Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Common issues in PKI implementations - climbing the "Slope of Enlightenment"

This paper is an attempt to go beyond the many conceptual papers published about Public Key Infrastructure (PKI) and look at the actual problems experienced when implementing it. In particular, it looks at issues connected with the design and roll-out of large scale, identity PKIs. The issues chosen and the examples used have been sourced from real-life experience, as well as from public records of two current large-scale implementations of Identity cards incorporating PKI: The Common Access Card currently being rolled...

Copyright SANS Institute
Author Retains Full Rights

AD

Beating the IPS?

Common issues in PKI implementations - climbing the "Slope of Enlightenment"

GSEC Practical v.1.4b, July 2003
Angela Keith

Abstract

This paper is an attempt to go beyond the many conceptual papers published about Public Key Infrastructure (PKI) and look at the actual problems experienced when implementing it.

In particular, it looks at issues connected with the design and roll-out of large scale, identity PKIs.

The issues chosen and the examples used have been sourced from real-life experience, as well as from public records of two current large-scale implementations of Identity cards incorporating PKI: The Common Access Card currently being rolled out to United States Department of Defense staff and the Estonian ID card.

Note to the Reader

If you are new to Public Key technology and infrastructure, you may want to get some background information before reading this paper.

Some suggested reading material:

Introduction to PKI, ArticSoft, 2 February 2003
http://www.articsoft.com/wp_pki_intro.htm

Reference on Business issues:
http://www.pkiforum.org/pdfs/deployment_paper1.pdf

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Table of Contents

Abstract 1
Note to the Reader 1

Introduction 3

Brief overview of the sample PKIs 4
Department of Defense Common Access Card 4
Estonia ID Card 5

Common Issues with PKI Implementations 6

1. Commercial Off-The-Shelf (COTS) versus Customised applications 6
2. Token logistics 8
3. Network issues - Traffic 10
4. Network issues - Encryption 11
5. Email address in certificate 12
6. Availability and storage of reliable user information 12
7. Certificate Validity Checking 14
8. PIN management 14
9. Encryption challenges 15
10. User acceptance 15
11. Legal & regulatory 17
12. Archiving/historic verification 18

Conclusion 19

References 20

© SANS Institute 2003, Author retains full rights.

Introduction

In the terminology of The Gartner Group's 'hype cycle', Public Key Infrastructure (PKI) made its journey from the "Peak of Inflated Expectations" of the mid- to late 90s to the "Trough of Disillusionment" in the last few years.

The decline from the height of the hype was evident through many critical reviews - even by early evangelists such as Bruce Schneier, author of the PKI bible *"Applied Cryptography"* (1995), who expressed serious doubts that PKI would be the revolutionary enabler he had envisaged. In his book *"Secrets and Lies"* (2000), Schneier promoted a more holistic view of IT security and risk management.

Some early PKI adopters made costly mistakes, such as Commercial CA start-ups that had lived by the "build it and they will come" principle. Governments struggled to comprehend the implications of the use of digital signatures, formulating laws for a technology which was still largely unproven. Due to the vacuum they were created in, these often became too vague, or too prescriptive. It was difficult to fit the technical standards to the real world, and on top of that, product vendors had gone in different directions in the absence of a dominating force.

Still, the realisation that PKI could not live up to the early expectations has not killed it - there is not yet a viable alternative to the strong authentication afforded by public key cryptography. Looking around, there is enough evidence that PKI is advancing on the next step of the cycle, the "Slope of Enlightenment". There is a lot of experimentation (although commercial organisations would probably prefer not to call it that), and a lot of hard work, trying to understand where the technology best fits, and how it can be adapted without compromising security.

As with any emerging technology, and especially so with infrastructure change, this is not happening without some major hiccups. This paper is written in order to present some of the more common issues that I have come across whilst being involved in various PKI projects, in the interest of assisting others in their uphill climb on the Slope.

For illustration, and in addition to own experience, I have used two PKI projects that are fairly well documented on public websites. Both were designed to provide secure, digital identities to a large number of humans forming part of a user community and both chose to use smartcards for storing user information and keys and certificates. In other respects they had very different requirements, so the examples are not so much for comparison, as for illustration of the issues and in some cases how they were solved.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Brief overview of the sample PKIs

Department of Defense Common Access Card

The U.S. Department of Defense (DoD) started its PKI project in early 1999, but it was not until late1999 when the project was merged with the Common Access Card (CAC) project that it really took off[1]. The CAC project had been tasked with replacing old DoD ID cards with smartcards that could provide the existing 'visual' uses, plus access control to buildings and networks, and any other future applications.

The fit between the two projects allowed for substantial savings and better integration.

In April 2002, DoD had over 3.3 million service members, but was also providing benefits to another 2 million retirees and families[2]. It is the biggest employer in the United States. In June 2003, approximately 2.4 million cards had been issued[3].

