3e Technologies International, Inc.

FIPS 140-2

Non-Proprietary Security Policy

3e-030-2 Security Server Cryptographic Core (Version 4.0)

June 2011

Table of contents

GLOSSARY OF TERMS

AP	Access Point
CO	Cryptographic Officer
DH	Diffie Hellman
IP	Internet Protocol
EAP	Extensible Authentication Protocol
FIPS	Federal Information Processing Standard
HTTPS	Secure Hyper Text Transport Protocol
LAN	Local Area Network
PRNG	Pseudo Random Number Generator
RSA	Rivest, Shamir, Adleman
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
SHA	Secure Hash Algorithm
SRDI	Security Relevant Data Item
TLS	Transport Layer Security
WLAN	Wireless Local Area Network
CSP	Critical Security Parameter
AES	Advanced Encryption Standard
KEK	Key Encryption Key
HMAC	Keyed-Hash Message Authentication Code
SHA1	Secure Hash Algorithm 1
SHA2	Secure Hash Algorithm 2
PMK	Pairwise Master Key
KAT	Known Answer Test

1. Introduction

1.1. Purpose

This document describes the non-proprietary cryptographic module security policy for 3e Technologies International's 3e-030-2 Security Server Cryptographic Core (Version 4.0), hereafter known as the Security Server. This policy is created to satisfy the requirements of FIPS 140-2 Level 1. This document defines 3eTI's security policy and explains how 3e-030-2 Security Server Cryptographic Core meets the Level 1 FIPS 140-2 requirements.

1.2. Overview

The Security Server Cryptographic Core is a software application that provides Authentication Service for IEEE 802.11i security (WPA2) with TLS-based EAP methods and utilizes RADIUS Key Wrap functionalities.

1.3. Definition

The Security Server Cryptographic Core is a software program that runs in the operational environment on a standard Intel-based computer running Linux or UNIX operating systems. Per section 4.5 of FIPS PUB 140-2, the cryptographic module is a multiple-chip standalone module. However, the cryptographic logic boundary is completely implemented in software and the physical security is solely provided by the host platform.

The physical boundary of the module is the case of the PC (Figure 1). The logical boundary of the module is the executable binaries "3e-provisionserver.exe", and "radiusd.exe". A high level architecture of the 3eTI Security Server Cryptographic Core, including the cryptographic boundary, is shown in Figure 2.

