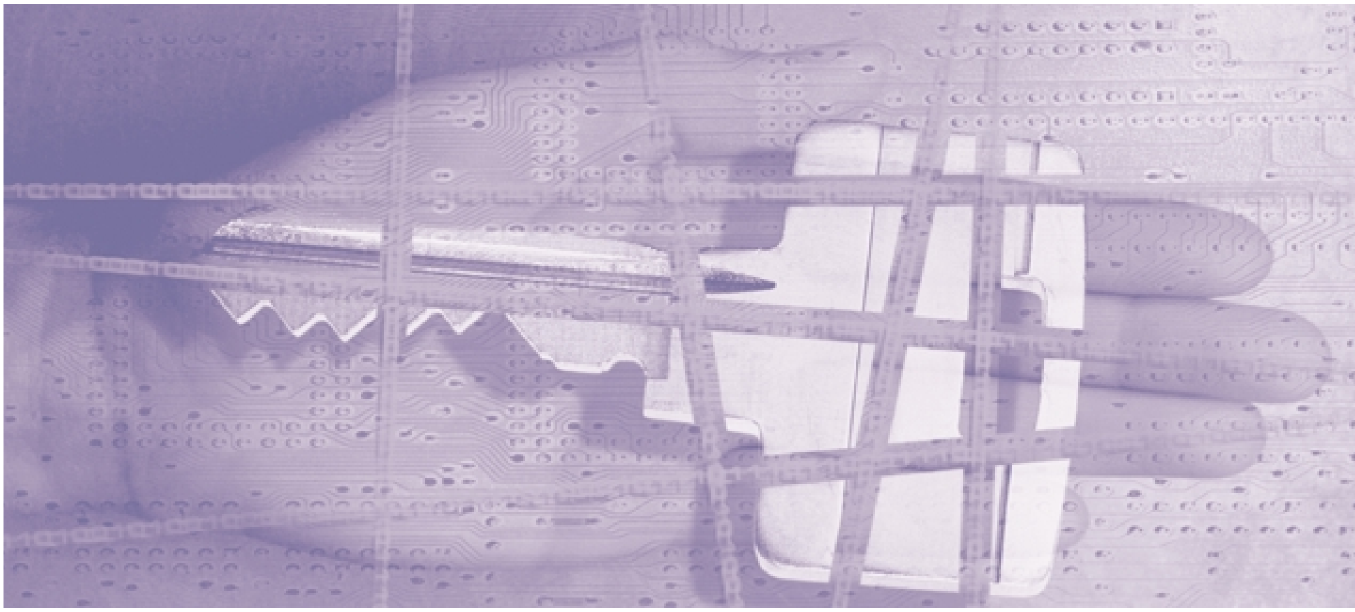

nCIPHER SECURITY WORLD WHITE PAPER

Introduction

As public key cryptography has become the basis of computer security, the weak point in enterprise security has shifted from the data itself to the keys which protect them. Data is only as secure as these keys. To address this issue, products have been developed to store keys securely within protected and trusted hardware modules.

Hardware security modules (HSMs) provide for the security of keys but, in themselves, do not completely address the issue of how keys are created, used, managed, stored and destroyed. This process of controlling key lifecycles, known as 'key management', requires software that interfaces between the HSM and the external world. In addition, the process of mediation between the physical environment of the HSM, the logical environment of the computer system whose keys the HSM protects, and the customer organisation itself requires careful design. This is the need filled by nCipher's Security World technology.

nCipher's Security World is a framework which maps security policies on to a flexible hardware-based security infrastructure. It provides for the total lifecycle management of security-critical encryption keys.

The nCipher Security World approach

The nCipher Security World approach provides many benefits for security architects planning and managing secure systems. The combination of physical and logical security enables architects to take a system-wide view of security across the enterprise and beyond. With extensions to the nCipher Security World, such as the nCipher Secure Execution Engine™ (SEE), security can be extended to computer code, and to the edge of the network and beyond. The deployment of strong security is not constrained by physical limitations about where HSMs can be located or how they are managed.

