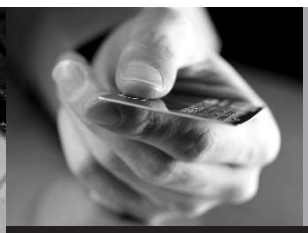
WHITE PAPER



NCIPHER netHSM

TECHNICAL ARCHITECTURE

nCIPHER™

I. Introduction

Hardware Security Modules (HSMs) are used for the protection of cryptographic key material. Typically HSMs have been directly connected to the servers requiring use of the key material. The demand for better manageability and improved return on investment has seen the emergence of 'shareable' HSM devices.

These network connected HSMs perform cryptographic functions on behalf of one or more remote servers over a network. As a potential single point of failure and performance bottleneck, shareable HSMs are required to be resilient, high capacity and above all secure. This document describes the architecture adopted by nCipher for its range of network-connected devices.

Additional information relating to best practice policy issues that relate to the use of HSMs and key management can be found in KPMG's 2002 white paper Key Management Policy and Practice Framework. (For the full text of this paper please visit http://www.ncipher.com/resources/downloads/files/white_papers/KPMG_wp.pdf)

About nCipher

nCipher develops Internet security products that optimize the use of cryptography to protect critical points of risk - Web servers, networks, applications, payment systems and databases - across the enterprise.

nCipher's FIPS 140-2 validated hardware security modules provide a secure, trusted environment to isolate keys and sensitive application software from attack. All of nCipher's dedicated and shared HSMs use a common key management framework, nCipher's Security World, ensuring compatibility across any deployment. This flexibility allows an organization to add, reconfigure or reallocate hardware to extend security to meet new business needs, maximizing the return on investment.

With netHSM, nCipher has designed both custom hardware and software to deliver an uncompromising, high security product.

2. AUDIENCE

This document is intended for security architects, security consultants or others interested in the design and implementation of nCipher's network connected HSM products.

3. TERMINOLOGY

Throughout this document, the server machine(s) that connect to a netHSM for cryptographic service are referred to as clients of the netHSM.

The term enrolment is used to describe the process of setting up a client to communicate with a netHSM; it always involves the registration of the netHSM to the client, it can involve the registration of the client to the netHSM.

4. PRODUCT PHILOSOPHY

There are different routes to create a network connected HSM. A product formed using 'off the shelf' hardware and software components is likely to result in compromised security, integrity and usability. With netHSM, nCipher has designed both custom hardware and software to deliver an uncompromising, high security product.

5. ARCHITECTURE OVERVIEW

5.1 Hardware architecture

The netHSM is a 1U high, 19" rack-mounted, secure network appliance comprising the following subsystems:

- An industrial Intel-based controller motherboard with integral solid-state disc
- A PCI bus
- FIPS 140-2 level 3 validated, tamper-resistant Hardware Security Module (HSM)
- An I/O board that controls all external interfaces other than the dual Ethernet ports
- A power supply
- A tamper-evident chassis with integral fans
- Integrated and secure user interface
- Smart card reader

The various interconnections between these components are shown below in Figure 1

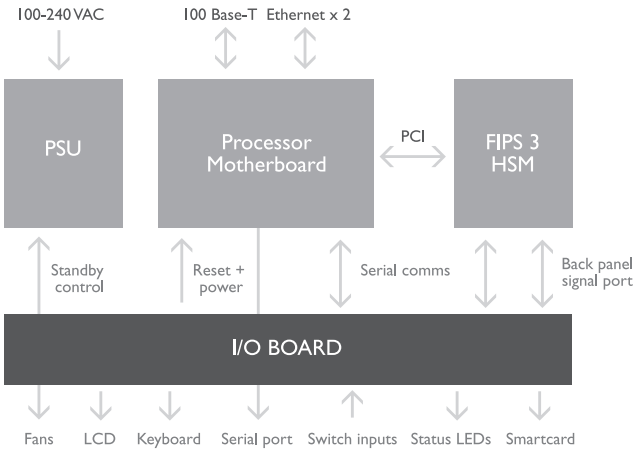


Figure 1 - Hardware architecture

5.2 Software architecture

Figure 2 below shows an outline view of the major software components at both the netHSM and each remote server. The remote client's 'Hardserver' mediates communication between server-based client application(s) and netHSM(s)

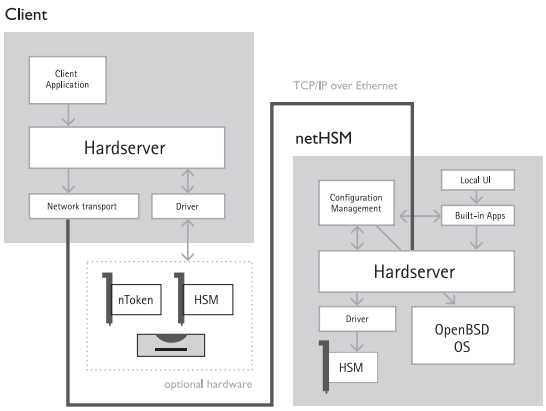


Figure 2 - Software architecture

6. SECURITY BOUNDARIES

The netHSM architecture provides a layered approach to security. It is possible to identify four discrete security boundaries in a typical netHSM deployment, these are:

