Rosetta USB

Model Number: USB110-GBL

Security Policy, Level 3

Version 1.4

Date: 20 June 2001

SPYRUS, Inc.
2355 Oakland Road
San Jose, CA 95131

Revision History

REV. #	DATE	DESCRIPTION
1.0	17 January 2001	Original (Firmware Version FUP03)
1.1	2 March 2001	Add in Load Key for Zeroize
1.2	8 March 2001	Edit Security Rules
1.3	18 April 2001	Edit Security Rules
1.4	20 June 2001	Addition of Model Number to Document

Contents

1 Introduction

1.1 Scope

This Security Policy specifies the security rules under which the Rosetta USB Cryptographic Module, herein identified as the Rosetta USB, must operate. Included in these rules are those derived from the security requirements of FIPS 140-1 and additionally, those imposed by SPYRUS, Inc. These rules, in total, define the interrelationship between the:

1. module operators,
2. module services, and
3. security related data items (SRDIs).

1.2 Rosetta USB Overview

The Rosetta USB (Figure 1-1) is a low cost cryptographic processor that can be used in place of a standard ISO7816 smartcard token. The Rosetta USB (Model Number USB110-GBL) provides the following features:

- Blue Over-molding (Visible LEDs)
- RSA Key Generation (non-deletable)
- FIPS 140-1, Level 3 Certification

The Rosetta USB is contained in a tamper resistant case and incorporates microprocessors, Ram, and EEPROM technologies. Communication is accomplished over a standard USB interface to a host computer running at 12M bits/s. The intended use of this product is to provide Rosetta Smartcard capabilities in an environment where existing SPYRUS smartcard technologies could not be deployed.

Figure 1-1
Rosetta USB

1.3 Rosetta USB Implementation

The Rosetta USB is implemented as a Multi-chip Stand-alone module as defined by FIPS 140-1.

1.4 Rosetta USB Cryptographic Boundary

The Cryptographic Boundary is defined to be the edge of the Rosetta USB. The cryptographic components are protected by a hard, opaque material compliant to the requirements of FIPS 140-1, Level 3 and over-molded with the USB case. The over-molding forms a non-removable case. Non-cryptographic H/W components are excluded from the requirements of FIPS 140-1. See Table 4.1 for F/W components that are excluded from the requirements of FIPS 140-1.

2 FIPS 140-1 Security Levels

The Rosetta USB cryptographic module passes FIPS 140-1 certification to the levels defined in Table 2.1. The FIPS 140-1 overall rating of the Rosetta USB is Level 3.

Table 2.1
Rosetta USB Security Levels

FIPS 140-1 Security Requirements Section	Level
1. Cryptographic Module	3
2. Module Interfaces	3
3. Roles and Services	3
4. Finite State Machine Model	3
5. Physical Security	3
6. Software Security	3
7. Operating System Security	N/A
8. Key Management	3
9. Cryptographic Algorithms	3
10. EMI / EMC	3
11. Self Tests	3

3 Security Rules

The Rosetta USB enforces the following security rules. These rules are separated into two categories, 1) those imposed by FIPS 140-1 and, 2) those imposed by SPYRUS, Inc.

3.1 FIPS 140-1 Related Security Rules

1. The Rosetta USB supports the following interfaces:
   - Data input interface.
   - Data output interface.
   - Control input interface.
   - Status output interface.
2. The Rosetta USB interfaces are logically distinct from each other.
3. The Rosetta USB inhibits all data output via the data output interface whenever an error state exists and during self-tests.
4. The Rosetta USB logically disconnects the output data path from the circuitry and processes performing the following key functions:
   - key generation, and
   - zeroization
5. The Rosetta USB enforces Identity-Based authentication.
6. The Rosetta USB supports a User role and a Cryptographic Officer role.
7. The Rosetta USB re-authenticates an identity when it is powered-up after being powered-off.
8. The Rosetta USB provides the following services:
   - Reference Table 4.1.
9. A hard, opaque material protects the non-excluded components within the cryptographic boundary.
10. The Rosetta USB contains production quality ICs with standard passivation.
11. The Rosetta USB is implemented as a production-grade multi-chip embodiment.
12. The Rosetta USB software is implemented using a high-level language except that limited use of a low-level language is used to enhance the performance of the module.
13. The Rosetta USB protects the following keys from unauthorized disclosure, modification and substitution:
    - secret keys.
    - private keys.
14. The Rosetta USB protects public keys against unauthorized modification and substitution.
15. The Rosetta USB generates keys using an approved FIPS 140-1 random number generator.
16. The Rosetta USB provides that:

- a key entered into,
- stored within, or
- output from

the Rosetta USB is associated with the correct entities to which the key is assigned.

