Vocera Communications, Inc.

Vocera Cryptographic Module

Firmware Version: 6.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.4

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 National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS) 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.

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Vocera.

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 6.0

2.1 Overview

The Vocera C1000 Badge is a wearable communication device that enables clinician agility and accelerates patient care. Small and lightweight (see Figure 1), the Badge redefines healthcare communications by bringing together voice calling, secure messaging, and alerts and alarms in a lightweight wearable.

Figure 1 – Vocera C1000 Badge

Purpose-built for patient care, the Badge is powered by the Vocera Platform, which enables data to be aggregated from most clinical and operational systems used in hospitals today (see Figure 2). The Badge allows patient information from those systems to be presented in parallel with notifications to enable real-time situational awareness and help reduce interruption fatigue.

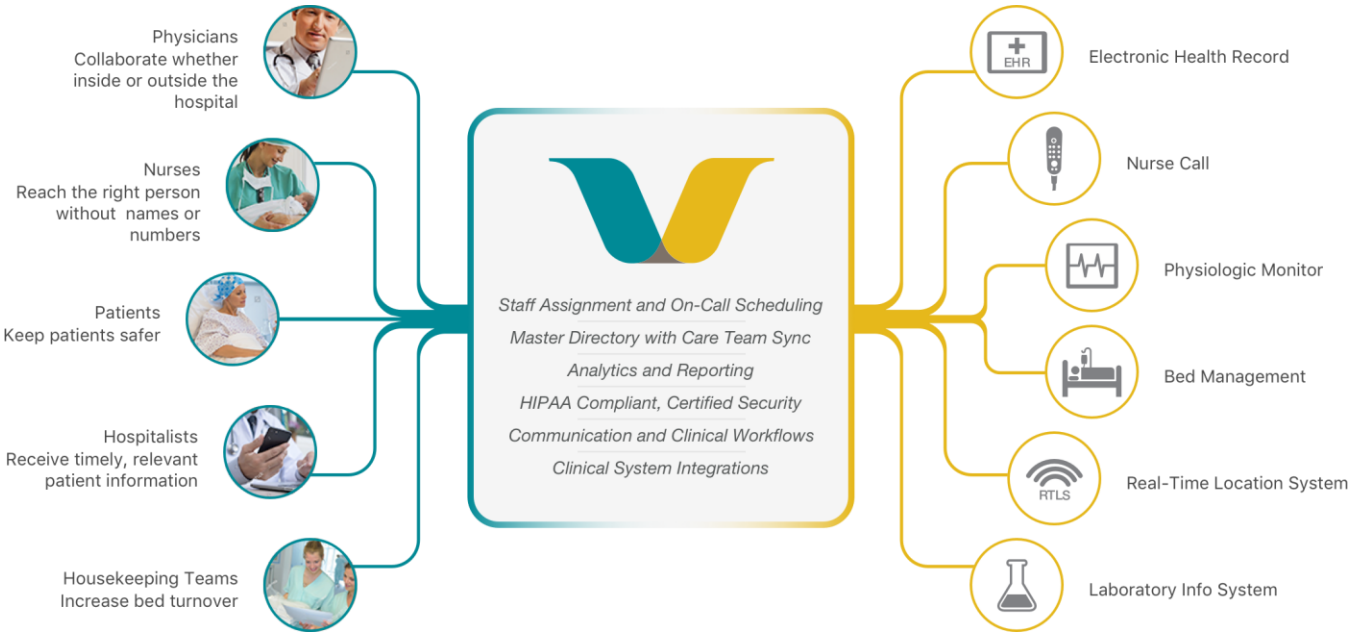

Figure 2 – Vocera Data Aggregation Mapping

The Vocera C1000 Badge enables care team members to make/answer calls hands-free and to connect with the right person by saying a name, role, or group name. It responds to more than 100 voice commands through an optimized speech-recognition engine. Broadcast messages can easily be sent to rapid response groups, and help can be summoned instantly via a dedicated panic button.

