SSL 100 SDK Module Security Policy

Issue Date: January 08, 2003

Revision 1.1C: January 08, 2003

Shiva Corporation
7755 Boul. Henri Bourassa Ouest
Saint-Laurent, Québec
Canada H4S 1P7

Table of Contents

DOCUMENT REVISION HISTORY

REVISION	ISSUE DATE	SECTIONS AFFECTED	DESCRIPTION
0.1	23 NOV 2001	ALL	FIRST DRAFT OF SECURITY POLICY
1.0	19 FEB 2002	ALL	INCORPORATION OF COMMENTS
1.1	28 NOV 2002	ALL	INCORPORATION OF COMMENTS FROM CMT (DOMUS) LAB
1.1B	16 DEC 2002	ALL	INCORPORATION OF FURTHER COMMENTS FROM CMT (DOMUS) LAB
1.1C	08 JAN 2003	3.2	INFO ADDED FOR SIMPLE ACCESS DSA KEY

1. Introduction

1.1 Company Overview

Simple Access Inc. (formerly Galea Secured Networks Inc) is a growing and innovative solutions manufacturer, based in the high-tech startup belt of Boston, with engineering offices outside Montreal, Québec that designs, builds and supplies high capacity security products for the new tougher applications of the Internet and Intranet, including:

- Sever Protector 100 (SP-100);
- Web Protector 100 (WP-100);
- SSL Accelerator Module (SSL-100 SDK); and
- Virtual Private Network 100 (VPN-100).

In short, Simple Access designs hardware and software firewalls, SSL and VPN solutions specifically designed for OEMing into e-commerce solutions. Each of their products provides a variety of security safeguard services to a customer's e-commerce implementation, enduring the integrity and confidentiality of transactions, and authentication of a variety of entities within the architecture. Although not currently validated, Simple Access has designed these products to meet international security standards.

1.2 Product Overview

The Simple Access e-Commerce SSL-100 SDK is a multi-chip embedded cryptographic module. It provides your organization with a very cost-effective solution to optimize the performance of your e-Commerce Web Servers. A single Simple Access e-Commerce SSL-100 SDK allows web servers to achieve sustained throughput of up to1600 SSL connection set-up per second with RSA 1024-bit operands. The Simple Access e-Commerce SSL-100 SDK offloads both SSL processing and the huge cryptographic computations from the server, freeing the CPU to respond to e-Commerce transactions immediately. This solution avoids slow response times, dropped connections and failed transactions, and therefore maintains the users loyalty to your Web site.

The Simple Access e- Commerce SSL-100 SDK advanced design allows Simple Access Inc. to maintain itself at the forefront of the competition in terms of price and performance. The cryptographic functions are located in the SSL-100 SDK module and can be easily replaced with a new generation of cryptographic chip whenever it is required. Customers obtain a good return of investment (ROI) by only upgrading the SSL-100 SDK plug-in module of a Simple Access e-Commerce SSL-100 SDK card and instantly increase its performance.

The Simple Access e-Commerce SSL-100 SDK offers flexibility and scalability to Linux, Windows, and other environments. It also comes in two speed configurations:

- A single engine SSL-100 SDK (can handle up to 800 new sessions per second), and
- A dual engine SSL-100 SDK (can handle up to1600 new sessions per second).

The FIPS 140-1 cryptographic validation testing was performed on the dual engine SSL-100 SDK, on a Red Hat LINUX v7.1.

Multiple Simple Access e-Commerce SSL-100 SDK cards can easily be installed in each Web Server to scale linearly. The load balancing capability of the Simple Access drivers allows it to smoothly balance the Web Server requests among the installed Simple Access e-Commerce SSL-100 SDK cards. The result is that your customers never have to wait for a secure connection. The Simple Access e-Commerce SSL-100 SDK offloads and accelerates the public-key cryptographic functions of SSL and other widely used security protocols from the Web server's CPU.

Key Benefits of the SSL-100 SDK include:

- Accelerates secure e-Commerce transactions,
- Ensures Web Server performance,
- Enables high throughput and fast response time,
- Easy to install, and
- Integrates with cryptographic developer tool kits and secure applications.

