due to mutual dependencies between the certification path and authenticated revocation information. Some trust architectures that can lead to such circularity are based on X.509 features such as self-issued certificates to facilitate Certification Authority (CA) re-key and use of separate keys for signing certificates and Certificate Revocation Lists (CRLs). In this paper, we explore such architectures and alternatives to validate the certification paths with circular dependencies.

1 INTRODUCTION

[X509] and [RFC3280] provide a rich set of mechanisms to manage PKIs in a flexible manner commensurate with the operational security needs of most Enterprises. Some of these mechanisms involve self-issued certificates, which are defined as certificates a CA issues to itself for purposes such as re-key and separation of certificate and CRL signing keys. Technically, a self-issued certificate is defined as a certificate with identical Issuer and Subject names. Self-signed certificates are a form of self-issued certificate for which the signature can be verified using the public key in that certificate. Self-issued certificates that are not self-signed must be verifiable to avoid the need for secure distribution.

Self-issued certificates can lead to circularity. We explore how trust architecture decisions can lead to circularity, how to avoid the circularity and how to deal with it when encountered.

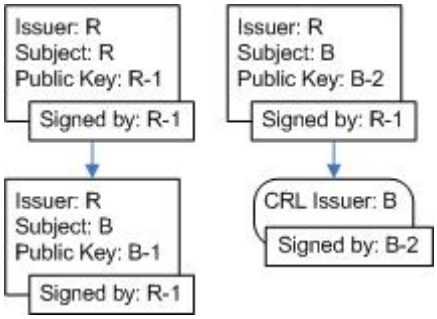
Section 2 describes the conventions used in this paper. Section 3 discusses the circularity problems due to self-issued certificates, including circumstances under which self-issued certificates can cause circularity problems and how to deal with them. Section 4 discusses the circularity problems due to indirect CRLs and describes techniques to deal with them. Section 5 discusses these issues and provides recommendations for Online Certificate Status Protocol (OCSP). Section 6 provides some recommendations regarding the development of certification paths for CRL issuers and OCSP Responders. Section 7 is a summary of findings and recommendations.

2 CONVENTIONS

In most PKIs, formal and semi-formal specifications such as notations and diagrams represent the certifying CA and subject using a single variable, which generally implies the name. This approach is found to be incomplete for this paper. The notation used throughout this document is a *(name, key)* 2-tuple, since a CA may use one of several keys to sign a certificate or CRL. Throughout, certificates are represented as a pair of rectangles that indicate the issuer name, subject name, public key, signing key and any extensions relevant to the example depicted by the diagram. Revocation information is represented by a rounded corner rectangle that indicates the revocation information issuer and signing key.

Certificate $(B, B\text{-}1)_{R, R\text{-}1}$ represents the certificate with subject DN B and public key B-1 that was issued by CA R with a signature that can be verified using the public key R-1. This is illustrated in Figure 2-1.

Similarly, $CRL_{B, B\text{-}2}$ represents a CRL issued by B with signature that can be verified using public key B-2. This is illustrated in Figure 2-2.



Figures 2-1 and 2-2: Certificate and CRL representation conventions

$OCSP_{O, O\text{-}1}$ represents an OCSP Response issued by OCSP Responder O with a signature that can be verified using public key O-1.

3 CIRCULARITY DUE TO SELF-ISSUED CERTIFICATES

A CA issues itself a certificate for a variety of reasons, such as:

1. The CA is a trust anchor (TA) and uses a self-signed certificate to promulgate a trusted public key of the CA.

2. The CA re-keys and issues a certificate to itself certifying the old key with the new and/or new key with the old in order provide trust paths to the relying parties. Figure 3-1 illustrates this scenario with the old key B-1 certifying the new key B-2.

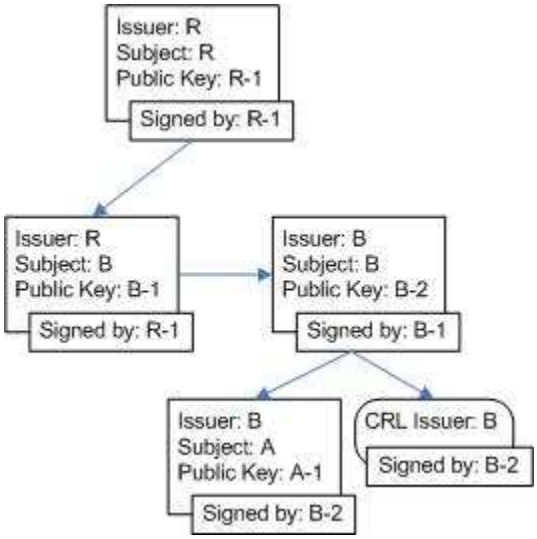


Figure 3-1: Self-Issued Certificate for CA Rekey

3. The CA issues itself a certificate for a CRL signing key. The CA may maintain separate certificate and CRL signing keys for the reasons for operational security. For example, a CRL signing key may be available on-line continuously, whereas the certificate signing key may only be used only when necessary and may be subject to stronger controls, i.e., two-person control. Additionally, using a separate CRL signing key provides less data for an attacker to perform cryptanalysis on the certificate signing key. Figure 3-2 illustrates this scenario.

Case 1 above does not lead to circularity and hence is not discussed further. Cases 2 and 3 are discussed in the following subsections in terms of how they can lead to circularity and how one can deal with it.

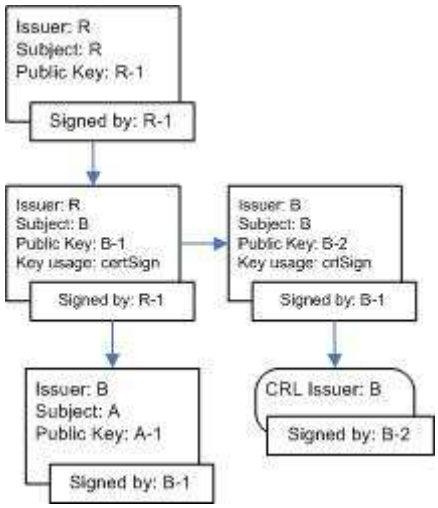


Figure 3-2: Self-Issued Certificate for CRL Signing Key

3.1 Circularity due to Self-Issued Certificates for Re-Key

In Figure 3-1, the certification path for A consists of the following ordered sequence of certificates[1]:

Certificate (B, B-1)$_{R, R-1}$; Certificate (B, B-2)$_{B, B-1}$; Certificate (A, A-1)$_{B, B-2}$ --- Path 1

The revocation status of Certificate (B, B-1)$_{R, R-1}$ can be obtained by using CRL$_{R, R-1}$. The revocation status of Certificate (B, B-2)$_{B, B-1}$ and Certificate (A, A-1)$_{B, B-2}$ can be obtained by using CRL$_{B, B-2}$. To verify signature on the CRL$_{B, B-2}$, a subset of the Path 1 must be validated:

Certificate (B, B-1)$_{R, R-1}$;
Certificate (B, B-2)$_{B, B-1}$ -- Path 2

Validation of Path 2 requires revocation status checking of the two certificates. The revocation status of Certificate (B, B-1)$_{R, R-1}$ can be obtained by using CRL$_{R, R-1}$. The revocation status of Certificate (B, B-2)$_{B, B-1}$ can be obtained by using CRL$_{B, B-2}$. Thus, in order to check the signature on this CRL we must validate the same CRL again, leading to circularity. There are several ways to mitigate the problem. The following

[1] Throughout, root certificates are not represented in certification paths and root-generated revocation information is not shown in the diagrams.

techniques are described in the subsections below, in order of preference:

- Obtain a certificate for the new key from the parent CA
- Sign CRLs using all valid keys
- Use a no-check extension
- Relax CRL checking requirements

3.1.1 Obtain Certificate from Parent

Under this approach, CA B will obtain a certificate for key B-2 from CA R. This scenario is depicted in Figure 3-3.

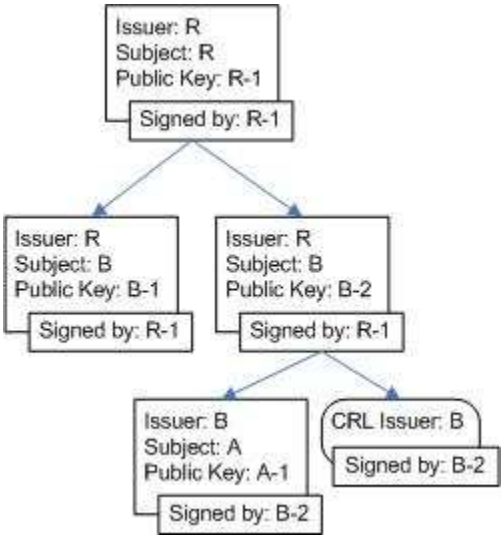


Figure 3-3: CA Re-Key: No Self-Issued Certificate

In Figure 3-3, the certification path to A consists of the following ordered sequence of certificates:

Certificate (B, B-2)$_{R, R-1}$; Certificate (A, A-1)$_{B, B-2}$

The revocation status of Certificate (B, B-2)$_{R, R-1}$ can be obtained by using CRL$_{R, R-1}$. The revocation status of Certificate (A, A-1)$_{B, B-2}$ can be obtained by using CRL$_{B, B-2}$. To verify signature on the CRL$_{B, B-2}$ a subset of the path to A must be validated:

Certificate (B, B-2)$_{R, R-1}$

Validation of this path requires revocation status checking of Certificate (B, B-2)$_{R, R-1}$ only, which can be obtained by using CRL$_{R, R-1}$.

This approach eliminates circularity problem by eliminating self-issued certificates. This also results in shorter certification paths.

A drawback of this approach is that CA R may not be available to issue Certificate (B, B-2)$_{R, R-1}$ for a period of time. If that period is not acceptable, some of the other approaches described in later sections can be used.

3.1.1.1 Trust Anchors

Self-issued certificates are also used to promulgate the new public key when a TA re-keys. So, the question arises, what to do if B is a TA? Let us assume B-1 and B-2 are the old key and the new key respectively for TA B. Then, self-issued certificates are useful to maintain chains of trust until:

1. Certificates issued under the old TA expire: Certificate (B, B-1)$_{B, B-2}$ is helpful in this scenario to aid the relying parties who do not have the old TA.

2. Relying parties install the new TA: Certificate (B, B-2)$_{B, B-1}$ is helpful in this scenario to aid relying parties who have not yet installed the new TA.

It is recommended that,

notAfter date in Certificate (B, B-1)$_{B, B-2}$ = latest *notAfter* in certificates signed by B-1.

This will ensure that a trust path for certificates issued under the old TA is available to the relying parties that do not have the old TA, but have the new TA. This is also secure from a crypto period viewpoint since it does not extend the life of B-1. The self-issued certificate will expire prior to the expiration of the crypto period for B-2. Note that this is assumes the crypto period for B-2 $\geq$ the crypto period for B-1, which must be assured when the re-key operation is performed.

notAfter date in Certificate (B, B-2)$_{B, B-1}$ $\leq$ latest *notAfter* date in certificates signed by B-1

Prior to expiry of this certificate, the relying party must install the new TA. It is assumed that the relying party with the old TA will have a need to obtain a new certificate for themselves and can obtain the new TA at that time.

In order to ensure proper checking of the revocation status of the self-issued Certificate

$(B, B\text{-}1)_{B, B\text{-}2}$, the list of revoked certificates should be signed using the new key.
In order to ensure proper checking of the revocation status of the self-issued Certificate $(B, B\text{-}2)_{B, B\text{-}1}$, the list of revoked certificates should be signed using the old key. This list needs to be signed until all the certificates signed using the old key have expired, including Certificate $(B, B\text{-}2)_{B, B\text{-}1}$.

Thus, the list of revoked certificates should be signed using both the old and the new key (resulting in two CRLs) until all the certificates signed using the old key have expired. This also aids in interoperation with MSFT CAPI, which requires CRLs to be signed using the same key as certificate checked using that CRL. This approach is same as the one described below in Section 3.1.2 for intermediate CAs. Note, the CRLs signed by the old key include entries for certificates signed by the new key, and vice versa.

3.1.2 Sign CRL with All Valid Certificates

Under this approach, B signs the same list of revoked certificates using all of its active private keys. A private key is considered active if there is a certificate issued by the CA that has not yet expired (including a self-issued certificate) and can be verified using the companion public key. Figure 3-4 illustrates this scenario.

In Figure 3-4, the certification path for A consists of the following ordered sequence of certificates:

Certificate $(B, B\text{-}1)_{R, R\text{-}1}$; Certificate $(B, B\text{-}2)_{B, B\text{-}1}$; Certificate $(A, A\text{-}1)_{B, B\text{-}2}$ --- Path 5

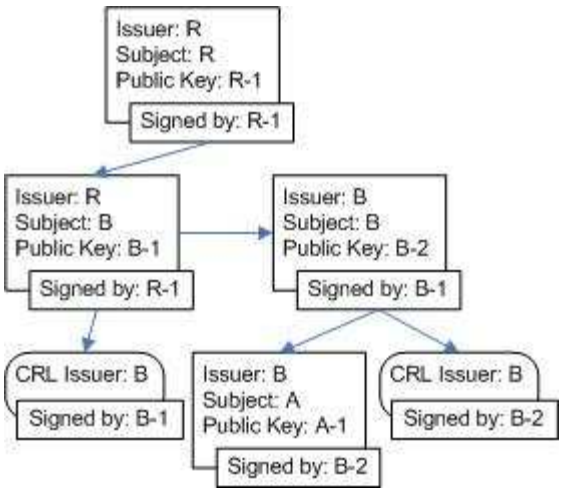


Figure 3-4: CA Re-Key: Multiple CRLs

The revocation status of Certificate $(B, B\text{-}1)_{R, R\text{-}1}$ can be obtained by using $CRL_{R, R\text{-}1}$. The revocation status of Certificate $(B, B\text{-}2)_{B, B\text{-}1}$ can be obtained by using $CRL_{B, B\text{-}1}$. Revocation status of Certificate $(A, A\text{-}1)_{B, B\text{-}2}$ can be obtained by using $CRL_{B, B\text{-}2}$.

To verify signature on the $CRL_{B, B\text{-}1}$ a subset of the certification path 5 above needs to be validated:

Certificate $(B, B\text{-}1)_{R, R\text{-}1}$ --- Path 6

To verify signature on the $CRL_{B, B\text{-}2}$ a subset of the certification path 5 above needs to be validated:

Certificate $(B, B\text{-}1)_{R, R\text{-}1}$;
Certificate $(B, B\text{-}2)_{B, B\text{-}1}$ -- Path 7

Since revocation status of Certificate $(B, B\text{-}2)_{B, B\text{-}1}$ can be verified using $CRL_{B, B\text{-}1}$, this approach eliminates circular dependencies. The approach has the advantage of being aligned with widely used toolkits such as Microsoft (MSFT) Cryptographic Applications Programming Interface (CAPI), which requires that a certificate and CRL be signed using the same key. This obviates the need to develop and verify certification paths for CRLs.

In general, since all active keys are used to sign the CRL, a self-issued certificate revocation status can always be determined by using the CRL signed by the same key as the self-issued certificate is signed with.

The approach also works if B is a TA as illustrated in Section 3.1.1.1.

The only drawback of the approach is that the CA must preserve all active keys and use them to issue the CRLs. If that is not feasible, other approaches described in this paper can be used.

3.1.3 Use No-Check Extension

Under this approach, self-issued certificates are valid for a limited period of time until the parent CA is available to certify the new public key. In interim time, the certificate contains a no-check extension, obviating the need for checking the revocation status of the certificate. Figure 3-5 illustrates this approach.

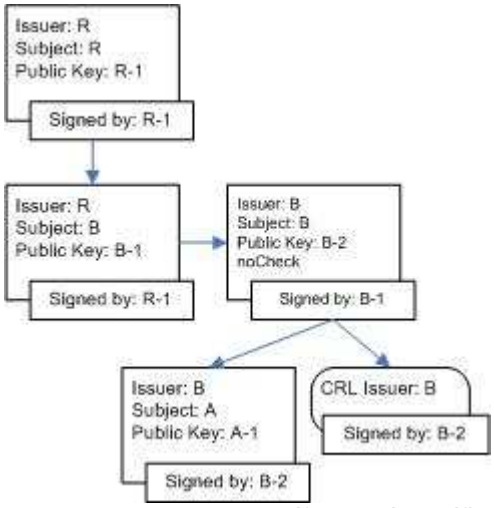


Figure 3-5: CA Re-Key: Self-Issued Certificate with No-Check

In Figure 3-5, the certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1)$_{R, R-1}$; Certificate (B, B-2)$_{B, B-1}$; Certificate (A, A-1)$_{B, B-2}$ --- Path 8

Revocation status of Certificate (B, B-1)$_{R, R-1}$ can be obtained by using CRL$_{R, R-1}$ (not depicted in Figure 3-5). Revocation status of Certificate (B, B-2)$_{B, B-1}$ is not required due to no-check. Revocation status of Certificate (A, A-1)$_{B, B2}$ can be obtained by using CRL$_{B, B-2}$.

To verify signature on the CRL$_{B, B-2}$ a subset of the certification path 8 above needs to be validated:

Certificate (B, B-1)$_{R, R-1}$;
Certificate (B, B-2)$_{B, B-1}$ --- Path 9

Since revocation status of Certificate (B, B2)$_{B, B-1}$ need not be checked, eliminating circular dependencies. This approach works when B is a TA.

This approach raises the following questions:

1. What to do if key 2 is compromised prior to expiry of (B, B-2)$_{B, B-1}$?
2. Is the approach in standards-compliant?
3. Do commercial products support this approach?

Item 1 can be mitigated by the CA operationally requesting revocation of B-1 when B-2 is compromised in advance of (B, B-2)$_{B, B-1}$ expiry. Thus, the CA is intrinsically linking B-1 and B-2. To reduce the temporal window, (B, B-2)$_{B, B-1}$ can be issued for a short period of time until R is ready to issue a certificate for (B, B-2). If B has to wait for a long period of time for R, B can keep on issuing certificates for short periods until R is ready. While the validity period of (B, B-2)$_{B, B-1}$ is dependent on the overall security requirements of the PKI being developed, based on the past experience it should be no greater than for the OCSP Responder, which is on the order of one month for many PKIs. When defining this period, it should be recognized that the compromise of key 2 can do more harm than the compromise of an OCSP Responder, but the mitigating factor is that the CA can always request revocation of key 1 eliminating trust paths involving (B, B-2)$_{B, B-1}$.

Item 2 is one of those gray areas. Strictly speaking, the use of no-check extension is limited to OCSP Responder certificates, but since effect is the same, one could argue that no or minimal change to the standard is required.

Item 3 requires further investigation with toolkits such as MSFT CAPI and PKIF.

3.1.4 Relax CRL Checking Constraint

Another approach is to relax CRL checking for self-issued certificates. This approach is as described with no-check in Section 3.1.3 except no-check is not required in the certificate.

This approach raises the following questions:

1. What to do if key 2 is compromised prior to expiry of (B, B-2)$_{B, B-1}$;
2. Is the approach in compliance with the standard; and
3. Do commercial products support this approach?

Item 1 can be mitigated by the CA operationally requesting revocation of key 1 when key 2 is compromised in advance of (B, B-2)$_{B, B-1}$ expiry. Thus, the CA is intrinsically linking key 1 and key 2. To further reduce the temporal window, (B, B-2)$_{B, B-1}$ can be issued for a short period of time until R is ready to issue a certificate for (B, B-2). If B has to wait for a long period of time for R, B can keep on issuing certificates for short periods until R is ready. While the validity period of (B,

B-2)$_{B, B-1}$ is dependent on the overall security requirements of the PKI being developed, based on the past experience it should be no greater than for the OCSP Responder, which is on the order of one month for many PKIs. When defining this period, it should be recognized that the compromise of key 2 can do more harm than the compromise of an OCSP Responder, but the mitigating factor is that the CA can always request revocation of key 1 eliminating trust paths involving (B, B-2)$_{B, B-1}$.

As for item 2, the approach violates [X509].

As for Item 3, products such as MSFT CAPI and PKIF do not work using this approach since the approach is not compliant with the standard and is insecure unless certain PKI operational, non-technical assumptions are made. There are some commercial products that support this. These products should be further investigated to determine under what circumstances they do not check the revocation status. If these products accommodate circularity for indirect CRL as discussed in Section 4 and subsections thereof, that would be an area of security concern.

3.2 Circularity due to Self-Issued Certificates for CRL Signing Key

In Figure 3-2, the certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1) $_{R, R-1}$*;*
Certificate (A, A-1)$_{B, B-1}$ *--- Path 10*

Revocation status of Certificate (B, B-1) $_{R, R-1}$ can be obtained by using CRL $_{R, R-1}$ (not depicted in Figure 3-2). Revocation status of Certificate (A, A-1)$_{B,1}$ can be obtained by using CRL$_{B, B-2}$. To verify signature on the CRL$_{B, B-2}$ the following certification path needs to be validated:

Certificate (B, B-1) $_{R, R-1}$*;*
Certificate (B, B-2)$_{B, B-1}$ *--- Path 11*

Validation of certification path 11 will require revocation status checking of the two certificates. Revocation status of Certificate (B, B-1) $_{R, R-1}$ can be obtained by using CRL $_{R, R-1}$ as stated previously. Revocation status of Certificate (B, B-2)$_{B, B-1}$ can be obtained by using CRL$_{B, B-2}$. Thus, in order to check the signature on this CRL we need this same CRL to be validated, leading to circularity.

There are several ways to mitigate the problem. These ways are described in the following subsections in order of preference. These techniques are the same as the ones for re-key scenarios described in subsections of Section 3.1 except signing CRL with all keys is not a viable alternative since the CA uses different keys for signing certificates and CRLs for operational security reasons. While the analysis is very similar to that of subsections of Section 3.1, due to subtle differences it is repeated here.

3.2.1 Obtain Certificate from Parent

Under this approach, CA B will obtain a certificate for its CRL signing key from CA R. This scenario is depicted in Figure 3-6.

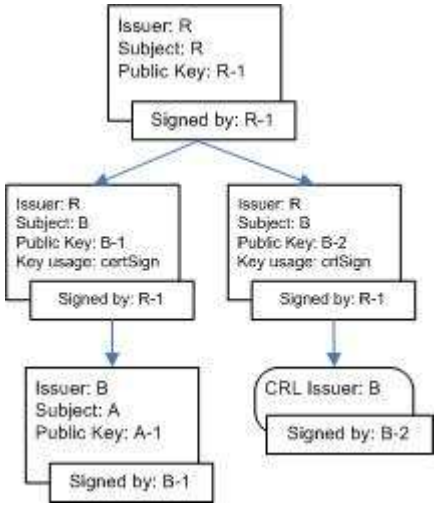


Figure 3-6: CRL Signing Key: No Self-Issued Certificate

In Figure 3-6, the certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1) $_{R, R-1}$*;*
Certificate (A, A-1)$_{B, B-1}$ *--- Path 12*

Revocation status of Certificate (B, B-1) $_{R, R-1}$ can be obtained by using CRL $_{R, R-1}$ (not depicted in Figure 3-6). Revocation status of Certificate (A, A-1)$_{B, B-1}$ can be obtained by using CRL$_{B, B-2}$. To verify signature on the CRL$_{B, B-2}$ the following certification path needs to be validated:

Certificate (B, B-2) $_{R, R-1}$*; --- Path 13*

Validation of certification path 13 will require revocation status checking of Certificate (B, B-2) $_{R, R-1}$ only, which can be obtained by using CRL $_{R, R-1}$.

This approach eliminates circularity problem by eliminating self-issued certificates.
A drawback of this approach is that CA R may not be available for some period of time to issue Certificate (B, B-2) $_{R, R-1}$. If that period is unacceptable to the PKI domain containing CA B, some of the other approaches described in later sections can be used.

If B is a TA, the simplest approach will be to use two TAs: (B, B-1)$_{B, B-1}$ for certificate signing and (B, B-2)$_{B, B-2}$ for CRL signing. A drawback of this approach is that there is no standard way to constrain (B, B-2) to no issue certificates, although toolkits such as MS CAPI may enforce *basicConstraints* and *keyUsage* extensions on TAs. Our response to the drawback is slightly different and is as follows: first, being a TA, (B,2) will not misbehave and will not sign certificates; and two, as the X.509 standard evolves to enforce the constraints in the TAs, (B, B-1)$_{B, B-1}$ could contain *keyUsage* extension with only *keyCertSign* bit set and and basicConstraints with cA = TRUE, and (B, B-2)$_{B, B-2}$ could contain *keyUsage* extension with only *cRLSign* bit set.

3.2.2 Use of No-Check Extension

A no-check extension can be used to accommodate a CRL signing key similar to the approach described in 3.1.3. In Figure 3-7, the certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1) $_{R, R-1}$*;*
Certificate (A, A-1)$_{B, B-1}$ *--- Path 14*

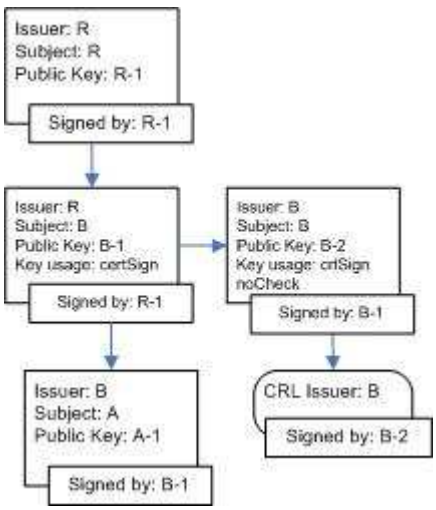


Figure 3-7: CRL Signing Key: Self-Issued Certificate with No-Check

Revocation status of Certificate (B, B-1) $_{R, R-1}$ can be obtained by using CRL $_{R, R-1}$ (not depicted in Figure 3-7). Revocation status of Certificate (A, A-1)$_{B, B-1}$ can be obtained by using CRL$_{B, B-2}$.

To verify signature on the CRL$_{B, B-2}$ the following certification path needs to be validated:

Certificate (B, B-1) $_{R, R-1}$*;*
Certificate (B, B-2)$_{B, B-1}$ *--- Path 15*

Since revocation status of Certificate (B, B-2)$_{B, B-1}$ need not be checked, eliminating circular dependencies. This approach also works when B is a TA.

This approach raises the following questions:

1. What to do if key 2 is compromised prior to expiry of (B, B-2)$_{B, B-1}$;

2. Is the approach in compliance with the standard; and

3. Do commercial products support this approach?

Item 1 can be mitigated by the CA operationally requesting revocation of key 1 when key is 2 compromised in advance of (B, B-2)$_{B, B-1}$ expiry. Thus, the CA is intrinsically linking key 1 and key 2. To further reduce the temporal window, (B, B-2)$_{B, B-1}$ can be issued for a short period and renewed. While the validity period of (B, B-2)$_{B, B-1}$ is dependent on the overall security requirements of the PKI being developed, based on the past experience it should be no greater than for the OCSP Responder, which is on the order of one month for many PKI. When defining this period, it should be recognized that the compromise of key 2 can do more harm than the compromise of an OCSP Responder (in case the CA is a TA), but the mitigating factor is that the CA can always request revocation of key 1 eliminating trust paths involving (B, B-2)$_{B, B-1}$.

Item 2 is a gray area. Strictly speaking, the use of no-check extension is limited to OCSP Responder certificates, but since effect is the same, one could argue that no or minimal change to the standard is required.

