Vocera Communications, Inc.

Vocera Cryptographic Module

Version: 3.1

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.6

Prepared for:

Vocera Communications, Inc.
525 Race Street
San Jose, CA 95126
United States of America

Phone: +1 408 882 5100
www.vocera.com

Prepared by:

Corsec Security, Inc.
13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

Phone: +1 703 267 6050
www.corsec.com

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Vocera Cryptographic Module from Vocera Communications, Inc. (Vocera). This Security Policy describes how the Vocera Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The Vocera Cryptographic Module is referred to in this document as the VCM or the module.

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Vocera website (www.vocera.com) contains information on the full line of products from Vocera.
- The search page on the CMVP website (https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Validated-Modules/Search) can be used to locate and obtain vendor contact information for technical or sales-related questions about the module.

1.3 Document Organization

The Security Policy document is organized into two (2) primary sections. Section 2 provides an overview of the validated module. This includes a general description of the capabilities and the use of cryptography, as well as a presentation of the validation level achieved in each applicable functional area of the FIPS standard. It also provides high-level descriptions of how the module meets FIPS requirements in each functional area. Section 3 documents the guidance needed for the secure use of the module, including initial setup instructions and management methods and policies.

2. Vocera Cryptographic Module

2.1 Overview

The Vocera Communications System is a breakthrough wireless platform that provides hands-free voice communications throughout an 802.11a/b/g/n-networked building or campus. The Vocera Communications System consists of two key components:

- The Vocera Server System Software, which runs on a standard Windows server, controls and manages call activity.
- The Vocera B3000n Communications Badge allows users to converse over a Wireless Local Area Network (WLAN).

A typical Vocera system deployment is shown in Figure 1 below. The following acronyms that are previously undefined appear in Figure 1:

- LAN – Local Area Network
- PBX – Private Branch Exchange

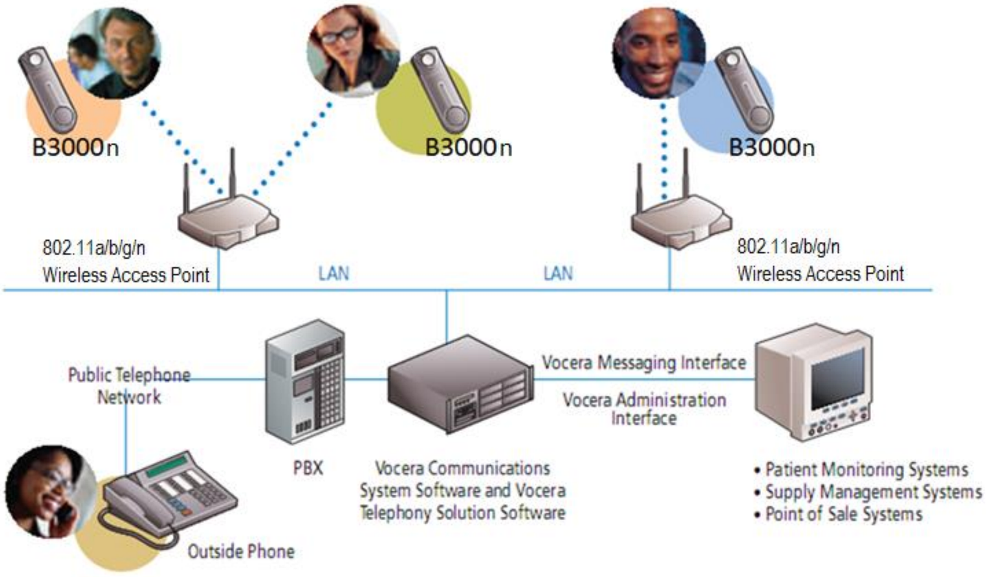

Figure 1 – Typical Vocera Communications System Deployment

The Vocera B3000n Communications Badge (see Figure 2) is a small, virtually hands-free wireless device that acts as the interface to the Vocera Communications System. The wearable badge is controlled using voice commands, and enables instant two-way voice conversation, text messaging, and alerts. The badge communicates with other Vocera communications devices or with the Vocera Server System Software (typically referred to as the Vocera Server) securely over a protected channel. With optional Vocera telephony solution software, the badge can also make and receive telephone calls through the Vocera Server via a PBX. The badge employs a high-performance antenna for improved transmit and receive sensitivity.

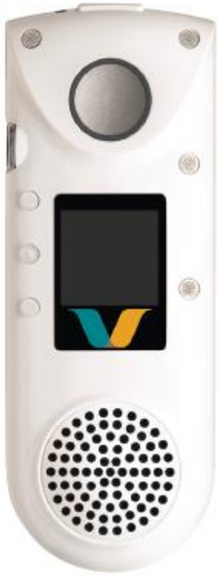

Figure 2 – Vocera B3000n Communications Badge

The security functionality required to protect badge communications is provided by the VCM embedded in the badge. Software libraries provide support for TLS[1] connections, while a WLAN chip provides support for the industry-standard wireless authentication protocols (including WPA-PSK[2], EAP[3]-TLS, and PEAP[4]) and encryption protocols (including AES[5]-CCMP[6]) used to protect the wireless traffic. Various applications on the Vocera B3000n Communications Badge make use of the VCM to establish a secure connection with the Vocera Server and with other Vocera communications devices. All cryptographic services needed by the badge are provided by the VCM.

The VCM is validated at the FIPS 140-2 Section levels shown in Table 1.

