Security Policy

for

Version 1.2.3

FIPS 140-2 Non-Proprietary

Focus Systems Corporation

Revision History

Date	Revision	Author	Description
2003/12/05	0.0.0	Yuichi Hagiwara	Initial release.
2003/12/18	0.1.0	Yuichi Hagiwara	Applied the requirements from InfoGard Laboratories.
2004/01/22	0.1.1	Yuichi Hagiwara	Added diagram for descriptions of cryptographic boundary.
2004/02/12	0.2.1	Yuichi Hagiwara	Reflected InfoGard Laboratories' comments.
2004/02/20	0.3.2	Yuichi Hagiwara	Reflected design modification. Optimized design.
2004/03/16	0.4.2	Yuichi Hagiwara	Reflected the approval of SHA-2 that could now be used in FIPS approved mode.
2004/04/15	0.5.2	Yuichi Hagiwara	Added Security Rule in regards to the DRNG.
2004/05/07	0.6.2	Yuichi Hagiwara	Reflected InfoGard Laboratories' comments.
2004/06/08	1.0.0	Yuichi Hagiwara	Initial public release.
2004/11/11	1.1.0	Yuichi Hagiwara	Reflected CMVP's comments.
2004/12/06	1.2.0	Yuichi Hagiwara	Reflected CMVP's comments.
2005/01/21	1.2.1	Yuichi Hagiwara	Included C4CS version 1.1.0.
2008/08/08	1.2.2	Takayuki Uchiyama	Reflected Company name change
2016/05/26	1.2.3	Akira Hasegawa	Reflected Company name change

Table of Contents

1. Module Overview

The Security Policy is prepared as one of the requirements of FIPS 140-2 validation. However Focus Systems Corporation intends other purposes also.
It allows entities to:

- Determine if the cryptographic module is implemented as stated in the Security Policy.
- Describe how the FIPS 140-2 requirements are actually implemented in the cryptographic module.

C4CS Version 1.0.0 and 1.1.0 is a software cryptographic module targeted for FIPS 140-2 Security Level 1 overall. In FIPS 140-2 terms, C4CS is a multi-chip standalone module and the physically contiguous cryptographic boundary is defined as the outer enclosure of a general purpose computing system. As a software-only cryptographic module, the logical boundary is defined as a Windows DLL. All I/O is managed through the cryptographic module API. An external user application (software outside of the logical boundary) links to the cryptographic module at runtime. The diagram below illustrates the cryptographic boundary.