FIPS boundary
the core security boundary for all cryptographic operations

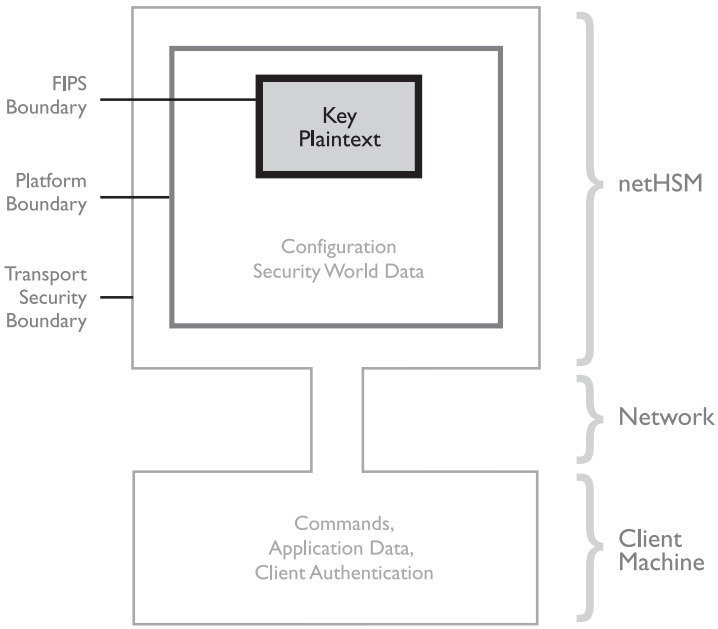
Platform boundary
the remaining systems components and physical chassis of the netHSM appliance

Transport boundary
the network connections linking remote servers to the netHSM

Client boundary
security features relating to the netHSM but hosted on the remote server

These boundaries are illustrated in Figure 3.

Figure 3 - Security boundaries



6.1 The FIPS security boundary
The innermost security zone is the FIPS (Federal Information Processing Standard 140-2, see Glossary for more details) security boundary. All requests for cryptographic processing by the remote servers along with sensitive internal netHSM control activities are handled within the FIPS security boundary; all plaintext key data and their associated Access Control Lists are only exposed within this boundary.

The netHSM can optionally contain a Secure Execution Engine (SEE) which can execute signed custom software. The SEE machine and its associated data and control logic execute within this hardware boundary.

The FIPS boundary is physically protected from tampering by embedding the circuitry in hardened epoxy resin, a process known as potting.

The value of an HSM is often placed on its ability to store keys in a physically protected way. However in real deployments at least equal benefit is derived from the techniques used to control access to those keys both from an administrative perspective and an operational perspective.

6.2 The platform security boundary
The steel outer case of the NetHSM chassis marks the boundary of the next security zone: the Platform Security Boundary. This boundary protects the embedded Intel microprocessor, its operating system and the user interface. The boundary is physically protected with tamper evident seals and cryptographically protected during upgrades using digital signature techniques. The netHSM keeps Security World data within the platform security boundary allowing control and audit via the netHSMs integral secure user interface.

6.3 Transport security
To prevent eavesdropping, it is vital to secure the communication between the netHSM and 'client' machines (the servers that require cryptographic processing to be performed inside hardware). It is also important that the netHSM and the client machines are mutually authenticated to prevent unauthorized use of keys. A security protocol known as Impath is used to provide secure, encrypted communication of all data transferred over the network. Impath is described in greater detail in section 9 of this document.

6.4 Client authentication
With any network connected HSM, the strength of 'client' authentication is critical; weak authentication allows the possibility of rogue servers or applications being allowed illegitimate access to key use. The netHSM security architecture supports different strengths of authentication, each appropriate for a different use.

7. FIPS BOUNDARY

FIPS 140-2 is the latest Federal Information Processing Standard that outlines security requirements for cryptographic modules. FIPS 140 was developed as part of the Cryptographic Module Validation Program (CMVP), a joint effort by NIST and the Communications Security Establishment of Canada. Initially developed for federal agencies using cryptographic-based security systems, the original FIPS 140-1 standard has become a widely used benchmark throughout the business world. To receive FIPS validation, cryptographic modules are tested by independent, accredited testing laboratories and receive a security level rating from 1 to 4. The range in security levels supports organizations' various security needs, based on the sensitivity of their data and on their computing environment.

The netHSM's FIPS boundary is fully validated to FIPS 140-2 level 3 (see the NIST website http://csrc.nist.gov/cryptval/ for details. Relevant certificates are 295 for the 1600 module and 297 for the SEE and payShield versions).

7.1 Key storage & protection
The netHSM FIPS boundary is designed to ensure that there is no single point of compromise within the key management environment. All cryptographic functions take place within the FIPS 140-2 Level 3 validated HSM. Critical private and secret cryptographic keys never appear in plain text outside the boundary of the cryptographic module. When not in use, application keys are encrypted and signed using a wrapper key before being stored on the 'client' computer's hard drive. This approach to wrapping is widely considered to be functionally unbreakable with any current or foreseeable technology. Allowing strongly encrypted application keys to be stored outside the netHSM helps ensure there is no single point of compromise in the system and that customers benefit from a scalable and flexible solution. This key management control meets FIPS 140 Level 3 requirements and complies with ISO standard 115681.

7.2 Access control
Access to security functions is managed using two-factor authentication, split responsibility and role separation. Threshold sets of smart cards, where 'k of n' cards must be presented along with individual pass phrases to enable a particular feature or to authorize a specific function. This means responsibility can be split between multiple security officers.

Clear separation of administration and operational functions through the use of distinct sets of smart cards ensures that there is no single 'super-user' with excessive access rights.