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	1

[1] TLS – Transport Layer Security
[2] WPA – Wi-Fi Protected Access - Pre-Shared Key
[3] EAP – Extensible Authentication Protocol
[4] PEAP – Protected Extensible Authentication Protocol
[5] AES – Advanced Encryption Standard
[28] CCMP – Counter with CBC-MAC Protocol

Section	Section Title	Level
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI[7]/EMC[8]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A[9]

2.2 Module Specification

The VCM is a software-hybrid cryptographic module with an overall security level of 1. As a software-hybrid, the module consists of disjoint software and hardware components, all contained within the same physical enclosure. The various module components are collectively versioned as 3.1.

The module supports an Approved and a non-Approved mode of operation. Please refer to section 2.2.4 of this Security Policy for details regarding the supported modes of operation.

The module executes on a Vocera B3000n Communications Badge with a Texas Instruments OMAP-L138 processor running Vocera Embedded Linux Version 3.1.

2.2.1 Physical Cryptographic Boundary

As a software-hybrid cryptographic module, the module takes on the characteristics of the host device, the Vocera B3000n Communications Badge. Thus, the physical cryptographic boundary is the hard plastic badge enclosure, and the module is defined as having a multiple-chip standalone embodiment.

The badge includes a Texas Instruments (TI) OMAP-L138 System-on-a-Chip (SoC) dual-core applications processor, 64MB[10] of SDRAM[11], and 128MB of NAND[12] Flash storage. The OMAP-L138 processor's dual-core architecture contains both ARM9 and DSP[13] processor cores, together with 256KB[14] of shared, on-chip SRAM[15].

2.2.2 Logical Cryptographic Boundary

The module's logical cryptographic boundary encompasses the module's disjoint software and hardware components. The software component comprises four software libraries, while the hardware component is a high-performance WLAN chip:

[7] EMI – Electromagnetic Interference
[8] EMC –Electromagnetic Compatibility
[9] N/A – Not Applicable
[10] MB – Megabyte
[11] SDRAM – Synchronous Dynamic Random Access Memory
[12] NAND – Negated AND
[13] DSP – Digital Signal Processor
[14] KB – Kilobyte
[15] SRAM – Static Random Access Memory

- The four software libraries are fips_libbadgecrypto.so, fips_rsa_sig_verify.so, mlan.ko, and sd8787.ko (collectively versioned as version 3.1). The software libraries are stored in Flash and execute on a Texas Instruments applications processor (part number OMAPL138). The fips_libbadgecrypto.so and fips_rsa_sig_verify.so libraries provide AES, HMAC[16] SHA[17], CMAC[18], DRBG[19], and RSA[20] signature verification services, whereas mlan.ko and sd8787.ko provide the API[21] to the Marvell chip.

- The WLAN chip is the Marvell Avastar WLAN SoC[22] (part number 88W8787). It holds and executes its associated firmware (sd8787_uapsta.bin), provides AES-CCM[23] encryption/decryption functionality, and supports wireless radio communications protocols, including WPA-PSK, PEAP, EAP-TLS, EAPOL[24], OKC[25], and CCKM[26]. The chip firmware is version 3.0.

The module is entirely contained within the physical cryptographic boundary described in section 2.2.1. Figure 3 below shows the logical block diagram of the module executing in memory and its interactions with surrounding software/firmware components, as well as the module's physical and logical cryptographic boundaries.

[16] HMAC – (keyed-) Hashed Message Authentication Code
[17] SHA – Secure Hash Algorithm
[18] CMAC – Cipher-based Message Authentication Code
[19] DBRG – Deterministic Random Bit Generator
[20] RSA – Rivest Shamir Adleman
[21] API – Application Programming Interface
[22] SoC – System on a Chip
[23] CCM – Counter with CBC-MAC
[24] EAPOL – Extensible Authentication Protocol over Local Area Networks
[25] OKC – Opportunistic Key Caching
[26] CCKM – Cisco Centralized Key Management

