

SafeSign Identity Client Middleware

White Paper



No part of this document may be reproduced or transmitted in any form or by any means electronic, mechanical or otherwise, including photocopying and recording for any purpose without written permission of A.E.T. Europe B.V.

Individuals or organisations, which are authorised by A.E.T. Europe B.V. in writing to receive this information, may utilise it for the sole purpose of evaluation and guidance.

Warning Notice

All information herein is either public information or is the property of and owned solely by A.E.T. Europe B.V. who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

This information is subject to change as A.E.T. Europe B.V. reserves the right, without notice, to make changes to its products, as progress in engineering or manufacturing methods or circumstances warrant.

Installation and use of A.E.T. Europe B.V. products are subject to your acceptance of the terms and conditions set out in the license Agreement which accompanies each product. Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/ or industrial property rights of or concerning any of A.E.T. Europe B.V. information.

Cryptographic products are subject to export and import restrictions. You are required to obtain the appropriate government licenses prior to shipping this Product.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, A.E.T. Europe B.V. makes no warranty as to the value or accuracy of information contained herein. The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, A.E.T. Europe B.V. reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

A.E.T. EUROPE B.V. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE INFORMATION CONTAINED HEREIN, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL A.E.T. EUROPE B.V. BE LIABLE, WHETHER IN CONTRACT, TORT OR OTHERWISE, FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER INCLUDING BUT NOT LIMITED TO DAMAGES RESULTING FROM LOSS OF USE, DATA, PROFITS, REVENUES, OR CUSTOMERS, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION CONTAINED IN THIS DOCUMENT.

© Copyright A.E.T. Europe B.V., 1997 - 2007.

All rights reserved.

SafeSign is a trademark of A.E.T. Europe B.V. All A.E.T. Europe B.V. product names are trademarks of A.E.T. Europe B.V.

All other product and company names are trademarks or registered trademarks of their respective owners.

Credit information:

This product includes cryptographic software written by Eric A. Young (eam@cryptsoft.com)

This product includes software written by Tim J. Hudson (tjh@cryptsoft.com).

Contact Information: A.E.T. Europe B.V.

IJsselburcht 3
NL-6825 BS
P.O. Box 5486
NL-6802 EL Arnhem
The Netherlands
Tel. +31-26-365 33 50
Tel. Support +31-26-365 35 43
Fax +31-26-365 33 51



info@aeteurope.nl / support@aeteurope.nl
<http://www.aeteurope.com/>

SafeSign Identity Client is a product developed by
A.E.T. Europe B.V.

Copyright © 1997 - 2007 A.E.T. Europe B.V.,
Arnhem, The Netherlands.
All rights reserved.



Document Information

Filename: SafeSign Identity Client Middleware
White Paper

Document ID: White_Paper_SafeSign-IC_v2.1

Document revision history

Version	Date	Author	Changes
1.0	08-12-2003	Drs C.M. van Houten	First edition
1.1	02-08-2004	Drs C.M. van Houten	Edited for SafeSign Identity Client Standard Version 2.0 for Windows (release 2.0.6)
2.0	08-04-2005	Drs C.M. van Houten	Edited for SafeSign Identity Client Standard Version 2.1 for Windows (release 2.1.3)
2.1	28-02-2007	Drs C.M. van Houten	Edited for SafeSign Identity Client Standard Version 2.3 for Windows (release 2.3.2)

WE RESERVE THE RIGHT TO CHANGE SPECIFICATIONS WITHOUT NOTICE

Table of contents

Warning Notice	I
Document Information	II
Table of contents	III
About the Product	V
About this Document	VI
1 Introduction	1
1.1 A.E.T. Europe B.V.	1
1.2 Products and Solutions	1
1.3 Location	1
2 Security Basics	2
2.1 Public Key Cryptography	3
2.1.1 Signing	4
2.2 Digital Certificate	5
2.3 Certificate Authority	6
2.4 Summary	6
3 Tokens	7
3.1 Smart Card	8
3.2 USB token	8
3.3 Benefits	8
3.3.1 Technical	8
3.3.2 Commercial	9
3.4 Summary	9
4 Middleware	10
4.1 The Standards	10
4.2 Summary	10
5 SafeSign Identity Client Middleware	11
5.1 SafeSign Identity Client structure	11
5.2 SafeSign Identity Client Components	12
5.2.1 SafeSign Identity Client PKCS #11 Library	12
5.2.2 SafeSign Identity Client CSP	13
5.2.3 SafeSign Identity Client Token Utilities	13
5.2.4 SafeSign Identity Client Store Provider	14
5.2.5 PKI applet	14
5.2.6 Applet loader	14
5.2.7 SafeSign Identity Client PKCS #15	14
5.3 SafeSign Identity Client Benefits	15
6 Applications	16

7	Case Study: Microsoft Windows logon.....	17
8	Case Study: Microsoft Windows 2003 Terminal Services	19
9	Case Study: Virtual Private Networking	20
9.1	Introduction	20
9.2	VPN and PKI	20
9.3	SafeSign Identity Client and VPN.....	20
9.4	Applications.....	21
9.4.1	Microsoft VPN	21
9.4.2	Cisco VPN	21
9.4.3	NCP VPN / PKI client	21
9.4.4	Check Point VPN: SecureClient/ SecuRemote	22
9.4.5	SafeNet SoftRemote	22
	Glossary	a

About the Product

SafeSign Identity Client is a software package that can be used to enhance the security of applications that support hardware tokens through PKCS #11 and Microsoft CryptoAPI.

The SafeSign Identity Client package provides a standards-based PKCS #11 Library and Cryptographic Service Provider (CSP), allowing users to store public and private data on a personal token, either a smart card, USB token or SIM card. It also includes the SafeSign Identity Client PKI applet, enabling end-users to utilise any Java Card 2.1.1 and higher compliant card with the SafeSign Identity Client middleware.

Combining full compliance with leading industry standards and protocols, with flexibility and usability, SafeSign Identity Client can be used with multiple smart cards / USB tokens, multiple Operating Systems and multiple smart card readers.

SafeSign Identity Client allows users to initialise and use the token for encryption, authentication or digital signatures and includes all functionality necessary to use hardware tokens in a variety of PKI environments.

SafeSign Identity Client Version 2.3 for Windows supports the following tokens (as described in the product description):

- STARCOS® smart cards developed by Giesecke & Devrient GmbH (G&D): SPK2.3, SPK2.3 RawRSA, SPK2.4, SPK2.4 FIPS, SPK2.5 Dual Interface (DI) and STARCOS 3.0;
- The G&D StarKey100 (M) and StarKey200 USB token with the completed STARCOS SPK 2.3 / 2.4 operating system;
- The G&D StarKey220 HID token with the completed STARCOS SPK 2.3 operating system;
- The G&D StarKey400 and StarKey400 M (with flash memory) USB token with Sm@rtCafé Expert 64k;
- The Eutron Cryptoidentity / CryptoCombo ITSEC-P with the completed STARCOS SPK 2.3 operating system, and the Cryptoidentity / CryptoCombo FIPS USB token with the completed STARCOS SPK 2.4 operating system;
- The KeyCorp Multos v4.2 48K card and the KeyCorp Multos v4.2 64K card;
- Java Card v2.1.1 / Open Platform 2.0.1 compliant Java smart cards:
Aspects OS755 v2.8, Axalto e-gate, Axalto Cyberflex Access Developer 32K, Axalto Cyberflex 64Kv1 and 64Kv2, Axalto Cyberflex Palmera, G&D Sm@rtCafé Expert 2.0, G&D STARSIM Java, Gemplus GemXpresso 211pk/Pro R3, IBM JCOP 20/21/30/31, MartSoft Java card, Oberthur CosmopolIC v4 and Orga JCOP 20/30.
- Java Card v2.2+ / GlobalPlatform 2.1.1 compliant Java smart cards:
Aspects OS755 (Java Card 2.2), G&D Sm@rtCafé Expert 64, G&D Sm@rtCafé Expert 3.0, G&D Sm@rtCafé Expert 3.1, IBM JCOP21 (Java Card 2.2), IBM JCOP31 (Java Card 2.2), IBM JCOP41, Oberthur ID-One Cosmo64 v5.2, Oberthur ID-One Cosmo 64 RSA D/T v5.4 and Oberthur ID-One Cosmo 32 RSA v3.6.

