PKI Design for the Real World

Peter Gutmann

Department of Computer Science
University of Auckland
New Zealand

pgut001@ cs.auckland.ac.nz

ABSTRACT

What would a PKI look like if it were designed for implementability and deployability rather than strict adherence to a particular theoretical or mathematical model? This paper presents and examines the results of a series of interviews in which a cross-section of experienced programmers, system administrators, and technical project managers with many years of practical, real-world experience were asked which technologies they would use to solve some of the major problems that occur in PKI implementation. The results of the interviews and various significant issues identified by them are presented and discussed. Finally, a PKI technology blueprint based on recommendations made by respondents is presented. The resulting design is noteworthy in that it is almost completely unlike the one proposed in X.509 and related standards, which would indicate that at least some of the deployment difficulties being encountered with X.509-style PKIs are due to their sub-optimal choice of technology.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection.

General Terms

Design, Security.

Keywords

PKI, certificate management.

1. INTRODUCTION

PKI users have for many years been trying to build PKIs using the methods and mechanisms described in the X.500 series of standards, and more recently profiled in an ongoing series of IETF RFCs and drafts that currently amount to over two thousand pages of text. During this time the feasibility and practicality of these technologies have been called into question as the result of numerous negative implementation experiences [1][2][3][4]. While PKI designs other than the X.509 one exist (examples being PGP [5], SPKI/SDSI [6][7] and AADS [8][9][10]), these are mostly based on different theoretical/mathematical models for handling certificates, or the use of PKI-like systems such as IBE [11][12].

NSPW 2006, September 19–22, 2006, Schloss Dagstuhl, Germany.

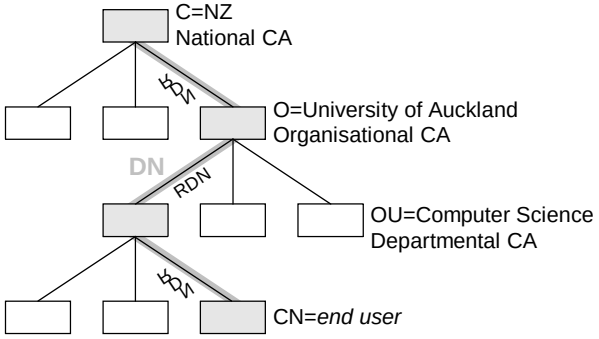

Figure 1: X.500 directory and certificate model

Although the global X.500 directory for which X.509 was designed never eventuated, PKI designers and users have had to live with the legacy X.509's origins ever since. In an attempt to determine which technologies would be the most suitable for implementing a real-world PKI, the paper therefore looks at the following question:

> *If you asked experienced programmers, system administrators, and technical project managers how they would implement certificate management, what would the technology framework for your PKI look like?*

The intent of this work is to take a cross-section of technical computer users with many years of practical, real-world IT experience and see which technologies they would select if asked to implement a PKI, or more specifically how they would solve the major problems that occur in PKI implementation. To this end the paper asks a series of "How" questions (such as "How would you store certificates") rather than "What" questions (such as "What policy would you employ for private-key handling?"). The intent is solely to determine the most practical means of solving common PKI-related technology problems without trying to address policy and legal issues, which are best left to upper management, lawyers, and lawmakers. Some of the technology issues that need to be addressed are relatively obvious and sufficiently well-known that they have their own names, examples being the "Which directory?" problem and the "Which John Smith?" problem. The background of the users involved in the study included medical, government, and university IT, telcos, financial institutions, and software houses, providing a good cross-section of potential PKI users.

The remainder of this paper is organised as follows. Section 2 covers the method used to interview respondents and the questions they were asked. Section 3 presents the raw results obtained from the interviews, and section 4 analyses various common issues that showed up in the information provided by

respondents, as well as presenting additional (unsolicited) feedback provided in areas that respondents saw as being a source of significant implementation or deployment difficulties. Finally, section 5 presents a PKI implementation blueprint based on the recommendations made by respondents. Readers are advised that any odd-sounding portions of the text are probably manifestations of the author's antipodean sense of humour.

1.1 Motivation

It has by now become generally acknowledged that PKI is extremely expensive [13], difficult to deploy [1], hard to use [14], and in many cases almost completely ineffective in providing any real security [15]. So why a paper on PKI design when better, cheaper, easier-to-use, and far more effective mechanisms are readily available?