The smart card used has a 32Kb chip with Java Card operating system and GlobalPlatform architecture and card management technology. As well as the chip, the CAC has a magnetic stripe for building access, 2 bar codes and a photograph. The chip can store various credentials such as X.509 certificates, biometric information and cryptographic keys. It also stores demographic information, entitlement information and various applets.

The DoD PKI consists of several hierarchies. Root Certificate Authorities (Root CAs) are operated by the National Security Agency (NSA). Subordinate CAs must be approved by the Policy Management Authority (PMA), are responsible for issuing and managing certificates, must submit a Certification Policy Statement for approval by the PMA, and operate in accordance with DoD Certificate Policies. Subordinate CAs can be managed by various DoD agencies, such as Army, Navy, Airforce and Marine Corps. There are also a number of external CAs that are approved by the PMA for use by non-DoD employees.

The DoD PKI currently issues 3 different types of certificate to CAC holders:

- An identity certificate used for authentication and non-repudiation services (e.g. two-factor login, remote access and web access to intranets).
- An email signing certificate used to create legally binding digital signatures in email.
- An email encryption certificate. Keys are generated in a Hardware Security Module (HSM) at the card issuance portal, and the private key is archived centrally for back-up purposes.

---

[1] DefenseLINK news article, July 26, 2002
[2] Dept of Defense, Introductory overview (DoD 101), 2002
[3] Govt Computer News, 9 June 2003

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Estonia ID Card

By law, possessing an ID card is mandatory for all Estonian citizens and permanent residents staying more than 1 year. In January 2002, the first smart card IDs were issued by the Citizenship and Migration Board, the aim being to make it the primary ID used in the country, and to encourage the use of digital signatures to assist the development of e-government services and e-commerce. In the future, it may also be possible to use it internationally, as the EU countries progress with their harmonization efforts.

The smart card is printed with card holder details, photograph, written signature and machine readable data. Additionally, the chip contains two sets of keys and certificates (X.509), one for authentication, the other for digital signing.

The Authentication key is intended to be used for signed & encrypted email and client authentication. The Digital Signature key is intended to be used for non-repudiation only (where a written signature would be required). The two key pairs are accessed by two different PINs.

As at the 7 July 2003, 254,818 cards had been issued[4], the total population of Estonia being an estimated 1.4 million.

In the following sections, these two PKI implementations will provide some illustrative examples for the common issues described.

[4] AS Sertifitseerimiskeskus, ID card information pages, card issuance statistics

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Common Issues with PKI Implementations

1. Commercial Off-The-Shelf (COTS) versus Customised applications

One decision that has to be made early on is whether to use Commercial-Off-The-Shelf products or whether to develop a product specifically for your application. The choice between COTS or customised products is usually one of cost versus usability.

One usability issue that has to be seriously considered is that of error messages. Unless PKI is built into applications so that it is completely transparent to users – and this is hard to do – the error messages provided are often such that users will require some understanding of the use of keys, certificates, Certificate Revocation Lists (CRLs) and directories/certificate repositories so that they can make informed decisions. For example, if the certificate repository is temporarily unavailable, users would have to determine from the error message that they may not be able to encrypt messages or verify signatures. The two problems would seem unrelated unless you knew what the repository is for and why you need to access certificates.

COTS pros and cons

The main driver for using COTS is to minimise the initial cost and the ongoing cost of ownership. It is true that using COTS products can save money, especially if the organisation already has licenses for the software and is using it. However, the cost of ownership can rise dramatically: Unless the organisation has a homogenous network environment and well documented procedures, users are bound to find out that 'this doesn't work with that'. Even if the organisation itself has implemented a standard operating environment, it will most likely want to use PKI services in its interactions with other organisations – this makes it hard to avoid incompatible PKI-enabled applications.

Although vendors have come a long way towards standardising PKI, common applications are still incompatible (e.g. Lotus Notes v. Microsoft Outlook). Most vendors will claim that their products are built on open standards, but until all components have been thoroughly integration tested, the application implementer is well advised to take these claims with some scepticism. Smart card interfaces (e.g. PKCS#11) are notoriously open to interpretation, but common protocols such as S/MIME are also frequently implemented in non-standard ways.

There is also the possibility that a COTS product vendor goes out of business or discontinues the product. When buying from a small or vulnerable company, the buyer is well advised insisting that source code is escrowed.

If the PKI is only implemented to support one or a few well-defined applications, COTS products may be perfectly satisfactory. From a security or usability point of view, COTS may not be the best solution, but it has the advantage that if flaws are found in a widely used product, there is likely to be a patch out fairly quickly.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Customisation pros and cons

The risks with customisation are no different from normal project management risks pertaining to security software: Scope, requirements and interfaces need to be well-defined, the solution thoroughly reviewed and tested for security - and usability issues need to be solved.