SafeSign Identity Client comes in a standard version with an installer for the following Windows environments¹:

- Windows 2000, Windows XP (Professional), Windows 2003 Server, Windows Vista.

In principle, SafeSign Identity Client supports any PC/SC compliant smart card reader. However, to avoid power problems, smart card readers must be capable to provide at least a current of 60mA. PC/SC driver software is available from the web site of the smart card reader manufacturer.

For more information, refer to the latest SafeSign Identity Client Product Description.

¹ Windows NT 4.0 is supported up to SafeSign Identity Client 1.0.9.04, in line with Microsoft's end-of-life policy. Windows 98 and Windows ME are supported up to SafeSign Identity Client 2.3.0 (< 2.3.0), in line with Microsoft's end-of-life policy.

About this Document

This document is a white paper describing the features of the SafeSign Identity Client middleware, and provides an overview of the concepts related to the use of cryptographic tokens, including Public Key Infrastructure (PKI), cryptography, digital certificates and key pairs.

It describes how and why the use of tokens provides a more secure means of authentication, by replacing the use of one-factor authentication (based on knowledge of username and password) with two-factor authentication (based on possession of token and knowledge of the PIN). It goes on to describe what middleware is and how the SafeSign Identity Client middleware enables the use of cryptographic tokens in PKI-enabled applications.

Finally, a number of usage cases is dealt with, intended to make it apparent for the user how tokens can be used in practice, taking into account the overview of cryptographic features and functions described before.

This document is not intended to give a full or accurate overview of public key cryptography and how the process of encrypting / decrypting and verification of signed messages works, but tries to give the reader some background as to the why and how of public key cryptography and the use of tokens.

1 Introduction

1.1 A.E.T. Europe B.V.

The Dutch company A.E.T. Europe B.V. (AET), headquartered in Arnhem and founded in 1998, is committed to IT security. AET offers its customers ingenious solutions based on worldwide accepted standards for e-commerce and information security.

1.2 Products and Solutions

AET is setting new standards in the crucial business area of developing middleware for smart cards and USB tokens with its *SafeSign Identity Client*. Another of AET's bestsellers is *BlueX Digital ID Management*. This well-known and extremely adaptable system supports the simplification and automation of individual steps for the digital ID management process.

For more information, please visit [http:// www.aeteurope.com](http://www.aeteurope.com).

1.3 Location

AET headquarters are based in Arnhem, close to the German border, and are easy accessible from all major directions. Apart from a well-equipped in-house test and demo facility, AET offices include a development centre.

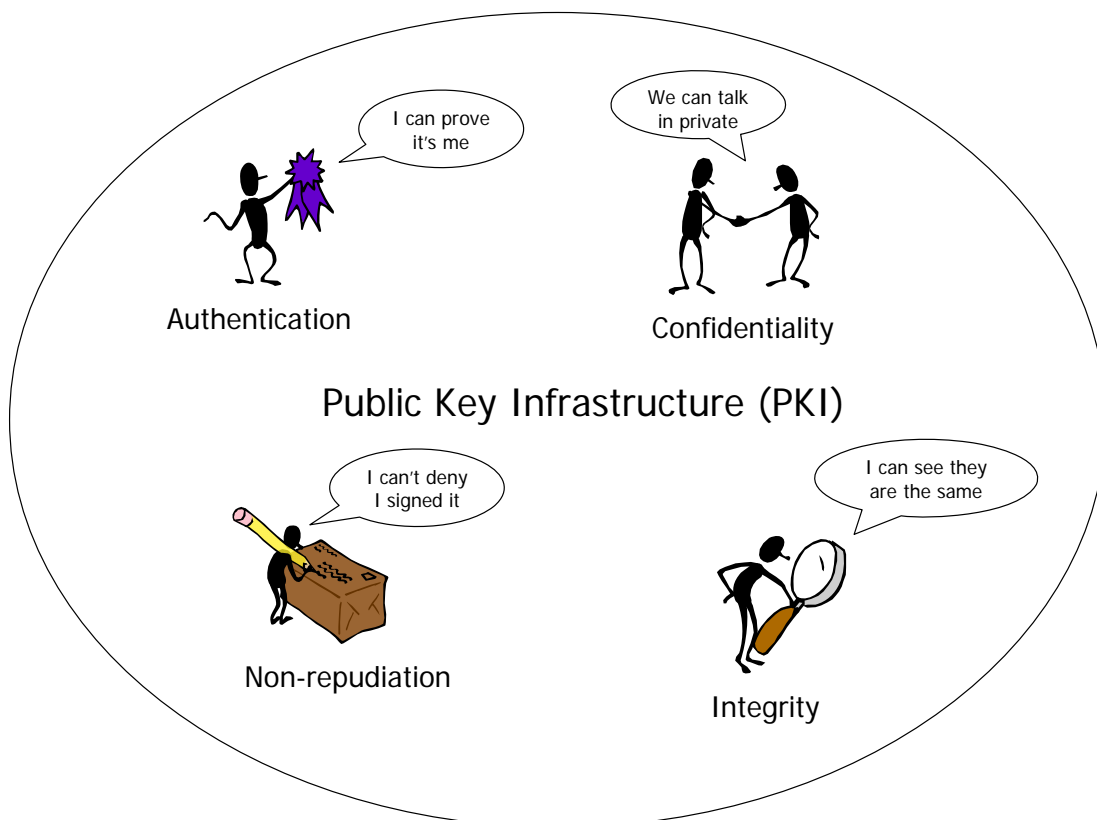
2 Security Basics

In our present time, we can hardly imagine doing business without using the Internet in one way or another. The advantages of this open and accessible channel are obvious. But more increasingly, the Internet is now not only used for publishing general company and product information, but also for software distribution and e-commerce. E-mail messages are now not only just memos and notes, but may also contain contracts and sensitive financial information. To secure this kind of communication and transaction using a medium so open and anonymous by nature, additional ways have to be found.

When we consider what is most important in communications face to face, this would be the fact that we are able to identify the other party with certainty, as well as rely on the integrity of information presented. In order to achieve the same for digital communications, this would require a system that should make it possible to identify another user (person, computer or other electronic entity).

Public Key Infrastructure (PKI) is such a framework that can be built into existing network systems (such as the Internet and a company's Intranet) and security policies and in which the identity of its participating members is the key factor. A PKI is a security architecture that has been introduced to provide an increased level of confidence for exchanging information over an increasingly insecure Internet and Intranet.

The benefits provided by a PKI revolve around a number of security policies, i.e. confidentiality, authentication, integrity, and non-repudiation. These policies together form the essence of PKI, creating a web of trust.



When giving these security policies a technical and business interpretation, they would serve the following purposes:

- **Authentication:** The sender and receiver can be sure that they are actually communicating with the entity they wish to be communicating with -> certainty of the source and destination of that information;
- **Integrity:** The sender and receiver can be sure that the message sent and the message received are the same -> certainty of the quality of information sent and received electronically;
- **Confidentiality:** The sender and receiver can be sure that only the entity intended to see the message may actually see it -> certainty of the privacy of that information;
- **Non-repudiation:** the receiver can be sure that the sender cannot deny having sent the data (at a later date) -> assurance that the information may be introduced as evidence in a court or law;

In order for a PKI to satisfy the policies described above, it uses public key cryptography. Public key cryptography is discussed in the next paragraph.

2.1 Public Key Cryptography

In order to fulfil the functions described above (for example to provide authentication, to ensure confidentiality), cryptographic techniques are used. Cryptography is the use of mathematical algorithms to encode or scramble information, to transform plain text into cipher text.

In symmetric key cryptography, also known as secret key cryptography, people use the same key for both encryption and decryption. This means that both sender and receiver need to share a key that is known only to each other. Key management is a significant problem in symmetric key systems. There are other significant problems in symmetric key cryptography systems. Symmetric key systems are easier to compromise, because symmetric keys are possessed (and used) by more than one person. If the system has been compromised, it is almost impossible for investigators to trace fraudulent activities back to the responsible person.

Symmetric key systems are very effective when only a few people must share their symmetric keys. However, they are impractical in large organizations. In 1976 two researchers, Whitfield Diffie and Martin Hellman, proposed an alternative to symmetric key cryptography. That alternative is called public key cryptography.

