nFORCE Ultra Asymmetric Module

nCIPHER™

Date: **20 February 2007**

Version: **1.0.1**

Patents

UK Patent GB9714757.3. Corresponding patents/applications in USA, Canada, South Africa, Japan and International Patent Application PCT/GB98/00142.

Contents

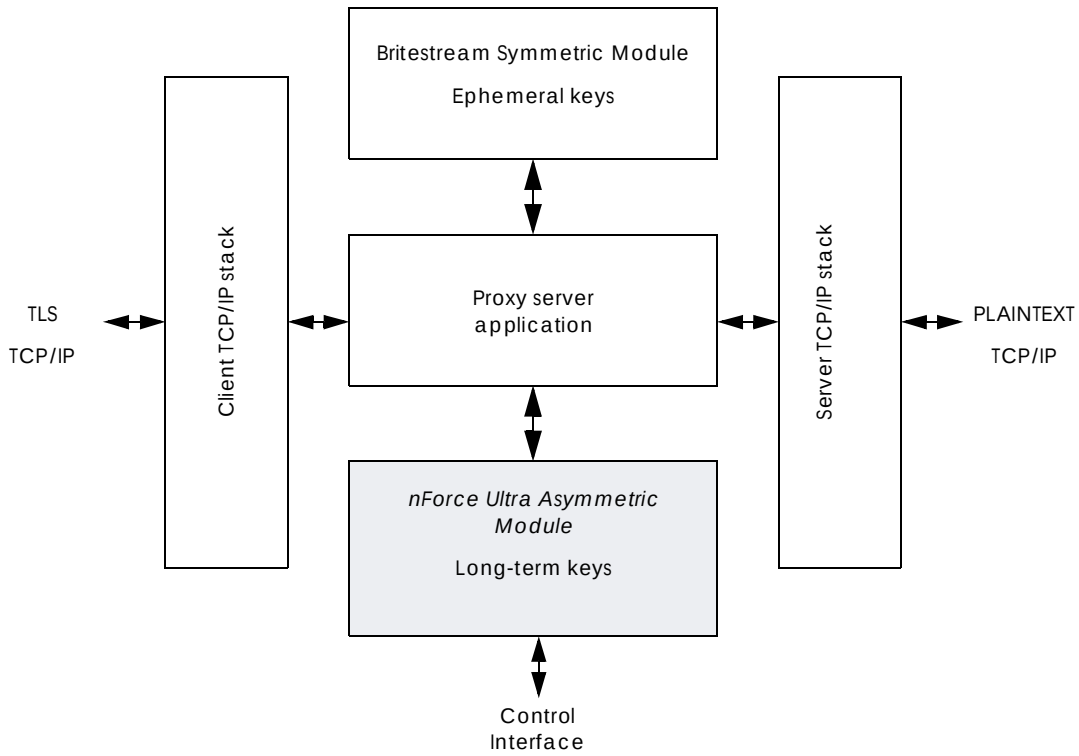
nForce Ultra Asymmetric Module

The *nForce Ultra Asymmetric Module* version 1.0.1 forms part of nForce Ultra products co-developed by nCipher and Britestream.

The nCipher nForce Ultra cards are PCI cards that act as TLS proxy servers, with secure TCP/IP communication on one ethernet port and plain text TCP/IP communication on a physically separate port. These cards completely off load the TLS processing from the host computer delivering secure internet communication at full line speeds.

The main components of the proxy server are physically resident on a single chip - the Britestream BN2010 chip - which has multiple processor cores plus dedicated hardware. This chip requires a small number of additional components, including memory, ethernet PHYs, etc.