Item 3 requires further investigation with toolkits such as MSFT CAPI and PKIF.

3.2.3 Relax CRL Checking Constraint

Yet another approach is to not worry about checking CRL for self-issued certificates. This approach is same as the one described with no-check in Section 3.1.3 except no-check is not required in the certificate.

This approach raises the following questions:

1. What to do if key 2 is compromised prior to expiry of $(B, B\text{-}2)_{B, B\text{-}1}$;

2. Is the approach in compliance with the standard; and

3. Do commercial products support this approach?

Item 1 can be mitigated by the CA operationally requesting revocation of key 1 when key is 2 compromised in advance of $(B, B\text{-}2)_{B, B\text{-}1}$ expiry. Thus, the CA is intrinsically linking key 1 and key 2. To further reduce the temporal window, $(B, B\text{-}2)_{B, B\text{-}1}$ can be issued for a short period and renewed. While the validity period of $(B, B\text{-}2)_{B, B\text{-}1}$ is dependent on the overall security requirements of the PKI being developed, based on the past experience it should be no greater than for the OCSP Responder, which is on the order of one month for many PKI. When defining this period, it should be recognized that the compromise of key 2 can do more harm than the compromise of an OCSP Responder ((in case the CA is a TA), but the mitigating factor is that the CA can always request revocation of key 1 eliminating trust paths involving $(B, B\text{-}2)_{B, B\text{-}1}$.

As for item 2, the approach is a violation of [X509].

As for Item 3, products such as MSFT CAPI and PKIF do not work using this approach since the approach is not compliant with the standard and is insecure unless certain PKI operational, non-technical assumptions are made. There are some commercial products that support this. These products should be further investigated to determine under what circumstances they do not check the revocation status. If these products accommodate circularity for indirect CRL as discussed in Section 4 and subsections thereof, that would be an area of security concern.

4 CIRCULARITY IN INDIRECT CRLS

[X509] and [RFC3280] provide a mechanism for a CA to nominate another CA or a CRL issuing authority to provide revocation status of the certificates issued by the CA. The way the mechanism works is that a CA nominates another authority by asserting the authority name in the CRL Distribution Point (CRLDP) extension of a certificate. Thus, nomination of the authority is on a certificate by certificate basis. Also, note that the nominated CRL issuing authority name is used. If the nominated authority key or key hash was included, the certificate may not be valid once the nominated CRL issuing authority re-keys. It should also be noted that neither the [X509] nor [RFC3280] require that the CA issue a certificate to the nominated CRL issuing authority, designating it to issue CRL. Figure 4-1 illustrates the concept of indirect CRL.

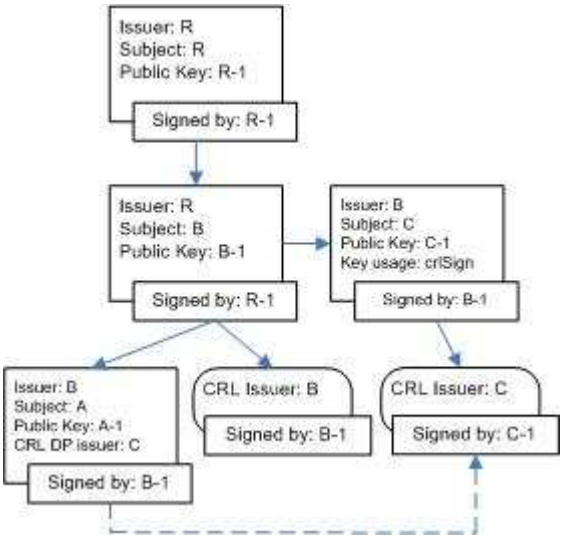


Figure 4-1: Indirect CRL Concept

The Indirect CRL can cause a circularity problem if the certificate issued to the indirect CRL issuing authority delegated the revocation status of that certificate also to the same indirect CRL issuing authority as illustrated in Figure 4-2.

In this example, certification path for A will consist of the following:

Certificate $(B, B\text{-}1)_{R, R\text{-}1}$;
Certificate $(A, A\text{-}1)_{B, B\text{-}1}$ – Path 16

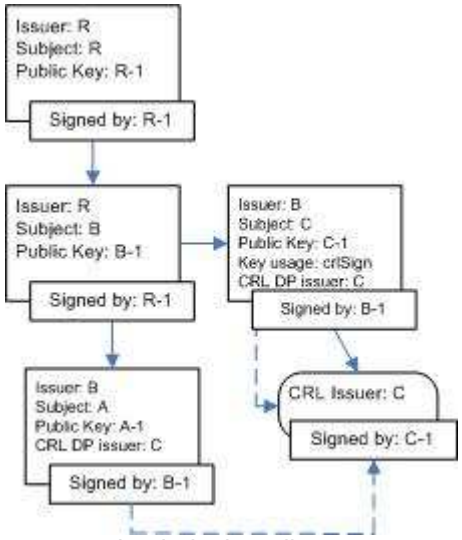


Figure 4-2: Circularity in Indirect CRL Concept

To check the revocation status of $(B, B\text{-}1)_{R, R\text{-}1}$ $CRL_{R, R\text{-}1}$ is required. To check the revocation status of Certificate $(A, A\text{-}1)_{B, B\text{-}1}$, $CRL_{C, C\text{-}1}$ (an indirect CRL) is required. To verify signature on $CRL_{C, C\text{-}1}$, the following certification path is developed:

Certificate (B, B-1) $_{R, R\text{-}1}$*;*
Certificate (C, C-1)$_{B, B\text{-}1}$ *– Path 17*

However, Certificate $(C, C\text{-}1)_{B, B\text{-}1}$ revocation status also requires $CRL_{C, C\text{-}1}$, resulting in circularity. This situation is akin to circularity due to self-issued certificate for CRL signing key (see Section 3.2), except in this case, the certificate issuing CA delegates CRL issuance to an authority with different name and not just a different key. The following sections describe the various ways to deal with the problem.

4.1 Issue a CRL for Indirect CRL Issuer

Under this approach, the CA does not nominate the indirect CRL issuing authority as the CRL issuing authority for the certificate issued to the indirect CRL issuing authority. This situation is illustrated in Figure 4-1; the absence of CRL issuer in the certificate (C, C-1) implies that B is the CRL issuer.

One drawback of this approach is that B is required to issue CRL. If B does not wish to issue CRLs, which may be one of the reasons B is nominating indirect CRL issuing authority for the certificates issued by B, this approach will not work.

4.2 Use No-Check Extension

Under this approach, a certificate is issued to the indirect CRL issuing authority for a limited period of time The certificate contains a no-check extension, obviating the need for checking the revocation status of the certificate. Figure 4-3 illustrates this approach.

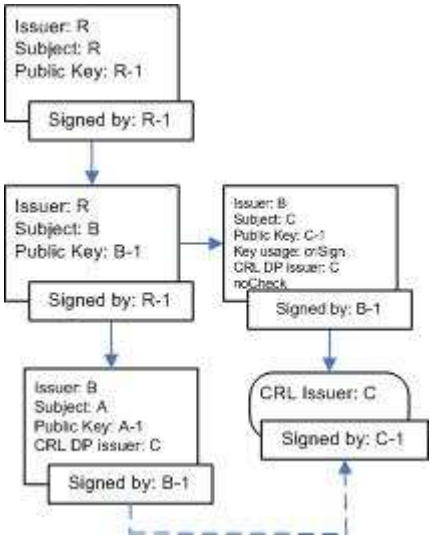


Figure 4-3: Removing Indirect CRL Circularity Using No-Check

In Figure 4-3, the certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1) $_{R, R\text{-}1}$*;*
Certificate (A, A-1)$_{B, B\text{-}1}$ *--- Path 18*

Revocation status of Certificate $(B, B\text{-}1)_{R, R\text{-}1}$ can be obtained by using $CRL_{R, R\text{-}1}$ (not depicted in Figure 4-3). Revocation status of Certificate $(A, A\text{-}1)_{B, B\text{-}1}$ can be obtained by using $CRL_{C, C\text{-}1}$ (an indirect CRL)

To verify signature on the $CRL_{C, C\text{-}1}$ the following certification path needs to be validated:

Certificate (B, B-1) $_{R, R\text{-}1}$*;*
Certificate (C, C-1)$_{B, B\text{-}1}$ *--- Path 19*

Since revocation status of Certificate $(C, C\text{-}1)_{B, B\text{-}1}$ need not be checked, eliminating circular dependencies. This approach also works when B is a TA.

This approach raises the following questions:

1. What to do if key Z is compromised prior to expiry of $(C, C\text{-}1)_{B, B\text{-}1}$?

2. Is the approach in compliance with the standard?

3. Do commercial products support this approach?

Item 1 can be mitigated by the CA operationally requesting revocation of key 1 when C reports to the CA B that key Z is compromised in advance of $(B, B\text{-}2)_{B, B\text{-}1}$ expiry. While the validity period of $(C, C\text{-}1)_{B, B\text{-}1}$ is dependent on the overall security requirements of the PKI being developed, based on the past experience it should be no greater than for the OCSP Responder, which is on the order of one month for many PKI.

Item 2 is a gray area. Strictly speaking, the use of no-check extension is limited to OCSP Responder certificates, but since effect is the same, one could argue that no or minimal change to the standard is required.

Item 3 requires further investigation with toolkits such as MSFT CAPI and PKIF.

4.3 Relax CRL Checking Constraint

Another approach is to not worry about checking CRL for delegated certificates issued to CRL issuing authority. While this approach has similar ramifications as those described in Section 3.2.3 when the certificate in question is issued by the delegating CA, the approach can lead to insecure results when the certificate to the CRL issuing authority is issued by a CA subordinate to the delegating CA. The insecure scenario is described below.

Let us assume that Figure 4-2 represents the intended trust structure from the infrastructure point of view. Let us say that CA A's key A-1 is compromised and the certificate is put on the indirect CRL as illustrated in Figure 4-4. The certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1) $_{R, R\text{-}1}$*;*
Certificate (A, A-1)$_{B, B\text{-}1}$ *--- Path 20*

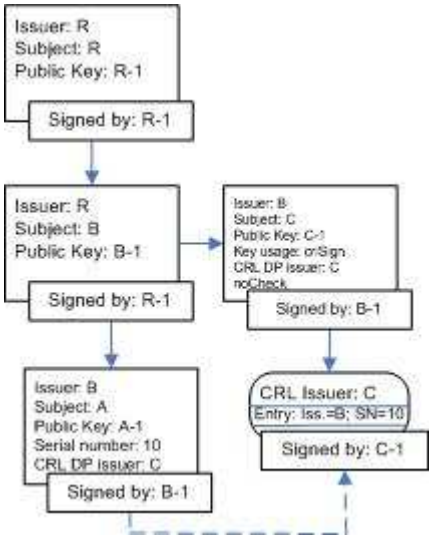


Figure 4-4: Successful Revocation of A

The revocation status of Certificate $(B, B\text{-}1)_{R, R\text{-}1}$ can be obtained by using $CRL_{R, R\text{-}1}$. The revocation status of Certificate $(A, A\text{-}1)_{B, B\text{-}1}$ can be obtained by using $CRL_{C, C\text{-}1}$ (an indirect CRL)

To verify signature on the $CRL_{C, C\text{-}1}$ the following certification path needs to be validated:

Certificate (B, B-1) $_{R, R\text{-}1}$*;*
Certificate (C, C-1)$_{B, B\text{-}1}$ *--- Path 21*

The revocation status of Certificate $(C, C\text{-}1)_{B, B\text{-}1}$ is not be checked to eliminate circular dependencies. The $CRL_{C, C\text{-}1}$ contains the certificate $(A, A\text{-}1)_{B, B\text{-}1}$, and thus path to A is properly rejected and security is preserved.

But, upon a second look, the party that has compromised the key A-1, can create a scenario described in Figure 4-5 in which the certification path for A consists of the following ordered sequence of certificates:

Certificate (B, B-1) $_{R, R\text{-}1}$*;*
Certificate (A, A-1)$_{B, B\text{-}1}$ *--- Path 22*

As before, the revocation status of Certificate $(B, B\text{-}1)_{R, R\text{-}1}$ can be obtained by using $CRL_{R, R\text{-}1}$. However, the revocation status of Certificate $(A, A\text{-}1)_{B, B\text{-}1}$ can now be obtained by using $CRL_{C, C\text{-}2}$ (an indirect CRL)

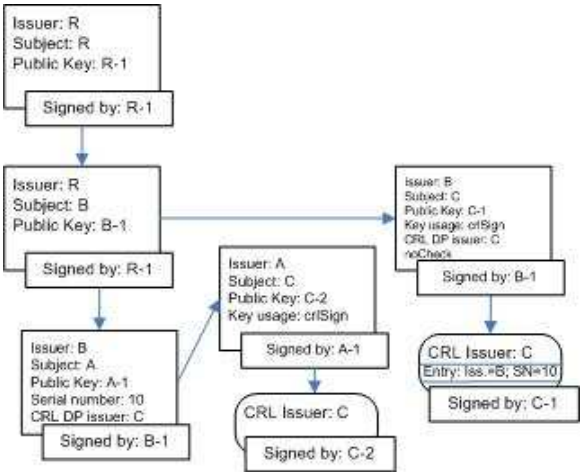


Figure 4-5: Successful Party Spoofing

To verify signature on the $CRL_{C, C-2}$ the relying party develops the following certification path:

Certificate (B, B-1) $_{R, R-1}$*; Certificate (A, A-1)*$_{B, B-1}$*; Certificate (C, C-2)*$_{A, A-1}$ *--- Path 23*

Certificate (A, A-1)$_{B, B-1}$ causes the circularity and if the revocation status of this certificate is not checked (due to circularity) during CRL signature validation, this path will lead to erroneous answer that Certificate (A, A-1)$_{B, B-1}$ is not revoked.

Since neither X.509 nor RFC 3280 impose any trust model for indirect CRL issuing delegation, this approach of ignoring circularity in indirect CRLs is not recommended.

5 CIRCULARITY IN OCSP RESPONDER CERTIFICATION PATH

Certification paths for OCSP Responder can lead to circularity. In Figure 5-1, the OCSP Responder (O, O-1) is the legitimate Responder and provides the revocation status of Certificate (B, B-1) $_{R, R-1}$. If B-1 is compromised, the compromised party can set up a rogue OCSP Responder (O, O-2) that provides the status as good.

In order to verify the signature on the OCSP Response for the status of Certificate (B, B-1) $_{R, R-1}$, the relying party could use one of the two paths listed below:

Certificate (B, B-1) $_{R, R-1}$*; Certificate (O, G)*$_{B, B-1}$*; --- Path 24*
Certificate (B, B-1) $_{R, R-1}$*; Certificate (A, A-1)*$_{B, B-1}$*; Certificate (O, H)*$_{A, A-1}$ *--- Path 25*

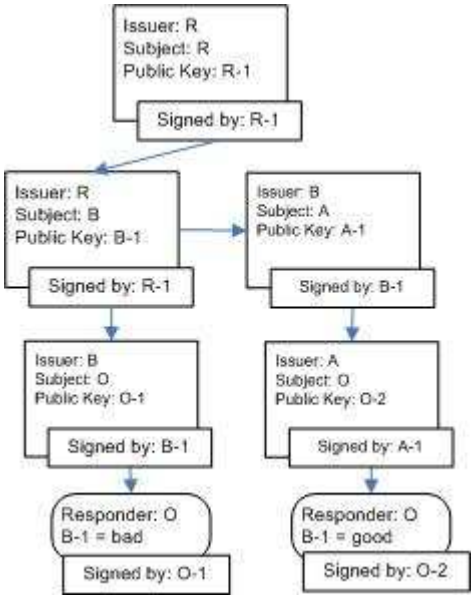


Figure 5-1: Circularity in OCSP Responses

Since B-1 is compromised, the compromising entity can give erroneously "good" status for the certificate.

It should be noted that [RFC2560] requires that the OCSP Responder certificate be limited to the following:

1. The CA that issued the certificate signs the OCSP response
2. A delegated Responder (i.e., the Responder that is issued a certificate by the CA that issued the certificate whose status is being checked) signs the OCSP response
3. Local policy based CA.

For items 1 and 2 above, the commercial products such as CoreStreet and Tumbleweed OCSP Clients further limit the CA to use the same key that was used to sign the certificates whose status the OCSP Responder provided to sign the OCSP Responder certificate. This provides simplest and strongest crypto binding and removes any circularity.

For item 2, in order to check the revocation status of the OCSP Responder Certificate, either the OCSP Responder certificate should contain a no-check extension, obviating the need to revocation checking, or a CRL issued by the CA should be used. However, using the CRL may defeat one of the key reasons for using OCSP, CRL size. If one were to download the CRL issued by the CA, one could simply check the

revocation status of the certificate, eliminating the need for OCSP altogether.

For Item 3, [RFC2560] is unclear. The two most popular OCSP clients only accept the following trust models for local policy:

1. Both clients accept OCSP Responder to be a TA under this option.
2. Tumbleweed also accepts Responders that have been issued certificates directly by a TA.

Thus, the two most popular OCSP clients will not be vulnerable to circularity scenarios in OCSP Responder trust architecture since they accept only the following trust models:

1. Responder is a TA
2. Responder is issued a certificate by a TA (only Tumbleweed)
3. Responder is using the same key as the CA uses to sign the certificate
4. Responder is issued a certificate by the CA (using the same CA key as used to sign the certificates) whose subject certificates' status is provided by the Responder.

In addition, in order to remove the circularity in OCSP Responder status checking, the OCSP Responder and any superior CA certificates must not point (in its *oCSP* field of *authorityInformationAcess* extension) to the Responder itself for revocation status checking.

6 CRL AND OCSP RESPONDER CERTIFICATION PATHS

Whenever the CRL or OCSP responder certificate are not signed using the same key as the certificate whose status is being checked, there is a need to develop a certification path for the CRL verification public key or for the OCSP Responder certificate. This can lead to problem of using wrong CRL as illustrated in Figure 6-1.

In Figure 6-1, certification path for E consists of the following:

Certificate (B, B-1) $_{R, R-1}$*; Certificate (A, K)* $_{B, B-1}$*;*
Certificate (E, M) $_{A, K}$ *--- Path 26*

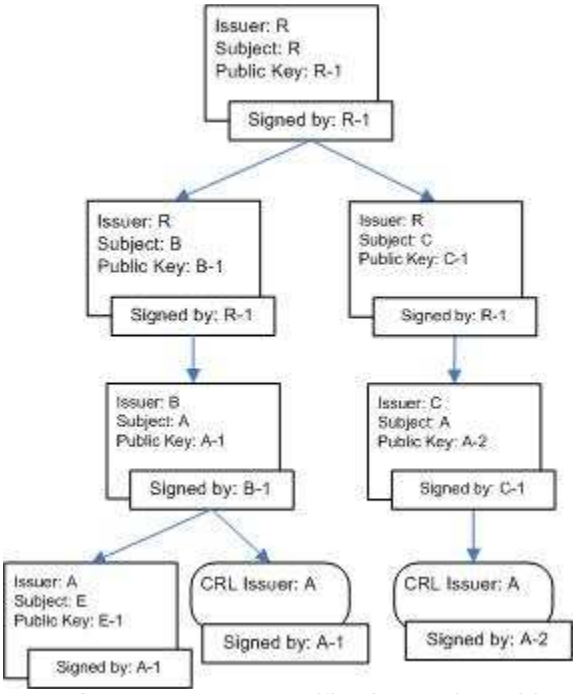


Figure 6-1: CRL Certification Path Problem

Since A has re-keyed and the CRL is signed using key L, a relying party could develop the following path for CRL to check $CRL_{A, L}$:

Certificate (B, B-1) $_{R, R-1}$*;*
Certificate (A, A-1) $_{B, B-1}$*; --- Path 27*

But, a relying party could also develop the following path for CRL and obtain and check $CRL_{A, A-1}$:

Certificate (C, C-1) $_{R, R-1}$*;*
Certificate (A, Z) $_{R, R-1}$*; --- Path 28*

In Figure 6-1, there are two distinct CAs with the name "A" and using path 28 will fetch the CRL from a different CA A than the CA A that issued a certificate to E. Thus, this CRL could give a wrong answer to the revocation status of Certificate $(E, M)_{A, K}$. Since neither [X.509] nor [RFC3280] mandate use of name constraints it is possible that different CA's will have the same name. The problem is exacerbated since most relying parties today have multiple TAs.

The simplest solution to the problem is to ensure that the CRL and OCSP Responder certificate are verified using the same key. MSFT CAPI uses this approach for the CRL and some OCSP clients use for the OCSP Responder certificate. Absent that, we propose the following algorithms for developing and validating the CRL and

OCSP Responder certification paths that are equally strong as the same key cryptographic binding, assuming a CA does not issue certificates with the same DN to distinct entities. This assumption is reasonable, because, if a CA were to violate it, relying parties would not be able to distinguish between the two entities, leading to additional problems.

The basic principle behind the approach is the certification path for the end entity (EE) and certification path for the CRL or OCSP Responder should start at the same TA DN, and all issuer and subject DNs in the path should match. Of course, the certification path size could be different and hence termination could be at different locations. It should also be noted that due to re-key and/or the use of separate keys for certificate signing and CRL signing, each path could have different set of self-issued certificates.

6.1 Direct CRL Certification Path

Let us say that we have a certification path for a certificate with starts with a TA A_0 and has a sequence of certificates. Let us assume that all self issued certificate issuer and subject DNs are ignored. Then we have the following list of issuer, subject DN 2-tuples for the certificates in the certification path. Note, since the first certificate is a TA it will be represented by a subject only, thus a 1-tuple.

$$(A_0), (A_0, A_1), (A_1, A_2) \ldots..(A_{n-1}, A_n)$$

Thus, the above is list of n+1 tuples representing the issuers and subjects in the certification path.

Now, let us say that, we develop a certification path for the CRL and after ignoring DNs in the self-issued certificates, we have the following list of issuer and subject DN 2-tuples for the certification path:

$$(B_0), (B_0, B_1), (B_1, B_2) \ldots..(B_{m-1}, B_m)$$

For the direct CRL, m = n -1 since the last certificate is the subject certificate whose status is being checked using the CRL. Our logic is based on the simple principle that if at each level of certification path we ensure that certificate and CRL certification path refer to the same CA, we will be ok. Thus, ensuring that the certification path for the certificate and CRL are correct, we use the following algorithm:

1. if (m != n-1) reject
2. For j = 0, j < m, j++
   If (A_j != Bj) reject

Using this algorithm in the Example in Figure 6-1, we have the following list of DNs for Certification path 26 for the certificate and n = 3

(R) (R, B) (B, A) (A, X)

For the CRL path represented by Path 27, we have the following as the list of DNs and m =2:

(R) (R, B) (B, A)

Thus, m = n-1 = 2 and the DNs for 0, 1, and 2 match. Thus, this CRL path is acceptable as stipulated previously.

Now, let us examine path 28. We have m =2 and the following as the list of the DNs:

(R) (R, C) (C, A)

Here also, m = n-1 = 2. So, that condition is met. For j=0, the node is same "R" for the EE certification path and for CRL certification path. But, for j=1, the EE certification path has B as the subject DN and the CRL certification path has C as the subject DN. This fails the algorithm and this CRL path and hence the associated CRL is aptly rejected.

While this additional logic increases computational complexity, it can be used to guide the CRL certification path as follows:

1. Develop certification path using path development logic native to the application.

2. Start the CRL certification with a Root whose DN is same as the DN of the Root of the certification path developed in step 1.

3. At each step, only consider certificates that are either self-issued or whose subject DN matches the next subject DN in the certification path developed in step 1. (Note: This next subject DN is considered after eliminating self-issued certificates).

4. Terminate when subject DN matches the last intermediate certificate in the step 1 path. From this point forward, only self-issued certificates can be considered.

The above approach eliminates extraneous paths and only considers paths that include the CAs in the EE certification path, resulting in fewer paths and hence reducing certification path development computational complexity.

6.2 Indirect CRL Certification Path

For indirect CRL issuers, the approach is the same as above for the Direct CRL issuers. While [X509] and [RFC3280] do not impose any constraints or offer recommendations in this area, the following should be used as guidelines for indirect CRL issuer[2]:

1. It could be a TA, leading to m = 0

2. It could be an authority directly certified by a TA leading to m = 1, however, this authority need not be in the EE certification path.

3. It could be an ancestor of the nominating CA. This would lead to m < n-1

4. It could be directly delegated (i.e., issued a certificate) by the nominating CA. This would lead to m = n. However, the last certificate (the one issued to the indirect CRL issuer) would not be in the EE certification path.

This leads to the following logic for indirect certification path matching using the same terminology as defined in the previous section.

1. If m = 1, set m = 0
2. If m = n, set m = m-1
3. For j = 0, j < m, j++
   If (A_j != Bj) reject

This approach does not completely eliminate the indirect CRL issuer name collisions, especially when one traverses two or more cross certified environments. While it may be a safe assumption that indirect CRL issuer names will not collide in an Enterprise PKI, it is not a safe assumption in cross certified environments. We considered two solutions to solve this problem:

1. Asserting the indirect CRL certificate issuer name in the CRL distribution point fields of the subject certificates. For example, in the case of Figure 4-1, the CRL distribution point extension in Certificate $(A, A\text{-}1)_{B, B\text{-}1}$ will contain C as the DN for indirect CRL issuer and B as the DN for certificate issuer to C[3]. The drawback of this approach is that while likelihood of having two pairs of B and C as certificate issuer to indirect CRL issuer and indirect CRL issuer are low, it is still possible. This approach does not completely remove the name collision problem.

2. Asserting key hash of the indirect CRL issuer public key in the CRL distribution point of the subject certificates and in the indirect CRL. For example, in the case of Figure 4-1, the CRL distribution point extension in Certificate $(A, A\text{-}1)_{B, B\text{-}1}$ will contain hash of key N and the indirect CRL issued by C will contain the hash of the key N. The relying parties can match the two key hashes to ensure that they have the correct indirect CRL. In order to account for indirect CRL issuer re-key, the CRL can contain a SEQUENCE or SET of key hashes. Furthermore, in order to reduce the number of key hashes in the CRL, the indirect CRL can stop asserting a hash in a CRL once all the certificates contains that key hash have expired. This approach provides mitigates the threat of name collisions assuming the CRL issuing authorities are not rogue.

Approach 2 is recommended for removing the name ambiguity for indirect CRL issuer. Using either approach will require an enhancement to the syntax and semantics of CRL distribution point extension, thus defining a new CRL distribution point extension with a new extension OID. It will also require using another extension of adding a field to the Issuing Distribution Point that contains the SEQUENCE or SET of key hashes for the indirect CRL issuer.

6.3 OCSP Responder Certification Path

For OCSP Responders, based on [RFC2560] and commercial implementations, the following constraints are recommended:

[2] The algorithm presented here may reject some paths that the standards permit, but there is no plausible means to ensure their security, i.e., avoid name collision.

[3] It should be noted that while we do not illustrate in this paper, the nominating CA (in this case CA B, need not be the certificate issuer to the indirect CRL issuer (in this case C).

1. It could be a TA leading to m = 0. Note that the OCSP Responder TA could be different from the EE certification path TA.

2. It could be a Responder directly certified by a TA. This would lead to m = 1 however, the Responder will not be in the EE certification path.

