LYNKS Series II Security Policy

TITLE: **LYNKS SERIES II SECURITY POLICY**	**REVISION** **D15** **DOCUMENT NO:** **550-160001-01**
CHECKED _____	**AUTHORIZED:** **JYOUNG** _____
DATE:	**DATE:** **05/08/2006**

LYNKS Series II Security Policy

Document No. 550-160001-01
Date: 4 May 2006

SPYRUS®
<info@spyrus.com>
<http://www.spyrus.com>

© Copyright by SPYRUS, Inc. 1998-2006. All Rights Reserved.

This document is provided only for informational purposes and is accurate as of the date of publication. This document may not be distributed for profit. It may be copied subject to the following conditions:

- All text must be copied without modification and all pages must be included.
- All copies must contain the SPYRUS copyright notices and any other notices provided herein.

Trademarks

SPYRUS, the SPYRUS logos, LYNKS Privacy Card, Security In A Box, SPEX/, SPYCOS, Multisession, Hydra Privacy Card, Hydra PC, Cryptocalculator, Talisman/DS, WebWallet, Rosetta, Signal Identity Manager, Personal Access Reader, Security to the Edge, Suite B On Board, Talisman/SAM, WEBREG, WEBSAFE, Terisa Systems, DeviceSSL, TLS Platinum and TLS Gold are either registered trademarks or trademarks of SPYRUS in the United States and/or other countries.

All other trademarks are the property of their respective owners.

Contents

Figures

1 Introduction

This Security Policy specifies the security rules under which the LYNKS Privacy Card® Series II, herein referred to as the LYNKS Series II or LYNKS Hardware Security Module (HSM), operates. Included in these rules are those derived from the security requirements of FIPS 140-2 and additionally, those imposed by SPYRUS, Inc. These rules, in total, define the interrelationship between the modules:

1. Operators
2. Services
3. Critical Security Parameters (CSPs)

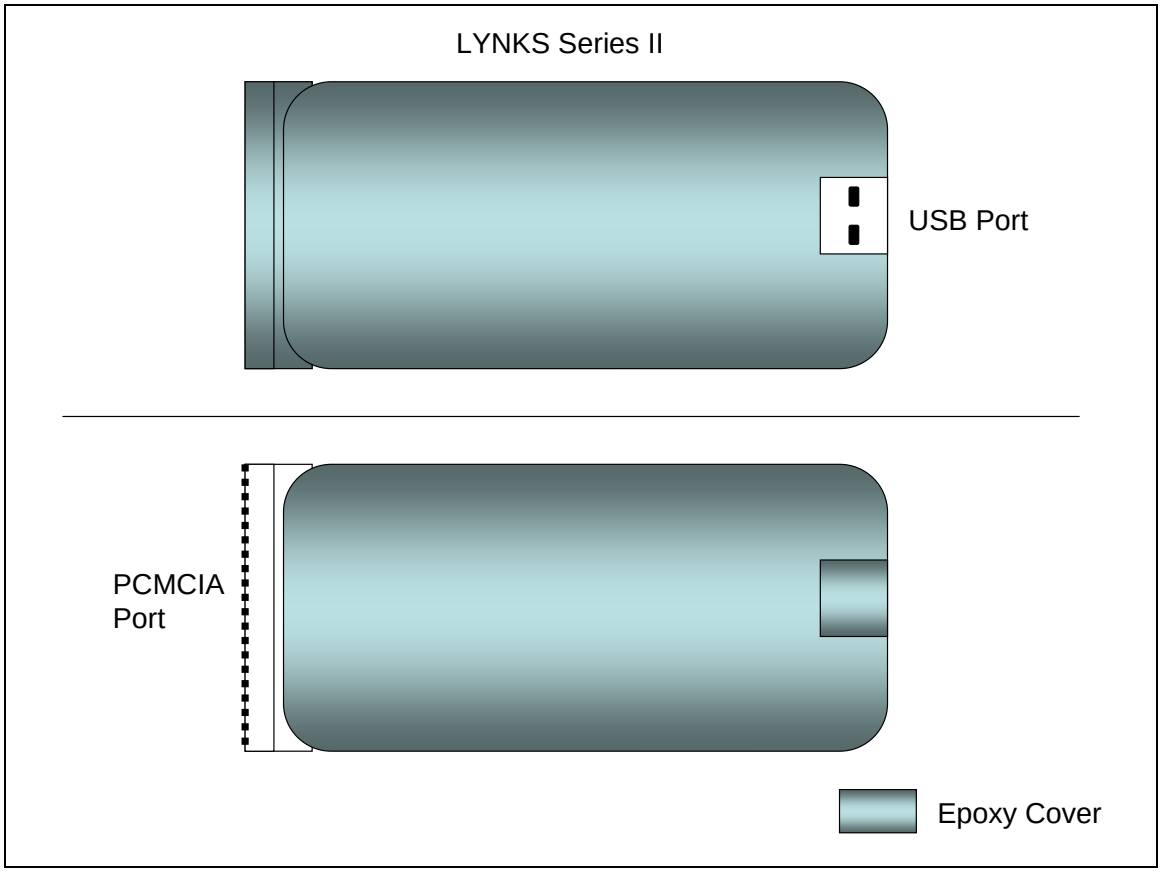

Figure 1-1 LYNKS Series II Interface types.