The proxy server on the BN2010 uses two separate cryptographic modules. The FIPS 140-2 level module *nForce Ultra Asymmetric Module* for long term asymmetric keys and the FIPS 140-2 level 1 *Britestream Symmetric Module* for ephemeral symmetric keys, logically divided as shown in the following diagram:



Note *This validation is for the nForce Ultra Asymmetric Module only. There is a separate FIPS 140-2 level 1 validation for the Britestream Symmetric Module. Refer to the security policy for that module, FIPS 140-2 certificate ###, for details of symmetric cryptographic operation.*

The *nForce Ultra Asymmetric Module* acts as the administrator user for the *Britestream Symmetric Module*.

All communication from the operator goes through the *nForce Ultra Asymmetric Module.*

All configuration commands for the *Britestream Symmetric Module.* and for the non-cryptographic portions of the B2010 are entered via the control interface of *nForce Ultra Asymmetric Module*.

In order to configure these components an operator must first log into the *nForce Ultra Asymmetric Module* and then enter commands at its control interface. The *nForce Ultra Asymmetric Module* forwards these commands to the other components of the chip.

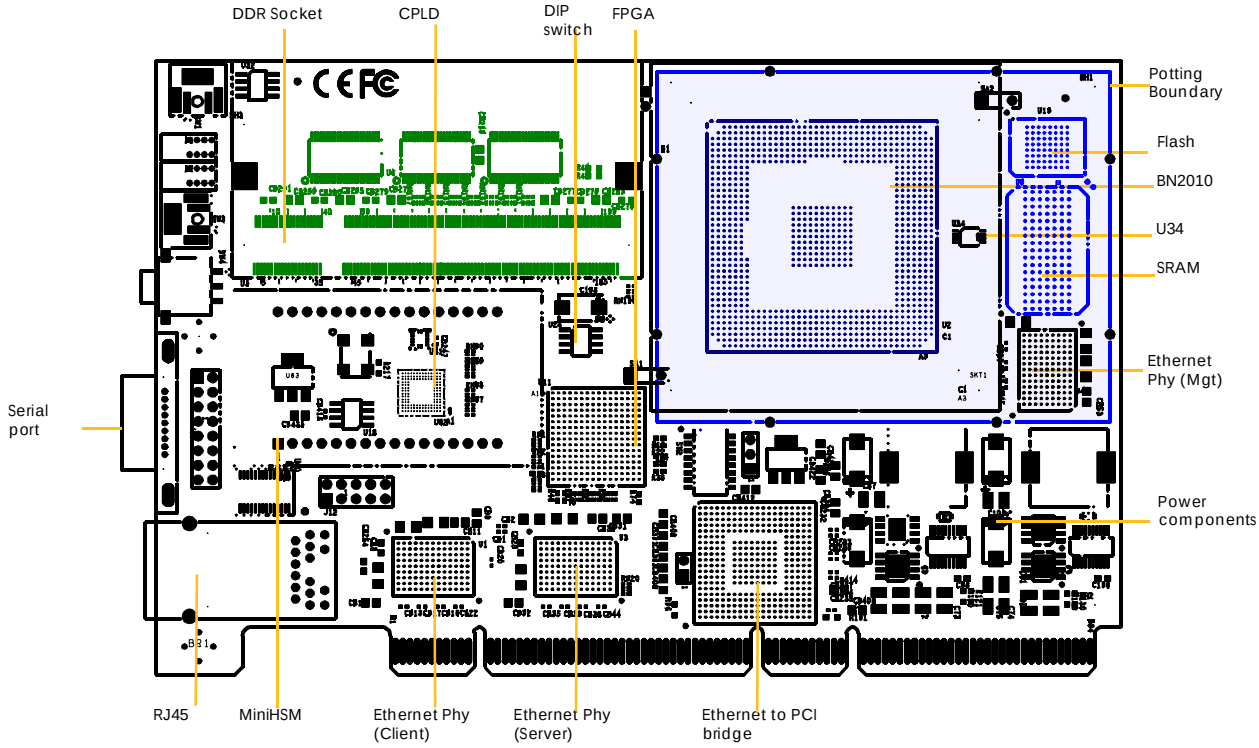
The commands for the *Britestream Symmetric Module* are listed in this security policy – as they are entered via this module. These commands are described in detail in the security policy for the *Britestream Symmetric Module*.