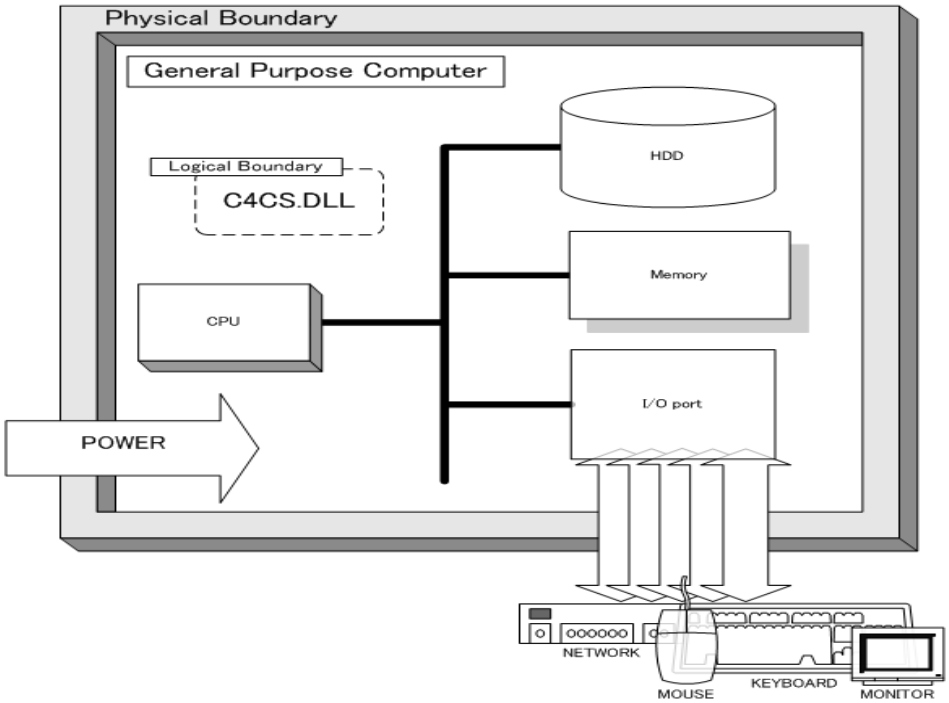

Diagram 1 - Cryptographic Boundary

2. Security Level

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2.

Table 1 - Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	1
Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	3
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

3. Modes of Operation

Approved mode of operation

In FIPS mode, the cryptographic module will support the following algorithms:

Table 2 - Approved modes of operation

AES	As defined in **FIPS PUB 197** with 128, 192, or 256 bit keys. AES will support the following modes; ECB, CBC, CFB, OFB, CTR
DH	**Diffie-Hellman** as a commercially available key establishment technique as allowed under **FIPS PUB 140-2 Annex D**.
DRNG	As defined in **ANSI X9.31, Appendix A.2.4** for generation of all cryptographic keys and for generating random numbers used by the user application for non-cryptographic functions.
ECDSA	As defined in **ANSI X9.62** for digital signature generation/verification.
HMAC-SHA-1	As defined in **FIPS PUB 198** for performing the power-up software integrity test. This functionality is ***not*** provided to the user application.
RSA	RSA will support the following modes; As defined in **RSAES OAEP** / **RSAES PKCS1v1.5** for encryption/decryption. This functionality is only supported for key wrapping as a commercially available key establishment technique as allowed under **FIPS 140-2 Annex D**. Encryption of bulk data is ***not*** supported. If the operator forces the module to encrypt non-key data, this Security Policy is violated. As defined in **RSASSA PKCS1v1.5** for digital signature generation/verification.
SHS	As defined in **FIPS PUB 180-2** for generating message digests with 160, 256, 384, 512 bit length.
SSS	**Secret Sharing Scheme** is used for split-knowledge procedures. If the operator forces the module to split non-key data, this Security Policy is violated. SSS will support the following modes; (k, n) threshold scheme, (k, L, n) threshold scheme

The C4CS cryptographic module may be configured for FIPS mode by making function calls associated with the algorithms listed above. If any of the non-Approved algorithms are accessed the module immediately switches to non-FIPS mode, and violates this Security Policy. Note that the module will ***not*** indicate if the module is operating in a FIPS approved mode or not.

Non-FIPS mode of operation

In non-FIPS mode, the cryptographic module provides non-FIPS Approved algorithms as follows:

Table 3 - Non-approved mode of operation

C4Custom	The **C4Custom** algorithm is a proprietary stream cipher of Focus Systems Corporation
RSA	As defined in **RSAES OAEP / RSAES PKCS1v1.5** for encryption/decryption of bulk data.
SSS	**Secret Sharing Scheme** used for splitting bulk data in the following modes; (k, n) threshold scheme, (k, L, n) threshold scheme

This module supports both FIPS approved and non-approved modes of operation. Note that the module ***must*** be rebooted to be in a FIPS mode, after once entering a non-FIPS mode.
See Section 6 for Access Control Policy.

4. Ports and Interfaces

The C4CS cryptographic module provides the following logical interfaces:

- Data input
- Data output
- Control input
- Status Output

The general purpose computing system that the cryptographic module executes on receives power from an external power supply.

