
# SUSK Security Module

(Software Version: 1.1)

# FIPS 140-2
# Non-Proprietary Security Policy

**Level 1 Validation**

**Document Version 0.5**

# Table of Contents

# 1 Introduction

## 1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the SUSK Security Module from Secured User, Inc. This Security Policy describes how the SUSK Security Module meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The Secured User website (http://www.secured-services.com) contains information on the full line of products from Secured User.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at: http://csrc.nist.gov/groups/STM/cmvp/index.html.

The SUSK Security Module is also referred to in this document as the cryptographic module, the software module, or the module.

 

# 2 SUSK Security Module

## 2.1 Product Overview

Secured User provides products for centrally-administered security environments for enterprises running mainframe, client-server, and web-based applications. The company's application environment provides:

- **Complete Identity Life Cycle Management:** Comprehensive identity administration, access control, and audit management functionality, built on a single, integrated platform. This makes for a unique cohesiveness in managing the entire life cycle of a digital identity.
- **A Single Identity, with True, Enterprise-Wide Reach:** The realization of true single sign-on to all application types, whether Web-based, client/server, or legacy mainframe.
- **Ease of Deployment:** Overcomes traditional barriers to identity management projects, deploying rapidly, non-disruptively, without changes to legacy code or directories, and with radically less risk and complexity than competing approaches.

Secured User's product performs all of its work by transparently intercepting and transforming the data stream between entities.

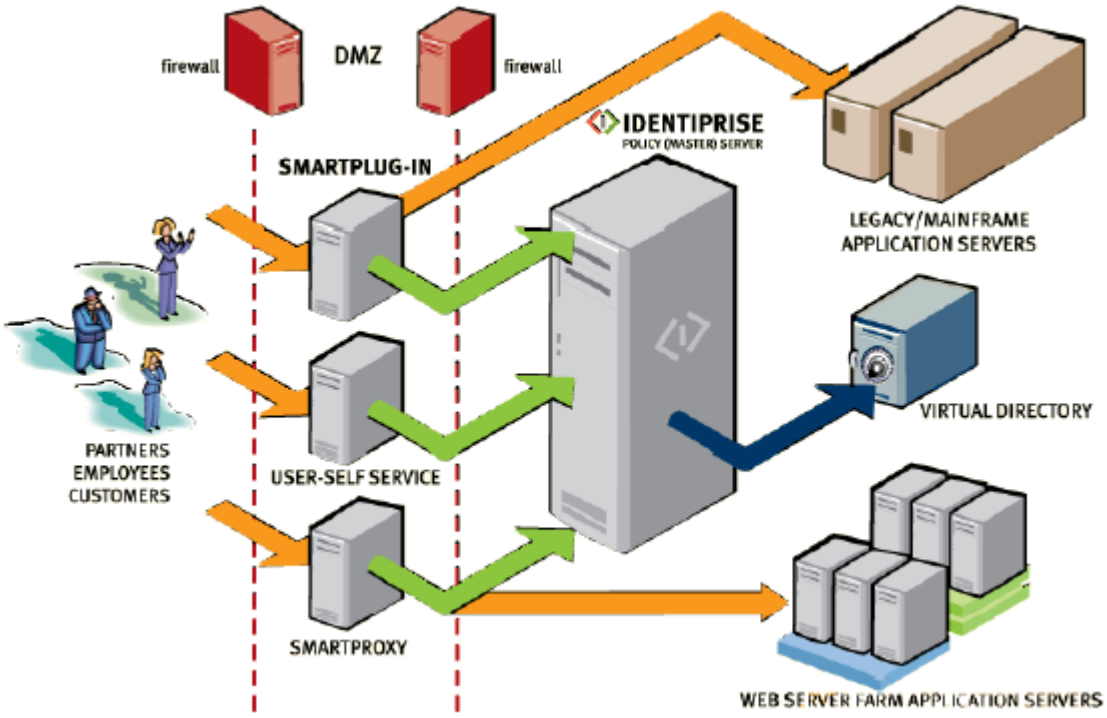


**Figure 1 – Secured User's Application Deployment**

All of the cryptographic functionalities of the Secured User product are provided by a central shared library, SUSK Security Module. The cryptographic module offers Transport Layer Security (TLS) services along with bulk encryption and hashing services exclusively to Secured User application. This application is considered as host application to the module.

## 2.2 Module Specification

The SUSK Security Module is a software-based cryptographic module tested for use on a standard server running the Windows Server 2003, HP-UX and Linux Operating Systems (OS). The module's software is entirely encapsulated by the logical cryptographic boundary as shown in Figure 2 below.

