Entrust

Cryptographic Module Security Policy

Entrust Authority™ Security Toolkit for C++

Version: 6.2

Author: Chris Wood, Guenther Kramer
Date: Dec 4, 2003
Feature: ENTOT00057410 – STCPP 6.2 Process Activity
Version: 1.10

Table of Contents

1 Revision History 3

2 References 3

3 Target Audience 3

4 Introduction 4

4.1 Purpose of the Security Policy 4

4.2 Cryptographic Module Definition 4

4.3 Cryptographic Module Description 6

5 Specification of the Security Policy 7

5.1 Identification and Authentication Policy 7

5.2 Access Control and Key Management Policy 7

5.3 Physical Security Policy 9

5.4 Operational Environment 9

5.4.1 Assumptions 9

5.4.2 Installation and Initialization 10

5.4.3 Policy 10

5.5 Self-Tests 10

5.6 Mitigation of Other Attacks Policy 10

1 Revision History

Authors	Date	Version	Comment
Chris Wood	Jan 13, 2002	1.0	First Version
Guenther Kramer	Feb 21, 2003	1.1	Updated for validation
Guenther Kramer	Feb 23, 2003	1.2	Restructured Access Control Tables
Guenther Kramer	Feb 25, 2003	1.3	Minor updates from validation feedback.
Guenther Kramer	Feb 25, 2003	1.4	Add Self-Test description.
Guenther Kramer	March 4,2003	1.5	Add copyright statement on cover
Guenther Kramer	May 16, 2003	1.6	Add version and designation statement.
Guenther Kramer	Oct 7, 2003	1.7	NIST/CSE comment updates
Guenther Kramer	Oct 29, 2003	1.8	Update 4.2 and 4.3
Guenther Kramer	Nov 3, 2003	1.9	Minor revision
Guenther Kramer	Dec 4, 2003	1.10	Move AES-MAC algorithm

2 References

Author	Title
NIST	[1] FIPS PUB 140-2: Security Requirements For Cryptographic Modules, May 2001
NIST	[2] Derived Test Requirements for FIPS PUB 140-2, November 2001
NIST	[3] Implementation Guidance for FIPS PUB 140-1 and the Cryptographic Module Validation Program, July 2001
Rational	[4] Simplifying the Process of Change, Rational Software Corporation, 2001 (http://www.rational.com/media/products/clearcase/D710_ClearCase_ProdFam.pdf)
Rational	[5] Working in Base ClearCase, Rational Software Corporation, (http://www.rational.com/docs/v2002/cc/cc_dev.ux.pdf)
Entrust	[DD] Security Kernel Design Description for the Entrust Authority Security Toolkit for C++ 6.2, Sept 2002
Entrust	[FD] Security Kernel Functional Description for the Entrust Authority Security Toolkit for C++ 6.2, Sept 2002
Dell	[SM] Dell OptiPlex GXa Systems Service Manual, Dell Computer Corporation, 1997 (http://docs.us.dell.com/docs/systems/dfuj/51555bk0.pdf).
Dell	[RIG] Dell OptiPlex GXa Mini Tower Systems with Enhanced Manageability (EM) Reference and Installation Guide, Dell Computer Corporation, 1997 (http://docs.us.dell.com/docs/systems/dfuj/88763.pdf).

3 Target Audience

This document is intended to be part of the package of documents that are sent for FIPS validation. It is intended for the following people:

- NIST and the FIPS validation group
- Developers working on the release
- Product Verification
- Documentation
- Product and Development Managers

- Security Assurance

4 Introduction

This document contains a description of the Entrust Authority™ Security Toolkit for C++ (STCPP) Cryptographic Module Security Policy. It contains a specification of the rules under which the STCPP cryptographic module must operate. These security rules were derived from the requirements of FIPS 140-2 validation [1].

4.1 Purpose of the Security Policy

There are three major reasons that a security policy is defined for and must be followed by the cryptographic module:

- It is required for FIPS 140-2 validation.
- It allows individuals and organizations to determine whether the cryptographic module, as implemented, satisfies the stated security policy.
- It describes the capabilities, protection, and access rights provided by the cryptographic module, allowing individuals and organizations to determine whether it will meet their security requirements.

4.2 Cryptographic Module Definition

This section defines the Cryptographic Module that is being submitted for validation to FIPS PUB 140-2, level 1.

The module consists of the following generic components:

1. A commercially available general-purpose hardware-computing platform. A generic high-level block diagram for such a platform is provided in Figure 1.
2. A commercially available Operating System (OS) that runs on the above platform.
3. A software component, the STCPP, that runs on the above platform and operating system. This component is custom designed and written by Entrust in the C++ computer language and is identical, at the source code level, for all identified hardware platforms and operating systems. The source code (see [FD] for list of classes) is compiled into a set of dynamic link libraries on the above OS. An Application Programming Interface (API) is defined as the interface to the cryptographic module.