Communications are protected via industry-standard communications protocols including TLS[1], DTLS[2], SRTP[3], EAP[4]-TLS, and PEAP[5]. The is a firmware module that provides the cryptographic primitives (including encryption/decryption, hashing, digital signature functions, and key derivation) needed to support the secure communication services to the system. The cryptographic module is installed on the Badge, and various applications on the Badge make use of the VCM to establish secure connections with the Vocera Server and with other Vocera communications devices.

The VCM version 6.0 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	2

[1] TLS – Transport Layer Security
[2] DTLS – Datagram Transport Layer Security
[3] SRTP – Secure Real-Time Transport Protocol
[4] EAP – Extensible Authentication Protocol
[5] PEAP – Protected Extensible Authentication Protocol

Section	Section Title	Level
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A[6]
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

2.2 Module Specification

The VCM is a firmware cryptographic module with an overall security level of the module is 1. The module consists of a firmware module based on the OpenSSL FIPS Object Module (FOM) 2.0.16 and an HMAC SHA-1 digest. The module is compiled into object form (*fipscanister.o*) and then linked to an instance of the OpenSSL cryptographic library at build-time. The larger library is then linked to the calling application. The HMAC SHA-1 digest is stored in *fipscanister.o.sha1*.

The module executes on a Vocera C1000 badge with an NXP i.MX 6 applications processor running Vocera's BadgeOS 6.0 as operating system.

2.2.1 Physical Cryptographic Boundary

As a firmware cryptographic module, the module has no physical components. Physically, the module takes on the characteristics of the host device, the C1000 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 C1000 includes an NXP i.MX 6ULZ applications processor (with a single-core Arm Cortex-A7), 512 MB[9] DDR3L[10], 4 Kb[11] EEPROM[12], and 256 MB NAND Flash. The module is stored in the badge's flash and executes on the applications processor.

2.2.2 Logical Cryptographic Boundary

The logical cryptographic boundary surrounds the firmware library and an HMAC SHA-1 digest. 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 firmware components, as well as the module's logical cryptographic boundary.

[6] N/A – Not Applicable
[7] EMI – Electromagnetic Interference
[8] EMC –Electromagnetic Compatibility
[9] MB – Megabyte
[10] DDR3L – Double Data Rate 3 Low Voltage
[11] Kb – Kilobit
[12] EEPROM – Electrically Erasable Programmable Read-Only Memory