The principal respects in which nCipher's Security World offers clear benefits to security architects and system managers are as follows:

- **Secure key lifecycle management:** the nCipher approach places the emphasis on the secure management of keys and their lifecycles, enhancing usability for security managers. This includes the creation and management of keys and also back-up and key recovery
- **Multi-factor protection:** by combining physical and logical security approaches, greater security is created
- **Secure key storage:** strong encryption and secret sharing ensure keys have the highest level of protection when stored. Because keys are stored as encrypted and protected files ('key blobs') outside the physical confines of the HSM, there is virtually unlimited key storage available
- **Controlled access to keys:** nCipher's unique Access Control Lists deliver very fine-grained control over who can do what and when. Access can be further controlled by requiring a specified number of key fragments, stored on tokens, to be presented at the same time
- **Easy-to-use, familiar tokens:** Administrative and Operational personnel are issued with smart card tokens which require matching passwords for activation
- **Non-hierarchical key management:** clear separation of administration and operational management functions means that there are no 'super-users' with excessive access rights
- **Module key uniqueness:** nCipher's hardware architecture provides a true hardware random number generator. This is used at module initialisation to create a truly random and externally unknowable module key
- **Broad system scalability:** additional modules can be added to a network and used together with other nCipher HSMs By configuring each module within a Security World to use the same module key, HSMs can be used and managed together across networks to provide centralised and consistent security management
- **Logical extension with the Secure Execution Engine™:** certain nShield™ modules can be used in conjunction with the nCipher SEE technology to develop advanced custom security infrastructures. SEE enables developers to create an environment where the same protection afforded to keys can be extended to trusted application program code, to create Trusted Agents™

The importance of secure key management

Protecting the keys at the heart of a security infrastructure is of vital importance. As accepted standards of security rise, and it becomes a given that keys must enjoy additional protection, the secure lifecycle management of the keys and their ongoing protection becomes a leading issue for security management.

An HSM is one obvious way to do this: by placing a physical barrier between the keys and the outside world, it ensures that the keys cannot be stolen, destroyed or tampered with.

While physical security is of great importance, it is not sufficient alone to provide for the security of valuable keys. Because cryptographic keys operate in a complex software environment, the security infrastructure needs to consider both logical and physical attacks on the stored keys, and the prevention of those attacks.

