LYNKS Privacy Card

Security Policy

LYNKS Privacy Card

Security Policy

Document No. 512-020000-01-A3
Revised November 9, 1999

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

Trademarks

Contents

1 Introduction

The SPYRUS family of LYNKS Privacy Card tokens provide high performance, high assurance cryptographic processing in a personal, portable PC card form factor. The LYNKS Privacy Card product enables security critical capabilities such as user authentication, message privacy and integrity, authentication, and secure storage in rugged, tamper-evident hardware. The device is used within the U.S. Government with the Defense Messaging System (DMS) and also in commercial applications.

The LYNKS Privacy Card implements the following cryptographic algorithms: Digital Signature Algorithm (DSA) FIPS PUB 186, Secure Hash Algorithm (SHA-1) FIPS PUB 180-1, Key Exchange algorithm (KEA), Skipjack, Data Encryption Standard (DES) and Triple Data Encryption Standard (Triple DES). The LYNKS Privacy Card communicates with a host computer via a PCMCIA 2.1 standard interface.

The remainder of this document describes the security policy for the LYNKS Privacy Card.

2 Product Overview

The LYNKS Privacy Card is designed to be a general purpose cryptographic module that executes U.S. domestic algorithms.

The Data Encryption Algorithm (DEA) in its ECB, CBC, and CFB modes meets the specifications of NIST FIPS PUB 46-2 (Data Encryption Standard (DES)). The LYNKS Privacy Card also supports Triple DES in the following modes: ECB, CBC, and CFB. The Rivest-Shamir-Adleman (RSA) digital signature, RSA key wrap, Diffie-Hellman (D-H) key agreement, and MD5 hashing algorithms are specified in the RSA Public Key Cryptosystem Standards (PKCS). D-H X9.42 is a variant of the classic D-H algorithm developed by the American National Standards Institute (ANSI) for use with financial services systems. The LYNKS Privacy Card also implements the FORTEZZA suite of algorithms (KEA, Skipjack, SHA-1 and DSA) using an embedded Capstone chip to ensure interoperability with government systems and the systems that communicate with them.

Table 1 summarizes the cryptographic algorithms on the LYNKS Privacy Card.

Key Transport	Key Wrap	Encryption/ Decryption	Hashing	Signatures
KEA	Skipjack	Skipjack	SHA-1	DSA
D-H	DES	DES	MD5	RSA
D-H X9.42	Triple DES	Triple DES		

Table 1. Algorithms Supported by the LYNKS Privacy Card

The LYNKS Privacy Card generally follows the FORTEZZA model for PCMCIA card configuration, initialization, user and SSO logons, state transitions, and hardware interface specifications.

2.1 Command Description

This section describes the cryptographic capabilities on the LYNKS Privacy Card provided through a set of primitive cryptographic functions.

The CI_ functions denote common functions that are FORTEZZA-compliant. The CIS_ functions are used specifically to access commercial algorithms and the Algorithm Agile™ card management functions. Several CI_ functions have CIS_ counterparts that offer extended functionality for the commercial algorithm variants. The services covered by the set of CI_ and CIS_ commands include encryption, decryption, hashing, key transport, digital signature, and key management.

2.1.1 FORTEZZA Algorithm Commands

CI_ChangePIN*	CI_GetCertificate	CI_Restore
CI_CheckPIN	CI_GetHash	CI_Save
CI_Decrypt	CI_GetPersonalityList	CI_SetKey
CI_DeleteCertificate	CI_GetTime	CI_SetMode
CI_DeleteKey	CI_Hash	CI_SetPersonality
CI_Encrypt	CI_InitializeHash	CI_SetTime*
CI_ExtractX*	CI_InstallX	CI_Sign
CI_GenerateIV	CI_LoadCertificate	CI_TimeStamp
CI_GenerateMEK	CI_LoadDSAParameters	CI_UnwrapKey
CI_GenerateRa	CI_LoadInitValues*	CI_VerifySignature
CI_GenerateRandom	CI_LoadIV	CI_VerifyTimeStamp
CI_GenerateTEK	CI_LoadX	CI_WrapKey
CI_GenerateX	CI_RelayX	CI_Zeroize
CI_FirmwareUpdate	CI_GetStatus	

2.1.2 Commercial Algorithm Commands

CIS_ConcealKey	CIS_GetcardOptions	CIS_RSAInstallPrivate
CIS_GenerateDHPublicPrivate	CIS_LoadCertificate	CIS_VerifySignature
CIS_GenerateDHTEK	CIS_LoadKey	CIS_SetCurrentMode
CIS_GenerateRSAPublicPrivate	CIS_LoadRSAPublicPrivate	CIS_Sign
CIS_GenerateRSASplit	CIS_LoadDHPublicPrivate	CIS_SignRSASplit
CIS_GetCertificate	CIS_RSAExtractPrivate	CIS_GetHash
CIS_OAEP_Decrypt	CIS_RevealKey	CIS_OAEP_Encrypt

*Available only when logged on as Crypto Officer.

