Storing and Protecting Certificates
A Sevan White Paper

The Sevan WSA authenticates users on the basis of their SSL certificates. This Sevan White Paper discusses options for storing, using, and protecting these certificates.

Although we talk about "certificates", public key authentication actually involves two components:

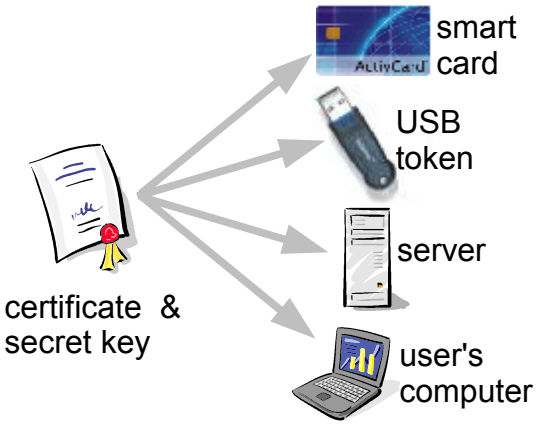
- the user's *SSL certificate* is typically a few thousand bytes of data
- the associated *private key*, which is a few hundred bytes

Unlike passwords, users cannot memorize their certificate nor private key, so both must be stored in a device. Furthermore, the private key must be stored in such a manner that others cannot discover its value. If another user were to obtain the value of your private key, he could impersonate you: your private key is your identity. Conversely, the certificate has no secret information, need not be protected, and must be available to anyone doing business with you.

There are a number of options for storing a SSL certificate and its associated private key. The most commonly mentioned are:

- a smart card
- a USB smart token
- a server
- the user's computer

The remainder of this paper discusses the pros and cons of the various storage options.



Smart Cards

The smart card has two major advantages:

- it is portable, so a user can carry his certificate with him
- it protects the private key from others

The smart card is the size of a credit card, so it can be carried in the user's purse or wallet. The user plugs the smart card into a smart card reader attached to his computer. When the WSA attempts to authenticate the user, the computer asks the smart card to perform the necessary authentication operations.

Security specialist find the smart card attractive because the private key is protected on the card. Cryptographic smart cards perform the public key operations right on the card, so the private key never leaves the confines of the card. Furthermore, a secure smart card is designed so that it is very difficult for a hacker to extract the private key from the card.

A smart card is protected by a password or PIN. The smart card will not operate unless the user provides the PIN. This protects the certificate and private key in the event that the card is lost or stolen; similar to the PIN of an ATM card.

The adoption of smart cards has been slow, especially in North America. The most commonly cited reasons are:

- cost, $20 to $50 per card,
- scarcity of computers equipped with smart card readers, and
- the propensity of users to misplace cards.

Since the PIN protects the smart card when it is stolen, the loss issue is not a security challenge. Rather it is an operations problem: the user cannot be authenticated while his card is lost.

USB Tokens

USB tokens are similar to smart cards, but they do not require a specialized reader. Rather the USB tokens plug into a standard USB port. Like smart cards, the USB token can be carried by the user and inserted into a computer when needed.

Although the USB tokens do not require special hardware, they do require special software. This means that you cannot plug your USB token into any computer and expect it to work.

The USB tokens has similar disadvantages as the smart card: cost, special software on the computers, and they are easy to misplace.

Certificate Servers

This option is seldom used, but it is often mentioned. The idea is to store certificates for all users on a common server. When a user needs his certificate, he “borrows” it from the server. This might involve the certificate being temporarily copied to the user's computer or the user's computer remotely accessing the certificate (and private key) on the server – the server acts like a giant, shared smart card.

There are few certificate servers deployed. The reasons are logistic and security. The logistic issues are easy: the user is deprived of his certificate when he cannot access the server. The security issues are more involved. These systems require that a user authenticate himself to the server, typically through passwords. If an imposture knows your password, he can use your certificate at will, so this approach is no stronger than password authentication. Secondly, the certificate server is a single-point of attack. If a thief gains access to the certificate store, he can impersonate all of the users.

User's Computer

Storing the certificate on the user's computer is the most commonly used option when the user has his own computer (more on this later). The benefits of this approach are:

- there is no incremental cost (no cards, tokens, or servers)
- users are less likely to lose a computer than a smart card or token
- the certificate is always available when user is at his computer.

The SSL certificate and private key are stored on the computer's disk drive. Depending on the operating system and the browser, the key and certificate is stored within a file or within the operating system. In either case, the value of the private key is generally hidden, so as to be difficult to find.

The utility of storing the certificate on the user's computer depends on the user's situation. Consider the following examples:

- When a user exclusively uses his own dedicated computer, it is natural to put the certificate on his computer. The certificate will always be there when the user needs it, and no other users are expected to use the computer. An example of this scenario is a consultant who lives and dies by his computer.

- When a user always uses his account on a shared computer, it is convenient to put the certificate on the computer. The certificate is accessible only through the user's account, and it is always available when the user is logged on. Note that the user relies on the security of the account login to protect his certificate from other users. An example of this scenario is a family member who shares a home computer.

  Note that similar to the certificate server, the certificate is protected by a password (in this case the account login password). However, the population having access to the shared computer (in this example, the family members) is usually much less than those who can attack a certificate server (possibly anyone with internet access). This reduces the likelihood and consequences of a successful attack.

- When the user exclusively uses a few computers, it makes sense to put a certificate on each computer. The user has an option of putting a copy of a single certificate on the computers or giving each computer its own certificate. In either case, the certificates will be there when the user needs them. An example of this scenario is an employee with an office and home computer.

- When the user access the web through shared or public computers, it makes no sense to put a user's certificate on the computers. No one wants his certificate (his authentication credentials) on a publicly available computer. In such cases, it makes more sense to store the certificate on a smart card or USB token, which the user can carry from computer to computer. An example of this scenario is a roaming nurse who uses the computer at the closest nurse's station.

  There has been a great deal of discussion concerning the advisability of using a smart card or USB token on a shared computer. There are practical issues that the card or token won't operate without the proper software and most shared computers prevent software downloads. There are also security issues raised by an untrusted shared computer. Even though the card or token protects the private key, the malicious code in the computer can intercept the user's PIN, copy information, or fool the card into performing a bogus authentication. As attractive as portable certificates sounds, most experts feel that reasonably secure portability is far off.

Since the most likely scenario is to store a certificate on the user's computer, the remainder of this paper considers issues raised when a certificate is stored on a user's computer.

Losing the Certificate

Previously we stated that one of the benefits of keeping the certificate on the user's computer is the certificate is there when the user needs it. In general this is true, but it is possible to lose the certificate through one of the following events:

- the user deletes the certificate (generally an browser configuration window),
- the computer is lost or stolen, or
- the certificate is destroyed by a software or hardware failure, such as a disk crash.

In the event the certificate is erased, lost, or destroyed, the user can no longer be authenticated. This is equivalent to a user losing his smart card or password.

When the user loses his computer, the user or the administrator must make sure that the certificate on the computer will no longer authenticate the user. This prevents the thief from authenticating himself as the user. The WSA has facilities for the user or administrator to “unenroll” a lost certificate, which prevents the thief from accessing the user's accounts.

When the computer is lost or fails, the user generally loses much more than his certificate. If the user has

the wherewithal to backup his disk drive, he will have a backup of his certificate. The certificate will be restored as part of disaster recovery.

Obtaining a SSL Certificate

Most computers do not have an SSL Certificate. There are three ways of obtaining a certificate:

1. Generate a *certificate* request, send the request to a certificate authority, receive the resulting certificate, and install the certificate into your computer.
2. Receive a private key and certificate from the certificate authority and install the certificate into your computer.
3. Request an on-line certificate from the WSA. You select a certificate name, and 3-clicks later the certificate is ready to use.

Generating a certificate request and installing the resulting certificate is too complicated for most users. It generally requires a number of steps and even some cutting and pasting of messages.

Installing a private key and certificate is far easier, but still requires above-average computer skills. Another issue is the that the certificate can be installed in multiple computers, which is generally not desirable (more on this later). Another draw-back is that that certificate authority has a copy of your private key. This means that a security breech of the certificate authority will expose your identity.

Generating a certificate on-line is a good balance of simplicity and security. It is simple enough for all users. Since the WSA loads a “non-exportable” certificate into the computer, we avoid the problem with certificate sharing. Finally, the private key never leaves the user's computer, so it is safe.

Copying the SSL Certificate

Most browsers or operating systems selectively allow certificates (and the associated private key) to be copied from one computer to another. This usually takes place in two steps: the certificate is *exported* to a file and the file is *imported* to the other computer.

Most browsers allow the user to specify whether or not a certificate is exportable. If the certificate is not exportable, the browser or operating system prevents the copying of the certificate and private key. In this manner the certificate is locked into the computer and cannot be readily copied. We use the modifier “readily” because it is conceivable that the certificate and private key could be copied using a hacker tool. As of yet, we have not found such a tool.

It is convenient to allow certificate copying. For example, you can copy your certificate from an old computer to a new one. But copying is also dangerous. If an attacker is able to copy a user's certificate, he can imitate the user without the user knowing. Another problem is that copying prevents the WSA from authenticating the user's access from a particular computer. This is a problem when the security policy restricts access to only certain computers or audit requirements require computer identity.

A SSL certificate generated by the WSA, is stored in the browser in a non-exportable format. This means that the user or hacker cannot directly copy the certificate to another machine. A WSA-generated SSL certificate is essentially bound to the computer.

Protecting the Certificate

Each user must make sure that others are not able to use his certificate – users must protect their private keys. If a thief gains access to your private key, he can impersonate you: access your bank account, e-mail, and medical records.

When the user stores his certificate on his computer, he must use the security features of the browser to prevent others from using his certificate. This generally involves a password or PIN. Most browsers (including Internet Explorer, Netscape, and Opera) allow a user to create a password to protect the certificate. When password protection is enabled, the user must provide the password before the browser will allow access to the certificate.

Most newer operating systems (including Windows XP, Windows NT, Windows 2000, Unix, Mac X, and Linux) can require users to login before being able to access their certificates: the certificate is protected by the user's login password. Furthermore, in situations where people share a computer, each user can have an account and a certificate. A user of a shared computer can rely on the fact that other users must have his account password in order to access his certificate. The account password protects the certificate as it does the user's files and applications.

In summary, the certificates and private keys are protected through a combination of restricting physical access to the computer, browser PIN, and operating system passwords.

There is a great deal of discussion concerning the security of commercial operating systems. The concern is that the security mechanisms can be circumvented through vulnerabilities or defeated through trojan horses. It is important to put the certificate's vulnerabilities into perspective. The computer's security mechanisms protect all of the information and applications on the computer, so a user's certificate is no more vulnerable than his other sensitive data. A breech of the computer's security mechanisms will more than likely expose all of the the computer's content. Therefore, users must maintain the security mechanisms, which in turn, will protect the users' certificate.

Issues with Certificate Passwords

Most likely a user will rely on a password or PIN to protect his certificate. We might conclude that since certificates involve passwords, we have all of the problems found in password-based user authentication systems. Specifically, passwords must be manually entered, and they can be forgotten or stolen. Fortunately, passwords that protect certificates are used in a very specific way, which makes them less troublesome and vulnerable than passwords used for user authentication. Consider the following differences:

- In most instances the user enters the password once to unlock the certificate. From then on, the certificate provides authentication. Passwords protecting certificates are not entered as often as passwords for user authentication, and are therefore less cumbersome for the user.

- A password that protects the certificate never leaves the user's computer. An attacker must gain access to the computer in order to steal the password. On the other hand, passwords that authenticate users are usually sent over networks and exist in one form or another in applications, directories, or servers. A certificate password provides far few opportunities for attack.

- A user typically has a single password to unlock the certificate. This certificate, in turn, can provide the user with access to many applications. Since there is only one password, it is much easier for the user to remember. A single password makes it is much less likely that the user will forget the password or be forced to write it down.

- Since the password does not leave the computer, there is little reason to periodically change it. This makes it far more likely that the user will remember his long-term password.

You can think of a certificate password as a personal password: it is only between the user and the device holding the certificate. This makes the certificate password more secure and far easier to use.

Losing a Certificate Password

In most browsers or operating systems, if you lose the password protecting the certificate, you lose the ability to use the certificate. You essentially lose the certificate. You business continuity plan must provide for these contingencies. One option is to backup the password. If you do so, you must also backup the certificate and the private key. The other option is to provide a mechanism for "resetting" the certificate: essentially abandoning the old certificate and begin with a new certificate. A certificate reset, like a password reset, is a chink in your security, and as such it must be carefully crafted to avoid exploits.

Sevan

Networks, Inc
1310 Hollenbeck Ave, Ste F
Sunnyvale, CA 94087
Tel: 408.830.1000
Fax: 408.830.1001
www.sevannetworks.com