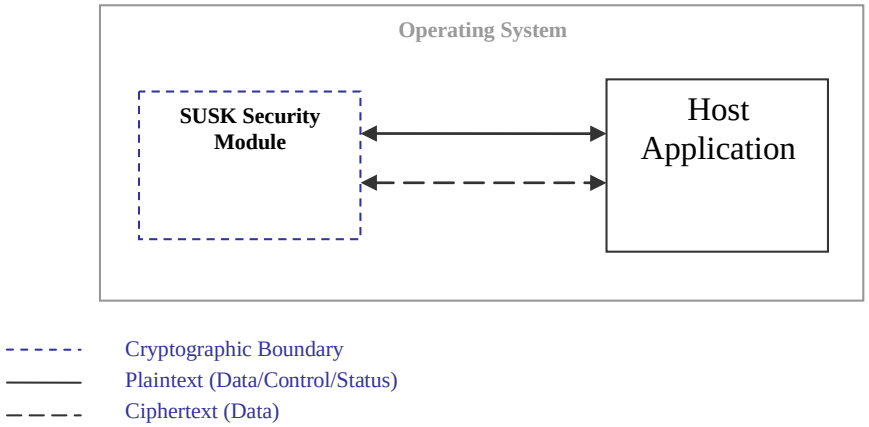


**Figure 2 - Logical Cryptographic Boundary**

In addition to the binaries, the physical device consists of the motherboard circuits, the central processing unit (CPU), random access memory (RAM), read-only memory (ROM), server case, expansion cards, and other hardware components included in the server such as hard disk, floppy disk, CD-ROM drive, power supply, and fans. The physical cryptographic boundary of the module is the hard opaque metal and plastic enclosure of the server. A physical block diagrams is shown below.

 

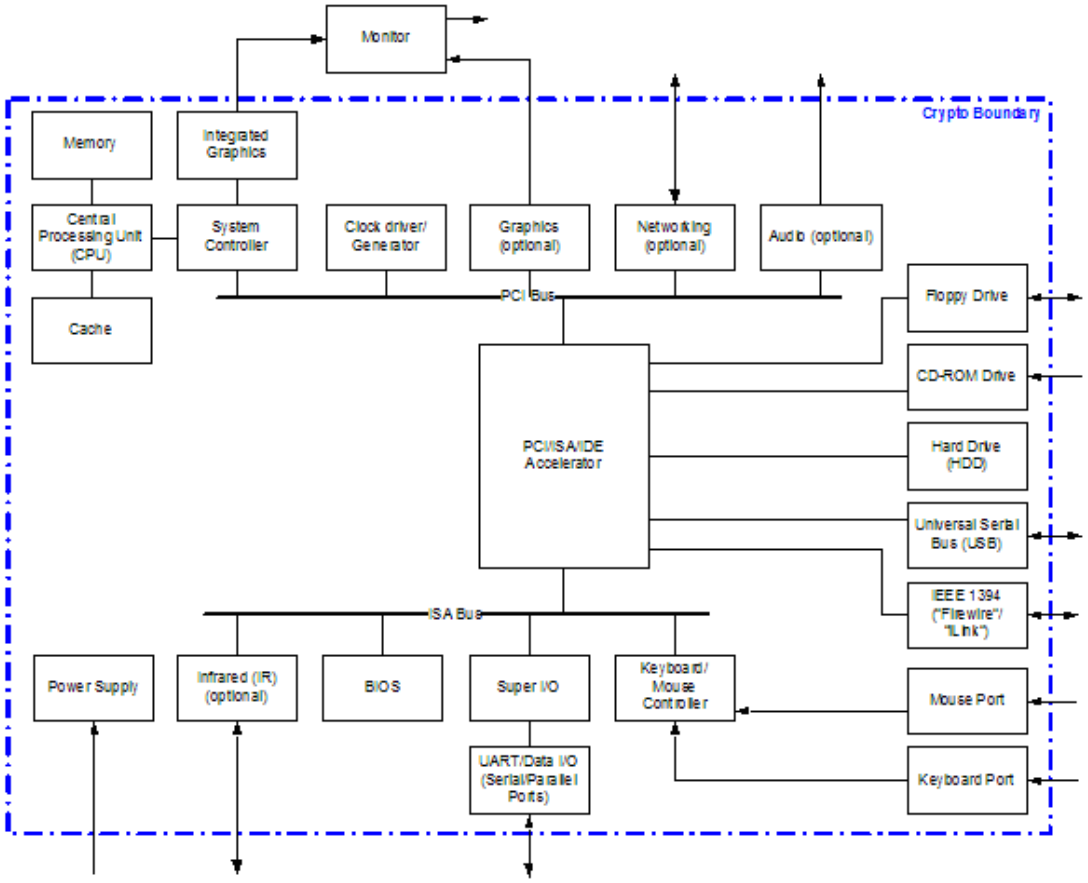


**Figure 3 - Standard Server Physical Block Diagram**

Per FIPS 140-2 terminology, the SUSK Security Module is a multi-chip standalone module that meets overall level 1 FIPS 140-2 requirements. The SUSK Security Module is validated at the following FIPS 140-2 Section levels:

