ValiCert Security Module 1.0
Security Policy

Revision 11/22/02

1.1 Introduction

This security policy was prepared as part of FIPS 140-1 certification of the ValiCert Security Module. The ValiCert Security Module meets all FIPS 140-1 level 1 requirements. This security policy describes how the ValiCert Security Module meets these requirements, and how it can be operated in a secure fashion.

The security policy defined in this document specifies the security rules under which the ValiCert Cryptographic Module **must** operate. It includes the rules derived from the security requirements of the FIPS PUB 140-1 standard, and additional security rules imposed by **ValiCert**. These rules define within which role(s), and under what circumstances (when performing which services), an operator is allowed to maintain or disclose each security relevant data item of the cryptographic module. For more detailed information about the ValiCert Security Module and all other ValiCert products that are based on the ValiCert Security Module, please visit the ValiCert web site at http://www.valicert.com. The web site contains non-technical descriptions of ValiCert products, product functionalities, product offerings and much more.

1.2 Specification of Security Policy

1.2.1 ValiCert Security Module

The ValiCert Security Module is a software cryptographic module implemented as a set of libraries. These libraries represent the cryptographic module, which present the same application programmer interface (**API**) to all other ValiCert products.

Supported FIPS approved algorithms are DES, Triple-DES, SHA-1, and RSA. Other algorithms that also are supported in a non-FIPS mode are MD2, MD5, RC2, and RC4. The ValiCert Security Module for the purpose of this FIPS testing is tested on Microsoft Windows 2000 Server and Sun Solaris 2.8.

1.2.2 FIPS 140 -1 Security Level

The table below lists the security levels to which the ValiCert Security Module has been certified.

Security Requirements Section	Level
Cryptographic Module	1
Module Interfaces	1
Roles and Services	1
Finite State Machine	1
Physical Security	1
Software Security	1

Operating System Security	1
Key Management	1
Cryptographic Algorithms	1
EMI/EMC	1
Self-Tests	1

1.2.3 Cryptographic Boundary

The physical cryptographic boundary is defined to be the computer case that the ValiCert Cryptographic Module is installed on. The module is therefore viewed in FIPS 140-1 terms as "multi-chip standalone module". The software cryptographic boundary is defined to be a subset of the ValiCert Cryptographic Module libraries. It does not include the functions associated with algorithms that are not used in a non-FIPS mode as well as functions that do not effect the security of the module. The cryptographic boundary is indicated in the cryptographic boundary diagram below (Figure 1):

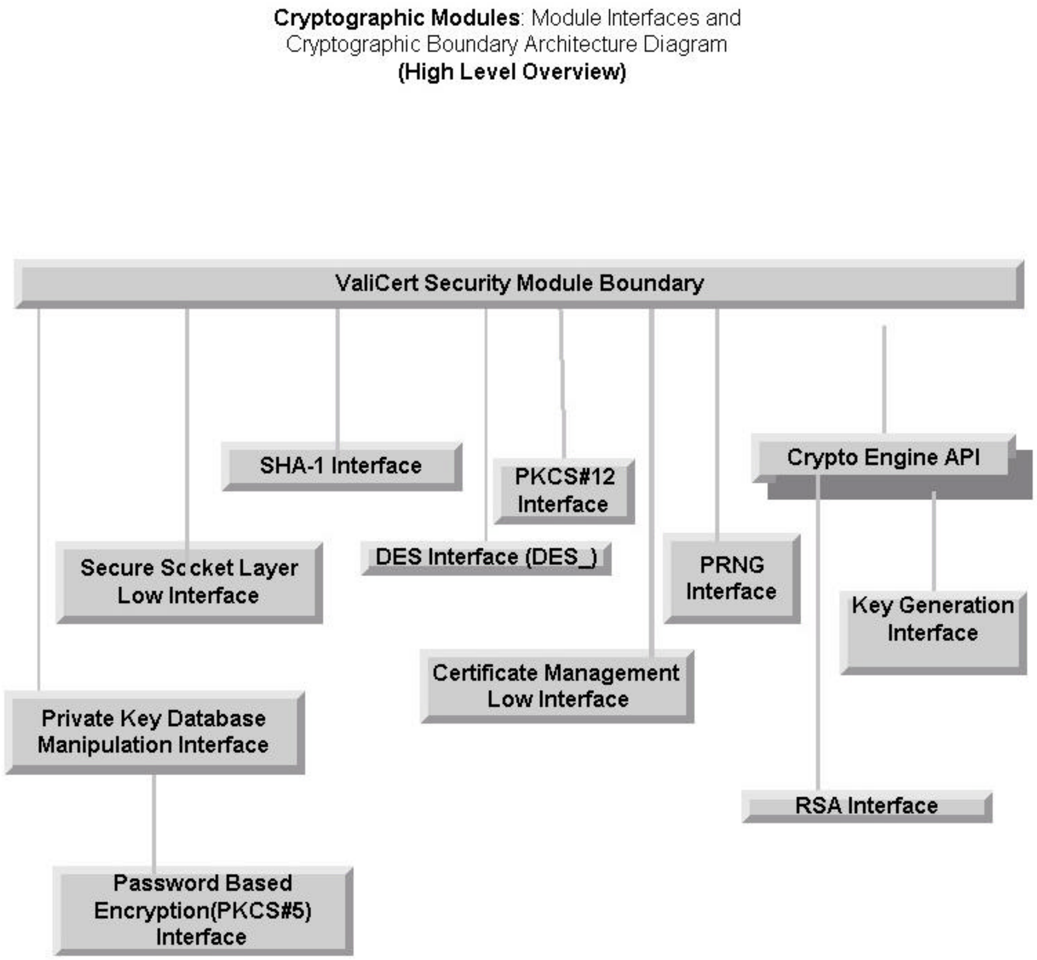
Cryptographic Modules: Module Interfaces and
Cryptographic Boundary Architecture Diagram
(High Level Overview)
ValiCert Security Module Boundary
SHA-1 Interface
PKCS#12 Interface
Crypto Engine API
Secure Sccket Layer Low Interface
DES Interface (DES_)
PRNG Interface
Key Generation Interface
Certificate Management Low Interface
Private Key Database Manipulation Interface
RSA Interface
Password Based Encryption(PKCS#5) Interface