3 Roles

The LYNKS Privacy Card supports two roles: Crypto-Officer and User, and enforces the separation of these roles by restricting the services available to each one. The role of the individual is determined by access control on the card prior to performing any services.

3.1 Crypto-Officer Role

The Crypto-Officer is responsible for initializing the LYNKS Privacy Card. The Crypto-Officer role is available during card initialization. Card initialization is typically performed on a Certificate Authority Workstation (CAW) that is secured according to the site security policy of the deploying organization. The Crypto-Officer has access to all services on the card. Before issuing a card to an end user, the Crypto-Officer initializes the card with private keying material, certificate information, and when needed, new firmware.

The LYNKS Privacy Card validates the Crypto-Officer role by requiring access using a Personal Identification Number (PIN). A valid PIN must be passed to the LYNKS Privacy Card before it will accept any initialization commands.

An uninitialized card from the factory contains a default PIN phrase, which is first used by the Crypto-Officer to load initialization values into the card. The card then transitions to an intermediate initialization state and the Crypto-Officer must change the PIN phrase for the Crypto-Officer role. Next the Crypto-Officer can set the time on the card, generate public/private key pairs, load the point-of-trust certificate of the certificate hierarchy into the card, load the user's certificates into the card, and load the user's certificate authority's certificates into the card. The Crypto-Officer is responsible for obtaining all certificate material in accord with the security policy of the organization.

Lastly, the Crypto-Officer must set the access control PIN phrase for the user. Only the Crypto-Officer may change a PIN phrase.

3.2 User Role

After the LYNKS Privacy Card has been loaded with a User PIN, the User role is available and the remainder of the cryptographic functions are enabled, with the exception of the following:

- Change PIN Phrase
- Extract X
- Load Initialization Values
- Set Time

The exclusion of these commands provides the mechanism for separation of the Crypto-Officer and User roles. The LYNKS Privacy Card validates the User role by requiring a Personal Identification Number (PIN) in order to access it. When the User has been successfully authenticated, he or she can then choose one of the available personalities on the card. Each personality corresponds to a separate public/private key pair plus other information. The Crypto-Officer sets up these user personalities during the initialization process.

4 Services

This section describes the cryptographic services provided by the Lynks Privacy Card.

4.1 Self-test

The LYNKS Privacy Card performs a self-test immediately upon power-up to ensure its integrity. Cryptographic and firmware checks are performed to ensure that the device is operating properly prior to communicating with the host computer. Any failures will cause the LYNKS Privacy Card to go into a non-operational error state. No authentication of the Crypto-Officer or user is required for the self-test.

4.2 Firmware Update

Prior to any cryptographic processing, the LYNKS Privacy Card must be loaded with SPYRUS developed firmware. Firmware should be loaded only at a facility designated by SPYRUS and in accord with the site security policy of the user's organization. After the firmware is loaded, the LYNKS Privacy Card is zeroized and ready for initialization. The firmware, which is digitally signed by SPYRUS, will be loaded into the LYNKS Privacy Card only if its signature can be verified. Only FIPS certified versions of the firmware should be loaded into the LYNKS Privacy Card. All firmware modifications require FIPS 140-1 re-validation of the LYNKS Privacy Card.

The LYNKS Privacy Card supports Firmware updates with the following command: CI_FirmwareUpdate

4.3 Initialization

The Crypto-Officer initializes the LYNKS Privacy Card prior to transferring it to a user. Initialization consists of the following:

- Authenticate the Crypto-Officer based on PIN input
- Load the Initialization parameters
- Load an X.509 certificate into a Trusted Certificate space
- Generate public/private key pairs for the user
- Obtain and load X.509 certificates for the user
- Specify a user PIN

Upon completion of the Initialization process, the LYNKS Privacy Card provides encryption/decryption, hash, key transport, digital signature, and key management cryptographic services.