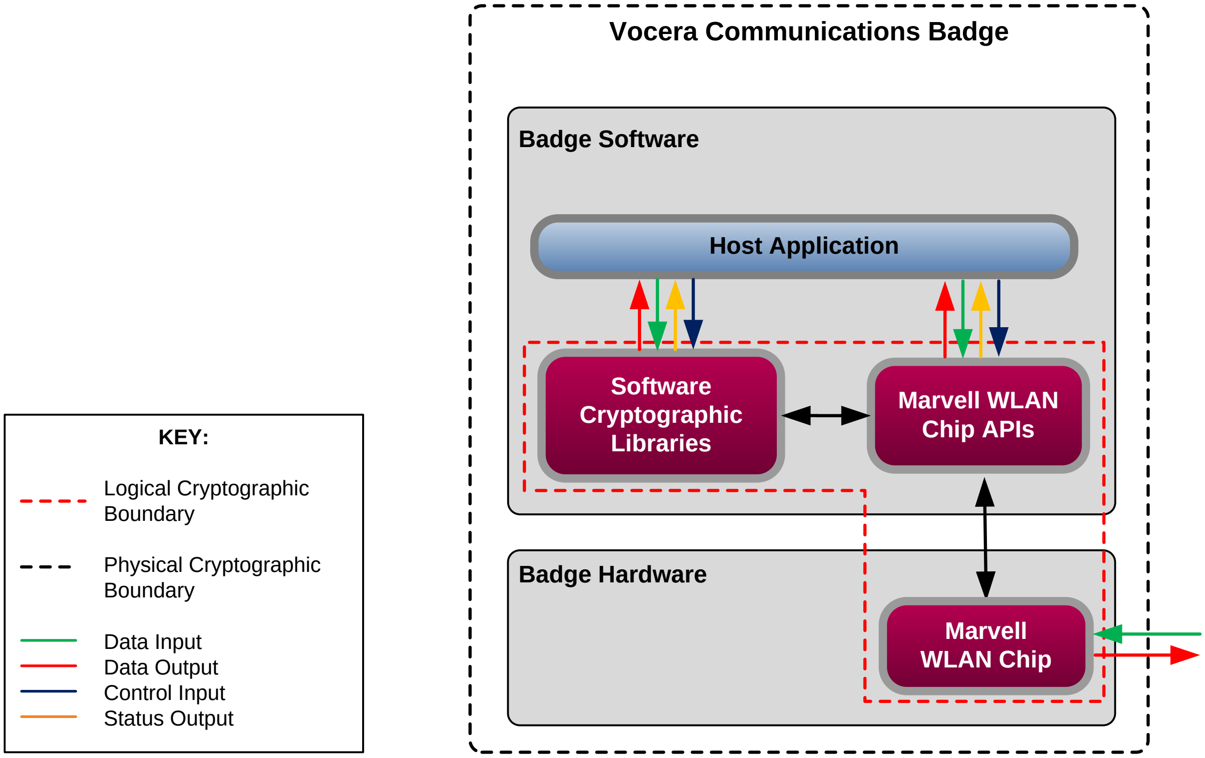

Figure 3 – Module Cryptographic Boundaries

2.2.3 Algorithm Implementations

The module implements the FIPS-Approved algorithms listed in Table 2 below.

Table 2 – FIPS-Approved Algorithm Implementations

Certificate Number		Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
Libraries	Chip					
[C2167](C2167)	-	AES	FIPS PUB 197	CBC[27]	128	encryption/decryption
			NIST SP 800-38B	CMAC	128	MAC generation/verification
-	3531	AES	FIPS PUB 197	ECB[28]	128	encryption/decryption
			NIST SP 800-38C	CCM	128	encryption/decryption
C2167	-	CVL[29]	NIST SP 800-135rev1	TLS 1.0/1.1 KDF[30]	-	key derivation *No parts of the TLS protocol, other than the KDF, have been tested by the CAVP or CMVP.*

[27] CBC – Cipher Block Chaining
[28] ECB – Electronic Code Book
[29] CVL – Component Validation List
[30] KDF – Key Derivation Function

Certificate Number		Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
Libraries	Chip					
C2167	-	DRBG	NIST SP 800-90Arev1	HASH-based	SHA2-256	deterministic random bit generation
C2167	-	HMAC	FIPS PUB 198-1	SHA-1, SHA2-256	KS<BS, KS=BS, KS>BS	message authentication
C2167	-	RSA	FIPS PUB 186-2	PKCS1.5[31]	1024, 1536, 2048, 3072, 4096	digital signature verification
			FIPS PUB 186-4	PKCS1.5	1024, 2048, 3072	digital signature verification
C2167	-	SHS[32]	FIPS PUB 180-4	SHA-1, SHA2-256	-	message digest

The module implements the non-Approved but allowed algorithms shown in Table 3 below.

Table 3 – Allowed Algorithms

Algorithm	Caveat	Use
MD5[33]	-	component of TLS handshake underlying algorithm within the module's key transport schemes (TLS, EAP-TLS, and PEAP) and, as such, are allowed for use per FIPS Implementation Guidance D.9
NDRNG[34]	-	entropy input source for the module's Approved DRBG *Each call to the entropy source requests 256 bits for entropy input or 128 bits for the DRBG's nonce.*
RSA (2048-bit)	key establishment methodology provides 112 bits of encryption strength; non-compliant less than 112 bits of encryption strength	key tranport

2.2.4 Modes of Operation

The module supports two modes of operation: Approved and Non-approved. The module will be in FIPS-Approved mode when all power up self-tests have completed successfully, and only Approved algorithms are invoked. See Table 2 and Table 3 above for a list of the Approved and allowed algorithms.

In the non-Approved mode of operation, the module employs the following non-Approved algorithm(s):

- 1024-bit RSA key transport
- SHA2-384 (non-compliant)
- SHA2-512 (non-compliant)

[31] PKCS – Public Key Cryptography Standard
[32] SHS – Secure Hash Standard
[33] MD5 – Message Digest 5
[34] NDRNG – Non-Deterministic Random Number Generator

No CSPs are shared between modes. Each invocation of an RSA function requires a key as input from the calling application. Any 1024-bit key used used by the calling application for the non-Approved RSA key transport functions shall not be for an Approved RSA function.

The module can alternate service-by-service between Approved and non-Approved modes of operation. The module will switch to the non-Approved mode upon execution of a non-Approved service. The module will switch back to the Approved mode upon execution of an Approved service. Table 7 below lists the services comprising the non-Approved mode of operation.

The module supports the Crypto Officer and User roles in both supported modes of operation.

2.3 Module Interfaces

The module interfaces exist at the module's logical cryptographic boundary. Thus, while included here for completeness, the Vocera B3000n Communications Badge is not within the logical boundary of the cryptographic module, and the module's interfaces are not implemented at this boundary. Only the components within the logical boundary illustrated in Figure 3 above comprise the module, and it is at this boundary where the module's interfaces are implemented.

