Global Information Assurance Certification Paper

A vulnerability assessment of roaming soft certificate PKI solutions

Stephen Wilson
March 2002

Abstract

In the past two or three years most major PKI technology vendors have released products which allow digital certificate holders with "soft certificates" to have their private keys stored at a central server and uploaded when needed to their local machine. This allows users to "roam" from one machine to another without having to manually manage the export and import of their keys onto temporary media like diskettes. Thus users gain much of the portability and usability advantages of hardware key media like smartcards and USB dongles but without the associated cost.

However, significant security compromises are entailed in any roaming soft certificate solution since fundamentally the key material is susceptible to sniffing or eavesdropping for at least some of the time. Careful security engineering and product deployment is needed to strike the right balance between cost/convenience and protection against identity theft. To date, little analysis of this balance appears in the public domain and the relative strengths and weaknesses of commercial solutions is difficult for users to determine. This paper highlights the security engineering and deployment considerations by presenting a systematic vulnerability assessment of the common roaming architecture.

Note: it is assumed that readers are familiar with the principles of digital certificates and public key infrastructure, as well as some common PKI applications.

Introduction: mobility and digital certificates

Conventional soft certificates

It is widely appreciated that the effectiveness and security of any PKI system depends critically on the care which users take in managing their private keys (see for example [1]). PKI systems with "soft certificates"[1] held in regular disk memory are vulnerable to various attacks where the private key may be stolen or substituted, usually without the user even being aware of it. The best known example of such PKI identity theft is the Caligula virus which is known to infect PGP users and surreptitiously upload copies of their private keys to the attacker's FTP site.

[1] Note that the term "soft certificate" can be technically misleading since the central issue is *not* the storage of the digital certificate but the users' *private key*. Nevertheless, "soft certificate" has wide currency and will be used as such in this paper, understanding that it is taken to mean that the private key is held in regular magnetic storage. Another effective synonym in the industry is "credential".

The vulnerability of many common PKI enabled applications to this type of attack rests on the fact that the users' private keys are often stored in a standard registry or file system location. The use of standard memory locations aids interoperability and software design, but it means that any programmer with a copy of the operating system manual can know the location of the private keys and thus write a Trojan Horse or virus that reads them off (for a good illustration of how easy this is, see the discussion of Caligula at [2]).

For mobile users, like tele-commuters and frequent travelers, conventional soft certificates represent a significant problem. While it is usually possible to export your credentials to a diskette and load them back into another machine, the process has many disadvantages:

- platform dependence – typically credentials exported from one browser can only be loaded back to a browser from the same manufacturer
- security – the exporting of a soft certificate usually leaves the original private key behind
- labor – it typically takes many steps, and additional user training is necessary.

Faced with these problems, mobile users are traditionally encouraged to move to portable hardware storage media for their private keys.

Portable hardware key storage

Smartcards are commonly understood to be the best medium in which to store, carry and utilise private keys, safeguarding the user against identity theft. The best smartcard technologies today allow for keys to be generated in the chip on the card, and subsequently invoked by digital signature firmware also in the chip, so that under no circumstances need the private key ever leave the secure environment of the smartcard. Access to the smartcard is controlled by a PIN (or possibly a biometric), providing optimum control for the user over their private key. It is unlikely for a diligent user to lose their smartcard without becoming aware of it, and even so, an attacker needs to overcome the PIN or biometrics in order to gain control over the private key.

The critical limitation of smartcards of course remains the low level of penetration of smartcard readers into the general PC environment. Thus smartcards are a major constraint on user mobility at this time and into the foreseeable future.

One response to the paucity of smartcard readers is the packaging of smartcard-equivalent cryptographic chips into "dongles" that interface to the user's machine via the USB (Universal Serial Bus) port. Several manufacturers have commercial products on the market; the cost is generally comparable to that of a smartcard plus low cost reader (i.e. less than US$50 each in small volumes). A USB dongle should have the same sort of cryptographic capability as a smartcard; in particular, it should provide on-chip key generation, a minimum 1024 bit RSA key length, and sufficient capacity for multiple keys and certificates.