Figure 1: Cryptographic Boundary

1.2.4 Roles and Services

1.2.4.1 Roles

ValiCert's Security Module utilizes a single user approach -- this user refered as **ValiCert User.** The module has a User role and a Cryptoofficer role. Both roles have same set of services. When the ValiCert User performs any of the modules services he is in both the roles of a Cryptoofficer and a User.

ValiCert Security Module does not have any service that is associated with maintenance and hence does not have a maintenance role.

1.2.4.2 Authentication

For Level 1, a cryptomodule is not required to employ authentication mechanisms to control access to the module. Hence the ValiCert Security Module does not employ any authentication mechanism to control access to the module.

1.2.4.3 Multiple Concurrent Operator Roles and Services

Since ValiCert applications always operate under a single user, the ValiCert Security Module does not have multiple concurrent operators at a time. The module does not support bypass mode.

1.2.4.4 Services

Services in FIPS mode may include, but not be limited to, the following:

Cryptographic operations such as encryption, decryption, hashing, digital signature generation, digital signature verification.

Key management operations such as key and parameter entry, key generation, key output , key archiving, key zeroization, and other key management functions.

Performance of self tests, such as cryptographic algorithm tests, continous random number generator tests, or any additional tests.

In each of the following services, since there is only one user, the user has access to **ALL** the services mediated by the application (for both client and server products).

Table: Services

Name of Service	Description of Service in Terms of Routines
Certificate Storage and Retrieval	This service consists of the following routines to perform certificate storage and retrieval: PEM_read_bio_X509, PEM_read_bio_X509_CRL, PEM_write_bio_X509, PEM_write_bio_X509_CRL, X509_free, X509_LOOKUP_load_file, X509_STORE_add_cert(), X509_STORE_free, X509_LOOKUP_by_subject, X509_LOOKUP_by_issuer_serial, X509_LOOKUP_by_fingerprint..

Digital Signatures	This service consists of the following routines to perform RSA signature generation, verification and for key encryption/agreement:RSA_sign(), RSA_verify(), RSA_public_encrypt(), RSA_private_encrypt(), RSA_public_decrypt(), RSA_private_decrypt().
Encryption/ Decryption	This service consists of the following routines used to perform DES: des-ecb-encrypt() and des-cbc-encrypt() This service consists of the following routines used to perform Triple-DES: des-ede3-cbc-encrypt() and des-ecb3-encrypt()
Hashing	This service consists of the following routines used to perform SHA-1 hashing: SHA1_Init, SHA1_Update, SHA1_Final
Key Generation	This service is utilized to perform RSA key generation and DES key generation and consists of the following routines: RSA_generate_key(), des_random_key().
PKCS #12 Personal Information Exchange	The PKCS #12 API will specify a standard interface based upon the PKCS #12 standard which allows this service to be used to exchange data such as private keys and certificates between two parties and consists of the following routines: PKCS12_create(), PKCS12_parse() can be used to parse a PKCS12 object, d2i_PKCS12() can be used to import a Distinguished Encoding Rules (DER) PKCS12 object and i2d_PKCS12() can be used to export a PKCS12 object to a DER encoded form. PKCS12_free is used to free a PKCS12 object.
Private Key Storage and Retrieval	This service is utilized to perform RSA private key storage and retrieval and consists of the following routines: PEM_read_bio_RSAPrivatekey() and PEM_write_bio_RSAPrivatekey(), RSA_free( ) is used to free an RSA key.
Pseudorandom Number Generation	This service consists of the following routines used for pseudorandom number generation: RAND_bytes()
TLS Session ID Cache (Secret Management)	This service consists of the following routines used to perform session ID cache management: ssl_get_new_session(), SSL_CTX_add_session(), SSl_CTX_remove_session(), SSL_SESSION_free, tls1_generate_master_secret.
Show Status	This service outputs the current state of the module. This service consists of the return value of each of the API calls.
Self Test	This service runs the complete set of power-up self tests. This service is run when the module is unloaded and reloaded.
Zeroization	RSA keys are zeriozed by the function RSA_free() and TLS Session ID cache is zeroized using SSL_SESSION_free(). To zeroize the key used by the FIPS_verify_dac() and