5. Identification and Authentication Policy

Assumption of roles

The C4CS cryptographic module supports two distinct operator roles (User and Crypto-Officer). The cryptographic module does not support operator authentication. The operator assumes a given role by making function calls associated with the role. The cryptographic module does not support a maintenance role.

Table 4 - Roles and Required Identification and Authentication

Role	**Type of Authentication**	**Authentication Data**
User	N/A	N/A
Crypto-Officer	N/A	N/A

Table 5 - Strengths of Authentication Mechanisms

Authentication Mechanism	**Strength of Mechanism**
N/A	N/A

6. Access Control Policy

Roles and Services

Table 6 - Services Authorized for Roles

Role	Authorized Services
User: The entity that has access to all crypto related functions supported by the crypto module. The operator ***implicitly*** selects this role by making function calls associated with this role.	● AES ● ANSI X9.31 DRNG ● Diffie-Hellman ● ECDSA ● RSA encrypt/decrypt (only supported for key wrapping) ● RSA signature generation/verification ● Self-tests ● SHS ● SSS (only supported for key splitting) ● Zeroization
Crypto-Officer: The entity responsible for management activities including installing the software onto the platform, configuring the OS, and checking status of the module. Authentication is not required. The operator ***implicitly*** selects this role by making function calls associated with this role.	● Show Status

Service - Purpose and Use

Table 7 - Service name, purpose, and use

Service Name	**Purpose and Use**
AES	Allows Users to encrypt/decrypt various data.
ANSI X9.31DRNG	Allows Users to generate deterministic random numbers, and generate keys for AES, DH, ECDSA, RSA.
Diffie-Hellman	Allows Users to agree keys.
ECDSA	Allows Users to sign/verify messages.
RSA encrypt/decrypt	Allows Users to wrap/unwrap keys.
RSA signature/verification	Allows Users to sign/verify messages.
Self-tests	Allows Users to determine if the module is functioning properly.
SHS	Allows Users to generate message digests.
SSS	Allows Users to encode/decode keys.
Zeroization	Allows Users to zeroize key data.
Show Status	Allows Crypto-Officers to let the module indicate its status.

Definition of Critical Security Parameters (CSPs)

The following are **CSP**s contained in the module:

- **AES key (128, 192, 256):** Used for encryption and decryption of various data in ECB, CBC, CFB, OFB, and CTR modes.
- **ANSI X9.31 DRNG Seed key (ADSK):** Used within the Approved ANSI X9.31 DRNG for generation cryptographic keys, and random numbers used within crypto processes, or by the user application.
- **Diffie-Hellman Private Key (DHPK):** Used as a commercially available key establishment technique allowed under FIPS PUB 140-2 Annex D.
- **ECDSA Private Key (EPrK):** Used to digitally sign data passed into the module by the User.
- **RSA Private Key (decrypt) (RPKD):** Used to unwrap keys as a commercially available key establishment technique allowed under FIPS PUB 140-2 Annex D. This key is not supported for decryption of bulk data.
- **RSA Private Key (sign) (RPKS):** Used to digitally sign data passed into the module by the User.
- **SSS Split Data (SSD):** Key data split by using SSS.

Definition of Public Keys

The following are the public keys contained in the module:

- **Diffie-Hellman Public Key:** Used as a commercially available key establishment technique allowed under FIPS PUB 140-2 Annex D.
- **ECDSA Public Key:** Used to verify digitally signed data passed into the module by the User.
- **RSA public Key (encrypt):** Used to wrap keys as a commercially available key establishment technique allowed under FIPS PUB 140-2 Annex D. This key is not supported for encryption of bulk data.
- **RSA public Key (verify):** Used to verify digitally signed data passed into the module by the User.