Because a not inconsiderable number of IT staff, for various political and bureaucratic reasons, are required to implement and deploy PKI, and like the dying gasps of OSI this phenomenon is likely to continue for some time yet. The intent of this work therefore isn't to revolutionise the world with another new PKI design (it's almost certainly too late for that), but to ease the pain of those still required to work with some form of PKI.

It has been observed that PKI is like opera, you pour in money and PKI comes out. When you stop pouring in money, the PKI stops (for this reason the US Federal Bridge PKI has been likened to a US government charity). In many cases the best alternative to PKI is to do nothing at all, or at least nothing too different from what we've already been doing. Mechanisms like RADIUS, EAP, and Kerberos are widely deployed and relatively effective, particularly when compared to PKI-based alternatives. Key continuity [16], a simple, effective site authentication mechanism best known through its use in SSH, is an effective way of achieving what SSL's expensive and complex PKI tries to do with little actual effect.

As one paper on authentication mechanisms observes, Kerberos was originally designed with three heads: authentication, authorisation, and audit (AAA). Kerberos had taken care of the first and was halfway through the second "when public-key cryptography came along. Then we all disappeared down a rabbit hole for twenty years, and we've just emerged now. The effect of public key was that we went back and did authentication again, but never re-did authorisation, or did audit at all" [17].

It's interesting to note that a panel evaluating security-related protocols for conformance to a set of AAA requirements created by pretty much a who's who of the computer and communications industries [18] never even considered PKI because it couldn't come close to meeting the AAA requirements (PKI architects are still too busy playing with authentication to have gotten around to authorisation or audit). So the simplest, most effective PKI may well be none at all. As the WOPR computer in the movie Wargames concludes, "The only winning move is not to play".

2. SURVEY METHOD

A series of interviews was conducted in which a number of programmers, system administrators, and technical project managers were asked various questions about their choice of technology for implementing certificate management. In order to avoid problems with the type of self-selecting survey often posted to Usenet newsgroups and web pages, the respondents were explicitly selected from among the employees of various companies and organisations rather than by soliciting responses from volunteers.

This proved to be somewhat more difficult in practice than it at first appeared. The straightforward approach of { obtain permission from a company to interview their IT people, work up and down the rows of cubicles } led to problems because many of the people working in the area weren't necessarily capable of architecting a solution to the particular problems presented by the survey (this is covered in more detail in the discussion of the results given in section 4.4). For example a network admin whose specialty was managing complex router and server configurations wasn't necessarily able to provide useful input on PKI design.

The lack of useful results from first attempt at a breadth-first survey led to a second approach, which was to ask managers in each organisations which people they'd consider best able to respond to the survey and then to only ask them. This somewhat more targeted approach considerably improved the quality of the answers, since now only the people with the expertise required to provide useful input (at least in the opinion of their managers) were being consulted.

The respondents were instructed to choose the technology they would use if they were responsible for implementing, deploying, and supporting the PKI. The intent of this was to emphasise the fact that they were being asked to design a practical, real-world system rather than one based on bleeding-edge or experimental technology. In addition they were told that there were no right or wrong answers to questions, and in particular that "I don't know" was a perfectly valid answer since it indicated that the question corresponded to a particularly tough problem. Despite the fact that most of the respondents were male primates, several of them did admit to not knowing the answers to various questions.

Selecting an appropriate survey methodology proved to be by far the most difficult part of the overall process, requiring both careful planning and some experimentation to try and find the best solution. The author is open to suggestions for techniques to apply to future work.

2.1 Bias Removal

The author has for some time been exposed to user feedback on preferred PKI implementation technology as a side-effect of involvement in numerous PKI implementation and deployment operations. This lead to concerns that the questions might (inadvertently) be phrased in a manner that influenced respondents into replying in a manner that matched the author's existing experience with users.

In order to ensure that the results weren't biased because of the way the questions were phrased, they were sent to representatives of every major and some minor PKI theologies for comment. The intent of this solomonic bias-removal process was to ensure that none of the theologies could later claim that they had been unfairly excluded from consideration because of the way that a particular question was phrased. For example instead of asking about revocation checking (which would bias the results towards a CRL-style solution), the question was phrased in terms of freshness/validity checking, which allowed

for a variety of answers, including CRLs. Feedback from the PKI representatives was applied to the initial questions (although almost no changes were deemed necessary), leading to the final survey questions given in the next section.