7.3 Functional separation
Every application key is bound to an individual Access Control List (ACL) to deliver fine-grained control over that key. Access to individual application keys can be further controlled through the use of Operator Card Sets (OCS). This allows different levels of security to be assigned to individual keys in direct relation to their importance.

As an alternative to OCS cards, a hardware token can be deployed at the client, allowing use of a particular key to be restricted to a specific server. Access would be granted only to a server that can strongly authenticate itself using a secret protected inside a specific pre-enrolled client hardware token. Together these controls ensure that individual keys or groups of keys can be isolated from one another through logical separation. This helps to reinforce the individual requirements of a given security policy, allowing access to individual keys only by authorized users or applications.

7.4 Compatibility
Keys can be securely shared between all nCipher's HSMs, both network-connected and dedicated devices. Security World compatibility ensures that keys made available to an existing nCipher installation of dedicated HSMs can be easily made available to a netHSM, to enhance or upgrade an existing installation.

7.5 Secure execution of code
In addition to protecting keys within this boundary, it is also possible to protect application software that executes within the physical HSM boundary. Using nCipher's SEE technology and CodeSafe toolkit, program code can be signed, optionally encrypted, and be given delegated rights to keys that are available within the boundary. SEE thus allows strongest possible protection for security-critical application logic.

SEE code is portable between nCipher's directly connected and network-connected HSMs. Correctly written code can be made available to multiple clients when deployed on netHSM platform.

8. PLATFORM BOUNDARY

The steel outer case of the NetHSM chassis marks the boundary of the Platform Security Boundary. This boundary protects the system CPU, internal operating system and the user interface. The boundary is physically protected with tamper-evident seals and logically protected using digital signature techniques.

8.1 Secure operating system

The operating system of any device that offers network connectivity is crucial in determining the resilience of the device to hostile attack. The embedded processor in the netHSM runs a minimal OpenBSD kernel; OpenBSD was chosen over other kernels as its design focuses on robustness and security rather than features.

OpenBSD is the winner of the 2003 Information Security Leadership Award for effective security testing of an operating system, http://www.sans.org/press/ISLA.php

The netHSM runs the OpenBSD kernel at security level 2, which prevents certain system-critical actions even with root privilege. It also takes a number of additional steps to defend against network-based attacks. All non-essential services such as the command line shell, telnet, FTP, NFS, mountd and Web Services are removed.

The only process allowed to connect to the external network is the nCipher Hardserver. This code is carefully checked, and provides many automatic data validation mechanisms designed to guard against buffer overruns and similar vulnerabilities.

Access to the netHSM is therefore restricted to well-defined commands accessible from the network interface or commands via the front-panel user interface. There is no shell accessible to the outside world, no user accounts to be managed or passwords to be forgotten. All access is mediated by Security World credentials i.e. the presentation of Administrator or Operator card sets (and optional pass phrases), whose secrets are kept safe within the inner FIPS security boundary.

8.2 Secure upgrade mechanisms

The security of a network-connected device can be compromised if there are weak mechanisms protecting the upgrade of internal software components. With netHSM, care has been taken to ensure that the firmware of the netHSM's internal components (the HSM, the embedded PC, and the User Interface controller) can be updated in the field, without compromise to security.

Only firmware authorised and signed by nCipher is accepted by the reprogramming mechanism. A single upgrade package contains images for each of the internal components. Firmware is always replaced as a set to avoid the possibility of version mismatch attacks. All internal code memory is erased and replaced with the new contents. The security boundaries around the internal components are maintained in the structure of the upgrade file. Separate signing keys are used for each firmware component, and each image is subject to security review by a team of specialists before its key is used.

The netHSM upgrade system uses the 'VSN' (Version Security Number) concept to prevent downgrade attacks. When a new version of firmware is issued with enhanced security properties, the VSN associated with that version is increased. The reprogramming mechanism will refuse to allow the current version of firmware to be replaced by a version with a lower VSN.

8.3 The secure user interface and management functions

The netHSM contains a front panel user interface from which Security World and device management functions can be performed. This comprises an LCD for output, a selection wheel and soft keys for input and an optional keyboard for more convenient data entry. The user interface is used for diagnostic and status information, for network and other configuration items and for Security World management.

When setting up a Security World (or, indeed setting up any other type of HSM which attaches to a host machine) trust has to be placed in the host machine to correctly convey the user's choices to the HSM and to relay important information back to the user. Until now it has been common practice to require a standalone machine with a fresh installation of software to perform setup functions. This is operationally expensive.

The netHSM's user interface is sufficiently rich to allow all Security World management functions to be performed directly (including creation and restoration of new or existing Worlds, creation of Operator Card Sets, and exercise of PIN or cardset recovery functions). As the display and input are directly coupled to the embedded secure operating system, all parameters are protected by the Platform boundary and network access to the netHSM is disabled when appropriate.

The netHSM can manage a Security World on behalf of a mixed network of netHSM clients and hosts with dedicated PCI or SCSI attached HSMs. A new World can be created and the resulting (KMdata) files distributed to each client. Alternatively an existing World can be imported and its parameters verified by comparing them to the secure user interface display. As all day-to-day administration functions can be performed at the netHSM, there is no need to insert Administrator cards into any other module; this is not only operationally convenient but helps to prevent misuse of these cards.

The request to generate application keys is usually performed on the client machine; this is done with nCipher's graphical KeySafe tool, its command-line equivalent, or directly from within an application itself. The secure display allows key generation certificates to be verified and its parameters checked in a trusted environment.