(c) 2006 SPYRUS, Inc. All rights reserved.

1.1 LYNKS Series II Overview

The LYNKS Series II (Figure 1-1) (model numbers and corresponding part numbers listed in Table 1-1 below, FW Version 2.2) is the latest addition to the SPYRUS family of LYNKS Privacy Card® cryptographic modules. LYNKS Privacy Cards provide high performance, high assurance cryptographic processing in personal, portable PC card and USB token form factors. LYNKS Privacy Cards are used within the U.S. Government in the Defense Message System (DMS) and also in commercial applications.

The LYNKS Series II enables security critical capabilities such as user authentication, message privacy and integrity, authentication, and secure storage in rugged, tamper-evident hardware. The LYNKS Series II communicates with a host computer via a PCMCIA 2.1 standard interface or USB interface.

The following is a summary of the ten configurations validated under FIPS 140-2 requirements for a Level 2 module:

Table 1-1
LYNKS Series II Product Configurations

Model Number	Part Number	Description	Interface	Algorithms	Features
PC500, PC600	906-160001-01 906-160002-01	LYNKS Series II PCMCIA	PCMCIA	AES,DES,TDES,SKIPJACK ECDSA,DSA RSA SHA-1,SHA-2 KEA	Command accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 20 key registers 50 Certificate locations
PC530	906-162001-01	Fortezza PCMCIA	PCMCIA	AES,DES,TDES,SKIPJACK ECDSA,DSA RSA SHA-1,SHA-2 KEA	Commands not accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 20 key registers 50 Certificate locations
PC530J	906-162002-01	Fortezza Jumbo PCMCIA	PCMCIA	SKIPJACK DSA SHA-1 KEA	Commands not accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 10 key registers 128 Certificate locations
PC530S	906-162004-01	Fortezza JumboMX	PCMCIA	SKIPJACK DSA	Commands not accessible prior to login

Model Number	Part Number	Description	Interface	Algorithms	Features
		PCMCIA		SHA-1 KEA	CI_getpersonalitylist CI_getcertificate CI_Setpersonality 10 key registers 400 Certificate locations
PC700, PC800	906-161001-01 906-161002-01	LYNKS Series II USB	USB	AES,DES,TDES,SKIPJACK ECDSA,DSA RSA SHA-1,SHA-2 KEA	Command accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 20 key registers 50 Certificate locations
PC730	906-162005-01	Fortezza USB	USB	AES,DES,TDES,SKIPJACK ECDSA,DSA RSA SHA-1,SHA-2 KEA	Commands not accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 20 key registers 50 Certificate locations
PC730J	906-162006-01	Fortezza Jumbo USB	USB	SKIPJACK DSA SHA-1 KEA	Commands not accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 10 key registers 128 Certificate locations
PC730S	906-162008-01	Fortezza JumboMX USB	USB	SKIPJACK DSA SHA-1 KEA	Commands not accessible prior to login CI_getpersonalitylist CI_getcertificate CI_Setpersonality 10 key registers 400 Certificate locations

1.2 LYNKS Series II HSM Implementation

The LYNKS Series II is implemented as a Multi-chip Stand-alone module as defined by FIPS 140-2.

The LYNKS Series II is available with either a PCMCIA 2.1 standard interface *or* a USB interface compliant to the *Universal Serial Bus Specification*, Revision 1.1,

(c) 2006 SPYRUS, Inc. All rights reserved.

dated 23 September 1998. Both Interfaces have been tested and are compliant with FIPS 140-2.

1.3 LYNKS Series II Cryptographic Boundary

The Cryptographic Boundary is defined to be the physical casing of the LYNKS Series II. The following two casings will be available and are chosen at the point of shipping:

- For the USB interface the cryptographic components are protected by a hard, opaque potting material compliant to the requirements of FIPS 140-2 Level 3. In addition to providing tamper evidence, the potting material forms a non-removable case which greatly enhances ruggedness and durability.
- For the PCMCIA form, the cryptographic components are protected by a hard, opaque potting material compliant to the requirements of FIPS 140-2 Level 3.

No hardware, firmware, or software components that comprise the LYNKS Series II are excluded from the requirements of FIPS 140-2.

1.4 Approved Modes of Operations

The LYNKS Series II supports an Approved mode of operation. The module operates in an Approved mode, except when the following services are invoked (in which case, the module assumes a non-Approved mode of operation):

- CIS_ GenRSAPublicPrivateSplitKeys
- RSA_Decrypt
- RSA_Encrypt
- CIS_SignRSASplitSignature
- RSA_CipherRAW

The LYNKS Series II modules support the following FIPS 140-2 approved algorithms:

Encryption & Decryption
Triple DES
AES
Skipjack
Digital Signatures
DSA, ECDSA, RSA
Hash
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
RNG
FIPS 186-2 RNG

The following services are available as “non-approved” algorithms:

Hash
MD5
Encryption & Decryption
RSA (Key Wrapping Only; key establishment methodology provides between 80-bits and 112-bits of encryption strength)
DES (non-approved; Legacy use)
Key Transport/Key Agreement
KEA
RNG
HW NDRNG (Myko Chip)

(c) 2006 SPYRUS, Inc. All rights reserved.