1.3 FIPS & the CMV Program

The Federal Information Processing Standards Publication standard 140-1 (FIPS 140-1) "*Security Requirements for Cryptographic Modules"* was approved on 4 January 1994. As the standard gained exposure, it was adopted by the Canadian Government in 1995, soon followed internally, and now is the defacto standard for cyrpto module security. This standard is to be used by federal organizations when these organizations specify that cryptographic-based security systems are to be used to provide protection for sensitive, non-classified data and specifies the security requirements that are to be satisfied by a cryptographic module.

The US and Canadian Governments, through their National Institute of Standards and Technology (NIST), and Communications Security Establishment (CSE) respectively, created the Cryptographic Module Validation (CMV) Program in July 1995. This program was established in order to test products against a certain Federal Information Processing Standards relating to cryptographic modules and algorithms, in order to verify that products incorporated into US and Canadian government networks were formally validated against the FIPS cryptographic requirements. Third-party laboratories, accredited as Cryptographic Module Testing (CMT) laboratories by the National Voluntary Laboratory Accreditation Program (NVLAP), handle all of the tests under the CMV Program, with final approval granted directly and jointly by NIST and CSE. There are currently six accredited CMT labs in North America and one in Europe.

The SSL-100 SDK is intended to meet FIPS 140-1 Level 2. This document provides details on the required elements of the "Security Policy" related assertions, primarily AS01.07, and Appendix A ("Cryptographic Module Security Policy") to the FIPS 140-1 DTRs.

1.4 Reference Documents

The following references were used as part of this project:

P/R1.	"Federal Information Processing Standards Publication 140-1: Security Requirements for Cryptographic Modules", NIST/US Department of Commerce, 11 January 1994;
P/R2.	"Derived Test Requirements for FIPS Pub 140-1: Security Requirements for Cryptographic Modules", NIST, March 1995;
P/R3.	"Implementation Guidance for FIPS PUB 140-1 and the Cryptographic Module Validation Program", NIST/CSE, 26 July 2001;
G/R1.	"Software Architecture Document #S100-102-01-Sec-SSL Apache PKSC#11" version 1.0, 2001/01/04;
G/R2.	"Software Architecture Document – Boot ROM" version 1.0;
G/R3.	"Inception Document – Project Mercury, Secured Network #M100-101-01-SEC", 25 Oct 2000;
G/R4.	Broadcom BCM582 e-Commerce Processor Web Promotional Material;

G/R5. Broadcom BCM582 Product Brief;

G/R6. Broadcom BCM582 Data Sheet, Document #5820-DS03-R'

G/R7. "SSL 100 SDK Module User Manual" Draft;

G/R8. "Software Architecture Document #S100-102-01 Embedded SSL", 4/10/2001; and

G/R9. "Software Architecture Document – Secured Key Management", version 1.0.

2. Cryptographic Module Definition

2.1 Technical Specifications

The following table provided descriptions of the features and technical specifications of the SSL-100 SDK:

Table 2.1-1: SSL-100 SDK Dual Engine Technical Specifications

SSL–100 SDK MODULE	
Features	
Transaction per second	Processing 1600 1024-bits private key RSA per second
Operating Systems	Linux Red Hat 7.1, Kernel level 2.42
Web Server	Apache
APIs and Toolkits	OpenSSL/SSLeay
Protocols Supported	SSL 2.0 (Secure Sockets Layer V2) [1]
	SSL 3.0 (Secure Sockets Layer V3) [2]
	TLS 1.0 (Transport Layer Security V1)
Cryptographic Functions	
Available Cryptographic Algorithms	Triple DES, TCBC, DES (using Tripe DES running in Single DES[3] configuration), SHA-1, DSA, RSA (PKCS#1), MD5 (in non-FIPS mode).
Modular exponentiation functions	DH, RSA, DSA
RSA modulus lengths	512-bit, 1024-bit, 2048-bit
RSA 1024-bit private-key throughput	1600 operation/sec
DH 1024-bit Modulo & 180b Exp-throughput	2400 key setup/sec
Random Number Generation	2 Mbits/sec
Physical	
Card	Small outline Dual Inline Memory Module (SODIMM)
Power consumption	2.5 A at 5 Volts
Operation temperature	0 to +50°C
Storage temperature	-40°C to +85°C
Humidity	5% to 95% non condensing

[1] SSL 2.0 is not currently FIPS approved.
[2] SSL 3.0 is not currently FIPS approved.
[3] Disclaimer: As specified in FIPS PUB 46-3, the use of single DES is no longer recommended and is currently permitted for legacy systems only.