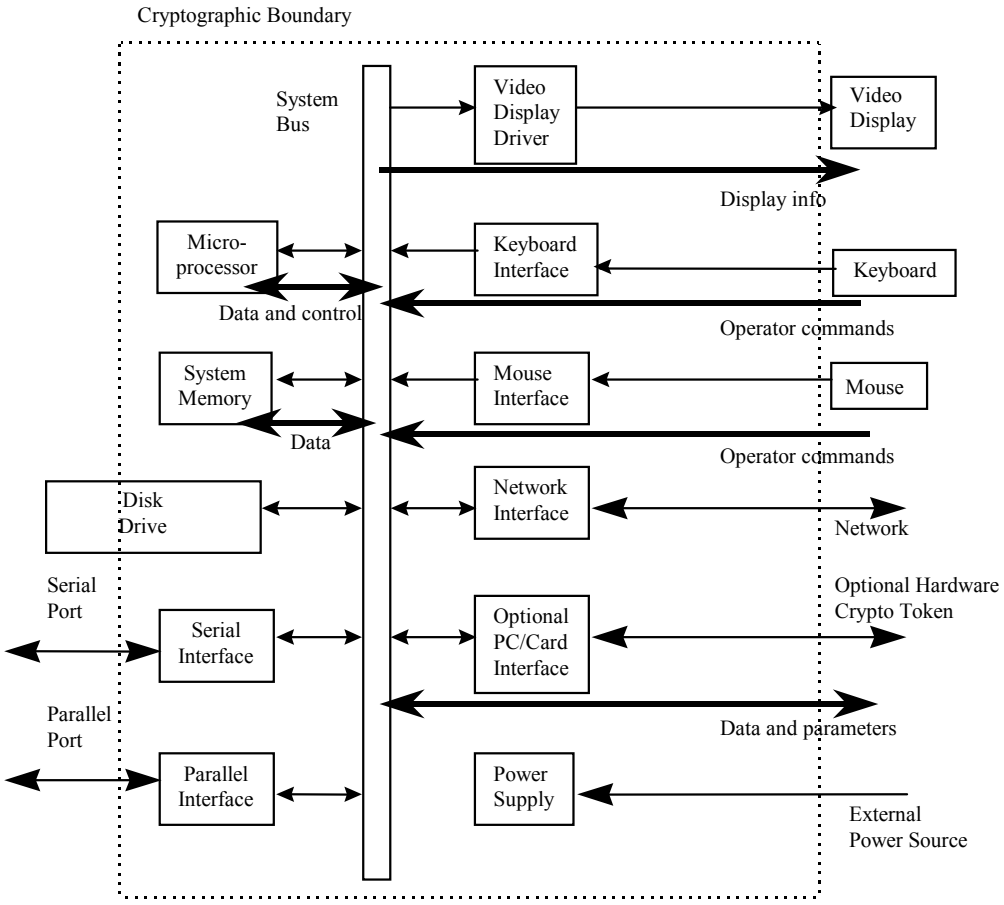
The cryptographic module contains the following hardware computing platform and operating system:

1. A Dell OptiPlex GXa Midsize Personal Computer system with:
    - An Intel Pentium II 400MHz processor,
    - 128MB system RAM (DIMM),
    - 2 serial ports and 1 parallel port,
    - 4.3GB hard drive,
    - A 3COM 3C509 Ethernet card,
2. Operating Systems:
    - Microsoft Windows NT 4.0 SP6a
    - Microsoft Windows 2000 SP3
    - Microsoft Windows XP SP1a

A detailed technical description of the Dell OptiPlex GXa platform is included in [SM] and [RIG].

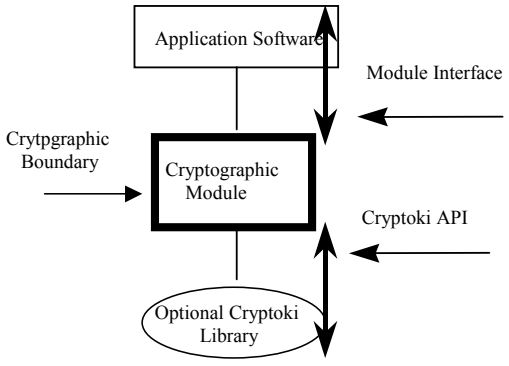
The STCPP cryptographic module is also suitable for platforms from the same or other manufacturers, based on compatible processors with equivalent or greater system resources, and equivalent or later

Operating System versions. Also, the STCPP cryptographic module used on all Microsoft Operating Systems is identical.