The module's physical interfaces are those of the host badge. Those interfaces are as follows:

- Badge display
- Buttons (call button, hold/DND[35] button, and menu buttons)
- Speaker
- Microphone
- Badge indicator light
- Call button halo
- Headset jack
- Battery contact pins (under battery compartment cover)
- WLAN unit

See Figure 4 below for an illustration of the badge's various physical features.

[35] DND – Do Not Disturb

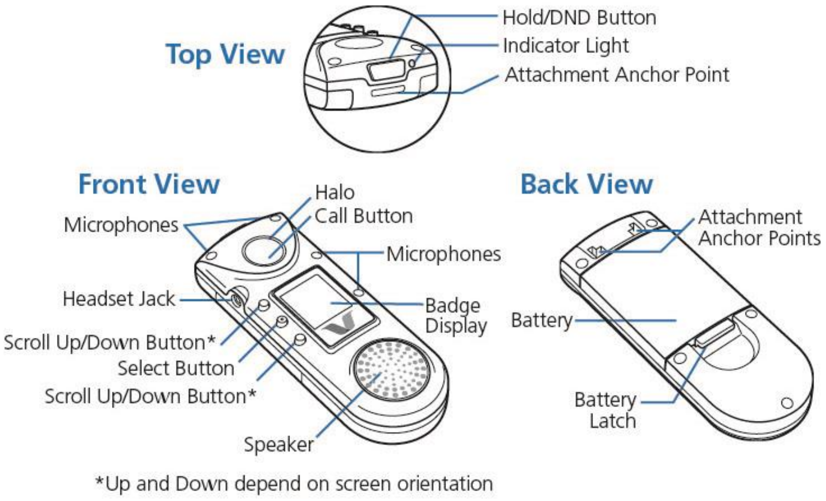

Figure 4 – Physical Features of the Vocera B3000n Communications Badge

The badge's physical interfaces (manual controls, physical indicators, and physical ports) map to logical interfaces supported by the module. The module's logical interfaces are at a low level in the software. The module isolates communications to logical interfaces that are defined in the software as APIs. The APIs are grouped into four logically distinct categories:

- Data Input
- Data Output
- Control Input
- Status Output

Per FIPS 140-2 Implementation Guidance, a hybrid module's control input and status output interfaces are provided only via the software component of the module. Thus, the module's interfaces consist solely of the APIs offered by the software libraries. The data and control inputs made via the badge microphone and buttons are translated into the logical data and control inputs made via the API calls to the software-hybrid module. Likewise, the data and status outputs made via API call returns from the software-hybrid module are translated into the data and status outputs made to the badge display, speaker, and status lights.

Table 4 below provides a mapping of the physical (i.e. badge) and logical (i.e. module) interfaces to the appropriate interface category.

Table 4 – Interface Mappings

Interface Category	Physical Interface	Logical Interfaces
Data Input	WLAN Unit, Microphone, Headset Jack	Function calls that accept, as their arguments, data to be used or processed by the module.
Data Output	WLAN Unit, Headset Jack, Speaker	Arguments for a function that specify where the result of the function is stored or returned as a return value.

Interface Category	Physical Interface	Logical Interfaces
Control Input	WLAN Unit, Call Button, Hold/DND Button (Hold to power-off), Menu Buttons	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output	Badge Display, Badge Indicator Light, Call Button Halo	Return values for function calls
Power Input	Battery Contact Pins	N/A

2.4 Roles and Services

The sections below describe the module's authorized roles and services.

2.4.1 Authorized Roles

The module is a library that provides cryptographic functions to calling applications, and the applications that link to the module are considered the module "operators". There are two roles supported the module that operators may assume: a Crypto-Officer (CO) role and a User role. Module operators assume their desired role implicitly, based on the module service selected for execution. The module does not allow concurrent operators.

2.4.2 Operator Services

The Approved services available to module operators are described in Table 5 and Table 6 below. Please note that the keys and Critical Security Parameters (CSPs) listed in the table indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

Table 5 – Crypto Officer Services

Service	Description	Input	Output	CSP and Type of Access
Self-test execution	Run power-up self-tests on demand	API call parameters or cycling power	Status output	None
Status monitoring	Monitor status	Command	Status output	None

Table 6 – User Services

Service	Description	Input	Output	CSP And Type Of Access
Zeroization	Zeroize keys utilized by the module	API call parameters or cycling power	Status output	RSA Server Public Key – W RSA Client Private Key – W TLS Authentication Key – W TLS Session Key – W 802.11i/r/w PMK – W 802.11i/r/w Temporal Key – W 802.11w Integrity Group Temporal Key – W AES CMAC Key – W HMAC Key – W DRBG Entropy Input String – W DRBG Seed – W DRBG "C" Value – W DRBG "V" Value – W
Crypto operation initiation	Create an environment to carry out cryptographic operation	API call parameters	Status output	None
Random number generation	Generate random number	API call parameters	Status output, random value	DRBG Entropy Input String – RX DRBG Seed – WX DRBG "C" Value – WX DRBG "V" Value – WX
Symmetric encryption	Encrypt plaintext data	API call parameters, key, context	Status output, encrypted context	TLS Session Key – RX 802.11i/r/w Temporal Key – RX
Symmetric decryption	Decrypt ciphertext data	API call parameters, key, context	Status output, decrypted context	TLS Session Key – RX 802.11i/r/w Temporal Key – RX
Hashing operation	Generate SHA-1 and SHA2-256 digests	API call parameters, context	Status output, hash	None
Signature verification	Verify a digital signature	API call parameters, signature	Status output	RSA Server Public Key – RX
Key transport operation	Perform key transport function	API call parameters, key	Status output	RSA Server Public Key – RX
MAC operation	Generate MAC value	API call parameters, key, context	Status output, MAC	HMAC Key – RX AES CMAC Key – RX
TLS key derivation	Perform TLS key derivation	API call parameters	Status output	RSA Server Public Key – RX RSA Client Private Key – RX TLS Pre-Master Secret – X TLS Master Secret – WX TLS Authentication Key – WX TLS Session Key –WX
EAPOL-Key message operations	Format EAPOL-Key message	API call parameters	Status output	802.11i/r/w Pair-Wise Master Key (PMK) – RX
EAPOL operation	Transmit and receive EAP messages using EAPOL	API call parameters	Status output	802.11i/r/w PMK – RWX
OKC operation	Performs OKC operation	API call parameters	Status output	802.11i/r/w PMK – RX