2.2 Platform Requirements

The SSL-100 SDK can be used with Windows, Linux, or Solaris operating systems. The minimum requirements for hardware to support each of these is as follows:

- **Operating Systems**: Windows NT or 2000; and
- **Hardware**: Celeron 500Mhz equipped with 64MB of SDRAM, with 10 Mb HD space.

FIPS 140-1 Conformance testing was performed using the Red Hat LINUX v7.1 operating system.

2.3 Block Diagram

Figure 2.3-1, below, illustrates the components of the SSL-100 SDK. For the purposes of the FIPS 140-1 Level 2 evaluation, the cryptographic boundary includes, as indicated in Figure 2.3-1, everything on the SSL-100 SDK board, with the exception of the Host PCI Connector.

The Host PCI Connector was excluded from the cryptographic boundary, as the application of the required physical security mechanisms (tamper evident opaque coating) was not possible without inhibiting the functionality of the component.

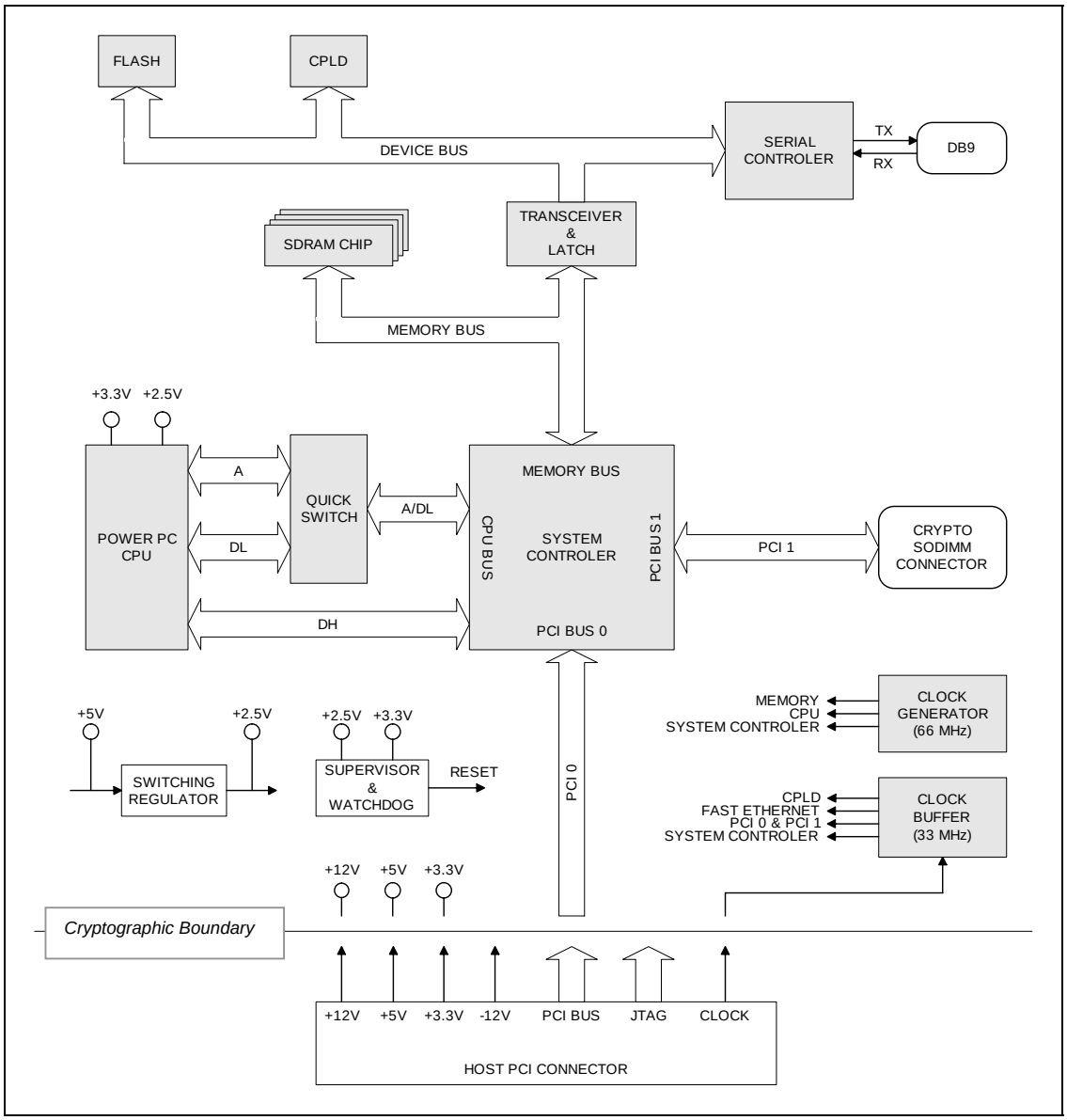

Figure 2.3-1 SSL-100 SDK Block Diagram

3. Roles & Services

3.1 Identification and Authentication

The FIPS 140-1 standard states that at a minimum, a cryptomodule must support at least the following :

- **User role:** The role assumed by an authorized user obtaining security services, performing cryptographic operations, or other authorized functions.

- **Crypto-officer role:** The role assumed by an authorized crypto officer performing a set of cryptographic initialization or management functions (e.g., cryptographic key and parameter entry, cryptographic key cataloguing, audit functions, and alarm resetting).

These roles do not necessarily need to be explicit, and may be implied based on the functions performed by the user.

The SSL-100 SDK uses a combination of identity-based and role-based authentication schemes.

A predefined role-based account, the "SO" is included in the board[4]. Using this management account, identity-based accounts can be created. These identity-based accounts are then assigned roles (called "privilege classes"), which define what type of transactions an account will be performing. The possible privilege classes include:

- Asymmetric,
- Symmetric, and
- Management.

The predefined "SO" account is assigned to the "Management" privilege class. Identity-based accounts may be assigned one, two, or all of the above privilege classes.

Thus, according to the standard, the SSL-100 SDK has the following:

- Crypto-officer role ("SO") (cryptographic officer of management privilege class); and