The USB port is common to almost all recent laptops and desk-top PCs, making the USB dongle a widespread mobile solution. On the other hand, the USB port is not always conveniently accessible, so the dongles can be awkward to use.

A more serious disadvantage is longevity. The USB port was originally designed for connecting peripheral devices where cables are inserted and removed only occasionally. A crypto dongle on the other hand gets inserted daily or even more frequently. The author has anecdotal evidence that the mean time between connection failures for USB crypto dongles is less than one year, so they may need to be replaced at least annually.

An overview of roaming soft certificate solutions

Given the cost and availability problems of hardware storage devices today, more sophisticated approaches to soft certificate storage have arisen. Most important of these is to store users' private keys on a central server and to upload temporary copies to the client side system as and when they need to be used, for instance at home, in an Internet café or at a remote office location. In overview, the approach works as follows.

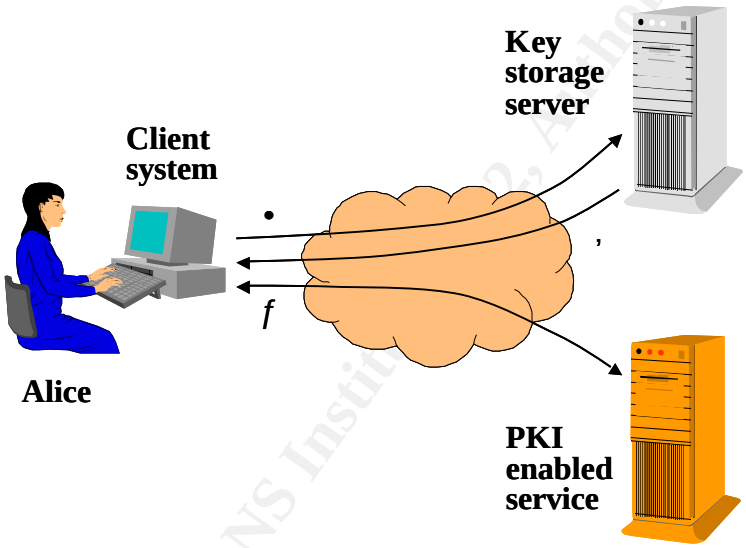

Figure 1: Basic roaming soft certificate architecture

When a user wishes to execute a PKI-enabled application on a local machine, they first authenticate themselves to the central key storage server (1). The server subsequently transmits an encrypted copy of the user's private key to the local machine, together with executable code of some sort to activate the key (2). The key is decrypted by the client and loaded into regular memory. For the remainder of the session, the local machine then behaves as it would had the private key been installed locally all along (3). Regular PKI enabled secure e-mail and web applications for example should work as normal. At the end of the session, the client system must destroy the local copy of the private key, so that when the user leaves the machine, their digital identity is not left behind.

Several vendors now offer roaming soft certificate solutions.[2] Most are superficially similar, with the main point of differentiation relating to the type of executable code used to decrypt and install the private key at the client side. Some products use a plug-in executable while others opt for a dynamic code fragment or applet uploaded from the key storage server at the beginning of each session. It is claimed (naturally enough) that the latter approach is more portable and faster in operation.

A more important area of difference is the means by which the user is authenticated to the key storage server. This issue is discussed in detail below.

All roaming soft certificate solutions have the following advantages:

- relatively low variable cost compared with hardware based key storage media
- good portability
- ease of use, and
- general interoperability across a range of PC/client platforms.

On the other hand, all share fundamental disadvantages deriving from the vulnerabilities of soft storage, and thus have to be carefully engineered to limit the possibility of interception of keys. Furthermore, a weak link in the system is the means for authenticating the user to the server. If this is done by conventional username and password, the risk of identity theft remains high.

These weaknesses and others are studied in more detail in the next section.