2 FIPS 140-2 Security Levels

The LYNKS Series II HSM cryptographic module complies with the requirements for FIPS 140-2 validation to the levels defined in Table 2.1. The FIPS 140-2 overall rating of the LYNKS Series II is Level 2.

Table 2-1
FIPS 140-2 Certification Levels

FIPS 140-2 Category	Level
1. Cryptographic Module Specification	2
2. Cryptographic Module Ports and Interfaces	2
3. Roles, Services, and Authentication	3
4. Finite State Model	2
5. Physical Security	3
6. Operational Environment	N/A
7. Cryptographic Key Management	2
8. EMI/EMC	3
9. Self-tests	2
10. Design Assurance	3
11. Mitigation of Other Attacks	N/A

3 Security Rules

The LYNKS Series II enforces the following security rules. These rules are separated into two categories: 1) those imposed by FIPS 140-2, and 2) those imposed by SPYRUS.

3.1 FIPS 140-2 Imposed Security Rules

1. The LYNKS Series II interfaces shall be logically distinct from each other.
2. The LYNKS Series II shall support the following four (4) interfaces:
   - data input
   - data output
   - control input
   - status output
   - power
3. The LYNKS Series II shall inhibit all data output via the data output interface whenever an error state exists and during self-tests.
4. The LYNKS Series II shall logically disconnect the output data path from the circuitry and processes performing the following key functions:
   - key generation and
   - key zeroization
5. The LYNKS Series II shall enforce Identity-Based authentication.
6. The LYNKS Series II shall re-authenticate an identity when it is powered-up after being powered-off.
7. The LYNKS Series II shall provide the following services:
   - Reference Table 4.1.
8. The LYNKS Series II shall contain production quality ICs with standard passivation.
9. The LYNKS Series II product shall be encapsulated within a hard, opaque potting material which shall form a non-removable enclosure such that attempts to remove or penetrate it shall have a high probability of causing serious damage to the LYNKS Series II.
10. The LYNKS Series II shall protect all CSPs from unauthorized disclosure, modification, and substitution:
11. The LYNKS Series II shall protect public keys against unauthorized modification and substitution.
12. The LYNKS Series II shall provide that:
    - a key entered into,
    - stored within, or
    - output from the LYNKS Series II is associated with the correct entities to which the key is assigned.
13. The LYNKS Series II shall provide the capability to zeroize all plaintext cryptographic keys and other unprotected critical security parameters within the LYNKS Series II.

(c) 2006 SPYRUS, Inc. All rights reserved.

14. The LYNKS Series II shall conform to the EMI/EMC requirements specified in FCC Part 15, Subpart B, Class B.
15. The LYNKS Series II shall perform the following self-tests, as applicable:
    - 1. Power up tests
    - * Cryptographic algorithm test
    - - TDES KAT
    - - AES KAT
    - - Skipjack KAT
    - - DSA Pairwise Consistency Test
    - - ECDSA Pairwise Consistency Test
    - - RSA Sign/Verify Pairwise Consistency Test
    - - RSA Encrypt/Decrypt KAT
    - - SHA-1 KAT
    - - SHA-2 KAT

    - * Random number generator test
    - - FIPS186-2 RNG KAT

    - * Software/firmware test
    - - SW Integrity Test, 32-bit CRC

    - * Critical functions test: N/A

    - 2. Conditional tests
    - * Pairwise consistency test
    - - DSA Pairwise Consistency Test
    - - RSA Sign/Verify Pairwise Consistency Test
    - - RSA Encrypt/Decrypt Pairwise Consistency Test
    - - ECDSA Pairwise Consistency Test

    - * Software/firmware load test
    - - DSA Signature Verification

    - * Continuous random number generator test
    - - FIPS 186-2 Continuous RNG Test
    - - NDRNG Continuous RNG Test

16. The LYNKS Series II shall enter an Error State and output an error indicator via the status interface whenever self-test is failed.
17. The LYNKS Series II shall not perform any cryptographic functions while in an Error State.
18. The power-up tests shall not require operator intervention in order to run and may be invoked on demand by power cycling the module.
19. The LYNKS Series II shall indicate that it is operational after all of the power-up tests have passed successfully.

20. The LYNKS Series II documentation shall include procedures for maintaining security while distributing and delivering the module.
21. The LYNKS Series II source code shall be annotated.
22. The LYNKS Series II software shall be implemented using a high-level language except that limited use of a low-level language is used to enhance the performance of the module.
23. The LYNKS Series II shall only permit the loading of trusted executable code that is DSA signed by SPYRUS, Inc.
24. The LYNKS Series II documentation shall provide Crypto Officer and User Guidance per FIPS 140-2, Section 4.10.4.