3. It could be an ancestor of the nominating CA. This would lead to m < n-1. This option is provided for the sake of completeness. It is recognized that for operational security, CAs generally do not sign OCSP responses.

4. It could be the CA. This would lead to m = n-1. This option is provided for the sake of completeness. It is recognized that for operational security reasons CAs generally do not sign OCSP responses.

5. It could be directly delegated (i.e., issued a certificate) by the nominating CA. This would lead to m = n. However, the last certificate (i.e., OCSP Responder) would not be in the EE certification path.

This leads to the following logic for indirect certification path matching using the same terminology as defined in the previous section.

1. If m = 0, exit with success.
2. If m = 1, set m = 0
3. If m = n, set m = m-1
4. For j = 0, j < m, j++
   If (A_j != Bj) reject

Approach 1 and 2, OCSP Responder as TA and OCSP Responder directly issued a certificate by a TA may the require Responders to collect revocation information from foreign (i.e., non-Enterprise) CAs. The discussion of this topic and OCSP architecture (in general) in Bridged environment is beyond the scope of this paper.

7 SUMMARY

In this paper, we have identified several pitfalls for certification paths. We have identified various solutions that are secure, interoperable, standards compliant, realizable using commercial products, and meet the operational constraints of the various PKIs. Specifically, we have identified the following:

- Self-issued certificates can lead to circularity. Not checking revocation status of self-issued certificates or using an untrusted public key to verify signatures to remove the circularity is non-compliant with the standards. There are several standards-compliant alternatives to remove circularity.

- Standards do not provide any constraints for trust models to support usage of indirect CRLs. This can lead to insecurity. We have proposed indirect CRL trust model constraints and associated extensions.

- Standards do not provide guidance on matching EE certification paths and CRL or OCSP Responder certification paths. This lack of guidance could lead to insecure results. We have provided a solution that can reduce the computational complexity for certification path development while enhancing security. We recommend that TA driven trust models for OCSP Responders not be used since they do not scale to cross-certified and Bridge environments, due to use of different relying party TAs.

- While the solutions presented in this paper are illustrated using simple examples of circularity, the solutions are applicable to mitigate convoluted circularities.

8 ACKNOWLEDGEMENTS

The issues described herein were uncovered during the development of PKI Framework (PKIF) toolkit and the PKI-related research for the United States Marine Corps. The authors would like to thank Captain Paul Fillmore and Mr. Mike Henry for funding and support. Ideas presented here were influenced by conversations with Tim Polk and David Cooper.

9 REFERENCES

[RFC2560] Myers, M., Ankney, R., et al., “X.509 Internet PKI Online Certificate Status Protocol - OCSP”, RFC 2560, June 1999.
[RFC3280] Housley, R., Polk, T., et al., “Internet X.509 PKI Certificate and Certificate Revocation List (CRL) Profile”, RFC 3280, April 2002.
[X509] Public key and attribute certificate framework.

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 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.

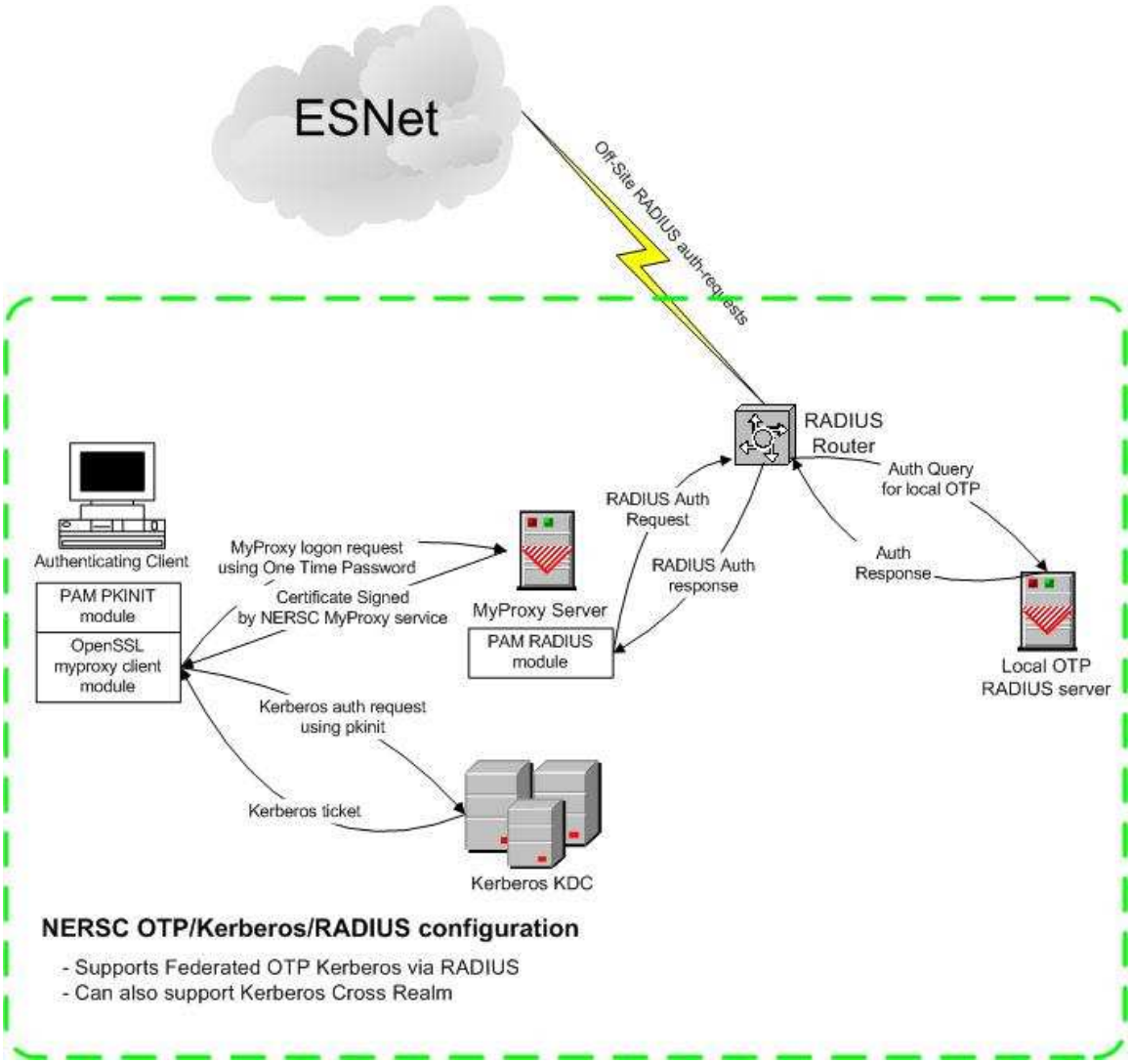


Figure 1: Logical Diagram of NERSC OTP/MyProxy environment

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 DNs 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**
Users are sometimes unaware of, or unmotivated by, the necessity for strong passphrases.	Passwords are in backend authentication system. Centralized password strength checking at backend.
Users are not always aware of the necessary filesystem permission settings on private keys to maintain security	PAM module handles short term certificates and keys on behalf of user. Long term certificates eliminated, avoiding those private keys entirely.
Credentials may be stored on shared network filesystems that are vulnerable to sniffing or authentication compromise	PAM module handles certificates – can be administratively configured to store creds in filesystem, memory, kernel keyring, HSM, etc.
Certificate revocation is not uniformly deployed by certificate authorities, nor is it uniformly checked by relying parties	Short lived (hours to days) certificates mitigate revocation issues. Configurable CA interface allows attributes such as OCSP URL to be added to certs.
If a user's passphrase is lost or forgotten, the only recourse is revocation and reissuance of the certificate.	Passphrase/password is in external authentication service (via PAM) and can be changed as appropriate.
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	Mitigated by short certificate lifetimes and the potential to embed OCSP URL attribute in certificate, enabling realtime revocation, without proving onerous to user.
Users need to have copies of their certificate and private key at every location where they will use the certificate for authentication.	MyProxy credential store is originally designed to mitigate this problem. Proposed solution builds on existing benefits.
Tools for manipulating PKI credentials (such as PGP and openssl) have usability issues.	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.
Sites typically have their own OTP systems, and cross vendor, cross realm compatibility is often lacking	Support for RADIUS fabric allows cross platform, cross site OTP authentication.
Asking users to authenticate with a different password every time they log into the same system may prove onerous in environments where Single	Certificate (or Kerberos ticket) provides persistent authentication token.

Sign-On authentication (Kerberos, Globus GSI, etc…) is already in place.	
OTP systems are not compatible with batch job schedulers, or many distributed systems platforms	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 Muruganantham, 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 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/

Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy

Tom Barton[1], Jim Basney[2], Tim Freeman[1], Tom Scavo[2],
Frank Siebenlist[1,3], Von Welch[2], Rachana Ananthakrishnan[3],
Bill Baker[2], Monte Goode[4], Kate Keahey[1,3]
[1]University of Chicago
[2]National Center for Supercomputing Applications, University of Illinois
[3]Mathematics and Computer Science Division, Argonne National Laboratory
[4]Lawrence Berkeley National Laboratory

Abstract

This paper describes the recent results of the GridShib and MyProxy projects to integrate the public key infrastructure (PKI) deployed for Grids with different site authentication mechanisms and the Shibboleth identity federation software. The goal is to enable multi-domain PKIs to be built on existing site services in order to reduce the PKI deployment and maintenance costs. An authorization framework in the Globus Toolkit is being developed to allow for credentials from these different sources to be merged and canonicalized for policy evaluation. Successes and lessons learned from these different projects are presented along with future plans.

1 Introduction

The Grid [11] communities have developed an international public key infrastructure (PKI) [18] as well as extensions to standard end entity certificates (EECs) [16] in the form of proxy certificates [40,42]. The combination of this PKI and proxy certificates is used to provide cross-domain authentication, single sign-on, and delegation for a number of large deployments (e.g., [7,31,39]).

As computational Grids have grown, there has been increasing interest in leveraging existing site authentication infrastructure to support this Grid authentication model. For example, Fermi National Accelerator Laboratory has successfully operated an online Kerberos Certification Authority for a number of years to allow its users to leverage existing Kerberos infrastructure for X.509 authentication [38].

In parallel, Shibboleth [37] has been developed by the Internet2 community and is increasingly deployed both in the U.S. and abroad as a mechanism for cross-site access control for web-based resources. Shibboleth utilizes OASIS SAML standards [21,22,29] for authentication and attribute assertions to achieve its purpose.

In this paper we cover recent work by two projects, GridShib [12,43] and MyProxy [1,25], working towards the integration of PKIs with both site authentication infrastructure and Shibboleth in order to achieve large-scale multi-domain PKIs for access control.[1] In section 2 we begin with a brief review of the Globus Toolkit and Shibboleth on which our work builds. In section 3 we summarize our work and lessons learned from the past year. We conclude in section 4 with our plans for the upcoming year.

[1] We stress this infrastructure is for access control and similar point-in-time decisions as opposed to long-term document signing, for example.

2 Prior Work

In this section we provide a brief overview of the Globus Toolkit and Shibboleth on which our work builds.

2.1 Globus Toolkit

The Globus Toolkit [10] provides basic functionality for Grid computing with services for data movement and job submission, and a framework on which higher-level services can be built. Over recent years, the Grid has been adopting Web Services technologies, and this trend is reflected in recent versions of the Globus Toolkit in implementing the Web Services Resource Framework [30] standards. This convergence of Grid and Web Services was part of our motivation for adopting Shibboleth, which is also leveraging Web Services technologies.

The Grid Security Infrastructure [44], on which the Globus Toolkit is based, uses X.509 end entity certificates (EECs) [16] and proxy certificates [42]. In brief, these certificates allow a user to assert a globally unique identifier (i.e., a distinguished name from the X.509 identity certificate). We note that in Grid scenarios there is often an organizational separation between the certificate authorities (CAs), which are the authorities of identity (authentication) and the authorities of attributes (authorization). For example, in the case of the Department of Energy (DOE) SciDAC program [36], a single CA, the DOE Grids CA [6], serves a broad community of users, while the attributes and rights for those users are determined by their individual projects (e.g., National Fusion Grid, Earth Systems Grid, or Particle Physics Data Grid).

Authorization in the Globus Toolkit is by default based on access control lists (ACLs) located at each resource. The ACLs specify the identifiers of the users allowed to access the resource. Also, higher-level services (such as CAS [32]) that provide richer authorization policies exist as optional configurations. As is discussed later, the GridShib project enhances the authorization options of the Globus Toolkit by adding standards-based attribute exchange for both authorization policies and service customization.

2.2 Shibboleth

Shibboleth[37] provides cross-domain single sign-on and attribute-based authorization while preserving user privacy. Developed by Internet2/MACE [19], Shibboleth is based in large part on the OASIS Security Assertion Markup Language (SAML). The SAML 1.1 browser profiles [17,21,34] define two functional components, an Identity Provider and a Service Provider[2]. The Identity Provider (IdP) creates, maintains, and manages user identity, while the Service Provider (SP) controls access to services and resources. An IdP produces and issues SAML assertions to SPs upon request. An SP consumes SAML assertions obtained from IdPs for the purpose of making access control decisions. Shibboleth specifies an optional third component, a "Where Are You From?" (WAYF) service to aid in the process of IdP discovery.

The Shibboleth specification [2] is a direct extension of the SAML 1.1 browser profiles [21]. While the SAML 1.1 browser profiles begin with a request to the IdP, the Shibboleth browser profiles are SP-first and therefore more complex [34].

In addition to the browser profiles, Shibboleth specifies an Attribute Exchange Profile [2]. On the IdP side, a Shibboleth Attribute Authority (AA) produces and issues attribute assertions, while a

[2] For the purposes of discussion, we adopt SAML 2.0 terminology [15] throughout this paper, although our work is currently based on SAML 1.1 technology.

subcomponent of the SP called an Attribute Requester consumes these assertions. Our work builds on Shibboleth attribute exchange with a focus on authorization and access control in the Globus Toolkit.

The current implementation of the specification is Shibboleth 1.3 (released July 2005), which has become our primary development platform. We describe extensions and enhancements to the Shibboleth Identity Provider and Service Provider components later in this paper.

3 Recent Results

In this section we provide a summary of our results from the past year.

3.1 MyProxy

MyProxy began as an online credential repository for X.509 proxy credentials encrypted by user-chosen passphrases [28]. Users authenticate to the MyProxy service to obtain short-lived (per session) proxy credentials that are delegated from credentials stored in the repository. This gives users convenient access to proxy credentials when and where needed, without requiring them to directly manage their long-lived credentials. The latter remain protected in a secure repository, where the repository administrator can monitor and control credential access.

In the past year, we have extended MyProxy to better integrate with existing site infrastructure and to make it easier for users to bootstrap their X.509 security context. New developments, described in the following sections, include management of trust roots, standards-based integration with site authentication, and the ability to act as a Certificate Authority (CA).

3.1.1 Managing Trust Roots

A user's X.509 security context includes an end entity or proxy credential, one or more trusted CA certificates, and certificate revocation information in the form of Certificate Revocation Lists (CRLs) [16] or online certificate status protocol (OCSP) [26] responses. Users can run the MyProxy Logon application to obtain their complete security context from the MyProxy service. The MyProxy administrator maintains a set of trusted CA certificates and configures the server to periodically fetch fresh CRLs. MyProxy Logon fetches the configured CA certificates and CRLs in addition to the user's end entity or proxy certificate and installs them in the local user's environment.

This work is inspired by Gutmann's "Plug-and-Play PKI" [13] which describes a PKI bootstrapping service aimed to make PKI enrollment as easy as adding a computer to the network with DHCP. Gutmann's PKIBoot service can use two methods to bootstrap mutual trust between the un-initialized client and the certificate issuer. The first method uses a shared secret (such as an enrollment password) to generate a Message Authentication Code (MAC) for each message. The second method is a variant of the "baby-duck security model" where the client trusts the first issuer it finds for the one-time bootstrap operation.

A drawback to the shared secret method is it becomes yet another password that users must remember. Common site authentication methods, such as Unix passwords, One-Time Passwords, and Kerberos, allow a service to verify a password entered by the user, but don't allow a service to lookup the user's site authentication password in advance for use in a MAC or other secure password protocol. Thus existing site passwords cannot be used and we must therefore have a unique password for the bootstrap service. In environments where users must bootstrap their PKI context repeatedly as they use different machines, it becomes necessary to maintain a long-lived password or dedicated

one-time password stream using S/Key or equivalent.

The baby-duck method is well known to SSH users, who learn the public keys of target hosts in the first connection attempt. This approach is generally accepted as "good enough" given the infrequency of connecting to a target host for the first time and the infrequency of man-in-the-middle attacks in practice relative to keystroke loggers, Trojan horses, viruses, etc.

MyProxy Logon currently supports two approaches to this initial bootstrapping. The first is to use an existing SASL mechanism that supports mutual authentication, such as Kerberos, for the bootstrap operation, leveraging existing site authentication infrastructure. The second is to distribute a trust root for the MyProxy service with the MyProxy client software distribution, recognizing that we trust this software distribution in any case not to capture passwords or otherwise misuse credentials. We have also prototyped the baby-duck approach and are considering it as a lighter-weight alternative.

3.1.2 Site Authentication

The MyProxy service can be configured to allow users to logon with existing site credentials, using Pluggable Authentication Modules (PAM) and/or the Simple Authentication and Security Layer (SASL). Through these mechanisms, users are not required to remember another username and password for the MyProxy service.

Unix/Linux vendors support many PAM modules, including Unix password, One-Time Password, Radius, Kerberos and LDAP. We have successfully tested our MyProxy PAM interface with Radius (and One-Time Passwords), Kerberos and LDAP. PAM also supports access control and monitoring modules to implement standard security policies across multiple services.

PAM authentication is based on user interaction, typically through one or more password prompts. In contrast, SASL provides a flexible protocol framework for supporting multiple authentication mechanisms. The primary SASL mechanism used by MyProxy is GSSAPI, which allows users to authenticate with a Kerberos ticket to obtain their X.509 credentials from MyProxy.

3.1.3 MyProxy Certificate Authority

For users that don't already have X.509 credentials to store in the MyProxy repository, the administrator can configure MyProxy to act as an online CA to issue certificates in real time based on site authentication. The administrator must provide a mapping of authenticated usernames to certificate subjects, either in a configuration file or through LDAP. The user authenticates via MyProxy Logon to the MyProxy service, and MyProxy issues a certificate to the user with the subject provided in the mapping file.

By leveraging existing site authentication infrastructure through PAM and SASL, the MyProxy CA provides a lightweight mechanism for sites to distribute X.509 credentials.

3.2 GridShib: X.509 and SAML Integration

GridShib is a software product that allows for interoperability between the Globus Toolkit and Shibboleth. The complete software package consists of two plug-ins: one for the Globus Toolkit (GT) and another for Shibboleth. With both plug-ins installed and configured, a GT Grid Service Provider may securely request user attributes from a Shibboleth Identity Provider. In this section, we briefly describe both software plug-ins

and then describe the profile by which they operate in greater depth.

3.2.1 GridShib for Globus Toolkit

GridShib for Globus Toolkit is a plug-in for Globus Toolkit 4.0. Its primary purpose is to obtain attributes about a requesting user from a Shibboleth attribute authority (AA) and make an access control decision based on those attributes. The plug-in implements a policy decision point (PDP) based on attributes obtained from the AA. A policy information point (PIP) does the actual work of requesting attributes. The separation between PIP and PDP allows the plug-in to be used in flexible ways within the toolkit's authorization framework.

3.2.2 GridShib for Shibboleth

GridShib for Shibboleth is a name mapping plug-in for a Shibboleth 1.3 identity provider. Its main purpose is to allow the servicing of attribute queries from Grid SPs based on the user's X.509 Subject distinguished name (DN). The plug-in allows the attribute authority to map the user's DN to a local principal name. Upon receiving an attribute query, the Shibboleth attribute authority uses this plug-in to map the DN and utilizes the resulting principal name to resolve attributes.

The name mapping is a memory-bound collection of name-value pairs. The name (key) is a canonicalized DN that conforms to RFC 2253 [41]. The value is the local principal name. The collection is initialized when the Identity Provider starts up. The current implementation of the name mapping construct is file-based, that is, the mapping entries are read from an ordinary text file. This text file is similar to the grid-mapfile used by Globus Toolkit.

3.2.3 GridShib Profile

The GridShib Profile is an extension of the Shibboleth Attribute Exchange Profile [2]. The primary difference is the use of X.500 distinguished names (DNs) to identify principals.

The GridShib Profile is designed for a standalone attribute requester, that is, an attribute requester that does not participate in a Shibboleth browser profile. Consequently, the Grid SP does not have access to an opaque handle typically issued by the IdP on the front end of the browser profile. In lieu of a handle, the Grid SP uses the DN obtained from the client's proxy certificate.

The primary use case we consider here is a Grid Client that already possesses an X.509 end entity certificate (EEC). As is often the case in grid-based scenarios, the established user uses their EEC to generate a proxy certificate as part of single sign-on. The proxy certificate is subsequently used to authenticate to Grid SPs as part of the act of requesting service.

We therefore make the following assumptions:

- The Grid Client and the Grid Service Provider (SP) each possess an X.509 credential.
- The Grid Client has an account with a Shibboleth Identity Provider (IdP).
- The IdP is able to map the Grid Client's X.509 Subject DN to one and only one user in its security domain.
- The IdP and the Grid SP each have been assigned a globally unique identifier called a *providerId*.
- The Grid SP and the IdP rely on the same metadata format and exchange this metadata out-of-band.

The GridShib protocol flow, depicted in Figure 1, consists of the following four (4) steps.

Step 1 is the beginning of a normal grid request/response cycle. As usual, the Grid

Client authenticates using their X.509 credentials to the Grid service provider. The Grid SP authenticates the request and extracts the client's DN from the credentials.

At step 2, the Grid SP formulates a SAML attribute query whose `NameIdentifier` element is the DN extracted from the client's certificate in step 1. The Grid SP uses its X.509 credential to authenticate to the AA.

At step 3, the IdP, or more specifically the attribute authority component of the IdP, authenticates the attribute request, maps the DN to a local principal name using the plug-in described earlier, retrieves the requested attributes for the user (suitably filtered by normal Shibboleth attribute release policies), formulates an attribute assertion, and sends the assertion to the Grid SP.

Finally, at step 4, the Grid SP parses the attribute assertion, caches the attributes, makes an access control decision, processes the client request (assuming access is granted) and returns a response to the Grid Client.

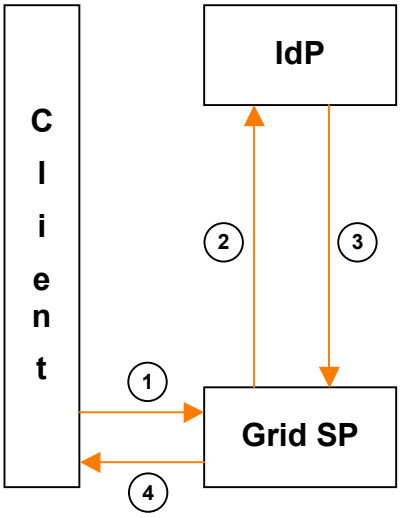


Figure 1 GridShib Protocol Flow

Both the IdP and the Grid SP rely on SAML 2.0 metadata [3,45] for their trust configuration (i.e., the certificates and public keys of the other entity). GridShib for Shibboleth supports a framework for consuming Grid SP metadata whereby the metadata file includes an `EntityDescriptor` element for each Grid SP that the IdP trusts. SAML 2.0 does not define a role for Grid SPs, however, so an extended role of type **AttributeRequesterDescriptorType** has been specified [35] for use with this profile. The defined role of each such entity is basically that of a standalone attribute requester.

3.2.4 GridShib Software

Beta software that implements the GridShib Profile is available for download from the GridShib web site [12]. Source code is available, licensed under the Apache License, Version 2.0.

3.2.5 Current Implementation Limitations

While we believe our current implementation to be sound from a security perspective, the following administrative limitations are recognized:

- The file-based name mapping doesn't scale. The fact that the DN-principal name pairs are read from a file is a major concern. Even if we were to provide administrative tools to manage the name mapping files, the overhead associated with this maintenance would be prohibitive for large user communities. Clearly, this overhead must be eliminated or at least reduced.
- IdP discovery must be generalized. In step 1 of the flow, we assume that a single IdP can assert attributes for all Grid Clients making requests of a Grid Service. A mechanism to allow a mapping between a user and their preferred IdP is needed.
- Metadata production and distribution needs to be automated or simplified.

Trust in a GridShib deployment is based on a bilateral arrangement between IdP and Grid SP. By virtue of the fact that the two entities exchange and consume each other's metadata, a trust relationship is established. The problem is that n entities give rise to $O(n^2)$ bilateral relationships, which does not scale well.

3.3 Globus Toolkit Authorization Framework

As the Globus Toolkit is used by many different projects and by many different Grid communities, it is clear that it cannot mandate the use of particular technologies and mechanisms. Specifically in the area of attributes and authorization policies, the toolkit has to be very flexible to accommodate local preferences regarding assertion formats and usage patterns.

This section enumerates the many certificate and assertion mechanisms that the toolkit has to support. It also describes an attribute collection and authorization framework that deals with the different mechanisms in a consistent manner and that is able to combine authorization decisions from many different sources to yield a single access decision for the invocation request.

3.3.1 Attribute Collection

When a client invokes a request to a service, that service may have to consider many different identity and attribute formats, like X.509 end entity certificates, X.509 attribute certificates, SAML attribute assertions, LDAP attributes, Handle System [14] attributes, and configuration properties.

As it is very common that client requests are made on behalf of other parties, some of the attribute values do not necessarily apply to the requester, but rather to other entities in the delegation chain.

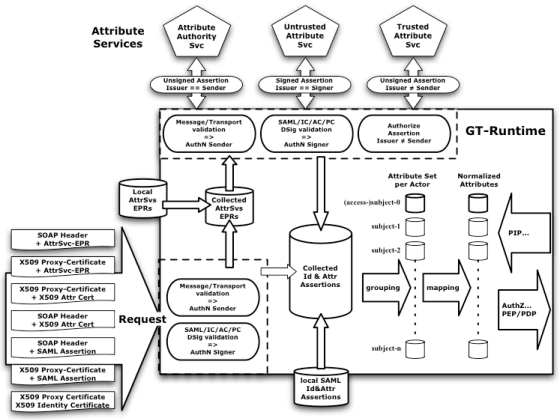


Figure 2. Attribute Collection Framework

Furthermore, the attributes can arrive at the service in a number of different ways. Some attributes are "pushed" by the requester, as in VOMS [8] or CAS [32], where the assertion bundle is included with the client request.

Other attributes are "pulled" by the service from attribute services like LDAP, SAML-compatible services like the Shibboleth Attribute Authority, or the Handle System. Note that each of the pull mechanisms uses different protocols.

Lastly, attributes can also be locally stored in (configuration) files on the service side.

The validation of the attribute binding is also dependent on the assertion format and how the information was received. Some attribute bindings are asserted through public key signatures, while others are received unsigned but embedded in protected messages or received over authenticated channels.

Finally, the attribute names and values have to be considered within the context of their definition as well as the context of the issuer. Besides the vocabulary, semantics, and ontology that apply to the attribute bindings, it is also important to understand clearly whether the assertion is only valid in the local context of the issuer or in a global

context that requires additional authorization during the validation process.