17. The Rosetta USB provides the capability to zeroize all plaintext cryptographic keys and other unprotected critical security parameters within the Rosetta USB.
18. The Rosetta USB supports the following algorithms:

Encryption & Decryption
DES (ECB64, CBC64)
Triple DES (2 Key and 3 Key CBC)
Skipjack (ECB64) (internal use only)
Key Wrap & Unwrap
DES (ECB64, CBC64)
Triple DES (2 Key and 3 Key CBC)
Digital Signatures
DSA
RSA (512-1024 bit)
Digital Signature Verification
RSA (512-1024 bit)
Key Transport / Key Agreement
RSA (512-1024 bit)
KEA (Primitives only)

19. The Rosetta USB conforms to the EMI/EMC requirements specified in FCC Part 15, Subpart J, Class B.
20. The Rosetta USB performs the following self-tests:
    Power-up and on-demand tests:
    - Cryptographic algorithm known answer tests (KAT).
    - Software/firmware test.
    - Statistical random number generator tests (on-demand only).

    Conditional tests:
    - Pairwise consistency test (RSA key generation).
    - KAT (DSA key generation).
    - Continuous random number generator test.
21. The power-up tests do not require operator intervention in order to run.
22. The Rosetta USB provides an indication via the "status output" interface if all of the power-up tests are passed successfully.
23. The Rosetta USB outputs an error indicator via the status interface whenever an error state is entered due to a failed self-test.
24. The Rosetta USB does not perform any cryptographic functions while in an Error State.

3.2 SPYRUS Imposed Security Rules

1. The Rosetta USB does not input/output plaintext private/secret keys or other critical security parameters.
2. The Rosetta USB does not support a multiple concurrent operators.
3. The Rosetta USB does not support a bypass mode.
4. The Rosetta USB does not provide a maintenance role/interface.
5. The Rosetta USB requires the re-authentication of identity when changing roles.
6. The Rosetta USB does not support the loading of Software/Firmware.

4 Roles and Services

4.1 Rosetta USB Supported Roles

The Rosetta USB 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 Rosetta USB. Initialization is typically performed using a Certificate Authority Workstation (CAW) that is secured according to the site security policy of the deploying organization.

Before issuing a Rosetta USB to an end user, the Crypto-officer initializes the Rosetta USB with private keying material and certificate information. The Rosetta USB validates the Crypto-officer identity & split key before accepting any initialization commands.

User Role: The User role is available after the Rosetta USB has been loaded with a User personality & split key.

The Rosetta USB validates the User identity & split key 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 Rosetta USB Services

The following table describes the services provided by the Rosetta USB.

Table 4.1
Rosetta USB Services

Service Name	Service Description
Get Public	The GET PUBLIC command returns header information / file size on a private key file.
Get Challenge	The GET CHALLENGE command returns a fixed length random number to be used in authentication schemes between the USB and applications. This command is excluded from FIPS operation.
Execute	The EXECUTE command runs executable code loaded by SPYRUS in EEPROM during USB F/W installation.
Generate Fast Random	The GENERATE FAST RANDOM generates random numbers without employing entropy managing F/W on the input of the FIPS RNG. The random number is provided in the Data Out-Block. This command is excluded from FIPS operation.
Self-test	The SELF TEST command initiates the power-on self tests and the Statistical Random Number Generator self-test.
Read Binary	The READ BINARY command allows the content of a file to be read by the application based on access conditions.
Update Binary	The UPDATE BINARY command updates the data in the selected file with the data provided based on access conditions.
Secure Update Binary	The SECURE UPDATE BINARY command is used to load an encrypted private key into a selected key file based on access conditions.
File Status	The FILE STATUS command retrieves the status information for the file currently selected.
Select File	The SELECT FILE command sets a current file within a logical channel based on access conditions.
Create File	The CREATE FILE command creates a file providing the parent directory access conditions for the create command have been fulfilled.
Delete File	The DELETE FILE command deletes a file/ directory based on access conditions. This command is also used to delete the contents of an encrypted private key file.
Extend	The EXTEND command extends the length of a file or directory based on access conditions.
Directory	The DIRECTORY command retrieves a directory listing from the current directory and sub-directories if the recursive mode is used providing appropriate