The risks are that the scope and cost blow out as the complexity grows. Also, the ongoing costs of maintenance could be prohibitive with a limited user base.

Often there is no choice other than to use a customised product; the functionality needed does not yet exist. The advantages are obviously that one can design a purpose-built system which maximises efficiency and minimises user impact.

Example

DoD CAC (1): In a scenario which would make a CFO pale in fright, it appears the DoD is having a difficult time getting the Defense Message System (DMS) just right. The project was always going to be massive, but the total costs and delays have caused a lot of criticism from both within and outside. According to a Federal Computer Week article dated April 21, 2003[5] a report by the Inspector General stated:

> "As of Sept. 30, 2002, DOD had spent about $9 billion in total program costs on DMS from fiscal 1990 through fiscal 2002. That amount includes investments of nearly $2.3 billion, operations and support costs of $150 million, and legacy phase-out costs of $6.65 billion.
>
> Therefore, instead of its planned savings of $453 million, "it will incur a negative return on investment of at least $266 million for general service messaging capabilities over the life of the DMS program through [fiscal] 2013," the report stated."

Military messaging is of course much more complex than PKI for a civilian environment. DMS has been rolled out and is being used, and DoD is committed to finish the project.

DoD CAC (2): In order to encourage the development of applications for the CAC, DoD have released card interface specification and a requirements specification for PKI enabled applications. With a user base of potentially 13 million[6], the CAC is an attractive market for application developers.

This approach combines some of the best features of COTS products with the flexibility of customised products for the implementer – without any investment.

[5] Federal Computer Week , 21 April 2003
[6] Smart Card Alliance DoD Case Study

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Estonian ID: The Estonian ID card project early on looked for existing applications that would satisfy their requirements, but did not find any suitable ones. They were also concerned that foreign ownership and control could endanger their country's aim of using digital signatures in daily life. Instead, they set about creating a new general purpose application called DigiDoc, based on signed XML standards (ETSI[7] standard XAdES, an extension of W3C[8] standard XML-DSIG). The libraries, specifications and applications developed are provided to the Estonian public free of charge. In order to promote the solution beyond Estonia's borders, a project was started, OpenXAdES (www.openxades.org) which provides documentation and software libraries for download to anyone who would like to develop applications that are compatible with DigiDoc and the Estonian ID card.

Again, this example illustrates that if there is a large enough market and interest in the solution, application developers may be encouraged to invest their own time & effort.

2. *Token logistics*

There are a number of advantages of implementing PKI on a hardware token, such as a smart card. These include security, portability, ease of use, and the potential to add other user specific applications and visual information.

However, the logistics of managing security tokens adds a considerable effort and cost to a PKI implementation. Typically, the PKI card lifecycle operations will include:

- Receiving cards from the manufacturer, registering and storing them
- Initialisation of the cards (e.g. replacing the transport key with the organisation's own key(s), loading applications, creating/loading public key pairs)
- Distribution to the user location
- Personalisation (loading user details, printing user details, photo, bar codes)
- PIN management (changing, resetting)
- Re-keying (creating/loading new keys & certificates when old ones expire)
- Revocation/suspension of certificates
- Destruction or recycling of cards (e.g. when a user leaves the community).

Some of these issues apply to PKI on soft tokens as well, however many of these logistical issues are specifically to the use of hardware tokens.

The extra hardware necessary, including smart card readers, personalisation equipment, and card printers also adds to the cost.

[7] European Telecommunications Standards Institute
[8] World Wide Web Consortium

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Key Generation/Certificate issuance

The point where keys and certificates are linked to their owner is a very critical point in a PKI. If a fraudulent certificate is issued by a registration officer and the certificate holder uses the certificate to commit a crime or prank, trust in the whole PKI hierarchy may be lost. The physical security requirements are high, and the registration officer, whether a person or a smartcard bureau, must be subject to strict security polices and practices.

In order to create a digital identity that can be relied upon, users need to correctly and securely take ownership of their key pair(s).

In PKI terms, this means that:

- the CA, usually through a trusted relationship with registration authority, must be sure of the applicant's identity
- a certificate request must be created, which has correct and unique user identification (or at least an auditable and permanent track leading to their identification)
- once the user has taken charge of their private authentication key(s), measures must be taken to ensure that this key cannot be used by anyone else. In particular, ensure that there is no possibility of any application or person at the issuing site (e.g. the Registration Officer) having taken a copy of the keys or the PIN protecting those keys.

Key generation must be both secure and based on a sufficiently random seed. When using crypto tokens it is preferable from a security point of view to generate at least the authentication/signing/non-repudiation key on the token, and prevent it from being exportable. An encryption key may require archiving/escrow, and so it may need to be generated centrally and in software.