The netHSM user interface also provides support for the creation, verification and management of payShield installations. The payment keys used in the installation can be generated, imported, checked or revoked from the front panel.

For CodeSafe applications, the netHSM allows code signing keys to be created and stored internally. The hash of this key can be read out from the secure display for external use and CodeSafe SEE Machine or User Data images can be accepted from the outside for signature. The integrity of the entire code signing process can therefore be protected within the Platform boundary.

As all day-to-day administration functions can be performed at the netHSM, there is no need to insert Administrator cards into any other module; this is not only operationally convenient but helps to prevent misuse of these cards.

8.4 Security issues related to status reporting and remote configuration

Configuration and management of netHSM

nCipher recognizes the need to balance manageability and restricted access to provide a solution that satisfies the need for strong authentication of users, secure system configuration and remote manageability. The netHSM offers:

- The ability to enrol the netHSM with a client via out-of-band authentication
- Restricted access during initial configuration & during Security World creation based on the use of an Administrator Card Set (ACS) that needs to be presented for authentication
- Facilities to remotely manage client and netHSM configuration, Security World administration and system monitoring

The netHSM is designed to allow network administrators to perform many configuration and monitoring tasks remotely, without compromising the security of the unit.

All 'control' parameters (such as the IP address, identification of client machines, etc.) are maintained in a configuration file internal to the netHSM. This configuration file can be edited from the front panel user interface and can be backed up and restored to or from a remote machine. If configured to be allowed, a designated remote machine can 'push' a new configuration file to the netHSM, which will update the system when it is received. Alternatively a configuration file can be sent from any remote machine at which appropriate credentials (smartcards) are presented to a directly-connected HSM.

Some security-critical operations may only be performed with direct access to the netHSM – e.g. initialising the device into a Security World. The security of the system benefits from the ACS cards only ever being presented to a trusted device.

Configuration and management of Client

The pre-configuration of the Client involves configuration of the local 'Hardserver'. The local Hardserver must be configured with the appropriate IP address(es) of netHSM(s) that are available to it; the netHSM needs to be configured with the IP address of the Client.

nCipher is committed to delivering high security and integrity products. Independent testing and validation is central to this commitment.

The process of configuring the netHSM and client machine to communicate with each other is known as enrolment. The netHSM must be enrolled to the client by either entering or confirming the public key of the netHSM (HKNETI). Optionally, a client can be enrolled to the netHSM using strongly protected credentials.

System monitoring and reporting

System monitoring is also fully supported. The netHSM provides data, via the nCore API statistics interface, to the nCipher SNMP agent or the Windows Performance Monitor plug-in. The netHSM maintains a system log file of events as they happen. This log file can be viewed from the front panel, or saved to a remote client machine at regular intervals. The level of detail in the logging can be configured as required.

8.5 3rd party security validations

nCipher is committed to delivering high security and integrity products. Independent testing and validation is central to this commitment. Please see http://www.ncipher.com/nethsm/index.html for further details.

9. TRANSPORT SECURITY

The communication between a netHSM and its clients is required to be authenticated and encrypted. nCipher use a protocol called 'Impath' for this communication.

9.1 The Impath protocol

Impath ("inter-module path") is a protocol developed by nCipher to provide communication between two secure endpoints through an insecure medium. In outline it is similar to SSL; a public-key based initial exchange is used to derive a secret, from which are derived symmetric encryption keys and HMAC keys. These are used to protect the integrity and confidentiality of session traffic between the endpoints. This guarantees that:

- Messages sent through the Impath cannot be read except by the parties at each endpoint
- Messages not sent by one of the endpoints will be rejected as invalid by the other
- A replay of a previously valid message won't be accepted as valid by either endpoint

When an Impath is set up, each side generates a random challenge. A setup message is then exchanged. This message contains:

- The challenge
- A freshly-generated Diffie-Hellman ephemeral key
- A variety of attributes about the endpoint (such as the IP address, HSM serial number or other identifier)

The message can be unsigned (if the endpoint is not authenticating itself) or signed with one or more keys.

9.2 Comparison between Impath and SSL / TLS

SSL and TLS are standard and well established security protocols used extensively for exchanging information over the Internet. Whilst adequate for most applications, they are not considered sufficiently strong or robust for protecting security critical applications such as the interface to a Security Module, on which the whole security system depends. Some of the deficiencies of SSL are listed below:

- The use of cryptography in SSL is not state-of-the-art. For example, SSL follows the traditional approach of applying authentication first and then encryption. Following the publication by H. Krawczyk, "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)"; which presented a theoretical concern, a practical attack on certain SSL implementations was achieved (See http://www.securitytracker.com/alerts/2003/Feb/1006151.html). Best practice[1] now applies encryption first and then authentication.
- SSL public keys are transported in X.509 certificate format. Much of the information contained within this certificate is designed for human presentation; it cannot easily be understood by machine. This can lead to a number of errors and potential security vulnerabilities (See http://securitytracker.com/alerts/2002/Sep/1005211.html). By contrast, all messages used in Impath use data-types from the nCore API and need no interpretation stage to be meaningful to the HSM. This leads to a more robust security implementation.