	FIPS_create_dat(), the operator would need to delete the binary in which the key is contained.

1.2.5 Access Control Policy

The access control policy enforced by the cryptographic module lets the operator and testers to know what **security relevant data items** the **operator** has access to while performing a **service**, and the **modes** of access he or she has to these data items. Also, the testers and operator must be able to know if and how the kinds of data items accessible changes when the service is invoked from each role in which it can be invoked.

1.2.5.1 Security Relevant Data Items

Security relevant data items consist of data types used for Certificate Storage and Retrieval, Digital Signatures, Encryption/Decryption, Generic Containers, Hashing, Key Generation, PKCS #12 Personal Information Exchange, Private Key Storage and Retrieval, Pseudorandom Number Generation, and TLS Session ID Cache (Secret Management).

All security relevant data items are identified by service, type, name, and description in the following table:

Table: Security Relevant Data Items

Service	Name of Data Item	Description of Data Iter
Certificate Storage and Retrieval	X509_STORE, X509_LOOKUP,X509, X509_CRL	X509_STORE holds the tables with certs and uses X50 certificates. X509 is a certificate object and X509_CRL
Digital Signatures	RSA	This data structure contains the private/public key for signing/encryption/verifying/decryption when using RSA
Encryption/ Decryption	des_key_schedule ks(1/2/3) des_cblock iv	The DES key(s) used to encrypt/decrypt data. The initilization vector used for CBC encryption/decrypt
Hashing	SHA_CTX	The structure representing the context of a SHA-1 hash c to performing a SHA-1 hash.
Key Generation	RSA des_cblock	The RSA_generate_key() function returns this data struc The data structure that contains the des key.
PKCS #12 Personal Information Exchange	PKCS12, X509_CERT, X509_STORE, EVP_PKEY, passphrase	The PKCS12 object that is created/managed by this int The X509 _CERT that is added to the PKCS12 object The X509_STORE of ca certificates that are added to The private key for the X509_CERT that gets added to The passphrase used to protect the PKCS12 object.
Private Key Storage and Retrieval	RSA	The structure representing the private key.
Pseudorandom Number	buf	The seed value used to seed the random number genera

Generation		
TLS Session ID Cache (Secret Management)	SSL_CTX SSL_SESSION, SSL	SSL_CTX is the structure that contains the context infor SSL_SESSION is the structure containing the data relev SSL is an internal SSL structure.

1.2.4.2 Service Relationships to Security Relevant Data Items Matrix

Table: Service Routine to Security Relevant Data Items Matrix

Service	Service Routine	Security Relevant Data Item	Read Access	Write Access
Certificate Storage and Retrieval	X509_STORE_add_cert(), X509_STORE_free	X509_STORE		X
	X509_LOOKUP_by_subject, X509_LOOKUP_by_issuer_serial, X509_LOOKUP_by_fingerprint.	X509_LOOKUP	X	
	X509_LOOKUP_load_file	X509_LOOKUP	X	
	PEM_read_bio_X509	X509	X	
	PEM_write_bio_X509, X509_free	X509		X
	PEM_read_bio_X509_CRL	X509_CRL	X	
	PEM_write_bio_X509_CRL	X509_CRL		X
Digital Signatures	RSA_sign(), RSA_verify(), RSA_public_encrypt(), RSA_private_encrypt(), RSA_public_decrypt(), RSA_private_decrypt().	RSA	X	
Encryption/ Decryption	des-ecb-encrypt()	ks	X	
	des-cbc-encrypt()	ks	X	
		IV	X	X