The LYNKS Privacy Card supports initialization with the following commands: CI_ChangePin, CI_CheckPin, CI_LoadInitValues, CI_Zeroize, CI_LoadCertificate, CI_GetCertificate, CI_GenerateX, CI_SetTime, CI_GetTime, CI_LoadDSAParameters

4.3.1 Encryption/Decryption

Symmetric cryptography uses one key to encrypt and decrypt data. The originator must ensure that the recipient has the key for decryption. This key must remain secure between the intended parties or the security of the message will be compromised. Successful encryption/decryption transformations require 100 percent bit integrity of all encryption and decryption functions and data. In some applications, this feature of encryption may be leveraged to provide an integrity check on the transmitted data.

The LYNKS Privacy Card supports symmetric encryption and decryption with the following commands: CI_Decrypt, CI_Encrypt, CI_GenerateIV, CI_LoadIV, CI_GenerateMEK, CI_Save, CI_SetMode, CI_DeleteKey, CI_SetKey, CI_Restore, CI_SetCurrentMode, CIS_LoadKey, CIS_OAEP_Encrypt, CIS_OAEP_Decrypt.

4.3.2 Key Transport

The LYNKS Privacy Card key exchange process uses public key cryptography to encrypt (*wrap*) the key used in the encryption algorithm. The wrapped key can then be securely transmitted to the recipient. Key exchange allows only the intended recipient to unwrap the needed key to decrypt data.

The LYNKS Privacy Card supports asymmetric key exchange for common key negotiation using the following commands: CIS_DHGeneratePublicPrivate, CIS_DHLoadPublicPrivate, CI_GenerateRa,CI_GenerateTEK, and CIS_DHGenerateTEK.

4.3.3 Hash

The hash function provides a check for data integrity. The card performs a mathematical hash function on the data, producing a 160-bit (20-byte) hash value. This hash value is unique for every message because any change in the data, even a single bit, changes the hash value. An important property of hashing is that data cannot be reconstructed from the hash value; hashing is a *one-way* function. This one-way property is key to generating unique digital signatures that cannot be forged.

The LYNKS Privacy Card supports hashing using the SHA-1 and MD5 algorithms and the following commands: CI_InitializeHash, CI_Hash, CI_GetHash , CIS_GetHash, CI_Save CI_SetMode, and CI_Restore.

4.3.4 Digital Signature

A digital signature allows a message originator to sign data (typically the hash value) and provides a recipient with the means to verify the originator's identity (user authentication and non-repudiation). Any change in the data causes a change in the hash value. A recipient can then use the cryptography to verify the originator's digital signature over the hash value to verify the integrity of the data.

Digital signatures do not encrypt, transform, or otherwise alter data, so sensitive data is usually signed first and encrypted second. The signature may or may not be encrypted.

Alternately (but not routinely implemented), a digital signature can be used as an integrity check value (ICV), wherein the data is encrypted before signing.

The LYNKS Privacy Card supports digital signature with the following commands: CI_GenerateX, CI_LoadX, CI_SetPersonality, CIS_RSAGeneratePublicPrivate, CIS_RSALoadPublicPrivate, CI_Sign, CI_VerifySignature, CIS_Sign CIS_VerifySignature, CIS_GenerateRSASplit, CIS_SignRSASplit, CIS_LoadCertificate, CIS_GetCertificate..

4.3.5 Key Management

The primary key management responsibility of the LYNKS Privacy Card is the administration of the DSA, KEA, RSA, and D-H public/private key pairs. The key management concept employed in the LYNKS Privacy Card provides a robust, yet practical solution. The card provides key generation and key archival functions and allows a user to perform revocation, expiration, notification, and authentication functions. The principal component in the key management architecture is the user's certificate.

The LYNKS Privacy Card supports management of symmetric and asymmetric keys using the following commands: CI_ExtractX, CI_InstallX, CI_RelayX, CI_GenerateX, CI_LoadX, CI_SetPersonality, CIS_RSAExtractPrivate, CIS_RSAInstallPrivate, CIS_ConcealKey, CIS_RevealKey, CI_Wrap, and CI_Unwrap.

4.4 Matrix of Cryptographic Functions

Table 2 summarizes the services performed by the LYNKS Privacy Card, including the roles for which the service is available and whether the service performs cryptographic functions.