Cost and the use of existing infrastructure

When implemented by itself, PKI with smart cards can become an expensive solution. In many cases, the security benefits of storing private keys securely may warrant it.

However, when smart card use is combined with other existing or desired uses such as physical access to a building, electronic purse, credit card or health benefits cards, the cost of managing cards is lower per application and the business case easier to make. If a token of some kind already exists, many of the logistical issues may already have a solution.

Example

DoD CAC: The DoD PKI did not gain momentum until the project was combined with the Common Access Card project. Token logistics still had to be solved, but there was already a cost in administering the current Defense ID, so there were savings to be gained from the consolidation.

Estonian ID: The Estonian Citizenship and Migration Board (CMB) already had a requirement for issuing ID cards, whether electronic or not. Two Estonian banks and two telecom companies founded AS Sertifitseerimiskeskus (SK), the Certificate Authority (CA) which provides the infrastructure for issuing and using the card.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

- CMB receives applications for a card - as before.
- A smart card bureau personalises the card, and SK issues certificates.
- The card holder can pick up their card from one of the two shareholder bank branches.

All parts of the process make use of existing infrastructure.

3. Network issues - Traffic

There is no doubt that the implementation of PKI will add to the network load, although just how much depends on the system architecture.

Potential additional traffic that should be considered includes:

- Certificate issuance
- Email usage
- CRLs
- Directory Replication

These are described in more detail below.

Certificate Issuance

Traffic here includes:

- Directory or database request(s) for user details and response(s).
- Certificate requests to CA

Peak loads at rollout and around renewal time must be considered. For an illustrative example, let's assume 4 million users and a 2-year certificate validity time (not considering workforce turnover or revoked/reissued certificates), the *average* number of certificates issued per week would be close to 40,000.

Email usage

Traffic here includes:

- Signed email
  - Sender: None
  - Recipient: Directory lookup for certificate/CRL
- Encrypted email
  - Sender: Directory lookup for certificate (and possibly CRL)
  - Recipient: None

In a large implementation, this quickly adds up. If 4 million users sent 10 signed & encrypted messages per day, there would be 80 million extra requests. It is possible to reduce the number of lookups by storing certificates locally and downloading/caching the CRL.

Also note that a signed or encrypted file is slightly larger than the original file, adding further to traffic volume.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

CRLs

In a large PKI, CRLs grow very big. For example, assume:

- a 4 million user community
- 15% of users have revoked certificates that have not yet expired (e.g. users that have left the community, that have lost their cards, whose keys have been compromised, whose certificate details change etc)
- 2 certificates each (signing & encryption)

CRL length = 4,000,000 x 0.15 x 2 = 1.2 million entries. If every entry is approximately 37 bytes (author's calculations) the total size is 44MB.

This will affect the amount of network traffic if each user regularly downloads the CRL, and also the amount of time in verifying a certificate. This can be eased through the use of OCSP or delta CRLs (See also Certificate Validity Checking).

Directory replication

Users need access to other users' certificates and CRLs (unless using OCSP) for encryption and verification of signatures. Certificates and CRLs are most commonly stored in LDAP directories.

In a large implementation, directories are usually replicated across the network for redundancy and availability. Although LDAP is optimised for quick lookups and replication, designing a solution that fits in with existing network security policies and bandwidth is not a trivial task. If replication is slow, there is also the risk of revocation information being further delayed.

4. Network issues - Encryption

Many organisations implement anti-virus software and content inspection on servers at the perimeter of their networks. Some have security policies that rejects or quarantines encrypted traffic.

To provide user-to-user confidentiality, messages will traverse networks with their payload hidden from inspection by virus and content checking.

Possible solutions include:

- Performing virus & content checking at the client machine
- Storing user confidentiality private keys on a secure key server which the gateway or mail server has access to
- Co-encrypting messages to a Gateway which then decrypts and checks the message before it is delivered.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

5. Email address in certificate

In order to use certificates for S/MIME signed/encrypted email, the users' email address must be in the certificate. Most people change their email addresses more frequently than the certificate. Unless a solution is built which allows users to keep the same email address over a long period, certificates would have to be re-issued every time a user changes email address.

S/MIME v.3 stipulates that the receiving application must check the *From*: or *Sender*: field in the mail header and compare it to an email address in the sender's certificate. If the check does not match, the mail application should perform another explicit check to ensure that the person who signed the message is indeed the person who sent it.

As usual, the 'devil is in the detail' when it comes to implementation.

Examples

DoD CAC: This issue seems to have created some problems. For example, the Army CPS mandated that the certificate be bound to a central mail account, located at the Army Knowledge Online (AKO) portal. However, users also have local email addresses that are administered locally.