	des-ecb3-encrypt()	ks1, ks2, ks3	X	
	des-ede3-cbc-encrypt()	ks1, ks2, ks3	X	
		IV	X	X
Hashing	SHA1_Init	SHA_CTX		X
	SHA1_Update		X	X
	SHA1_Transform		X	X
	SHA1_Final			X
Key Generation	RSA_generate_key()	RSA	X	X
	des_random_key()	des_cblock	X	X
PKCS #12 Personal Information Exchange	PKCS12_create, PKCS12_parse, i2d_PKCS12, d2i_PKCS12,PKCS12_free	PKCS12 X509_CERT X509_STORE EVP_PKEY passphrase	X X X X X	X
Private Key Storage and Retrieval	PEM_read_bio_RSAPrivatekey()	RSA	X	
	PEM_write_bio_RSAPrivatekey()	RSA		X
Pseudorandom Number Generation	RAND_seed()	buf	X	
TLS Session ID Cache	SSL_CTX_add_session()	SSL_CTX		X
	SSl_CTX_remove_session()	SSL_CTX		X
	SSL_CTX_flush_sessions()	SSL_CTX		X
	ssl_get_new_session()	SSL_SESSION/SSL	X	X
	SSL_SESSION_free()	SSL_SESSION		X
	tls1_generate_master_secret	SSL	X	

1.2.6 Security Policy Rules

Given below are the security policy rules, which will be adhered to by each ValiCert product. The rules are categorized based on the sections of the FIPS 140-1 standard.

Rules:

I Cryptographic Module:

- *ValiCert's cryptographic module consists of a set of binary software libraries compiled for each supported platform and utilized by ALL ValiCert products.*
- *The modules FIPS 140-1 mode of operation is accomplished by using only the FIPS approved algorithms in 1.2.6 Rule IX.*

II Module Interfaces:

- *The ValiCert Security Module ensures that critical errors are not allowed to compromise security.*

III Roles and Services:

- *The single user (ValiCert User) who is in the role of a User and a Cryptoofficer shall have access to ALL the services supplied by the cryptographic module.*

IV Finite State Machine:

- *Even though the FSM does not show transitions from all other service states to Un-Initial State, they are implied.*

V Physical Security:

- *The Valicert Security Module is installed on a workstation that meets standard physical secuirty requirements for level 1 (Standard production quality ICs and all chips have standard passivation techniques).*

VI Software Security:

- *None.*

VII Operating System Security:

- *The authenticity of the cryptographic module is verified using a SHA-1 HMAC.*
- *The operating system shall be configured in a single user usage mode. The operating system shall be configured to allow only a single user to access the cryptographic module.*

VIII Key Management:

All passwords and private keys shall be stored in an encrypted form in secondary storage.

- *The user's password shall act as the key material to encrypt/decrypt private key material.*
- *The cryptographic module shall extract private keys wrapped with a password according to PKCS #12.*
- *The FIPS PUB 140-1 cryptographic module shall utilize pseudorandom number generation as defined via the mechanisms described in Appendix 3.1 of FIPS PUB 186.*
- *SSL master secrets (private key data) shall be protected within the boundary of the cryptographic module (the TLS secure session ID cache shall be considered within the boundary of the cryptographic module).*

IX Cryptographic Algorithms:

- *The FIPS 140-1 cipher suite implemented by the module consists solely of DES and 3DES for encryption/decryption.*
- *PKCS#1 signatures are used with RSA algorithm.*
- *SHA-1 hash is used for secure hash, with cryptographic module HMAC, and in the context of TLS protocol.*
- *The module in a non-FIPS mode can support MD2, MD5, RC2, and RC4.*
- *If any of the functions associated with the algorithms MD2, MD5 and RC2 and RC4 are called then the module will not be in the FIPS Mode; however, in some portions of the TLS key derivation process MD5 is used in a non security relevant manner.*
- As defined in TLS RFC 2246, the following TLS cipher suites are the only ones allowed in the FIPS 140-1 mode of operation:
  - TLS_RSA_WITH_DES_CBC_SHA
  - TLS_RSA_WITH_3DES_EDE_CBC_SHA
- 3DES is the prefered encyrption algorithm used by the module, and DES is used only for legacy and compatibility purposes.

X EMI/EMC:

- *The ValiCert Security Module is installed on a workstation that complies with FCC requirements.*

XI Self-Tests:

- *Upon initialization of the FIPS PUB 140-1 cryptographic module, the following power-up self-tests shall be performed and the following known answer tests will be performed for the following algorithms:*

  (1) RC2-ECB Encrypt/Decrypt,
  (2) RC2-CBC Encrypt/Decrypt,
  (3) RC4 Encrypt/Decrypt,
  (4) DES-ECB Encrypt/Decrypt,

(5) DES-CBC Encrypt/Decrypt,
(6) triple DES-ECB Encrypt/Decrypt,
(7) triple DES-CBC Encrypt/Decrypt,
(8) MD2 Hash,
(9) MD5 Hash,
(10) SHA-1 Hash,
(11) RSA Encrypt,
(12) RSA Decrypt,
(13) RSA Signature,
(14) RSA Signature Verification
(15) Module Software self test using SHA-1 HMAC

- *The conditional tests that are performed are pair-wise consistency test and continuous random number test.*