	Roles		Cryptographic Functions	
Services	**Crypto-Officer**	**User**	**Yes**	**No**
Self-test	X	X	X	
CI Change PIN Phrase	X		X	
CI Check PIN Phrase	X	X	X	
CI Decrypt	X	X	X	
CI Delete Certificate	X	X		X
CI Delete Key	X	X		X
CI Encrypt	X	X	X	
CI Extract X	X		X	
CI Firmware Update	X	X	X	
CI Generate IV	X	X	X	
CI Generate Mek	X	X	X	
CI Generate Ra	X	X	X	
CI Generate Random Number	X	X		X
CI Generate TEK	X	X	X	
CI Generate X	X	X	X	
CI Get Certificate	X	X		X
CI Get Hash	X	X	X	
CI Get Personality List	X	X		X
CI Get Status	X	X		X
CI Get Time	X	X		X
CI Hash	X	X	X	
CI Initialize Hash	X	X	X	
CI Install X	X	X	X	
CI Load Certificate	X	X		X
CI Load DSA Parameters	X	X	X	
CI Load Initialization Values	X		X	
CI Load Iv	X	X	X	
CI Load X	X	X	X	
CI Relay	X	X	X	
CI Restore	X	X	X	
CI Save	X	X	X	
CI Set Key	X	X	X	
CI Set Mode	X	X	X	
CI Set Personality	X	X	X	
CI Set Time	X			X
CI Sign	X	X	X	
CI Timestamp	X	X	X	

CI Unwrap Key	X	X	X	
CI Verify Signature	X	X	X	
CI Verify Timestamp	X	X	X	
CI Wrap Key	X	X	X	
CI Zeroize	X	X	X	
CIS Conceal Key	X	X	X	
CIS Generate DH Public/Private	X	X	X	
CIS Generate DH TEK	X	X	X	
CIS Generate RSA Public/Private	X	X	X	
CIS Generate RSA Split Key	X	X	X	
CIS Get Certificate	X	X		X
CIS OAEP Decrypt		X	X	
CIS Get Hash	X	X	X	
CIS Get Card Options	X	X		X
CIS Load Certificate	X	X		X
CIS Load Key		X		X
CIS Load RSA Public/Private	X	X	X	
CIS Load DH Public/Private	X	X	X	
CIS RSA Extract Private	X	X	X	
CIS Reveal Key	X	X	X	
CIS RSA Install Private	X	X	X	
CIS Verify Signature	X	X	X	
CIS Set Current Mode	X	X	X	
CIS Sign	X	X	X	
CIS Sign RSA Split Key	X	X	X	
CIS OAEP Encrypt		X	X	

Table 2. Matrix of Services, Roles, and Cryptographic Functions

5 Key Lifecycle

This section describes the lifecycle of the keys that are generated, stored, and used by the LYNKS Privacy Card.

5.1 KFEK

Each LYNKS Privacy card contains a Key File Encryption Key (KFEK) unique to that card. This key is an internally generated and maintained value and is not available at the card interface. The KFEK is used to wrap/cover and unwrap/uncover working values within the card's internal memory space.

5.2 Skipjack Keys

Skipjack is a symmetric encryption algorithm. Two types of keys are used to support Skipjack operation, Message Encryption Keys (MEK) and Token Encryption Keys (TEK). MEKs are used in bulk encryption operations, for example via the Encrypt and Decrypt commands. TEKs are used to wrap (encrypt) MEKs. The following describes the life-cycles of MEKs and TEKs.

5.3 MEK

An MEK is created using the GenerateMEK command. The loading of skipjack MEKs is prohibited on the LYNKS Privacy Card. This MEK is a random value, stored in one of the ten internal key registers. Once an MEK is loaded onto the card, it's plaintext value is never exposed. An MEK is deleted using the DeleteKey command. When an MEK is required to be offloaded from the card, it must first be wrapped using the WrapKey command. A wrapped MEK is loaded back onto the card using the UnwrapKey command. When an MEK is wrapped, it is encrypted with TEK, protecting the contents of the original MEK. When power is removed from the card, the contents of any MEK stored in one of the ten internal key registers is lost.

5.4 TEK

A TEK is created as a result of a key exchange (KEA is the key exchange algorithm used for generating Skipjack TEKs). A TEK is stored in one of the ten internal key registers. Once a TEK is generated in the card, its plaintext value is never exposed. Multiple TEKs may be stored simultaneously internally in the card. A TEK may never be extracted from the card, in any form. When power is removed from the card, the contents of any TEK stored in one of the ten internal key registers is lost. Skipjack TEKs may be used only to wrap skipjack MEKs.

5.4.1 Ks

The card storage key (Ks) is a special case of a TEK. Ks occupies key slot 0 and may only be loaded by the SSO. Ks may never be exported off of the card in any form.