Service	Description	Input	Output	CSP And Type Of Access
CCKM operation	Performs CCKM operation	API call parameters	Status output	802.11i/r/w PMK – RX
Four-way handshake processing	Process four-way handshake	API call parameters	Status output	802.11i/r/w PMK – RX 802.11i/r/w Temporal Key – WX 802.11w Integrity Group Temporal Key – WX
WPA-PSK operation	Perform WPA-PSK operation	API call parameters, message traffic	Status output	WPA-PSK Pre-Shared Key – RX
PEAP operation	Perform PEAP operation	API call parameters, message traffic	Status output	RSA Server Public Key – RX RSA Client Private Key – RX TLS Authentication Key – X TLS Session Key – X 802.11i/r/w PMK – RWX DRBG Entropy Input String – RX DRBG Seed – WX DRBG "C" Value – WX DRBG "V" Value – WX
EAP-TLS operation	Perform EAP-TLS operation	API call parameters, message traffic	Status output	RSA Server Public Key – RX RSA Client Private Key – RX TLS Authentication Key – X TLS Session Key – X 802.11i/r/w PMK – RWX DRBG Entropy Input String – RX DRBG Seed – WX DRBG "C" Value – WX DRBG "V" Value – WX
TLS operation	Perform TLS operation	API call parameters, message traffic	Status output	TLS Authentication Key – X TLS Session Key – X

Table 7 below lists the services available in the non-Approved mode of operation.

Table 7 – Non-Approved Services

Service	Operator		Input	Output	Description
	CO	User			
PEAP operation (non-compliant)		✓	API call parameters, message traffic	Status output	Perform PEAP operation using 1024-bit RSA key transport
EAP-TLS operation (non-compliant)		✓	API call parameters, message traffic	Status output	Perform EAP-TLS operation using 1024-bit RSA key transport
TLS operation (non-compliant)		✓	API call parameters, message traffic	Status output	Perform TLS operation using 1024-bit RSA key transport
Hashing operation (non-compliant)		✓	API call parameters, context	Status output, hash	Generate SHA2-384 or SHA2-512 digest

The module will switch to the non-Approved mode of operation when any service in Table 7 is invoked. The module will switch back to the Approved mode of operation upon invocation of any service in Table 6.

2.4.3 Authentication

As a level 1 cryptographic module, authentication mechanisms are not supported.

2.5 Physical Security

The VCM is a software-hybrid module, which FIPS defines as a multiple-chip standalone embodiment. The hardware components of the module consist of production-grade components that include standard passivation techniques. Further, the module is entirely contained within the hard plastic badge enclosure, which blocks physical access to the module.

2.6 Operational Environment

The module executes on a Vocera B3000n Communications Badge running Vocera Embedded Linux Version 3.1. The software libraries run on the OMAP-L138 processor, while the Marvell WLAN chip runs its own special-purpose firmware.

The module is considered to be operating a single-user environment due to the fact that only one operator can be in possession of a given communications badge (which hosts the module) at any given time. The module is not intended to operate on any platform other than the Vocera B3000n Communications Badge. As such, all keys, intermediate values, and other CSPs remain only in the process space of the operator using the module. The operating system (OS) uses its native memory management mechanisms to ensure that outside processes cannot access the process space used by the module.

2.7 Cryptographic Key Management

The module supports the CSPs listed below in Table 8.

Table 8 – Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
RSA Server Public Key	1024, 1536, 2048, 3072, 4096-bit RSA public key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Signature verification; Key transport during TLS handshake for PEAP and EAP-TLS phase 1
RSA Client Private Key	2048, 4096-bit RSA private key	Externally generated and input in plaintext	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Used with client-side certificates for authentication in EAP-TLS
TLS Pre-Master Secret	384-bit random value	[for RSA cipher suites and module acting as client] Internally generated via FIPS-Approved DRBG [for RSA cipher suites and module acting as server] Externally generated and imported in encrypted form via RSA key transport	[for RSA cipher suites and module acting as client] Exits the module in encrypted form via RSA key transport [for RSA cipher suites and module acting as server] Never exits the module	Not persistently stored	Unload module; remove/cycle power	Derivation of the TLS Master Secret
TLS Master Secret	384-bit shared secret	Internally derived via TLS KDF	Output in plaintext to the calling application	Not persistently stored	Unload module; remove/cycle power	Derivation of the TLS Session Key and TLS Authentication Key
TLS Session Key	128-bit AES key	Internally derived via TLS KDF OR Externally generated and input in ciphertext	Output in ciphertext during TLS handshake OR Never exits the module	Not persistently stored	Unload module; remove/cycle power	TLS session encryption/decryption of authentication-related messages in PEAP Phase 2 and EAP-TLS