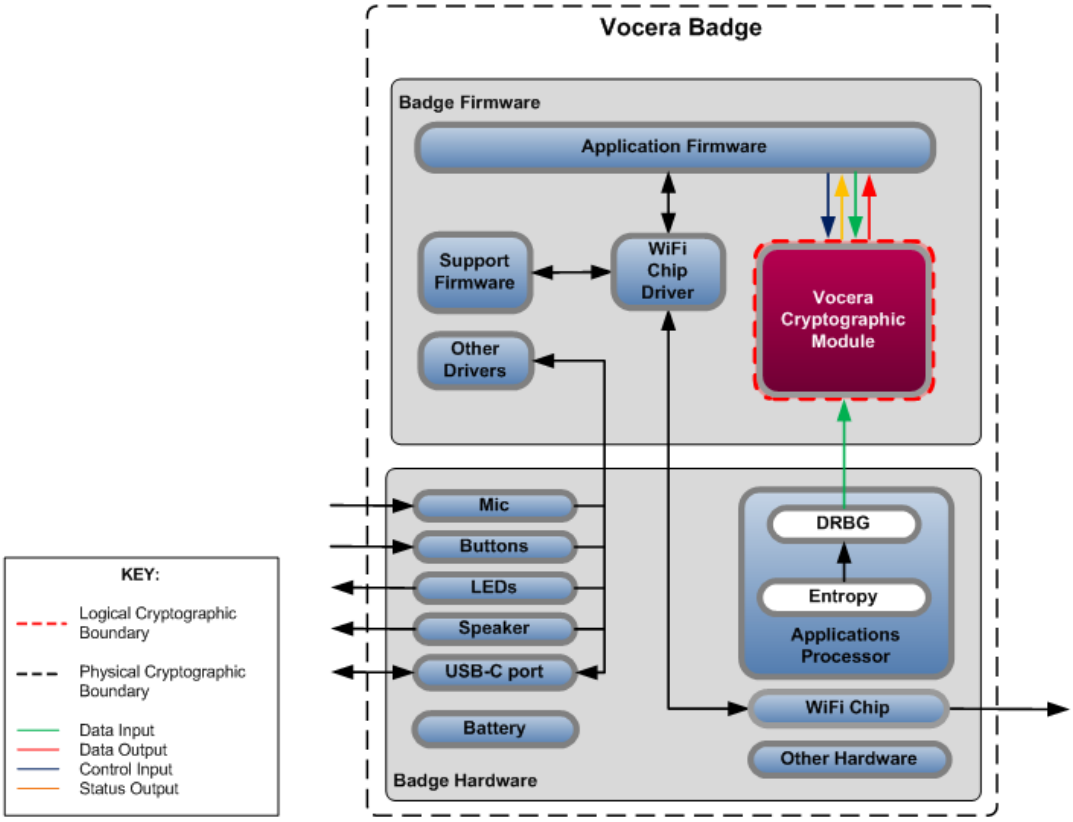

Figure 3 – VCM Cryptographic Boundaries

2.2.3 Algorithm Implementations

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

Table 2 – FIPS-Approved Algorithms

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
A2059	AES[13]	FIPS PUB 197	CBC[14], CTR, ECB[15]	128, 192, 256	encryption/decryption
		NIST SP 800-38B	CMAC[16]	128, 192, 256	MAC generation/verification
		NIST SP 800-38C	CCM[17]	128, 192, 256	encryption/decryption

[13] AES – Advance Encryption Standard
[14] CBC – Cipher Block Chaining
[15] ECB – Electronic Codebook
[16] CMAC – Cipher-based Message Authentication Code
[17] CCM – Counter with CBC-MAC

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
		NIST SP 800-38D	GCM[18]	128, 192, 256	authenticated encryption/decryption
A2059	CVL	NIST SP 800-135rev1	SRTP, TLS 1.2	-	key derivation *No parts of the SRTP and TLS protocols, other than the KDFs[19], have been tested by the CAVP or CMVP.*
A2059	ECDSA	FIPS PUB 186-4	SHA2-224, SHA2-256, SHA2-384, SHA2,512	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	digital signature generation *Used only to support the power-up ECDSA PCT.*
			SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2,512	B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521	digital signature verification
A2059	HMAC[20]	FIPS PUB 198-1	SHA-1[21], SHA2-224, SHA2-256, SHA2-384, SHA2-512	-	message authentication
A2059	KAS-SSC[22]	NIST SP 800-56Arev3	ECC[23]	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	shared secret computation *Key establishment methodology provides between 128 and 256 bits of encryption strength.*
A2059	KTS[24]	NIST SP 800-38D	AES-GCM	128, 192, 256	key wrapping (authenticated encryption/decryption) *Per FIPS 140-2 Implementation Guidance D.9, AES-GCM is an Approved key transport technique.*
		FIPS PUB 197 NIST SP 800-38F FIPS PUB 198-1	AES with HMAC	128, 192, 256	key wrapping (encryption/decryption + authentication) *Per FIPS 140-2 Implementation Guidance D.9, AES with HMAC is an Approved key transport technique.*
A2059	RSA[25]	FIPS PUB 186-2	PKCS1-v1_5[26]	1024, 1536, 2048, 3072	digital signature verification
		FIPS PUB 186-4	PKCS1-v1_5, PSS[27]	2048, 3072, 4096	digital signature verification