3.2 SPRYUS Imposed Security Rules

1. The LYNKS Series II shall not support multiple concurrent operators.
2. The LYNKS Series II shall not provide a maintenance role/interface.
3. The LYNKS Series II shall not support a bypass mode.
4. The LYNKS Series II shall not be required to mitigate other attacks including, but not limited to:
   - Power Analysis,
   - Timing Analysis,
   - Fault Induction, or
   - TEMPEST.

(c) 2006 SPYRUS, Inc. All rights reserved.

4 LYNKS Series II Roles and Services

4.1 Roles

The LYNKS Series II supports two roles, Crypto-officer (also called Site Security Officer (SSO)), and User, and enforces the separation of these roles by restricting the services available to each one.

Crypto-officer Role: The Crypto-officer is responsible for initializing the LYNKS Series II. Before issuing a LYNKS Series II to an end user, the Crypto-officer initializes the LYNKS Series II with private keying material and certificate information. The LYNKS Series II validates the Crypto-officer identity & PIN before accepting any initialization commands.

User Role: The User role is available after the LYNKS Series II has been loaded with a User personality & PIN.

The LYNKS Series II validates the User identity & PIN before access is granted. Each personality corresponds to a separate public/private key pair plus other information. The Crypto-officer may set up these User personalities during the initialization process.

4.2 Services

The following table describes the services provided by the LYNKS Series II.

Table 4.1
LYNKS Series II Services

Service	Description
Change PIN Phrase	Enables the SSO to change either the User PIN or SSO PIN. The SSO must provide the original and new PIN phrases. When the user PIN is successfully changed or the command fails, the SSO is automatically logged out.
Check PIN Phrase	Inputs a PIN Phrase to authenticate the SSO or the User.
CIS_ConcealKey	RSA wraps a specified Symmetric key.
CIS_GenRSAPublicPrivate	Used to create an RSA key pair. Currently, 1024- to 2048-bit modulus sizes are supported.
CIS GenRSAPublicPrivateSplitKeys	Generates a Boyd's RSA variation public/private keypair.

Service	Description
CIS_GenShroudedRSAKeypair	Generates an RSA public/private key pair. The private component is encrypted by a Triple DES symmetric key and returned. The symmetric key must exist in a key register prior to execution of the command.
CIS_GetCardOptions	Returns a bit mapped representation of algorithms and modes supported by the token. This function may be called at anytime, regardless of whether the card has been initialized, and regardless of whether a user is logged on or not. This service is not accessible in the Zeroized State and Power-up State.
CIS_GetCertificate	Returns the specifically requested certificate .
CIS_GetHash	Hashes the last or only block of data and returns the raw (unencoded) hash value. The hash algorithm used is either the default or the specified hashing algorithm set by *CIS_SetCurrentMode*. The default is SHA1.
CIS_GetPublicKey	Retrieves from a specified key index the public key value corresponding to the private key stored in the index. For DSA and KEA keys, the returned public key consists of the P,Q,G, and Y values. For RSA keys, the public key is returned in PKCS #1 encoded format.
CIS_LoadCertificate	Loads the specified certificate into the non-volatile memory of the Card.
CIS_LoadKey	Allows a user to load an AES, DES, or Triple-DES key into a key register on the Card. A Skipjack MEK cannot be loaded through this command.
CIS_LoadRSAPublicPrivate	Used to create an RSA public key from the private key data input.
CIS_RevealKey	Employs RSA to unwrap a Symmetric key using the select personalities private key.
CIS_RSAExtractPrivate	Typically used to back up an RSA private key or to load the same key into multiple Cards for the same functionality. This function wraps an RSA private key using the RSA public key of the recipient.
CIS_RSAInstallPrivate	Enables a recipient to act as a backup or alternate source for the private key that was extracted using the *CIS_RSAExtractPrivate* command.
CIS_SetCurrentMode	Sets the type of operation on the Card.
CIS_Sign	Computes an RSA signature in accord with PKCS#1 or a DSA signature in accord with FIPS PUB 186-2.
CIS_SignRSASplitSignature	Performs a Boyd's RSA variation signature.
CIS_Unwrap	Decrypts an AES-wrapped key into a key register using the AES algorithm.
CIS_VerifySignature	Validates a DSA or RSA signature against a locally

(c) 2006 SPYRUS, Inc. All rights reserved.