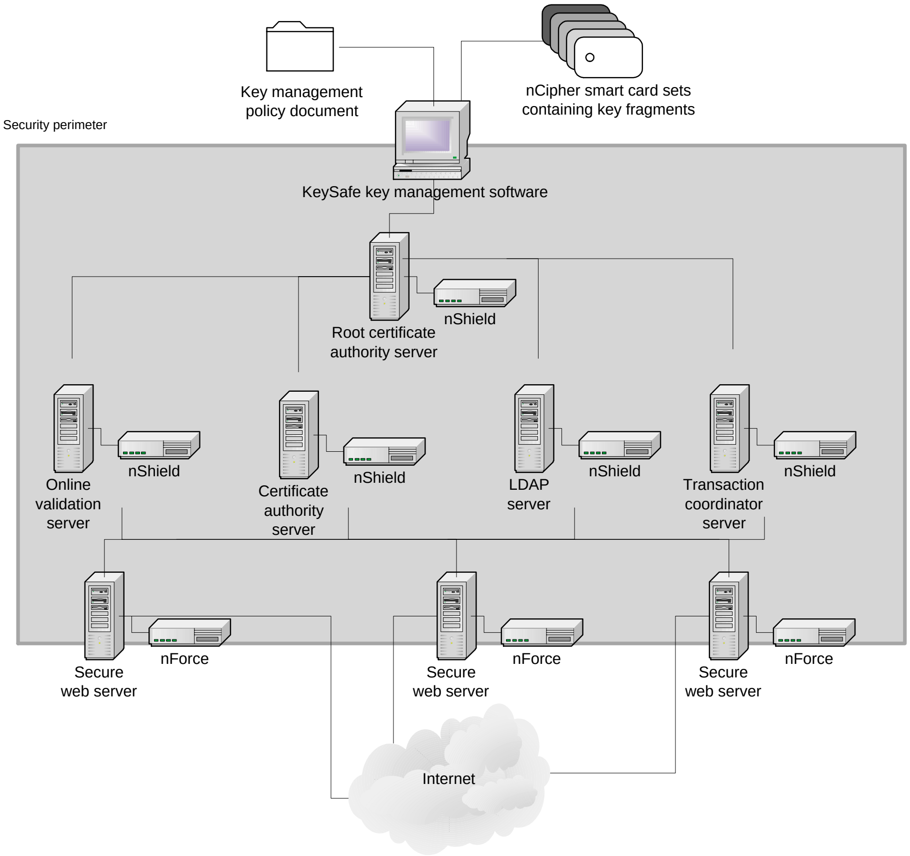

Figure 1: the nCipher Security World

In the nCipher Security World these needs are addressed by nCipher's advanced library of key management techniques, which provide an interface to the application programming, key generation and management functions built into the HSM. Key management features supported by the nCipher Security World include:

- Secure generation of new keys
- Setting the capabilities and security limits of new keys
- Implementing procedures for key backup and, if required, key recovery
- Preparing keys for storage. This is described below in the section Exporting keys
- Revoking keys before the end of their planned lifetime
- Retiring keys at the end of their planned lifetime

Human access to the capabilities provided by the nCipher Security World is also provided through the graphical user interface of nCipher's management software, KeySafe™.

Multi-factor protection - logical and physical security

nCipher's nShield HSMs combine logical and physical security techniques. Keys are only ever stored outside the physical protection of the HSM in an ultra-secure format, the encrypted key file or 'key blob'. Key blobs can only be decrypted inside the HSM when certain conditions (as specified by the operator) are met. For example, a key may be stored in several fragments, of which a pre-specified number must be presented, along with appropriate signed credentials, to reconstitute the key inside the HSM, thus ensuring that it never appears in the clear.

However, the emphasis on logical security is not at the expense of physical security. nShield is validated at FIPS 140-1 level 3, with physical security features to achieve this stringent validation including epoxy-coated hardware and tamper-evident seals.

Employing physical security has several benefits; it is very clear when the unit has been attacked, for example. However, physical security should only form part of a comprehensive approach to security to help mitigate broader risks; for example, physical attacks on an HSM could result in the destruction of the key. If the key was not backed up (as would be the case if the key could not be exported in a secure format), this would create significant problems as the key would effectively be lost.

Secure key storage

The secure storage of keys is the foundation of a robust security architecture. This security goes far beyond protecting keys when they are being used inside the HSM. The security perimeter must extend widely enough to deliver a secure framework of key management procedures, which includes backup and key recovery options. It must also allow for the secure storage of large quantities of keys in order to meet real-world storage capacity requirements. If security is limited by the storage capacity of an individual HSM, scalability becomes a significant problem.

Potential weaknesses of key storage within the HSM

It has traditionally been thought that the most secure way to store a key was to protect it exclusively within the physical confines of a hardware security module. However, this approach has several limitations:

- The safety of the key is immutably bound to the safety of the HSM. If the HSM is attacked, the key is destroyed
- The ability to implement secure key recovery and backup policies is mutually contradictory with a security approach where keys cannot be exported. In practice, many units whose keys purport never to leave the module in fact rely on export facilities, or require access to the manufacturer's shared secrets to achieve recovery and back-up
- The number of keys which can be created, used and stored is restricted by the capacity of storage built into the HSM unit
- Establishing the HSM's own original key (in nCipher terminology, the 'module key') is also a problem with physically-restricted architectures. In practice, root keys are often pre-installed and thus known to the HSM manufacturer. This results in a chain of trust that is not entirely under the control of the HSM administrator, or may even result in multiple separate installations that share the same root key or back-up key. Administrators cannot be sure that external parties could not gain access to these keys if the vendor's trust is breached.