[18] GCM – Galois/Counter Mode
[19] KDF – Key Derivation Function
[20] HMAC – (keyed-) Hashed Message Authentication Code
[21] SHA – Secure Hash Algorithm
[22] KAS-SSC – Key Agreement Scheme - Shared Secret Computation
[23] ECC – Elliptic Curve Cryptography
[24] KTS – Key Transport Scheme
[25] RSA – Rivest Shamir Adleman
[26] PKCS1-v1_5 – Public Key Cryptography Standard #1 version 1.5
[27] PSS – Probabilistic Signature Scheme

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
A2059	SHS[28]	FIPS PUB 180-4	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	-	message digest

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

Table 3 – Allowed Algorithms

Algorithm	Caveat	Use
RSA	key establishment methodology provides between 112 and 152 bits of encryption strength	key encryption and decryption *The RSA algorithm may be used by the calling application for encryption or decryption of keys. No claim is made for NIST SP 800-56Brev2 compliance, and no CSPs are established into or exported out of the module using these services.*

2.2.4 Modes of Operation

Upon successful completion of the module's power-up self-tests, the module operates in the Approved mode of operation. Further, when following all installation, configuration, and initialization guidance provided in this Security Policy, the module does not support a non-Approved mode 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 C1000 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 boundary features the physical interfaces of a host badge. Those interfaces are as follows:

- Buttons (see Figure 4 below)
- Speaker
- Microphone
- LEDs[29]
- USB-C headset/charging port
- Battery connector
- Bluetooth/WLAN[30] unit

[28] SHS – Secure Hash Standard
[29] LED – Light-Emitting Diode
[30] WLAN – Wireless Local Area Network

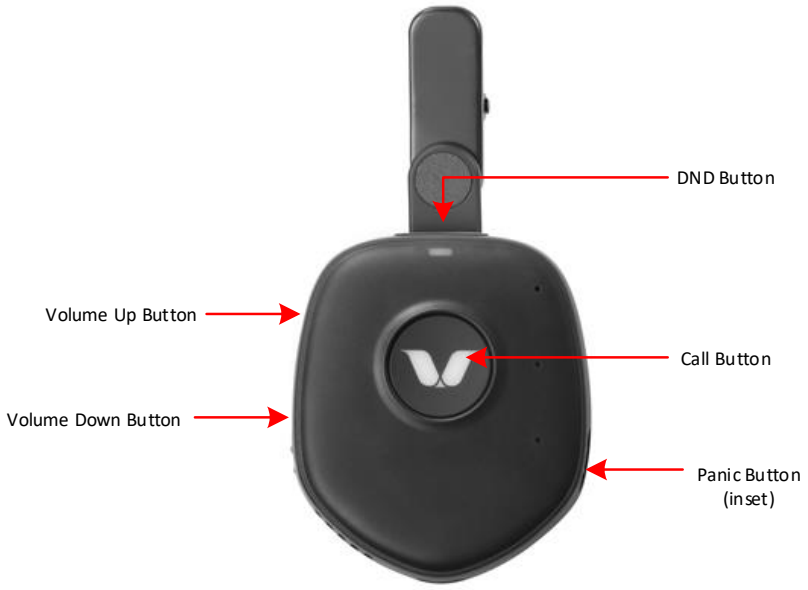

Figure 4 – Vocera C1000 Badge Buttons

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 firmware. The module isolates communications to logical interfaces that are defined in the firmware as an API[31]. The API is mapped to the following four logical interfaces:

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

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	Microphone, USB-C Headset/charging port, Bluetooth/WLAN unit	Function calls that accept, as their arguments, data to be used or processed by the module.
Data Output	Speaker, USB-C Headset/charging port, Bluetooth/WLAN unit	Arguments for a function that specify where the result of the function is stored or returned as a return value.