2.2 Survey Questions

Before being asked the questions themselves, the respondents were asked about any existing exposure to PKI that might bias the response. The actual questions that followed were broken down into five groups covering enrolment, identification of certificates, storing and obtaining certificates, checking certificate validity, and a miscellaneous section. Although these questions don't represent an exhaustive enumeration of all possible PKI technology issues (it's unlikely that any finite question list can), they cover all of the major areas and, in their answers, provide a good solid technology framework on which to build a practical PKI.

The identification, storing and obtaining certificates, and validity checking question groups were presented in that order because the results from one group tended to affect the following ones. For example a choice of domain names as a certificate identifier early on would lead logically to the use of a DNS-based certificate distribution mechanism in the following question group, and a DNS-based validity checking mechanism in the group which followed that.

Questions within each group were ordered logically and in some cases anticipated answers to earlier questions. For example experience with users indicated that email addresses were popularly used to identify certificate owners, so one of the questions that followed the initial identification question was how someone (or something) that didn't have an email address would be identified. Finally, two questions about the cost and/or complexity of the solutions given in previous questions were added in order to discourage impractical and extravagant schemes.

Enrolment

1. How do people sign up?
2. Can this be bypassed/made less labour-intensive?

Identification

2. How would you identify certificate owners?
3. What if there's more than one John Smith?
4. What if they don't have an email address?
5. How would you get an ID which is globally unique?

Obtaining certificates

6. How would you store a collection of certificates?
7. How would you access a collection of certificates?
8. How would you locate a collection of certificates?

Freshness/validity checking

9. How would you check the validity/freshness of a certificate?
10. How would you handle the cost of doing this?

Miscellaneous

11. How would you check that an operation was valid at a given time in the past?

Respondents who were particularly quick to leap in with an answer were asked to justify their choice, mostly by being reminded that they would be responsible for implementing and supporting their choice of technology. As mentioned earlier, the intent behind the explicit step of making users eat their own dogfood was to weed out technology that they were only aware of through trade magazines or vendor literature in favour of technology that they were familiar with and believed was practical to deploy and maintain in the field.

3. RESULTS

When it came to exposure to existing technology, it proved almost impossible to find anyone who hadn't been exposed to PGP and (to a lesser extent) ssh. In an attempt to locate someone who hadn't been subject to these potential sources of bias, the net was cast wider and wider, eventually landing non-technical managers who, unfortunately, weren't able to provide answers to most of the technical questions. However, since both PGP and ssh have little in the way of infrastructure, the results were probably not affected by preconceived notions of how a PKI was supposed to be implemented.

Only one respondent had had any significant exposure to X.509, and his responses to the questions differed markedly from the other responses (details are given further on). None of the respondents were aware of other PKI systems such as SPKI/SDSI or AADS, or PKI-like systems such as IBE.

The remainder of this section presents the responses to the questions provided by users. The section that follows this one contains an analysis of the results.

3.1 Enrolment

This group of questions was seen by most as being policy rather than technical questions. As a result, the non-technical managers were able to answer them while most of the remainder saw it as being an issue that was best handled by others. This position wasn't taken because they were trying to avoid work or responsibility, but because they were used to such decisions being made by management, the clients, or other external agents. In other words, they were unfairly being asked "What" rather than "How" questions.

The responses that were provided tended to be domain-specific. One respondent who worked for an organisation with a large number of clients suggested a paper-mail-based enrolment system in which the (known to the organisation) clients were enrolled using traditional paper documents, with all certificate details being filled out from the organisation's records. Another respondent with a healthcare background suggested creating certificates based on existing health records. The responses tended to leverage existing mechanisms for use in the enrolment process as much as possible, both for ease of deployment and because they represented established business practice and were therefore likely to be looked on favourably if a dispute over enrolment details arose. The main design goals of these schemes appeared to be a combination of ease-of-use and risk avoidance.

3.2 Identification

Almost all users immediately suggested the use of an email address as the primary identifier for a certificate. One or two users suggested domain-specific identifiers, for example in healthcare the patient ID or medical registration number might be used as the identifier, and an organisation with known clients such as a financial institution would use the clients' account number.