	access conditions have been fulfilled.
Get Response	The GET RESPONSE command provides a generic method for transmitting APDU(s), or part of APDU(s) from the USB to the application when the available protocols cannot be used.
Rehabilitate (Enable)	The REHABILITATE command enables a previously disabled file based on access conditions. If executed on a file that is not disabled, no change is made to the file state and the command returns a success response code.
Invalidate (Disable)	The INVALIDATE command disables all operations on a file based on access conditions.
Block PIN	The BLOCK PIN command allows a PIN to be blocked.
Unblock PIN	The UNBLOCK PIN command allows a PIN that has been blocked using the BLOCK PIN command or after too many unsuccessful CHECK PIN attempts to be unblocked.
Check PIN	The CHECK PIN command inputs a split key to be used to decrypt an internally stored encrypted Pin Phrase to authenticate an operator to the USB.
Change PIN	The CHANGE PIN command allows the old split key to be changed to a new split key providing the old split key is correct.
Set Key	The SET KEY command facilitates setting an MEK to be the first, second or third key for use with the DES engine.
Load Key	The LOAD KEY command is used to overwrite (with FFs) a selected volatile key register to zeroize a session MEK.
RSA Key Generate	The RSA KEY GENERATE command generates an RSA public/private keypair.
RSA Wrap Key	The RSA WRAP KEY command facilitates public key wrapping of an MEK. This command will create a PKCS#1 compatible key object.
Generate Random	The GENERATE RANDOM command will cause the USB to generate a random number and return the value in the Data Out-Block.
Generate RA	The GENERATE RA command creates an RA for use in applications. User must load the required cryptographic variables prior to key generation.
Generate TEK	The GENERATE TEK command creates a pairwise bit pool for use in symmetric wrapping and unwrapping operations.
Generate X	The GENERATE X command creates DSA and/or KEA Public and Private Keying material. The required cryptographic variables must be loaded prior to key generation.
Load Cryptographic Data	The LOAD CRYPTOGRAPHIC DATA command supports the loading of cryptographic variables required by other commands.
Encrypt	The ENCRYPT command performs the encryption process on the input data, and returns the ciphertext data.
Decrypt	The DECRYPT command performs a decryption process on the input data and sets up the plaintext data for retrieval.
DSA Sign	The DSA SIGN command computes a digital signature on the input data using the DSA algorithm.

RSA Sign	The RSA SIGN command computes a digital signature on the input data using the RSA algorithm.
RSA Verify Signature	The RSA VERIFY SIGNATURE command verifies the RSA signature on the input data.

5 Identification and Authentication

5.1 Identification

The Crypto-officer installs personalities into the Rosetta USB during initialization. These personalities are used to define the identity and privileges of the Users.

5.2 Authentication

During initialization, the Crypto-officer loads the PIN file that contains the encrypted information required by the authentication process. The Crypto-officer provides the User with a split key required by the authentication process.

During login, a User selects a desired personality and provides the appropriate split key. If authentication is verified, the User is granted access to services defined by the assumed personality.

6 Access Control

6.1 Security Relevant Data Items (SRDIs)

Table 6.1
Rosetta USB SRDIs

SRDI	Description
Private keys (Asymmetric)	Cryptographic keys used in public key (asymmetric) algorithms. These keys are used in digital signature and decryption (during key exchange) operations.
Secret keys (Symmetric)	Cryptographic keys used in secret key (symmetric) algorithms. These keys are used in data encryption/decryption operations.

6.2 SRDI Access Type

Table 6.2
Rosetta USB Access Types

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

6.3 Access Matrix

Table 6.3
Rosetta USB Access Matrix

Service Name	Roles		SRDI's	
	CO	User	Private Keys	Secret Keys
Get Public	X	X		
Get Challenge	Excluded			
Execute	X	X		
Generate Fast Random	Excluded			
Self-test	X	X	U	U
Read Binary	X	X		
Update Binary	X	X		
Secure Update Binary	X	X	U	U
File Status	X	X		
Select File	X	X		
Create File	X	X		
Delete File	X	X	Note: Can be used to delete a file containing an encrypted private key.	
Extend	X	X		
Directory	X	X		
Get Response	X	X		
Rehabilitate (Enable)	X	X		
Invalidate (Disable)	X	X		
Block PIN	X			
Unblock PIN	X			U
Check PIN	X	X		U
Change PIN	X	X		U
Set Key	X	X		
Load Key	X	X		D
RSA Key Generate	X	X	G	U
RSA Wrap Key	X	X	U	U
Generate Random	X	X		
Generate RA	X	X		
Generate TEK	X	X		G
Generate X	X	X	G	U
Load Cryptographic Data	X	X		

Encrypt	X	X		U
Decrypt	X	X		U
DSA Sign		X	U	
RSA Sign		X	U	
RSA Verify Signature	X	X		