Security vulnerabilities of roaming soft certificates

Despite the obvious tradeoffs entailed in these solutions, there is actually little critical analysis in the public domain of the strengths and weaknesses of roaming soft certificates.

- A search of several leading security web sites, in addition to Alta Vista, uncovered only vendor information, plus one early article on the topic (see next bullet). Sites searched included the SANS Institute (www.sans.org), Counterpane (www.counterpane.com) and the Queensland University of Technology Information Security Research Centre (www.isrc.qut.edu.au).
- One independent article was discovered, by Gary Kessler in December 1999 [3]. It introduces the topic but has nothing to say about security.
- In what is otherwise one of the better recent PKI text books, Adams and Lloyd [4] devote a mere three paragraphs in total to the topic and only offer the simple recommendation that "it is important to ensure that the technology vendor can support [roaming] in a secure and robust manner". They do not canvass even the

[2] Roaming solutions were canvassed in detail for the major PKI vendors Baltimore Technologies (www.baltimore.com), Entrust (www.entrust.com) and RSA Security (www.rsasecurity.com). It is understood that solutions are also marketed by Arcot, Hush, SingleSignOn and Vasco, among others.

basics of server storage of private keys, let alone provide any framework for understanding what “robust” and “secure” could mean in this context.

To date the major vendors have released little or nothing in the way of technical white papers or competitive comparisons to help users understand the security tradeoffs involved in roaming soft certificates. Therefore it is time for an independent analysis and security assessment.

A new analysis

To systematically analyse the security vulnerabilities inherent in this class of solution, we need to first understand the steps that take place in establishing a session under the architecture common to all products considered. Subsequently, we can perform a Threat & Impact Assessment.

The following diagram illustrates the transactions that occur between the three main actors in a roaming soft certificate session (the alphabetic labels on the diagram are referenced later in the Threat & Impact Assessment).