Definition of CSPs Modes of Access

Table 8 defines the relationship between access to **CSP**s and the different module services. The modes of access shown in the table are defined as follows:

- **Generate (g):** a cryptographic key is generated using the Approved ANSI X9.31 DRNG.
- **Enter (e):** a cryptographic key is entered into the module.
- **Established via DH (dh):** a cryptographic key is established using Diffie-Hellman which is a commercially available key establishment technique allowed under FIPS PUB 140-2 Annex D.
- **Use (u):** a cryptographic key is used to perform cryptographic operations within its corresponding algorithm (as described in Section 3 of this document).
- **Output (o):** a cryptographic key is output from the module.
- **Zeroize (z):** a cryptographic key is destroyed.

Table 8 - CSP Access Rights within Roles & Services

Role		Service	Cryptographic Keys and CSPs Access Operation						
C.O.	**User**		**AES**	**ADSK**	**DHPK**	**EPrK**	**RPKD**	**RPKS**	**SSD**
	X	AES	e, g, u						
	X	DRNG	g, o	e, u	g, o	g, o	g, o	g, o	g, o
	X	DH	dh						
	X	ECDSA				e, g, u			
	X	RSAES					e, g, u		
	X	RSASSA						e, g, u	
	X	Self-Tests							
	X	SHS							
	X	SSS							e, g, u
	X	Zeroization	z	z	z	z	z	z	z
X		Show Status							

7. Operational Environment

The FIPS 140-2 Area 6 Operational Environment requirements are applicable since the C4CS software executes on general purpose Operating Systems.
The cryptographic module was tested and validated on the following platforms:

- Operating System: Windows® XP Service Pack 1
  - Windows RSAENH.DLL Ver. 5.1.2600.1029
- Operating System: Windows® 2000 Service Pack 3 with Hotfix 326886
  - Windows RSAENH.DLL Ver. 5.0.2195.3839
  - Windows RSABASE.DLL Ver. 5.0.2195.3839

As a 32bit DLL, the module will have compatibility with other 32bit Windows® Operating Systems.

The Crypto-Officer must ensure that the operating system is configured to run in a single user mode.

8. Security Rules

The C4CS cryptographic module's design corresponds to the C4CS cryptographic module's security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Security Level 1 module.

1. The cryptographic module shall provide two distinct operator roles. These are the User role, and the Crypto-Officer role.
2. The cryptographic module shall not provide authentication.
3. The cryptographic module shall support RSAES for key wrapping, and not data encryption.
4. The cryptographic module shall support SSS for splitting key data and not for bulk data. Values of k, L, n should be $n \geq k > 0$ for (k, n) threshold scheme and $n \geq k > L > 0$ for (k, L, n) threshold scheme.
5. The output of plaintext cryptographic keys shall require two independent internal actions.
6. The seed and seed key shall not assume the same value.
7. The same RSA key pair shall not be used for both key wrapping and digital signature operations.
8. The key establishment methods must employ 80 bits of security at minimum. I.e., for DH and RSA, (key size) ≥ 1024, and for ECDSA, (key size) ≥ 160.
9. The cryptographic module shall perform the following tests:
    A. Power up Self-Tests:
        1. Software Integrity Test (HMAC-SHA-1 verification)
        2. Cryptographic algorithm tests:
            a. AES Known Answer Test
            b. HMAC-SHA-1 Known Answer Test
            c. SHA-2 Known Answer Test
            d. ANSI X9.31 DRNG Known Answer Test
            e. RSA Known Answer Test (signature generation/verification)
            f. RSA Known Answer Test (key wrap/unwrap)
            g. ECDSA Known Answer Test
            h. DH Known Answer Test
        3. Critical Functions Tests:
            a. SSS Critical Function Test
    B. Conditional Self-Tests:

1. Continuous Random Number Generator (RNG) test
   – performed on ANSI X9.31 DRNG
2. RSA Pair-wise consistency test
3. ECDSA Pair-wise consistency test
4. DH Pair-wise consistency test

10. At any time the cryptographic module is in an idle state, the operator shall be capable of commanding the module to perform the power-up self-test.