The 'middleware' (the interface between the application and the smartcard) required the "Reply To" field in the header to match the SubjectAlternateName (where email is specified) in the certificate. Users were required to provide this address to their local Exchange administrator, so that they could enter it as the default "Reply to" address in the individual's local mail account. Users were also required to set their forwarding address for their AKO mail account to their local email address. Many users had not understood the procedure, causing many calls to the helpdesk regarding 'lost' emails, inability to encrypt etc.[9]

Estonia ID: Every ID card holder is allocated an email address consisting of *<firstname>.<lastname>_NNNN@eesti.ee*, where *NNNN* are 4 random numbers. This address is guaranteed to be available to a person for their lifetime. There is no email service behind this address – it functions solely as a relay address which users can configure to forward to up to five 'real' email addresses. These can be changed at any time, using an online service.

6. Availability and storage of reliable user information

For an identity certificate scheme, names in certificates need to be unique, meaningful - and correct. Few large user communities have all their member details in a central and accurate database or directory, and the exercise of consolidating, checking and updating all user data can turn into a massive and expensive exercise.

It is not uncommon that a PKI implementation goes hand-in-hand with the implementation of an organisation-wide online directory. While this task should not be under-estimated, it provides an opportunity to improve business processes and add security features such as Role Based Access Control.

---

[9] Picatinny Arsenal, Lessons Learned

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Implementing a central database will inevitably raise privacy issues, as such a repository presents an attractive target for hackers and identity thieves. If the repository is also available the user community and/or the public, careful attention must be paid to access control and protection of data.

The full Dname indicates where in the directory a user's certificate will be stored. In a case where the directory is not used as the source of the certificate Dname, and for the sake of directory management, the following situations should be considered:

- If the directory already contains user details, and the Dname in the certificate issued by the CA is not the same as a user's existing entry, the directory will either reject the certificate posting, or create a new node based on the certificate Dname.
- If the directory is not populated ahead, and the Dname entries are manually entered at the time of registration, the directory is likely to become chaotic and unmanageable.

For this reason, certificate Dname values are usually populated from an accurate master directory or database.

Uniqueness of names

Uniqueness in names can present a challenge in a large community where clashes between names are likely. Many PKIs use a combination of <firstname>.<lastname><unique identifier> to ensure uniqueness.

Ideally, the <unique identifier> is a number or sequence that is unique within the community, centrally issued and previously tied to the user, such as a staff id number. This would also make it more likely that the user is able to remember their Dname (Distinguished Name, X.500) in an urgent situation where they need to revoke their certificate. A manual method of allocating Dnames is prone to errors as it requires Registration Officers to check issued Dnames.

Example:

DoD CAC: Fortunately for the DoD PKI, a central database, "Defense Enrolment Eligibility Reporting System" (DEERS), had already been created in a response to a Congressional mandate to improve the management of health benefits provided to service personnel and their families. It contains over 23 million records.

Another application, "Real-time Automated Personnel Identification System" (RAPIDS) had been developed to provide a more secure method of providing IDs to eligible persons. RAPIDS workstations are deployed in DoD personnel offices worldwide, and allow for updates and additions to the DEERS database. The application and its location made it a prime candidate for issuing CACs and act as the registration client for certificate requests. During registration, RAPIDS will lookup the DEERS database for a user's details, populate the certificate request and pass it to the CA.

Estonia ID: Although no information appears to be publicly available to show exactly how user details are verified, being a government it is likely to have reliable information on its citizens from birth certificates and other official sources. Also, the need for the applicant to fill out a form with their personal details guarantees that the information is up to date.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

7. Certificate Validity Checking

CRLs have been the conventional method of providing certificate validity checking. CRLs do not scale very well as discussed earlier, but are usually kept for backward compatibility, archiving/historical verification and for use in off-line mode.

The other issue with CRLs is that they are generally issued at certain intervals of 6, 12 or 24 hours, causing a time lag from the time a certificate is revoked until it appears on the published CRL. This may present a security risk, as a certificate may verify correctly after it has been reported as compromised and revoked; (however some would argue that the time from actual compromise until the discovery and reporting of it would in most cases be a more significant lag).

The Online Certificate Status Protocol (OCSP) (RFC2560) allows a client to query an OCSP responder for the current status of a certificate. This saves searching through a large CRL and can save bandwidth if the CRL would normally be downloaded - although it may increase network traffic. Most OCSP responders are based on CRLs and thus do not solve the problem of time lag as outlined above.

Example

DoD CAC: The DoD PKI X.509 Certificate Policy specifies one CRL publishing interval for 'normal' circumstances, and another, shorter interval for cases where the revocation reason is "Key Compromise".

It allows CAs to optionally provide OCSP responders, however CAs must provide CRLs for locations that do not have online communications.