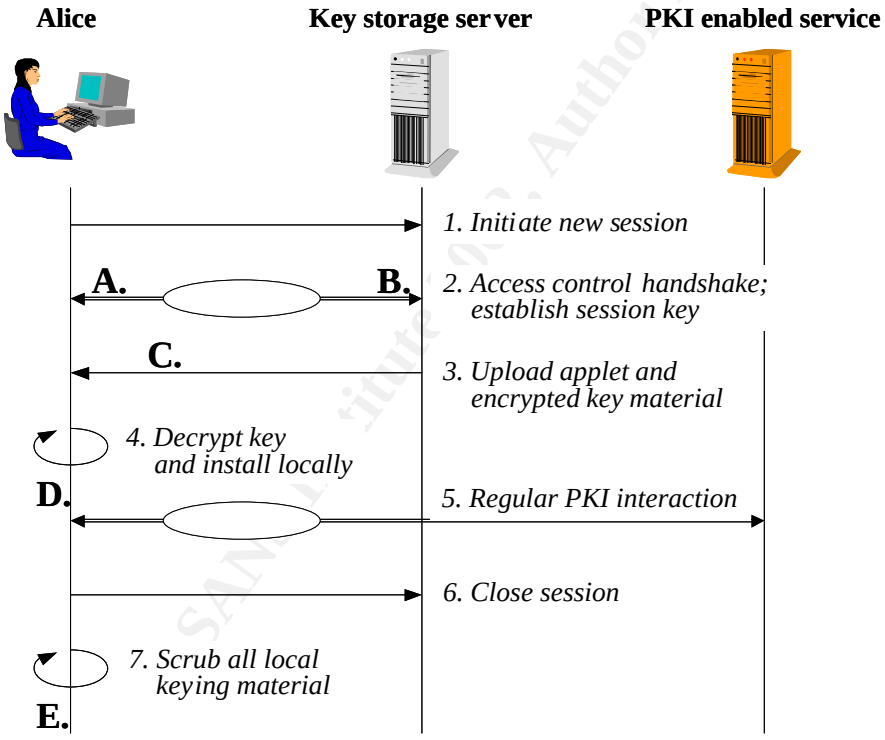

Figure 2: Actor diagram for roaming software certificate session

1. User Alice first initiates a new session for her client software to access the key storage server;

2. The server and Alice's client mutually authenticate each other, through some sort of access control handshake (see below), and thereby establish a session encryption key to be used to secure the private key transfer to follow;
3. Once server and client are authenticated, the server uploads the applet (if applicable) and Alice's private key, encrypted under the session key;
4. Alice's client software decrypts her private key and loads it into memory (the storage location depends on the design of the roaming solution but typically a standard operating system location is used so that the key is seamlessly available to a wide range of applications, as if it had been loaded in the machine all along);
5. Alice then goes on to conduct typical PKI-enabled transactions with another service (the details of which are immaterial to this analysis);
6. At the end of the PKI transactions, the client software closes the session with the key storage server;
7. The client side software scrubs (that is, securely deletes) the local copy of the private key, together with any other cryptographic data possibly related to it.

Threat & Impact Analysis

In accordance with conventional risk management standards [5] we can now tabulate the potential threats to any such roaming solution, the impact on users of each threat, the vulnerability of the system to each threat, and the available means for mitigating those vulnerabilities.[3] The analysis will only focus on the security of the roaming private key management and will ignore the possibility of attacking the PKI transactions once underway, since the latter is generic.

In the following table, certain vulnerabilities are cross-referenced alphabetically against the preceding actor diagram; e.g. (D). Explanatory notes appear at the end of the table and are cross-referenced numerically; e.g. (4).

A note about "risk": a full blown analysis of this type usually delivers a rating of the risk associated with each vulnerability – reflecting both the severity of impact and the likelihood of occurrence – to help prioritise corrective actions. Threat & Risk Assessment methodologies (as per [5] for example) allow security engineers to measure and weight both severity and likelihood according to the context of their business. However, in a general analysis such as the present paper, it is not possible to determine these factors, and therefore we stop short of assigning risk ratings here.

[3] In line with AS/NZS 17799, we adopt the following terminology:
- a security *incident* is any event which adversely affects the confidentiality, availability and/or integrity of an information system;
- a *threat* is an unrealised security incident;
- a *vulnerability* is the means by which a threat can become an incident;
- *risk* is a measurement, specific to the business in question, of the seriousness of an incident, taking into account its impact on the business and its likelihood of occurrence.

In this context, risks can generally be mitigated or they can be accepted. Only in rare circumstances can a given information security risk be eliminated altogether.

Threat	Business impact	Possible vulnerabilities	Available mitigations
Identity theft	Impersonation and fraud.	Spoofing Alice during session establishment (A) • by brute force attack • by password guessing • by password sniffing.	• Two factor authentication (1); • Enhanced password protection protocols; e.g. SPEKE (2); • Sound user password management (3).
		Interception of private key material during upload (C).	Strong encryption of private key with e.g. 128 bit session key (4).
		Retrieving private key from disk during session (D).	• Personal firewall to safeguard against Caligula-like viruses (5); • Special key storage location in fat client (6); • Keep sessions as short as practicable to minimise exposure; • Separate one's roaming certificate from other fixed credentials (7).
		Retrieving private key (or remnants of it) from disk after session closes (E).	Scrub (securely delete) all key material from disk when session closes (8).
Spoof key storage server	• Denial of service, with loss of or delays to business; • Identity theft (and subsequent impersonation) via password sniffing.	IP or web address spoofing (B).	SSL server certificate for key storage server.
Fail to upload private key	Denial of service, with loss of or delays to business.	Transmission failure (C).	• Sound communications protocol design; • High performance server for key storage.
		Encrypted data payload blocked at user firewall (C).	Use "firewall friendly" standard protocol (SSL).
Cannot reach key storage server	Denial of service, with loss of or delays to business.	• Conventional denial of service attack on server; • Conventional outage.	• High availability server design; • High quality ISP; • Quality perimeter defence in depth.