In order to manage the attribute collection in a consistent manner, the Globus team is in the process of developing a framework depicted in Figure 2. Its purpose is to accept and validate the various attribute assertion formats and mechanisms, to group all the attributes that apply to the same entity together, to translate the names and values into a single format, and finally to make the attribute collections available to the subsequent authorization decision processing phase.

3.3.2 Authorization Mechanisms

As was the case for attribute collection, the processing of the authorization policy enforcement is a similar challenge because of the fact that many formats and mechanisms have to be supported. The applicable authorization policy can come from many different sources, like the resource owner, the resource domain, the requester, the requester's domain, the virtual organization, or intermediaries.

Authorization decisions can be evaluated within the same hosting environment as the policy enforcement point, or can be evaluated by external authorization services. External policy decision points (PDPs), like PERMIS [46], are accessed through the SAML 1.1 authorization query protocol or by using the SAML 2.0 Profile of XACML v2.0 [9].

We have the common delegation-of-rights scenario where one subject can empower others to work on her behalf through the issuing of policy statements. As a consequence, there can be multiple policies and decisions that have to be combined to yield a single decision about the access rights of the requester.

The requester can push some of these policy statements or decisions as authorization assertions, which have to be evaluated by the resource owner. Proxy certificates are simple examples of such authorization assertions. CAS uses SAML authorization decision assertions that are either embedded in proxy certificates or communicated in the SOAP header.

There are many different mechanisms and languages used to express authorization policies, like grid-mapfiles, proxy certificates, SAML authorization decision assertions, CAS policy rules, XACML policy statements, PERMIS policies, and simple ACLs. Note that previously collected identity and attribute values have to be available for the authorization policy evaluations.

3.3.3 Authorization Decision Evaluation

After all the attributes and authorization assertions are collected, and internal and external authorization services are identified, the authorization decision for the access request can be determined.

In order to be able to deal with different authorization mechanisms, the authorization framework uses a PDP abstraction having the same semantics as the one defined in XACML, requiring that each authorization mechanism provides a PDP interface to the framework, each having its own custom decision evaluator that understands the intrinsic semantics of the policy expressions. The PDP abstraction allows the framework to use a common interface to interact with the different mechanism-specific authorization decision evaluators, keeping the mechanism-specific evaluations encapsulated. This common interface is mimicked after the XACML request context interface, which essentially presents the decision request as a collection of attribute values for the subject, resource and action. The PDP's evaluated decision result can

have the values of *permit*, *deny* or *not-applicable*. Note that the PDP's decision is associated with either the issuer of the policies that were evaluated or with the identity associated with an (external) authorization service.

For each received authorization assertion and for each authorization service, a mechanism-specific PDP instance is created. As each of those PDP instances is queried through the same interface to evaluate authorization decisions, the mechanism-specific details are all hidden behind the abstraction.

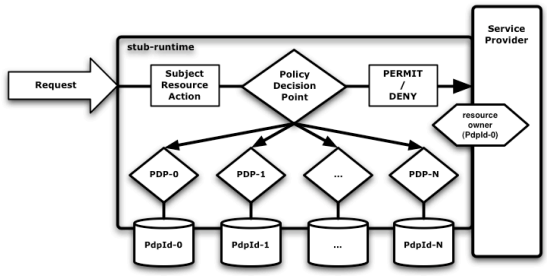


Figure 3. Authorization Framework with PDP Abstraction of Authorization Mechanisms

As shown in Figure 3, a separate *Master PDP* abstraction is used to combine all the different decisions from the various PDP instances in such a way that a single decision reflects the overall evaluated policy. In essence, this Master PDP queries the different PDP instances about the access rights of the requester and potential delegates, and searches for valid delegation decision chains that originate from the resource owner's policy and end with a statement that speaks to the access rights of the requester. The existence of such a valid delegation chain essentially states that the expressed delegation is allowed.

Note that through the use of PDP abstractions, the framework is able to evaluate decisions about delegated access rights for the requester, without the need for explicit support of delegation in the policy languages used in the authorization mechanisms.

3.3.4 Current and Future GT Support

The currently shipping GT 4.0 implementation includes a simplified version of the described attribute collection and authorization framework, but does not fully support attribute-based authorization and has no support for fine-grained delegation of rights. It includes support for proxy certificate delegation, call-out support to SAML 1.1-compliant authorization services, grid-mapfile authorization, and an XACML evaluator.

Enhancements to support Shibboleth and SAML attribute assertions have been added as part of the GridShib effort, and are included in the GridShib beta release.

The full-featured authorization framework is under active development, has produced a number of prototypes, and will ship with our next major release GT 4.2.

4 Next Steps

In this section we discuss our plans for work in the forthcoming year for enabling the seamless integration of Shibboleth/SAML and Grid Security/X.509.

4.1 GridShib

The limitations noted in the previous sections are being addressed. First of all, the file-based name mapping system will be augmented with a database implementation. This will not solve the maintenance problem, but it will make it easier to provide administrative tools. A database implementation will also facilitate the load-balancing of IdPs. (Load-balancing a cluster of IdPs is an ongoing issue in the Shibboleth Project. We do not want to exacerbate this problem.)

One approach to the IdP discovery problem is to include the IdP providerId in the user's X.509 certificate itself. Thus we are planning a modification to MyProxy that produces certificates containing this information. For this to work, we assume initially that MyProxy resides in the same security domain as the IdP. Further work will attempt to relax this restriction.

As mentioned earlier, metadata is an important aspect of GridShib (or any federated identity management system, for that matter). Therefore the following enhancements are being considered:

- provision attribute release policies (ARPs) from Grid SP metadata;
- consume IdP metadata and provision Grid SP configuration; and
- produce SP metadata from the underlying Grid SP configuration.

On the IdP side, tools to produce and consume metadata are being designed. In particular, a tool to automatically produce IdP metadata would be very helpful. (Other projects such as MAMS [24] are working on ARP tools that could take advantage of the attribute requirements called out in SP metadata.) Similar tools for the Grid SP are being developed.

Testing a classic, browser-based Shibboleth deployment remains a challenge. Testing GridShib on top of Shibboleth is even more difficult. To address this problem, we provide a command-line testing tool that tests both a Shibboleth AA and a GridShib AA. A discriminating test strategy is being built around this tool.

To further simplify testing, centralized test services will be deployed. For example, we hope to stand up an on-line GridShib IdP that new Grid SP deployments can leverage for testing purposes.

4.2 Need for Name Binding

In the simplest case, access to a grid service is managed by providing all users with an X.509 end entity certificate (EEC) from a recognized CA, mapping the names in these EECs to another namespace local to the grid service, and using these local names in access control lists. GridShib provides a means of augmenting this approach to identity-based access control with an attribute-based capability: attributes bound to the *distinguished name* in the EEC are marshaled using Shibboleth and filtered through an access control policy to determine access to the grid service.

To broaden the availability of the grid service to more users, additional naming authorities may be recognized. In particular, we wish to enable use of established naming authorities, such as those local to a user's home organization, and authentication tokens other than X.509 EECs. However, we are constrained by the requirement that an EEC must be presented to the grid service, and that only attributes correlated with the distinguished name in that EEC can be marshaled.

This presents two problems. One is the exchange of an original authentication token for a suitable EEC to be presented to the grid service, which is treated elsewhere in this article. The other is mapping the distinguished name in this EEC to the name in the original authentication token, called the *principal name*, so that attributes bound to the principal name can be marshaled by the grid service. Because the principal namespace is not local to the grid service, and to support pseudonymous access scenarios, we propose to collocate this distinguished name to principal name mapping function with the authority for the principal namespace and the attributes that are bound to principal names. This will replace the grid-mapfile associated with the

Shibboleth IdP in the initial GridShib beta product and will also support dynamic binding of principal names to distinguished names in EECs in a manner that enables the Shibboleth AA to map the distinguished name back to its principal name, enabling it to provide attributes for that principal.

4.3 Direct Client-server Use Case

There are two distinct but equally important scenarios in which this name binding must take place. In the first scenario, which we discuss in this section, the client application communicates directly with the service. The second scenario, which we discuss in the next section, involves a web portal intermediary.

When the client application and service communicate directly, end-to-end X.509 authentication is performed as part of the protocol (which is either based on TLS or SOAP with message-level security based on WS-Security [27]). The difficulty in this case is binding the identifier in the user's X.509 credential back to the principal name so that attributes may be obtained.

In this case, we believe that the online CA functionality in MyProxy (described in section 3.1) can be used to solve this problem. As shown in Figure 4, the user obtains short-lived X.509 credentials initially by authenticating to the MyProxy online CA using their principal name and password.[3] The MyProxy CA would then issue the X.509 credential, embedding into it the user's principal name. The service would then extract the principal name and use it when communicating back to the Shibboleth Attribute Authority.

[3] We use "password" here generically to indicate a static or one-time password, Kerberos credential, or any shared secret.

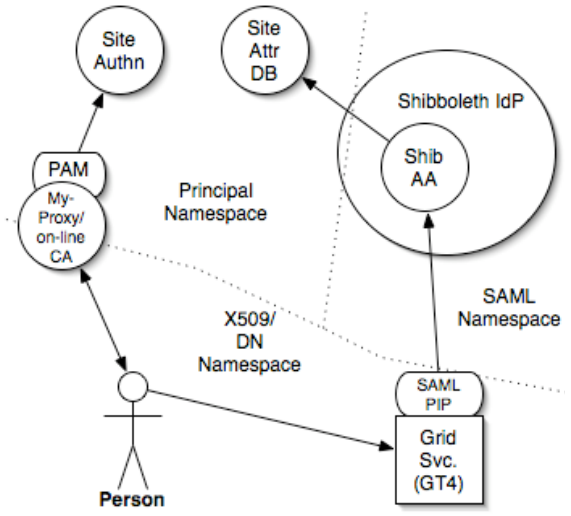


Figure 4: Different namespaces involved in an integrated MyProxy/Grid Service/Shibboleth transaction. The principal name used for authentication (at left) must be transmitted and used for attribute retrieval (upper right).

We note that this approach has a distinct advantage over the current implementation in that the Shibboleth AA does not need to maintain a DN-to-principal name mapping since the principal name is in the SAML query.

One approach is to use CryptoShibHandle [5], a modified Shibboleth handle that encrypts the principal name (along with a nonce and expiration time) into the handle itself. Encryption relies on a symmetric key shared with the Shibboleth Attribute Authority. Used in combination with a non-identifying X.509 DN, CryptoShibHandle preserves privacy by concealing user identity from the Grid service.

An open issue is the appropriate mechanism for embedding the principal name into the X.509 certificate. Current options being considered are to use the Subject Alternate Name or the Subject Information Access extension (sections 4.2.17 and 4.2.2.2 of [16] respectively). One could also embed the principal name into the DN itself (in fact the LionShare security profile [20] specifies precisely this), however we are concerned

about placing requirements on the contents of the DN.

We also note that it would be desirable to embed the providerId of the Shibboleth Attribute Authority in the proxy certificate, allowing the Grid service to easily locate the Attribute Authority. This solves the IdP discovery problem discussed earlier

4.4 Portal Use Case

The other use case mentioned in the previous section involves the client using a web browser to access a web server, which in turn accesses Grid services on behalf of the client. This use case is becoming more common as a means to allow for easy access to Grid services with a minimal footprint installation on the client system.

The primary observation in this case is that the portal effectively functions as a "chasm" that must be bridged. Either X.509 or Shibboleth/SAML can be used to authenticate to the portal, but neither has a delegation method that allows for the delegation of authority from the user of a web browser to a portal (see, however, recent work of Cantor [4]). This is the so-called *n*-tier problem ($n > 2$), an active research area.

We note that MyProxy has been used traditionally in the Grid community to enable a portal to use a client's username and password to obtain X.509 credentials for the client. Recent work [23] has also shown that this can be extended to web single sign-on using PubCookie [33]. We believe this approach can be adapted to allow Shibboleth-issued SAML authentication assertions to be used to obtain X.509 credentials from MyProxy[4].

As in the previous section, these X.509 credentials would have the principal name, taken from the `NameIdentifier` element in the SAML assertion, embedded in them. This would allow the Grid service to query the SAML Attribute Authority in an identical manner as described previously.

5 Conclusions

We have presented recent results from the GridShib and MyProxy projects. The goal of both projects is to ease PKI deployment costs by leveraging existing site infrastructure for the establishment of multi-domain PKIs to facilitate policy enforcement.

6 Acknowledgments

The GridShib work is funded by the NSF National Middleware Initiative (NMI awards 0438424 and 0438385). Opinions and recommendations in this paper are those of the authors and do not necessarily reflect the views of NSF.

The MyProxy work was funded by the NSF NMI Grids Center and the NCSA NSF Core awards. The online CA work was implemented at LBNL.

We thank the Internet2 Shibboleth development team for their continued cooperation.

7 References

1. Basney, J., Humphrey, M., and Welch, V. "The MyProxy Online Credential Repository," Software: Practice and

[4] The newly formed "ShibGrid" projects, ShibGrid and SHEBANGS, sponsored by the UK Joint Information Systems Committee has similar goals and we expect to collaborate on or leverage their work in this area.

Experience, Volume 35, Issue 9, July 2005, pages 801-816.
2. Cantor, S. et al., Shibboleth Architecture: Protocols and Profiles. Internet2-MACE, 10 September 2005. Document ID internet2-mace-shibboleth-arch-protocols-200509 http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-latest.pdf
3. Cantor, S. et al., Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS SSTC, 15 March 2005. Document ID saml-metadata-2.0-os http://www.oasis-open.org/committees/security/
4. Cantor, S. SAML 2.0 Single Sign-On with Constrained Delegation. Working Draft 01, 1 October 2005. Document ID draft-cantor-saml-sso-delegation-01 http://shibboleth.internet2.edu/docs/draft-cantor-saml-sso-delegation-01.pdf
5. CryptoShibHandle https://authdev.it.ohio-state.edu/twiki/bin/view/Shibboleth/CryptoShibHandle
6. DOEGrids Certificate Service, http://www.doegrids.org/
7. Enabling Grids for E-sciencE (EGEE), http://public.eu-egee.org
8. EU DataGrid, VOMS Architecture v1.1. 2003. http://grid-auth.infn.it/docs/VOMS-v1_1.pdf.
9. Anderson, A. and Lockhart, H. SAML 2.0 Profile of XACML v2.0. OASIS Standard, 1 February 2005. Document id: access_control-xacml-2.0-saml-profile-spec-os
10. Foster, I. Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005.
11. Foster, I., and Kesselman, C. (eds.). *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004.
12. GridShib: A Policy Controlled Attribute Framework http://gridshib.globus.org/
13. Gutmann, P. Plug-and-play PKI: A PKI your Mother can use. Presentation given at the 12th USENIX Security Symposium, Washington, 2003.
14. The Handle System, http://www.handle.net/, 2005.
15. Hodges, J. et al. Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005.
16. Housley, R., Polk, W., Ford, W., and Solo, D., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, IETF, April 2002.
17. Hughes, J. et al. Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1. OASIS, May 2004.
18. International Grid Trust Federation, http://www.gridpma.org/, 2005.
19. Internet2 Middleware Architecture Committee for Education (MACE) http://middleware.internet2.edu/MACE/
20. The LionShare Project http://lionshare.its.psu.edu/main/
21. Maler, E. et al., Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1. OASIS, September 2003.
22. Maler, E. et al., Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V1.1. OASIS, September 2003.
23. Martin, J., Basney, J., and Humphrey, M. Extending Existing Campus Trust Relationships to the Grid through the Integration of Pubcookie and MyProxy. 2005 International Conference on Computational Science (ICCS 2005), May 22-25, 2005. Emory University, Atlanta, GA, USA.
24. Meta-Access Management System (MAMS) http://web.melcoe.mq.edu.au/projects/MAMS/
25. MyProxy Credential Management Service http://grid.ncsa.uiuc.edu/myproxy/
26. Myers, M. et al. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP). RFC 2560, IETF, 1999.
27. Nadalin, A., et. al., Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), March 2004.
28. 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 Computer Society Press, 2001.
29. OASIS Security Services (SAML) TC http://www.oasis-open.org/committees/security/
30. OASIS Web Services Resource Framework (WSRF) TC http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
31. OpenScienceGrid, http://www.opensciencegrid.org
32. Pearlman, L., Welch, V., Foster, I., Kesselman, C. and Tuecke, S., A Community Authorization Service for Group Collaboration. IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.
33. Pubcookie: open-source software for intra-institutional web authentication http://www.pubcookie.org/
34. Scavo, T. et al., Shibboleth Architecture: Technical Overview. Internet2-MACE, 8 June 2005.
35. Scavo, T. et al., SAML Metadata Extension for a Standalone Attribute Requester. Committee Draft 01, 11 April 2005.
36. Scientific Discovery through Advanced Computing (SciDAC), http://www.scidac.org, 2001.
37. The Shibboleth Project http://shibboleth.internet2.edu/
38. Skow, D., Use of Kerberos-Issued Certificates at Fermilab. GGF-15 Community Activity: Leveraging Site Infrastructure for Multi-Site Grids. October 3, 2005. http://www.ggf.org/GGF15/presentations/DDS_20051003_kca.ppt
39. TeraGrid Project, http://www.teragrid.org, 2005.
40. Tuecke, S., Welch, V. Engert, D., Pearlman, L., and Thompson, M., Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, RFC 3820, IETF, June 2004.
41. Wahl, M., Kille, S., Howes, T., Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names, IETF, December 1997. http://www.ietf.org/rfc/rfc2253.txt
42. 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. Proceedings of the 3rd Annual PKI R&D Workshop, 2004. http://middleware.internet2.edu/pki04/proceedings/proxy_certs.pdf
43. Welch, V., Barton, T., Keahey, K., Siebenlist, F., Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration, Proceedings of the 4th Annual PKI R&D Workshop, 2005.
44. 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 Twelfth International Symposium on High Performance Distributed Computing (HPDC-12). IEEE Computer Society Press, 2003.
45. Whitehead, G. and Cantor, S., Metadata Profile for the OASIS Security Assertion Markup Language (SAML) V1.x, Committee Draft 01, 15 March 2005.
46. D.W. Chadwick and A. Otenko. The PERMIS X.509 role based privilege management infrastructure. Future Generation Computer Systems, 19(2):277-289, February 2003.

PKI Interoperability by an Independent, Trusted Validation Authority

Jon Ølnes

DNV Research, Veritasveien 1, N-1322 Høvik, Norway
jon.olnes@dnv.com

Abstract. Interoperability between PKIs (Public Key Infrastructure) is a major issue in several electronic commerce scenarios. This paper suggests an approach based on a trust model where an independent Validation Authority (VA) replaces Certification Authorities (CA) as the trust anchor for the receiver of a PKI certificate (the Relying Party, RP). By trusting the VA, the RP is able to trust all CAs that the VA can answer for. The main issue is not technical validation of the certificates but assessment of quality, trustworthiness and risk related to certificate acceptance. The RP obtains a one-stop shopping service – one point of trust, one agreement, one bill, one liable actor. As an additional benefit, the need for certificate path discovery and validation may disappear.

1. Introduction

Public key cryptography used with a PKI (Public Key Infrastructure) carries the promise of authentication, electronic signatures and encryption based on sharing of only non-secret information (public keys, names and other information in certificates[1]). The same information (the certificate) may be shared with all counterparts, to replace separate, shared secrets.

The requirements on a counterpart (RP for Relying Party – relying on certificates) are that it must be able to validate the authenticity and integrity of the certificate and interpret the certificate's content. The RP also needs to assess the risk related to acceptance of the certificate, determined by the quality of the certificate, the trustworthiness of the issuer (the CA – Certification Authority), the liabilities taken on by the CA, and the possibilities for claiming liability in case of mistakes by the CA; all related to the security and business requirements of the operation in question.

In this picture, PKI interoperability is an important issue. An RP may need to accept certificates from a large number of PKIs. Consider DNV as an example: DNV is an international company with customers and partners in more than 100 countries all over the world. As an RP, DNV must be able to assess the risk related to acceptance of certificates from in most cases several CAs per country. In our work on the interoperability problem, DNV has concluded that a different approach is best suited to address these concerns, where interoperability is offered by means of an independent Validation Authority (VA).

The idea of a VA is not new, but in our approach, the VA replaces CA(s) as the trust anchor for the RP. In common PKI practice, the trust model is reversed: a VA is delegated trust from the CAs it handles, and only CAs may be directly trusted.

In our trust model, it is important that the VA is neutral with respect to CAs, i.e. the VA service must be offered by an independent actor. A VA should be able to answer for validity, quality and liability related to certificates issued by "any" CA, thus providing RPs with the necessary information for their risk assessment. The requirement for independence with respect to CAs particularly applies for quality classification. VA services may additionally cover verification of signed documents (not only certificates) and may be extended to notary (trusted storage) and various related services [27].

A VA service may be general ("one size fits all") or customisable. Customisation may consist of defined quality profiles per RP and/or explicit specification of criteria (e.g. nationality) for CAs that shall be trusted or not by the specific RP.

In the following, we clarify DNV's position in 2, describe requirements in 3, review existing approaches

[1] Another term is "electronic ID". A PKI-based electronic ID usually consists of two or three certificates and corresponding key pairs, separating out the encryption (key negotiation) function and possibly also the electronic signature (non-repudiation) function to separate key pairs/ certificates. To a user, this separation is normally not visible. This paper uses the term "certificate", to be interpreted as covering the electronic ID term where appropriate.

in 4, describe the independent VA in 5, and look closer on the commercial and legal issues for a VA in 6. We conclude in 7.

2. DNV's Position and Role

DNV (Det Norske Veritas, http://www.dnv.com) is an independent foundation offering classification and certification services from offices in more than 100 countries. The maritime sector and the oil and gas industry are the main markets. DNV is also among the world's leading certification bodies for management systems (ISO 9000, ISO 14000, BS 7799 and others), delivering services to all market sectors.

DNV seeks to extend its existing position as a supplier of trusted third party services to digital communication and service provisioning. The first version of a VA service along the lines described in this paper will be offered to pilot customers mid-2006. This paper does not describe this pilot service but rather the research leading to the decision to launch the pilot service.

3. Requirements for Interoperability

3.1 The PKI Interoperability Challenge

The PKI interoperability challenge can be described from two viewpoints:

- A certificate holder should be able to use the certificate towards all relevant counterparts, regardless of the PKI used by the counterpart.
- An RP should be able to use and validate certificates from all relevant certificate holders, regardless of the PKI used by the certificate holder.

The word "relevant" is the key to the severity of the interoperability challenge. In many cases, the set of relevant counterparts is limited by such criteria as nationality, business area, application area (e.g. banking) or any other criteria that an actor may find relevant. CAs may also put restrictions on use of certificates. Note however:

- Unlimited interoperability may be viewed as the ultimate goal, likened to the ability to make phone calls internationally.
- A service provider as an RP may want to accept certificates from as many CAs as possible, in order to reach as many customers as possible.
- A certificate holder may want to use one certificate for "any" service internationally.
- When a digitally signed document is created, the parties involved may be able to identify the relevant CAs. However, the document may need to be verified later by another actor, who may not have any relationship to any of these CAs.

Service providers as RPs may want to solve this situation unilaterally by requiring use of a certain PKI by its counterparts. This may be unacceptable to a counterpart (be that an individual customer or a business partner) that already has a certificate, and that does not want to acquire another one (or several more if different RPs pose such requirements).

3.2 PKI Deployment and International Aspects

PKIs are deployed in various contexts: Society infrastructures for the general public (individuals, but also for businesses), corporate infrastructures (business internal), and community infrastructures (for particular purposes, e.g. banking). Interoperability is relevant where communication requires use of certificates across infrastructures.

PKIs as society infrastructures are being deployed in probably most developed countries for national electronic IDs. Society infrastructures cover at least individual citizens but may also cover businesses and individuals in the role of employees. The infrastructures are either based on PKIs run by public authorities or on services obtained from the commercial market. Society infrastructures are almost exclusively national, although some international coordination takes place. Notably, the EU Directive on electronic signatures [9] defines the concepts of qualified signatures/certificates as means to achieve legal harmonisation across the EU in this area.

Even in countries with (plans for) public authority PKIs, the usual situation is several (2-15 is typical for European countries) public, commercial CAs competing in a national market. While PKI interoperability thus may be a challenge even at a national level, the scaling may be manageable. However, interoperability at an international level remains a severe challenge.

The topic is on the agenda. In Europe, interoperability of certificates and electronic signatures is identified as a key issue in creating an internal market[2] in the EU. One example is the IDABC (Interoperable Delivery of European E-government Services to Public Administrations, Businesses and Citizens) programme's statement on electronic public procurement [5]: "The interoperability problems detected [for qualified electronic signatures] despite the existence of standards, and the absence of a mature European market for this type of signatures pose a real and possibly persistent obstacle to cross-border e-procurement." Other examples can be found.

[2] Coined as "the SEEM" (Single European Electronic Market) in EU terms.

Internationally oriented businesses face the same challenges. Mandatory requirements for signatures are rare in the private sector but businesses can benefit a lot from electronic signatures and PKI-based authentication. In an increasingly global society, restricting these mechanisms to a national level is too narrow. Solutions are being developed for particular commercial sectors, such as the SAFE Bridge-CA for the pharmaceutical industry [20]. The SAFE initiative shows that groups of actors may manage to work together towards interoperability in international communities.

However, in general the interoperability problem remains an issue. If not solved otherwise, the problem is left to the individual RP, but an RP acting by itself has a challenge handling the problem with confidence, i.e. with definable risk. This paper suggests VA services as a promising approach at solving the interoperability problem.

3.3 The Challenges to the RP

The interoperability challenges are best described from the viewpoint of an RP. With respect to a certificate, the RP must perform:

- Parsing and syntax checking of the certificate and its contents, including some semantic checking like use of certificate compared to allowed use (key usage settings) and presence of mandatory fields and critical extensions.
- Assessment of the risk implied by accepting the certificate, determined by the CA's trustworthiness, the quality of the certificate, and the liability situation, relative to the operation in question.
- Validation of the CA's signature on the certificate. This requires a trusted copy of the CA's own public key, either directly available, or obtained from further certificates in a certificate path (see 4.1).
- A check that the certificate is within its validity period, given by timestamps in the certificate. For real-time checking, this must be compared against the current time. For old, signed documents, it is the time of signing that is of interest.
- A check that the certificate is not revoked, i.e. declared invalid by the CA before the end of the validity period. For real-time checking, the current revocation status is checked. For old, signed documents, status at the time of signing is checked.
- Semantic processing of the certificate content, extracting information that shall be used either for presentation in a user interface or as parameters for further processing by programs. The name (or names) in the certificate and interpretation of naming attributes are particularly important.
- In the case of certificate paths, this processing must be repeated for each certificate in the path (see 4.1).

Syntactic parsing and checking of validity period are usually straightforward operations. All other steps in the certificate processing more or less have problems related to scaling, i.e. handling of certificates from a high number of CAs.

Management of information about CAs and their services (trustworthiness, quality of certificates, liability, possibility of enforcing liability, and trusted copy of public key) gets increasingly difficult with the number of CAs. The liability situation can in general only be safely assessed through agreements, but it would be difficult for an RP to have explicit agreements with all relevant CAs. A consortium of RPs, e.g. in an industry sector, may be able to find approaches to diminish the problem.