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
TLS Authentication Key	160-bit HMAC SHA-1 key	Internally derived via TLS KDF OR Externally generated and input in ciphertext	Output in ciphertext during TLS handshake OR Never exits the module	Not persistently stored	Unload module; remove/cycle power	Data authentication for TLS sessions for PEAP Phase 2 and EAP-TLS
WPA-PSK Pre-Shared Key	128-bit AES key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; remove/cycle power	WPA-PSK authentication and encryption/decryption
802.11i/r/w Pair-wise Master Key	256-bit shared secret	[for WPA-PSK] Externally generated and input in plaintext [for PEAP and EAP-TLS] Internally generated	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Partial input to construct 802.11i/r/w Temporal Key used in 802.11i protocol
802.11i/r/w Temporal Key	128-bit AES key	Internally generated	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Used to create secure tunnel for wireless data transmission.
802.11w Integrity Group Temporal Key	128-bit AES CMAC key	Internally generated	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Used for data integrity check for group addressed Management frame
HMAC Key	128, 160, 256-bit HMAC SHA-1 key 256-bit HMAC SHA-256 key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Used for Keyed-Hash Message Authentication in the module
AES CMAC Key	128-bit AES CMAC key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Used for Keyed-Hash Message Authentication in the module
DRBG Entropy Input String[36]	256-bit value	Internally generated by the NDRNG	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Entropy material for DRBG seed generation

[36] The min-entropy is 3.463 bits of entropy per 4 bits of raw noise data. As long as there is at least one bit of entropy per four bits of raw noise data, the entropy provided by each call to the entropy source can be considered to contain full entropy. Thus, when the DRBG requests 256 bits of entropy for seeding, the entropy source returns 256 bits of entropy.

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
DRBG Seed	440 bits of seed value	Internally generated using nonce along with entropy input	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Seed value generated by the DRBG
DRBG “C” Value	Internal DRBG state value	Internally generated	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Generation of random number
DRBG “V” Value	Internal DRBG state value	Internally generated	Never exits the module	Not persistently stored	Unload module; remove/cycle power	Generation of random number

2.8 EMI / EMC

The module is a software-hybrid module and depends on the target platform for its physical characteristics. The VCM is not a radio device. However, the module's target platform is a Vocera B3000n Communications Badge, which is considered a radio device. The Vocera B3000n Communications Badge has been tested and found compliant with Subparts B, C, and E of Part 15 of the Federation Communications Commission (FCC) rules (CFR[37] title 47) for Class A digital devices.

2.9 Self-Tests

Cryptographic self-tests are performed by the module when the module is first powered up and loaded into memory as well as when a random number or asymmetric key pair is created. The following sections list the self-tests performed by the module, their expected error status, and error resolutions.

2.9.1 Power-Up Self-Tests

The module performs a series of FIPS-required self-tests at power-up. These tests are performed automatically, without the need for operator intervention. The module is capable of performing the power-up self-tests on-demand via power cycle, which restarts the module.

The VCM performs the following self-tests at power-up:

- Library Software Integrity Check using HMAC SHA-1
- Chip Firmware Integrity Check using HMAC SHA-1
- Known Answer Tests (KATs)
  - AES ECB KATs for encrypt and decrypt
  - AES CCM KATs for encrypt and decrypt
  - AES CBC KATs for encrypt and decrypt
  - AES CMAC KATs for generate and verify
  - SHA-256 KAT
  - HMAC SHA-256 KAT
  - DRBG KAT
  - RSA signature verification KAT

The module's SHA-1 and HMAC SHA-1 implementations are fully tested by the module's HMAC SHA-1 power-up integrity tests. Thus, as allowed per section 9.2 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, no independent KATs are needed for the module's SHA-1 and HMAC implementations.

2.9.2 Conditional Self-Tests

The module performs a series of FIPS-required self-tests operationally when the module generates a random value. These tests are performed automatically, without the need for operator intervention. The VCM performs the following conditional self-tests:

[37] CFR – Code of Federal Regulations

- CRNGT for the non-Approved NDRNG
- CRNGT for the FIPS-Approved DRBG

2.9.3 Critical Functions Self-Tests

The module performs health checks for the DRBG's Generate, Instantiate, and Reseed functions as specified in section 11.3 of *NIST SP 800-90Arev1*. The module also performs a KAT for the TLS KDF implementation. These tests are performed at module power-up.

2.9.4 Self-Test Failure Handling

Failure of any power-up self-test or the conditional CRNGT for the non-Approved NDRNG will result in the module entering a critical error state immediately. For the conditional CRNGT for the Approved DRBG, a newly-produced set of random bits is compared to the previously-produced set of random bits. If they are equal, then the test is failed and the module will enter a soft error state. The module will then generate a new set of random bits and perform the comparison again. If the test is failed a second time, the module will enter a critical error state.

Upon reaching the critical error state, the module outputs a failure indicator over the module's status output interface (this indicator maps to a failure message that is displayed on the badge). The module will then immediately terminate the host application and either (1) restart the host application or (2) reboot the badge. See Table 9 below for a list of self-test failure actions and messages.