In cases where the certificate owner didn't have an email address, various (obvious) solutions such as the DNS name or IP address were suggested (an alternative way of phrasing question 4 was "What if the certificate owner is a printer?" when the respondent had immediately suggested using an email address as the answer to question 3). Other responses included MAC and IPv6 addresses (the latter because they included more information than IPv4 ones). One respondent provided a nice generalisation to "the name you saw when you first encountered the device", so that if the printer mentioned earlier appeared on the local network as "Wallet Buster 300" (the name used in one department for a photographic-quality printer with a particularly high per-page printing charge) then it would also be identified in the certificate as the "Wallet Buster 300". Although names such as these were meaningful only in the local context, the fact that the identified item was only visible locally made this issue irrelevant (the user had thus independently rediscovered SDSI/SPKI's local names).

Some users had problems coming up with an identifier. One user suggested using a personal name or company name and asking the user to select a certificate if several matching ones were found, but wasn't able to provide a solution that would be amenable to automated processing. It's probable that they misunderstood the nature of the question, however the author was reluctant to provide further prompting for fear of influencing the results.

Most users immediately suggested the use of a GUID (Globally Unique ID) as a unique value to identify certificates. Two users weren't aware of GUIDs but described (in some detail) an identifier built up in much the same way as a GUID.

3.3 Obtaining Certificates

As with the email addresses, almost all users immediately suggested using a database as the certificate storage mechanism, seasoned to taste ("Anything but Oracle", "ODBC, because it's on every Windows machine", "Whatever the company's using at the moment", "Oracle, DBAs are a dime a dozen", and so on). One respondent suggested LDAP "because that's what you store certificates with" but was unable to provide further information, and had no actual LDAP experience. Another respondent (the one with X.509 experience) also suggested using LDAP "because that's what you use", but immediately followed it up with the comment that it wouldn't work in his organisation because users typically occupied multiple roles (leading to a multitude of possible entries in the directory, which could change several times a year), and the only way they had found to resolve the problem was to use the directory as a flat database. Another respondent who was a strong OSS advocate initially suggested "Whatever I can find on SourceForge", but eventually settled on a database like most of the others because of the availability of open-source solutions such as Berkeley DB and MySQL.

The unanimous consensus for the access mechanism was HTTP ("That's reality"). One user commented that they'd really prefer XML and SOAP if it were a bit more widespread, and another user suggested ODBC as another possibility, while acknowledging that there would be some problems due to it being a mostly Windows-only solution and having some problems with Internet traversal. Several users expanded their basic answer to address reliability and scalability issues ("We've had zero downtime for our web pages in the last year (except for link outages) even though individual servers have occasionally gone down"). They provided sketches for web architectures to handle almost any eventuality, based on their existing experience with web technology.

Most respondents suggested fetching the certificate from what can be generalised to "the most obvious place". For example if the certificate belonged to someone at a given organisation, they would query the organisation's web server. If they needed a certificate for someone at their own organisation, they would query their main corporate file or web server. If they had an email address, they would query the corresponding web server, for example `www.hotmail.com` for a Hotmail email address.

Some of the respondents who worked for organisations with known clients indicated that their (custom) client software would be configured when it was built or deployed so that it would talk to their own servers. For example one user's organisation had an application that used a particular EDI protocol to talk to a given server, so certificate retrieval would be piggybacked on top of this existing mechanism in the form of an HTTP-style `GET` using EDI instead of HTTP. One respondent also suggested using the DNS as a certificate storage mechanism, but quickly decided that it wouldn't work for much of the standard reasons that DNS-based certificate storage has been regarded as impractical. Two users suggested the possibility of a certificate search engine that worked like existing web crawlers and indexers, extracting certificate information and providing a single portal from which multiple disparate certificate stores could be accessed.

Finally, several respondents commented that if all else failed, users would have to manually set preferred server URLs in the same way that they set preferred home pages in web browsers.

3.4 Freshness/Validity Checking

As with the previous responses, the almost unanimous response to the question of validity checking was to use the certificate store in the manner of a trusted directory. Freshness and validity checking would be performed through a simple fetch operation, with the result being either a known-good certificate or some form of error indication. This goes back to the original 1970s concept of public-key distribution in which keys were to be held in public directories or key distribution centres (KDCs) that handed out only known-good keys in response to queries [19]. The user with X.509 experience suggested using CRLs, but immediately followed it with the observation that they didn't really work, and something better would have to be found.