Thus, it is evident that whilst the idea of keys 'never leaving the module' has a basic appeal, in fact this policy is not sustainable for any practical operation of the HSM, where key back-up and key recovery must be properly available and controlled. An architecture which does not consider this fact in its basic design is more likely to suffer from security flaws which can be exploited by attackers.

Key backup and key recovery

Because the nCipher Security World stores keys outside the HSM in a considered and secure way, key backup and key recovery procedures are easier to implement in a consistently secure manner. Keys are exported using documented and secure procedures, not by 'back doors'. Because the keys are not physically tied to the HSM, the loss or compromise of the HSM does not automatically mean that keys are lost.

This approach means that nCipher's key management architecture is ideal for situations where well-documented key recovery systems are required. In any situation, the loss of the module does not necessarily mean the loss of the keys. By creating a network infrastructure where multiple HSM devices at separate locations share the same module key, keys can be loaded into another server in the event of the loss of the primary server. This architectural flexibility is a boon to security architects who can design network infrastructures to cope with multiple risk scenarios, whilst preserving resilience and scalability.

Protecting stored keys

How can keys be properly protected in storage when they can also be removed from the HSM? The nCipher approach employs various cryptographic techniques to provide security for exported keys. These include:

- Strong encryption: the key data itself is stored in a triple-DES encrypted format
- Fragmentation of keys: secret-sharing enables key fragments to be stored separately on tokens so that 'k of n' key fragments are required in order to load or reconstitute the key being protected
- Access Control Lists (ACLs): each key has its own list of operations which can be performed and keys which are entitled to request that operation, enabling a multi-layered and finely controlled security infrastructure to be created

This structure ensures that when keys are not physically protected within the module (as they usually are when being created, used and managed), they are instead logically protected. Keys can only be exposed in the clear within the secure confines of the HSM. Through the use of strong, multi-factor encryption and the addition of schemes such as secret-sharing across multiple password-protected operator tokens, the security of encrypted keys is exceptionally high.

When a key is transported outside the HSM, the confidentiality and integrity of the key data and its ACL are protected by encryption, hashing and signing. A key object encrypted in this way is known in the nCipher Security World as a "key blob"; the detailed process of creating key blobs is described below in the section *Exporting keys*.

The power of the Access Control List

The nCipher Security World stores an associated *Access Control List* (ACL) with every key blob. The key payload and the ACL are encrypted together and signed together by a module key (in this context usually referred to as the "wrapper key"). The ACL describes the operations that can be performed with, or upon, the key. Access Control Lists are a fundamental tool in nCipher's Security World, allowing very fine-grained control over keys that can be closely matched to the individual requirements of a given security policy.

As well as determining whether or not an operation can be carried out at all, various threshold techniques are also available to increase flexibility. For example, each operation permitted by the ACL can also specify how many times that particular operation can be performed, enabling different levels of access to be provided to different classes of user or for different types of key operation. The authorisation limit contained in the ACL can be:

- A hard limit: once the limit has been used up, that operation can never again be performed with that key

- A per-authorisation limit: the user just has to reload the key to restart the count
- A time limit, setting a time period after which the key will expire

Different limits can be placed on each operation, or group of operations. The ACL also controls whether or not the user can alter the ACL itself. In general, the ACL is configured to enable administrators to remove permissions, but not to add permissions.