Table 9 – Self-Test Failure Actions and Messages

Self-Test	Failure Action		Failure Message
	Restart Host Application	Reboot Badge	
Chip Firmware Integrity Check		✓	"Integrity check of [module file name] Power On Self Test failed. Rebooting system in <n>* seconds."
Library Software Integrity Check		✓	"Integrity check of [module file name] Power On Self Test failed. Rebooting system in <n> seconds."
AES ECB KAT		✓	"AES ECB Decryption Power On Self Test failed. Rebooting system in <n> seconds."
		✓	"AES ECB Encryption Power On Self Test failed. Rebooting system in <n> seconds."
AES CCM KAT		✓	"AES CCM Encryption Power On Self Test failed. Rebooting system in <n> seconds."
		✓	"AES CCM Decryption Power On Self Test failed. Rebooting system in <n> seconds."
AES CBC KAT		✓	"AES CBC Encryption Power On Self Test failed. Rebooting system in <n> seconds."
		✓	"AES CBC Decryption Power On Self Test failed. Rebooting system in <n> seconds."
AES CMAC KAT		✓	"AES CMAC Power On Self Test failed. Rebooting system in <n> seconds."

Self-Test	Failure Action		Failure Message
	Restart Host Application	Reboot Badge	
SHA-1 KAT		✓	[refer to Library Software Integrity Check]
SHA-256 KAT		✓	"SHA256 Power On Self Test failed. Rebooting system in <n> seconds."
HMAC SHA-1 KAT		✓	[refer to Library Software Integrity Check]
HMAC SHA-256 KAT		✓	"HMAC SHA256 Power On Self Test failed. Rebooting system in <n> seconds."
Hash-based DRBG KAT		✓	"DRBG Power On Self Test failed. Rebooting system in <n> seconds."
	✓		"RBG[38] KAT for Instantiate failed. Restarting. Restarting application in <n> seconds."
	✓		"RBG Invalid parameter specified for Instantiate. Restarting. Restarting application in <n> seconds."
	✓		"RBG KAT for Reseed failed. Restarting. Restarting application in <n> seconds."
	✓		"RBG Invalid parameter specified for Reseed. Restarting. Restarting application in <n> seconds."
	✓		"RBG KAT for Generate failed. Restarting. Restarting application in <n> seconds."
	✓		"RBG Invalid parameter specified for Generate. Restarting. Restarting application in <n> seconds."
	✓		"RBG KAT for Uninstantiate failed. Restarting. Restarting application in <n> seconds."
RSA Signature Verification KAT		✓	"RSA Sigver Power On Self Test failed. Rebooting system in <n> seconds."
CRNGT for DRBG	✓		"RBG Continuous Test failed. Restarting. Restarting application in <n> seconds."
CRNGT for NDRBG		✓	"RBG Entropy input continuous test failed. Rebooting. Rebooting system in <n> seconds."
	✓		"Unable to process RBG entropy source. Restarting. Restarting application in <n> seconds."
		✓	"Unable to open RBG entropy source. Rebooting. Rebooting System in <n> seconds."
		✓	"Unable to get requested RBG entropy. Rebooting. Rebooting System in <n> seconds."
	✓		"Unknown entropy source specified. Rebooting. Restarting application in <n> seconds."
TLS KDF KAT		✓	"TLS KDF Power On Self Test failed. Rebooting system in <n> seconds."

*<n> is a countdown from 10 to 1.

[38] RBG – Random Bit Generation

If the error state persists through the automatic restart/reboot, an operator may attempt to manually clear the self-test error by restarting the module (which requires power-cycling the host badge); however, if the error does not clear, then the badge must be sent to Vocera for service.

2.10 Mitigation of Other Attacks

This section is not applicable. The modules do not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

3. Secure Operation

The VCM meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation. **Operating the module without following the guidance herein (including the use of undocumented services) will result in non-compliant behavior and is outside the scope of this Security Policy**.

Please note that the Vocera Cryptographic Module is not delivered to end-users as a standalone offering. Rather, it is an integrated component of the Vocera B3000n Communications Badge. The module is pre-installed on the badge at the factory prior to delivery to end-users. Vocera does not provide end-users with any mechanisms to directly access the module or its APIs.

3.1 Secure Management

The following paragraphs describe the steps necessary to ensure that the Vocera Cryptographic Module is running in its validated configuration.

3.1.1 Installation

The module is part of a product application package that is factory-installed. Thus, the module has no independent installation steps that end-users must follow.

3.1.2 Badge Configuration

While the module requires no configuration, the Vocera B3000n Communications Badge must be configured to support the use of the module in its FIPS-Approved mode. The CO is responsible for configuring the communications badge to make proper use of the module.

The CO shall enable FIPS support on the badge properties via the Vocera Server Software System. Instructions to manage the communications badge via the Vocera Server Software System are provided in the *Vocera Badge Configuration Guide* available to the CO via Vocera's online Support Portal. The Vocera Server Software System provides user-friendly utility tools and a web-based administrator console to configure and manage the entire Vocera system.

Vocera B3000n Communications Badges are configured to make use of the VCM by updating a badge configuration file called "badge.properties". This update is accomplished via a utility called the Badge Properties Editor. Instructions on updating the badge.properties file to employ the module are as follows:

1. From the Windows **Start** menu, choose Start > All Programs > Vocera > Badge Properties Editor. The Badge Properties Editor will appear.
2. From the **Badge Type** drop-down menu, choose "B3000n".
3. Select the **Security** tab and do the following:

    - Check the "Enable FIPS" checkbox.

- From the **Authentication** drop-down menu, select “WPA-PSK”, “WPA-PEAP”, or “EAP-TLS”.
- From the **Encryption** drop-down menu, select “AES-CCMP”.