Public key cryptography uses a pair of mathematically related cryptographic keys, a key pair: a public key that is widely available, and a different private key known only to the person, application or service that owns the keys.

If you have the public key, you cannot easily calculate the corresponding private key, but vice versa, if you have the private key, you can easily calculate the corresponding public key. The public key can be transmitted unencrypted over insecure lines, since it is not a secret, while the private key must be kept secret. Together, the public and the private key are called a key-pair.

If one key is used to encrypt information, then only the related key can decrypt that information, or with one key you 'sign' data and with the other one you verify the integrity of that data. This is the essence of public key cryptography: an operation performed with the one key can only be reversed with the other key. One key is used for encoding or encrypting information, the other key is used to decode or decrypt information. This is the basis for both secure e-mail and digital signatures.

2.1.1 Signing

Here is an example of how this works in case of sending a signed e-mail message:

The sender will first create a unique abstract / compressed version of the message (a so-called message digest or fingerprint) using a hash algorithm. This uniquely identifies the message, as changing anything in the message would lead to a different message digest. The sender then signs the message digest with his private key and sends the resulting value along with his e-mail message as a digital signature.

The recipient of the message will decrypt the signed message digest with the public key of the sender and compare this message digest with the message digest he derives by using the same hash algorithm on the message. Thus, he can be sure that the message did actually come from the sender, for only the sender has the private key used to create the digital signature (authentication) and the message has not been modified since it was signed, for the message digest he arrives at is the same (integrity).

Obviously, this is not something the sender and receiver will do manually; usually their e-mail program will do this for them. Suppose you want to send a signed message, all you have to do is to select this option in your e-mail program (for example Outlook Express) and it will be signed. Vice versa when you receive a signed message.

When the sender signs a message with his private key, the recipient can be sure that it is the actual sender he is communicating with. However, when an e-mail message is signed, third parties may still be able to read the message, as it is in plain text. In order to protect against unauthorized or unwanted monitoring, you will need to encrypt the message. When encrypting a message, you transform the plain text of the message into cipher text, so that only the intended recipient can read it. This is done using the public key of the recipient, so that the recipient can transform the text encrypted with his public key back to plain text using his private key.

From the above, it is clear that the private key is the element of a key pair that should be kept in a safe place. The private key is used for authentication and non-repudiation, in the same way a hand-written signature could be used to authenticate the signer. Private keys can be stored on a computer's hard disk (protected by a password) or on a special cryptographic device called a token (see [chapter 3](#)). If a private key is stored on a token, it can only be used while the token is inserted into the computer and when the user enters the correct PIN¹ for the token. Public keys are usually embedded in digital certificates. Digital certificates are easy to distribute, either as an attachment to an e-mail message or through a Web browser (see [paragraph 2.2](#)).

¹ Or biometric credential, where a biometric feature might replace the PIN or be additional (three-factor authentication: something you have, something you know, something you are).

2.2 Digital Certificate

A digital certificate may be considered as the digital equivalent of a driver's license or passport, which can be used to authenticate or identify its owner in a digital world. It binds the physical existence of a person to his electronic existence or presence.

According to IETF (Internet Engineering Task Force) RFC 3280 paragraph 3.2, certificates and PKI are: " *Users of a public key require confidence that the associated private key is owned by the correct remote subject (person or system) with which an encryption or digital signature mechanism will be used. This confidence is obtained through the use of public key certificates, which are data structures that bind public key values to subjects. The binding is asserted by having a trusted CA¹ digitally sign each certificate. The CA may base this assertion upon technical means (a.k.a., proof of possession through a challenge-response protocol), presentation of the private key, or on an assertion by the subject. A certificate has a limited valid lifetime, which is indicated in its signed contents. Because a certificate's signature and timeliness can be independently checked by a certificate-using client, certificates can be distributed via untrusted communications and server systems, and can be cached in unsecured storage in certificate-using systems.*"

In short this means that due to the process of verification by a CA, a public key is associated to an identity. This association is done by the CA by signing / binding the identity and the public key together. A certificate is the bound public key and the identity signed by a CA. Through this association of information on its owner and any relevant details with the public key, the certificate becomes a means for other people to verify that the owner does indeed hold the private key, and therefore, must really be who he / she says he / she is.

Digital certificates also identify who issued the certificate, as they are issued by a trusted licensing body, called a certificate authority (CA)². The CA digitally signs every certificate it issues. Anyone can test the CA's signature by retrieving the CA's public key and decrypting the signature.

Besides identification data and a public key, a certificate contains more information, for example:

- So-called *key usages*. Key usages indicate for which usage(s) a certificate may be used, for example it is possible to create a certificate that may only be used for signing purposes, or it is possible to create a certificate that can be used for 'smart card logon' to a network.
- A certificate contains the validity time period of the certificate.

¹ A Certificate Authority, as described in paragraph 2.3.

² The integrity of a CA and the corresponding trust in it, lies in the fact that it is understood that the CA has followed certain procedures to issue certificates and that these are indeed implemented and followed correctly, so that it is not possible that a certificate is issued without the correct procedure(s). The integrity of a CA may be safeguarded for example by the fact that: the private key of the CA is generated and protected by a Hardware Security Module (HSM), the computer on which the CA is installed is not accessible from a network or only via additional authentication means; the computer on which the CA is installed is located in a protected room.

2.3 Certificate Authority

Certificate Authorities (CAs) are an essential component of a PKI. They are trusted authorities that issue digital certificates. CAs vouch for the identity of the individual/enterprise to whom they are issuing a certificate.

The CA verifies your personal information and the integrity of your public key. After the verification process, the CA signs your public key, stores appropriate personal information and your public key on the digital certificate, and issues your digital certificate to you.

2.4 Summary

With the emergence of doing business over the Internet, protection of sensitive data and corporate assets is becoming increasingly important. To protect computers and files based on usernames and passwords is no longer secure enough and requires the implementation of an overall management and security architecture, where the identity of its many different participants becomes the most important factor.

For the end-user, it is important that whatever system has been deployed takes care of the essential security policies of a PKI and protects his identity. He will most likely be working with PKI-enabled applications that do this for him, without having to personally hash the data he is sending. However, it must also be clear that his key pair and certificate are the essential components of an identity-based security system and that these need to be protected by adequate means. This will be dealt with in the next chapter, where the importance and convenience of the use of a token is described.

3 Tokens

Public key cryptography and PKI are important means for creating a security system that centres around the identity of the individual and that is capable of being used across an organisation, for all kinds of applications, including signed and encrypted e-mail, secure remote access, VPN, electronic forms, desktop security etc.

In such a system, where security lies in the identification and authentication of its participants involved, the means for an individual to identify himself and be identified, needs to be protected very carefully. It is obvious from the above, that if anyone would be able to obtain someone else's private key, he would be able to impersonate the other person, intercept and change mail. And if this does not sound serious when only viewed in the light of for example secure e-mail, imagine a scenario in which you give your bank an order to withdraw money from your account to transfer to a third party and that this is intercepted and changed to transfer a higher sum of money to another person.

Usernames and passwords are a traditional and well-known means to protect information, but they may not be strong enough to provide the security required by the implementation of a PKI for many different users, ranging from sales people who need access to pricing information available on the home office server, support people working from a customer site who need access to the central support database at the home office to employees who have to be able to log on different computers at different locations. The drawbacks of the use of usernames and passwords are equally well known: if users have to remember multiple passwords (which have to fulfil certain requirements), they tend to write down their password somewhere, which may present a serious security leak (the famous example of the yellow post-it notes on the PC display). Not to imagine the cost of password management systems and calls to the helpdesk to retrieve passwords (if one even knows the password is compromised).

Use of a username and password is one-dimensional; authentication is based on the fact that somebody has knowledge (of the username and password). To step up authentication, you can introduce another factor, not just knowledge, but also possession, i.e. the possession of a token and the knowledge of a password (or in token terms, a PIN). Even better, if the token that is used does not serve just as a storage medium (of username and password or of private and public key) but is actually able to perform cryptographic operations (such as generating a key pair) on the token itself and not in software. This would guarantee that without either the token or the PIN, no one would be able to access private information and that the private key cannot be extracted or read from the token.