Exporting keys

As noted earlier, layered cryptographic techniques are used within the nCipher Security World to permit keys to be safely exported from the physical protection of the HSM. The following sequence illustrates the steps taken to establish a key in an encrypted key file, known as a 'key blob' (see also Figure 2, below):

1. The target key is encrypted using strong (Triple-DES) encryption. Its Access Control List is also, separately, encrypted.
2. The key and ACL are encrypted together and the result is signed with a wrapper key (usually the module key), to form the key blob. A Message Authentication Code (MAC) is stored with the key blob, ensuring that tampering is detectable.
3. The wrapper key in turn is associated with another Access Control List, which determines who can access it.
4. If required, key fragments can each be wrapped with their own access control mechanisms.
5. Encrypted and protected key blobs can now be exported and stored either on smart cards or server storage such as hard discs.
6. If required, key fragments can be stored separately so that multiple tokens are required to access a valuable key. Operational use of such keys would require a chosen number of smartcards (k) out of a total set (n) to be presented to the module

These processes are in accordance with the procedures laid down by FIPS 140-1 and also by the ISO 15782 international standard, each governing procedures for the safe handling of private signing keys.

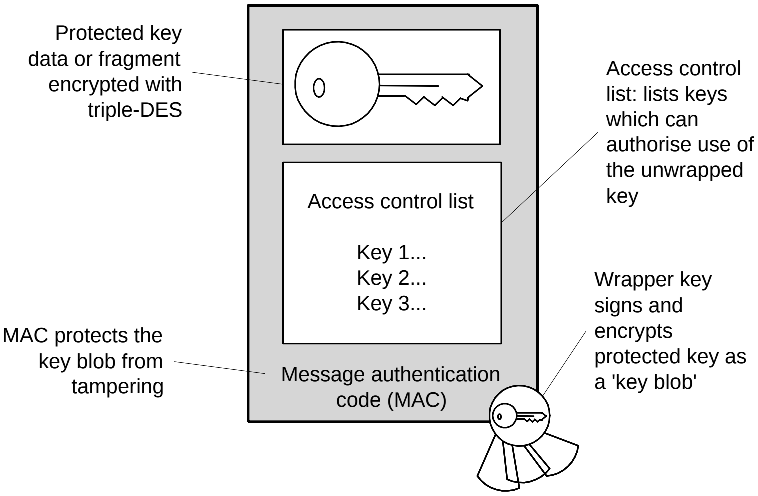
Protected key data or fragment encrypted with triple-DES
Access control list
Key 1...
Key 2...
Key 3...
Access control list: lists keys which can authorise use of the unwrapped key
MAC protects the key blob from tampering
Message authentication code (MAC)
Wrapper key signs and encrypts protected key as a 'key blob'

Figure 2: Components of a key blob

Access to key blobs is physically controlled:

- By the use of smart card *tokens*, which must be presented in order to load the key blob into the HSM and unwrap it for decryption
- By using the module key: any key data stored on physical tokens is encrypted with the module key, so that theft of the tokens and knowledge of the key blob is not by itself sufficient to recover the original key object. Module keys are held securely within the HSM. This 'personalises' a set of tokens to work with a particular HSM, or group of HSMs
- By using a pass phrase which the operator must supply in addition to the token. Instead of just encrypting the data on the token with the module key, the HSM can combine the module key with either a phase supplied by the user or the hash of it

The reverse procedure from storing keys is used to gain access to a key, as shown in Figure 3 below.