The X.509v3 standard [16] defines syntax of certificates, but leaves many options, and only partly defines semantics of fields, attributes and extensions. Even though recommended profiles for X.509 certificates exist, certificates from different CAs often differ in content. This particularly applies to naming of subjects. An RP must either be able to use (parts of) names in a certificate directly for identification, or a name in a certificate must be reliably translated to a derived name that is useful to the RP. The security/quality of the translation process must preserve the quality of the certificate, i.e. the confidence in the derived name must be as if the derived name had been included in the certificate.

3.4 Legal Issues and Risk

An RP must not only be able to validate a certificate, but also be able to assess the risk involved in accepting the certificate for a given purpose. This raises legal and commercial concerns.

A question which an RP always faces is to know with confidence the liability taken on by the CA, and what recourse the RP has if the CA fails to fulfil its responsibility. An unknown liability situation may constitute a serious risk. An actor offering an interoperability service should on one hand be able to take liability for its own actions (which on the commercial side means that it must have sufficient income or funding to cover the liability), and on the other hand at least provide guidance with respect to the liability taken by the CAs it covers. Preferably, the interoperability service should take on the CAs' liabilities and be able to transfer these to the responsible CA when appropriate, thus providing risk management for the RPs.

CA liability is described in certificate policies and may be governed by (national) law. Additionally,

agreements between a CA and RPs may control liability. In an international setting, certificate policies may be written in a foreign language and refer to foreign legislation with respect to the RP, and as cited above, it would be difficult for an RP to have agreements with all CAs on which it may want to rely. Thus, the RP's risk situation can be complex.

Current approaches to PKI interoperability may solve technical problems but they all have challenges on the commercial and legal side (see 4). In the context of a VA, these issues are discussed in 6.

4. Approaches to PKI Interoperability

4.1 Trust Models and Certificate Paths

Present PKI practice focuses on only CAs being trusted. Given a large number of CAs, direct trust in each of them by an RP (trust list approach, see 4.5) becomes difficult. Present approaches seek to solve the scaling problems by trust structures among the CAs: peer-CA cross-certification (mutual recognition), hierarchy, or bridge-CA. Hybrid models are possible but are not discussed in depth in this paper.

Trust structures are created by issuance of certificates to the CAs themselves; by peer-CAs, a bridge-CA, or a CA at a higher level of a hierarchy. The idea is that an RP should be able to discover and validate a certificate path from a directly trusted CA (typically the root-CA of a hierarchy) to any CA (may be previously "unknown") that is a member of the same trust structure. In this, trust is regarded as a transitive property. The number of CAs directly trusted by an RP can be reduced.

A general comment on trust structures is that certificate path discovery may be a very difficult task [24]. Sufficient support for path discovery is lacking in many PKI implementations. Also, certificate path validation may be very resource demanding due to the need for repeated certificate processing (the steps described in 3.3). Caching of previously validated trust paths can mitigate this problem.

Certificate path validation, possibly also path discovery, may be performed by a validation service (delegated path validation/discovery [25]). Note that the trust model suggested by this paper (see 5.2) eliminates certificate path processing.

"Trust" in this context mainly means the ability to find a trusted copy of a CA's public key in order to validate certificates issued by the CA. To some extent, trust models can address quality (e.g. by policy mapping) but liability is in practice still left as an issue between the RPs and the individual CAs.

4.2 Peer-CA Cross-Certification

Practical experience with peer-CA cross-certification (mutual recognition) has shown that the effort needed is very large, in particular when the CAs are competitors. The author was involved in a project where three CAs in Norway managed to establish a cross-certification regime, but repeating this effort is not recommended.

Large-scale cross-certification would create trust structures ("web of trust", similar to the trust model used by e.g. PGP) that would be particularly complex with respect to path discovery. However, the technical issues are not the most important ones.

Commercially, no CA is really interested in solutions that improve market access for its competitors. Cross-certification may be tempting in cases where both CAs can gain from an increased market. In other cases, the commercial incentive simply does not exist, and the attitude will be to refrain from cross-certification if possible, i.e. unless cross-certification is imposed by e.g. national authorities.

Cross-certification with policy mapping means that the two CAs' services are regarded as equal with respect to quality. The complexity involved in the policy mapping depends on the differences in the policies. There are a few common frameworks [4] [6] [7] for structuring of policies. Mapping between the frameworks is not too complicated, and most CAs adhere to one of the frameworks. Still, the real content of policies may differ quite a lot.

Cross-certification may imply that the CAs provide guarantees for one another, so that a customer of one CA may claim liability related to certificates issued by the other CA. This is governed by the cross-certification agreement, but competing CAs may be reluctant to enter such agreements.

On an international level, peer-CA cross-certification as a scalable solution to interoperability must be regarded as unfeasible. The main use may be in situations where the CAs are non-commercial, e.g. corporate PKIs of co-operating businesses.

4.3 Hierarchy

In a hierarchy, CAs are assembled under a common root-CA, which issues certificates to subordinate CAs. Although a hierarchy may in theory have an arbitrary number of levels, practical systems usually have two levels: root-CA and certificate issuing CAs.

Hierarchies scale well, but if an indication of quality of service of CAs shall be implied by the hierarchy, all CAs involved must have equal quality. This is usually enforced by a common base policy defined by the root-CA. As one example, all CAs for qualified certificates

approved by the German government are placed under a root-CA run by the Regulatory Authority for Telecommunications and Post [2].

However, the root-CA need not put restrictions on the CAs, as shown by the EuroPKI initiative [17]. In this case, an RP can draw few conclusions (except that the CA's public key is authentic) from the fact that the CA is a member of the hierarchy.

There is no reason to believe in a world-wide hierarchy as the solution to PKI interoperability. However, hierarchies reduce the number of CAs that must be directly trusted.

The weak point in a hierarchy is the root-CA. This part is technically simple, but legally and commercially very difficult. Although CAs may be willing to pay some amount to join a hierarchy, it is not possible to gain much income from operating a root-CA. A root-CA may run on governmental or international funding, or by a limited company jointly owned (cost and risk sharing) by the CAs beneath the root-CA. Without an income, the owner of a root-CA, even if it is a governmental agency, will be reluctant to take on much liability, and liability may remain an issue between the RP and the individual CAs in the hierarchy.

At an international level, one may devise establishment of yet another level in the form of international root-CAs on top of national root-CAs, or alternatively cross-certify between (the root-CAs of) hierarchies. Such structures will create complex certificate paths, and cross-certification between actors that do not take on liability (the root-CAs) may be a questionable approach. A better approach in this case is to use bridge-CAs to connect hierarchies.

4.4 Bridge-CA

A bridge-CA is a central hub, with which CAs cross-certify. The bridge-CA should be run by some neutral actor, and it shall itself only issue cross-certificates. An RP may always start a certificate path to a given CA by starting at its own root of trust, and then proceed to a certificate issued by its root to the bridge-CA. For hierarchies, the usual situation is cross-certification between the bridge-CA and the root-CA. Thus, complicated certificate paths may occur even when using a bridge-CA.

Cross-certification between a CA and a bridge-CA is considerably simpler than peer-CA cross-certification, as the bridge-CA has no (competing) role in issuing of certificates to end entities.

Indication of quality may be done by requiring a CA to cross-certify with the bridge-CA at the appropriate quality level. As an example, the Federal Bridge CA (FBCA) in the USA defines five policy levels [11].

The FBCA is not liable to any party unless an "express written contract" exists ([11] section 9.8). A commercial bridge-CA, such as the SAFE Bridge-CA [20], may take on more liability, but commercially a bridge-CA suffers from the same problems as the root-CA of a hierarchy: It may be difficult to get an income from issuance of cross-certificates, and liability must usually be balanced by an income. Mainly, liability remains an issue between the RP and the individual CAs.

The FBCA does not provide validation services, but test suites are defined for path discovery [23] and path validation [22] related to the FBCA. A list of products that have passed the test is found on FBCA's web site. A bridge-CA might provide directory services and VA services [19] similar to those described in this paper. We argue that with such VA services, the bridge-CA functionality is actually obsolete and the VA functionality is sufficient.

Bridge-CAs have so far either a regional scope (as USA or EU) or a defined business scope (may be international, as for the SAFE Bridge-CA), which means that there is a need to link bridge-CAs in order to achieve general, global interoperability, thus creating more complex trust models. The FBCA has defined guidelines for such cross-certification (part 3 of [10]). As argued for hierarchies, cross-certification between actors that do not take on liability (the bridge-CAs) may be a questionable approach.

4.5 Trust List Distribution

A trust list consists of named CAs and their public keys. All CAs on the list are trusted. The CA may be the root of a hierarchy, in which case all CAs in the hierarchy can be trusted. An example is the list of more than 100 CAs included in distributions of Microsoft OSs. An RP may manage a trust list entirely on its own or modify existing lists such as adding or removing CAs from Microsoft's standard list.

Trust list management may also be done by a third party, which should regularly distribute lists to its subscribers. Interoperability is achieved by installation of compatible trust lists at all actors. In Europe, the IDABC Bridge/Gateway CA (EBGCA) actually is a trust list distribution service [3] based on the study in [13][3] and ongoing work in ETSI [8]. The primary purpose of the EBGCA is to list nationally approved or registered issuers of qualified certificates but other CAs may be added. The status of the CA (such as issuer of qualified certificates) is indicated as extra

[3] This study disapproves of a VA solution to interoperability. However, in this case the VA is an OCSP service with few similarities to the VA concept presented in this paper.

quality parameters of the trust list. Quality information is a fairly straightforward extension for any trust list. An example of a simpler service is TACAR [18] for the academic sector in Europe. This is simply a repository of CA keys and policies, available for download to organisations.

The EBGCA is particular in that it defines itself as a trust anchor for the RP and takes on some liability with respect to the RP. In other cases, the CAs on the list take the trust anchor role, and liability remains an issue between the RP and the individual CA. As for quality information, liability information may in principle be distributed with the trust list; however the distribution service is unlikely to help in claiming liability.

We have not seen evaluations on the possibilities of making a trust list distribution service profitable. The subscribers will use the service only occasionally (regular but infrequent updates, or notification and download upon changes), and CAs may be reluctant to pay to get on the list. A service run by a publicly funded agency (national or international) may be an alternative. Correspondingly, a distribution service will usually be reluctant to take on much liability for its own service. RPs may download trust lists, and use them at their own risk.

5. The Independent Validation Authority

5.1 Outsourcing Certificate Validation

Certificate processing at an RP may be very resource consuming (see 3.3). This particularly applies to certificate path processing and revocation checking by use of CRLs (Certificate Revocation List [16]). A more efficient revocation checking protocol, OCSP (Online Certificate Status Protocol) [21], has been developed to enable outsourcing of the revocation checking part.

While OCSP was primarily designed for services provided by one CA, OCSP services that can answer about revocation status for certificates from several CAs are also in use. According to the OCSP specification, such a service must present a certificate from the given CA to prove that it has been delegated responsibility to answer about revocation status. Since OCSP only transfers identification of certificate and issuer, not the complete certificate, the protocol cannot be used to support outsourcing of more of the steps in the RP's certificate processing.

SCVP (Standard Certificate Validation Protocol – should be released as a "proposed Internet standard" in the near future) is developed to provide richer functionality for validation. SCVP allows the complete certificate (or even a certificate chain) to be transferred. SCVP has been severely delayed, and support for the protocol seems to be low. Delegated certificate path processing is envisaged by the PKIX (Public Key Infrastructure X.509) working group of the IETF (Internet Engineering Task Force) [25] but the complexity is troublesome [24].

The main problem in our view is that the validation authority resides with the CAs. Below, we describe the advantages of decoupling the VA role from the CAs.

5.2 Revising the Trust Model for the RP

In our view, a fundamental flaw in present PKI practice is that a CA is the only actor that can serve as a trust anchor; i.e. a trust decision must ultimately always be linked to a trusted CA. This requirement leads to the necessity for trust structures and certificate paths in order to navigate from a trusted CA to an "arbitrary" CA.

The CA as the trust anchor is the right model for a certificate holder, who selects the CA(s) to obtain certificate(s) from. However, an RP should aim at acceptance of "any" CA's certificates, regardless of its relationships to other CAs.

This paper instead suggests a trust model where an independent Validation Authority (VA) is the trust anchor for the RP. Upon trusting the VA, the RP is able to trust any CA that the VA handles. The VA handles each CA individually, regardless of any trust structure that the CA may participate in. Certificate path discovery and validation are irrelevant (although the VA may use such processing internally to aid in classification and other tasks) since there is no need to prove a path to a "trusted CA".

This trust model resembles a two-level hierarchy or use of a bridge-CA, but the VA does not issue certificates. It is an on-line service answering requests from RPs. As opposed to other interoperability services, an on-line VA may be able to run a profitable business by providing real risk management services to the RP. The RP is provided with one-stop shopping for validation of certificates: One point of trust, one agreement, one point of billing, one liable actor.

5.3 Using a VA Service for Interoperability

Given this trust model, the state of the art in VA services may be considerably advanced. The RP outsources all (or parts of, see 3.3) its certificate processing to the VA, regardless of the CA that has issued the certificate. The VA checks validity with the appropriate CA, but returns its own answer, not an answer originating from the CA. The answer includes information on quality, trustworthiness, and liability, and possibly auxiliary information derived from certificates. Such information may be other names for the certificate holder (the name in the certificate need

not in itself be useful to the RP) or further information related to certificate holder, such as age, sex, or credit check. Auxiliary information may originate from the CA as well as from other sources, and the information may be general or RP specific.

Thus, the VA acts as a clearinghouse for information about CAs and their certificates, with a possibility for further, value-added services. The main feature is support for risk management for the RPs. A VA may be provided in a "one size fits all" manner, or it may be configurable to meet requirements of individual customers (RPs). The VA does not remove the complexity of interoperability, but it handles the complexity in one place, for all RPs who have outsourced certificate processing to the VA. Internally, the VA operates a trust list of the CAs it is able to answer for.

5.4 Classification Related to VA Services

As noted, a VA shall not only return an answer about validity, but also indication of quality, trustworthiness and liability related to a certificate.

The quality of a CA's certificates is mainly derived from its certificate policy [4] [6] [7]. Trustworthiness is determined by an assessment of the actor running the CA, e.g. to confirm that the CA is able to fulfil its liability in case of errors. Other documentation may also be of relevance, such as certification practice statements and agreements with certificate holders and other actors (including membership in hierarchies and cross-certification regimes). Liability is discussed in 6 below.

The documentation must be measured against a classification system, defined as a set of quality and trustworthiness parameters, and criteria for meeting certain levels related to these parameters. In the simplest case, the resulting classification may be mediated as a number (say, classes 1-10), but it is also possible to define data structures in order to mediate a more fine grained classification with respect to the parameters. An RP may be allowed to define its requirements in the same manner (either as "at or above level x" or "according to the values in this structure"). The VA may compare the RP's requirements to the classification. The result may be a yes/no answer or a report on deviations from the desired quality profile. A particular classification is assessment of compliance with national or international legislation, e.g. that requirements for qualified certificates/signatures [6] are met.

Such a classification system resembles policy mapping for cross-certification, but the system is more flexible. The classification system rates certain characteristics of a CA and its services to obtain either an overall score or a descriptive structure, whereas a policy mapping needs to determine compliance between two policies. A classification system with just a few discrete classes may be close to a policy mapping scheme (e.g. the five levels of the FBCA), while a more fine grained classification allows CAs to differ in policies but still fit in the classification scheme. Since agreed quality levels, like qualified level in Europe and FBCA levels in the USA, are regional in scope, a flexible classification system may be important for international interoperability.

Note that the documentation only presents the quality and trustworthiness claimed by the CA. A classification must include an "evaluation assurance level" to indicate to what degree an assessment of actual operation has been done. Levels may be: self-assessment by CA (possibly augmented by acceptance of a surveillance authority such as demanded by the EU Directive on electronic signatures [9]), report from a surveillance agency or a third party auditor, and certification (such as BS7799[4] [1], ISO15408 [14], ISO9000 etc.). Classification criteria for CAs may be used to develop specific criteria for quality certification of CAs. The evaluation assurance level may be incorporated in the quality indication (higher assurance implies higher quality) or it may be mediated as a separate parameter.

DNV is among the world's leading actors in classification and certification, and work is ongoing on development of classification criteria and a classification system for CAs in conjunction with VA services. At present, we leave open the question of whether a classification system should be standardised or be left as a competitive element for a VA. In DNV's present services, classification may be based on standards (e.g. certification to ISO 9000 or similar standards) or competitive (e.g. DNV's own class rules for ships).

5.5 A Note on Openness of PKIs

A VA is based on the assumption that the CAs provide open PKIs. Our basic criterion for technical openness is that an RP should be able to use any standards-based software to process certificates and signed documents. PKI support is included in almost all platforms, and the RP should be able to base its processing on such built-in functionality (with enhancements if needed) regardless of the CA.

[4] Information security management is usually developed according to ISO/IEC 17799 [15], which is based on BS7799 part 1. However, certification is still done according to BS7799 part 2, since the certification part has not yet been approved by ISO.

This assumption is unfortunately broken by many PKIs, which require particular software to be installed at the RP in order to accept and process certificates and documents issued/signed under the PKI. Such PKIs are in effect closed in that the certificates can only be used between parties that have all installed the software. Examples are solutions that require particular Java applets or similar to be transferred from a service provider (the RP) to a certificate holder, and solutions that use proprietary protocols between certificate holder and RP and/or between RP and CA.

It is clear that such PKIs cannot properly support interoperability, since one cannot expect all possible RPs to install the software. Also, an RP (typically a service provider) cannot be expected to install such software related to more than a few PKIs. In some cases, such software (e.g. to process signed documents) may be installed at a VA instead of at the RP, but in many if not most cases the RP is stuck with the extra software. We believe that such closed solutions eventually must be changed, but in the short to medium term they will cause a major problem to interoperability.

Some CAs require explicit agreements[5] with all RPs. The CA's policy states that the CA takes no liability unless the RP has such an agreement. Large-scale interoperability cannot be achieved, as it is not possible to have agreements with every possible RP. A VA may sign a "bulk agreement" with such CAs; one agreement covering all RPs using the VA. This may solve the agreement issue, but the CA has to approve the solution (see also 6.1 below).

A VA may solve some, but not all, issues related to closed PKIs. However, an approach based on trust structures and certificate paths cannot solve any of the issues since the problems are related to processing and validation of certificates and signatures, not to path discovery and path validation.

5.6 Implementation, Performance, Availability

The technical realisation of a VA service is not a central topic of this paper. However, the following observations are made:

– A VA is an on-line trust service subject to severe requirements for availability and security. These requirements are enforced on the software and hardware used as well as on the operational environment of the service.
– A VA needs to handle the heterogeneity encountered in the PKI area, including support for various certificate profiles, cryptographic algorithms and protocols.
– For scaling, a VA must be replicated. Synchronisation between instances of the VA service and optimisation of collection of revocation information and auxiliary information must be in place.

Outsourcing certificate processing to a VA may improve performance since an optimised and dedicated installation is used at the VA. The avoidance of certificate path discovery and validation procedures greatly improves speed in cases where this would normally be needed. However, the VA solution must scale, and performance is influenced by factors like the communication link between RP and VA.

When RPs operating critical services rely on a VA, the VA's availability must be guaranteed. There are two main issues involved:

– Availability of the VA towards the RPs. This is similar to availability of other critical systems, and measures are reliable systems and communication links, redundancy, protection against DoS attacks and so on.
– Availability of updated status information from the CAs. If a CRL download or an OCSP request fails, the VA must either report an error to the RP or risk an answer based on the old, cached status information. If a CRL download is too slow, the VA may also need to answer based on old information. Optimising status information updating is very important, see 5.7.

5.7 Interfacing a VA

For the interface between an RP and a VA, today's standard validation protocol, OCSP [21], clearly has too limited functionality. The successor, SCVP, has been severely delayed, and support for the protocol seems to be low.

A better approach, in our opinion, is to provide VA services as Web Services. The XKISS part of XKMS [12] is a good starting point for the VA interface. The XML documents exchanged with the VA may in the future be subject to standardisation. In any case, a VA should publish its XML specifications in order to enable integration software produced by "anyone". The desired level of standardisation may be limited by the heterogeneity of different VA services, and by the possibility of tailoring VA services to specific customers.

For performance, a VA must optimise gathering of information from CAs (and possibly other sources for auxiliary information) and answer requests as far as

[5] This is almost always the case for PKIs that require particular software to be installed. An agreement covers both purchase of software and acceptance as an authorised RP.

possible based on information cached locally. The preferred option is CRL download, with OCSP requests to the CA as a fallback alternative. CRL download must be configurable and be done by a separate process. A polling strategy may be used in order to catch CRLs issued out of or before schedule. Delta-CRLs and CRL push mechanisms should be exploited wherever available.

All interfaces to and from a VA must be secured. The communication links should be protected by use of SSL (or similar means), and it must be possible to sign requests and responses between the RP and the VA and between the VA and CAs. Authentication of the RPs (and the VA towards the CAs) is done either when the SSL channel is established or through signatures on requests.

The RPs may be authenticated by certificates issued by their preferred CA. The VA's own certificates can either be obtained from one or several CAs (may be needed to authenticate towards CAs), or the VA may authenticate by a self-signed certificate to pinpoint its position as an independent trust anchor.

5.8 Privacy and Identity Management

Miscellaneous scenarios can be used to illustrate potential relationships between a VA and identity management services. A VA may take on the role of an Identity Provider according to the Liberty Alliance framework. In this case, the XML document produced as a response to a request will be a SAML V2.0 token including certificate information and auxiliary information. A VA may also be placed "behind" an Identity Provider, enabling the Identity Provider to outsource certificate processing. Even in this case a SAML V2.0 token may be the appropriate answer from the VA.

The VA must reliably log all actions performed, since the VA must be prepared to supply evidence in case of disputes. Disputes need not involve the VA itself; an RP involved in a dispute with a customer may consult the VA for evidence. The log information will include information on all certificate validations with identification of certificate, RP and time. Thus, a VA by necessity obtains personal information.

The privacy issues for a VA are rather similar to those faced by an Identity Provider. A VA does not in itself provide identity federation and therefore has no user consent procedures. It is clear that a VA will in principle be able to track use of certificates across all RPs that the VA handles. However, the VA has no need for this information since its customers are the individual RPs. The only practical purpose of tracking use of a particular certificate may be to trace misuse of the certificate across RPs. Consequently, this functionality may be disabled.

A VA needs a published and carefully tailored privacy policy. The VA should gather and store personal information only to the extent needed, and all information, including logs, must be subject to adequate security mechanisms. In particular, log information must only be available to the correct RP.

6. Commercial and Legal Issues, Liability

6.1 Risk, Liability and Agreements

A VA must take on responsibility and liability with respect to its services. One reason for using a trusted third party service is risk management and risk reduction on the RP side. The VA should ideally provide a one-stop shopping service, where all relevant liability related to certificate validation is taken on by the VA. The VA should then be able to transfer liability to the CAs (or other information providers) if an erroneous answer from the VA is caused by erroneous information from such actors. The VA's liability must be clearly stated and accepted in the VA's agreement with the RP, and the cost to an RP may depend on the level of risk that the VA takes. Thus, the RP faces a clear risk picture and is provided with some risk reduction. However, a VA will definitely limit its liability.

A VA is an on-line service, and there is a clear risk that this will constitute a single point of failure for the RP. Unavailability of the VA will disable use of certificates for all RPs affected by the situation. This situation must be covered by service level agreements between the RPs and the VA. Additionally, the VA actor must ensure a service with very high availability, as discussed in 5.6.

An RP must also evaluate the risks related to continuation of the VA's service offering, such as bankruptcy of the actor behind the VA. A competitive environment should exist for VAs (see 6.2 below), and interfaces should be published and openly available to ensure that an RP is able to change to another VA. Change from a VA model to a non-VA model (based on trust structures such as bridge-CAs) may however require more work on the RP side. The agreement between an RP and a VA should ensure that logs and other material of potential evidential value can be transferred to the RP if the agreement is terminated.

The jurisdiction for an agreement between an RP and the VA will preferably be determined by the VA, but an RP may demand an agreement according to its own legal environment when the VA and the RP are in different jurisdictions (e.g. different countries).

A VA will on the other hand in most cases need agreements with the CAs (and other information providers). Relying on general statements in a CA's policy will be too risky. An agreement will in most cases be according to the CA's jurisdiction since the agreement resembles a relying party agreement with respect to the CA.

Note that such an agreement additionally provides risk management for the CA. As one example, the EU Directive on electronic signatures [9] mandates in principle unlimited liability for a CA issuing qualified certificates. Today, the only way for such a CA to control liability is to require agreements with all RPs. With a VA, the chain of agreements from a CA to a VA and on to the RPs may be used to limit liability.

Thus, a VA should aim at a situation where all relationships between actors are covered by agreements, providing a clear risk picture.

A VA is not an issuer of certificates and thus can assess the validity and quality of a certificate, but not the correctness of a certificate's content. The VA can take on liability for certificate content, but only if this liability can be transferred to the appropriate CA.

Operation of a VA as described in this paper may depend on changes in national legislation. As one example, the German legislation [2] requires a foreign CA to cross-certify with a German CA in order to have its qualified certificates accepted in Germany. The Regulatory Authority for Telecommunications and Post must approve the cross-certification. This is an unfortunate implementation of the paradigm that only a CA may be a trusted actor in PKI. However, an interpretation where a VA may take the CA's role, and the requirement for a cross-certificate as mechanism is relaxed, will solve the situation.

6.2 Customers, Payment, Competition

The liability that the VA takes on, and the operational costs of a VA, must be balanced by an income if the VA shall be able to make a profit out of the service. A VA provides on-line services. The RP will pay for the VA services according to the business model agreed (transaction based, volume based or fixed), and the VA in turn may pay CAs and other information providers according to agreements.

PKI interoperability problems are faced by service providers (government and business), requiring PKI-based authentication and signatures from the customers, and by businesses for (signed) B2B communication. However, VA services to the general public, e.g. to verify signed email no matter the CA of the sender, is also interesting. It is recognised that to the general public, anonymous access is beneficial, but note that most auxiliary information that can be returned from a VA need to be subject to access control, and will require authentication. At present, payment also requires authentication.

CAs are off-line services. A CA might prefer a low price for issuing of certificates combined with a fee for use of certificates, where this fee is collected from the RPs. Pay for use is only possible for on-line services, which for a CA are revocation checking and directory services. If revocation checking is based on CRLs, an RP will typically download CRLs periodically to a cache and perform further revocation checking from the cache. If the RP instead uses a VA, the VA may provide per use billing even for CAs that only provide CRLs.

An RP should need to trust and have a contract with only one VA. A competitive market exists for certificates (CA services), and correspondingly a competitive market should exist for VA services. Competition should be based on cost and quality of service (QoS). In addition to customary QoS parameters like response time and availability, QoS elements for a VA may be e.g. the number of CAs handled, responsibility/liability taken on by the VA, the classification scheme used, possibilities for auxiliary information, and the interface(s) offered.

Competition is limited if interfaces offered by a VA are closed and proprietary, necessitating a "deep integration" with systems at the RP. We suggest use of Web Services with published XML specifications to interface a VA (see 5.6).

7. Conclusions

An alternative approach at PKI interoperability is suggested, where interoperability is offered by means of an independent, trusted Validation Authority (VA). The trust model for the PKI Relying Party (RP) is revised, and the RP takes direct trust in the VA, not CAs. The RP is then able to trust all CAs that the VA handles. The VA handles all CAs individually, thus eliminating the need for trust structures among CAs and the resulting certificate path discovery and validation procedures.