Since this question required a bit more information than the basic "Use HTTP"-type response to question 7, users provided a fair bit of detail on the operations involved. For example one

respondent suggested communicating a checksum (meaning a cryptographic hash) rather than the full certificate to save bandwidth, and several used the GUID (the unique ID from question 5) to fetch the certificate. What the respondents were in effect describing was a form of distributed hash table, a data structure capable of answering the question "Is element e in set S?" [20]. Other respondents added use-by dates to certificates to indicate the interval after which it should be re-fetched from the certificate store (again, an independent rediscovery of a SDSI/SPKI concept), or suggested the use of HTTP-type cache control mechanisms that served a similar purpose.

Interestingly, none of the respondents considered the further refinement of using something like Diffie and Hellman's original Public File approach [21], which sidesteps the need for certificates altogether. Another approach, proposed by Davies and Price in the late 1970s in which a CA (or more specifically its predecessors at the time, arbitrators and key registries) provided a dispute resolution mechanism to relying parties by issuing an interactive certificate attesting to the validity of a key in the context of a particular transaction [22], was also not considered by respondents. This was probably a side-effect of the way in which the questions were structured, since they presuppose the need for certificates. A further set of tests with a fresh set of users would have been necessary to resolve this issue, however it wasn't considered productive since the goal was to determine how to make existing PKI technology more practical and not to design yet another PKI-alternative.

As with questions 1 and 2, most of the respondents regarded question 10 as being a policy issue and therefore someone else's problem ("That's beancounter material"). The main motivation for adding this question, as with question 2, was to discourage excessively extravagant solutions in the answers to the previous question.

Those who did answer this question suggested a variety of approaches such as a multi-tiered charging structure similar to the pricing schemes used by ISPs and for web hosting in which different levels of service and usage were billed in different ways. Other suggestions were to use per-query charging, to specifically charge the relying party rather than the certificate owner, or (in recognition of the fact that charging for queries would discourage use) the use of cost-sharing schemes to avoid one party carrying the cost while another party obtained all the benefits.

3.5 Miscellaneous

Again as with earlier responses, the unanimous response to the historical-query question was that the certificate store should maintain audit logs of certificate histories and use those to resolve historical queries. This is a logical extension of the distributed hash table-like mechanism from the previous section in the form of a persistent authenticated dictionary, a data structure capable of satisfying the extended query "Was element e in set S at time t?" [23]. Since auditing was built into most databases and the certificate store was the ideal place to maintain this information, the consensus was that this was a job for whatever or whoever was managing the certificate store.

As with question 9, some respondents provided a fair amount of detail on the operations involved. For example one user suggested charging for the length of storage of historical information, and another user came up with the novel idea of storing historical information for previous certificates in the current certificate, so that anyone obtaining an end entity's current certificate (via the mechanism from question 9) could also use it to answer historical queries. While this is in theory impractical due to bandwidth considerations, in practice having a 2Kb vs. 1Kb certificate would make no difference with a PC-based application, affecting only highly constrained devices such as smart cards.

3.6 Summary of Results

A summary of the results or the survey, which contrasts the approach provided by the X.509 standards (which represent the most commonly-used PKI blueprint) with the approach suggested by programmers, system administrators, and technical project managers, is shown in Table 1.

Table 1. Survey Results Summary

	X.509	Survey response
Identifier	X.500 DN	email address/DNS name/IP address
Unique ID	X.500 DN	GUID
Storage	X.500 directory	Database
Access	LDAP	HTTP
Validity check	CRL	Repository presence check
Historical query	(Timestamping)	Authority records

4. DISCUSSION

This section analyses the responses from users given in the preceding section. The major trends that were apparent in the responses are presented in their own subsections, with miscellaneous comments gathered at the end.

4.1 Consistency of Results

The most remarkable thing about the results presented in the previous section is the fact that almost all of the respondents agreed on one particular solution to the problem presented by each question. So consistent were the answers (somewhat akin to finding a straight line on a double-log graph) that the author felt it necessary to locate and question further respondents, leading to the eventual extension of the survey to non-technical managers as described in section 3.

The fact that the respondents had been specifically instructed to select the technology they felt was the most practical and feasible probably helped produce this consistent result. A few of the respondents were later informally asked what they would have suggested if they had been allowed to choose any technology (no matter how impractical), and came up with very different answers such as CORBA (although this was suggested as a joke by someone whose employer had a customer with a particular obsession with CORBA).

Almost all users suggested using a GUID or GUID-equivalent as a unique identifier for a certificate. This came as something of a surprise to the author, since the conventional approach (at least in PGP, SPKI/SDSI, and recently-issued X.509v3 certificates) is to use a value derived from the certificate's public key. When

asked why they hadn't used the public key, the users responded that they hadn't considered it, but that that would also work. It appears that the widespread acceptance and use of GUIDs as general-purpose unique identifiers lead to this being the immediate choice for unique certificate identifiers as well.