Using such a token would safeguard that the key(s) stored on it may be accessed and used only within the token. The private key is directly created on the token and all cryptographic operations that require the use of the private key are performed directly on the token. The token's operating system prevents the key from being exposed outside the token. A user can access the token functions only via a PIN code.

This feature makes cryptographic tokens especially interesting in conjunction with public key based authentication systems to implement a two-factor authentication scheme, in which access can only be granted as the result of "something you have" (a token containing the key) and "something you know" (the PIN which unlocks the token).

So do you need to have a PKI in place to employ tokens or do you need tokens to employ a PKI? No: tokens can secure and enhance PKI technology (by storing your digital credentials on a tamper-resistant device), and at the same time PKI technology can enhance the use of tokens (as it provides a management architecture). Moreover, there are PKI-enabled applications available that do not need a PKI, but that do use cryptography and allow you to store your keys on a token.

It is important to realise that these kinds of tokens are very different from, for example, data storage cards (with a chip). These cards are not capable of performing cryptographic operations on board. So-called smart cards allow for more data storage (memory) and the data they contain, can be protected against unauthorized access and tampering. Memory functions such as reading, writing, and erasing can be linked to specific conditions, controlled by both hardware and software (for example the entry of a PIN before being allowed to read the token contents). Another advantage of smart cards is that they are more reliable and have longer expected lifetimes.

Tokens may come in two major forms: as a smart card, a credit-card sized electronic device; or a USB token, a key-like token that fits into the USB slot of a computer and that is small enough to be attached to a key chain.

3.1 Smart Card



A smart card is an electronic device the size of a conventional credit card. A smart card has a microprocessor chip to store and process electronic data and applications. When it is inserted into a smart card reader, the smart card chip receives power to communicate with the computer.

The advantage of the use of smart cards is that they have a familiar form factor that is easy to use. A disadvantage is that you need a smart card reader to work with them.

3.2 USB token



USB Smart tokens contain a tiny computer chip for securely storing information. They are technologically identical to smart cards, with the exception of their form factor and interface.

USB Smart Tokens are typically the size of a house key and are designed to interface with the universal standard bus (USB) ports found on millions of computers and peripheral devices.

The advantage of the use of USB tokens is that they are extremely portable. A disadvantage is that it is not possible to print USB tokens like you would smart cards, so you cannot personalize them.

3.3 Benefits

What are the benefits of the use of tokens from a technical point of view (“what can I do with tokens?”) and from a commercial point of view (“what can tokens do for me?”).

3.3.1 Technical

- The main technical benefit of the use of a token is the “double authentication” factor: through possession of the token and knowledge of the PIN, two-factor authentication is achieved, based on something you have (possession of the token), as well as something you know (knowledge of the PIN)¹. Someone in possession of only one of these elements will still not be able to access information that he is not intended to read. If someone would gain possession of the token, he would still need the PIN to access its functions and if someone would gain knowledge of the PIN, he would not be able to use it without possession of the token.
- The use of tokens is easy and familiar: people are used to cards (bank cards, insurance cards) with or without a protective code/ password / PIN.
- Tokens can easily be carried around (portability). This allows users to utilize their credentials at different locations. Unlike the use of a username and a password, if the user is in possession of his token, a person with malicious intentions cannot obtain it: you cannot steal something that is not there (where you might need it, for example, to get access to someone's computer).
- Tokens are tamper resistant, whereby physical control over private data is increased.
- Tokens are multi-application and multi-function.
- Storing credentials on a token is more secure, when compared to storage on hard drives, which are more vulnerable (may crash, become obsolete, are susceptible to theft and copying/deletion).

¹ There is also three-factor authentication possible, based on something you have (the token), something you know (the PIN) and something you are (a biometric property: fingerprint, iris etc.)

3.3.2 Commercial

- Smart cards and USB tokens currently offer the best combination of flexibility, security and cost among token technologies and will continue to offer more functionality at decreasing costs.
- Return on investment.
- The use of tokens decreases cost of administration and management of passwords.
- Through the use of tokens, companies can provide their customers with commercial applications, such as secure banking and access control to protected web sites.

3.4 Summary

Where the use of public key cryptography and the implementation of a Public Key Infrastructure (PKI) are involved, security and protection of digital identity becomes a top priority. Replacing usernames and passwords by stronger authentication, i.e. the use of a secure and tamper-resistant token, two-factor authentication is provided, based on something the user knows (the PIN for the token) and something the user has (possession of the token).

Both token form factors have advantages and disadvantages as opposed to each other. To mention a few, a USB token does not require a smart card reader and is easily transportable, as it fits on a key chain. Smart cards on the other hand, while having the same benefits with regard to security and functionality, may contain additional identification information on its owner (such as a photograph).

Note that it is not necessary to have a PKI in place when you want to use tokens. But when you do have a PKI in place, the use of tokens may upgrade your security and the integrity of the system.

When you want to use a token, you would need software to enable you to personalise and use your token with your PKI-enabled applications. This is what middleware is and does.

4 Middleware

Though the possibilities and benefits of the use of tokens to protect personal and private credentials may be clear from the above, the question remains, how to achieve the interaction between the token (and the smart card reader) and the PKI systems or PKI-enabled applications. In order to do so, you need a piece of software that is often referred to as middleware (for cryptographic tokens).

This middleware should be capable of communicating both with the lower level related to the smart card and reader as well as to the higher level, the applications, to enable smart card interoperability. Thus, the middleware sits on a strategic level and is on the one hand dependent on the PC/SC layer and reader drivers functioning properly, as well as the applications (for example Internet Explorer) to correctly interpret the standards.

4.1 The Standards

Two standards are important when it comes to integrating tokens into applications and using middleware: PKCS#11 and Microsoft CryptoAPI. An application that wants to include support for tokens will include either of these two.

PKCS is a set of standard protocols developed by RSA Laboratories¹ for making secure information exchange possible. The standards include RSA encryption, password-based encryption, extended certificate syntax, and cryptographic message syntax for S/MIME (RSA's proposed standard for secure e-mail). PKCS #11 is the Cryptographic Token Interface Standard. The standard is available at: <http://www.rsa.com/rsalabs/node.asp?id=2133>.

The Microsoft CryptoAPI application programming interface (API), and CAPICOM provide services that enable developers to add security based on *cryptology* to applications. CryptoAPI includes functionality for encoding to and decoding from *ASN.1*, encrypting and decrypting data, authentication using *digital certificates*, and digital certificate management using certificate stores. CryptoAPI and CAPICOM support both PKI and symmetric key cryptography.

4.2 Summary

When tokens (either in the form of a smart card or USB token) are to be deployed in an organisation, stable software that is based on industry standards and protocols is required.

This software should provide either a standards-based PKCS#11 and/or Microsoft CryptoAPI (CSP) implementation, fully compliant with leading industry standards and protocols, while being so flexible that it can be used with multiple smart cards / USB tokens, multiple Operating Systems and multiple smart card readers.

This is where the SafeSign Identity Client middleware comes in (see [Chapter 5](#)).

¹ RSA Laboratories is the research center of RSA, The Security Division of EMC, and the security research group within the EMC Innovation Network.

5 SafeSign Identity Client Middleware

The SafeSign Identity Client middleware is designed to integrate cryptographic / digital signature tokens into numerous PKI-enabled applications in order to provide secure two- or three-factor authentication on all major platforms.