**Table 1 - Security Level per FIPS 140-2 Section**

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

## 2.3 Module Interfaces

The module's logical interfaces exist in the software as an Application Programming Interface (API). Physically, ports and interfaces are considered to be those of the server. Both the API and physical interfaces can be categorized into following logical interfaces defined by FIPS 140-2:

- Data Input Interface
- Data Out Interface
- Data Control Interface
- Status Output Interface
- Power Interface

All of these logical interfaces are described in the following table:

**Table 2 - FIPS 140-2 Logical Interfaces**

| FIPS 140-2 Logical Interface | SUSK Security Module Port/Interface | Physical Port/Interface |
|---|---|---|
| Data Input | Function calls that accept, as their arguments, data to be used or processed by the module. | Keyboard, mouse, CD-ROM, floppy disk, and serial/USB/parallel/network ports |
| Data Output | Arguments for a function that specify where the result of the function is stored. | Floppy disk, monitor, and serial/USB/parallel/network ports |
| Control Input | Function calls utilized to initiate the module and the function calls used to control the operation of the module. | Keyboard, CD-ROM, floppy disk, mouse, and serial/USB/parallel/network port |
| Status Output | Return values for function calls | Floppy disk, monitor, and serial/USB/parallel/network ports |
| Power Interface | Not applicable | Power Interface |

## 2.4 Roles and Services

Two roles are supported by the module (as required by FIPS 140-2) that operators may assume: a Crypto-Officer (CO) role and a User role. The operator of the module must assume either of the roles based on their operation without any authentication. Both of the roles and their responsibilities are described below.

### 2.4.1 Crypto-Officer Role

The Crypto-Officer (CO) is expected to install and configure the module in the FIPS mode of operation. The CO is also responsible for monitoring the module's configuration and operational status from the Windows event log. Please see the *Secure Operation* section for a complete list of the Crypto-Officer's responsibilities.

Descriptions of the services available to the Crypto-Officer role are provided in the table below.

**Table 3 – Mapping of Crypto-Officer Role's Services to Inputs, Outputs, CSPs, and Type of Access**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Installation | Installing the software module | Command | Result of installation | None |
| Uninstall | Uninstall the software module | Command | Module uninstalled | None |
| SUSK_SSL_library_init | Enables the FIPS mode of operation. | API call | The module executes in FIPS mode of operation | None |
| SUSK_selftest | Performs power-up self-tests | API call | Self-tests performed | None |

### 2.4.2 User Role

The User role accesses the module's cryptographic services that include TLS operation, encryption/decryption, and hashing. The following table lists the services available to the User role.

**Table 4 – Mapping of User Role's Services to Inputs, Outputs, CSPs, and Type of Access**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| SUSK_TLS_connect | Initiates the TLS handshake with a server | API call | TLS connection completed | Diffie-Hellman (DH) key pair – Read<br>RSA key pair – Read<br>Session Key – Write |
| SUSK_TLS_close | Terminates a TLS connection | API call | TLS connection terminated | None |
| SUSK_get_error | Obtains result code for TLS operation | API call | Status of TLS connection | None |
| SUSK_TLS_read | Reads bytes from a TLS connection | API call with input data | Data read from a TLS session | Session Key – Read |
| SUSK_TLS_write | Writes bytes to a TLS connection | API call | Data sent over a TLS session | Session Key – Read |
| SUSK_TLS_accept | Accepts TLS handshake from a client | API call | TLS handshake initiated | DH key pair – Read<br>RSA key pair – Read<br>Session Key – Write |
| SUSK_keygen | Generates a random key | API call | Key generated | Pseudo Random Number Generator (PRNG) seed – Read/Write<br>Symmetric Key – Write |
| SUSK_encrypt | AES cipher operation | API call with a symmetric key and input data | Data encrypted /decrypted | Symmetric key – Read |
| SUSK_hashinit | Initiate hashing mechanism | API call | Hashing context initializes | None |
| SUSK_hashupdate | Updates hashing digest | API call with input data | Hash value updated | None |
| SUSK_hashfinal | Finalizes hash value | API call | Returns digest | None |
| SUSK_Rijndael_encrypt | AES encryption | API call with a symmetric key and input data | Data encrypted | Symmetric key – Read |
| SUSK_Rijndael_decrypt | AES decryption | API call with a symmetric key and input data | Data decrypted | Symmetric key – Read |

Operator spaces on the module are separated by the Operating System. The OS must be configured for single user mode in FIPS-approved mode of operation.