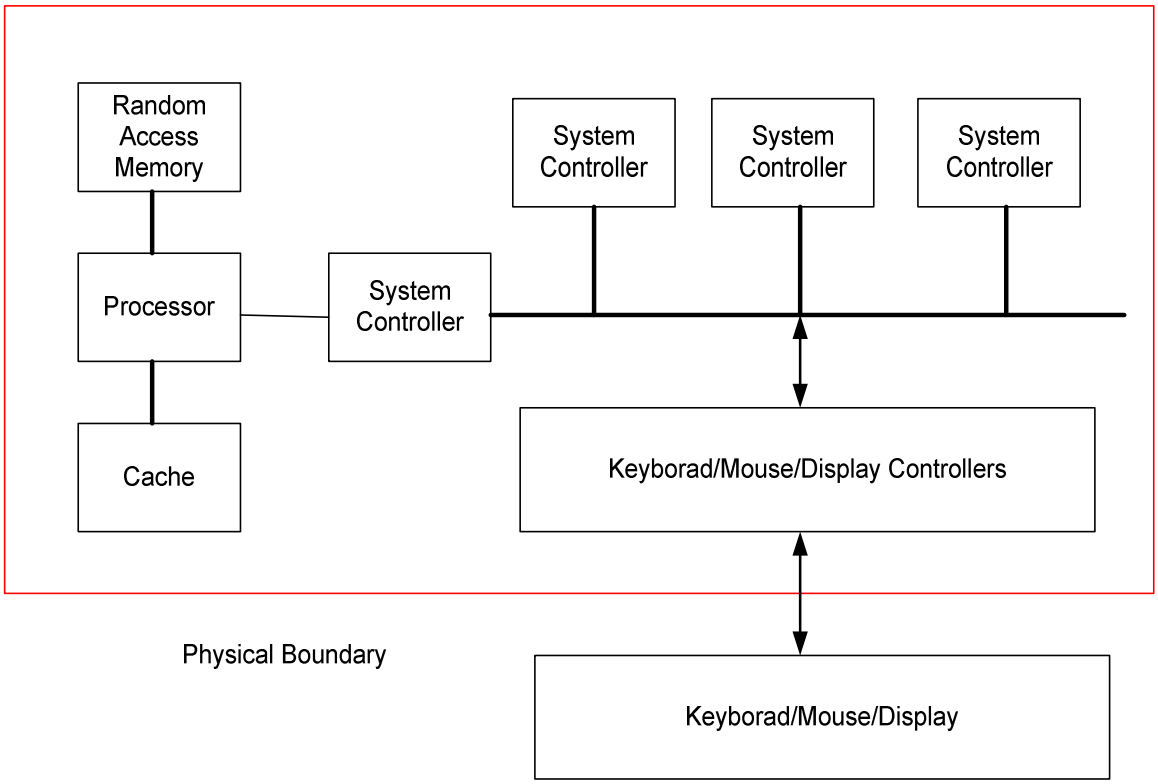

Figure 1: PC Hardware Diagram that contains 3eTI Security Server Application

Software is stored on the hard drive of the system, and loaded into random access memory for execution. The processor component shown in Figure 1 executes all software.

Figure 2 shows the 3eTI Security Server fits into the overall operational environment and illustrates the logical cryptographic boundary.

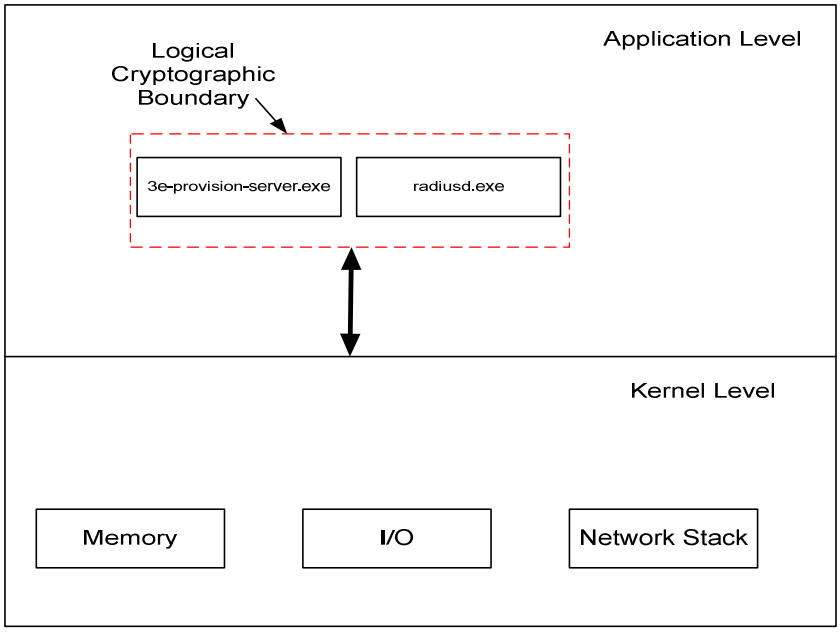

Figure 2 Cryptographic Boundary

3e-030-2 Security Server software provides authentication service for wireless users. The Security Server verifies user's digital certificate against a chain of certificate issuers and their corresponding Certificate Revocation List. The Security Server grants user access to the wireless network only upon successful authentication.

1.4. Security Level

This cryptographic module meets the overall requirements to Level 1 security of FIPS 140-2. The table below summarizes the subcategory specifications

Security Requirements Section	Level
Cryptographic Module Specification	1
Module Ports and Interfaces	1
Roles, Services and Authentication	3
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1

EMI/EMC	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

1.5. Modes of Operation

The 3eTI Security Server operates in FIPS 140-2 approved mode only. In FIPS mode of operation, the cryptographic module supports the following FIPS Approved algorithms:

- RSA with 1024, 2048, 3072 and 4096 bit keys for digital signature generation and verification. (CAVP certificate # 749)
- ECDSA with NIST defined P-256, P-384 curves for digital signature and verification. (CAVP certificate # 191)
- DSA with 1024, 2048, and 3072 bit keys for digital signature and verification. (CAVP certificate # 478)
- AES CBC mode with 128, and 256 bit keys for bulk encryption and decryption.
- AES (ECB mode; 128-bit key) for key transfer/management. (CAVP certificate # 1546)
- HMAC-SHA-1 for message authentication/integrity. (CAVP certificate #897, HMAC)
- SHS FIPS 180-3 (CAVP certificate #1371)
- Triple-DES (option 1 key bundle) for bulk encryption and decryption. (CAVP certificate #1016)
- FIPS 186-2 Random Number Generator (CAVP certificate #834)

During the module initialization step, a message is logged in the log file with the following FIPS module initialization success string: "**Security Server successfully initialized in FIPS mode**".