Implementation

The *nForce Ultra Asymmetric Module* is an embedded multi-chip module as defined in FIPS 140-2, which consists of the management ARC processor on the BN2010 chip, flash memory and SRAM memory. It provides asymmetric key operations as part of the TLS protocol.

The following diagram shows the nCipher nForce Ultra card with the components that form the *nForce Ultra Asymmetric Module* and the potting boundary highlighted in blue.

Figure 1 PCI board layout



The BN2010 chip - shown in the darker blue - contains several ARC processors. Only one processor - the management ARC - is used by the *nForce Ultra Asymmetric Module*, the other processors and the components in green form the *Britestream Symmetric Module* which is validated separately.

The *nForce Ultra Asymmetric Module* has the following version numbers:

- Hardware version 010-00007 a.00
- Firmware version 610-00014 1.0.0.

Ports and interfaces

The module has the following ports:

Logical Port	Physical Port
Data in:	Internal bus 1
Data out:	Internal bus 1
Command In:	Internal bus 2, serial port, reset pin, factory state pin
Status Out:	Internal buses 1 & 2
Power In:	Power pins

The internal bus 1 connects to the SPP arc within the BN2010 chip.

The internal bus 2 connects to two separate logical ports on a TCP/IP interface, one for commands in and status out and one solely for status out.

If the serial port is enabled the TCP/IP interface is disabled and vice versa. The serial port is only used to load new firmware in cases where the loaded firmware is corrupt. This is controlled by DIP switches.

Status information can be routed to the management port, or through the proxy server to the server TCP/IP port. This routing is set by the administrator user.

The chip can be reset using the reset pin. This pin is connected to a push button on the face of the PCI card.

The module can be reset to factory state by starting the chip with the factory state pin connected to 5V. This pin is connected to one of the DIP switches.

Roles

The module supports the following roles.

Administration role

To connect in the Administration role you must connect on the external management port and supply a user id and password for a user with administrator privileges.

The administrator can change the password for the administrator and junior administrator user.

The Administration role can configure the module, configure non-cryptographic functions on the TLS security chip, configure connections to a key server, initiate connections to a key server, load keys from a key server and obtain status messages.

Junior administration role

To connect in the Junior Administration role you must connect on the external management port and supply a user id and password for a user with junior administrator privilege.

A junior administrator cannot connect until the administrator user has set up their pass phrase. A user with junior administrator privileges cannot change passwords.

The Junior Administration role can configure the module, configure non-cryptographic functions on the TLS security chip, configure connections to a key server, initiate connections to a key server, load keys from a key server and obtain status messages.

TLS user role

The TLS user role is assumed by authenticating the TLS user application with a RSA digital signature.

Connections to the module, via the TLS user role, are made on the dedicated internal channel.

The module's verification of a RSA signature confirms the application as an approved TLS user.

The TLS user role can initiate sign and verify operations. The user role refers to keys by index - it has no access to private key material.

Services

In the following description of the services available, the following terms are used to describe access to critical security parameters.

Key access	Description
Create	Creates a in-memory object, but does not reveal value.
Erase	Erases the object from memory, smart card or non-volatile memory without revealing value
Export	Discloses a value, but does not allow value to be changed.
Regenerate	Generates a new value for a CSP and writes this value in the same location as the existing value, thus erasing the original values; this operation does not reveal either the new or the old value.
Set	Changes a CSP to a given value
Use	Performs an operation with an existing CSP - without revealing or changing the CSP

Unauthenticated commands

The following commands are available without authentication. They have no access to Keys or CSPs.

Service	Description
connect	Opens a connection to a specific module identified by IP address and port.
exit	Closes an open connection.

Authentication command

The following command is used to identify a user and authorize them to assume the administrator or junior administrator role.