- System users of symmetric or asymmetric privilege class.

The services that each of these roles can perform is as follows:

Table 3.1-1 SSL-100 SKD Roles

Role	Services
User	Authentication, symmetric key generation, key destruction, symmetric encryption/decryption, hashing, MAC generation/verification, self-test, login
Cryptographic Officer	Configuration of cryptographic services (e.g. set/generate initialization vector); Key pair generation, asymmetric encryption/decryption (sign/verify), password change

3.2 Security Relevant Data Items

The following are the security relevant data items within the SSL-100 SDK module:

- Simple Access Public Key (DSA),
- Key Backup Key (3-DES CBC algorithm with a triple length 3-DES key and a random IV. Both are randomly generated onboard)
- Key Signing Key (3-DES; note that it is impossible to export the KBK and KSK key otherwise than by using a perfect secret sharing scheme[5])

[4] The pass phrase for the predefined SO account is provided directly to the end user by the vendor.
[5] A share secret scheme is perfect if when n shared secrets are needed to reconstruct the secret then the knowledge of any n-1of these n secrets should not yields any information on the secrets itself. One implication of a perfect share secret scheme is that the size of the each shares secret must be at least the size of the secret itself.

- Master Authentication Key (3-DES; the value of the Master Authentication Key is determined from a passphrase using a derivation mechanism)
- Master Authentication Key Passphrase (DES)
- Symmetric Keys (SHA-1, DES, 3DES and MD5 {*in Non-FIPS mode only*}) and
- Asymmetric Key Pairs (DSA, Diffie-Hellman, DSA)

The cryptographic module provides the following FIPS approved basic services:

- Cryptographic data hashing using SHA1 and MD5 (in non-FIPS approved mode);
- Bulk data encryption and decryption using 3DES and DES (using 3DES running in Single DES mode);
- MAC calculation through HMAC using SHA1 or MD5 (in non-FIPS approved mode); and
- Signing and signature verification using DSA and RSA (in non-FIPS approved mode).

The module also provides the following services:

- Key wrapping and unwrapping using RSA.
- Key agreement using Diffie-Hellman,
- User authentication (as described above).
- Random number generation using a FIPS 180-2-compliant hardware -based generator.

3.3 Services & Correspondence Mapping

The following table maps roles to services to SRDI.