The product also supports the following **non-Approved but FIPS allowed** cryptographic algorithms:

- MD5 hashing in HTTPS over TLS
- Diffie-Hellman

- Elliptic Curve Diffie-Hellman

2. Ports and Interfaces

The physical ports of the module are provided by the general purpose computer on which the module is installed. The logical ports and interfaces are defined as follows:

1. Authentication Data Input & Output port, where RADIUS messages are exchanged with an 802.1X authenticator. This logical port uses the UDP port 1812 (default value) which is provided by the host PC Operating System to send and receive RADIUS messages. Each RADIUS message is authenticated using an authentication attribute created by HMAC-SHA1 using a shared authentication key between authenticator and the Security Server.

2. Management Control & Status port is provided through HTTPS with mutual authentication between the Web client and security server for cryptographic officer to configure, control and view the security server operational status.

3. Certificate-Status Data Input & Output port where a remote validation authority is consulted to get the status of a client certificate (when configured).

The security server will perform self tests during the module initialization process. No cryptographic operation will be performed by the module until the self tests are completed with success status. If the self tests fail, the module will enter an error state with error messages logged to indicate the initialization failure. No input or output of data over ports or interface will occur in this state. The self-test results are written to the audit log files. CryptoOfficer may view the audit logs after the self-test to find out the results.

3. Roles, Services, and Authentication

3.1. Assumption of Roles

The 3e-030-2 supports two roles (Cryptographic Officer and User).

The Cryptographic Officer role is assumed by the module when performing any of the following functions:

- Cryptographic Module Initialization

- Cryptographic Algorithm Testing
- Auditing
- Operator Authentication
- Key & SRDI Management
- Service Management

. The User role is assumed when the authentication service authenticates wireless clients.

3.2. Roles and Services

The 3e-030-2 Security Server supports the following roles for operators:

Cryptographic Officer Role: The Crypto Officer role performs all security functions provided by the 3e-030-2 Security Server. This role performs cryptographic initialization and management functions (e.g., server initialization and modifying server configuration). The Crypto Officer must operate within the Security Rules specified in Sections 6.1. Only one Cryptographic Officer is defined in the 3e-030-2 Security Server. .

The 3e-030-2 Security Server provides the following major services:

- **Authentication service**

  Execute TLS-based EAP method employing mutual authentication.

- **Key Management**

  Cryptographic Officer performs key input, and key zeroization.

- **Cryptographic Configuration**

  Cryptographic Officer changes password, inputs user credentials, specifies authentication algorithms and all other critical and sensitive security parameters.

- **Service Management**

  Start and stop the authentication service.

- **Access Auditing Logs & view status**

  Read audit log file and change audit log settings**.**

- **Algorithm Tests**

The follow table outlines the services that are available to each role:

Role	**Authorized Services**

Cryptographic Officer	Start and Stop Authentication service
User	Authentication Services, use of all cryptographic keys
Cryptographic Officer	Key management , set/delete keys
Cryptographic Officer	Cryptographic Configuration, change of authentication data
Cryptographic Officer	Access auditing logs, view status
Cryptographic Officer	Run On-Demand Algorithm Tests

3.3. Authentication

Identity based authentication is implemented. The Crypto Officer is authenticated by username and password over secured and trusted HTTPS path. Passwords for Crypto Officer shall be configured to be 10 or more characters (maximum of 40), including at least one character from numeric, uppercase, lowercase, and special character sets. There are possibly 4.11E+16 passwords combinations (1 numeric [10], 1 upper [26], 1 lower [26], 1 special [20], 6 any [82]), which results in a 2.43E-17 chance of a single random success. At 4 login trials per second, that'd be 1 in 1.71E+14, for a 5.84E-15 chance of a random success per minute.