Service	Description
	generated hash value using the signer's public key (Y) in accord with FIPS PUB 186-2 or PKCS#1, respectively.
CIS_Wrap	Encrypts the selected key with the AES algorithm.
CISGetCurrentMode	Returns the current mode of all of the cryptographic services provided by the card. If a particular cryptographic mode has not been set via the SetMode command, this command will return the card default value
CISGetPublicKey	Retrieves from a specified key index the public key value corresponding to the private key stored in the index. For DSA and KEA keys, the returned public key consists of the P,Q,G, and Y values. For RSA keys, the public key is returned in PKCS #1 encoded format.
Decrypt	Supports symmetric cryptographic decryption modes.
Delete Certificate	Overwrites with nulls the data in an indexed storage location. The *Delete Certificate* command passes the certificate index to the Card; after the data is deleted, the corresponding certificate index flag is cleared.
Delete Key	Actively overwrites the key in the indexed key register with zeros. The *Delete Key* command passes the register index to the Card. Key Register 0 contains K_s; therefore, Key Register 0 is not valid for the *Delete Key* command.
EES_ECCSignCmd	Generates an ECDSA signature from the provided hash data and length. The returned signature is encoded using the SPYRUS Elliptic Curve Raw encoding
EES_ECCVerifyCmd	Verifies an ECDSA signature returned from the *ECCSignCmd* and from the provided hash data and length. The signature must be in the SPYRUS Elliptic curve raw format.
EES_GenECCKeypairCmd	Generates an elliptic curve public private key pair for a specified curve and returns the public in the SPYRUS raw encoded form.
Encrypt	Supports symmetric encryption modes.
Extract X	Archives a private X value, encrypted with Skipjack. Only X values loaded or generated by the SSO can be extracted.
Firmware Update	Updates the firmware in the Card. All firmware updates are signed by DSA to prevent the loading of untrusted code. The *destructive flag*, 0000 00FF, indicates to the Card that all non-volatile memory should be zeroized (including certificates) and the non-destructive flag, 0000 FF00, indicates that

Service	Description
	certificates should be maintained during the update.
Generate IV	Generates then writes the IV value into the crypto-logic.
Generate MEK	Uses the internal random number generator to produce a key for encrypting messages. The command results are stored in the key register index designated for future use. The MEK is not available in plaintext outside the Card. The MEK is available for use immediately after it is generated.
Generate Ra	Used with *Generate TEK* to support the public/private key exchange. The *Generate Ra* command generates a 1024-bit random value that is written to the Data-Out Block.
Generate Random Number	Generates a 160-bit random number and returns it in the Data-Out Block. This command can be executed before logging on to a Card.
Generate TEK	Supports public/private key exchanges. The command is used by both initiator and recipient after the key exchange. The *Generate TEK* command generates an 80-bit key based on the parameters passed with the Data-In Block. The key is stored in the key register index indicated.
Generate X	Allows the SSO or user to generate unique X & Y components before loading a certificate. The certificate index and the component type to be generated (KEA, DSA or both) are parameters passed in the Data-In Block.
Get Certificate	Returns the data associated with the specified certificate. Certificate data may be in any format because the Card does not read the certificate.
Get Hash	Returns the current hash value for a block(s) of hashed data. Depending on the mode SHA-1, SHA-256, SHA-384, SHA-512 or MD5 (non-approved) hashing algorithms are performed.
Get Personality List	Provides the host system with a list of stored personalities. The Card returns an entry for each certificate location, including Index 0. All the certificate names stored in non-volatile memory are sent to the host for the user to select and are 32 bytes in length.
Get Status	Allows the SSO or user to obtain the current status of the Card. Status information returned includes current state, serial number of the crypto-processor, mode, personality, key registers in use and certificate slots in use.
Get Time	Enables a user or SSO to retrieve the time from the on-board real-time clock.
Hash	Hashes the data provided with the command. It

(c) 2006 SPYRUS, Inc. All rights reserved.

Service	Description
	continues to hash using the current hash state as a starting point. The hash value may be reset by the *Initialize Hash* command or restored by the Restore command. The *Hash* command does not output the hash value. To obtain the hash value, use the *Get Hash* command.
Initialize Hash	Initializes the on-board hash function according to the NIST Secure Hash Standard.
Install X	Used by either the SSO or a user to: ▪ Restore an archived, private DSA or KEA *X* value ▪ Support remote rekey operations ▪ Duplicate personalities between cards
Load Certificate	Enables the SSO or a user to load user certificates for storage in a Card's non-volatile memory. The certificates loaded define the personalities available to the Card user. After a *Generate X* or *Load X* command has generated the certificate's unique X and Y values, the user's certificate is loaded in the index location containing the X value and associated with a user personality name.
Load DSA Parameters	Enables a user to load externally supplied p, q and g parameters for signature verification outside the domain of the currently selected personality.
Load Initialization Values	Enables the SSO to load the Card's initialization parameters—a 64-bit random seed value and a 80-bit user storage key, K_s.
Load IV	Writes the IV value into the crypto-logic. The command is used before an *Encrypt* or *Decrypt* command.
Load X	Enables the SSO or user to load the secret X value in the card during initialization. For the X value loaded, the command must specify the associated certificate and the mode (DSA or KEA). Input parameters are the following: Type of X: DSA, KEA or both; b) X value; c) X value's p, q and g parameters.
Relay	Used to restore an archived private X value, support remote rekey operations and duplicate personalities.
Restore	Renews the state of the cryptologic operation as specified by the Crypto Type input parameter. If the Data-In Block contains data, the Card uses the value of the input parameter as the input value. If the length of the restore data is zero, the Card uses the stored data (the result of a *Save* command).