Note: All arrows indicate data flow, however; only bold arrows indicate data (plaintext and encrypted) flows into and out of the Cryptographic Module

Figure 1: Cryptographic Module block diagram for hardware.



Note: Bold arrows indicate data (plaintext and encrypted) flows into and out of the Cryptographic Module

Figure 2: Cryptographic Module block diagram for software.

4.3 Cryptographic Module Description

The STCPP consists of a set of shared libraries (.dll) files. The cryptographic module portion of the STCPP consists of a staticly linked C++ library that is included in the core shared library file (EtkCore.dll). The cryptographic module provides a set of functions (API) that allows developers to integrate the cryptographic module security features into the applications they design. The cryptographic module API is described in detail in the Cryptographic Module Functional Description [FD] companion document.

The purpose of the cryptographic module is to provide application developers with the access to cryptographic algorithms, and the ability to integrate security into the applications they design. The types of cryptographic algorithms provided include:

- Symmetric Ciphers (encryption/decryption/key generation)
- Asymmetric Ciphers (encryption/decryption/key generation)
- Message Digests (hashing)
- Signatures (signing/verification)
- Message Authentication Codes (creation)
- Random Number/Seed Generation
- Key Agreement

5 Specification of the Security Policy

5.1 Identification and Authentication Policy

The Cryptographic Module does not identify nor authenticate any user (in any role) that is accessing the Cryptographic Module. This is only acceptable for FIPS 140-2 level 1 validation.

Role	Type of Authentication	Authentication Data
User	N/A	N/A
Cryptographic Officer	N/A	N/A

Table 1: Roles and Required Identification and Authentication

Authentication Mechanism	Strength of Mechanism
None	N/A

Table 2: Strengths of Authentication Mechanisms

5.2 Access Control and Key Management Policy

The Cryptographic Module supports two roles: User and Cryptographic Officer. An operator performing a service within any role can read/write/execute critical security parameters (CSP) only through the invocation of a service by use of the Cryptographic Module API.

An operator is explicitly in the User or Cryptographic Officer role based upon the services chosen. If any of the User specific services are called, then the operator is in the User role; otherwise the operator is in the Cryptographic Officer role.

The critical security parameters (CSP) defined for this module consist of cryptographic keys and random number seeding material.

Each service within each role can only access the cryptographic keys and CSPs that the service's API defines. The following cases exist:

- A cryptographic key or CSP is provided to an API as an input parameter; this indicates read/write access to that cryptographic key or CSP.
- A cryptographic key or CSP is returned from an API as a return value; this indicates read access to that cryptographic key or CSP.

In order to protect sensitive keying information, the module does not allow any secret or private keys to be exported in raw/plaintext form. To export keys, another FIPS approved symmetric encryption key must be provided, which is used to encrypt the key before it is returned.

The module also internally zeroizes all intermediate results of sensitive operations. CSPs are zeroized when no longer needed by overwriting the CSP with zeros. The application using the cryptographic module can also explicitly zeroize keys by calling the Zeroize() function.

The module also obfuscates the internal representation of private keys that or stored in allocated stack or heap memory. This is an extra level of security to hide sensitive information in memory that an attacker may gain access to.

The cryptographic module supports 4 types of access to a service:

1. read – allows read access to the CSP's
2. wrapped read – allows read access to CSP's which encrypted by an encryption key (wrapped)
3. write – allows write access to the CSP's
4. execute – allows execution of the service

The FIPS approved services corresponding to each of the supported roles, along with the types of access are detailed in **Table 3**. The

Table 3: FIPS 140-2 Approved Services Authorized for Roles

Approved Service	*CSP's*	*Certificate Number*	*Accessible Roles*	*Type of Access*
Symmetric Encryption/Decryption				
	AES Key	59	User / Crypto Officer	wrapped read, execute
	DES Key (legacy systems only)	56	User / Crypto Officer	wrapped read, execute
	Triple-DES Key	6	User / Crypto Officer	wrapped read, execute
Asymmetric Key Wrapping				
	RSA Public Key	Note 1	User / Crypto Officer	read, execute
	RSA Private Key	Note 1	User / Crypto Officer	wrapped read, execute
Digital Signature Generation/Verification				
	DSA Public Key	10	User / Crypto Officer	read, execute
	DSA Private Key	10	User / Crypto Officer	wrapped read, execute
	RSA Public Key	Note 1	User / Crypto Officer	read, execute
	RSA Private Key	Note 1	User / Crypto Officer	wrapped read, execute
Hash Generation				
	SHA-1	10	User / Crypto Officer	read, execute
MAC Generation				
	DES MAC	56	User / Crypto Officer	read, execute
	Triple-DES MAC	6	User / Crypto Officer	read, execute
	HMAC-SHA1	10	User / Crypto Officer	read, execute
Random Number Generation				
	Seeding data (as defined by FIPS 186-2 Appendix 3.1)		User / Crypto Officer	read, write, execute
Key Agreement				
	Diffie-Hellman public parameters	n/a	User / Crypto Officer	read, write, execute
	Diffie-Hellman private keys		User / Crypto Officer	wrapped read, execute
Module Initialization	n/a	n/a	User / Crypto Officer	execute

Note 1: Vendor Affirmed PKCS #1 v1.5

The other services that are not FIPS approved are detailed in **Table 4** below.