Table 3.3-1: Roles to Services Correspondence Mapping

ROLE	SERVICE	SRDI
System User – Asymmetric Privilege Class	RSA Sign	User Account Data Asymetric Key Pair
	RSA Verify	User Account Data Asymetric Key Pair
	RSA Encrypt	User Account Data Asymetric Key Pair
	RSA Decrypt	User Account Data Asymetric Key Pair
	RSA Key Generation	User Account Data Asymetric Key Pair
	Diffie Hellman Key Generation	User Account Data Asymetric Key Pair
	Diffie Hellman Share Generation	User Account Data Asymetric Key Pair
	DSA Sign	User Account Data Asymetric Key Pair
	DSA Verify	User Account Data Asymetric Key Pair
	DSA Key Generation	User Account Data Asymetric Key Pair

ROLE	SERVICE	SRDI
System User – Symmetric Privilege Class	SHA1 Key Generation	User Account Data Symmetric Key
	MD5 Key Generation (non-FIPS mode only)	User Account Data Symmetric Key
	SHA1 Hash	User Account Data Symmetric Key
	MD5 Hash (non-FIPS mode only)	User Account Data Symmetric Key
	Triple DES/DES Key Generation	User Account Data Symmetric Key
	Triple DES /DES Encrypt	User Account Data Symmetric Key
Cryptographic Officer – Management Privilege Class	Module Initialization – Load Image	Simple Access Key Master Authentication Key Master Authentication Key Passphrase
	Execute Self Tests	Master Authentication Key Master Authentication Key Passphrase
	Show Status	Master Authentication Key Master Authentication Key Passphrase
	Master Password Change	Master Authentication Key Master Authentication Key Passphrase
	Add User	Master Authentication Key Master Authentication Key Passphrase User Data
	Delete User	Master Authentication Key Master Authentication Key Passphrase User Data
	Get User Information	Master Authentication Key Master Authentication Key Passphrase User Data
	Set User Privilege	Master Authentication Key Master Authentication Key Passphrase User Data
	Import Key	Master Authentication Key Master Authentication Key Passphrase User Data
	Export/Backup Key	Master Authentication Key Master Authentication Key Passphrase User Data Key Backup Key Key Signing Key

ROLE	SERVICE	SRDI
	Delete Key	Master Authentication Key Master Authentication Key Passphrase

4. Security Policy

4.1 I&A Policy

The SSL-100 SDK enforces a combination identity and role-based authentication. This policy requires any user attempting to execute any SSL-100 SDK service or access any SRDI first be authenticated to the module, using an UserID and Password, before being granted access to the cryptographic module.

4.2 Access Control Policy

Access to SSL-100 SDK services and SRDIs is restricted by assigned privilege class as per Table 3.3-1 in Section 3.3.

Users do not have access to any SRDIs not included within the list of given services for a given role.

4.3 Module Zeroization

Zeroization of the module can be performed by having the Security Officer (SO) call the function:

gsnHwCrypto_DeleteAllKey()

When this function is invoked, all keys, permanent and session keys, are erased from memory by setting the corresponding memory to 0x00.

This function also deletes the Master Authentication Key.

5. Physical Security Requirements

The SSL-100 SDK is coated with an opaque, tamper-evident potting. This coating is designed to detect any attempts by providing clear evidence of any such tampering. It will also prevent tampering attempts and will cause components to be damaged if tampering is attempted.

The potting coating covers the entire SSL-100 SDK module, except for the Host PCI connector and the heat fan.

Heat sinks (one each) are used for heat dissipation for the two Broadcom chips. Heat Dissipation for the microprocessor chip is performed using a fan.

6. Installation Guide

6.1 Hardware Installation

To install the SSL-100 SDK, follow these steps:

1. Shut down the computer.
2. Remove the computer's cover. There is no need to unplug the computer if it is not running. Leave the computer plugged, its chassis remains electrically grounded. This facilitates your work in later steps.
3. Choose an empty PCI slot. The PCI bus of the *SP-100* is 32-bit wide. The slot should be chosen in such a way that you could insert the card without bending it. The card must not be in contact with any moving parts, or close to a heat sink.
4. Remove the cover bracket matching the chosen PCI slot, usually by unscrewing the screw that secures the bracket to the computer frame.
5. Insert the *SSL-100 SDK* into the SODIMM slot 1 of SP-100 PCI card. Push the SSL-100 SDK card into the slot until it is firmly seated. Secure the card with the two brackets on the side.