## 2.5 Physical Security

The physical security requirements do not apply to this module since it is a software module and does not implement any physical security mechanisms.

Although the module consists entirely of software, the FIPS 140-2 tested platform is a standard server, which has been tested for and meets applicable Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined in Subpart B of FCC Part 15.

## 2.6 Operational Environment

SUSK module runs on the general purpose Windows Server 2003 and Linux Operating Systems. The module was tested on the following Operating Systems for FIPS 140-2 requirements:

- Windows Server 2003 (32 and 64 bit)
- Linux Red Hat 2.6, Fedora 6, and SuSe 10
- HP-UX 11.11

The OS must be configured for single user mode per NIST CMVP guidance for FIPS 140-2 compliance. Single user mode configuration instructions for the OS can be found in the Secure Operation section.

## 2.7 Cryptographic Key Management

The SUSK Security Module implements the following FIPS-approved algorithms:

- AES ECB/CBC mode – FIPS 197 (certificate #474 and #770)
- SHA-1 – FIPS 180-2 (certificate #542)
- HMAC SHA-1 – FIPS 198 (HMAC certificate #231, SHA-1 certificate #542)
- Deterministic Random Number Generator (RNG) – Appendix A.2.4 of ANSI X9.31 (certificate #257)

Additionally, the software module utilizes the following non-FIPS-approved algorithm[1] implementations:

- Diffie-Hellman (key agreement, key establishment methodology provides 80-bits of encryption strength)
- RSA (key wrapping, key establishment methodology provides 80-bits of encryption strength)

[1] In order to operate in an Approved mode of operation compliant to FIPS 140-2, Diffie-Hellman and RSA keys of 1024-bits are used.

 

The module supports the following critical security parameters:

**Table 5 - List of Cryptographic Keys, Cryptographic Key Components, and CSPs**

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Diffie-Hellman Key pair | Diffie-Hellman key pair (1024 bits) | Generated for TLS session establishment | Never exit the module | Plaintext in volatile memory | Zeroized after session is over or by power cycle | Used by the module in establishing a session key during TLS handshake |
| Session Key | AES key (256 bits) | Established during TLS session establishment | Exits the module in plaintext | Plaintext in volatile memory | Zeroized after session is over or by power cycle | Used to encrypt the TLS session. |
| Symmetric Key | AES key (128, 192, 256 bits) | Generated internally or entered electronically in plaintext | Exits the module in plaintext | Plaintext in volatile memory | By power cycle | Encrypts/decrypts data |
| Asymmetric Key | RSA key pair (1024 bits) | Generated internally | Exits the module in plaintext | Plaintext in volatile memory | By power cycle | Used by the module in establishing a session key during TLS handshake |
| Software Integrity Test Key | HMAC key (16 bytes) | Hard coded into the module | Never exit the module | Stored on hard drive in plaintext | Uninstalling the module | Checks integrity of the software |
| PRNG Seed | 8 bytes of seed value | Prearranged value | Never exit the module | Plaintext in volatile memory | By power cycle | Generate ANSI X9.31 A.2.4 PRNG |

## 2.8 Self-Tests

In order to prevent any secure data from being released, it is important to test the cryptographic components of a security module to ensure that all components are functioning correctly. The module performs the following self-tests:

- Power-Up Self-Tests:
  - Software integrity check using HMAC SHA-1
  - Known Answer Tests (KATs)
    - AES KATs
    - SHA-1 KAT
    - RSA Pairwise Consistency Test
    - X9.31 PRNG KAT

The SUSK Security Module performs the following conditional self-tests:

- Continuous RNG for FIPS approved PRNG
- RSA Pairwise Consistency Test
- Diffie-Hellman Pairwise Consistency Test

Status output of self-tests are logged in the Windows Event log file, unless the Crypto-Officer configures the module to record the status output into a local file. On Linux platforms, status output is always logged into a local file.

## 2.9 Design Assurance

Configuration management for all of the Secured User's source code files and user manual is provided by Subversion. The source code revisions are maintained in a private Subversion repository with write access restricted to the core development team.