Service	Description Access to CSPs Key Types used
login	Login into the module establishing identity and opening a session. ***Uses*** *password*

Administrator and junior administrator roles

The following services are available to the administrator and junior administrator roles after they have connected and logged in.

Service	Description Access to CSPs Key Types used
logout	Logs out ending a session. *No access to CSPs*
passwd	Changes the password for a user - administration user only. ***Sets** password*
run	Starts the level 3 module and level 1 module. *No access to CSPs*
halt	Stops the TLS security firmware - preventing access to all ports, other than management port. *No access to CSPs*
softReset	Resets the level 3 module, clearing all keys and causing all self tests to be run and disconnecting all users. ***Clears** all session keys*
readOpStatus	Displays the status of the level 3 module. *No access to CSPs*
nResetKDP	Regenerate KDC and KDI. ***Regenerates** KDI, KDC* *DSA, Diffie-Hellman*
nClearKDP	Zeroize all the secure certificates and keys in memory and flash. ***Clears** keys*
getKDPEnrollInfo	Read KDP enrollment information. ***Exports** public halves of KDI and KDC* DSA, Diffie-Hellman
getFIPSMode	Display current FIPS mode. *No access to CSPs*
viewCertificate	View the certificate details including the SHA-1 hash of the key. ***Uses** a server key* SHA-1

Service	Description Access to CSPs Key Types used
getAllCert	List all existing X.509 certificates, does not list key material. ***Uses*** *a server key* SHA-1
delCertificate	Delete the certificate and associated keys ***Erases*** *server key*
loadSecureCertificate	Set the EKM key identifier for the certificate/key. Causes module to fetch encrypted key from KDCP proxy server. ***Uses*** *KDI, KDC, and session keys,* ***sets*** *server key* DSA, Diffie Hellman, AES, SHA-1
upgradeFW	Download firmware to BN2010. The module will only accept firmware if it can verify the RSA signature on the firmware image. ***Uses KBS***, ***replaces*** *firmware.* RSA
readFWInfo	Read firmware information
saveOpConfig	Save the current operational configuration writing a HMAC to validate. HMAC SHA-1
eraseOpConfig	Delete the saved operational configuration.

The following commands configure the *Britestream Symmetric Module* and non-cryptographic portions of the BN2010 chip.

These functions are included here as they can only be accessed through the *nForce Ultra Asymmetric Module* and the user must log into the *nForce Ultra Asymmetric Module* before issuing these commands.

These commands do not have access to CSPs.

Service	Description
setPassThru	En/Disable passthrough traffic - see level 1 security policy
getPassThru	Passthrough traffic setting - see level 1 security policy
syncClock	Set BN2010 clock to the system clock
getClock	Get BN2010 clock
getSystemInfo	Read system information
setWatchDog	En/Disable watchdog feature
getWatchDog	Watchdog feature setting

Service	Description
setProxy	Setup a TCP proxy
setBackChProxyID	Set up a backchannel proxy
getProxy	List the TCP proxy
delProxy	Delete the TCP proxy
getAllProxy	List all existing proxies
setProxySSL	Set up SSL attribute for the proxy
getProxySSL	Display SSL attribute for the proxy
setSessionIDTimeout	Set session ID timeout
getSessionIDTimeout	Get session ID timeout
setPortBlocking	Setup a blocked TCP/IP address entry
getPortBlocking	Get a blocked TCP/IP address entry
delPortBlocking	Delete the blocked TCP/IP address entry
getAllBlocking	List all existing blocked TCP/IP address entries
setRehandshakeMaxTimeOut	Set maximum time before SSL rehandshake
getRehandshakeMaxTimeOut	Get maximum time before SSL rehandshake
setRehandshakeMaxSeqNum	Set maximum sequence number before SSL rehandshake
getRehandshakeMaxSeqNum	Get maximum sequence number before SSL rehandshake
setMgmtTCPIP	Set mgmt port TCP/IP address at next powerup
getMgmtTCPIP	Display mgmt port TCP/IP address
getMgmtTCPIPStored	Display mgmt port TCP/IP address at next powerup
getTCPMaxConn	Display maximum TCP connection setup
setICMP	En/Disable ICMP
getICMP	Get ICMP setting
setIPFragment	Pass-through or discard IP fragments
getIPFragment	IP fragment setting
setLBMode	Set Load Balancing Mode
getLBMode	Get Load Balancing Mode
setTCPOption	Set TCP option
getTCPOption	Get TCP option
setMACAddr	Set MAC address for ports at next powerup
getMACAddr	Get MAC address for ports
getMACAddrStored	Get MAC address for ports at next powerup
getMACStatus	Get MAC status for ports