Service	Description
Retrieve Sequencer Data	Retrieves the sequencer information from the token. This command will return the Sequencer tag and the Sequencer number (Big Endian format) for the index supplied as part of the data in block parameters. If a private key is generated using *CIS_GENRSAPublicPrivate* without the parameter of CIS_SIGNATURE_SEQ_TYPE this command will return an error.
RSA CipherRAW	RSA encrypt or decrypt data.
RSA_Decrypt	Performs RSA decryption
RSA_Encrypt	Performs RSA encryption.
RSATimestamp	Computes the rsa digital signature over the host provided hash value, Sequencer data, and date and time obtained from the on-board clock. This command uses SHA-1 Hashing only.
Save	Keeps the state of the cryptologic operation specified by the Crypto_Type input parameter.
Set Key	Loads the key value from the specified register index into the crypto-logic for encryption/decryption.
Set Mode	Sets encrypt, decrypt and hash command modes.
Set Personality	Enables a user to select a personality and can be executed any time during a session.
Set Time	Enables an SSO to set the current date and time for the on-board real-time clock; users are not allowed to change the clock.
Sign	Computes the digital signature values r and s over the host provided data. The data is signed with the private key associated with the certificate selected by the *Set Personality* command.
Timestamp	Computes the digital signature values r and s over the host provided hash value and the date and time obtained from the on-board clock.
Unwrap Key	Decrypts key data using the key indicated by the register index.
Verify Signature	Validates a digital signature that has been received.
Verify Timestamp	Validates a digitally signed date/time value that has been received.
Wrap Key	Uses the key indicated by the first register index to Skipjack encrypt the key indicated by the second register index. The result of the Wrap Key command is returned in the Card's Data-Out Block as a 96-bit value.
Zeroize	Clears the Card's data and internal buffers, key management information, and puts the Card in the zeroized state. Invoking this service and power cycling the module embody the zeroization service.

(c) 2006 SPYRUS, Inc. All rights reserved.

5 Identification and Authentication

5.1 Initialization Overview

All LYNKS HSMs are initialized at the factory with a Default SSO PIN Phrase. The SSO (Site Security Officer) must change the default value during logon to make the HSM ready for initialization. During initialization the HSM allows the execution of only the commands required to complete the initialization process.

Before a user can access or operate an HSM, the SSO must initialize it with operating parameters, the user ID and user certificates. The SSO is authorized to log on to the HSM any time after initialization to change parameters. The HSM allows 10 consecutive failed SSO logon attempts before it zeroizes all key material and initialization values. In the *zeroized* state, the SSO must use the Zeroize PIN Phrase to log on to the HSM and must reinitialize all HSM parameters.

A user must log on to a card to access any on-board cryptographic functions. To log on the user must provide the correct User PIN Phrase and select a *user personality*. The HSM allows 10 consecutive failed logon attempts before it deletes the stored User PIN Phrase. Certificates and other information stored in the HSM in non-volatile memory remain resident. After entering a correct PIN phrase, a user can select one of the stored user personality certificates for operations requiring DSA or KEA exchanges.

5.2 User Identity Authentication

User authentication is accomplished by PIN entry by the user. On invocation by the user, the LYNKS Series II waits for authentication of the user or SSO role by entry of a PIN phrase. Once a valid PIN phrase has been accepted the LYNKS Series II waits for the Personality to be set. Setting a Personality renders the LYNKS Series II ready for user commands.

The LYNKS Series II stores the number of logon attempts in non-volatile memory. The count is reset after every successful entry of a User PIN Phrase by a user and after every successful entry of the SSO PIN Phrase by the SSO. If the user fails to logon to the LYNKS Series II in 10 consecutive attempts, the LYNKS Series II will zeroize the User PIN and then transitions to a state that is initialized only for the SSO to perform restorative actions. To restore operation to the LYNKS Series II, the SSO Enabled User or SSO will have to reload the User PIN phrase. If the SSO Enabled User fails to logon to the LYNKS Series II in 10

consecutive attempts, the LYNKS Series II will zeroize all of the certificates, Private Components, Key Registers and disallow User access. When the LYNKS Series II is inserted after a zeroize, it will power up and transition to the Zeroized State, where it will only accept the Zeroize Default PIN phrase. After the Zeroize Default PIN phrase has been accepted, the LYNKS Series II transitions to the Uninitialized State and must be reinitialized, as described in section 5.1.

5.3 Strength of Authentication

The strength of the authentication mechanism conforms to the following specifications.

5.3.1 Single Random Attempt

For each attempt to use the authentication mechanism, the probability is less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur (e.g., guessing a password or PIN).