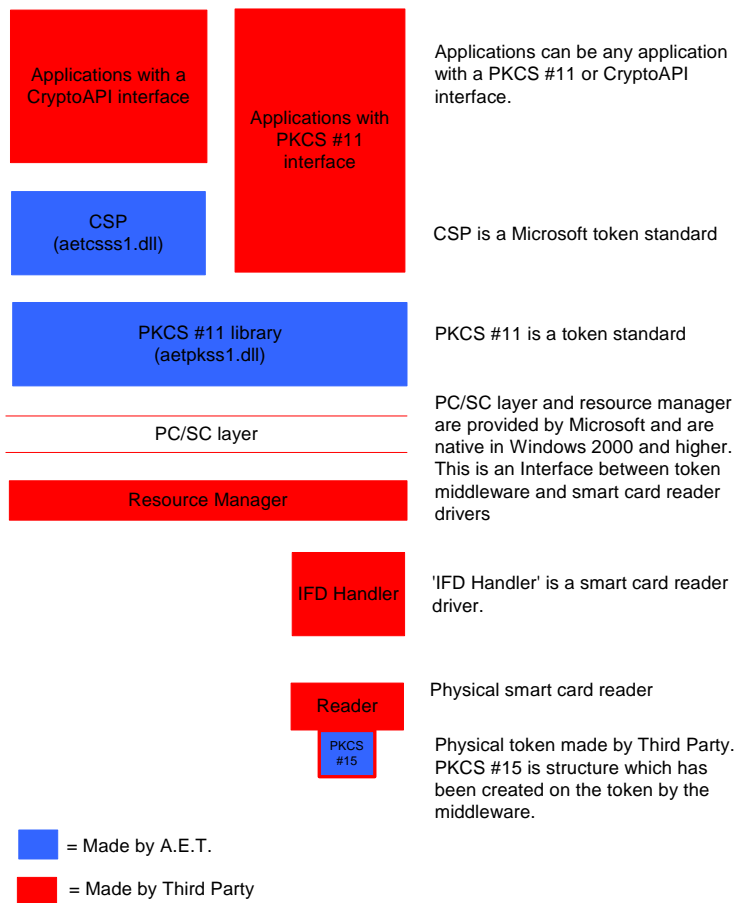
SafeSign Identity Client combines optimum flexibility with usability, by supporting a wide range of Operating Systems, applications, smart cards, USB tokens and smart card readers. Enhancing while at the same time simplifying security, SafeSign Identity Client is a reliable stronghold in a complex world of PKI.

SafeSign Identity Client provides a standards-based PKCS#11 and Microsoft CryptoAPI (CSP) implementation, fully compliant with leading industry standards and protocols, while being so flexible that it can be used with multiple smart cards / USB tokens, multiple Operating Systems and multiple smart card readers.

Basically any application that either supports PKCS#11 and/or CSP to work with tokens on any of the supported platforms can make use of the benefits and features of SafeSign Identity Client. One software version fits all.

5.1 SafeSign Identity Client structure

In order to understand how the SafeSign Identity Client middleware relates to the components it works with or depends upon, please view this diagram:



The SafeSign Identity Client middleware sits between the applications that interoperate with tokens using either PKCS#11 or Microsoft CryptoAPI and the interface between token and smart card reader, including the enabling layer of software on the Windows platform and the smart card reader drivers. The structure on the smart card (PKCS#15) is also created by the SafeSign Identity Client middleware.

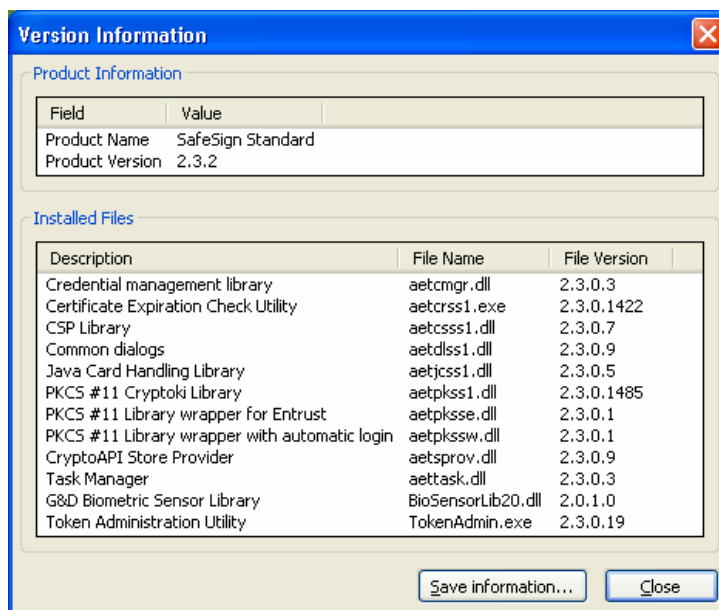
It is obvious from the above that the correct functioning of SafeSign Identity Client depends on the correct functioning of the PC/SC layer. At the same time, SafeSign Identity Client supports basically every smart card reader with a PC/SC interface and with enough power supply (with a minimum of 60 mA).

5.2 SafeSign Identity Client Components

SafeSign Identity Client consists of the following main components:

- PKCS #11 (v2.11) Library
- Cryptographic Service Provider (CSP)
- Token Management Utility (TMU) / Token Administration Utility (TAU)
- Store provider (automatic certificate registration)
- PKI applet (on Java cards)
- Applet loader
- PKCS #15 compatible card structure

The *Versions Info* menu option in the Token Management Utility / Token Administration Utility gives a full overview of the components pertaining to a particular version of SafeSign Identity Client:



5.2.1 SafeSign Identity Client PKCS #11 Library

PKCS #11 defines a technology-independent programming interface, called Cryptoki, for cryptographic devices such as smart cards. The SafeSign Identity Client PKCS #11 Library is the SafeSign Identity Client implementation of the PKCS #11 standard, integrating each PKCS #11 compatible application with a SafeSign Identity Client supported hardware token.

The SafeSign Identity Client PKCS #11 Library is built directly on top of PC/SC and can be regarded as the main component of SafeSign Identity Client. All cryptographic operations are handled by the PKCS #11 Library. Even when an application interacts with the CSP, the CSP will delegate cryptographic operations to the PKCS #11 Library.

The SafeSign Identity Client PKCS #11 Library is multi-application and multi-threaded, which means that multiple applications can access the PKCS #11 Library at the same time.

5.2.2 SafeSign Identity Client CSP

The SafeSign Identity Client CSP is built on top of the SafeSign Identity Client PKCS#11 library. Its primary aim is to provide cryptographic functionality that involves private credentials stored on the SafeSign Identity Client Token. AET has a clear design principle when it comes to CSP functionality. We only implement functionality when it is actually used (that is, when it is being used by an application).

The SafeSign Identity Client CSP integrates each Microsoft CryptoAPI compatible application with a SafeSign Identity Client supported hardware token.

The SafeSign Identity Client Cryptographic Service Provider (CSP) is included in the SafeSign Identity Client product to enable Microsoft CryptoAPI applications to access the PKI functionality provided by the SafeSign Identity Client Token. Note that this does not include only Microsoft applications (such as Internet Explorer and Microsoft VPN), but also other applications that use Microsoft CryptoAPI to integrate tokens, such as Check Point VPN, Novell Groupwise etc.

5.2.3 SafeSign Identity Client Token Utilities

SafeSign Identity Client Standard for Windows provides two separate token utilities, for end-users and administrators (in separate installers):

- The Token Management Utility (TMU) has been specifically designed for (end-)users. It allows users to perform some basic token operations (such as initialise token, change PIN) and provides users with an easy tool for viewing, importing and transferring their Digital IDs. Note that an administrator may have further restricted the functions available by default to the user.
- The Token Administration Utility (TAU) has been specifically designed for administrators, allowing them to perform advanced token operations.

The token utilities provide management functions for Digital IDs stored on the hardware token. The main function of the token utilities is to enable you to initialize the token with the SafeSign Identity Client PKCS#15 file structure and to personalize your token to be part of your secure applications.

The SafeSign Identity Client TMU and TAU are the central management utilities for both end users and administrators. Administrators have the ability to remove certain functionality for the end-user, such as the ability to change the PUK of the token or wipe its entire contents. It provides a user-friendly view of the digital IDs on the token, .i.e the key pair and certificate stored on the token.

Basic functionality provided by the token utilities is such functionality as viewing the registered Digital IDs, with associated actions such as checking expiration, viewing and deleting Digital IDs; token functions, such as change PIN / PUK, view PKCS#11 objects (administrator only), and such information as version info.

The user may not even see much of the token utilities, nor would he have to use them frequently, as he will be using SafeSign Identity Client (the components PKCS#11 and CSP) with his secure applications. Only if the user himself should initialize the token or be allowed to change his PIN, would he require the use of the token utility (TMU).

The ease-of-use and flexibility of the TMU reduces support calls, as it only offers the functionality the user really needs. The administrator is able to use the advanced options of the TAU to view the objects on the token (keys, certificates, data objects, etc.), to add an as yet unrecognised version of an already supported Java card and to dump the token contents. If support is needed, the TMU / TAU give a detailed overview of all SafeSign Identity Client components and their versions.