Service	Description
setProxyState	En/Disable proxy processing
getProxyState	Current proxy processing status
setAllProxyState	En/Disable processing for all proxies
getAllProxyState	Current processing status for all proxies
setBlockState	En/Disable port blocking processing
getBlockState	Current port blocking processing status
setAllBlockState	En/Disable processing for all blockings
getAllBlockState	Current processing status for all blockings
getEthernetStats	Ethernet statistics info
getNetworkStats	Network statistics info
getSSLTLSStats	SSL/TLS statistics info
setStatsControl	Set up statistics refreshing
getStatsControl	Display statistics refreshing setup
setAlertControl	En/Disable alert messaging
getAlertControl	Display alert messaging setting
setBackChTCPIP	Set back channel TCP/IP address
getBackChTCPIP	Display back channel TCP/IP address
readThermal	Read the thermal sensor
clearThermal	Clear all thermal records
setThermalWatch	Set thermal watch config
getThermalWatch	Get thermal watch config
setThermalAlert	En/Disable thermal alert
getThermalAlert	Get thermal alert setting
setGlobalCipher	Set up global cipher suites for the level 1 module
getGlobalCipher	Display global cipher suites for the level 1 module

TLS user Role

The following services are available to the TLS user role.

Service	Description Access to CSPs Key types
TLS	Uses a key, loaded by the administrator user, for TLS setup. ***Uses** a server key* RSA private key

Keys

The *nForce Ultra Asymmetric Module* does not generate the RSA private keys used for the TLS protocol. These keys must be imported - in encrypted form - from an external module acting as a key server. This is usually an nCipher nShield module.

The level 3 module and the key server communicate using nCipher's KDP/KDCP protocols under the control of the Administrator user. The level 3 module is identified by a DSA signature key and Diffie Hellman key-exchange key, which are generated by the level 3 module under the instruction of the Administrator user. The administrator user must export the public half of these keys and transport them to the key server.

Once a RSA private key has been decrypted inside the module there is no mechanism to export keys or access key material, except to use the key.

KBS

The Britestream private key, used to sign the module firmware.

This is a 4096-bit RSA key, which is used for firmware authentication and to authenticate the TLS user.

The private half is stored securely at Britestream and is never revealed.

The public half is written into the bootloader code stored in the module ROM at chip manufacture.

There is no access to the public key value.

KDI

The is a DSA key pair used to identify this module.

The private half of this pair is never revealed. The public half can be retrieved by the Administration user, this is exported and sent to the key server.

Whenever the level 3 module connects to the key server, it signs a message with the private key to prove its identity.

KDC

A 2048-bit Diffie Hellman key used in key exchanges to establish a symmetric wrapping key.

Keys are transferred from the key server to the *nForce Ultra Asymmetric Module* using the nCipher Key Distribution Protocol (KDP).

KDP specifies the cryptography used in the transfer. A separate protocol Key Delivery Control Protocol (KDCP) controls the communication layer.

The KDP protocol uses Diffie Hellman keys to agree a symmetric wrapping key that is used to encrypt the message. The protocol includes various nonces to protect against replay and uses signing keys (KDI) to identify the end points to ensure keys are only ever delivered to the correct modules.