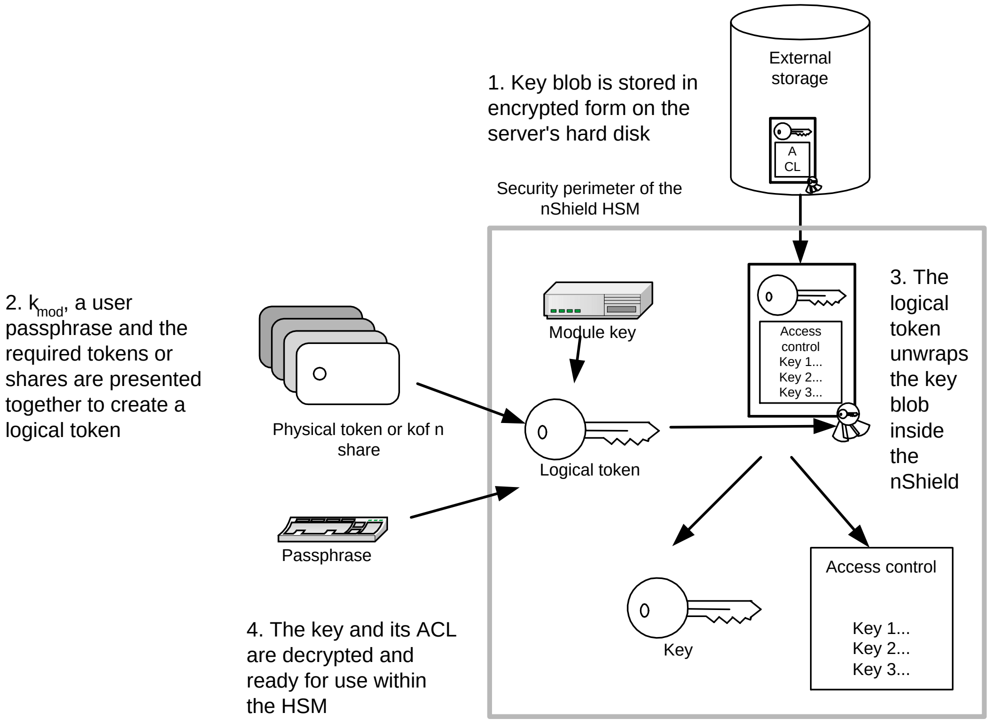
1. Key blob is stored in encrypted form on the server's hard disk
External storage
A CL
Security perimeter of the nShield HSM
2. k_{mod}, a user passphrase and the required tokens or shares are presented together to create a logical token
Module key
3. The logical token unwraps the key blob inside the nShield
Access control
Key 1...
Key 2...
Key 3...
Physical token or kof n share
Logical token
Passphrase
4. The key and its ACL are decrypted and ready for use within the HSM
Key
Access control
Key 1...
Key 2...
Key 3...

Figure 3: Loading a key from storage

Non-hierarchical key management

A fundamental benefit of the nCipher architecture is that there is a clear separation between administrative and operational functions. There is no concept of a 'super-user', so no single user can have the powers to override security functions. Access to administrative functions provides no access to operational functions; the two card sets (containing the keys which authorise each set of actions) and policy statements can be kept entirely separate.

In a typical installation, there will be at least two sets of cards (containing tokens to load keys): an Administrator card set and one or more Operator card sets.

- The Administrator card set will belong to the Security Officer and is used to enable administrative functions to be performed. Examples are the key recovery and data recovery functions
- The Operator card set(s) will be used by operational staff to establish their rights to perform application functions, such as signing keys and generating certificates using keys protected by the nCipher Security World. There may be more than one Operator card set, or set of Operator keys, within an nCipher Security World

The ability to establish multiple sets of Operator cards is of particular value in creating easily-manageable nCipher Security Worlds. For example, one card set can provide very limited access, perhaps to a routine administrative aspect of day-to-day certificate issuance, while further sets which provide access to higher-level functions may have stronger security rights associated with them, perhaps therefore requiring the use of multi-party secret-sharing to perform some functions.

Multiple card sets can also be managed separately according to the security requirements of the organisation. For example, card sets which are widely distributed to many staff can be replaced periodically by revoking the keys they contain and issuing new ones; other card sets can contain keys with a longer life, replaced less frequently. The key lifecycle required can be matched to the security needs of the functions to which the card sets permit access, according to system policy.

Administrator and Operator keys and card sets can only be used by units sharing the same module key. Operator card sets likewise only provide access to the functions permitted by the Access Control Lists for the keys they contain. The process of exporting module keys from one module to another is managed securely using nCipher's KeySafe software. Each nCipher installation therefore forms a completely separate nCipher Security World and is independent of the actual number of physical HSM devices deployed. Each nCipher Security World, on the other hand, can be securely extended so that it can be centrally managed by the same Security Officer, and can share keys belonging to the same trust chain. Keys and card sets are tied to their originating module keys, so will cease to be valid if the nCipher Security World is re-initialised.