Estonia ID: According to the Estonian Digital Signatures Act, a Certificate Service Provider must provide "a method of verifying certificate validity online"[10]. AS Sertifitseerimiskeskus (SK), as an issuer of certificates for the ID card, provides three services:

- CRLs: Conventional CRLs are made available for download
- An LDAP directory containing only valid certificates: This directory is updated in real time, that is, when a certificate is revoked, it is immediately removed from the directory. This directory also allows users to look up other card holders' email addresses
- OCSP: The OCSP responder does not make use of CRLs, and has been implemented so that it reflects the actual status of its CA database at any one time.

8. PIN management

This is not specifically a PKI issue but a universal security issue. However, PIN management for smart cards may cause some new problems, as they cannot (should not) be resettable by a remote administrator. Again, it is the non-repudiation requirement – if someone else can access my private signing key, how do I know they haven't misused it?

---

[10] AS Sertifitseerimiskeskus, Estonian ID Card Whitepaper

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example

DoD PKI: Users have software for changing PINs and are planning to implement the use of biometrics for PIN reset or even as a PIN replacement[11]

9. Encryption challenges

Archiving/escrow of encryption keys

If the user, and in some cases their organisation or a Law Enforcement Agency, require access to encrypted files, they will need to have access to the private key that can decrypt it. If the private key cannot be accessed (e.g. the holder is not available, has lost the smart card or forgotten the password protecting it) this will not be possible. For this reason, encryption keys are often archived or held in escrow. In addition, this solution requires that keys are not generated in a secure token (unless export of private keys is allowed – which is not very secure).

DoD CAC: The DoD PKI archives the Private Email Encryption key. This key is generated at a central CAC Issuance Portal and is transported to the card using SSL, then injected on the smart card. Authentication and Email Signing keys are generated on the smart card.

In addition, the DMS now provides a tool for converting Signed & Encrypted Messages to Signed only, so that they can be archived without the need for the recipient's encryption key.[12]

Estonia ID: The Estonia card project took a different view of encryption; it assumed that encryption would only be used to provide secure transport for documents, and not for storage. Therefore there was no need to archive keys – it puts the onus on the user to make sure that they do not store important documents encrypted, or manage old encryption keys themselves.

10. User acceptance

The shift from paper-based trust to electronic trust can be confusing to end users, and the results in terms of usage and willingness to accept, disappointing. Users cannot "see" a digital signature, it does not appear on the printed page and they are only given a moment's choice to verify the signature on a message or a file. The correct verification of a sender's certificate is left to software to establish. Users have little choice but to trust that the public key technology has been implemented correctly.

Evidence of Identity

Users are often required to provide Evidence of Identity (EOI) in order to establish their identity beyond doubt and be issued their keys and certificates. This can prove a very hard sell – primarily because it can appear to be questioning an already established relationship.

---

[11] Activcard DoD Case Study

[12] Defense Message System, Public Web Site

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

It is often possible to find other ways of leveraging on existing relationships in the identification process - and making it more secure in the process: Most people have 100pts of Identification in their wallets, which means a stolen wallet is an invitation to identity theft. Biometrics are likely to play a part in the future, but the process need not be that intrusive. There is often a shared secret that can be used to strengthen identification, for example, a bank could make use of a verification of an existing password used by the bank customer (preferably hashed to remain obscured).

Introducing complexity in known, simple operations

From a user perspective, PKI can turn known, simple operations such as sending an email may into much more complex procedures that are prone to producing confusing error messages.

One way to identify how important these issues might be would be to run a pilot program. However this itself could introduce other considerations, such as the situation where the pilot application is not critical – users are asked to sign and/or encrypt insignificant messages, making the exercise seem pointless.

The risks of resentment

Resentful users present one of the most (if not *the most*) serious security risks there are. While most users will not consider actively carrying out sabotage on a system, they are likely to bypass or ignore security policies that get in their way or that they don't understand. This can severely undermine the improved security that a PKI is intended to deliver. Methods of reducing the likelihood of this happening include:

- Implementing well-planned and executed security awareness programs
- Carrying out consultation and obtaining buy-in from users from an early stage of the project
- Hiding as much of the complexity as possible – this may require developing custom applications or interfaces
- Clearly stating what users can do with their keys and certificates and also, what they cannot do
- Describing well defined and tested use cases for applications, producing the best possible user documentation and help files
- Ensuring that any new applications are developed to existing standards.

<u>*Example:*</u>

DoD CAC: Judging from the FAQs that are publicly available, user acceptance issues experienced were mostly caused by problems pertaining to the sending and receiving of signed/encrypted emails. For example, from the Picatinny Arsenal FAQ:

> *"Why do some email recipients report they cannot read my digitally signed message even though I'm not sending encrypted messages?"*
>
> *"Why do I get an error asking to insert the CAC, even though I have it already in?"*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

"Why am I unable to read an encrypted message that I sent to myself?"