A VA must be offered by an actor independent from the CAs. The VA should provide to an RP: Status on validity of certificate, quality classification of the certificate, and a clear picture of the liability issues. A VA must take on liability for its actions, thus providing risk reduction for the RPs. A commercial VA must have an income or funding to be able to cover liability and expenses and run a profitable business. Thus, the added value to the customers must be sufficient for them to be willing to pay. The main achievement to an RP in addition to risk reduction is one-stop shopping

(agreement, billing, complaining, trust, liability) for acceptance of certificates.

The VA scheme is based on agreements, between the VA and the RPs on one hand and the VA and CAs on the other hand. Thus, unlike other approaches to PKI interoperability, the RP obtains an agreement for acceptance of certificates from any CA.

References

1. British Standards Institute: Specification for Information Security Management Systems. British Standard BS 7799-2:2002 (2002)
2. Bundesnetzagentur: Ordinance on Electronic Signatures. (2001)
3. Certipost: Certification Practices Statement, European IDABC Bridge/Gateway CA for Public Administrations v2.0. EBGCA-DEL-015 (2005)
4. Chokani S., Ford W., Sabett R., Merrill C., Wu S.: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. RFC3647 (2003)
5. Commission of the European Communities: Action Plan for the Implementation of the Legal Framework for Electronic Public Procurement. Communication from the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions (2004)
6. ETSI: Policy Requirements for Certification Authorities Issuing Qualified Certificates. ETSI TS 101 456 v1.4.1 (2006)
7. ETSI: Policy Requirements for Certification Authorities Issuing Public Key Certificates. ETSI TS 102 042 v1.2.1 (2005)
8. ETSI: Electronic Signatures and Infrastructures; Provision of Harmonized Trust Service Provider Information. Draft ETSI TS 102 231 v1.2.1 (2005)
9. EU: Community Framework for Electronic Signatures. Directive 1999/93/EC of the European Parliament and of the Council (1999)
10. Federal PKI Policy Authority (FPKIPA): US Government Public Key Infrastruture: Cross-Certification Criteria and Methodology Version 1.3. (2006)
11. Federal PKI Policy Authority (FPKIPA): X.509 Certificate Policy for the Federal Bridge Certification Authority (FBCA) Version 2.1. (2006)
12. Hallam-Baker P., Mysore S.H. (eds.): XML Key Management Specification (XKMS 2.0). W3C Recommendation. (2005)
13. IDA: A Bridge CA for Europe's Public Administrations – Feasibility Study. European Commission – Enterprise DG, PKICUG project final report (2002)
14. ISO: Evaluation Criteria for IT Security. ISO 15408 Parts 1-3 (1999)
15. ISO/IEC: Information Security Management – Code of Practice for Information Security Management. ISO/IEC 17799 (2000)
16. ITU-T | ISO/IEC: OSI – the Directory: Authentication Framework. ITU-T X.509 | ISO/IEC 9594-8 (2001)
17. Lioy A., Marian M., Moltchanova N., Massimiliano P.: The EuroPKI Experience. EuroPKI 2004 – First European PKI Workshop (2004)
18. Lopez D.L., Malagon C., Florio L.: TACAR: A Simple and Fast Way for Building Trust Among PKIs. EuroPKI 2004 – First European PKI Workshop (2004)
19. Malpani A.: Bridge Validation Authority. ValiCert White Paper. (2001)
20. McBee F., Ingle M.: Meeting the Need for a Global Identity Management System in the Life Sciences Industry – White Paper. SAFE BioPharma Association. (2005)
21. Myers M., Ankney R., Malpani A., Galperin S., Adams C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. RFC2560 (1999)
22. NIST: Public Key Interoperability Test Suite (PKITS) Certification Path Validation. (2004)
23. NIST: Path Discovery Test Suite Draft Version 0.1.1. (2005)
24. OASIS: Understanding Certification Path Construction. White Paper from PKI Forum Technical Group (2002)
25. Pinkas D., Housley R.: Delegated Path Validation and Delegated Path Discovery Protocol Requirements. RFC3379 (2002)
26. TeleTrusT Deutschland e.V.: Bridge-CA Certificate Practice Statement (CPS) (2002)
27. Ølnes J.: DNV VA White Paper: PKI Interoperability by an Independent, Trusted Validation Authority. DNV Report 2005-0673 (2005)

Achieving Email Security Usability

Phillip Hallam-Baker

Principal Scientist, VeriSign Inc.

Abstract

Despite the widespread perception that email security is of critical importance cryptographic email security is very seldom used. Numerous solutions to the problem of securing email have been developed and standardized but these have proved difficult to deploy and use.

One of the main reasons for this difficulty is that each piece of the required technology has been developed independently as a generic platform on which security solutions may be built. As a consequence the user is left with an unacceptably complex configuration problem.

This paper proposes a means of providing transparent email security without the need for additional configuration based on existing security standards (XKMS, S/MIME, PGP, PKIX) and the recent DKIM standards proposal. Although the client deployment mode is considered the same approach would be equally applicable to an edge security configuration. Possible extensions of the protocol allow support for document level security approaches and to resist attack by quantum cryptanalysis.

The Usability Problem

It is a truth universally acknowledged that an Internet user in possession of an email application must be in want of encryption.

Despite the strong and nearly universal belief in cryptographic security within the information security field, users have proven exceptionally reluctant to use the encryption features built into practically every major email program for close to a decade.

It is time for the security community to recognize that the users do not reject cryptographic solutions out of ignorance. They reject them because they are too difficult to use and often fail to meet their real security needs.

The cost of public key infrastructure that impedes deployment is mental rather than financial. Users do want security. But they are not prepared to do their work any differently or learn any new tools to achieve this. Users demand security that is completely seamless and transparent, built into the fabric of the Internet infrastructure.

The need for ubiquitous Internet security has never been more apparent or more acute. Internet crime is now a professional business conducted for profit. The twin engines of Internet crime are spam and networks of compromised computers (botnets). The lack of a ubiquitous email authentication infrastructure allows phishing gangs to steal credit card numbers and access credentials by impersonating trusted brands.

The demand for usable security is critical even in classified applications that have traditionally relied on sophisticated operating systems designed to be secure at all costs[1].

What is usability?

A secure application should require no more training and be no more difficult to use than an insecure one.

In order to realize these goals it is necessary to:

- Employ consistent and familiar communication methods
- Eliminate all non-essential interaction
- Communicate all essential security information

While these goals may not prove to be sufficient it is clear that they are necessary and that current email security implementations do not achieve them.

How current systems fail

Instead of being presented with a solution that provides security automatically and reliably the user is given a ‘self assembly kit’.

Once the user has selected a Certificate Authority and enrolled for a digital certificate S/MIME allows her to sign individual email

messages or set a policy of signing all outbound email. If there is a digital certificate available for the recipient she may choose to send the message encrypted, or not.

For the average user this already represents a bewildering array of decisions but the user is still far from having a fully functional email security solution. She has not yet configured her LDAP directory or her SCVP interface. She has not loaded her smartcard drivers. And after completing all these tasks she will have to renew her certificate a year later when the original expires.

PGP suffers from similar usability problems, notably described by Whitten and Tygar[2] in 1999. Like most S/MIME interfaces the PGP 5.0 interface described in the paper is designed with the goal of allowing the user to use cryptography as if this was the end rather than merely the means.

Later versions of PGP, notably PGP Universal have attempted to overcome the usability deficit. However this has been achieved by having "declared peace in the certificate and message format debates"[3] and essentially implementing every variant of every standard. As such PGP universal is agnostic on the critical question as to which software architecture is most likely to enable a ubiquitous Internet wide email security infrastructure.

Traditional PGP offers the non-technical user an even more puzzling requirement. Before they can use their key they should get it signed by one or preferably several other PGP users that they already know.

Enterprise strength PKI systems allow network administrators to substantially mitigate this pain for the enterprise user. The personal Internet user is left on their own. Their perception of their security needs and thus their tolerance for deployment pain is very substantially lower, yet as the problem of phishing demonstrates personal Internet users have more than sufficient assets to be the target of professional Internet criminals. Personal users may have less confidential information to be stolen but they have money that can be stolen and they are much more likely to be tricked into parting with it.

The deployment problem

"Philosophers have only interpreted the world in various ways, the point is to change it" – Karl Marx

In the mid 1990s a considerable effort went in to ensuring that every major email client supported the S/MIME protocol. But even though this top-down 'deployment' was almost completely successful in making secure email available to over a billion users it was entirely unsuccessful in persuading them to use it.

The bottom-up deployment strategy of PGP was only marginally more successful. PGP persuaded a significant minority within the technical community to install and configure a security plug in. But even amongst this community security is the exception, not the rule. Only a tiny number of PGP key holders use it every day. Neither protocol has succeeded in achieving ubiquitous use today, nor is there reason to believe that this will change in the future.

Metcalf's law and its corollary

Metcalf's law states that the value of a network is proportional to the number of people it reaches. Metcalf's law is often quoted in the context of breathless pitches for 'viral marketing' programs premised on the fact that once a network has gained 'critical mass' its growth becomes self-sustaining.

The unfortunate corollary to Metcalf's law is the chicken and egg problem. The same process of positive feedback can cause a network that has not reached critical mass to quickly *loose* members. The Internet now has over a billion users and 'critical mass' for an application is likely to be several tens of millions of active users.

The problem of network effects is even more acute when a new network is in competition with an established one. If an S/MIME signature is added to an email there is a small but significant risk that the receiver will not be able to read it. Some email programs cannot process messages in S/MIME format. Other programs can process the message but display it to the user in a distinctly unhelpful fashion. An early version of the Internet access software provided by one major ISP displays a helpful message 'warning' the user that a signed email has been received.

The installed base

As we have seen the success of any new security infrastructure depends in large measure on how it interacts with the existing infrastructure.

In particular the development cycles for client applications are typically three years or more[i] and at any given time at least half of the installed base of applications is three years old or more.

It is clearly desirable for a security proposal to be as compatible with the installed base as is possible. But it is unrealistic to expect that legacy systems will be as secure as those that are updated.

It is important that a secure email protocol be compatible with the legacy infrastructure but it is also important that expectations be realistic. It is essential for legacy users to be able to communicate and interact with secured systems. It is neither essential nor realistic to expect a new security protocol to offer infallible protection for the user who does not have an up to date application or whose machine has been compromised by a Trojan.

Essential criteria

- Provide acceptable security and usability when used with an aware client
- Provide acceptable usability when used with a non-aware client

Non-Essential criteria

- Provide protection against bug exploits in legacy applications or platforms.
- Provide protection when the user's machine has been compromised by a Trojan.

Early adopter community

The usual solution to this corollary is to identify a community of early adopters with an urgent need for an email security solution that meets a particular need within that community.

The early adopter generally targeted for this approach is government, in particular the United States Government. In the early days of the Internet the US government and government funded research institutions represented a clear majority of Internet users.

The problem with this approach is that the needs of early adopter communities tend to be specialized. A solution that meets these needs may not meet the needs of Internet users as a whole. Early adopter communities are also likely to be tolerant of usability problems that are show stoppers for Internet users as a whole.

The problem of specialist needs is particularly acute in the US government. In addition to being considerably larger and more complex than the largest corporation the US government has considerably more information to protect and a greater need to keep it secure. The military alone has over 1.4 million active duty personnel, 1.2 million reservists, a further 654,000 civilian employees and indirectly employs a similar number of contractors[5]. In addition approximately two million retirees and family members receive benefits. In comparison Wal-Mart, the worlds largest corporate employer has 1.6 million employees[6].

Early adopter communities can also be unrepresentative of even their own needs. The US government certainly has a need for a security infrastructure that allows confidential and classified information to be protected. But it is not clear that these needs are met by an email security protocol. A classified document should be encrypted whether it is stored on disk or traveling over the Internet. This requirement is more appropriately met by document level security systems being developed in the context of Trustworthy Computing and Digital Rights Management.

It appears that S/MIME has failed to meet government needs by offering too little even as it has failed to achieve widespread deployment by requiring too much.

Pain Point

Deployment of new Internet infrastructure is expensive and time consuming. This expense is only likely to be met by a security protocol if it meets a critical pain point that is urgently felt at the time it is being deployed.

Unlike the 'early adopter' strategy which attempted to identify a subset of users for whom the proposal represents a 'killer application' in the 'pain point' strategy we attempt to identify particular functionality that addresses an issue of immediate and urgent concern for the community of Internet users as a whole.

---

[i] For example consider the release cycle of Microsoft Windows for home use, major updates occurring in 1995, 1998 and 2001[4]

The pain that is being felt most urgently on the Internet today is caused by Internet crime, in particular spam and phishing[7].

Bootstrap strategy

Addressing an urgent pain point is a necessary requirement for achieving a critical mass of support. If we are not careful however we may end up with a proposal that meets the requirements for addressing the pain point and only those requirements. Instead of establishing a ubiquitous and pervasive security infrastructure for all email we will have only succeeded in meeting our current needs with no plan for extending the solution scope in the future.

Future-proofing a solution is particularly important in the context of Internet crime. Professional Internet criminals seek the largest return for the least amount of effort. Phishing spam is not their first criminal tactic to exploit the lack of security in email and unless we have a comprehensive email security plan it is unlikely to be the last.

Accountability not Control

Since its beginnings the field information security has been dominated by government needs and in particular academic perception of military needs. This has led to the development of security systems designed to control access to information:

Control Approach

- Authentication: Who is making the request?
- Authorization: Is the request permitted for this party?

The control approach is based on the assumption that there is a clearly defined set of parties, a clearly defined set of rules that are to be applied and that both the rules and the parties to which they are to be applied are known in advance.

There is no set of rules that can be written *in advance* that will infallibly identify spam email without mistake yet it is easy to recognize spam when it is received.

Not only do these assumptions fail when applied to a public network, they also fail for a large number of real world situations. Motorists are deterred from speeding through fines, license suspensions and prison terms rather than being prevented from speeding using a speed limiter. Even if every motorist was required to install a speed limiter this would only prevent one type of traffic violation; it would still be necessary to use the deterrence approach to control reckless driving, driving under the influence of alcohol.

The glue that holds social networks together is *accountability* rather than control. Control based security systems are not applicable to the principle security issues facing the Internet today: the problems of Internet crime, in particular spam and phishing. Nor should it be a surprise that the Internet security problems that have not been solved today are the ones which the control approach is not suited for. The problems for which it is suited have already been solved.

The accountability approach to information security is better suited to applications where the consequences of individual security failures are small but the aggregate consequences of many small security failures are significant.

Accountability Approach

- Authentication: Who should be held accountable?
- Authorization: What the likelihood of compliance?
- Consequences for default

As in the control approach the first two steps in the accountability triad are authentication and authorization. The principle difference is that in the control approach authorization is the last step in the process. The authorization decision is binary; access is either granted or withheld.

In the control approach there is a bias towards refusing access unless the criteria for granting it are met. The Internet security problems that have proved intractable using the control approach are problems where the consequences of incorrectly granting access on a single occasion are small (a single spam is an annoyance) but the consequences of incorrectly granting access on a large number of occasions are severe (a thousand spam messages a day is a crisis).

In the accountability approach there is a bias towards granting access, provided that we are confident that there will be significant consequences if the other party defaults. This is a much closer match to our typical ‘real world’ behavior than the principle of ‘do nothing until

completely sure' that characterizes the control approach.

The consequences of default may be loss of use, civil actions or even criminal prosecution. What is important in the accountability approach is that the perceived probability of the consequences being imposed and the consequences themselves be sufficient to deter an unacceptable rate of default.

The Responsibility Problem

Domain Keys Identified Mail (DKIM[8]) is an email authentication technology that allows an email sender, forwarder or mailing list to *claim responsibility* for an email message. A party that claims responsibility for an email message informs the recipient that they can be held accountable and thus may increase the probability that the intended recipient will accept it.

Although DKIM does not and cannot solve the spam problem directly, DKIM allows email senders who volunteer to be held accountable to distinguish themselves from likely spammers. The spammers have a vast array of tactics but each and every one is designed to avoid the spammer being held accountable.

The DKIM message signature format allows a signature to be added to an email message without requiring modification of the message body. This ensures that (unlike S/MIME or PGP) the addition of a signature to an email does not negatively impact any recipient. Another significant departure from previous schemes is that recipients are advised to treat a message carrying a signature that cannot be verified as if it were unsigned.

The DKIM sender signature policy record allows a domain name owner to explicitly deny responsibility for unsigned mail message by stating that all authentic mail is signed. This makes it possible for an email recipient to conclude that an unsigned message is likely to be a forgery, a conclusion that is not possible with any of the previous cryptographic email security proposals.

Edge Architecture

Unlike the traditional approaches that attempted to identify the individual responsible for sending the email, DKIM is designed to identify a domain name owner that take responsibility for the email. The Internet has a billion users, attempting to hold each and every user accountable for sending unwanted email is a futile effort. Holding ISPs, Corporations, Schools and Universities accountable for policing their own users is much more promising.

In particular the DKIM architecture is designed to the assumption that messages are signed at the outbound email edge server of a network rather than by individual who sent it. On the receiving side the design is optimized to meet the needs of a signature verification filter at the incoming email edge server. In most cases this filter would be a part of a spam and virus filtering solution.

The edge architecture of DKIM allows for rapid deployment as an organization can deploy DKIM through an infrastructure upgrade limited to the email servers.

DNS Key Distribution

DKIM is a highly focused proposal designed to solve the responsibility problem using minimal extensions to existing protocols and infrastructures. Instead of proposing deployment of a new Public Key Infrastructure for key distribution DKIM keys are distributed through the DNS using unsigned public key values stored in a standard text record.

Using the DNS to provide the key distribution mechanism allows any email sender to start accepting responsibility for outbound email by signing it without requiring the sender to deploy any new infrastructure beyond adding the email signature module to their outbound mail server and adding a small number of text records to their DNS.

The disadvantage to this approach is that the key distribution mechanism is limited by the architecture of DNS which is designed to provide a fast response to contemporaneous requests. The DNS has no concept of history and there is no way to ask 'what did this DNS record look like two months ago'. While this is not a significant constraint when an email message is being validated in-transit (e.g. at the inbound email edge server) the DNS is not an ideal infrastructure for serving the key distribution needs of an email client which might want to verify a signature on an email opened hours, days or even months after it was originally sent.

The Authenticity Problem

Traditional email security approaches consider confidentiality and integrity to be complimentary tasks that are equally important. This assumption introduces a subtle bias into the architecture as it is assumed that senders and receivers must both upgrade their email clients to exchange secure mail.

This assumption certainly holds for encrypted mail where a recipient must have the means to decrypt the message in order to read it. But the assumption that a recipient must have the means to check the signature on a signed mail before reading it is a major departure from existing practice. It has led to a situation where S/MIME signatures cannot be used against the problem of phishing because of the minority of email readers that are unable to present a signed message to the user in an acceptable fashion.

The problem of phishing highlights the need to consider authenticity separately from the problem of integrity. It is much more important that a recipient be able to identify the sender of an email than know with certainty that the content has not been modified in any respect since.

Traditional email security approaches have attempted to identify the sender of an email by means of an X.500 distinguished name or an RFC 822 email address. The second approach has proved more successful than the first but still allows email senders to be impersonated through use of 'cousin' or 'look-alike' domains. DKIM allows 'AnyBank' to prevent an attacker successfully impersonating anybank.com. DKIM does not prevent the attacker registering a similar domain name such as any-bank.com or anybank-security.com. The introduction of internationalized domain names[9] provides additional scope for this type of attack.

A phishing impersonation attack is directed at the weakest link in the security chain, the gap between the computer screen and the user's head. To close that gap the authenticity of the message must be demonstrated using cues that are familiar to the user. A user cannot and should not be expected to recognize AnyBank by its Domain name any more than by its telephone number or ABA routing number. Customers recognize businesses in the physical world by their brands. Every large bank has a team of people whose sole job is ensuring that every piece of information issued by the bank, every letter, every credit card, every ATM is consistently branded with the current logo. To solve the authentication problem the same cues must be applied to Internet communications.

Secure Internet Letterhead

Secure Internet Letterhead is a proposal for a comprehensive Internet authentication infrastructure that allows every trustworthy Internet communication to be securely marked by a trusted brand.

The SSL padlock interface is designed to tell the user '*if* the padlock icon is present *the domain name component in the address bar* can be trusted'. The Secure Internet Letterhead approach is direct: '*if* the trusted brand logo appears in the secure area of the browser *it* can be trusted'.

For a user interface component to be trustworthy it must always be trustworthy. DNS Domain Names and X.500 distinguished names were both designed to provide a directory function. Attempting to overload this function and in addition use them as a security indicata is doomed. Secure Internet Letterhead introduces a new indicata whose sole purpose is to provide a security indicata.

If the authentication mechanism is to be successful it must be applied consistently and ubiquitously. In addition to its application to email described in this paper work is underway to apply the same principles and underlying technology to Web transactions (using SSL) and to Internet Messaging, telephony and Video.

Secure Internet Letterhead is a realization of the PKIX LogoType extension proposed by Stefan Santesson et. al., expected to be accredited as an IETF draft standard in the near future.[10] The PKIX LogoType extension allows a certificate issuer to embed links to one or more logos representing the brands of the certificate subject and/or issuer.

Linking a certificate record to a DKIM public key record[11] allows the DKIM signature format to be used as a vehicle for applying secure letterhead. The brand of the message sender is only shown if the message signature verifies and the signature key is authenticated by an X.509v3 certificate carrying the corresponding LogoType extension that is issued by a trusted certificate issuer (Figure 1).

Figure 1: DKIM Secure Letterhead

The prototype implementation of Secure Internet Letterhead was developed as a Web Mail interface. This approach was chosen to further the deployment strategy. If one or more of the principal providers of Web Mail services were to deploy Secure Internet Letterhead critical mass would be achieved instantly. Even adoption by a single Web Mail provider would provide a compelling business case for Financial Institutions targeted by phishing to obtain a Secure Letterhead certificate.

Qui Custodiet Custodes?

The security of Secure Internet Letterhead is critically dependent on the trustworthiness of the certificate issuers. If an attacker can persuade a Certificate Authority to issue them a certificate with a logo that impersonates a trusted brand the introduction of letterhead makes the phishing problem considerably worse.

Various control based mechanisms have been proposed to ensure that Certificate Authorities carry out their duties accurately and effectively. Like all control based security approaches these suffer from the weakness that they can only define minimum standards for compliance. Control based security does nothing to encourage the development of improved authentication criteria above and beyond the minimum.

The most appropriate way to ensure the trustworthiness of Certificate Authorities in an accountability based security scheme is to apply accountability principles to the problem. Displaying the issuer logo to the user, either directly in the email message dialog or through a ‘pop-up’ or ‘mouse-over’ window forces the Certificate Authority to put its own brand on the line every time a certificate is issued (Figure 2).

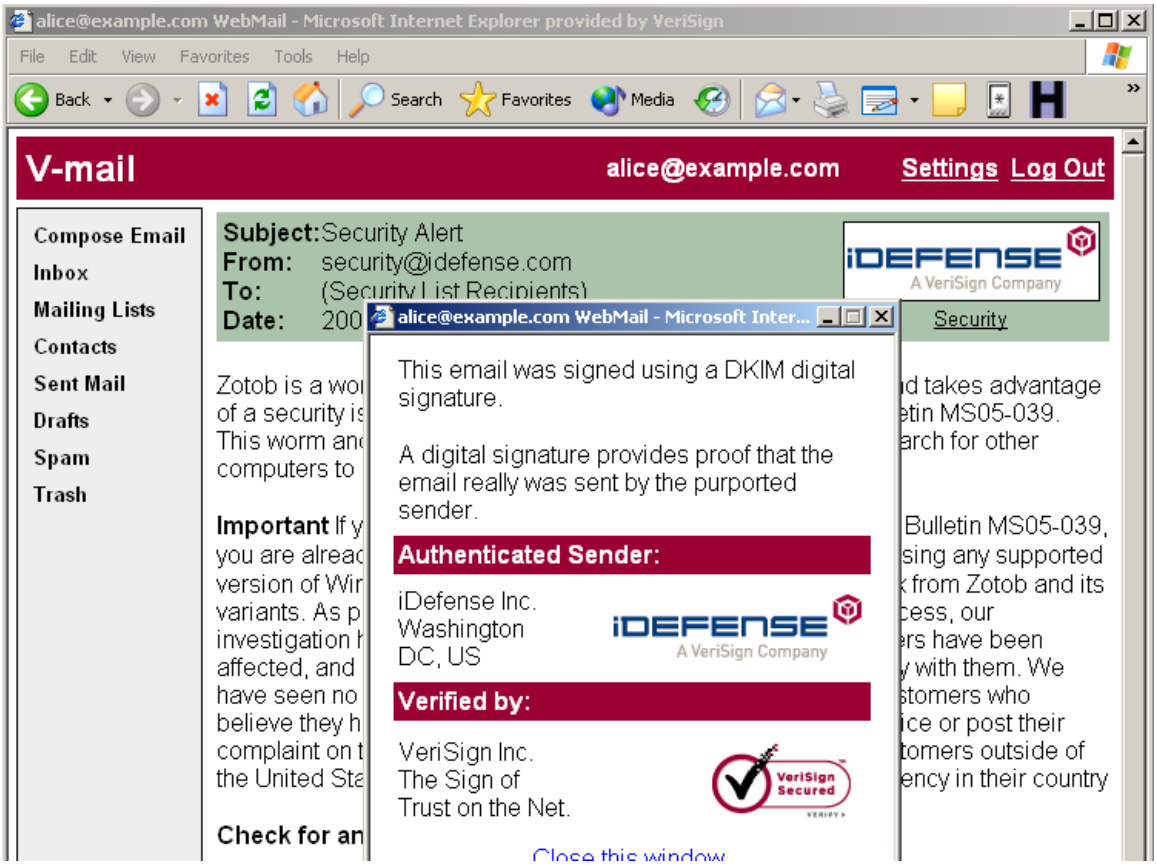


Figure 2 DKIM Secure Letterhead Issuer Logo

While effective authentication processes and rigorous quality control can minimize the risk of issuing a fraudulent certificate no amount of prior investigation can ensure that the Certificate subject will not default at a future date. Even the best known and trusted brand can be acquired by a company that is later discovered to be run by crooks and swindlers. For secure Letterhead to be trustworthy as well as merely trusted it is essential for the Certificate Authority to support rapid revocation of keys that are used fraudulently. For example by supporting a real time certificate status protocol such as OCSP[12].

Client Application Validation

The DKIM protocol combined with Secure Letterhead provides a robust solution to the authentication problem for users of hosted Web Mail services. As previously discussed however, DNS begins to show weaknesses as a key distribution infrastructure when signature verification is performed offline in the email client rather than during the transaction flow by the messaging infrastructure. A signature verifier can expect a DNS record to still be available minutes or hours after the message was sent. Demanding records to be available at an indefinite time in the future represents a significant change to the operational requirements of DNS.

For signature validation in the client application to be viable, persistent credentials are required. DNS is not designed to provide a persistent credential repository but other existing PKI protocols are. In particular XKMS[13] was designed to provide a persistent store for PKI credentials that is entirely agnostic with respect to the architecture of the underlying PKI. Like the DKIM DNS based key distribution model, XKMS realizes a key centric PKI model similar to the original Public Key Directory model proposed by Diffie and Hellman[14]. XKMS may also be used as a gateway to a traditional certificate based PKI following the Kohnfelder model[15].