Notes to the table

(1) Initial access to the key storage server is ideally controlled using a challenge-response device or one-time password generator, in addition to username and password. These measures mitigate against password guessing, and against replay attack following password sniffing.

All vendors researched for this paper quite rightly indicate in their technical material the inherent weakness of single factor authentication of the user to the key storage server. Most go on to advocate two factor authentication. Some offer a bundled two factor solution (such as RSA Keon with RSA SecureID) while others claim to support arbitrary external solutions via an API.

(2) There exist a number of so-called Zero Knowledge Proof (ZPF) algorithms for mutual authentication whereby two parties can prove to one another that they know a shared secret, without having to reveal what that secret is. This avoids the vulnerability of transmitting clear text passwords over the network. A family of techniques has evolved from Exponential Key Exchange [6], of which the most relevant here is Small Password-authenticated Exponential Key Exchange (SPEKE) [7]. While most EKE variants use a long random value to seed the key exchange, SPEKE allows short passwords to be used, making it significantly more user friendly. Nevertheless, SPEKE cannot protect against password guessing and so does not obviate the need for sound password management.

Of the major vendors, only Entrust at this time promotes its use of a sophisticated shared secret protocol like SPEKE [8].

(3) In all cases, sound password management by users is essential, including the use of non-obvious phrases and regular rolling of old passwords.

(4) Some vendors go into more detail than others regarding how they encrypt the uploaded key material and applet, including the specification of multiple encryption of the session key [9,10]. Such crypto-theoretic details are beyond the scope of this paper. Moreover, they are beyond the understanding of the majority of users, so the point must be made that such documentation does not really help customers make decisions about the candidate solutions.

(5) The possibility of having one's private key stolen from the local machine during the course of a roaming session – via any of the mechanisms where a permanent soft certificate can be stolen – is probably the most serious threat of all. The best defence might be a personal firewall (the Caligula virus has a distinct signature [2] and variations on Caligula are likely to be similarly readily detectable). However, it is in the nature of roaming that the user is unlikely to be sure to have a firewall wherever they happen to be working.

(6) As discussed, the great vulnerability of most fixed soft certificates derives from the fact that they employ standard registry or file system locations to hold the private key. If on the other hand, a fat client roaming solution was to use proprietary storage locations, known only to the software developers, it would become far harder to write a Caligula-like virus. The proprietary approach is hardly perfect since it is vulnerable to a social engineering attack on the developer, but it would help to keep the vendor ahead in the "arms race". There is also an obvious interoperability tradeoff.

(7) Given that roaming soft certificates are fundamentally more vulnerable to attack than fixed soft certificates (which might at least be afforded a robust firewall) it would be prudent to obtain one or more separate certificates for roaming use. Careful partitioning of certificate use makes forensic investigation simpler too in the event that an identity is stolen.

(8) Careful attention must be paid to the manner in which the private key is deleted. It is widely appreciated now that to properly purge magnetic media of unwanted data, it is necessary to overwrite the physical data concerned at least three times with different bit patterns (the National Computer Security Centre recommends three overwrites[11]; Schneier recommends seven [6]).

Interestingly, none of the vendors touched on this topic. Candidate vendors should be questioned about the security of their scrubbing before a real life product selection is made. Ideally, independent security evaluation and/or certification of roaming soft certificate systems would cover the quality of the key scrubbing.

An academic legal risk?

Finally, let us address the possibility of additional legal risk in the use of roaming soft certificates. Orthodox PKI analysts have long advocated that private signing keys should remain under the sole control of the user. Otherwise, they argue, it may be possible for a user to mount a spurious case that a given transaction may have been signed by someone else, merely because *in principle* their private key might have been copied. This possibility is claimed to weaken the desirable property of non-repudiation in the PKI.