[1] *H. Krawczyk, "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)"; also see http://www.securitytracker.com/alerts/2003/Feb/1006151.html*

- The initial protocol negotiation steps of SSL are complex and require a number of mechanisms to prevent an attacker exploiting flaws in earlier protocol versions. By contrast, the Impath session setup messages include well-defined means to accommodate future protocol changes, cipher suites, and new endpoint attributes. The data to do this is signed as part of the key exchange process, making it hard to attack.[2] (See V Klima et al "Attacking RSA based sessions in SSL/TLS", CHES 2003)
- The implementation of SSL is complex; for instance, many implementations contained buffer overruns in the ASN.1 parsing code used to process X.509 certificates (see http://securitytracker.com/alerts/2002/Jul/1004879.html for example).

Impath is much simpler in implementation (not least because all the messages are encoded in nCipher's robust 'marshalled' format); to date no such vulnerabilities have been found.

Full details of the Impath protocol can be found in the Impath.pdf document, available for download at http://www.ncipher.com/resources/downloads/index.html

10. CLIENT AUTHENTICATION

10.1 Client authentication options

It is important, in a network-connected system, for each party to be able to identify each other reliably so that an attacker cannot impersonate a legitimate user. netHSM supports a variety of options for authenticating both the server and the application, or user, requesting key use. This option allows an appropriate level of security to be applied dependent on the environment and customer preference. These options include:

- The ability to connect a server as a 'soft' client with no hardware present
- The use of a discrete hardware 'token' ('nToken') to provide strong protection of the secret associated with the Impath connection
- The use of a directly-connected nCipher HSM e.g. nShield allowing presentation of smart cards to authorise key use

10.2 General principles

Only clients that have been previously enrolled can connect to the netHSM. The nature of the Impath connection set-up protocol means that all traffic using that connection can only be read (or can only have been generated) by the party which undertook the key exchange process. This means, for instance, that consecutive commands from a client to a netHSM can be trusted to be from the same source (and in sequence). Identification of that party (i.e. binding to some other access control information) is done by that party signing the Impath set-up messages with a particular key.

The netHSM maintains a key-pair for signing its Impath session setup messages (KNETI); this is unique to that netHSM and is only renewed when a netHSM is returned to 'factory state'. The key hash of this (HKENTI) can be read from the netHSM's front panel and transferred to (or checked against) the remote server. Thus the remote server can be confident that it is talking to a particular netHSM when setting up the connection.

10.3 Software only clients

A server machine that has no locally connected nCipher HSM is referred to as a 'software-only' or 'soft' client. For soft clients, the netHSM does not require that the server sign its connection set-up message. In other words, at the point the connection starts, the client is anonymous to the netHSM. Instead, the client sends nCore API commands via the Impath which contain various forms of credentials for identification. These credentials can include:

- nCore API command certificates (i.e. a given request comes with a signed credential)
- Passphrases or PINs when reading shares from user
- The contents of a key blob (see section 12) for loading

The main advantage of this method is that when multiple applications are running on a given client machine, each application can supply its own credentials which are then kept separate from those associated with other applications. Different application keys can also be authorised with different credentials. If the only identification were of a particular server machine (i.e. of the endpoint of the Impath), it would be hard to separate different applications' requests for different keys. In other words, with a 'soft' client the Impath is not used to identify a client, merely to provide

[2] *See V Klima et al "Attacking RSA based sessions in SSL/TLS", CHES 2003*

a safe conduit through which it can supply its identification. It is necessary to identify the netHSM, however, so that the client only sends its credentials to a trusted endpoint.

It may be believed that additional protection could be established by requiring each client machine to identify itself (with a signing key) to the netHSM when setting up sessions. However, in a software-only client, such a key would have to be stored somewhere on the disk – it would be alongside other files containing the sort of credentials mentioned above, and would be no harder for an attacker to compromise. This would thus provide no greater security but provide a much coarser level of access control.

Operationally this arrangement is convenient. Clients can be connected without requiring hardware to be installed and it delivers the most appropriate and secure authorization possible, given that there is no hardware to protect enrolment secrets.

10.4 'nToken' hardware at client

In some situations, trusting a client's operating system to store its credentials, as described above, is insufficient. Therefore the netHSM architecture allows the use of the lightweight hardware nToken on the client to store an identifying key-pair. When an Impath is set up, the token signs the key-exchange message, which is verified by the netHSM. A particular machine is enrolled to the netHSM by allowing its nToken to generate a new key-pair and entering its key hash via the front panel user interface (or, equivalently, transmitting it over the network and verifying it via the user interface).

Application keys can then be generated in such a way as to be only usable by a client when its Impath is authenticated with a particular nToken. The critical difference that the nToken makes is that the key used to identify the client is stored in hardware on the client machine, not on the disk. This gives an immediate security increase: an attack can only be mounted whilst the given token is installed and powered on, from within that machine. The private key material is never stored outside the token itself; in the event of a compromise it can be permanently revoked by generating a new key in its place. It is easy to confirm (by physical inspection) the identity of the machine which uses a particular token; there is no need to trust the network setup, file system configuration, etc.

Operationally this arrangement allows the benefits of hardware enforced authentication of servers. This strong authentication allows strongly enforced assignment of keys to particular clients.

10.5 Existing HSM hardware at client

Systems with existing nCipher PCI and SCSI modules are able to use the Remote OCS feature with the netHSM. This allows an Operator Card from a Security World to be inserted into an HSM local to the client machine, and the data transferred securely (via Impath) to the netHSM. This provides the credentials required to load the keys associated with that Operator Card Set.

Operationally this arrangement allows strong, fine-grained control over key use (with credentials presented where convenient) at any location that has an nCipher HSM with smartcard slot available.