Wireless User is authenticated to the Cryptographic Module through the EAP-TLS session using its X.509 certificate. A single attempt success rate for 1024 bit RSA is: 1 in 1.797693134862315907729305190789e+308 and per minute (at 4/sec) is: 1.335044315104320829854139630399 5e-306 chance of a random success per minute

4. Definition of CSPs, SRDI and Mode of Access

Role		Service	Cryptographic Keys, CSPs, Access Operation
Crypto Officer	Server Application in User Role	AES Encryption	AES Key Use Destroy
	x	AES Decryption	AES Key Use Destroy
	x	AES Key Wrap	AES KEK Use Destroy

	x	Generate random number	PRNG Seed	Use Destroy
	x	RSA Sign	RSA Private Key	Use Destroy
	x	RSA Verify	RSA Public Key	Use
	x	ECDSA Sign	ECDSA Private Key	Use Destroy
	x	ECDSA Verify	ECDSA Public Key	Use
	x	DSA Sign	DSA Private Key	Use Destroy
	x	DSA Verify	DSA Public Key	Use
	x	HMAC-SHA1	HMAC Key	Use Destroy
	x	SHA1	No access to CSPs	
	x	SHA2	No access to CSPs	
x		Zeroize	Destroy all plaintext CSPs	
x		Set all CSPs	Set	Destroy

5. Cryptographic Key Management

5.1. Key Agreement

The 3eTI Security Server uses Diffie-Hellman and ECDH for key agreement between the module and Web Access Client or the wireless supplicant client.

5.2. Key Transport

The PMK sent from the cryptographic module to 802.1X authenticator is AES key wrapped with the AES KEK and integrity and authenticity is provided by HMAC-SHA1.

The authentication and key encryption keys enter the module and are guarded by a CRC.

5.3. Key Storage

Those plaintext keys and CSPs are promptly destroyed after their usage in RAM. Keys and CSPs persisted in files in plain text format.

5.4. Key Destruction

Keys except public security parameters such as public key are zeroized and freed once no longer needed in RAM. Keys and CSPs may also be destroyed upon the Security Server's un-installation by following the steps listed below:

1. log into system as root.
2. Issue command: "**killall 3e-provisioning-server**" (this kills radius too)
3. Issue command: "**shred /opt/3eti/security-server/db/server.can –x -z**"
4. Issue command: "**shred /opt/3eti/security-server/db/auth.db –x -z**"
5. Issue command: "**shred /opt/3eti/security-server/db/certs/Server.cert –x -z**"
6. Issue command: "**shred /opt/3eti/security-server/db/certs/*.0 –x -z**"
7. Issue command: "**shred /opt/3eti/security-server/db/certs/Dh –x -z**"
8. Issue command: "**shred /opt/3eti/security-server/db/web/Server.cert –x -z**"
9. Issue command: "**shred /opt/3eti/security-server/db/web/*.0 –x -z**"
10. Issue command: "**shred /opt/3eti/security-server/rel-1.0.0/sbin/ file provisioning-server-key.pem –x -z"**

The following table shows the Cryptographic Keys and CSPs:

ID	TYPE	Storage Location	Form	Zeroizable	Protection Mechanism	Function
Module Integrity Key	ECDSA P-384 Public Key	RAM / HARD DISK	Plaintext (inaccessible, hard-coded)	N/A	N/A	Authenticate and verify integrity of Module code.
Web TLS Master Secret	AES CBC 128, 256 Triple-DES CBC	RAM	Plaintext	Y	Zeroized	Derive Web TLS session keys
802.1x TLS Master Secret	AES CBC 128, 256 Triple-DES CBC	RAM	Plaintext	Y	Zeroized	Derive EAP-TLS session keys
PMK	EAP MSK	RAM	Plaintext	Y	Zeroized Transfer encrypted using	Transferred to Authenticator in order to derive PTK

					approved Keywrap and guarded by approved keyed hash	
802.1x DH Prime	DHE Private Exponent 2048 bit	RAM HARD DISK	Plaintext	Y	Inaccessible	
802.1x ECDH Prime	ECDHE Private Exponent 256, 384 bit	RAM HARD DISK	Plaintext	Y	Inaccessible	
Web DH Prime	DHE Private Exponent 2048 bit	RAM HARD DISK	Plaintext	Y	Inaccessible	
Web ECDH Prime	ECDHE Private Exponent 256, 384 bit	RAM HARD DISK	Plaintext	Y	Inaccessible	
Web ECDH Private Key	ECDH Private Key P-256, P-384	RAM	Plaintext	Y	Inaccessible	Derive TLS Master Secret.
802.1x ECDH Private Key	ECDH Private Key P-256, P-384	RAM	Plaintext	Y	Inaccessible	Derive TLS Master Secret.
HMAC SHA-1 secret	“Authentication AP/Security Server Shared Secret”	RAM / HARD DISK	Plaintext	Y		Authentication between 802.1X authenticator (AP) and Security Server.
Key Encryption Key	AES 128 bit Key Wrapper key.	RAM / HARD DISK	Plaintext	Y		To encrypt EAP MSK before transit to authenticator
Web Access Server Certificate	RSA, DSA, ECDSA Certificate & public key	RAM / HARD DISK	Plaintext	Y	N/A	Offered during TLS handshake for authentication.
802.1x Server Certificate	RSA, DSA, ECDSA Certificate & public key	RAM / HARD DISK	Plaintext	Y	N/A	Offered during TLS handshake for authentication.