5.5 DES Keys

DES is a symmetric encryption algorithm. Unlike Skipjack keys, only one type of DES key exists, thus any DES key may wrap any other DES key. Also unlike Skipjack keys,

DES keys may be loaded into the card in their plaintext form, using the LoadKey command. Of course, DES keys may be randomly generated using the GenerateMEK command. The WrapKey and UnwrapKey commands are used to wrap/unwrap DES keys for passage out of and back onto the card. DES keys may be wrapped only with other DES keys. A DES key resides in one of the ten internal key registers. Once a DES key is loaded onto the card, it's plaintext value is never exposed. When power is removed from the card, the contents of any DES key stored in one of the ten internal key registers is lost.

5.6 KEA/DSA Keys

KEA is an asymmetric algorithm used for key exchange and the generate of TEKs. DSA is an asymmetric algorithm used for the generation and verification of digital signatures. As asymmetric keys, DSA and KEA keys consist of 2 discrete but related parts, the private (X) value and type public (Y) value. KEA/DSA private keys maybe be randomly generated on the card using the GenerateX function or may be loaded directly onto the card in their plaintext form using the LoadX command. Once a KEA/DSA private key is loaded onto the card, it's plaintext value is never exposed. When an KEA/DSA private key is required to be extracted from the card, it must first be wrapped using the ExtractX command. An extracted private key is restored to the card using the InstallX command. When a private key is extracted, it is encrypted with TEK and a password, protecting the contents of the original private key. When power is removed from the card, the contents of any private key stored in one of the personality registers is maintained. These private keys are deleted using the Zeroize command.

5.7 Diffie-Hellman Keys

Diffie-Hellman (DH) is an asymmetric algorithm used for key exchange and the generation of TEKs. As asymmetric keys, DH keys consist of 2 discrete but related parts, the private (X) value and type public (Y) value. DH private keys maybe be randomly generated on the card using the GenerateDHPrivate function or may be loaded directly onto the card in their plaintext form using the LoadDHPrivate command. Once a DH private key is loaded onto the card, it's plaintext value is never exposed. When a DH private key is required to be extracted from the card, it must first be wrapped using the ExtractDHPrivate command. An extracted private key is restored to the card using the InstallDHPrivate command. When a private key is extracted, it is encrypted with a TEK and a password, protecting the contents of the original private key. When power is removed from the card, the contents of any DH private key stored in one of the personality registers is preserved. These private keys are deleted using the Zeroize command.

5.8 RSA Keys

RSA is an asymmetric algorithm used for both key exchange and the generation and verification of digital signatures. As asymmetric keys, RSA keys consist of 2 discrete but related parts, the private (X) value and type public (Y) value. RSA private keys maybe be randomly generated on the card using the GenerateRSAPublicPrivate function or may be loaded directly onto the card in their plaintext form using the LoadRSAPublicPrivate command. Once an RSA private key is loaded onto the card, it's plaintext value is never exposed. When an RSA private key is required to be extracted from the card, it must first be wrapped using the ExtractRSAPrivate command. An extracted private key is restored to the card using the InstallRSACommand command. When a private key is extracted, it is encrypted with another RSA key and a password, protecting the contents of the original private key. When power is removed from the card, the contents of any private key stored in one of the personality registers is maintained. These private keys are deleted using the Zeroize command

5.9 RSA Rules of Operation

The use of RSA Algorithm will place the mode of operation in a non-FIPS mode.

6 Physical Security

The LYNKS Privacy Card is easily portable and can be carried readily in a pocket or a Portable Computer (ie Laptop). However the User should be aware that the loss of the LYNKS Privacy Card can place the information that is protected at risk, or could enable an unauthorized person to imitate the legitimate user. While the LYNKS Privacy Card employs a Personal Identification Number (PIN) to prevent use of the card by unauthorized users, no PIN – based system us absolutely foolproof. A hostile entity, which obtains your card or other implementation, could possible extract the PIN or User Certificates, and then use the LYNKS Privacy Card to decrypt information protected by that individual card.
The LYNKS Privacy Card is physically packaged in a PCMCIA Type 2 Standard. This packaging allows for Tamper evident but is not Tamper proof. If any evidence of physical tampering or if the card is lost the User should report at once to the cognizant Security Officer or Certificate Authority.

7 Revision History

REV. #	DATE	DESCRIPTION
1.0	3/11/99	Initial Release
A1	4/13/99	Revised
A2	9/27/99	Revised
A3	11/9/1999	Revised