10.6 Summary of encryption and authentication options

Hardware at Client	Strong Encryption	netHSM authentication	Impath used for:	Client authentication	Client authentication credentials
No hardware	Yes	Yes – strong	Encryption	Yes – software based	KMdata file on disk or user-entered passphrase or nCore command certificate
"nToken"	Yes	Yes – strong	Encryption and Authentication	Yes – strong	Key held on nToken device
Directly connected HSM, using smart card slot	Yes	Yes – strong	Encryption and Authentication	Yes – strong	OCS smart card device

11. RESILIENCY ARCHITECTURE

All nCipher HSMs, whether directly-connected or network-connected, use 'Security World' mechanisms for the protection of key material. This ensures that access to critical keys is not dependent on a single hardware entity, protecting against complete HSM / netHSM failure. Any HSM within the same Security World has access to the same module key and hence can use all keys within that Security World to which it has permission.

Scalability, back-up and resiliency are automatically provided as part of the normal database and file system environment. No special procedures are required. When a new HSM or netHSM is required to be introduced to replace a failed device, it is initialised into the World using the World ACS and can then be operational immediately with access to the same keys as the failed unit.

nCipher HSMs provide load-balancing between available HSMs as standard. The same mechanism also allows automatic failover or 'High Availability' (HA) as standard, with the deployment of 2 or more HSMs. Load-balancing and HA with nCipher HSMs is managed by a nCipher supplied client software component known as the Hardserver. In order to load-balance across available HSM's the Hardserver being used by the application must be configured to be aware of the relevant network-connected HSMs. Any locally connected HSMs are automatically detected and available.

For a commercial application using a 'standard' interface (e.g. PKCS#11) the library will organize the load-sharing provided that the environment is correct e.g. the PKCS#11 environment variables are set. Available HSMs, either dedicated or network-connected, simply appear as 'slots' to a PKCS#11 application. Thus load-balancing and HA is possible across a mixed deployment of dedicated HSMs and netHSMs. If an HSM should fail or become isolated from the rest of the infrastructure, the remaining HSM(s) will automatically service the cryptographic load.

12. KEY MANAGEMENT AND PROTECTION FOR NETWORK CONNECTED HSMs

It is possible to protect keys in an HSM by the simple mechanism of holding them internally, protecting a small and finite number of keys. For directly-connected HSMs this mechanism has operational disadvantages, for a network connected device these operational disadvantages become more serious.

Networked devices will typically protect keys for multiple servers. This aggregation means that more keys require protection, significantly increasing the required storage capacity within the HSM. Network connected devices are likely to be installed where regular physical access is inconvenient. This means that Key backup mechanisms that require physical presentation of backup tokens may be operationally unacceptable. Similarly it is difficult to manage a resilient system if this requires physically transporting backup tokens between devices that may not be co-located.

To overcome these issues, nCipher developed the Security World key management architecture that cryptographically protects keys, binding them with fine-grained access control lists and usage parameters to form a key blob. Since key blobs are cryptographically protected using a higher level module key, they can be safely stored on an external file system. Any HSM within the same Security World has access to the same module key and hence can automatically use all keys within that Security World to which it has permission. Scalability, back-up and resiliency are automatically provided as part of the normal database and file system environment. No special procedures are required.

Host-side software, known as the Hardserver, automatically stores and fetches key blobs and load balances tasks between available netHSMs or dedicated HSMs transparently to the host applications.

12.1 Dynamic key storage vs 'partitioning'
The Security World system provides an existing, flexible and scalable way of separating keys into groups using Operator Card Sets (OCS). This compares to the rigid and inflexible partitioning technique used by some other HSMs.

Each OCS can have the 'k' and 'n' secret-sharing parameters individually set, allowing a flexible choice between maximum security and operational convenience. Keys protected by one OCS cannot be modified or used by presenting another OCS. The total number of card sets, and the number of keys in each set, is limited only by disk storage capacity - with this approach several thousand card sets and tens of thousands of keys can be supported easily.

For systems with nToken hardware, it is possible to run 'unattended' (i.e. without the insertion of cards) and still provide hardware-enforced separation of keys use. This feature allows the applications or clients using a key to be identified to the HSM; a key can therefore be locked down so that it is only usable by a Client possessing a particular nToken. When a Client machine then connects to the netHSM, it will use the nToken's key to identify it and look up its privileges; the netHSM will then allow access to those keys which require this credential. So, a particular group of machines can be identified as 'Web Servers' and another as 'Certificate Authorities'; SSL keys can be created that are only accessible to the first group and CA keys that are only accessible to the second. Another machine may be designated a 'management machine', and will have membership of both groups. There are no practical limitations on the number of different groups which can be set up in a netHSM system.

12.2 Location of encrypted key files
The data held in a nCipher Security World consists of secrets held within the HSM itself, shared secrets distributed amongst the ACS and OCS card sets and data files held on the client machine's disk. The data files include:

- Control files for the World itself, the modules and operator card-sets
- Key files containing encrypted key blobs of all the application keys

For the netHSM, all files except the key files are kept in non-volatile storage inside the platform security boundary. The internal Security World utilities accessible from the front panel operate on these files. This 'master copy' of the Security World data can be transferred to other client machines for local use. It is also possible to import existing Security World files into the netHSM.

Encrypted key blob files are not, in general, stored inside the netHSM. These key files are generated at the client using the KeySafe tool, or directly by the application, and are used at the host server. This architecture allows the system to scale to protect many thousands or millions of keys without requiring large amounts of storage inside the netHSM. Key files may, however, be imported to the netHSM for examination and display using the secure user interface to verify correct generation.