The mandatory minimal PIN size is 12 characters. If a random attempt to perform the CheckPIN command for either a SSO or user (i.e., login), the probability of success on that single attempt is equal to twice the reciprocal of the size of the PIN-space. This is due to the fact that there are normally two operant PINs that allow access at any time (unless the LYNKS Series II has not been initialized for the user, in which case only the SSO PIN is operable). Thus if there are N possible PINs available, the probability of success in entering the SSO or User PIN is $2 / N^{12}$. The character set available for PINs is at least all Alphanumeric characters (upper and lower cases) and 31 special keyboard characters comprising the set {~ ! @ # $ % ^ & * ( ) _ + - = { } [ ] | \ : ; ” ’ < , > . ? /}. This results in a PIN-space of $(26 + 26 + 10 + 31)^{12} = 93^{12} = 4.1859629748 \times 10^{23}$. The probability of a single successful random attempt is therefore $2.388936563 \times 10^{-24}$ to 10 significant figures, which is less than one in 1,000,000, or 1.0×10^{-6}.

5.3.2 Multiple Attempts

For multiple attempts to use the authentication mechanism during a one-minute period, the probability is less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.

The probability of success of a multiple entry attack in one minute is limited by the policy that no more than 10 failed attempts at login are permitted. Any more attempts would result in transition to the user uninitialized state, or in the case of the SSO, transition to the zeroized state. In either case, no further PIN entries could be successful.

(c) 2006 SPYRUS, Inc. All rights reserved.

To determine the probability of success of a multiple attack, the following observation is relevant: the attack is decomposed into ten mutually exclusive events that entail the success on the i^{th} try after i - 1 unsuccessful attempts. Thus the probability of ultimate success over a set of ten mutually exclusive and exhaustive events is the sum of the probabilities of the 10 attack events described.

For i ranging from 1 to 10, the i^{th} attack event has a probability equal to the probability P_i of i - 1 unsuccessful previous attacks multiplied by the probability of a successful attack on the i^{th} try. Thus $P_i = [(N – i)/N] \times [1 / (N – i)]$, or by cancellation, 1 / N, where N is the size of the PIN-space calculated in 5.3.1. The total probability of a multiple attack, given that either SSO or User PIN may be guessed, is then 2 x 10 / N, or 20/N. Thus, the probability of a multiple attack resulting in success (i.e., entering either the user or SSO PIN), is $4.777873125 \times 10^{-23}$ to 10 significant figures. This is less than one in 100,000, or 1.0×10^{-5}, as required.

5.3.3 Obscuration of Feedback

Feedback of authentication data to an operator is obscured during authentication (e.g., no visible display of characters result when entering a password). The PIN value is input to the CheckPIN command as a parameter by the calling application. No return code or pointer to a return value that contains the PIN is provided.

5.3.4 Non-weakening Effect of Feedback

Feedback provided to an operator during an attempted authentication shall not weaken the strength of the authentication mechanism. The only feedback provided by the CheckPIN command is a return code denoting success or failure of the operation. As is evident in the calculations of 5.3.1 and 5.3.2, this information in no way affects the probability of success or failure in either single or multiple attacks.

5.3.5 Generation of Random Numbers

The Generate Random Number command can be invoked either before authentication of the user or after the user has completed authentication as described above (see section 5.2). In the former case, i.e., prior to authentication, a random number is returned upon invocation to the calling application. The FIPS 186-2 (Change Notice version) algorithm is used for all unauthenticated calls. In the authenticated case, the same behavior and outcome occurs, but the output is generated by an embedded module.

The unauthenticated Generate Random Number command is isolated from the authenticated version, and its use prior to authentication is justified by the following:

- The result of the call does not reside within the cryptographic boundary of the LYNKS Series II;
- The call does not generate CSPs, or affect other CSPs within the cryptographic boundary of the LYNKS Series II or the next call of Generate Random Number;
- The states of the respective random number generators for unauthenticated calls and authenticated calls of Generate Random Number are independent and cannot influence each other due to the difference of algorithm and separation of all internal variables; and,
- Use of the generated random number is determined by the calling application and is subject to the security policy of the domain of the application, which is outside the control of the LYNKS Series II.

Non-user-invoked internal RNG calls are serviced by a FIPS186-2 RNG and hardware RNG. The hardware RNG is used for seeding other RNGs.

(c) 2006 SPYRUS, Inc. All rights reserved.

6 Access Control

6.1 Critical Security Parameters (CSPs)

CSP Designation	Algorithm(s) / Standards	Symbolic Form	Description
SSO PIN Phrase	N/A	PIN	A secret 12 byte value used for SSO authentication.
KEA Private Components	KEA	rA, rB	Employed during the key exchange operation.
Message Encryption Key (MEK)	SKIPJACK, AES, DES, TDES	MEK	Employed for data encryption or wrapping of keys.
RNG Key	FIPS 186-2	XKEY	Used to seed the FIPS 186-2 RNG.
DSA Private Key	DSA	X	The Private Key of the User employed in digital signing operations.
ECDSA Private key	ECDSA X9.62	d_{ECDSA}	The Private Key of the User employed in Elliptic Curve digital signing operations.
RSA Private key	RSA X9.31	d_{RSA}	The Private Key of the User employed in RSA digital signing operations.
Storage Key	SKIPJACK	K_S	Used to encrypt all symmetric keys stored in FLASH
Token Encryption Key (TEK)	SKIPJACK	TEK	Token Encryption Key used for key encryption
User PIN Phrase	N/A	PIN	A secret 12 byte value used for user authentication.