SafeSign Identity Client includes a PKCS#12 import function, to import Digital IDs, such as key pairs and certificates. This feature allows users to transfer Digital IDs generated on the PC onto the token. This greatly increases security and is very cost-effective, as no new Digital IDs have to be generated by the organisation, when the use of tokens is implemented.

5.2.4 SafeSign Identity Client Store Provider

The SafeSign Identity Client CryptoAPI Store Provider provides automatic registration of certificates when the hardware token is inserted and automatic deregistration of certificates when the hardware token is removed.

Especially in a multi-user environment, it is convenient to de-register certificates once the user has removed his token, or else all certificates will remain in the certificate store, which may be confusing if another user wants to connect to a secure web site and he is presented with all certificates registered in the Microsoft certificate store.

5.2.5 PKI applet

The SafeSign Identity Client PKI applet enables end-users to utilise any Java Card 2.1.1 and Java Card 2.2 (and higher) compliant card with the SafeSign Identity Client middleware. The applet manages the SafeSign Identity Client PKCS #15 file structure and the on-card RSA key-pairs. The applet is shipped together with SafeSign Identity Client in several different flavours that match specific classes of Java Cards and their specific benefits and limitations.

5.2.6 Applet loader

SafeSign Identity Client includes a universal Java Card applet loader. This applet loader can load the SafeSign Identity Client PKI applet out-of-the-box onto a variety of Java Cards equipped with a VISA/OP test key set (this includes most sample cards that can be purchased from Java Card vendors). The applet loader allows end-users rapid on-the-spot access to SafeSign Identity Client's capabilities. With the applet loader, SafeSign Identity Client offers end-users the possibility of testing a variety of cards without a hassle, allowing them to take the time to select the card that best matches their wishes.

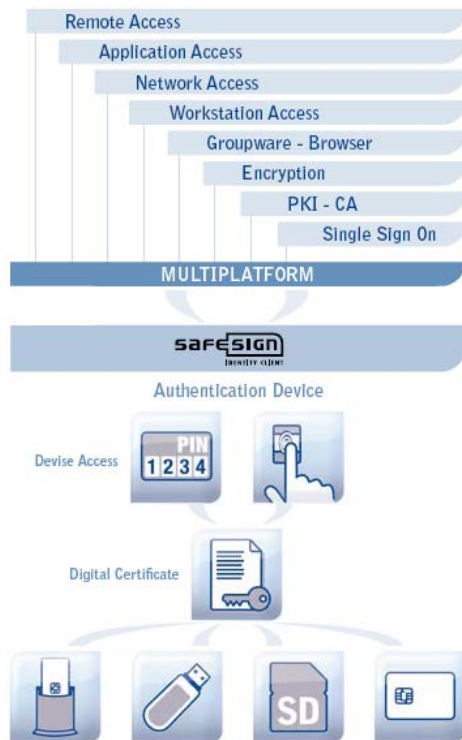
The applet loader can also be used in a production environment during a mass rollout of Java Cards. It can be configured to load applets onto cards with a production key set, and can even be used to change the key set of a card on the fly.

5.2.7 SafeSign Identity Client PKCS #15

The PKCS #15 Standard is the Cryptographic Token Information Syntax Standard and is intended to standardize the use of cryptographic tokens to identify themselves to multiple, standards-aware applications - regardless of the application's cryptographic token interface provider. The format specifies (a file and directory format) how keys, certificates and other application-specific data may be stored on cryptographic tokens. In doing so, it allows users to be able to use their tokens for identification purposes in all applications where this is necessary.

By the implementation of a PKCS#15 file structure on the token, SafeSign Identity Client not only adheres to an industry standard interface, but ensures that users can use SafeSign Identity Client and the tokens its supports with any application.

5.3 SafeSign Identity Client Benefits



The major advantage of SafeSign Identity Client is the fact that it is extremely flexible: it supports an ever-growing list of tokens, smart card readers and applications and can be used on multiple platforms, while it remains easy to use. SafeSign Identity Client is smart card / USB token vendor neutral, supporting both proprietary as well as Java Card operating systems from all leading vendors, which is precisely the reason why it is used throughout the world.

SafeSign Identity Client virtually works out of the box with all PKCS#11 and CSP interoperable applications and adheres to major industry-standards, which guarantee its interoperability. There is only one SafeSign Identity Client PKCS#11 and SafeSign Identity Client CSP library and that does it all: there is no need to use a different CSP for a particular application.

SafeSign Identity Client is the first multi-platform software, currently available on Linux, Mac OS X, Sun Solaris and Windows CE (.NET) / PocketPC / Windows Mobile.

SafeSign Identity Client is the only multi-language middleware, supporting over ten different languages. Languages can even be added on request: just hand in your translation and you will be able to use SafeSign Identity Client in your local language. Through its support for Unicode, Chinese (both Simplified and Traditional), Japanese, Thai, Turkish and Russian has been added.

SafeSign Identity Client is well known for its speed. Not only in its quick and straightforward installation, but also in its operation. Depending on token and reader combination used, Windows logon times may be as fast as 4 seconds. Maintaining and focusing on its high standards of interoperability and quality assurance, AET development is constantly seeking to improve the quality and speed of SafeSign Identity Client.

SafeSign Identity Client is value for money, not only is SafeSign Identity Client stable, flexible and user-friendly, but the excellent support for SafeSign Identity Client users and integrators is the reason why many customers buy SafeSign Identity Client today. SafeSign Identity Client is delivered complete with both user and administrator documentation, with installer documentation and programmer's guide is available on request.

6 Applications

SafeSign Identity Client supports an ever-increasing list of applications.

Refer to the latest product description for a full overview of applications supported. Just to mention a few:

- Citrix Presentation Server
- Microsoft Standalone and Enterprise Certificate Server
- Microsoft Internet Explorer
- Microsoft Outlook (Express)
- Microsoft VPN
- Mozilla Firefox and Thunderbird
- NCP VPN/PKI Client
- RSA Keon PKI
- Windows 2003 Terminal Server
- And many more . . .

7 Case Study: Microsoft Windows logon

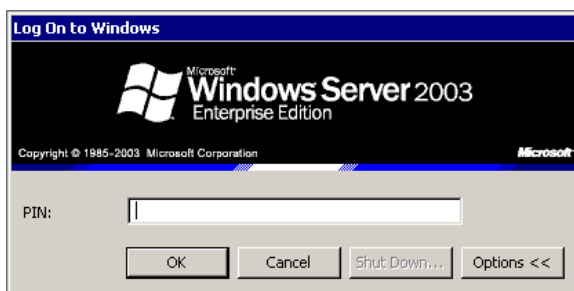
Microsoft Windows 2000 Server and 2003 Server integrate smart card capabilities in the Operating System. The Microsoft Windows 2000 / 2003 operating system includes a native Public Key Infrastructure (with its own Certificate Server) and introduces smart card authentication as an alternative to passwords to achieve strong network authentication.

Windows 2000 / 2003 enables administrators to set up an internal Certification Authority (CA) on the Windows 2000 / 2003 server and to issue digital certificates to users. Through the Smart Card Enrollment Station, administrators (with a valid enrollment agent certificate) can request certificates for users and can store each user's certificate directly on an individual token. A certificate can be specified for logon authorization (Smart Card Logon) only, or for both logon authorization and email security (Smart Card User). These certificates can be used to authenticate the user when logging on to the network, and for securing email with digital signatures and encryption. Integration with SafeSign Identity Client is easy and works virtually out of the box: when SafeSign Identity Client is installed on the Smart Card Enrolment Station, the SafeSign Identity Client CSP will allow authorized agents to generate keys and store certificates directly on a SafeSign Identity Client compatible token.

A significant element of the architecture is the CryptoAPI, through which applications can access strong cryptographic services for providing the required security characteristics. Users logging on to a domain must authenticate themselves, which may happen using a username and passwords. When using a smart card to do so, further possibilities are provided for, in particular, the login process can verify if the user has the proper credentials for accessing the system (authorization) and can check the Certificate Revocation List to confirm that the certificate presented is still valid. The user does not only authenticate himself in this way with a certificate, he should be known in the Active Directory of the Windows server, where authorization can be set what he may or may not access. Moreover, the administrator can configure smart card removal behaviour on the server, for example, to lock the workstation when the user removes his smart card to go to lunch.