4.2 Universality of WWW Technology

The penetration of the web into all aspects of computer use was very obvious in the responses. All respondents regarded HTTP as the universal glue to tie the PKI together. The use of web technology went far beyond the basic transport mechanism. Users suggested the use of HTTP cache-control mechanisms to handle certificate re-validation, web search engine technology to make locating certificates when their exact location wasn't known easy, and the use of various standard reliability and scalability-enhancing techniques such as round-robin DNS to address availability concerns.

The ability of the respondents to design sophisticated web-based solutions without too much effort reflects the extensive practical expertise available in this area, backed up by a large number of tools (both commercial and open source, to suit all tastes) and background technical information. For example the level of scalability planning extended beyond the basic bullet-point-on-a-PowerPoint-slide level to "we'll use these servers and this software because we've done it before and we know that it works", a good indication that the resulting design would be practical under real-world conditions.

4.3 Key Management Issues

Several users expressed concern about the complexity involved in the key and certificate setup process. One user proposed a certificate-vending-machine type mechanism for which the only user interface task consisted of entering some form of authenticator and clicking a button labelled "Click here to generate a key and obtain a certificate". This was to be implemented using an HTTPS interface to the CA, submitting the public key and reading the resulting certificate back from the certificate store. Another user suggested "look at how *browser-name* does it and then do the exact opposite", a reference to the complexity of the browser-based enrolment process used to obtain certificates from some public CAs [24].

Yet another user, from a healthcare background, commented that many of their users would require per-site (rather than per-user) keys, since doctors expected many of the operations requiring the keys to be performed by nurses or administrative assistants, and keys were expected to be associated with roles such as "Duty doctor" (covering several GPs and assistants) rather than a particular individual. The inevitable result of this inability of per-user certification to match existing practice was that "they'll take whatever doctor turns up first in the morning's key and use that for the rest of the day", an observation arising from many years of experience with equivalent (non-public-key-based) solutions.

Other users also expressed concern about the enrolment/setup process. One user, working for a large organisation with known users, commented that a one-click enrolment process ("assuming Amazon hasn't patented that too") would be an absolute requirement, with an automated phone call-back being used to confirm that the user had indeed required the certificate ("cumbersome but functional").

This is clearly an area that needs further study to determine how low-impact the enrolment process can be made while still satisfying various legal concerns. Without any rigorous (and workable) framework for this area, users are coming up with solutions such as the alarming practice of having the CA generate the end user's private key and then sending it to them via email, either in plaintext form or with the password attached [25].

4.4 Miscellaneous

The fact that some respondents worked in a particular area influenced their replies to policy (rather than purely technical) questions. For example people working for organisations with clients or members tended to think of end entities in that role, with the organisation managing certificate issuance by taking advantage of its existing knowledge of users.

Several respondents spontaneously evolved SPKI/SDSI-type concepts such as local and global naming and timed re-validation of certificates, even though they had no previous exposure to PKI design. This mirrors experience with psychological studies of non-programmers who spontaneously evolved programming-language-like constructs such as control statements when they were asked to create descriptions of algorithm-like tasks [26].

The innate tendency of system administrators and technical managers to build in disaster-planning has already been pointed out elsewhere [27]. This was also apparent in many of the architectures laid out by respondents, with multiple development paths being possible in order to arrive at the final goal (one user summed it up with "Postgres if possible, Perl, Apache, and MySQL if they need it by Monday"). This was further reinforced by the fact that a number of respondents planned in future extensibility to handle scalability and reliability issues. The initial survey requirement that users would be required to eat their own dogfood appears to have been a powerful influence in both the choice of technology and the overall architecture.

Another fact that became apparent from the replies to the questions (although it's not directly relevant to this paper) was that the respondents' job position often matched their ability to provide answers to the questions. For example respondents who were working as programmers often had difficulty in architecting solutions to some of the more complex problems like identification or billing, while respondents with a similar amount of work experience who had been migrated into technical project management had little difficulty in this area. Although this has little effect on the results presented here (the respondents were chosen from a general cross-section of technical users without concentrating on one particular area), it's interesting to note that people seemed to have drifted into the job role they were most suited to, at least as determined by their ability to answer the survey questions.