WARNING: *The SSL-100 SDK can be damaged by electrostatic discharges. When manipulating, an easy way to prevent this is to touch an electrically grounded material, such as the chassis of your plugged, but still turned OFF computer. Being in touch with a grounded material, your body does not carry any electrostatic charges.*

Appendix A - Definitions

Automated key distribution:

the distribution of cryptographic keys, usually in encrypted form, using electronic means, such as a computer network (e.g., down-line key loading, the automated key distribution protocols of ANSI X9.17).

Compromise:

the unauthorized disclosure, modification, substitution or use of sensitive data (including plaintext cryptographic keys and other critical security parameters).

Confidentiality:

the property that sensitive information is not disclosed to unauthorized individuals, entities or processes.

Control information:

information that is entered into a cryptographic module for the purposes of directing the operation of the module.

Critical security parameters:

security-related information (e.g., cryptographic keys, authentication data such as passwords and PINs) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Cryptographic boundary:

an explicitly defined contiguous perimeter that establishes the physical bounds of a cryptographic module.

Cryptographic key (key):

a parameter used in conjunction with a cryptographic algorithm that determines:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a data authentication code (DAC) computed from data.

Cryptographic key component (key component):

a parameter which is combined via a bit-wise exclusive-OR operation with one or more other identically sized key component(s) to form a plaintext cryptographic key.

Cryptographic module:

the set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

Cryptographic module security policy:

a precise specification of the security rules under which a cryptographic module must operate, including the security rules derived from the requirements of this standard and the additional security rules imposed by the manufacturer.

Data authentication code (DAC):

a cryptographic checksum, based on DES (see FIPS PUB 113); also known as a Message Authentication Code (MAC) in ANSI standards.

Data key:

a cryptographic key which is used to cryptographically process data (e.g., encrypt, decrypt, sign, authenticate).

Data path:

the physical or logical route over which data passes; a physical data path may be shared by multiple logical data paths.

Digital signature:

a non-forgeable transformation of data that allows the proof of the source (with non-repudiation) and the verification of the integrity of that data.

Electromagnetic compatibility (EMC):

the ability of electronic systems to operate in their intended environments without suffering an unacceptable degradation of the performance as a result of unintentional electromagnetic radiation or response.

Electromagnetic interference (EMI):

electromagnetic phenomena which either directly or indirectly can contribute to a degradation in the performance of an electronic system.

Environmental failure protection (EFP):

the use of features designed to protect against a compromise of the security of a cryptographic module due to environmental conditions or fluctuations outside of the module's normal operating range.

Environmental failure testing (EFT):

the use of testing to provide a reasonable assurance that a cryptographic module will not be affected by environmental conditions or fluctuations outside of the module's normal operating range in a manner that can compromise the security of the module.

Electronic key entry:

the entry of cryptographic keys into a cryptographic module in electronic form using a key loading device. The user entering the key may have no knowledge of the value of the key being entered.

Encrypted key (ciphertext key):

a cryptographic key that has been encrypted with a key encrypting key, a PIN or a password in order to disguise the value of the underlying plaintext key.

Error detection code (EDC):

a code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.

Finite state machine (FSM):

a mathematical model of a sequential machine which is comprised of a finite set of states, a finite set of inputs, a finite set of outputs, a mapping from the sets of inputs and states into the set of states (i.e., state transitions), and a mapping from the sets of inputs and states onto the set of outputs (i.e., an output function).

FIPS approved security method:

a security method (e.g., cryptographic algorithm, cryptographic key generation algorithm or key distribution technique, authentication technique, or evaluation criteria) that is either a) specified in a FIPS, or b) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.

Firmware:

the programs and data (i.e., software) permanently stored in hardware (e.g., in ROM, PROM, or EPROM) such that the programs and data cannot be dynamically written or modified during execution. Programs and data stored in EEPROM are considered as software.

Hardware:

the physical equipment used to process programs and data in a cryptographic module.

Initialization vector (IV):

a vector used in defining the starting point of an encryption process within a cryptographic algorithm (e.g., the DES Cipher Block Chaining (CBC) mode of operation).

Integrity:

the property that sensitive data has not been modified or deleted in an unauthorized and undetected manner.