These are fairly typical issues in a COTS product environment, and while easy to rectify, an annoyance to users.

11. Legal & regulatory

This paper has barely touched on the non-technical issues until now – yet these are perhaps the most formidable challenges. Many of them are not in the hands of the PKI implementers though, and the environment varies a lot depending on where in the world we operate. The issues are many, and beyond the scope of this document to deal with. However, to give the reader an idea of the types of issues there are, a local example:

The Australian Government strategy for the regulation of CAs, ("Gatekeeper"), is undergoing major changes in some part based on the experiences of the CA operators.

For example, RFC 2527 sets out the recommended format for Certificate Policies (CPs) and Certificate Practice Statements (CPSs). Early PKI regulators, including Gatekeeper, interpreted this standard to the letter and insist upon use of its every clause and in particular, its language. However, CPs in particular are semi contractual documents. In most jurisdictions, and in order to enter into a valid contract, the user must have a reasonable chance of understanding what that contract means. The RFC is written by technical specialists and is by no means set out in plain English. The size of the CP can also be extremely intimidating – Gatekeeper CPs that accord with the standard can be up to 70 pages.

In practice, PKIs who are nervous about the contractual holes this approach may open up, but who do not wish to deviate from the standard, have taken to issuing the CP in conjunction with a *PKI Disclosure statement.* While this does go some way to mitigating the risk of having users not understand what they've signed up to, it also adds to the already burdensome paperwork that the PKI is asking users to read.

Users could be forgiven for asking themselves why the use of their written signature doesn't require them to read the equivalents of CPs?

Also, the understanding of legal implications of transitioning from paper based to digital trust systems is evolving. As an example, non-repudiation is often thought to have been achieved through ensuring that private authentication keys are exclusively in the possession of the key holder and never accessed by anyone else. It is possible to design a PKI that implements this technically, but how can we prevent someone stealing the key holder's smart card and password and use it fraudulently?[13] .

The reality is that the law seldom takes such a narrow view – and how could it? Written signatures can so easily be forged, and yet the paper based system served us well for so long. A court case would most probably take other forensic and circumstantial evidence into account.

[13] The use of Biometrics may help prevent such fraud in the not too remote future – the subject of biometrics is beyond the scope of this paper.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example:

DoD CAC: The changing views of Non-repudiation can be illustrated by the definition in the DoD X.509 Certificate Policy (2002) of the services that the DoD PKI will provide:

> "The DOD PKI must support five primary security services: Access control, confidentiality, integrity, authentication and technical non-repudiation"

The definition of technical non-repudiation in the glossary is:

> "The contribution public key mechanisms make to the provision of technical evidence supporting a non-repudiation security service"

This is a prime example of how a more mature understanding of PKI can make it fit better in the 'real world'. The ambition to achieve Non-repudiation often has a major influence in PKI design considerations, however while it is important to keep the integrity of private authentication keys, the system must be viewed as a whole. Again, the adjustment of expectations and the balancing of security and usability is paramount.

12. Archiving/historic verification

Digital signatures need to be verifiable even after the keys used to sign have expired.

Likewise, we need to be able to verify that the certificate was valid at the time the data was signed. This means we would need to archive:

- the signed file
- the public key certificate of the signer
- the CRL that was valid at the time of signing
- a reliable timestamp to prove the accuracy of the time of signing
- the hardware environment that can run the software that was used at the time…et cetera

Think of the average lifetime of recent operating systems and compare it with the legal retention period for contracts and other signed documents, and you begin to understand that the last point requires some thought!

This issue is often put in the 'too hard' basket, but robust solutions will need to be developed for when the acid test eventually occurs. The example below shows what may be possible with some invention.

Example:

Estonia ID Card: In a clever design, the system uses standard OCSP for time-stamping. Quoted from AS Sertifitseerimiskeskus "The Estonian ID card…" white paper:

> *"SK chose to base its time-stamping implementation on standard OCSP. This enables the service provider to conveniently deliver certificate validity and time information in one convenient query response. The OCSP protocol query format contains a Nonce field, which protects against replay attacks. Instead of cryptographically random data, the Nonce field is set to contain the hash of the data to be signed, because it can also be interpreted as just a random number. According to the RFC, the OCSP responder signs its response which in SK's case,*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

contains the original Nonce (document hash), response providing/signing time and ID of the certificate used to give the signature, binding the three pieces of data together and providing the validity confirmation for the digital signature. SK stores the signed response in its log as evidence material.The main features of the above concept are:

- *it guarantees long-time digital signature and document validity*
- *it is based on standard protocols and standards*
- *verification process is lightweight and the document is self-contained – no additional verification services are needed*

SK has implemented all of the above, including both client and server parts, in its DigiDoc digital signature architecture."