13. PERFORMANCE ARCHITECTURE

Correct hardware design and software architecture is necessary if a network connected HSM is to deliver satisfactory cryptographic service to multiple clients. The netHSM features the following performance-related design points:

- Latest 1600 TPS (transactions per second) FIPS validated internal HSM, with high speed PCI interface
- Simple, single process software architecture on Intel-based system controller motherboard maximises use of CPU power
- Queues of multiple outstanding commands can be held at the Client, on the system controller motherboard and on the internal HSM to remove the effects of network latency on throughput
- Aggregation of multiple commands and replies into single packet for transport efficiency
- Client Hardserver process balances load amongst dedicated and netHSMs, automatically responding to levels of activity on network.

14. GLOSSARY

ACS (Administrator Card Set)
Administrator smart cards (ACS) are used to control access to Security World configuration and recovery operations and are required for FIPS level 3 compliance. Every Security World will have one ACS. An ACS is created using a k of n model to provide resiliency against lost or damaged cards. With netHSM, ACS cards are only ever presented directly to the local smart card slot on the netHSM. All security critical ACS functions execute within the netHSM.

Client
In a system involving netHSMs the term 'client' is used to describe server machines that connect to a netHSM for cryptographic operations. A single netHSM may have multiple 'clients', meaning that multiple servers can connect to a single or multiple netHSM(s).

CodeSafe
CodeSafe is the nCipher brand name for a cryptographic toolkit that allows application code to be executed within a cryptographic module with strongly bound rights to key material. The CodeSafe toolkit is thus used in conjunction with nCipher's Secure Execution Engine (SEE) technology. Versions of netHSM are available featuring SEE.

FIPS
FIPS 140-2 is the latest Federal Information Processing Standard that outlines security requirements for cryptographic modules. FIPS 140 was developed as part of the Cryptographic Module Validation Program (CMVP), a joint effort by NIST and the Communications Security Establishment of Canada. Initially developed for federal agencies using cryptographic-based security systems, the original FIPS 140-1 standard has become a widely used benchmark throughout the business world. To receive FIPS validation, cryptographic modules are tested by independent, accredited testing laboratories and a report submitted to NIST.

Hardserver
The Hardserver is the communications service between applications and nCipher modules. It is installed automatically as part of the netHSM client software. Once correctly configured the Hardserver administers the connection between client applications and directly-connected and/or netHSM modules. It provides failover and loadsharing between any available netHSM and directly-connected modules. Connections are secured between the client Hardserver and a netHSM using the Impath protocol.

Impath
Impath is an abbreviation of 'Inter-module path', a secure protocol used to communicate over IP networks. Impath connections are used between netHSM and clients. Data sent through such a channel is secure against both eavesdroppers and active adversaries. Impath is based on an ephemeral DH key exchange and is described in more details here (http://www.ncipher.com/resources/downloads/index.html)

KMdata
The Security World design keeps all critical plaintext data securely within the FIPS boundary of the HSM. Other data is kept in control files on disk within the Platform Security Boundary of the NetHSM or on the client as required, in the KMdata directory. Where necessary, data files in the KMdata directory are encrypted and/or MAC'd with keys kept within the module or on card sets. KMdata files include control files for card sets and modules, and files containing securely encrypted key blobs. By storing data on disk in this way, the number of keys and/or Operator Card Sets within in a Security World is effectively unlimited.

KNETI / HKNETI
KNETI is the netHSM integrity key. This is a signature key maintained within the netHSM and regenerated whenever a Security World is installed, or the unit returned to factory state. The netHSM uses KNETI to sign its Impath setup messages, in order for the client to be sure it is connecting to a genuine netHSM. The hash of KNETI (HKNETI) is available from the front panel menu of the netHSM; it can then be transferred out of band to the client avoiding the need to trust the network at any stage of setup.

Module key
A module key is an encryption key that is stored in the key management module and used to protect the data stored on Card Sets. A module can store multiple module keys.

nCore API
The programming interface used to access nCipher modules. The interface is based around a set of commands; the application using the module constructs a command block containing the command type and parameters, and submits it for processing. When complete, a reply is returned containing result data.

nToken
nToken is the name for an optional hardware device installed at a netHSM client machine. This hardware exists for the protection of a private key that is used in establishing the Impath connection between a client and a netHSM. Protecting this key in hardware provides strong authentication of the client to the netHSM.

OCS (Operator Card Set)
An OCS is the nCipher mechanism for strongly restricting the use of a key to a legitimate user. To restrict key access to a particular user, a set of smart cards known as an Operator Card Set is created. There is no limit to the number of Operator Card Sets that can be created within a Security World. An Operator Card Set belongs to a specific Security World. It cannot be read, erased, or even formatted except in a module from its Security World. An Operator Card Set stores a number of symmetric keys that are used to protect the working keys. These keys are Triple DES keys. Each card in an Operator Card Set stores only a fragment of the Operator Card Set keys. These keys can only be re-created if you have access to enough of their fragments. Because cards sometimes fail or are lost, the number of fragments required to re-create the key (k) must be less than the total number of fragments (n). The k and n values can be chosen when the card set is created. With netHSM it is possible to present OCS cards 'remotely' for the netHSM - i.e. at the slot of an HSM directly connected to the client.

Open BSD operating system
See www.openbsd.org for more details.