Web Access Server Private Key	RSA, DSA, ECDSA Private Key	RAM / HARD DISK	Plaintext	Y		Private key for authentication
Web Access Private Key Password	Used to decrypt private asymmetric key	RAM / HARD DISK	Plaintext	Y		Password used to decrypt a server private key.
802.1x Server Private Key	RSA, DSA, ECDSA Private Key	RAM / HARD DISK	Plaintext	Y		Private key for authentication
802.1x Server Private Key Password	Used to decrypt private asymmetric key	RAM / HARD DISK	Plaintext	Y		Password used to decrypt a server private key.
Crypto Officer password	Used to authenticate Crypto Officer	RAM/HARD DISK	Hashed value	Y		Password used to authenticate the cryptographic officer
FIPS 186-2 seed key		RAM	Plaintext in RAM	Y		Used to initialize FIPS PRNG Zeroized every time a new random number is generated using the FIPS PRNG
Non-Approved RNG seed	20-byte value		Plaintext in RAM	Y		Used as Seed for non-Approved RNG which provides seed key for FIPS PRNG. Zeroized every time a new random number is generated

6. Secure Operation and Security Rules

6.1. FIPS mode of operation and Secure Installation

The 3e-030-2 Security Server is software that should be installed on the X86 Linux Server. The following steps must be performed to install and initialize the module for operating in a FIPS 140-2 compliant manner:

1. The operating system must be configured to run in single-user mode.

2. The paging file (Virtual memory) must be configured to reside on a local drive on the workstation, not a network drive.

3. After the Cryptographic Officer log in to the Security Server for the first time using the default authentication (See User's Guide). The crypto officer must change the password.

4. The Crypto Officer shall configure the Security Server to enforce mutual authentication between the web client and security server over HTTPs.

7. Physical Security

The module itself is comprised of software only and thus does not provide any physical security mechanisms.

8. Operational Environment

The 3e-030-2 Security Server is software that has been tested on an X86 Linux Server running Red Hat Linux Enterprise OS version 5.5.

The Red Hat Linux Enterprise OS, as a general purpose/modifiable operation environment together with Security Operation Rules stated in Section 6, meets all FIPS 140-2 section 4.6.1 requirement for security Level 1.

9. Self-tests

Self tests occur during module initialization (Power on), or at Crypto Officer's command (Manual).

9.1. Power on Self Tests

The power-on self test consists of software integrity test and known answer tests for the cryptographic algorithm implementations.

9.2. Software Integrity Test

The Software Integrity check is performed when the Crypto Module is initialized. The module implements an integrity test for the module software by verifying its ECDSA signature(s). The software integrity test passes if and only if the signature verifies successfully using the embedded ECDSA-P384 public key.

9.3. On Demand Self Tests

The cryptographic officer may initiate algorithm tests at any time. All other cryptographic activity will cease until the tests have run successfully. Upon failure of any test, the security server will audit the failure and abort execution immediately.

9.4. Cryptographic Algorithm Self-tests

- AES ECB - encrypt/decrypt KAT

- AES CBC -encrypt/decrypt KAT
- Triple-DES CBC –encrypt/decrypt KAT
- SHA-1 KAT
- SHA-2 KAT
- HMAC-SHA-1 KAT
- PRNG FIPS 186-2 (Appendix 3.1, 3.3) KAT
- RSA Sign/Verify KAT
- DSA Sign/Verify Pair-Wise Consistency Test
- ECDSA Sign/Verify Pair-Wise Consistency Test
- Software Integrity Test (power-up)

9.4.1. Conditional Self-tests

- CRNGT for Approved PRNG

9.4.2. Critical Functions tests

- DH pair wise consistency test (power-up)
- ECDH pair wise consistency test (power-up)

10. Design Assurance

All source code and documentation are stored in version control system CVS. The module is coded in C with module's components directly corresponding to the security policy's rules of operation. Functional Specification is also provided.

11. Mitigation of Other Attacks

The module has not been designed to mitigate any specific attack beyond the scope of FIPS 140-2 requirements.