This solution requires a different kind of archive strategy which allows comparison of the original document and the stored hash - this is not part of the description but would present a less onerous problem than conventional archive strategies.

Nevertheless, it illustrates how a tricky issue can be solved with some ingenuity - and without abandoning standards.

Conclusion

This paper has outlined a *few* of the issues that PKI implementation presents. In particular, we have looked at the issues with large scale, identity schemes using secure tokens – there are many other kinds of PKI that have their own issues.

In doing the research, the discovery of the sheer number and scale of PKI initiatives around the world came as something of a surprise. As well as real implementations, there is a lot of discussion going on in government bodies, standards committees, news groups and at seminars. And, last but not least, a lot of money is invested in PKI, especially by governments, as illustrated in some of the examples. (These observations influenced the title and direction of the paper from an initial "Common PKI implementation issues" to the reference to the climb up the "Slope of Enlightenment".)

The only way for PKI to claw its way up the "Slope" is to tackle the issues in real-life implementations as they arise. Hopefully, a more mature and common sense approach to the new terms and conditions for the transition from paper-based to digital transactions will ultimately prevail. Harmonization of various PKI and smart card standards are underway, although they appear to be taking somewhat different directions in North America and Europe. The challenge will be to re-invent while keeping the integrity of public key technology, and understand which parts can be changed and which ones must not be tampered with.

Finally, the IT security community as a whole is learning that there is no substitute for vigilance, and that principles of prevention and detection must be incorporated in every solution. And perhaps the biggest lesson of all in the evolution of PKI is that no matter how clever the original idea is - it will be of no use unless it is usable.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References

General

- Schneier, Bruce, "Applied Cryptography", Second Edition, John Wiley & Sons, 1997
- Schneier, Bruce, "Secrets & Lies", John Wiley & Sons, 2000

Department of Defense Common Access Card (DoD CAC)

- Activcard Case Study, "The Common Access Card Program at the U.S. Department of Defense", Activcard solutions for large-scale smart card and Digital Identity Provisioning, http://www.activcard.com/solutions/id_cards.html (Free Registration required)
- Smart Card Alliance's Digital Security Initiative Case Study: "Department of Defense to issue up to 13 million Common Access Cards for smart card-enabled PKI", http://www.smartcardalliance.org/alliance_activities/dsi_case_studies.cfm (Free Registration required)
- DoD Public Key Infrastructure Program Management office, "X.509 Certificate Policy for the United States Department of Defense". Version 7.0. 18 December 2002 http://iase.disa.mil/cpmwg/doclibrary/doclibrary.htm
- DefenseLINK news article: "Chief Information Officer Stenbit Demonstration Of PKI", 26 July 2002, http://www.defenselink.mil/news/Jul2002/t07262002_t0726pki.html
- Department of Defense, An Introductory Overview (DoD 101), 2002 data, http://www.dod.gov/pubs/dod101/dod101_for_2002.html
- Government Computer News, "BIG DEAL: DoD puts millions of smart cards in play", 9 June 2003, http://www.gcn.com/22_14/cover/22332-1.html
- Dan Caterinicchia, "DMS, despite failings, wins IG support", Federal Computer Week , 21 April 2003 http://www.fcw.com/fcw/articles/2003/0421/tec-dms-04-21-03.asp
- Picatinny Arsenal, N.J. website: Common Access Card (CAC) Experiences & Lessons Learned, http://w4.pica.army.mil/cac/, last modified 13 November 2002
- Defense Message System Public Website, http://www.disa.mil/apps/apm/
- Dan Caterinicchia, "Army taps DMS for wartime comm", Federal Computer Week, 27 March 2003, http://www.fcw.com/fcw/articles/2003/middle_east/web-dms-03-27-03.asp

Estonian ID Card

- AS Sertifitseerimiskeskus "The Estonian ID Card and Digital Signature Concept, Principles and solutions", Whitepaper, Version: 5 June, 2003, http://www.id.ee/pages.php/03031002
- AS Sertifitseerimiskeskus ID card information pages, card issuance statistics, http://www.id.ee/pages.php/03030102
- AS Sertifitseerimiskeskus Website: http://www.sk.ee
- Citizenship and Migration Board ID Card Website: http://www.pass.ee
- Finnish Population Register Centre Electronic ID card: http://www.fineid.fi/default.asp?todo=setlang&lang=uk

Notes

1. The links were all active as at the 16th July 2003.
2. The author is grateful for the spirit of disclosure on the internet, and apologises that if the DoD examples appear more fraught with issues, it is partly due to the fact that the project is open to public scrutiny and issues are being made public, whereas other projects are able to keep a lower profile. The author also acknowledges that media reports such as the ones referenced, may not be objective and may contain factual errors. They were included to illustrate examples and not to provide facts.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: September 3rd, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced