

*White Paper*



***The Security Advantages of  
Hardware Tokens over Software  
Tokens for PKI Applications***



# Table of Contents

Abstract .....	3
Background .....	4
Soft Tokens and Their Vulnerabilities .....	5
Hardware Tokens and Smart Cards .....	8
References .....	11
About Aladdin .....	12
About eToken .....	12



# The Security Advantages of Hardware Tokens over Software Tokens for PKI Applications

Yehuda Lindell

## Abstract

In this document, we compare the security of software-based PKI solutions to the security of hardware-based PKI solutions. The essence of our argument is that computers today are *inherently insecure* mediums, and thus software-based PKI solutions are vulnerable to attacks. In contrast, hardware-based solutions – and in particular smart card tokens (in USB or card form factor) – are able to protect the secrecy of keys even in insecure environments. This is crucial for the success of eCommerce.

## About the Author

Yehuda Lindell is an Assistant Professor in cryptography at Bar-Ilan University, Israel. He received his B.Sc. and M.Sc. (both in Computer Science) from Bar-Ilan University, and his Ph.D. in Computer Science and Applied Mathematics from the Weizmann Institute of Science, Israel. Upon completing his Ph.D., Yehuda spent two years as a Raviv postdoctoral fellow in the Cryptographic Research Group at the IBM T.J.Watson Research Center. Yehuda has performed extensive research and written numerous publications on the topic of cryptography.

More information about Yehuda Lindell can be found on his web site: <http://www.cs.biu.ac.il/~lindell/>

## Background

In asymmetric cryptography, each user has two distinct keys: a *public key* that is used for carrying out “public operations” (such as encrypting or verifying the validity of a digital signature), and a *private key* that is used for “private operations” (such as decrypting or generating a digital signature). PKI has become synonymous with the notion of asymmetric cryptography, although strictly speaking, they are very different. Specifically, a *public-key infrastructure* (PKI) is the way that we bind public keys to entities. To see why this is important, note that when we encrypt a message, it is crucial that we use the public encryption key that belongs to the intended recipient and not to some attacker. A PKI, typically implemented via certificate authorities, is essentially a way of distributing public keys in a secure way. In short, a PKI enables us to ensure that when we encrypt a message, it is indeed encrypted under the public key belonging to the intended recipient.

Likewise, upon receiving a signed document, we are able to verify the identity of the signer.

In this document, we will consider the *key management* side of asymmetric cryptography. That is, we will discuss where keys should be stored and why. We will ignore the issue of how public keys are distributed, and how we know what key belongs to what entity. As mentioned above, this is the role of a public-key infrastructure, and we will just assume that such a mechanism is in place. We begin by presenting some very brief background. However, we assume that the reader is familiar with the basics of asymmetric cryptography (encryption and digital signatures), and with PKI.

## Applications of Asymmetric Cryptography

Prior to the invention of the notion of asymmetric cryptography by Whitfield Diffie and Martin Hellman [2], and prior to its implementation in RSA [5], cryptography belonged almost solely to the military and to intelligence organizations.

Given only symmetric keys (where the same key is used for encryption and decryption), the problem of key distribution was unsolvable. Parties that wished to communicate with each other would have to physically meet in order to exchange keys. This is fine for “closed environments”, like the military. However, it is of no help whatsoever when it comes to applications like eCommerce.

In an eCommerce setting, parties need to communicate securely even though they have never physically met. Customers purchase items over the Internet, and need to securely send their credit card information to the vendor. The whole point of Internet shopping is that it is not necessary to physically go to a store. Indeed, many shopping sites have no store, and no physical meeting ever takes place between the store owners and their customers. In order to facilitate secure communication in such a setting, asymmetric encryption is essential, as is a working PKI. Thus, asymmetric encryption provides a solution to the problem of online shopping. However, eCommerce encompasses much more than just online shopping, and secure digital signatures are a necessary component of these solutions. Take the simple issue of email, for example.

**Asymmetric cryptography enables secure eCommerce by facilitating secure communication between different parties without ever requiring them to physically meet.**

Much business today takes place over email and users naively believe that the sender's name that appears in the received mail is reliable. However, in reality, it is trivial to send an email in someone else's name. This could have devastating effects. Essentially the only way to ensure the integrity of email is to sign each message with a secure digital signature scheme. Another crucial property of digital signatures is that of *non-repudiation*. This means that if a signed document is received, then the signer cannot later deny that he or she indeed signed the document. This property enables digital signatures to be presented in court, and provides a way of sending "online orders", for example, that cannot be rescinded by mere denial.

There are other very important and widespread uses of asymmetric cryptography, beyond its direct application to eCommerce. One prime use is that of *authentication*. In this setting, a user's ability to sign or decrypt a message serves as a proof of his or her identity. For example, in the case of Windows smart card logon, the user receives a credential (or "ticket granting ticket" in the Kerberos terminology) that is encrypted under his or her public key. This credential is only useful once decrypted. Therefore, only the private key owner is able to decrypt and possession of this credential is considered a proof of identity. Asymmetric authentication is also used for web authentication and is an integral part of the SSL protocol. Of course, asymmetric cryptography has many other applications as well, like securing email via encryption and protecting laptops via file encryption and boot protection.

## The Security of Private Keys

A fundamental fact of cryptography is that private keys must be kept private. If an attacker can obtain a party's private key, then it can easily decrypt all messages sent to that party. Furthermore, the attacker can sign any message it wishes in the name of the party, and can successfully impersonate them. Thus, all security is lost. A more subtle issue that arises is that non-repudiation for digital signatures is lost as soon as it is possible for a user to *reasonably claim* that his or her private key was stolen (it is irrelevant whether or not an actual theft took place). This is due to the fact that if a private key could have been stolen, then a claim that the signed order is a forgery will hold up in court. Of course, the existence of such claims would be devastating for the use of digital signatures in legally binding transactions.

**A fundamental fact of cryptography is that private keys must be kept private.**

## Key Management and This Document

In this document we will compare two alternatives for storing private keys. The first option is to store the private key in software, and the second is in an external, special-purpose hardware device. As we will see, this implementation choice has a huge effect on the security of the system.

## Soft Tokens and Their Vulnerabilities

In this section, we will discuss the security ramifications of storing private keys (for signing or decrypting) in software. The software encasing for a user's private keys is often called a *soft token*. In the context of PKI, it is also sometimes called a *software certificate*. In this case, in order to sign or decrypt, the user's

**Soft tokens – which must be duplicated onto all computers that the user wishes to work on – are exposed to security threats on those machines.**

private key is retrieved from the soft token and the desired operation is carried out. Note that from a usability perspective, this means that the soft token must be duplicated onto all machines that the user wishes to work on. As we will see below, this significantly increases the vulnerability of the keys.

## The Basic Threats

In order to understand the security of software-based solutions, it is important to note that *personal computers are an inherently insecure medium*. This is due to the following two basic threats:

1. *Physical access*: The first threat is based on the fact that in many cases it is possible to physically access a user's computer. Given such physical access it is possible to obtain the desired soft token. We remark that physical access can be obtained if a user's laptop is lost, through criminal means (e.g., by stealing the user's laptop, or breaking into their home), or just due to the fact that multiple users work on the same machine. Physical access is also available in almost any office environment. Thus, a user's soft token may be stolen (or, more accurately, copied) by fellow employees or even members of the cleaning staff.<sup>1</sup> It is important to note that locking the user's machine does not suffice because it is easy to manually extract the hard disk and read the files that are written there. (Of course, we can make it hard for the attacker to extract the soft token, as we describe in the countermeasures below. Nevertheless, it will still always be possible.)
2. *Malicious software*: A second major threat on soft tokens is due to the proliferation of malicious software. Viruses, worms, trojan horses and more are common in today's computing environment. Infection by any type of malicious software can be devastating, as the software can simply read the stored soft token and send it over the Internet to the attacker. This attack can also be made widespread and thus many keys from many computers can be captured by a single worm. We note that in the late 90s a virus named *Caligula* was released [7]; this virus targeted the PGP keyring, which is essentially a soft token used for storing private keys in the PGP system [10].

## The Basic Countermeasures and Their Security

Due to the above threats, soft tokens are typically not plain files that are stored in a user's personal directory – this would make stealing a soft token an overly easy task. Rather, they are protected using one or both of the following techniques:

1. *Password encryption*: The soft token is not stored in plain text on the user's machine. Rather, it is encrypted under a password that is chosen by the user. More specifically, a key is derived from the user's password

---

1. If you are wondering why cleaning staff would be interested in a soft token, then you should think of a targeted attack on a given company. In such a targeted attack, a natural strategy is to have a good hacker impose as a cleaner. With a small fee, the cleaning company will assign them to the desired location and they will then have physical access. We note that targeted attacks have been carried out in the past, and similar tactics have been employed.

(using a hash function), and the resulting key is used to encrypt and decrypt the soft token.

2. *Obfuscation*: Essentially, the soft token is hidden in a scrambled way on the user's hard disk. The aim of this method is to make it hard for hackers to find the soft token. (There are actually many ways of doing this, but the effect is the same in all cases. For example, one strategy is to encrypt the soft token with a secret encryption key, and then to hide the key in different places on the user's disk.) See [9] for more information on how obfuscation works in general.

Let us consider the security of each of these countermeasures. First, we remark that obfuscation can be of help. However, with enough effort, it is always possible to break it. Thus, relying on obfuscation is dangerous and it should be viewed only as an additional measure that may be employed in order to slow attackers down. This is suitable for applications connected to digital rights management (DRM), but not for eCommerce and PKI. Regarding password encryption, the security of this methodology depends on the quality of the password used. In order to understand this well, note that once an attacker has a copy of the user's encrypted soft token, it can guess a password and attempt to decrypt according to this password. It can then check if the result is a valid soft token. If yes, then it has succeeded; if not, it can try a new password. Such an attack is called an *offline dictionary attack* and is highly effective. The only way to prevent such an attack is to use a long random password. Unfortunately, most users are incapable of remembering such long passwords (especially since they usually have many passwords to remember). Thus, in most cases, offline dictionary attacks are very successful. For a recent example of how password-encrypted files (including soft tokens) can be broken, see [8].

## Conclusions on the Security of Soft Tokens

The main problem with soft tokens is that they rely on the integrity of the computer that they reside on. However, personal computers are inherently insecure: once physical access is obtained or a virus infects the machine, the user's private key may be completely compromised. To make things worse, this compromise may take place without the user even detecting the attack. Thus, the user may continue using his or her private key, even though it is known to an attacker. This solution therefore provides a relatively low level of security, and its use should be minimized. In particular, non-repudiation is not achieved with the use of soft tokens. Thus, eCommerce solutions that rely on asymmetric cryptography and PKI are far weaker when software tokens are relied upon.<sup>2</sup>

We conclude with a remark related to the threat of malicious software. It is important to note that it is possible to significantly reduce the risks due to malicious software by using appropriate anti-virus software and by behaving responsibly on the Internet (e.g., by not downloading executable files from

---

2. We stress that as with all security solutions, a proper risk analysis must be carried out in order to choose the appropriate solution. There are some cases where soft tokens may suffice. For example, consider a setting where users' machines are well protected from viruses and where users are well practiced in using long passwords. In such a setting, it may be reasonable to use a soft token, as long as the security threat is not too great.

unreliable sources). Within an organization, it is also possible to enforce such behavior, thereby further reducing the risk. Despite the above, virus infection is bound to occur. Furthermore, since the potential damage of such infection is so great when a software token is used, we believe that the risk is too great when a high level of security is needed.

## Hardware Tokens and Smart Cards

A hardware token is an auxiliary device that is used for storing a user's private key. There are a number of different ways that such an auxiliary device can be used, and we will consider the three main alternatives. In short, the alternatives are:

1. *External storage*: The external device is used for storing a user's private keys. These keys are imported to the local machine for signing or decryption.
2. *Hardware tokens with cryptographic capabilities*: The external device is used for the actual cryptographic operations. Thus, signing and decryption take place on the external device and not on the local machine.
3. *Smart card tokens*: These are hardware tokens with cryptographic capabilities, as above. However, smart card tokens also come with significant mechanisms for preventing physical attacks. We remark that smart card tokens come in a number of different forms (e.g., in USB or card form factor). Our discussion here holds irrespective of the format.

We proceed to describe each solution and its security.

### External Storage

Essentially, in this solution, a soft token is stored on an external disk (e.g., USB flash drive) and downloaded to the local computer whenever it needs to be used. This has a usability advantage over a regular soft token in that it is now fully portable. At first glance, it may also seem that it has a security advantage because now the user's private key is not stored on the user's laptop or local machine. Thus, an attacker needs *physical access* to the user's external device, and such a device is arguably easier to protect than a laptop or personal computer.

Despite this, we argue that the security advantages of this solution over a regular soft token are very mild. First, if the external device is obtained – even for a short amount of time – it may be possible to extract the soft token from it (without the user detecting this). Second, the fact that the private key is downloaded to the computer means that it is vulnerable to any malicious software that sits on that computer. Furthermore, in practice, much secret information finds its way onto swap files. Thus, using methods from computer forensics, it may be possible to find the user's secret key from any computer where it was used.<sup>3</sup> Finally, due to the portability advantages, users may be

3. There are ways of preventing a secret key from being swapped out. However, not all applications will do this.

External storage hardware devices provide portability of users' keys but do not greatly enhance security.

tempted to use their keys on untrusted machines, something that can have disastrous effects. We therefore conclude that the security of such a system is only mildly better than that of a soft token.

We remark that such a solution may also have password protection. This improves the security, but still not much beyond the security of a password-encrypted soft token. Furthermore, the private key will still be vulnerable to viruses when it is imported.

## Hardware Tokens with Cryptographic Capabilities

The main problem with the above system is that the private key is imported to the user's local computer. The solution to this is therefore to have the external device carry out the cryptographic operations by itself. That is, the external device includes a microprocessor. In order to decrypt a message, the local machine sends the encrypted message to the external device which decrypts it and hands it back (this operation is typically also password protected). This means that the *private key is never exported from the device*. This is a fundamental step forward towards a highly secure solution. Let us consider again the threats of malicious software and physical access:

1. *Malicious software*: Assume that a user connects a hardware token with cryptographic capabilities to an infected machine, and assume also that the user enters its token password at some stage. In this case, a virus that resides on the machine may be able to use the token (to decrypt or sign a message). However, the damage is limited to the time that the token is connected to the machine, and such a virus is non-trivial to write.
2. *Physical access*: First note that since the key is never exported, the concern here is only when an attacker obtains physical access to the device; the local machines used have no information on the key. Now, if an attacker manages to obtain a user's device, the situation is problematic. Security in this case comes down to the question of what physical protection is provided by the device. Basic hardware tokens provide only minimal physical protection and by breaking them open it is possible to extract their internal code and secrets without too much difficulty. On the positive side, this will often require the attacker to destroy the token, and thus the attack will be detected by the user (whose token has gone missing).<sup>4</sup> In some cases, this price is also too high.
3. A potentially more damaging type of attack on hardware tokens is called a *side channel attack*. Given physical access to a token, an attacker can learn the user's private keys by measuring phenomena like the time and power consumed during decryption and signing operations (although this sounds far-fetched, such attacks are actually very effective and do not damage the token). Simple hardware tokens provide little protection against these types of attacks and are therefore relatively vulnerable to them. We refer the reader to [1, 3, 4] for more information on these types of attacks.

**Hardware tokens with cryptographic capabilities provide higher security by performing on-board cryptographic operations, so private keys are never exported from the device.**

<sup>4</sup>. Of course, whenever a user loses its token, it is necessary to replace all of the user's certificates. We hope that this is the common practice.

Smart cards are designed and rigorously tested to defend against various types of attacks, providing the highest level of protection of users' keys.

## Smart Card Tokens

A smart card token is essentially a hardware token with a smart card, which provides cryptographic capabilities and also includes protection against different types of attacks that can be carried out given physical access to the card. Modern smart cards are highly sophisticated and provide protection against a wide variety of attacks; see [6] for a good survey of the different threats and protections. As with almost any security solution, with enough time and money, it is also possible to break almost any smart card. However, with a good smart card, the cost of such an attack is likely to be greater than the benefit.<sup>5</sup>

Regarding malicious software, the situation here is the same as for hardware tokens. Thus, there is some threat. However, as we have mentioned, this threat can be significantly reduced by using appropriate anti-virus software and so on. Since the damage of infection in this case is lower, the security provided by smart card tokens is still very high. Another important point regarding smart cards is that they are typically tested very rigorously. Thus, software bugs that could cause security holes are unlikely to exist.

## Conclusions on the Security of Hardware Tokens

Based on our above analysis, it is clear that any secure solution must be based on an external device with cryptographic capabilities. In some settings, a simple hardware token with cryptographic capabilities may be sufficient. However, a smart card token is highly preferable due to the fact that it provides stronger protection against physical and other attacks. Of course, no security solution is foolproof. However, smart card based solutions provide very strong guarantees that are sufficient for almost all industrial applications.<sup>6</sup> Given that secure eCommerce requires that private keys remain private, it is only really possible to securely deploy PKI-based solutions with strong smart card tokens that ensure the privacy of keys. The highly important property of non-repudiation is also preserved in its strongest sense when smart card tokens are used. (Of course, it is always possible to claim theft. However, the burden of proof is likely to be on the smart card holder in such a case.)

---

5. Note that every smart card contains different private keys. Therefore, any physical attack has to be carried out separately – from scratch – on every target card. When the attack is quick and cheap, as in the case of a simple hardware token, such attacks may be cost beneficial. However, when the attack is very expensive, it will rarely be worth carrying out.

6. Military and intelligence need higher levels of security and will thus often ensure that their machines are not connected to any external network. This greatly reduces the risk of virus infection. Of course, needless to say, physical access to such facilities is also greatly limited.

## References

- [1] R. Anderson and M. Kuhn. Tamper Resistance – A Cautionary Note. The *2nd USENIX Workshop on Electronic Commerce*, pages 1–11, 1996.<sup>7</sup>
- [2] W. Diffie and M.E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22, pages 644–654, 1976.
- [3] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology – CRYPTO'96*, Springer-Verlag (LNCS 1109) pages 104–113, 1996.
- [4] P. Kocher, J. Jaffe and B. Jun. Differential Power Analysis. In *Advances in Cryptology – CRYPTO'99*, Springer-Verlag (LNCS 1666) pages 388–397, 1999.
- [5] R.L. Rivest, A. Shamir, and L.M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [6] M. Witteman. Advances in Smartcard Security. *Information Security Bulletin*, July 2002.
- [7] The Caligula Virus. <http://www.internetnews.com/bus-news/article.php/64191>.
- [8] B. Krebs. DNA Key to Decoding Human Factor. *The Washington Post*, March 28, 2005. <http://www.washingtonpost.com/wp-dyn/articles/A6098-2005Mar28.html>.
- [9] Obfuscated Code, Wikipedia. [http://en.wikipedia.org/wiki/Obfuscated\\_code](http://en.wikipedia.org/wiki/Obfuscated_code).
- [10] Pretty Good Privacy (PGP), <http://www.pgp.com>.

---

7. This paper cites attacks on smart cards. However, the attacks mentioned there are typically only applicable today to simple hardware tokens and not to modern smart cards. This is due to progress made in the last decade.

## **About Aladdin**

Aladdin (NASDAQ: ALDN) is a leader in digital security, providing solutions for software digital rights management and Internet security since 1985. Serving more than 30,000 customers worldwide, Aladdin products include: eToken™, providing cost-effective strong user authentication and password management solutions; the eSafe® line of integrated content security solutions, protecting networks against malicious, inappropriate and non-productive Internet-borne content; and HASP®, a digital rights management (DRM) suite of protection and licensing solutions featuring the number one hardware-based system in the world.

## **About eToken**

Aladdin eToken provides cost-effective strong user authentication and password management solutions. It provides enhanced security and ensures safe information access; improved password and ID management; and secure and convenient mobility of digital credentials for both PKI-based and non-PKI authentication solutions.

About the size of an average house key, the smart-card-based eToken is easy to use and highly portable, providing users with powerful strong authentication and management of digital credentials. It is used for secure online transactions, secure network logon, secure VPN access, single sign-on, secure email, and numerous other applications. eToken devices are available in both USB and traditional smart card form factors, with physical access (RFID) capabilities and one-time password technology.

For more information please visit [www.Aladdin.com/eToken/](http://www.Aladdin.com/eToken/)



For more contact information, visit: [www.Aladdin.com/contact](http://www.Aladdin.com/contact)

<b>North America</b>	T: 1-800-562-2543, 1-847-818-3800	F: 1-847-818-3810
<b>International</b>	T: +972-3-636-2222	F: +972-3-537-5796
<b>UK</b>	T: +44-1753-622-266	F: +44-1753-622-262
<b>Germany</b>	T: +49-89-89-4221-0	F: +49-89-89-4221-40
<b>Benelux</b>	T: +31-30-688-0800	F: +31-30-688-0700
<b>France</b>	T: +33-1-41-37-70-30	F: +33-1-41-37-70-39
<b>Spain</b>	T: +34-91-375-99-00	F: +34-91-754-26-71
<b>Israel</b>	T: +972-3-636-2222	F: +972-3-537-5796
<b>Asia Pacific</b>	T: +852-2166-8605	F: +852-2166-8999
<b>Japan</b>	T: +81-426-607-191	F: +81-426-607-194



0 7 3 8 9