In this context, it must be considered that in any roaming soft certificate solution, the user is not in sole possession of their private key. Perhaps careful cryptographic design can preclude anyone but the legitimate user ever accessing the private key in the clear. Even so, the question of sole custody of the private key is not as clear-cut as it is in the case of hardware storage tokens, and on that basis, there may be new legal uncertainties and risks associated with digital transactions applied by roaming users.

In the view of the author at least however, rather too much emphasis has been given historically to sole custody of the private key (see also [12]). "Non-repudiation" has more to do with legal principles and arguments than it has with technology per se. In

deciding whether a claim to repudiate a given digital signature is legitimate or not, we must weigh the sum total of evidence indicating where and how the transaction originated. The probability that a signature originating from the claimant's private key was actually applied by that person or by someone else is a function of how well the key is protected and controlled.

A well engineered key storage server, under careful management by a reputable organisation, should not introduce any serious doubts as to unauthorised access to the keys held on that server. Thus, it is not necessarily the case that just because a user is not in absolute sole possession of their keys that they are able to repudiate their signed transactions as they please.

Conclusion

The high level threat and vulnerability analysis presented here indicates that if properly engineered and implemented, a roaming soft certificate solution can bring the usability and portability advantages of smartcards or USB dongles, at a reduced cost.

The analysis indicates three major risk management tactics, at least two of which would ordinarily be under the control of the user:

1. the option of two factor authentication for accessing the key storage server should be taken up if available from the vendor,
2. roaming soft certificate users should separate the credentials they use when roaming from those used at their fixed locations, and
3. ideally a personal firewall should be in place wherever practicable at the client's remote site and configured to block Caligula-like viruses (or more crudely, to block all FTP services).

The analysis also indicates two major design issues for vendors:

1. the applet or client side plug-in (as applicable) must scrub (securely delete) all private key material and related data at the close of a roaming session, in compliance with accepted secure data deletion standards, and
2. care must be taken designing the cryptographic security for the private key upload.

Despite the rapid maturation of this sector of the PKI market and its strong appeal, very little information is yet in the public domain to help prospective buyers select a roaming soft certificate solution. As with all security product selection, independent evaluation or certification should be sought wherever possible. Customers should seek specific information from vendors as well as external specialists about each candidate product's data scrubbing and cryptographic design.

References

[1] Ellison, Carl and Schneier, Bruce. *Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure* in *Computer Security Journal* Vol 16, no. 1. 2000. Also available at www.counterpane.com/pki-risks.html.

[2] Internet Security Systems (ISS) Vulnerability Alert *Windows Backdoors Update II: NetBus 2.0 Pro, Caligula, and Picture.exe*. February 19, 1999. See www.iss.net/security_center/alerts/advise20.php.

[3] Kessler, Gary C. *Roaming PKIs: Harbinger of Virtual VPNs?* in *Information Security Magazine*. February 2000. Also available at www.garykessler.net/library/roaming_pki.html.

[4] Adams, Carlisle and Lloyd, Steve. *Understanding Public Key Infrastructure*. New Riders, 1999.

[5] *Code of Practice for Information Security Management*. AS/NZS 17799:2001. Standards Australia, 2001.

[6] Schneier, Bruce. *Applied Cryptography Second Edition*. John Wiley & Sons, 1997.

[7] *A peek at SPEKE*. Integrity Sciences Inc. www.integritysciences.com/peek.html.

[8] *Integrity Sciences Licenses SPEKE Cryptography to Entrust® Technologies.* Joint press release of Entrust Inc. and Integrity Sciences Inc. June 15 1999. www.integritysciences.com/PKI50.html.

[9] Entrust Authority Roaming Server Data Sheet www.entrust.com/authority/roaming/datasheet.htm.

[10] Baltimore UniCERT Roaming Data Sheet www.baltimore.com/unicert/technology/roaming.asp.

[11] *A Guide to Understanding Data Rememberance in Automated Information Systems*. NCSC-TG-025 Version 2. National Computer Security Centre. 1991.

[12] Wilson, Stephen. *Comparison of Authentication Technologies in E-business* in *Asia Business Law Review* No. 33. July 2001.