## 2.10 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 level 1 requirements for this validation.

# 3 Secure Operation

The SUSK Security Module meets Level 1 requirements for FIPS 140-2. The sections below describe how to configure and maintain the module in a FIPS-approved mode of operation. Operating the module without following this guidance will remove the module from the FIPS-approved mode of operation.

## 3.1 Crypto-Officer Guidance

### 3.1.1 Initial Setup

The Crypto-Officer is responsible for installing/uninstalling, configuring, and managing the module. The SUSK Security Module is available on a Compact Disc (CD) to the Crypto-Officer from Secured User through shipping using a standard carrier (i.e. FedEx or UPS). The Crypto-Officer is responsible for inspecting the CD and its packaging upon receipt for signs of tampering. If evidence of tampering is found, the Crypto-Officer should contact Secured User immediately and should not use the module. The software module will be provided to users by Secured User for exclusive use in their products only. The module is installed during installation of the host application, Identiprise SecuredUser. With the delivered software, the Crypto-Officer also receives a complete documentation library, which includes the Installation Guidance for the host application.

Before installing the module, the CO must configure the OS for single user mode as instructed below.

To configure Windows Server 2003 for single user mode, the CO must ensure that all remote guest accounts are disabled in order to ensure that only one operator can log into the OS at a time. The services that should be disabled are –

- Server services
- Terminal services
- Remote registry service
- Remote desktop and remote assistance service

The specific procedures for each of the Linux variants are described below.

1. The CO has to log in as the “root” user.
2. All the users except “root” and the pseudo-users should be removed from the system files /etc/passwd and /etc/shadow. Password fields in /etc/shadow for the pseudo-users should be either a star (*) or double exclamation mark (!!). This prevents login as the pseudo-users.
3. The system file /etc/nsswitch.conf needs to be edited such that “files” the only option for “passwd”, “group”, and “shadow”. This disables NIS and other name services for users and groups.
4. In the /etc/xinetd.d directory, the value of “disable” need to be set to “yes” for the following files: “rexec”, “rlogin”, “rsh”, “rsync”, “telnet”, and “wu-ftpd”.
5. The system has to reboot for the changes to take effect.

The specific procedure for HP-UX is described below.

1. The CO has to log in as the “root” user.
2. All the users except “root” and the pseudo-users should be removed from the system files /etc/passwd and /etc/shadow. Password fields in /etc/shadow for the pseudo-users should be either a star (*). This prevents login as the pseudo-users.
3. The system file /etc/nsswitch.conf needs to be edited such that “files” the only option for “passwd”, “group”, and “shadow”. This disables NIS and other name services for users and groups.
4. In the /etc/inetd.conf file, lines for remote login, remote command execution, and file transfer daemons such as telnetd, rlogind, remshd, rexecd, ftpd, and tftpd need to be removed.

5. The system has to reboot for the changes to take effect.

### 3.1.2 Management

The Crypto-Officer should monitor the module's status by regularly checking the event log file of the module. If any irregular activity is noticed or the module is consistently reporting errors, then Secured User customer support should be contacted.

## 3.2 User Guidance

The User accesses the module's cryptographic functionalities only. Although the User does not have any ability to modify the configuration of the module, they should report to the Crypto-Officer if any irregular activity is noticed.

# 4 Acronyms

**Table 6 - Acronyms**

| Acronym | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| ANSI | American National Standards Institution |
| CD | Compact Disk |
| CMVP | Cryptographic Module Validation Program |
| CO | Crypto-Officer |
| CPU | Central Processing Unit |
| CRNGT | Continuous Random Number Generator Test |
| CSP | Critical Security Parameter |
| DH | Diffie-Hellman |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| FCC | Federal Communication Commission |
| FIPS | Federal Information Processing Standard |
| FSM | Finite State Machine |
| HMAC | (Keyed-) Hash MAC |
| KAT | Known Answer Test |
| LED | Light Emitting Diode |
| NIST | National Institute of Standards and Technology |
| NVLAP | National Voluntary Laboratory Accreditation Program |
| OS | Operating System |
| PRNG | Pseudo Random Number Generator |
| RAM | Random Access Memory |
| RNG | Random Number Generator |
| ROM | Read Only Memory |
| RSA | Rivest, Shamir, and Adleman |
| SHA | Secure Hash Algorithm |
| SUSK | SecuredUser Security Kernel |
| TLS | Transport Layer Security |
| USB | Universal Serial Bus |
| VSS | Visual Source Safe |