Table 6.1 Critical Security Parameters
LYNKS Series II CSPs

6.2 Other Key Management Parameters

Key Management Parameter	Algorithm(s) / Standards	Symbolic Form	Description
DSA Public Key	DSA	Y_{DSA}	The Public Key of the user employed in digital signature verification operations.
ECDSA Public key	ECDSA X9.62	Q	The Public Key of the user employed in Elliptic Curve digital verification operations.
SPYRUS DSA Public Key	DSA X9.62	$Y_{FWupdate}$	A hard-coded Public Key employed in Firmware Update operations. Stored in plaintext.
KEA Public Components	KEA	RA, RB	Employed during the key exchange operation.
RSA Public Key	RSA ANSI X9.31	Y_{RSA}	The Public Key of the user employed in digital signature verification operations.

Table 6.2
Other LYNKS Series II Key Management Parameters

6.3 CSP Access Type

Table 6.3
LYNKS Series II Access Types

Access Type	Description
Generate (G)	"Generate" is defined as the creation of a CSP
Delete (D)	"Delete" is defined as the zeroization of a CSP
Use (U)	"Use" is defined as the process in which a CSP is employed. This can be in the form of loading, encryption, decryption, signature verification, or key wrapping.

(c) 2006 SPYRUS, Inc. All rights reserved.

6.4 Access Matrix

The following table shows the services (see section 4.2) of the LYNKS Series II, the roles (see section 4.1) capable of performing the service, the CSPs (see section 6.1) that are accessed by the service and the mode of access (see section 6.3) required for each CSP.

Table 6.4
LYNKS Series II Access Matrix

Service Name	Role		Access	
	SSO	User	CSPs	Access Mode
Change PIN Phrase	X		PIN	G
Check PIN Phrase			PIN	U
Cis_ConcealKey		X	KS, MEK	U, U
CIS_GenRSAPublicPrivate		X	d_{RSA}	G
CIS_GenRSAPublicPrivateSplitKeys		X		
CIS_GenShroudedRSAKeypair		X	d_{RSA}, MEK, KS	G, U
CIS_GetCardOptions				U
CIS_GetCertificate				U
CIS_GetHash		X		U
CIS_GetPublicKey		X		
CIS_LoadCertificate	X	X		U
CIS_LoadKey		X	MEK, KS	U
CIS_LoadRSAPublicPrivate		X	d_{RSA}	U
CIS_RevealKey		X	KS, MEK	U, U
CIS_RSAExtractPrivate	X	X	$d_{RSA,}$ MEK	U, U
CIS_RSAInstallPrivate	X	X	$d_{RSA,}$ MEK	U
CIS_SetCurrentMode		X		U
CIS_Sign		X	$d_{RSA,}$ X	U
CIS_SignRSASplitSignature		X		
Cis_Unwrap		X	MEK, KS	U, U
CIS_VerifySignature		X		U
Cis_Wrap		X	MEK, KS	U, U
CisGetCurrentMode		X		
CisGetPublicKey		X		
Decrypt		X	MEK, KS	U
Delete Certificate	X	X		D
Delete Key		X		D

Service Name	Role		Access	
	SSO	User	CSPs	Access Mode
Encrypt		X	MEK, KS	U
EES_GenECCKeypairCmd		X	d_{ECDSA}	G
EES_ECCSignCmd		X	d_{ECDSA}	U
EES_ECCVerifyCmd		X		U
Extract X	X		X, TEK	U
Firmware Update	X	X		G
Generate IV		X		G
Generate MEK		X	MEK	G
Generate Ra		X	rA	G
Generate Random Number	X	X	KEA, MEK, X, d_{RSA}, d_{ECDSA}	G
Generate Random Number			XKEY	U
Generate TEK		X	TEK	G
Generate X	X	X	X	G
Get Certificate	X	X		U
Get Hash		X		G
Get Personality List				U
Get Status				U
Get Time				U
Hash		X		G
Initialize Hash		X		G
Install X	X	X	X, TEK	U
Load Certificate	X	X		U
Load DSA Parameters		X		U
Load Initialization Values	X		Ks	G
Load IV		X		U
Load X	X	X	X	U
Relay	X	X	X, TEK	U
Restore		X		U
Retrieve Sequencer Data	X	X		
RSA_CipherRAW		X		
RSA_Decrypt		X		
RSA_Encrypt		X		
RSA Timestamp		X	d_{RSA}	U
Save		X		U
Set Key		X	MEK, KS	U
Set Mode		X		U
Set Personality	X	X		U
Set Time	X			U

(c) 2006 SPYRUS, Inc. All rights reserved.

Service Name	Role		Access	
	SSO	User	CSPs	Access Mode
Sign		X	X	U
Timestamp		X		U
Unwrap Key		X	TEK, MEK	U
Verify Signature		X		U
Verify Timestamp		X		U
Wrap Key		X	TEK, MEK	U
Zeroize			All CSPs	D