Wrapping key

A wrapping key is either:

- a 128-bit AES key providing 128 bits of encryption strength

or

- a three key Triple DES key providing 112 bits of encryption strength.

Wrapping keys are used to protect KDP Session Keys in transit from the key server to the *nForce Ultra Asymmetric Module*.

Wrapping keys are discarded at the end of each session.

KDP Session keys

A KDP session key is a 2048-bit Diffie-Hellman private key used within key exchange within a KDP session. To minimize the use of KDC and to ensure security on the key server, KDC is only used for the initial exchange of messages.

The key server generates a KDP session key and sends this to the *nForce Ultra Asymmetric Module*. The KDP session key is used to establish a session wrapping key used in subsequent messages in the session.

Session KDP keys are discarded at the end of each session.

Session wrapping key

A session wrapping key is either:

- a 128-bit AES key providing 128 bits of encryption strength

or

- a three key Triple DES key providing 112 bits of encryption strength.

The session wrapping key is always the same format as the wrapping key negotiated at the start of the KDP session.

Session Wrapping keys are used to protect Server Private keys in transit from the key server to the *nForce Ultra Asymmetric Module*.

Wrapping keys are discarded at the end of each session.

Server private keys

Server private keys are the RSA private keys used within the TLS protocol.

Server private keys are transported from a key server encrypted under symmetric wrapping key established using KDC or a wrapping key.

Server private keys are not used to wrap other keys.

Server private keys are transported with metadata that indicates if have a timeout, and whether they can be stored in flash or must be reloaded from the server.

Passwords

The module uses passwords to identify the Security Officer and Junior Security O9fficer users.

These passwords are stored in NVRAM.

Each character can be one of 255 values. Values that cannot be typed directly from the keyboard can be entered by holding down the Alt key and entering the hex value on the numeric key pad.

One random guess would result in 1:262 trillion chance getting the correct value.

One login takes ~0.16 sec. therefore an attacker can make no more than (1/0.16)* 60 = 375 login attempts per minute.

All login attempts are sequential.

375:262trillion = 1:699 billion/minute

Even if you restrict choice of character to the 96 characters that can by directly typed, there are 885,842,380,864 possible six character passwords and the chance of success in a minute is approximately 1:2 billion.

Authentication of TLS User

The TLS user role is assumed by the TLS application authenticating to the module prior to setting up TLSuser sessions. The module checks a signature made on the application using the key KBS, see *KBS* on 14.

In order to to make multiple attempts to forge this signature, the attacker must load a new firmware image containing the forged signature between attempts. If the internal image does not contain a valid singature, the module does not accept commands and a new firware image must be loaded using the serial port.

The module checks the size of the image - and will reject an image that is too small. Give the amount of data to be loaded and the speed of the serial port, loading a firmware image takes significantly more than one minute to complete.

The attacker can therefore only make one guess per minute.

In order for an unauthorized application to assume this role, it would have to have a fake signature that randomly verified. Without knowledge of the private key the chance of creating a random byte block that is a valid signature which is equivalent to breaking a 4096–bit key: that is it provides 150-bits of encyption strength.

Rules

The module cannot generate the RSA keys used directly in TLS. These keys must be generated in a separate key server.

The module does generate the Keys used to identify the module and protect the keys in transit.

Before the module can be used, the Administrator must generate a KDI/KDC pair and enroll the module with a key server.

The Administrator configures the KDCP parameters. These are the TCP/IP transport parameters that tell the module how to connect to the Key Server.

To load a key, the Administrator uses the LoadSecureCertificate command. This causes the module to communicate with the key server and request the encrypted key.

The level 3 module uses DSA signed Diffie Hellman to establish a secure connection to the key server.

Self tests

The module performs power up and conditional self tests. All data output is inhibited while the module is performing self tests or if it enters an error state as a result of a test failing.