The DKIM signature format allows additional key distribution mechanisms to be specified by means of an attribute. In a typical application both key distribution mechanisms would be

supported. This allows in-transaction signature verification filters to acquire keys quickly while ensuring that the needs of offline clients for a persistent and dependable key distribution infrastructure are both met.

Per User Signatures

Support for signature verification in the email client extends the scope of the DKIM signature to the receiving end of the communication. It is logical to look for ways in which the scope of the security context can be extended to the sending end of the communication, allowing the individual email sender to sign their correspondence with their own individual key.

Even though support for 'per-user' keying is outside the scope of the initial DKIM charter the base specification provides all the mechanism necessary to sign messages with individual user keys and to use them for message validation.

What the base DKIM specification lacks is support for management of the private key lifecycle. This is not a major concern for deployment at the edge. Even a large enterprise is unlikely to need more than a ten or a hundred domain keys. With 'per user' keying even a moderately sized enterprise may quickly find that it is managing hundreds, thousands or even hundreds of thousands of keys. Domain names tend to be relatively stable but students, employees and customers come and go. Unless the secure email client application provides support for key lifecycle management per user-keying quickly becomes unmanageable.

Key Lifecycle management with XKRSS

Fortunately XKMS also provides for key lifecycle management. The XML Key Registration Service Specification (XKRSS) component of XKMS is designed to support registration, reissue, revocation and recovery of private keys.

An XKRSS client may be written from scratch in a few days if an XML parsing library is available and open source toolkits are available for many languages.

The Configuration Problem

As the experience of S/MIME deployment demonstrates, support for a security feature is unlikely to be used if the end user is required to make an effort to configure it. XKMS supports automatic discovery of the local XKRSS registration service using the DNS service discovery (SRV) record[16].

If the user's email address is alice@example.com an XKMS aware client can discover the DNS address of the local XKRSS service by requesting the SRV record _XKMS_XKRSS_SOAP_HTTP._tcp.example.com. Once the XKRSS service is located the email client can register keys for any purpose they are required for: signature, encryption or key exchange.

The development of a prototype implementation revealed a minor shortcoming in this aspect of the XKMS design. The only way that the XKMS client can discover the features supported by the XKMS service is to attempt each one in turn. A richer service description language would allow the XKMS service to tell the client which services are available.

Encryption

DKIM, X.509 certificates and XKMS provide all the support necessary to support a comprehensive yet completely user friendly email authentication mechanism. Adding support for encryption completes the requirements for secure email as they are traditionally understood.

Instead of proposing yet another email message encryption format however we observe that the existing S/MIME[17] and PGP[18] message formats provide almost everything that is needed. While either message format would meet the technical requirements support for both formats is required to meet the political constraints created by the S/MIME vs. PGP standards war. To date this struggle has reached a stalemate, S/MIME dominates deployment but PGP dominates in mindshare. The quickest way to resolve this stalemate is to declare both formats winners and move on.

Problems

Although the S/MIME and PGP message formats are entirely sufficient both protocols have significant usability defects that must be addressed if our deployment criteria are to be met.

Key Distribution

The principle defect in the most commonly used implementations of the traditional email encryption formats is that both lack an effective mechanism for key distribution. Given an email address alice@example.com there is no simple process for locating the encryption key to use to send email to that address.

XKMS, and two recent PKIX extensions, PKIXREP[19] and the proposed CERTStore[20] extension solve this problem by allowing the email sender to discover the location of the key distribution service for the recipient using the same SRV mechanism used to discover an XKMS registration service.

Once the key distribution mechanism is made automatic an email client can be configured to automatically encrypt outgoing messages whenever an encryption key is available for the recipient. Email encryption becomes entirely seamless and automatic.

Encryption is Message Body Only

In S/MIME and PGP the SMTP encryption is applied to the message body alone, the subject line is left unencrypted despite the fact that the subject line is very likely to contain confidential content. As a result the legitimate expectations of the user are not met.

Solving this particular problem requires only the recognition that it is more important to meet the security expectations of the user. The solution adopted in the prototype is to introduce a confidentiality option into the email composition window. If the confidentiality option is selected the email client ensures that the entire message is encrypted by moving the subject line into the message body and adding a new subject line 'Confidential' or if applicable 'Client confidential – Attorney work product privilege asserted'.

If the confidentiality option is selected and it is not possible to send the message encrypted the user is warned. The user is given the option of canceling the message sending the message without encryption. The user might also be given the option of having the message printed out and sent by courier or sending the recipient a notice telling her to retrieve the message from a secured Web site.

Security is End to End Only

Although some effort has been made to introduce an edge-to-edge model to both PGP and S/MIME both specifications are essentially predicated on an end-to-end security model.

This causes particular difficulty where encryption is concerned since many enterprises do not want to accept encrypted email messages unless they are certain that they do not contain a virus or other form of executable code. Nor is end-to-end encryption likely to be acceptable to end users if it renders spam filtering measures inoperative.

Another source of difficulty with end to end encryption is the current trend towards receiving email on a wide variety of portable and mobile devices. It is not unlikely for a user to require access to their email by means of a desktop, laptop and PDA. The end to end principle is also inappropriate in the context of a Web mail service.

The XKMS architecture allows the domain name owner to control key distribution infrastructure for and hence the use of encryption in their domain. If the domain name owner wants to ensure that encrypted email can be read by virus scanning or compliance systems at the incoming edge server this can be achieved by returning the public key of the edge server in response to key location requests.

While this violates a core premise of the traditional email security protocols, that the end user should be empowered to control their own security, domain names are inexpensive. The user who feels the need for 'empowerment' and has the ability and inclination to control their own security can readily do so by obtaining their own domain name.

After decryption at the email edge server the message may be re-encrypted under the end-user's key. The resulting 'encryption with a gap' need not mean a weaker security solution than the traditional end to end approach. For most enterprises the risk of trojan code bypassing their firewall and anti-virus filters is considerably greater than the risk of unintended disclosure of confidential information. If a trojan is loose inside the enterprise the security of the email system is moot in any case.

In cases where the 'encryption gap' is a concern, the process of decryption, scanning for active code and re-encryption could be performed by

trustworthy hardware configured to refuse any administrative interference.

Complex Trust Infrastructures

The protocol profile described so far allows authentication and encryption capabilities to be added to an email application with a minimum of code and without affecting usability. While these capabilities are likely to be sufficient to meet the security needs of most enterprises they do not necessarily meet the needs of an enterprise which has already achieved a substantial deployment of a sophisticated PKI built on traditional principles.

Fortunately XKMS provides an answer to these cases as well. All that is necessary is for the email application that is attempting to locate an encryption or signature key to delegate the task to a local XKMS Validate service discovered using the same DNS SRV mechanism used to discover Locate and Registration services.

During the development of the prototype a minor bug was discovered in the XKMS specification which only defines a single SRV prefix for identifying an XKISS Locate or Validate service. While these functions might be combined in a single server the Locate service is primarily concerned with servicing external requests and the Validate service is like the Registration service essentially an exclusive service for the local domain.

It is therefore more likely that a Validate service would be combined with a Registration service than a Locate service. A simple solution to this oversight is to define a separate SRV prefix for the Validate service:

_VALIDATE_XKMS_XKRSS_SOAP_HTTP

DNS Security

A possible objection to the use of the DNS as a key distribution or service discovery mechanism as described in this paper is that the security of the key distribution infrastructure is ultimately dependent on the security of the DNS, a protocol that does not currently have a deployed cryptographic security infrastructure. While DNS security has not proved to be a source of chronic security problems as email has it is clearly unsatisfactory for the security of a cryptographic security protocol to rely on an insecure infrastructure.

Fortunately DNSSEC[21] meets this objection for both XKMS and the DKIM DNS key distribution. The principal obstacle to DNSSEC deployment has been the lack of a compelling use case for the domain name owner. The professional Internet criminal attacks the weakest, most profitable link in the chain. Until the systemic security failures of email are addressed the security shortcomings of the DNS are practically irrelevant. Using the DNS as the lynchpin of a ubiquitous cryptographic security system for email creates one of the strongest business cases imaginable.

Responding to change

As previously mentioned one of the most important tests of a security infrastructure is its ability to respond to changing needs. While it is impossible to foresee every need a system that is designed to meet the foreseeable needs is much more likely to meet unforeseen needs as well.

Document Lifecycle Security

The next major step forward in Information security is likely to be a transition from transport and message based protection to schemes that protect the integrity and confidentiality of *documents* throughout their entire life cycle. While an email message *may* contain sensitive information an attached spreadsheet titled ‘Accounts’ is almost certain to.

Various schemes for ‘Digital Rights Management’ or ‘Content Management’ have been proposed but in practice most effectively end at the enterprise border. Without the ability to exchange the necessary key information across the open Internet it is not possible for the CFO to send a document to external counsel for review, a sales person to send confidential contract proposal to a customer or meet many similar real world business security needs.

Although the XKMS based key distribution system and SRV discovery mechanism described in this paper is applied to the PGP and S/MIME encryption formats it could in principle be extended to support DRM or CM encryption formats as well. Alternatively if this approach proved to be too constraining the same SRV discovery mechanism could be applied to a SAML[22] service publishing the appropriate authorization assertions.

Incremental Advances in Cryptology

An ongoing concern for every developer of a cryptographic protocol is that advances in cryptanalysis might result in the underlying cryptographic algorithms being compromised.

Fortunately there is good reason to believe that DKIM and XKMS both offer realistic mechanisms for achieving a transition from one encryption algorithm to another. A paper simulation of a transition from the current RSA based signature algorithm to an ECC algorithm was conducted with satisfactory conclusions[23].

Quantum Computing

The worst case scenario for developments in cryptanalysis is the development of a quantum computer capable of performing calculations of significant complexity. Such a machine could in principle break every public key algorithm currently in use and it is prudent to assume that this represents an intrinsic property of public key algorithms.

Fortunately quantum computing is not currently believed to threaten symmetric key algorithms in the same degree and even the best quantum computer cannot factor an RSA public key it does not know. These premises and a minor modification to the XKMS key information protocol allow an XKMS configuration to be established which is secure even if the adversary has a quantum computer yet remains compatible with legacy systems.

In the standard public key model everyone who wants to send an encrypted message to Alice uses the same public key. In the modified model a separate key pair is established for each correspondent. The key Alice discloses to Bob is different from the key she discloses to Carol. The use of separate key pairs for each bilateral relationship allows the keys to be kept confidential so that Alice's public key used to receive encrypted email from Bob is only disclosed to Bob. Mallet cannot then cryptanalyze the key no matter how effective his quantum computer might be.

In effect the XKMS services at both ends of the communication act in the manner of a Kerberos[24] Key Distribution Center. The keying material that Bob receives from Alice's XKMS Locate service has an additional element carrying the private key encrypted under a symmetric key shared only by Alice and the XKMS Service.

The requirement for public keys to be kept private effectively eliminates the flexibility and convenience that makes public key cryptography such an attractive technology. In effect the parties end up with the convenience of a symmetric system and the performance of an asymmetric one. This is however an acceptable price to pay in the context of a worst case scenario in which the objective is to transition the network from the use of public key based technology to a symmetric system without a loss of service or functionality.

The only addition required to the XKMS protocol is the specification of appropriate algorithm identifiers and (as keys are now specific to a relationship between two users rather than just a key holder) a mechanism to allow the counterparty to the communication to be specified. A possible objection to this approach is that each message would have to contain both a public and a private key. The use of a public key encryption mechanism such as ECC that supports a more compact public key would meet this objection.

Conclusions

The problems of deploying ubiquitous email security are significant but as this paper demonstrates may be met by using a combination of existing protocols which are with the sole exception of DKIM all existing standards. The challenge of email security is thus similar to the challenge facing the field of networked hypertext applications in the early 1990s. The components all exist. The challenge that must be met is integrating those components in such a way that the user experience is fluent, seamless and learned automatically.

Despite the insistence that the user interface be at least as simple as the user interface for insecure email the system described in this paper offers at least as much security as existing schemes. It is not only possible to achieve usability and security, it is impossible to achieve security in practice unless an uncompromising approach is taken to both.

Acknowledgements

This paper has greatly benefited from the work and insights of many people. In particular Nico Popp, Siddharth Bajaj, Alex Deacon and Jeff

Burstein at VeriSign and Mark Delaney, Miles Libbey (Yahoo), Jim Fenton (Cisco), John Levine, Harry Khatz (Microsoft), Barry Leiba (IBM) and Stephen Farrell (Trinity College Dublin) in the DKIM working group. The Secure Letterhead concept was developed from concepts originally proposed by Stefan Santesson and refined by Amir Herzberg at Haifa University.

---

1 **Central Intelligence Agency Inspector General** *Report Of Investigation Improper Handling Of Classified Information By John M. Deutch* February 18, 2000

[2] **Alma Whitten and J. D. Tygar**. Why Johnny Can't Encrypt, 8th Usenix Security Symposium, 1999

[3] **Jon Callas**, PGP Inc. CTO, http://www.pgp.com/library/ctocorner/automagical.html

[4] **Microsoft Inc**, http://www.microsoft.com/windows/lifecycle/default.mspx

[5] **US Department of Defense** statistic, see e.g.http://www.defenselink.mil/pubs/dod101/dod101_for_2002.html

[6] **WalMart Inc.** see: http://walmartstores.com/GlobalWMStoresWeb/navigate.do?catg=1

[7] **Phillip Hallam Baker**, *The dotCrime Manifesto, To be Published*

[8] **E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas**, *DomainKeys Identified Mail (DKIM)*, IETF Draft, July 9, 2005

[9] **P. Faltstrom, P. Hoffman, A. Costello**, *FRC 3490 Internationalizing Domain Names in Applications (IDNA)*, March 2003 http://www.ietf.org/rfc/rfc3490.txt

[10] **S. Santesson, R. Housley, T. Freeman**, *RFC 3709 - Internet X.509 Public Key Infrastructure: Logotypes in X.509 Certificates*, IETF, February 2004, http://www.ietf.org/rfc/rfc3709.txt

[11] **Phillip Hallam-Baker**, *Use of PKIX Certificates in DKIM*, September 2004, http://www.ietf.org/internet-drafts/draft-dkim-pkix-00.txt

[12] **M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams**, *RFC 2560 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*, IETF, June 1999. http://www.ietf.org/rfc/rfc2560.txt

---

[13] **Phillip Hallam-Baker, Shivaram H. Mysore**, *XML Key Management Specification (XKMS 2.0)*, W3C Recommendation 28 June 2005, XKMS http://www.w3.org/TR/xkms2/

[14] **W.Diffie and M.E.Hellman**, *New directions in cryptography*, IEEE Trans. Inform. Theory, IT-22, 6, 1976, pp.644-654.

[15] **Kohnfelder**, *Toward a Practical Public Key Cryptosystem*, in Department of Electrical Engineering. 1978, MIT.

[16] **A. Gulbrandsen, P. Vixie, L. Esibov**, *RFC 2782 A DNS RR for specifying the location of services (DNS SRV)*. IETF, February 2000. http://www.ietf.org/rfc/rfc2782.txt.

[17] **B. Ramsdell**, RFC 3851 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification, IETF, July 2004, http://www.ietf.org/rfc/rfc3851.txt

[18] **J. Callas, L. Donnerhacke, H. Finney, R. Thayer**, OpenPGP Message Format, IETF, November 1998, http://www.ietf.org/rfc/rfc2440.txt

19 **S. Boeyen and P. Hallam-Baker**, *Internet X.509 Public Key Infrastructure Repository Locator Service*, RFC 4386, http://www.ietf.org/rfc/rfc4386.txt

[20] **Peter Gutmann**, *Certificate Store Access via HTTP*, RFC 4387 http://www.ietf.org/rfc/rfc4387.txt

[21] **R. Arends, R. Austein, M. Larson, D. Massey, S. Rose**, *RFC 4033 DNS Security Introduction and Requirements*, IETF, March 2005, http://www.ietf.org/rfc/rfc4033.txt

[22] **E. Maler et al.**, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML).* OASIS, September 2003. Document ID oasis-sstc-saml-core-1.1 http://www.oasis-open.org/committees/security/

[23] **Phillip Hallam-Baker**, DKIM Transitions, To be published

[24] **B. Clifford Neuman and Theodore Ts'o**, *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications, 32(9) pp33–38. September 1994, http://gost.isi.edu/publications/kerberos-neuman-tso.html

CAUDIT PKI Federation

A higher Education Sector Wide Approach

Dr Rodney McDuff

The University of Queensland

Viviani Paz

AusCERT

Abstract

Australian Higher Education Institutions, in common with other research institutions around the world, need to collaborate with each other and with global research partners. Cross-disciplinary research is also increasingly important between intra and inter-institutional groups and yet, mechanisms for communication between such groups are often insecure. Insecure communication methods are of particular concern for research because of the need to protect intellectual property.

The deployment of PKI in the higher education sector in Australia has been measured. Taking this early stage of PKI adoption into consideration AusCERT in conjunction with CAUDIT has been working on a Public Key Infrastructure (PKI) Project to establish a National Certificate Authority Framework for Australian and international universities and research groups interoperation. The first phase of this project (called CAUDIT PKI Federation pilot) included the development of policies and guidelines, the implementation of a prototype certificate management system and preliminary research into interoperation issues.

The intent of this framework is to minimize PKI up taking costs, minimize surprises once we move into a production environment and provide clear guidelines for implementation to avoid retrofitting.

This paper will discuss the basic implementation used and will look at some vital issues on how to enable secure interoperation amongst the Higher Education sector in Australia while drawing on the experience gained while implementing this pilot project.

1 Introduction

The CAUDIT PKI Federation project is part of a larger effort from Australian Higher Education Sector with support from AusCERT, CAUDIT, Grangenet and the Australian government to develop an environment in which Universities can collaborate at low cost and low risk to business-like institutions.

Our aim is to develop and ultimately implement a PKI for CAUDIT universities (which includes universities in Australia, New Zealand, Fiji and Papua New Guinea). To achieve this goal we are working closely with other projects such as Meta Access Management System Project (MAMS) and Middleware Action Plan and Strategy (MAPS) and are taking a phased approach to test interoperability and find out issues regarding PKI enabled applications.

This phased approach has enabled us to receive support from a number of organizations and to promote extensive research in the proposed PKI architecture and how it would perform in the higher education environment.

Further funding of $649,000 has recently been awarded to the University of Queensland by the Hon Dr Brendan Nelson MP, Minister for Education, Science and Training to develop an e Security Framework for Research which will enable a production PKI infrastructure to be built for the sector using the architecture and policies and procedures that have been developed in this pilot project.

The purpose of this follow on project is to implement secure access, authentication and authorisation for researchers who access services and infrastructure across global networks. This project seeks to establish an E-Security Framework to integrate two types of security systems, PKI and Shibboleth, to foster collaboration and enable the secure sharing of resources and research infrastructure within Australia and with international partners. The project will leverage off existing work in both areas,

build on the advantages of these different systems and create a platform to enable the secure sharing of resources for a research infrastructure.

2 CAUDIT PKI Federation Architecture

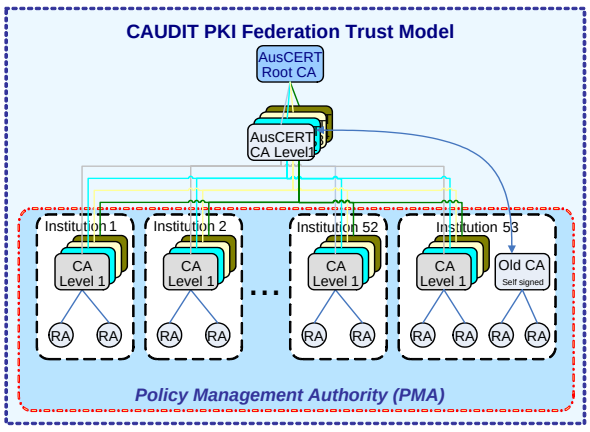
A given PKI can support a number of services in an organisation. The CAUDIT PKI pilot implementation provided three core services:

- **Authentication** – the assurance that the entity proves who they are (or claim to be).
- **Integrity** - that data has not been modified (intentionally or unintentionally) in transit).
- **Confidentiality** – the assurance of data privacy.

These services enable entities to demonstrate they are who they claim to be, to be assured that data is not undetectably modified, and to be certain that data sent to another entity is only read by the intended entity.

The CAUDIT PKI Federation has used a combination of trusted models to develop its own operational model. It is comprised of a single Root Certification Authority (CA), four Subordinate CAs corresponding to each level of certification and Institutions' CAs. The four Sub-CAs issue CA certificates to Institutions CAs within CAUDIT. Institutions within CAUDIT inherit the Certificate Policies and Certificate Practice Statement from the Root CA and four Sub-CAs, or comply with them. The trust model is described in detail on section 4.

The following diagram illustrates the architecture chosen.



3 Certification Levels

We believe that a fundamental issue for a successful PKI implementation is the identity of the end user (or entity) and the degree of identity checking and verification. CAUDIT PKI Federation proposed:

- **Use several identity certification levels** corresponding only to the strength of the identification process of the end entity; rather than what they are or what they do within the institution. Each level will also correspond to a different signing private key for the appropriate CA.
- **Base the identification process** on the Australian 100 points of identity system (described in the Financial Transaction Reports Act 1988 and Financial Transaction Reports Regulations 1990) using a modified Form 201 that requires completion and identification proof in the institutions' RA's presence.
- **Use four certification levels** as detailed below.

The default operating certification level, called Level 3, is granted once an end entity has successfully accrued at least 100 points of identification. In most institutions, staff on its payroll should proffer a birth certificate or passport (70 points) on induction or have a driver's license (40 points) or a credit card (35 points) and so will easily fall within this level. Similar most students (and others within the institution's circle) should be able to proffer enough credentials to eventually be certified to Level 3.

It made sense to consider certification levels both greater and lesser than Level 3. Certification Level 4 is used when there is a need from relying parties for identification process greater that Level 3. For example consider a relying party that is a digital repository containing confidential and very sensitive intellectual property. That relying party may insist that the end user have more than just 100 points of identifications but should also have a recent background check which indicates that this individual has no prior history of intellectual property violations. Information regarding the agency executing the background checks and check type can be encoded into the end users certificate within a X.509 extension attribute.

Certification Level 2 encompasses end entities that cannot for one reason or another provide enough credentials to meet the 100 points criteria. These users may still need a public certificate to access low risk resources where only the possession of a valid certificate is required. It would be discriminatory to deny these users access to these types of resources.

Certification Level 1 where end entities who are still with the institution's circle have not directly provided to the institution any credentials at all. However these entities should have provided identification credentials to another body (not within the CAUDIT PKI circle of trust), which has an agreement of mutual trust with that institution. An example of this is the process of enrolling new students into a university. In Australia state secondary education bodies transfer to the university enough information about new prospective students so that they can be enrolled and if necessary accounts created. However this information usually has not been vetted by the university for veracity at this stage. The university trusts the state body that the information provided is correct.

The table below summarises the CAUDIT PKI Certification Levels.

Certificate Level	***Description***
Level 1	• No proactive identity check provided to the RA. • Identity information provided by a body that the RA has a trust relationship. • Example: A student being enrolled in at least one subject is sufficient for the certificate issuing however identity information has only been supplied by QTAC (or similar state body).
Level 2	• Subject must provide proof of identity by appearing IN PERSON at the RA. • Individual cannot provide the required 100 points of identification. • Example: Short term contractors at an institution requiring access to PKI-protected systems whose credentials are insufficient credentials to meet the 100 points check but can provide some credentials (e.g. drivers licence, credit card, etc).
Level 3	• Subject must provide proof of identity by appearing IN PERSON at the RA. • Individual must accrue at least 100 points of identity. • Example: Foreign staff with valid passports and written references from acceptable referees.
Level 4	• Subject must provide the same information for Level 3 certification in addition to character background check. • For example a positive check is also conducted by an appropriate external agency.

4 Trust Model

A key benefit of PKI is the ability to construct a “sense of trust” between a relying party and an end entity (whoever or whatever they may be). This sense of trust has several aspects ranging from the technological to psychological. At both technological and psychological level a “trusted” connection must be made between a trust anchor of the relying party and a trust anchor of the end entity.

At a technology level, trust anchors are normally either the CA that signed the end entities’ own certificate or a set of CAs that the relying parties either explicitly trust or that the relying parties’ software’s vendor explicitly trusts.

Relying parties must attempt to construct either a direct or indirect path between the presented end entity certificate and its own trust anchor.

This process is trivial when the relying party and end entity share the same trust anchor. If the relying party and the end entity do not share the same trust anchor, the relying party must find a continuous chain of valid and appropriate CAs, starting from the end entity’s CA, and terminating at its trust anchor. If this path cannot be constructed and validated then the relying party must be alerted to the absence of trust.

This process is called “Certificate Path Processing” and it is a major function of any PKI. If the same CA signs all end entity certificates, Certificate Path Processing is trivial and requires limited consideration. However reality is more complicated with thousands of active CAs having complex and opaque relationships.

For a relying party to transverse a chain link between two CAs (and therefore infer a level of trust between them), they must have previously setup a trust relationship between themselves; either by being a subordinate CA to the other or by (unilaterally or bilaterally) cross-certifying themselves.

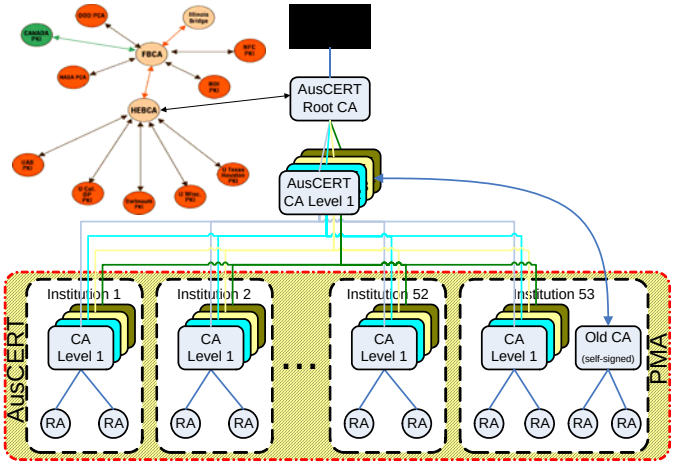
CAs should not arbitrarily setup relationships as this weakens the chain of trust. Inference of trust must also be carefully handled. If CA_A trusts CA_B and CA_B trusts CA_C then the inference that CA_A trusts CA_C is not necessarily correct all the time.

CA certificate extension attributes (e.g. nameConstraint and policyConstraint) can be used to correct faulty trust inference logic; however problems also occur if the trust chain is too long including:

- **Path processing** - becomes more intensive for the relying party.
- **Trust erosion** - at each transition of a link of the chain the erosion of trust is a possibility as the policies and procedures of each CA may not perfectly align to relying party expectations. The CA certificate extension attribute pathLengthConstraint can be used to mitigate this problem.

4.1 CAUDIT PKI Trust Model

The CAUDIT PKI Federation is a combination of models:



Core CAUDIT PKI architecture - the Hierarchical CA model provides good flexibility to the members of the CAUDIT PKI and a reasonably simple trust topology for Certificate Path Processing.