It is outside the scope of this paper to discuss the requirements and procedures how to set up Windows 2000 / 2003 to issue certificates, but SafeSign Identity Client provides comprehensive user guides to do so, please refer to the *SafeSign Identity Client User Guide for Windows 2000* and the *SafeSign Identity Client User Guide for Windows 2003*.

Interactive Logon using a smart card begins when a user inserts a smart card into a smart card reader. This signals the Windows 2000/XP/2003/Vista operating system to prompt for a Personal Identification Number (PIN) instead of a username, domain name and password:



The card insertion event is equivalent to the familiar Ctrl-Alt-Del secure attention sequence used to initiate a password-based logon. However, the PIN the user provides to the logon dialog is used to authenticate only to the smart card and not to the domain itself. A public key certificate stored on the smart card is used to authenticate to the domain. After a user inputs a PIN to the logon dialog, the operating system begins a sequence of actions to determine whether the user can be identified and authenticated based on credential information the user has provided (PIN and smart card), among which certificate verification, digital signature verification and user account lookup.

SafeSign Identity Client seamlessly integrates with Windows 2000 / 2003 Certificate Services (PKI) and the Windows Smart Card Logon service enabling strong two-factor authentication that can be deployed for Windows 2000/XP/2003/Vista services and applications:

- Secure user authentication from a Windows 2000, XP, 2003 or Vista client to the Windows 2000/2003 domain
- Secure VPN client logon for remote access to corporate network
- E-mail encryption and e-mail signing with Microsoft Outlook
- Windows 2003 Terminal Services (see [Chapter 8](#)).

8 Case Study: Microsoft Windows 2003 Terminal Services

Terminal Services is a multi-session environment that gives remote computers access to a server desktop through "thin client" software.

In Terminal Server mode, you can access Windows-based applications or the Windows desktop itself on virtually any computing device - including those that cannot run Windows. When a user runs an application on Terminal Server, all of the application execution takes place on the server, and only keyboard, mouse, and display information traverses the network. Users see only their own individual sessions, which are managed transparently by the server operating system, and remain independent of any other client session. Through terminal emulation, Terminal Services allows the same set of applications to run on diverse types of desktop hardware.

An improved feature of Terminal Services in Windows 2003 Server is that it is now possible to log on to the Windows environment using smart card logon, as was not possible with previous versions of the Microsoft Terminal Server (Windows 2000 Server). A smart card that contains Windows logon credentials can provide those credentials to a Windows Server 2003 remote session for log-on. Note that this feature requires a client OS that can recognize the smart card first: Windows 2000, Windows XP, Windows 2003, Windows Vista and Windows CE .NET.



For smart card logon with Terminal Services in Windows 2003 Server to work, the following prerequisites are to be fulfilled:

- Active Directory installed and configured;
- Microsoft Certification Authority installed and configured;
- SafeSign Identity Client software installed on the Windows 2003 (Terminal) Server;
- Drivers for the smart card reader(s) installed on the Windows 2003 (Terminal) Server and on the client;
- A smart card reader installed on the client (if a smart card is used);
- A token (smart card or USB token) that is supported by SafeSign Identity Client, personalized with a Digital ID which supports smart card logon (see the *SafeSign Identity Client User Guide for Microsoft Windows 2003* for more details) for use on the client;
- Windows 2000, Windows XP, Windows Server 2003 or Windows Vista used as a client to connect to the Windows 2003 (Terminal) Server.

Once these requirements are met, users can start up a Terminal Server Connection and authenticate themselves by means of a token. Example: a user with a PC with Windows XP installed can set up a remote connection by inserting his token and entering the PIN. But it may even go further than that: the user can first use his token to logon to the domain (his Windows XP client is part of) and then set up a terminal connection for those applications and services that run on the server with the same token. And even when connected to the Terminal Server, SafeSign Identity Client can be used on the terminal itself: for example, the user can go to a secure web page and authenticate himself with his token and PIN.

9 Case Study: Virtual Private Networking

9.1 Introduction

A VPN is a private connection (a “secure tunnel”) between two or more machines that sends private data over a shared or public network, e.g. the Internet. The act of creating and configuring a VPN is known as ‘virtual private networking’.

VPN connections allow users working at home or on the road to connect in a secure fashion to a remote (corporate) server using the routing infrastructure provided by a public network (such as the Internet). The secure connection across the network appears to the user as a private network communication—despite the fact that this communication occurs over a public network—hence the name *virtual private network*.

When deploying a remote networking solution, an enterprise needs to facilitate controlled access to corporate resources and information. The solution must allow roaming or remote clients to connect to LAN resources, and the solution must allow remote offices to connect to each other to share resources and information (router-to-router connections). In addition, the solution must ensure the privacy and integrity of data as it traverses the Internet. The same concerns apply in the case of sensitive data traversing a corporate network.

9.2 VPN and PKI

Because the Internet facilitates the creation of VPNs from anywhere, private networks need strong security features to prevent unauthorised access and to protect private data as it is sent across the public network. PKI (Public Key Infrastructure) can provide the secure infrastructure to fulfil the above requirements.

A VPN creates a secure “tunnel” through the Internet. The VPN makes sure that the computer or device at each side is what it says it is. VPNs may authenticate users by what they know (passwords), what they have (a special code or key that both parties share, smart cards, or digital certificates issued by a trusted third parties), or some physical or biometric characteristic (fingerprint, voice print, or iris image). After each side agrees on the details for transmitting data—the form, encryption algorithm, and so on—the VPN creates a secure connection based on the rules defined for this specific path. The data is encrypted, sent through the secure tunnel, and then decrypted on the other end. When the parties complete their transmission, this virtual private network terminates.

The two factor authentication process may be crucial in this respect, as it requires users to have both a token and a PIN that goes with it, and because it takes place at the set-up of the communication process. If one of the two is missing then it is impossible to dial-in to the corporate network. All data are encrypted from the very beginning and transmitted with protection in an end-to-end tunnel between Secure Client and the central gateway. The corporate headquarters can be accessed directly via public dial-in networks and/or the Internet.

9.3 SafeSign Identity Client and VPN

SafeSign Identity Client integrates into many VPN solutions on the market today and can provide the following benefits when used in combination with a SafeSign Identity Client supported token:

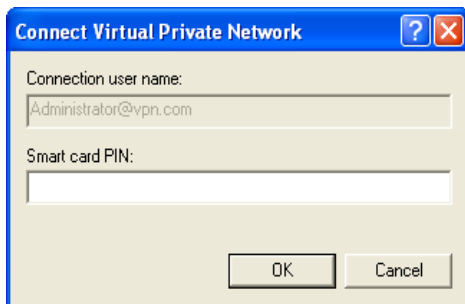
- Security of VPN access is increased through hardware-based two-factor authentication
- The use of a token reduces the risk of tampering with personal and corporate information
- A token is both a flexible and portable authentication device, ideally suited for remote access from any location.

9.4 Applications

9.4.1 Microsoft VPN

Microsoft includes extensive support for virtual private network (VPN) technologies in its Windows family of operating systems.

SafeSign Identity Client seamlessly integrates into Microsoft VPN through its Cryptographic Service Provider (CSP). In order to set up a connection, the user needs to insert his token and enter the correct PIN, after which he will be connected.



Source: [Microsoft](#)

9.4.2 Cisco VPN

The Cisco VPN Client is software that enables customers to establish secure, end-to-end encrypted tunnels to any Cisco Easy VPN server. This thin design, IP Security (IPsec) compliant implementation is available from Cisco.com for customers with SMARTnet® support, and is included free of charge with the Cisco VPN 3000 Concentrator. The client can be preconfigured for mass deployments and initial logins require very little user intervention. VPN access policies and configurations are downloaded from the central gateway and pushed to the client when a connection is established, allowing simple deployment and management.

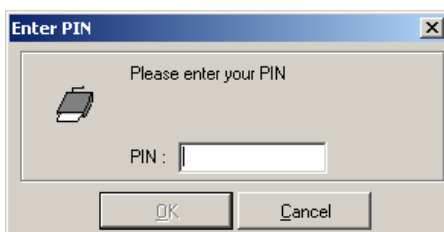
Starting with version 3.5, the VPN Client supports authentication with digital certificates through a smart card or an electronic token. As the VPN Client supports digital certificates via CAPI, it works out-of-the-box with SafeSign Identity Client.