11. If the module enters error state due to failing power-up self-test, the module shall be power cycled or reloaded in order to perform its service. The on-demand self-tests will only recover error states due to failure of conditional self-tests.

12. The module must be operated with an Operating System including RSAENH.dll with version 5.0.2195.3839 or higher, and/or RSABASE.dll with version 5.0.2195.3839 or higher. If the Operation System does not include the above DLL(s), the module will not be able to operate in FIPS approved mode when using AES, RSAES, DH, RSASSA, ECDSA, and SSS if keys are generated internally.

13. Prior to each use, the internal DRNG shall be tested using the conditional test specified in FIPS 140-2 §4.9.2.

14. Data output shall be inhibited during key generation, self-tests, zeroization, and error states.

15. Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.

16. The module shall not support concurrent operators (this is a Security Level 1 module).

17. The module shall inhibit cryptographic operations and data output in all error states.

18. The module shall be operated with an Operating System configured in Single User mode.

9. Physical Security Policy

Physical Security Mechanisms

Physical security requirements are not applicable to this software-only module. However, when installing the module, the crypto-officer must ensure that the computer system is stored in a secure environment. Since a software cryptographic module cannot equip physical security, the crypto-officer should stress on physical environment of the computer system.

Operator Required Actions

There are no operator required actions, as physical security is not applicable.

Table 9 – Inspection/Testing of Physical Security Mechanisms

Physical Security Mechanisms	**Recommended Frequency of Inspection/Test**	**Inspection/Test Guidance Details**
N/A	N/A	N/A

10. Mitigation of Other Attacks Policy

The module has ***not*** been designed to specific attacks outside the scope of FIPS 140-2.

Table 10 – Mitigation of Other Attacks

Other Attacks	**Mitigation Mechanism**	**Specific Limitations**
N/A	N/A	N/A

11. References

- National Institute of Standards and Technology, “FIPS PUB 140-2, Security Requirements for Cryptographic Modules”, May 25, 2001
- National Institute of Standards and Technology, “Derived Test Requirements for FIPS PUB 140-2, Security Requirements for Cryptographic Modules. Draft”, March 24, 2004
- National Institute of Standards and Technology, “FIPS PUB 197, Advanced Encryption Standard (AES)”, November 26, 2001
- National Institute of Standards and Technology, “FIPS PUB 186-2, Digital Signature Standard (DSS)”, October 5, 2001
- RSA Laboratories, “PKCS #1: RSA Encryption Standard. Version 2.1”, June 14, 2002.
- American Bankers Association, “Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)”, ANSI X9.62-1998.
- National Institute of Standards and Technology, “FIPS PUB 180-2, Secure Hash Standard (SHS)”, August 1, 2002
- National Institute of Standards and Technology, “FIPS PUB 198, Keyed-Hash Message Authentication Code (HMAC)”, March 6, 2002
- American Bankers Association, “Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)”, ANSI X9.31-1998.
- Adi Shamir, “How to share a secret”, Communications of the ACM, 612-613, 1979.
- Hirosuke Yamamoto, “Secret Sharing System Using (k, L, n) Threshold Scheme”, IEICE Trans., vol.J68-A, no.9, pp.945-952, September 1985.

12. Definitions and Acronyms

Table 11 – Definitions and acronyms

AES	**A**dvanced **E**ncryption **S**tandard
C4CS	**C4 C**ertified **S**uite
C4Custom	A proprietary stream cipher developed by Focus Systems Corporation
DH	**D**iffie-**H**ellman key agreement
DRNG	**D**eterministic **R**andom **N**umber **G**enerator
ECDSA	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm
RSA	A public key cryptosystem invented by Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman
SHA-2	**S**ecure **H**ash **A**lgorithm including SHA-256, SHA-384, and SHA-512.
SHS	**S**ecure **H**ash **S**tandard – the message digest will be 160, 224, 256, 384, or 512 bits (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512).
SSS	**S**ecret **S**haring **S**cheme