Table 4: Other Services Authorized for Roles

Approved Service	*CSP's*	*Accessible Roles*	*Type of Access*
Symmetric Encryption/Decryption			
	CAST key	User / Crypto Officer	wrapped read, execute
	CAST3 key	User / Crypto Officer	wrapped read, execute
	CAST5 key	User / Crypto Officer	wrapped read, execute
	IDEA key	User / Crypto Officer	wrapped read, execute
	RC2 key	User / Crypto Officer	wrapped read, execute
	RC4 key	User / Crypto Officer	wrapped read, execute
Asymmetric Encryption/Decryption			
	RSA Public Key	User / Crypto Officer	read, execute
	RSA Private Key	User / Crypto Officer	wrapped read, execute
Digital Signature Generation/Verification			
	ECDSA Public key	User / Crypto Officer	read, execute
	ECDSA Private key	User / Crypto Officer	wrapped read, execute
Hash Generation			
	SHA-256	User / Crypto Officer	read, execute
	MD2	User / Crypto Officer	read, execute
	MD5	User / Crypto Officer	read, execute
	RIPEMD-160	User / Crypto Officer	read, execute
MAC Generation			
	AES MAC	User / Crypto Officer	read, execute
	CAST MAC	User / Crypto Officer	read, execute
	CAST3 MAC	User / Crypto Officer	read, execute
	CAST5 MAC	User / Crypto Officer	read, execute
	HMAC-MD5	User / Crypto Officer	read, execute
	HMAC-RMD160	User / Crypto Officer	read, execute
	IDEA MAC	User / Crypto Officer	read, execute
	RC2 MAC	User / Crypto Officer	read, execute
	RC4 MAC	User / Crypto Officer	read, execute
Key Agreement			
	SPEKE public key	User / Crypto Officer	read, execute
	SPEKE private key	User / Crypto Officer	wrapped read, execute

Detailed information on which Cryptographic Module APIs belong to each role can be found in the Security Kernel Functional Description [FD].

5.3 Physical Security Policy

The physical security of the cryptographic module is provided by the PC that it is being used on. Physical Security requirements for FIPS 140-2 Level 1 modules are not applicable.

5.4 Operational Environment

5.4.1 Assumptions

The following assumptions are made about the operating environment of the cryptographic module:

- Unauthorized reading, writing, or modification of the module's memory space (code and data) by an intruder (human or machine) is not feasible.

- The module is initialized to the FIPS 140-2 mode of operation

5.4.2 Installation and Initialization

The following steps must be performed to install and initialize the cryptographic module for operating in a FIPS 140-2 compliant manner:

- The operating system must be configured to allow only a single user.
- All the dynamic link libraries shipped with the STCPP must be copied locally to the machine on which the STCPP is being used.
- The module must be initialized and self-tests run to enter FIPS mode.

5.4.3 Policy

The following policy must always be followed in order to achieve a FIPS 140-2 mode of operation:

- The cryptographic module must only be used by one human operator at a time, and must not be actively shared among operators at any time. Also, there must be only one instance of the cryptographic module loaded into RAM at any give time on any given machine.
- Only FIPS 140-2 approved/vendor affirmed algorithms can be used.
- Virtual memory that exists on the machine when the cryptographic module runs must be configured to reside on a local, not a networked, drive.
- The above conditions must be upheld at all times in order to ensure continued system security after initial setup of the validated configuration. If the module is removed from the above environment, it is assumed to not be operational in the validated mode until such time as it has been returned to the above environment and re-initialized by the user to the validated condition.

5.5 Self-Tests

Below is the self tests provided by the STCPP:

1. Power-Up Tests
    a) Cryptographic Algorithm Known Answer Tests (Encrypt/Decrypt test for all algorithms)
    b) Software Integrity Test (Triple-DES MAC)
    c) Critical Functions Tests (RNG Test at initialization)

2. Conditional Tests
    a) Pairwise Consistency Tests (RSA, DSA)
    b) The Continuous Random Number Generator Test

5.6 Mitigation of Other Attacks Policy

The cryptographic module is not designed to mitigate any specific attacks.

Other Attacks	Mitigation Mechanism	Specific Limitations
None	N/A	N/A

Table 5: Mitigation of Other Attacks