4. Press “Apply” and then “OK” to save any changes.
5. Restart the Vocera Server from the web-based administrator console as instructed in the *Vocera Administration Guide*. The document can also be found in Vocera’s online Support Portal. The badges.properties file on any connected badges will be automatically updated upon Server restart.

The badge operator must use the “Info Menu” on the badge home screen to see the status of FIPS Mode. At this point, FIPS Mode should display that it is set to “on” without operator intervention. The version will display as “VERSION: 3.1”.

NOTE: The “Vocera Only” option from the badge menu must not be used when running the badge in its FIPS configuration.

3.1.3 Initialization

The module has a defined default entry point (DEP) containing code that the OS loader executes automatically when the library is loaded into memory for execution (but prior to the calling application assuming process control over the library). With the badge configured to enable FIPS operation, the module runs the power-up self-tests automatically when it is powered up. If the power-up self-tests complete successfully, the module is deemed to be operating in a FIPS-Approved mode of operation. Successful power-up self-tests displays the following message on the badge display screen:

```
“Power On Self Tests successful.”
```

Upon completion, the module will be in its FIPS mode of operation. Failure of any of the initialization actions will result in a failure of the module to load for execution.

3.2 Operator Guidance

The following sections provide guidance to module operators for ensuring that the module is operating in its FIPS-Approved configuration. The CO is responsible for making sure the module is running in FIPS-Approved mode of operation using the steps provided in Section 3.1 above.

3.2.1 Crypto Officer Guidance

The CO is also responsible for monitoring that the Vocera B3000n Communications Badge’s FIPS configuration is maintained by using only FIPS-Approved functions. To maintain the FIPS configuration, the CO must ensure that that only those algorithms documented in Section 2.7 (Cryptographic Key Management) of this Security Policy are in use.

3.3 User Guidance

The User does not have any ability to modify the configuration of the module. However, if any irregular activity is noticed or the module is consistently reporting errors, then Vocera Customer Support should be contacted.

Users are not responsible for the badge's configuration; this is the responsibility of the CO. Users employ the secure communications services provided by the module. For guidance on using the Vocera B3000n Communications Badge, please refer to the *Vocera Badge User Guide*. The document can be found in Vocera's online Support Portal.

3.3.1 General Operator Guidance

The following provide further guidance for the general operation of the module:

- The module does not store any CSP persistently (beyond the lifetime of an API call). When the API call is complete, zeroization of any temporarily-stored CSPs is performed automatically by the API call itself. Additionally, any CSPs in SDRAM can be zeroized by removing the Smartbadge's battery or unloading the module from memory.

  Keys used in support of TLS and 802.11 communications are automatically zeroized upon session termination. Additionally, the HMAC Integrity Key is used only in the performance of a power-up self-test, and thus is not subject to FIPS zeroization requirements as per FIPS Implementation Guidance 7.4.

- As the module supports service-by-service mode switching, the module's current mode of operation is determined by the service being executed at any given time. Execution of an Approved or allowed service means that the module is in Approved mode; execution of a non-Approved service means that the module is in non-Approved mode.

- To execute the module's power-up self-tests on-demand, the module can be unloaded and reloaded into memory, which will trigger te initiation of the self-tests. Power-cycling the badge by removing and re-inserting the battery will also restart the module and initiate the power-up self-tests.

  Additionally, the calling application can invoke the `POST_DoTests()` API to execute the power-up tests on-demand. If all self-tests complete successfully, the module will send a return value of "true" to the calling application. If any self-test fails, the module will immediately terminate the calling application and initiate a reboot of the badge in order to trigger a full re-initialization of the module.

3.4 Additional Guidance and Usage Policies

The notes below provide additional guidance and policies that must be followed by the calling application (acting as the module's sole authorized operator):

- As the module does not persistently store keys, the calling application is responsible for the storage and zeroization of keys and CSPs passed into and out of the module.

4. Acronyms

Table 10 provides definitions for the acronyms used in this document.

Table 10 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CCKM	Cisco Centralized Key Management
CCM	Counter with CBC-Message Authentication Code
CCMP	Counter with CBC-Message Authentication Code Protocol
CFR	Code of Federal Regulations
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CO	Crypto-Officer
CRNGT	Continuous Random Number Generator Test
CSE	Communications Security Establishment
CSP	Critical Security Parameter
DND	Do Not Disturb
DRBG	Deterministic Random Bit Generator
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol over Local Area Networks
ECB	Electronic Code Book
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
HMAC	Hash-based Message Authentication Code
KAT	Known Answer Test
KDF	Key Derivation Function
LAN	Local Area Network
MD5	Message Digest 5
N/A	Not Applicable
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology

Acronym	Definition
OKC	Opportunistic Key Caching
OS	Operating System
PBX	Private Branch Exchange
PEAP	Protected Extensible Authentication Protocol
PKCS	Public Key Cryptography Standard
PMK	Pairwise Master Key
PSK	Pre-Shared Key
RSA	Rivest, Shamir, Adleman
SDRAM	Synchronous Dynamic Random Access Memory
SHA	Secure Hash Algorithm
SoC	System on a Chip
SP	Special Publication
TLS	Transport Layer Security
VCM	Vocera Cryptographic Module
WLAN	Wireless Local Area Network
WPA-PSK	Wi-Fi Protected Access - Pre-Shared Key

Prepared by:

Corsec Security, Inc.

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com