"OpenBSD's operating system consistently gets rave reviews from users who find that they do not face the same constant fear of hackers that owners of other operating systems encounter. Although it is often difficult to prove that an operating system has had thorough security testing, an extraordinary event provided relevant data. In the 2003 competition among military academies and grad schools, in which they competed to provide the best defence against cyber attacks launched by National Security Agency specialists, the judges acknowledged that in the final analysis, use of OpenBSD was a determining factor in the winner's ability to fight off attacks. This award demonstrates that even in complex software like operating systems, unequivocal, uncompromising attention to writing safe software does work."

Security World
Security World is a secure key management framework that allows the generation of cryptographic keys within netHSM; sets the capabilities and security limits of keys; implements key backup and recovery options.
A Security World consists of:

- One or more nCipher netHSM; payShield, nForce or nShield modules
- A set of Administrator smart cards used to control access to Security World configuration and recovery operations (FIPS level 3 compliant Security Worlds require the use of Administrator smart cards to authorize most operations)
- An optional set, or sets, of Operator smart cards used to control access to application keys
- Some cryptographic key and certificate data, encrypted using the Security World key, stored on a client computer or computers.

SEE
SEE (Secure Execution Engine) is the name for a unique nCipher technology that allows the protection of both key material and security sensitive application logic with the FIPS security boundary of a netHSM. More can be learnt about SEE at:
http://www.ncipher.com/technologies/

NCIPHER netHSM TECHNICAL ARCHITECTURE

nCipher has used various open source software components and is required to identify those individuals and organizations that have contributed to this open source material.

This product includes software developed by:
Aaron Brown; Aaron Campbell; Adam Glass; Alice Group; Allen Briggs; Amancio Hasty; Ben Harris; Berkeley Software Design, Inc.; Bernd Ernesti; Bill Paul; Brad Pepers; Bradley A. Grantham; Brian Dunford-Shore; C Stone; Charles D. Cranor; Charles Hannum; Christian E. Hopps; Christopher G. Demetriou; Christos Zoulas; Colin Wood; Computer Systems Engineering Group at Lawrence Berkeley Laboratory; Computer Systems Laboratory at the University of Utah; Dale Rahn; Daniel Widenfalk; David Jones; David Miller; Dean Huxley; Denis A. Doroshenko; Eric S. Raymond; Eric Young; Ericsson Radio Systems; Ezra Story; Frank van der Linden; Gardner Buchanan; Garrett A. Wollman; Gordon Ross; Harvard University and its contributors; HAYAKAWA Koichi; Hellmuth Michaelis; Herb Peyerl; Iain Hibbert; Ignatios Souvatzis; Information Technology Division, US Naval Research Laboratory; Internet Research Institute, Inc.; James R. Maynard III; Jason Downs; Jason L. Wright; Jason R. Thorpe; Jim Lowe; Joachim Koenig-Baltes; Job de Haas; Joerg Wunsch; John P. Wittkoski; Jonathan Stone; Juergen Hannken-Illjes; Jukka Marin; Julian Highfield; Kari Mettinen; Kenneth Stailey; Klaus Burkert; Kungliga Tekniska Högskolan; LAN Media Corporation; Leo Weppelman; Lutz Vieweg Manuel Bouyer; Marc Espie; Marc Horowitz; Mark Tinguely; Markus Wild; Martin Husemann; Mats O Jansson; Matthew R. Green; Matthias Drochner; Matthias Pfaller; Michael L. Hitch; Michael Shalayeff; Michael Teske; Michael van Elst; Mika Kortelainen; NetBSD Foundation, Inc.;Network Research Group at Lawrence Berkeley Laboratory; Niels Provos; Niklas Hallqvist; Nivas Madhur; Paul Kranenburg; Per Fogelstrom; Peter Galbavy; Philip A. Nelson; Roar Thronæs; Rodney W. Grimes; Roger Hardiman; Roland C. Dowdeswell; Rolf Grossmann; Ross Harvey; Scott Bartram; Scott Reynolds; Scott Turner; Seth Widoff; SigmaSoft; SMCC Technology Development Group at Sun Microsystems, Inc.; Stephan Thesing; Steve Murphree, Jr.; Takashi Hamada; Tatoku Ogaito Terrence R. Lambert; Th. Lockert; Theo de Raadt; Theodore Ts'o; Thomas Skibo; Tobias Weingartner; TooLs GmbH; University of California, Berkeley and its contributors; University of Lule, Sweden and its contributors; University of Vermont and State Agricultural College; Waldi Ravens; Wasabi Systems, Inc.; Washington University; William F. Jolitz; Winning Strategies, Inc.; Wolfgang Solfrank; Zembu Labs, Inc.

Customers should check with their nCipher sales contact for the availability of functionality described in this document.

NCWP/NTA/OCT2003

nCipher Inc.
500 Unicorn Park Drive
Woburn, MA 01801 USA
Tel: +1 (781) 994 4000
E-mail: ussales@ncipher.com

nCipher Corporation Ltd.
Jupiter House, Station Rd.
Cambridge, CBI 2JD UK
Tel: +44 (0) 1223 723600
E-mail: int-sales@ncipher.com

nCipher Corporation Ltd.
15th Floor, Cerulean Tower,
26-1 Sakuragaoka-cho, Shibuya-ku,
Tokyo 150 8512 Japan
Tel: +81 3 5456 5484
E-mail: int-sales@ncipher.com

Visit our Web site at
www.ncipher.com – today!

Redefining cryptographic security

nCIPHER™