5. PKI IMPLEMENTATAION BLUEPRINT

Using the results presented in the preceding sections, we can now look at how a PKI might be implemented with a particular goal of using the most practical real-world technology in order to increase the chances of successful deployment. As was already mentioned earlier, this implementation blueprint covers

only the "How" aspect and leaves issues such as policy and legal concerns to the appropriate entities.

The basic certificate-management system is built on top of the database of choice, and uses an HTTP (or HTTPS) interface for communication. Certificates are generally identified by user name (CommonName in X.500 terminology) and email address, with alternatives such as an account number, IP address, or device name being used where this isn't feasible.

Certificate issue is handled via a minimal one-click interface, which can be accomplished on most systems in a reasonably automated manner by reading the user name and email address from the user account information (for example the GCOS field under Unix or the Windows user information), and using it to populate the certificate request. The generated certificate is obtained by fetching it from the certificate store.

The process of obtaining a certificate is also the mechanism used for freshness/validity checking, with the certificate store returning only known-good certificates. Historical queries and similar issues are handled through the standard auditing and accounting mechanisms built into the database, which are used to track certificate additions and deletions and similar operations.

The basic mechanisms presented here can (obviously) be garnished to taste. For example some CAs may require a private-key proof-of-possession operation before issuing a certificate, which may require a two-stage process to be used when requesting a certificate. Potential implementers should however bear in mind that the goal of this work was to determine how to build a practical, deployable PKI. A workable (but not quite theoretically perfect) practical PKI is still better than theoretically perfect vapourware.

A number of CAs and PKIs are in fact already employing some of these mechanisms, although their use is often hidden from public view. For example many large public CAs use (and an unknown number of non-public ones) already use databases as their underlying certificate store. As an example of this practice Verisign, the world's largest CA, is built on top of Oracle, with LDAP being merely a shim on top of the database [28] (Verisign used the same approach for their "LDAP" Whois service when they found that LDAP wasn't up to the task [29]). Many other CAs have taken a similar approach, with Oracle, Ingres, and MS SQL Server being popular certificate store solutions.

6. CONCLUSION

This paper has presented the results obtained from asking a number of technically skilled users with extensive practical IT experience how they would implement a certificate-management system. The resulting design is noteworthy in that it is almost completely unlike the one proposed in X.509 and related standards, although it does bring in some concepts that also appear in SDSI/SPKI. This would indicate that at least some of the deployment difficulties being encountered with X.509-style PKIs are due to the sub-optimal choice of implementation technology. To address this problem, the paper proposes a new certificate management technology blueprint based on information in the responses from users. This blueprint makes use of widely-utilised, mature technology and the extensive experience that users have working with it.

7. ACKNOWLEDGEMENTS

The author would like to thank various PKI theology representatives for providing feedback/bias removal on the questions asked, Paul de Bazin, Tony Bryant, Tom Bowden, Nick Brooker, Russell Fulton, Paul Kendall, Suad Musovich, Edwin Ng, Steven Perreau, Steven Robb, Raymond Sellars, Chris Stephens, Russell Street, Clifford Wilson, and Stuart Woolford for putting up with the questioning, and NSPW attendees for feedback and comments on the paper.

8. REFERENCES

[1] "Advances and Remaining Challenges to Adoption of Public Key Infrastructure Technology", United States General Accounting Office report GAO-01-277, February 2001.

[2] "Solution and Problems: (Why) It's a long Way to Interoperability", Jürgen Schwemmer, *Datenschutz und Datensicherheit*, **No.9**, **2001** (September 2001).

[3] "Prime-Time Player?", Leo Pluswich and Darren Hartman, *Information Security Magazine*, March 2001.

[4] "PKI: An Insider View", Ben Rothke, *Information Security Magazine*, October 2001.

[5] "OpenPGP Message Format", RFC 2440, Jon Callas, Lutz Donnerhacke, Hal Finney, and Rodney Thayer, November 1998.

[6] "SPKI Requirements", RFC 2692, Carl Ellison, September 1999.

[7] "SPKI Certificate Theory", RFC 2693, Carl Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian Thomas, and Tatu Ylönen, September 1999.

[8] "PKI Account Authority Digital Signature Infrastructure", Anne Wheeler and Lynn Wheeler, `draft-wheeler-ipki-aads-01.txt`, 16 November 1998.

[9] "Account-Based Secure Payment Objects", ANSI X9.59 draft, 28 September 1999.