The two card sets, Administrator and Operator, are not interchangeable and access to one provides no access to the other.

Initialisation key uniqueness

When an HSM is first installed and initialised, it generates its own truly random initialisation key. This key, the module key (k_{mod}), is unique to that particular HSM. The module key is not known to nCipher or anyone else and cannot be exported because no access to it is provided through the programming interface (API). It is stored in the HSM's non-volatile RAM and remains valid until the module is re-initialised. This approach improves security by making the HSM more self-reliant.

Before the nShield can be used for key management, it must be initialised by setting a Security Officer key which will be used to check the signature on all commands that must be authorised by the Security Officer. This key cannot be changed without reinitialising the module, a process which clears all the non-volatile storage in the unit and destroys all the module keys. Without a module key, tokens cannot be loaded, and without a token key blobs cannot be loaded; therefore reinitialising the unit invalidates all the keys it has issued since it was last initialised.

The Security Officer managing the nCipher Security World is identified to the module by the public half of the Security Officer key pair. This key pair is established during the initialisation process and a hash of the public key part is stored alongside the module keys. The private Security Officer key likewise is stored inside the module, but use of it is controlled by means of the Administrator Card Set.

Access to initialisation procedures is controlled both logically (by requiring digital credentials) and physically (by a mode switch). This ensures that the unit cannot be reinitialised without the appropriate rights. Module keys can only be added or removed with the permission of the Security Officer, thus preserving the integrity of this security model.

Some HSM architectures require that the original module key be installed by the HSM vendor, or created in a way which is knowable by them. The chain of trust, rather than ending at the HSM or the security infrastructure it supports, goes outside the perimeter of the organisation and back to the vendor, who may already know the key or have a backdoor to extract the key in emergency situations. Clearly avoiding this kind of security model is highly desirable.

Scalability

Another particular advantage of the nCipher Security World approach is that it is truly scalable. It is possible to share module keys across a series of HSMs, so that a large-scale installation, possibly geographically diverse, can be assembled and consistently managed. The use of multiple servers is commonplace both within enterprises and in hosting environments. By attaching an HSM to each server, and making each device part of the same nCipher Security World, the modules can be managed together with no reduction in security. nCipher's architecture can encompass the following situations, amongst others:

- Multiple HSMs are required on a single server, to handle a high volume of secure transaction requests (for example, an online certificate validation server)
- Multiple servers on the same local area network need to be managed together, each having one or more HSMs directly attached to it
- Multiple servers separated across a wide area network, or over the Internet, each having one or more HSMs directly attached to it

HSMs can be chained together by using the same nCipher Security World module key for each module, enabling them to share keys and operate together. The operation of different HSMs can be restricted using Operator card sets to determine which functions can be carried out with which units and at which sites.

This approach also has administrative benefits, in that all units sharing the same module key can be managed together. So, for example, a head office can manage the security of branch office systems remotely using key management software to administer the keys involved (Fig.4).

There is no need to determine the final size of the nCipher Security World at the outset. Additional HSM devices can be added to a server at will, and HSMs can also be attached to additional servers. The ability to store nearly unlimited numbers of keys also means that the number of keys within the system does not become a limiting factor on future growth.

Extending the nCipher Security World with SEE

While the nCipher Security World described in this white paper is principally concerned with the protection of digital keys, of course public key technology is used for a far wider range of security needs. Keys and certificates can be used to sign and protect a wide range of data for different purposes.

Currently, most security architectures are concerned with the protection of keys and data. However, nCipher's Secure Execution Engine takes the security concept a stage further and uses the nCipher Security World's system of signed access control lists, delegated rights and strictly-enforced policies to control the execution of application code. SEE technology enables security managers to create a set of policies and rights which can be delegated to authenticated program code, enabling the code to perform security functions without operator supervision. This technology and its features are described in more detail in the nCipher Secure Execution Engine white paper.