Interface:

a logical section of a cryptographic module that defines a set of entry or exit points that provide access to the module, including information flow or physical access.

Input data:

information that is entered into a cryptographic module for the purposes of transformation or computation.

Key encrypting key:

a cryptographic key that is used for the encryption or decryption of other keys.

Key loader:

a self-contained unit which is capable of storing at least one plaintext or encrypted cryptographic key or key component which can be transferred, upon request, into a cryptographic module.

Key management:

the activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs, counters) during the entire life cycle of the keys, including their generation, storage, distribution, entry and use, deletion or destruction, and archiving.

Manual key distribution:

the distribution of cryptographic keys, often in a plaintext form requiring physical protection, but using a non-electronic means, such as a bonded courier.

Manual key entry:

the entry of cryptographic keys into a cryptographic module from a printed form, using devices such as buttons, thumb wheels or a keyboard.

Microcode:

the elementary computer instructions that correspond to an executable program instruction.

Operator:

an individual accessing a cryptographic module, either directly or indirectly via a process operating on his or her behalf, regardless of the specific role the individual assumes.

Output data:

information that is to be output from a cryptographic module that has resulted from a transformation or computation in the module.

Password:

a string of characters used to authenticate an identity or to verify access authorization.

Personal Identification Number (PIN):

a 4 to 12 character alphanumeric code or password used to authenticate an identity, commonly used in banking applications.

Physical protection:

the safeguarding of a cryptographic module or of cryptographic keys or other critical security parameters using physical means.

PIN:

see Personal Identification Number.

Plaintext key:

an unencrypted cryptographic key which is used in its current form.

Port:

a functional unit of a cryptographic module through which data or signals can enter or exit the module. Physically separate ports do not share the same physical pin or wire.

Private key:

a cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public.

Public key:

a cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public.

Public key certificate:

a set of data that unambiguously identifies an entity, contains the entity's public key, and is digitally signed by a trusted party.

Public key (asymmetric) cryptographic algorithm:

a cryptographic algorithm that uses two related keys, a public key and a private key; the two keys have the property that, given the public key, it is computationally infeasible to derive the private key.

Secret key:

a cryptographic key used with a secret key cryptographic algorithm, uniquely associated with one or more entities, and which shall not be made public. The use of the term "secret" in this context does not imply a classification level, rather the term implies the need to protect the key from disclosure or substitution.

Secret key (symmetric) cryptographic algorithm:

a cryptographic algorithm that uses a single, secret key for both encryption and decryption.

Security policy:

see Cryptographic Module Security Policy.

Software:

the programs, and possibly associated data that can be dynamically written and modified.

Split knowledge:

a condition under which two or more entities separately have key components which individually convey no knowledge of the plaintext key which will be produced when the key components are combined in the cryptographic module.

Status information:

information that is output from a cryptographic module for the purposes of indicating certain operational characteristics or states of the module.

System software:

the special software (e.g., operating system, compilers or utility programs) designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system, programs, and data.

Trusted path:

a mechanism by which a person or process can communicate directly with a cryptographic module and which can only be activated by the person, process or module, and cannot be imitated by untrusted software within the module.

Zeroization:

a method of erasing electronically stored data by altering the contents of the data storage so as to prevent the recovery of the data.

Appendix B - Abbreviations

ANSI	American National Standards Institute
ATM	Automated Teller Machine
CBC	Cipher Block Chaining
DAC	Data Authentication Code
DES	Data Encryption Standard
DOC	Department of Commerce
DOD	Department of Defense
EDC	Error Detection Code
EFP	Environmental Failure Protection
EFT	Environmental Failure Testing
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EPROM	Erasable Programmable Read-Only Memory
E^2PROM	Electronically-Erasable Programmable Read-Only Memory
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
FIPS PUB	FIPS Publication
FSM	Finite State Machine
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITSEC	Information Technology Security Evaluation Criteria
IV	Initialization Vector
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MAC	Message Authentication Code
NBS	National Bureau of Standards
NIST	National Institute of Standards and Technology (formerly the National Bureau of Standards)
NSA	National Security Agency
PC	Personal Computer
PIN	Personal Identification Number
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
ROM	Read-Only Memory
SRDI	Security Relevant Data Item
TCB	Trusted Computing Base
TCSEC	Trusted Computer System Evaluation Criteria