[10] "CONSEPP: Convenient and Secure Electronic Payment Protocol Based on X9.59", Albert Levi and Çetin Kaya Koç, *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01)*, December 2001, p.286.

[11] "Identity-Based Cryptosystems and Signature Schemes", Adi Shamir, *Proceedings of Crypto'84*, Springer-Verlag Lecture Notes in Computer Science No.196, p.47.

[12] "Identity-Based Encryption from the Weil Pairing", Dan Boneh and Matthew Franklin, *Proceedings of Crypto 2001*, Springer-Verlag Lecture Notes in Computer Science No.2139, p.213

[13] "PKI Billion-Dollar Boondoggle?", *Information Security Magazine*, February 2004, p.14.

[14] "In Search of Usable Security: Five Lessons from the Field", Dirk Balfanz, Glenn Durfee, Rebecca Grinter, and D.K. Smetters, *IEEE Security and Privacy*, **Vol.2**, **No.5** (September/October 2004), p.19.

[15] "Hardening Web Browsers Against Man-in-the-Middle and Eavesdropping Attacks", Haidong Xia and José Brustuloni, *Proceedings of the 14th international conference on the World Wide Web (WWW'05)*, May 2005, p.489.

[16] "Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable", Simson Garfinkel, PhD thesis, Massachusetts Institute of Technology, May 2005.

[17] "Anonymous Authentication (Transcript of Discussion)", Bruce Christiansson, Proceedings of the 12th International Workshop on Security Protocols (Protocols'04), Springer-Verlag Lecture Notes in Computer Science No.3957, April 2004, p.306.

[18] "Criteria for Evaluating AAA Protocols for Network Access", RFC 2989, Bernard Aboba, Pat Calhoun, Steven Glass, Tom Hiller, Peter McCann, Hajime Shiino, Glen Zorn, Gopal Dommety, Charles Perkins, Basavaraj Patil, David Mitton, Serge Manning, Mark Anthony Beadles, Pat Walsh, Xing Chen, Sanjeevan Sivalingham, Alan Hameed, Mark Munson, Stuart Jacobs, Byung-Keun Lim, Brent Hirschman, Raymond Hsu, Haeng Koo, Mark Lipford, Ed Campbell, Yingchun Xu, Shinichi Baba, and Eric Jaques, November 2000.

[19] "Cryptography: A New Dimension in Computer Data Security", Carl Meyer and Stephen Matyas, John Wiley & Sons, 1982.

[20] "Chord: A Scalable Peer-to Peer Lookup Protocol for Internet Applications", Ion Stoica, Robert Morris, David Liben-Nowell, David Karger, M.Frans Kaashoek, Frank Dabek, and Hari Balakrishnan, *Proceedings of ACM SIGCOMM 2001*, August 2001, p.149.

[21] "New Directions in Cryptography", Whitfield Diffie and Martin Hellman, *IEEE Transactions on Information Theory*, **Vol.22**, **No.6** (November 1976), p.644.

[22] "Security for Computer Networks : An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer", Donald Davies and W.Price, John Wiley and Sons, 1984.

[23] "Persistent Authenticated Dictionaries and their Applications", Aris Anagnostopoulos,Michael Goodrich, and Roberto Tamassia, *Proceedings of the 4th International Information Security Workshop (ISW'01)*, Springer-Verlag Lecture Notes in Computer Science No.2200, October 2001, p.378.

[24] "Plug-and-Play PKI: A PKI Your Mother Can Use", Peter Gutmann, *Proceedings of the 12th USENIX Security Symposium*, August 2003, p.45.

[25] "Lessons Learned in Implementing and Deploying Crypto Software", Peter Gutmann, *Proceedings of the 11th Usenix Security Symposium*, August 2002, p.315.

[26] "What non-programmers know about programming: Natural language procedure specification", Kathleen Galotti and William Ganong III, *International Journal of Man-Machine Studies*, **Vol.22**, **No.1** (January 1985), p.1.

[27] "Apropos: Re-Routed Packets", Tina Darmohray, *;login*, **Vol.26**, **No.8** (December 2001), p.3.

[28] "Extranet Directory Delivers Digital IDs for Millions of Customers", `http://wp.netscape.com/solutions/business/profiles/verisign.html`.

[29] "Replacing the Whois Protocol", Andrew Newton, *IEEE Internet Computing*, **Vol.10**, **No.4** (July/August 2006), p.79.