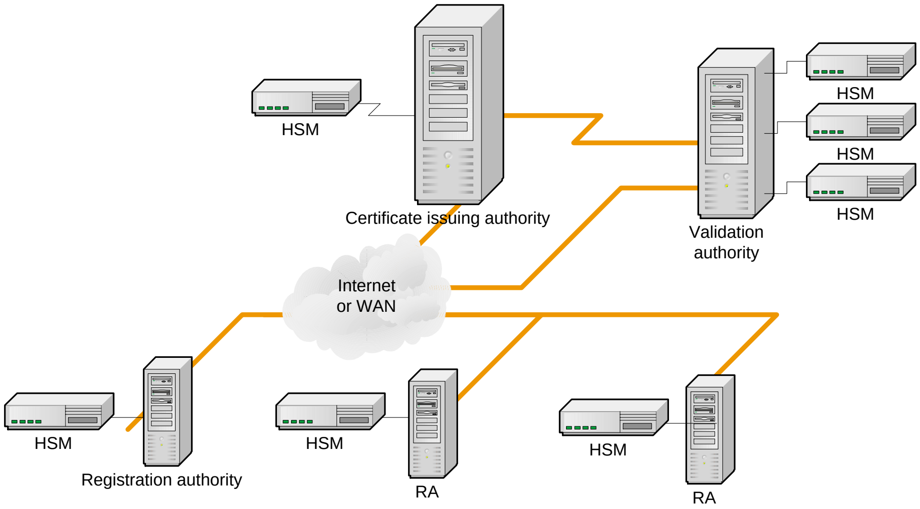

Figure 4: A multi site system protected by an nCipher Security World

Conclusion

There are several approaches to building security architectures. Different approaches combine physical and logical security in different ways, and draw the boundaries of security areas in different places.

nCipher's Security World approach maximises the flexibility and scalability of the security system by using a thoroughly secure key management architecture.

The nCipher approach of storing keys within strongly encrypted key blobs and enabling them to be transported outside the HSM provides enhanced security, distributes physical risk, and enables the provision of a more logical structure for key management, backup and recovery.

Enterprises are exploiting the cost-reducing and security-enhancing benefits of digital certificates and public-and private-key cryptography ever more widely. As they do so, the value of the expandability, flexibility and scalability of the nCipher Security World is becoming increasingly evident.

Abbreviations and glossary

ACL	Access control list
CA	Certification authority
DES	Data Encryption Standard - An industry standard symmetric cryptographic algorithm used to encrypt key material in the nCipher Security World
HSM	Hardware security module
Key blob	Encrypted and protected data consisting of a key and its Access Control List, signed by a module key and only readable by units which have the same module key
LDAP	Lightweight Directory Application Protocol - an industry standard for directory services applications, for example those used to deliver public keys on demand
Module key	A key generated by each nCipher HSM on initialisation, used to wrap up key blobs and key fragments for tokens. Module keys can be shared across several HSMs to create a larger nCipher Security World
NVRAM	Non-volatile RAM
PKI	Public Key Infrastructure
RA	Registration authority
Security Officer	The system administrator who manages the administrator card set and the management of the Security World
SEE	Secure Execution Engine
Triple DES	Highly secure DES (q.v.) variant in which the message is encrypted three times

NCIPHER™

CORPORATE HEADQUARTERS

Europe, Middle East & Africa
nCipher Corporation Ltd.
Jupiter House
Station Rd.
Cambridge, CBI 2JD
United Kingdom
Tel: +44 (0) 1223 723600
Fax:+44 (0) 1223 723601
E-mail: sales@ncipher.com

The Americas & Asia Pacific
nCipher, Inc.
500 Unicorn Park Drive
Woburn, MA 01801
United States
Telephone: 1 800 NCIPHER (1 800 624 7437)
or + 1 781 994 4000
Fax: +1 781 994 4001
E-mail: ussales@ncipher.com

Every effort has been made to ensure the information included in this paper is true and correct at the time of going to press. However, the products described herein are subject to continuous development and improvement, and the right is reserved to change their specifications at any time.