- **Trust anchor** – AusCERT operates as the trust anchor for all the CAUDIT PKI due to existing trust relationships. AusCERT is seeking to have either its Root CA accepted into a broad range of vendors' trust lists or to have its Root CA signed by a well-known CA already in a broad range of vendors' trust lists.
- **Subordinate CA certificates** - from the AusCERT Root CA certificate, there are subordinate AusCERT CA certificates for each Certification Level implemented. This allows AusCERT and the CAUDIT PKI members more control over how PKI networking is achieved over the various Certification Levels by using various X.509 constraint extensions. Each institution will also have a separate CA certificate corresponding to each implemented Certification Level chained back to the corresponding subordinate AusCERT CA certificate.
- **Established PKIs** - institutions with an established PKI will implement their part of the above design and use it to sign new end entity certificates. End entities issued by the institution's old PKI can be transferred to the new design by cross-certifying their old CA certificate to the appropriate AusCERT subordinate CA certificate. This way these old end entities will still recognize the old CA as their trust root (and continue to function) and relying parties elsewhere can construct a chain to them.
- **PMA** - as each member of the CAUDIT PKI is its own self-contained organisation, AusCERT acts as a Policy Management Authority (PMA) to help maintain the trust fabric by periodically auditing the policies and procedures of each member.
- **Cross certification** - the AusCERT Root CA Certificate will eventually be cross-certified to other PKI federations (e.g. HEBCA and various GRID PKIs) to allow collaboration between parties at national, international and global levels.

5 Additional Design Considerations

There are many other design considerations to consider other than the identity certification levels and the trust model. We briefly discuss some of these issues below that are organized in around the various stages of the typical management lifecycle of a certificate [ADAMS2003]; namely initialisation, issuing and cancellation.

5.1 Initialisation Phase

This phase contains:

- **Registering** of the end entities;
- **Generating** of the key pairs;
- **Creating** certificates and distributing to the end entities (possibly including private key distribution);
- **Disseminating** the public certificates for use by relying parties; and
- **Backup of the keys**.

5.1.1 Registration

Our identity registration method is based on the Australian "100 points of Identification" system with credentials offered to a RA in person.

This method scales well while the CAUDIT PKI is small where RAs (used by end users to register) are distributed over various institutions and key organisational units. However it will become intractable when the CAUDIT PKI encompasses many end users.

Consider a situation of mandatory issue of personal certificate(s) for every student. This situation will require bulk certificate creation that will obviously comprise Certification Level 1, which is designed to handle this type of situation. End users with a bulk created certificate at Level 1 who require higher certification can present themselves to an RA and have another certificate issued. To minimize this certificate promotion, Level 1 certification must be sufficient for normal use.

Institutions are expected to employ a CMS capable of bulk key/certificate generation to prepare for large scale PKI deployment.

There are also issues regarding bulk creation of key pairs - particularly for certificates used for signing and non-repudiation. Typically the certificates for the key pair are generated on the end user's computer or crypto-token. Key pair generation by a third party implies knowledge of the private key and will weaken strength of non-repudiation.

5.1.2 Key Pair Generation

Key generation can occur at the:

- **End user's computer or crypto-token;**
- **RA**; or
- **CA**.

Depending on the use of the key there are factors that impact where it is generated.

Although losing a signing private key is inconvenient (as only its corresponding verification certificate is needed after signing data, the CA should hold a copy of this certificate), it may be disastrous if a decryption private key is lost resulting in permanent loss of corporate data.

If the signing private key is known to anyone other than the end user then the requirement of non-repudiation (ie "to prove to the satisfaction of a third party that the private key could not possibly have been used by anyone other that the owner of the private key") is compromised even if the "other" is the CA itself.

The CAUDIT PKI will issue separate keys/certificates for signing/non-repudiation, which can also be used for authentication since at its core authentication with X.509 certificates relies on signing a challenge from a party and returning it to be verified, and encryption to end users. To ensure that each certificate is only used for its appropriate purpose the issuing CA should set the appropriate X.509 keyUsage attributes.

At this stage we recommend generating signing key pairs on the user's computer or crypto-token; however we also recognise this may be problematic for large scale PKI production and there will be security issues to consider. We expect the onus be on the end user to ensure their signing key is appropriately backed up.

Encryption keys should be generated at either the RA or CA to enable automatic safe and secure archive. If an encryption key must be created on user's computer or crypto-token, the user must make all reasonable attempts to supply this key to the institutions CA for archival purposes.

5.1.3 Certificate Creation and Key/Certificate Distribution

After generating a key pair, the public key must be securely transferred to the CA for placement in a certificate and signing by the CA and the certificate relayed back to the user. Issued certificates should be published in the institution's directory so other users wanting to communicate with the user can easily locate it.

However if the key pair was generated at the RA or CA, the private key must also be securely communicated to the end user. This can be achieved using the X.509 PKI Certificate Management Protocol [RFC2510] or using Public key Cryptography Standard (PKCS)7 [RFC2315] or 10 [RFC2986]. The CMS employed by an institution should support at least one of these standards.

Although the ideal situation is to store private keys on a crypto-token (e.g. smart card that can be used for swiping and proximity but need a special reader, or USB key which have the advantage of being compatible with virtually all recent personal computers) rather than an encrypted file on the computers hard drive, we acknowledge these devices may still be relatively expensive for a University environment.

We also recognise that if the whole of CAUDIT and its encompassing staff and students are to eventually embrace the CAUDIT PKI Federation, the CAUDIT PKI Federation must embrace crypto-token technology. We recognise that the crypto-card option may impact various internal policies regarding student and staff identity cards. A workaround may be to deploy crypto-cards in parallel to established identity cards.

5.1.4 Certificate Dissemination

It is essentially important that the University community can readily find the certificates of people they want to securely communicate with. Public certificates should be published in the institution's directory; however although this aids intra-institution searches, it does not aid inter-institution searches and ideally a single location to search for certificates for all of CAUDIT's members is required.

One solution being investigated is for AusCERT to run a "directory of directories" service or a directory proxy. A "directory of directories" is an LDAP directory populated only with referrals to other directories. The searching application can follow the referrals to the target directory and in some applications these hopes are in vain. Also it is difficult to instigate a search for an individual across several institutions.

A directory proxy service takes the request (re-writes the request if necessary) and executes the search on the user's behalf at various institutions' directories. Results are re-written (if required), collated and returned to the user. A simple web interface (e.g. similar to the EuroPKI interface) will allow greater accessibility.

Another approach being investigated is using Google as a Web File (also called the "Public File") as suggested by Peter Gutmann [Gutmann04]. This approach embeds or links the user's certificate to the user's personal web page. As this page contains the user's name (and possibly a picture) a Google search will easily locate the information. To encourage this AusCERT is looking into developing a simple CGI script with a URL that embeds an identifier for the user's certificate that can be simply added to a personal web page.

This option would also be relevant for institutions planning or deploying web-based staff portfolio pages.

Privacy is a difficult aspect of certificate dissemination and it comes in two parts:

- **Encoded information** - identification certificates contain user information (e.g. name and email address) encoded in the certificate; and the certificate is useless without it. However after the certificate is disseminated it cannot be recalled (only revoked) and can remain in the public domain forever. There are schemes in which one put either an anonym or pseudonym in the certificate (rather than the veronym) to protect privacy; however this approach virtually cripples potential certificate use.
- **Searching** - privacy issues also arise by allowing everyone to browse and search the CAUDIT PKI directories and web pages for certificates. This issue is complex enough just within a single institution. We suggest that CAUDIT instigates a study of solutions to this problem across all its members.

5.1.5 Key Backup

Key backup is a key issue and we recommend backing up encryption keys at creation by the institution's CA. However, this implies the institution's CMS is capable of this function. Provided this process is secure, institutions are free to implement their own procedures, which will regularly be audited by the CAUDIT PKI Federation PMA.

To protect non-repudiation signing private keys should not be backed up by the institution at their creation; however we recommend backing up and archiving of the signing public certificate.

Users should backup either of these keys using an encrypted format and a strong pass phrase.

5.2 Issued Phase

After a private key and its corresponding public certificate have been disseminated they enter the "issued" phase that includes:

- **Retrieving** the certificate from a remote repository (where necessary)
- **Validating** the certificate whenever it is used
- **Recovering** the private key id lost; and
- **Updating** the certificate prior to expiration.

5.2.1 Certificate Retrieval

Certificate Dissemination is the act of publishing public certificates for use by others. Certificate Retrieval is the complementary operation where a relying party or end user retrieves the certificates from various repositories. The infrastructure for certificate retrieval is identical as that required for certificate dissemination and we make no further recommendation.

5.2.2 Certificate Validation

It is vitally important that any relying party can successfully perform Certificate Path Processing on certificates issued by CAs in the CAUDIT PKI Federation. Every effort must be made to create and maintain the necessary infrastructure for achieving this goal while considering the following:

- AusCERT will either place its Root CA Certificate in trust lists for well known applications or have its "Root" CA certificate chained to a well known CA certificate that already exists in the trust lists in well known applications.
- SSLv3/TLSv1-enabled servers must be configured to supply certificate chains to the relying party. This approach means relying parties do not need to inspect individual certificates to locate the certificates to traverse the CAUDIT PKI hierarchy to the top.
- S/MIME enabled mail clients must be configured to embed certificate chains with the PKCS#7 MIME attachment. This way relying parties do not need to inspect individual certificates to locate the certificates to traverse the CAUDIT PKI hierarchy to its top.
- All issued certificates must use the following X.509 extension attributes:
    - Authority Information Access Extension (AIA) to supply to the relying party:
        - Location of certificate chains and cross-certificate pairs.
        - Location of CRLs and OCSP responders
    - CRL Distribution Points Extension to supply to the relying party
        - Location of CRLs.
- All issued CA certificates and cross-certificates must be published in either a X.500 or LDAP directories so that relying parties and DPP/DPV servers can locate them. If LDAP servers are used then a "Directory of Directories" or Directory Proxy service will be necessary.
- Institutions must publish regular and timely CRL information. If revocation list grows large they should consider using CRL partitioning and Delta CRLs to minimise bandwidth. Institutions will be expected to run an OCSP responder.
- There must be a single point of CRL and OCSP information for applications that cannot discover their locations via information in the certificates. These services may be provided using Indirect and Redirect CRLs and OCSP proxy.

5.2.3 Key Recovery

End users will lose private key and forget pass phrases protecting private keys. In this situation, the RA or CA may need to retrieve the key from the key archive and securely transmit the key to the owner to prevent permanent loss of information. We recommend institutions deploy a CMS capable of key backup and recovery.

5.2.4 Key Update or Renewal

When a certificate is near to expiration and the end entity still needs a certificate, the CA can either:

- **Renew the certificate** – in this operation the user's original public key is placed in a new certificate and issued back to the end user prior to certificate expiration. This operation can be automatically initiated by the CA prior to the end user's certificate expiration; or
- **Update the certificate** – in this operation a new key pair is generated and a new certificate is issued. For this operation to take place the end user must send a certificate update request to the RA.

Institutions can select the best method for itself, its staff and students that provide a balance between security and convenience. Either way the end entity must be notified of the impending expiration in advance so they can initiate key update or renewal. For scalability issues, this process should be as automated as possible and as transparent to the end entity as possible.

5.2.5 Cancellation Phase

This phase covers the natural expiration of a certificate (and revocation if required) in addition to reissuing or renewing expired or expiring certificates.

The cancellation phase also involves the records management task of maintaining a history of keying material so data encrypted by now-expired certificates can be decrypted in the future (if required) as well as for dispute resolution purposes.

5.2.6 Certificate Expiration

The aim is to maximise the number of naturally expiring certificates and minimise the number of certificates that must be revoked (e.g. users leaving the CAUDIT PKI, etc.). CAs should also aim to minimise certificate renewals and updates.

For example, consider certificates issued to students and the following options:

- **Issuing certificates on the 1st January valid for approximately one year** - each year new students must be issued with certificates and continuing students must renew or update their certificates. During the year the CA must track students permanently leaving and revoke their certificates. However some proportion of students graduate and leave each year at or about when their certificates naturally expire and require no revocation. For this option the process of renewing or updating certificates for continuing students is an intensive task while the revocation of certificates has less impact.
- **Setting the student certificate validity period to approximately 3 years** - to coincide with the average university degree period. In this situation, new students are issued certificates as normal and for a large majority as they graduate their certificates should be also expiring. Certificates for the minority remaining longer than 3 years can be renewed or updated for each extra year at the institution. Certificates must still be revoked for students leaving before the three years. This option is lighter on certificate renewal or update as compared to the previous option; however it is heavier on the process of revocation. This option also creates CRLs that are significantly larger than the previous Option.

Selecting an optimal validity period for staff is more difficult due to irregular staff employment terms. While some staff members have fixed term employment (and therefore a predictable expiry date), the majority may leave the institution before their certificates expire naturally and therefore require revocation.

We recommend institutions carefully select validity periods and revocation policies that best suit each institution needs.

5.2.7 Certificate Revocation

Under the CAUDIT PKI Federation certificates can be revoked for the following common reasons:

- **Compromise of end entity's private key** - due to a stolen computer or crypto-token or the computer upon which the private key is held has been comprised, the affected certificate should be revoked as soon as possible. It is the duty of the end entity to contact the RA or CA immediately once they realize the computer/crypto-token has been stolen or otherwise compromised. However the institution must publish precise instructions to be followed in this case. If the end entity has misplaced or lost the computer/crypto-token where their private key(s) reside, they also should contact the CA or RA as soon as possible to revoke the certificates. Authorized administrators must also be able to initiate revocation if they suspect compromise of a private key.
- **Termination of institution association** - most institutions are dynamic bodies with staff and students regularly entering and leaving the institution. End users will inevitably terminate their employment and/or studies before natural certificate expiration. In this situation, certificates should also be revoked. Most institutions have well defined staff termination procedures and checklists that could be updated to include processes for revoking staff certificates; however students pose problems as they generally have less well-defined procedures.
- **Changing certificate information** - information in a certificate will inevitably change (Certificate Perishability) and it may become necessary to revoke that certificate (and reissue another certificate) before the certificate naturally expires. Examples of such changes include name, email address or affiliation changes. To counter this situation, institutions should minimise the use of attributes with the potential to change regularly (e.g. refraining from adding attributes in an ID certificate for authorisation purposes). Attribute certificates or access management systems like Shibboleth are better suited for this.

5.2.8 Key History and Archive

We recommend institutions' CAs should archive all keying materials or encryption certificates and the public certificate for signing certificates including renewed certificates and updated key pairs.

Archiving allows the institution to decrypt encrypted data when private keys are lost. Also signed documents can still be verified in the future even when the user has updated or renewed their certificates and have removed or deleted the older versions.

6 Approach used

We have developed a phased approach to ensure that the production implementation is not only feasible, but also useful to each individual university.

- **Pilot Phase** - extensive research is being undertaken to understand interoperability issues with PKI enabled applications that may arise in a production environment.

- **Pre-Production Phase** – investigate inclusion of Root CA into web browsers certificate authorities and compliance requirements to the appropriate FIPS. Investigate Higher Education requirements for authorization certificates including short-lived authorization certificates. Investigate alignment of Shibboleth into the CAUDIT PKI Federation Trust fabric, which will be performed in collaboration with MAMS project.
- **Initial Production Phase** – deploy an environment that enables Universities collaborative research in a safer manner. Empower Universities with the necessary information to train their users.

While these phases are very distinct they are also interconnected in a way that the results from one phase will impact and direct future phases. Using this phased approach we hope to be able to map and document any technical and philosophical problems that may hinder a PKI implementation.

We understand that one of the major hurdles of deploying a large PKI is not so much the technical intricacies of PKI enabled technology available to date, but the support from management and end users.

We all agree that PKI is not a simple implementation and that end users may be reluctant to accept and adopt new technologies, however we hope to develop an infrastructure that is as simple as possible to fit in with existing individual Universities infrastructures.

7 Conclusion

As we progress in the implementation of the CAUDIT PKI Federation Project we face technical and business challenges. Many applications do not cope with PKI as expected. We are looking into ways to scale CRL dissemination across all members of CAUDIT PKI. We expect that existing business processes will need to be re-evaluated and possibly new processes will need to be in place before this project is taken into production.

We have finalized the Pilot Phase in which draft Certificate Police/Certificate Practice Statement have been developed and feedback sought from the participant universities and other PKIs from around the world. This phase also included the development of a PKI test environment in which CA certificates where issued to participant institutions that in turn issued end user certificates.

Preliminary interoperability tests included encryption and signing of emails at a client level, browser client authentication, online validation of certificates, server side certificates and CRL and OSCP implementations.

At the time of writing this paper we have entered the Pre-production Phase in which we are further developing the draft CP/CPS and pursuing the avenues to include the Root CA into web browsers. We are investigating Higher Education requirements for authorization certificates including short-lived authorization certificates and, in collaboration with MAMS, we are exploring the alignment of Shibboleth into the CAUDIT PKI Federation Trust fabric.

We are however optimistic that with the continued support we have received from the CAUDIT universities participating in the Pilot Phase that we'll be able to implement an efficient PKI solution across the higher education sector in Australia.

Our phased approach has enabled us to receive support from a number of organizations, which keeps the momentum with the Higher Education Sector in Australia moving forward.

References

[ADAMS2003]	C. Adams and S. Lloyd, Understanding PKI, Addison-Wesley, 2003
[ADAMS2004]	C. Adams and M. Just "PKI: Ten years later" http://middleware.internet2.edu/pki04/proceedings/pki_ten_years.pdf
[AS4539.1.2.1]	AS 4539 Part 1.2.1 (2001) Information technology – Public Key Authentication Framework (PKAF) General – X.509 Certificate and Certificate Revocation List (CRL) profile. Standards Australia.
[AS 4539.1.3]	AS 4539 Part 1.3 (1999) General – Information technology – Public Key Authentication Framework (PKAF) - X.509 supported algorithms profile. Standards Australia.
[DIFFIE]	W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, Vol 22, No 6, November 1976
[FBCA]	Public X.509 Certification Practice Statement (CPS) For The Federal Bridge Certification Authority (FBCA) - http://www.cio.gov/fpkipa/documents/fbca_cps.pdf
[FIPS 140-]	Security Requirements for Cryptographic Modules, 1994-01 http://csrs.nist.gov/fips/fips1401.htm
[HEBCA]	X.509 Certificate Policy for the Higher Education Bridge Certification Auth (HEBCA) - http://www.educause.edu/ir/library/pdf/NET0309.pdf
[KOHNFELDER]	L. Kohnfelder, "Towards a Practical Public-key Cryptosystem", MIT Thesis May 1978
[MUCA]	Monash University Public Key Infrastructure: Certificate Practice Statement - http://www.its.monash.edu.au/security/certs/CPS_v1_1.doc
[Gutmann04]	P, Gutmann, How to build a PKI that works, 3rd Annual PKI R&D Workshop 2004
[PKCS#12]	Personal Information Exchange Syntax Standard, April 1997. Http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-12.html
[RFC 2459]	Internet X.509 Public Key Infrastructure - Certificate and CRL Profile http://www.ietf.org/rfc/rfc2459.txt
[RFC 3280]	Housley, et al. (2002) Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280. IETF Network Workgroup – PKIX. http://www.ietf.org/rfc/rfc3280.txt
[RFC 3647]	Chokhani, et al. (2003) Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. RFC 3647. IETF Network Workgroup – PKIX. http://www.ietf.org/rfc/rfc3647.txt
[VERISIGNCPS]	VeriSign Certification Practice Statement - http://guardent.com/repository/CPS2.3/VeriSignCPS2.3.pdf

Appendix A

Financial Transaction Reports Act 1988 (FTR Act)

Identification Record for a Signatory to an Account

'100 Point Check' (201)

Following are some of the checks that may be made towards the prescribed verification procedure (100 Point Check), pursuant to the *Financial Transaction Reports Act 1988* (FTR Act), for the purpose of obtaining an identification record (section 20A(1)(b)(i) of the FTR Act) for a signatory to an account. Refer to the *Financial Transaction Reports Regulations 1990* for a complete list.

Please Note: Special provisions may apply to particular signatories. Refer to AUSTRAC account opening model form 202 and to Regulations 4, 5, 6, 7, 8, 9, 10A and 10B of the FTR Regulations for more details.

How to complete this form:

- Record the points scored for the checks carried out
- Total the points scored
- In Parts A and B, record the appropriate details for the checks carried out
- In Part C, indicate if verification has or has not been achieved

The AUSTRAC Help Desk can be contacted on 1800 021 037 if you require general assistance to complete this form.

Name of Signatory	
Account Name	
Account Number	

Type of check		Tick if satisfactory	Details to be recorded
1.	**PRIMARY DOCUMENTS**	**70 POINTS**	
	NAME of the signatory verified from one of the following: • Birth Certificate • Birth Card issued by the New South Wales Registry of Births, Deaths and Marriages • Citizenship Certificate • International Travel Document: - a current passport - expired passport which has not been cancelled and was current within the preceding 2 years - other document of identity having the same characteristics as a passport (e.g. this may include some diplomatic documents and some documents issued to refugees)	☐	Provide details in A overleaf, or keep a copy of the document.
	Note: Do not score additional points for more than one document.		Regulation 4(1)(e)
2.	**Signatory is a known customer of at least 12 months standing**	**40 POINTS**	
	Note: This procedure may only be used by authorised deposit-taking institutions (ADIs), banks, building societies, credit unions or registered corporations within the meaning of the *Financial Corporations Act 1974*.	☐	Provide details in B overleaf. Regulation 4(1)(h)
3.	**NAME of signatory verified from a written reference from one of the following, signed by both the person giving it and the signatory:**	**40 POINTS**	
	• Another financial body certifying that the signatory is a known customer • Another customer who has been verified as a signatory by the cash dealer • An acceptable referee (refer to AUSTRAC Guideline No. 3 and Information Circular No. 3)	☐ ☐ ☐	Provide details in A overleaf, or keep a copy of the document.
	Note: Customer must be known for at least 12 months by any of the above		Regulation 4(1)(j)
4.	**NAME of signatory verified from one of the following (but only where they contain a photograph or signature that can be matched to the signatory);**	**40 POINTS**	
	• A licence or permit issued under a law of the Commonwealth, a State or Territory (e.g. an Australian driver's licence) • An identification card issued to a public employee • An identification card issued by the Commonwealth, a State or Territory as evidence of the person's entitlement to a financial benefit • An identification card issued to a student at a tertiary education institution	☐ ☐ ☐ ☐	Provide details in A overleaf, or keep a copy of the document.
	Note: Additional documents can be awarded 25 points (see category 8 overleaf)		Regulation 4(1)(f)
5.	**NAME and ADDRESS of signatory verified from any of the following:**	**35 POINTS**	
	• A document held by the cash dealer giving security over the signatory's property • A mortgage or other instrument of security held by another financial body	☐ ☐	Provide details in A or B overleaf, or keep a copy of the document.
			Regulation 4(1)(a)(iii)-(iv)

Form 201 - June 2003

Page 1 of 2

List of Acronyms

AA	Attribute Authority
ABUSE	Attribute Based, Usefully Secure Email
AC	Attribute Certificate
ACL	Access Control List
ARP	Attribute Release Policies
ASN.1	Abstract Syntax Notation One
B2B	Bridge-to-Bridge
BBWG	Bridge-to-Bridge Working Group
CA	Certification Authority
CAS	Community Authorization Service
CAUDIT	Council of the Australian University Directors of Information Technology
CMP	Certificate Management Protocol
CP	Certificate Policy
CPFCA	Common Policy Framework Certificate Authority
CPS	Certification Practices Statement
CRL	Certificate Revocation List
CRLDP	CRL Distribution Point
DAM	Draft Amendments
DHCP	Dynamic Host Configuration Protocol
DIT	Directory Information Tree
DKIM	Domain Keys Identified Mail
DN	Distinguished Name
DNS	Domain Name Systems
DNSSEC	DNS Security Extensions
DNV	Det Norske Veritas
DPV	Delegated Path Validation
DRM	Digital Rights Management
DSL	Digital Subscriber Line
ECC	Elliptic-Curve Cryptography
EE	End Entity
EEC	End Entity Certificates
EELA	European Commission’s E-Infrastructure Shared Between Europe and Latin America
ENUM	Telephone Number Mapping (IETF WG)
FBCA	Federal PKI Bridge Certificate Authority
FICC	Federal Identity Credentialing Committee
FIPS	Federal Information Processing Standard
FPKIPA	Federal PKI Policy Authority
GSSAPI	Generic Security Service Application Program Interface
GT	Globus Toolkit
HEBCA	Higher Education Bridge Certification Authority
HSM	Hardware Security Module

HSPD-12	Homeland Security Presidential Directive 12
IdP	Identity Provider
IDABC	Interoperable Delivery of European E-Government Services to Public Administrations, Businesses and Citizens
IP	Internet Protocol
IEEE	Institute for Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGTF	International Grid Trust Federation
ISO/ITU-T	International Organization for Standardization/International Telecommunication Union - Telecommunication Standardization Sector
KCA	Kerberos Certification Authority
KDC	Key Distribution Center
LBNL	Lawrence Berkeley National Laboratory
LDAP	Lightweight Directory Access Protocol
LHC	Large Hadron Collider
MAC	Message Authentication Code
MACE	Middleware Architecture Committee for Education
MAMS	Meta Access Management Systems Project
MAPS	Middleware Action Plan and Strategy
MIME	Multipurpose Internet Mail Extensions
MSFT-CAPI	Microsoft Cryptographic Applications Programming Interface
NCSA	National Center for Computing Applications
NERSC	National Energy Research Scientific Computer Center
NIST	National Institute of Standards and Technology
NMI	NSF Middleware Initiative
NSF	National Science Foundation
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OSG	Open Science Grid
OTP	One Time Passwords
PAM	Pluggagble Authentication Modules
PDA	Personal Digital Assistant
PDAM	Proposed Draft Amendments
PDP	Policy Decision Point
PEM	Private Enhancements for Internet Electronic Mail
PERMIS	Privilege and Role Management Infrastructure Standards Validation
PGP	Pretty Good Privacy
PIP	Policy Information Point
PK-APP	Aka KX509: X.509 Certificates via Kerberos
PKC	Public Key Certificate
PK-CROSS	Public Key Cryptography for Cross-Realm Authentication in Kerberos
PKCS-12	Public-Key Cryptography Standard Number 12
PKI	Public Key Infrastructure

PKIF	Public Key Infrastructure Framework
PK-INIT	Public Key Authentication in Kerberos
PKIX	Public Key Infrastructure X.509 Working Group
PMI	Privilege Management Infrastructure
QoS	Quality of Service
RA	Route Attestations
RADIUS	Remote Authentication Dial In User Service
RP	Relying Party
SAFE	Secure Access For Everyone
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Level
SCVP	Standard Certificate Validation Protocol
SEEM	Single European Electronic Market
SHA-1	Secure Hash Algorithm, as specified in FIPS 186-1 (also denoted SHA1)
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol (XML protocol)
SP	Service Provider
SPKI	Simple Public Key Infrastructure
SSH	Secure Shell
SSL	Secure Sockets Layer protocol
TA	Trust Anchor
TLS	Transport Layer Security
VA	Validation Authority
VOMS	Virtual Organization Membership Service
W3C	World Wide Web Consortium
WASP	Web Activated Signature Protocol
WAYF	The “Where are you from?” problem
WIP	Works-in-Progress
X.509	The ISO/ITU X.509 standard
XACML	Extensible Access Control Markup Language
XER	XML Encoding Rules for ASN.1
XKMS	XML Key Management System
XML	Extensible Markup Language
XKISS	SML Key Information Service Specification