Source: [Cisco](#)

9.4.3 NCP VPN / PKI client

With NCP Client software you extend your local area network to include remote workstations. Regardless of whether this involves stationary or mobile units. All network applications and network functionalities are available to teleworkers on their remote Client PC - 1 : 1; just like at an office workstation in the corporate headquarters.

The Secure VPN/PKI Client offers the highest possible security level currently available when accessing the central network resources as an all-in-one solution: Data encryption, VPN end-to-end tunnelling, and electronic certificates in a PKI. Strong user authentication via electronic certificates replaces the entry of user ID and password (weak authentication).



NCP VPN / PKI client supports smart cards through PKCS#11.

Source: [NCP Secure Communications](#)

9.4.4 Check Point VPN: SecureClient/ SecuRemote

VPN-1 / FireWall-1 provides the ideal platform for enterprise VPN deployments, enabling encrypted communications and guaranteeing data privacy, integrity and authenticity. In addition to site-to-site VPN capability, VPN-1 / FireWall-1 Gateway deployments provide access to remote users when used with Check Point's VPN-1 SecureClient and SecuRemote software. Digital certificate support is included for organizations with Public Key Infrastructure (PKI) deployments.

Using the SafeSign Identity Client token in Check Point VPN-1 SecuRemote / SecureClient enhances security by storing a user's digital credentials (private / public key pair and X.509 certificate) on a PIN-protected token. Furthermore, the SafeSign Identity Client software provides a flexible tool for personalizing the token and placing digital credentials on a token, taking maximum advantage of the possibilities offered by the Check Point VPN-1 / FireWall-1 solution with regard to digital certificates.

For remote access to a site secured by Check Point VPN-1 / FireWall-1, users simply have to insert the token and enter the password / PIN for the token when asked to do so. Credentials are exchanged and verified, and once the user is authenticated, he will be allowed to access the resources available to him.

Check Point SecuRemote/SecureClient supports tokens through CAPI.

Source: <http://www.opsec.com/solutions/partners/aet.html>



9.4.5 SafeNet SoftRemote

SafeNet SoftRemote is a remote access and end-point security product that secures communications on the Internet and other public networks to create a *virtual private network* (VPN) between users. The SoftRemote VPN client secures data communications sent from a desktop or portable computer across a public or private TCP/IP network. SoftRemote protects the office computer user and the home and mobile workforce.

SoftRemote supports secure client-to-gateway or client-to-client communications. For example, employees can telecommute from their homes to the office through the Internet or other dial-in remote access devices for secure client-to-gateway communications. Organizations that require a low-cost solution for secure communications among their employees or members across a private LAN, WAN, or individual dial-up connections can use SoftRemote for secure client-to-client communications.

SafeNet SoftRemote supports the use of tokens through CAPI.

Source: [SafeNet Enterprise Solutions](#)

Glossary

Authentication	Authentication is the process used to confirm the identity of a person or to prove the integrity of specific information. Certificates, issued to entities by a Certificate Authority (CA), are used to identify the author of a message or the entity providing information. People or applications who receive a certificate can verify the identity of the certificate's owner or the validity of the certificate. This process is called authentication. The action of verifying information such as identity, ownership or authorization.
CAPI	Microsoft CAPI, or Cryptographic Application Programming Interface, is an interface to a library of functions that software developers can call upon for security and cryptography services within Microsoft Windows platform products. In the case of smart card interfaces, Microsoft CAPI allows cryptographic information stored on smart cards to be used by Microsoft applications.
Certificate	A digital document (i.e. a formatted file) that binds a public key to a person, application or service. Certificates are used to verify the identity of an individual, organisation, or Web server. Three major kinds of certificates are used in a PKI: CA certificates, server certificates (also referred to as SSL certificates), and end-entity certificates.
Certificate Authority	The authority (trusted party) that issues and manages certificates within a PKI. Often, this refers to the computer that signs certificates.
Certificate fingerprint	A digest of a certificate. A certificate digest is a cryptographic hash (or „fingerprint“) of the certificate.
Client	A Client is the user (man or machine) who uses a service offered by a server or service provider e.g. a web browser is a client of a web server.
Digital Signature	Encrypting a message (or message digest) with the private key and attaching this value (checksum or hash) to a message is called 'signing', adding a digital signature. The computed value is unique for a message. This means that if the message is changed in any way, the same computation will always result in a different value. This enables user identity verification and the integrity of data (= authentication). A digital signature functions for electronic documents as a handwritten signature for printed documents.
Encryption	The scrambling of a message or data to prevent unauthorised access by making it unreadable (for unauthorised persons).
Entity	A person, organization, or device (such as a router). In a PKI, an entity may be thought of as anything you can issue a certificate to.
Hash algorithms	Message Digest 5 (MD5) and Secure Hash Algorithm (SHA-1) are examples of hash algorithms. Hash algorithms are used to reduce a piece of data (such as a file or message) to a fixed-length value. This value is called a message digest or fingerprint.
Key	A key is a series of numbers (bits) in electronic format (a mathematical algorithm / formula and a value) used to perform cryptographic functions on electronic data.
Library	A collection of computer routines.
Non-repudiation	The author of a message cannot deny having created that message at a later date (i.e. repudiation cannot occur). Digital signatures help establish the non-repudiation of transactions.
PC/SC reader	PC/SC is the architecture used by Microsoft to make it possible to use smart card readers from any vendor (that supports PC/SC). A smart card reader that supports PC/SC is called a PC/SC compliant reader (in short PC/SC reader).
PIN	Personal Identification Number
PKCS#11	The cryptographic token interface standard. This defines a technology independent programming interface for cryptographic devices such as smartcards. The standard is available at: http://www.rsa.com/rsalabs/node.asp?id=2133 This standard specifies an API to devices that hold cryptographic information and perform cryptographic functions. It's called Cryptoki and it follows a simple object-based approach, addressing the goals of technology independence (any kind of device) and resource sharing (multiple applications accessing multiple devices) with a common, logical view of the device called a cryptographic token.
PKCS#12	The personal information exchange syntax standard. This describes a portable format for storage and transportation of user private keys, certificates etc. The standard is available at: http://www.rsa.com/rsalabs/node.asp?id=2138

PKCS#15	The cryptographic token information format standard. This describes a standard for the format of cryptographic credentials stored on cryptographic tokens. PKCS #15 is intended to standardize the use of cryptographic tokens to identify themselves to multiple, standards-aware applications — regardless of the application's cryptographic token interface provider. The PKCS #15 standard defines the PKCS #15 Cryptographic Token Information Format. The format specifies how keys, certificates and other application-specific data may be stored on an ISO/IEC 7816 compliant smart card. The standard is available at: http://www.rsa.com/rsalabs/node.asp?id=2141
PKI	A Public Key Infrastructure (PKI) is a set of protocols, standards, and services that support interoperable applications of public key cryptography. These elements are designed to protect the integrity and privacy of sensitive data and, more importantly, to provide non-repudiation of important online transactions.
Private key	The private part of a key pair. Private keys must be securely stored to prevent unauthorised access and accidental deletion. In general, information encrypted with the private key can only be decrypted with the corresponding public key.
Public key	The public and widely distributed part of a key pair. For example, a certificate contains information about the certificate subject, the certificate's signer, and a public key value. In general, information encrypted with the public key can only be decrypted with the corresponding private key.
Public Key Cryptography	The system that uses a public-private key mechanism. In this mechanism, a combination of two keys is used. One key is kept secret (the private key) while the other key can be made widely available (the public key). Everything encrypted with the public key can only be decrypted with the private key and vice versa.
PUK	PIN Unlocking Key
Smart card	An electronic device the size of a conventional credit card, containing a microprocessor chip to store and process electronic data and applications. The major advantage of a smart card is security (the keys and certificates are stored on the card, not on the computer's hard disk and cannot be copied). Other advantages include: Password/PIN protection, portability and ease of use.
SSL	Secure Sockets Layer (SSL) is a protocol layer created by Netscape to manage the security of message transmissions on a network. Security is achieved via encryption. The "sockets" part of the term refers to the sockets method of passing data back and forth between client and server programs in a network or between program layers in the same computer.
Token	The logical view of a cryptographic device defined by Cryptoki.