Power up self test

At power up the module performs the following tests:

- board hardware tests
- firmware integrity - RSA signature verification
- algorithm known answer tests, DSA, RSA signature and verification, AES, Triple DES, SHA-1 and HMAC
- pRNG known answer test.

Conditional self tests

When the module generates a new DSA or Diffie-Hellman key pair, it performs a pairwise consistency check.

The module also performs a continuous test on the pRNG output whenever a random number is requested.

Delivery and operation

The module is supplied with a default configuration.

In order to use the module in FIPS mode, the administrator must install the card in a server, log into the module, change the default password to a secure password, configure the connection to a key server and load at least one private key.

Installing the card

The nForce Ultra is a PCI board. The board fits into any standard PIC socket - though preferably a 66-bit 66MHz socket.

Once the card has been physically installed, you must install the host software and drivers.

See the installation guide for full details.

Changing password

The module is supplied with a default password.

The Administrator must connect to the card, log on using the default password and then use the password command to change the password to a secure value. The module will not allow any other operations until the administrator has changed the password.

The password is an string of at least 6 characters: see *Passwords* on 16.

Configuring the key server

The level 3 module does not generate server private keys. It requires a key server to supply the server private keys over an encrypted channel.

The key server must be configured with an nCipher security world, see the security policy for those module for details.

You must also run the nCipher key server application, this application manages the communication between the MiniHSM and the *nForce Ultra Asymmetric Module*.

You must enroll the *nForce Ultra Asymmetric Module* with the key server. Use the `getenrolmentinfo` command to write the public half of KDI and KDC to a file and then use the key server's utility to enrol these values and bind them to an network address.

You can generate a key for use by the *nForce Ultra Asymmetric Module*, using any of the standard tools provided by nCipher. Keys for use with the *nForce Ultra Asymmetric Module* must have the `apptype` KPM.

Loading a private key

Use the loadSecureCertificate command to load a key. You must supply the IP address and port or the Key server and the name of the key.

Submitting this command causes the module to open a secure channel, protected by an AES key established using Diffie Hellman key exchange using KDC, to the key server and send a request for the named key.

Assuming that the key server verifies the module's signature - a DSA signature made using KDI, and also verifies that the key's ACL allows export to this module. The key server performs a Diffie Hellman key exchange with the module to establish a session key and sends required RSA private key, the X.509 certificate and meta data to the module wrapped with this session key. The module decrypts the application key and stores it in working memory.

If any of these operations fails, the module returns an error code.

If the meta data sent with the key indicates that the key is cacheable, the module writes a copy of the key data into its flash memory. If the module is reset it loads any cached keys from flash, these keys are then available to a user once the user's identity has been validated.

If the metadata with the key does not allow caching, the key is only written to ephemeral memory and must be refetched from the key server if the module is reset.

To determine which keys are loaded, use the readAllCertificates command to list the loaded keys. This command displays the SHA-1 hash of the key and the associated X.509 certificate chain.

Configure the proxy

Once the private key is loaded you must set up the proxy server to use this key. Use the SetProxy command to define the key identifier to use for a specific TCP address. Use SetProxySSL to set the SSL settings for the proxy, which will define the key types used by the level 1 module.

Physical security

All security critical components of the module are covered by epoxy resin.

Strength of functions

Users connecting on the management port must provide a password to prove their identity.

Before the TLS operator can connect on the internal bus, the module verifies a 4096 bit RSA - SHA-1 signature on the application firmware.

Keys transported from the key server are protected by a 2048-bit Diffie Hellman key exchange. The module and server identify themselves using 1024-bit DSA signature keys.

Algorithms

The module uses the following algorithms:

DSA

Certificate 138

Diffie Hellman

Key agreement, key establishment methodology provides 112 bits of encryption strength.

RSA

Certificate 103 for RSA signature verification only

SHA-1

Certificate 343

RNG

Certificate 96

AES

Certificate 264

Triple DES

Certificate 346

HMAC

Certificate 76