[31] API – Application Programming Interface

Interface Category	Physical Interface	Logical Interfaces
Control Input	Call button, Hold/DND button, Up/Down volume buttons, USB-C Headset/charging port, Panic button	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output	LEDs, USB-C Headset/charging port	Return values for function calls
Power Input	Battery connector, USB-C Headset/charging port	N/A

2.4 Roles, Services, and Authentication

The sections below describe the module's authorized roles, services, and operator authentication methods.

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. The module supports role-based authentication, and roles are assumed explicitly based on the login credentials.

As a shared library, the module provides cryptographic functions to multiple calling applications simultaneously, all executing as different processes under different process IDs assigned by the operating system. The module can service multiple processes; however, the module only allows one operator per process at any given time. Separation of roles and services across multiple processes is maintained via the process and memory management functions of the underlying operating system.

2.4.2 Operator Services

Descriptions of the services available are provided in Table 5 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 – Mapping of Operator Services to Inputs, Outputs, CSPs, and Type of Access

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Initialize	X	X	Perform initialization of the module	API call parameters	Status	CO password – RX User password – RX
Run self-test on-demand	X	X	Perform power-up self-tests	API call parameters	Status	None
Show status	X	X	Return the current mode of the module	API call parameters	Status	None

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Zeroize	X	X	Zeroize and de-allocate memory containing sensitive data	Power cycle; unload module	None	AES key – W AES CMAC key –W AES GCM key – W AES GCM IV – W HMAC key – W ECDSA public key – W RSA public key – W RSA private key – W
Import random value	X	X	Retrieve and return the specified number of random bits from external source	API call parameters	Status, random bits	None
Generate message digest	X	X	Compute and return a message digest using SHS algorithms	API call parameters, message	Status, hash	None
Generate keyed hash (HMAC)	X	X	Compute and return a message authentication code	API call parameters, key, message	Status, hash	HMAC key – RX
Generate symmetric digest (CMAC)	X	X	Compute and return a cipher message authentication code	API call parameters, key, message	Status, hash	AES CMAC key – RX
Perform symmetric encryption	X	X	Encrypt plaintext using supplied AES key	API call parameters, key, plaintext	Status, ciphertext	AES key – RX
Perform symmetric decryption	X	X	Decrypt ciphertext using supplied AES key	API call parameters, key, ciphertext	Status, plaintext	AES key – RX
Perform authenticated symmetric encryption	X	X	Encrypt plaintext using supplied AES GCM key and IV	API call parameters, key, plaintext	Status, ciphertext	AES GCM key – RX AES GCM IV – RX
Perform authenticated symmetric decryption	X	X	Decrypt ciphertext using supplied AES GCM key and IV	API call parameters, key, ciphertext	Status, plaintext	AES GCM key – RX AES GCM IV – RX
Verify signature	X	X	Verify an ECDSA or RSA digital signature	API call parameters, key, signature, message	Status	ECDSA public key – RX RSA public key – RX
Perform public key encryption	X	X	Perform asymmetric encryption for RSA key transport	API call parameter	Status, ciphertext	RSA public key – RX
Perform private key decryption	X	X	Perform asymmetric decryption for RSA key transport	API call parameter	Status, plaintext	RSA private key – RX

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Establish TLS master secret	X	X	Establish and return TLS master secret	API call parameters, pre-master secret	Status, TLS keys	ECC CDH[32] public component – RX ECC CDH private component – RX TLS pre-master secret – WRX TLS master secret – W
Establish TLS keys	X	X	Establish and return TLS session and integrity keys	API call parameters, master secret	Status, TLS keys	TLS master secret – RX TLS session key – W TLS integrity key – W
Establish SRTP keys	X	X	Establish and return SRTP session and integrity keys	API call parameters, master secret	Status, SRTP keys	SRTP master key – RX SRTP session key – W SRTP integrity key – W

***NOTE**: The module itself does not generate random numbers. Rather, the module fulfills such requests from calling applications by importing random numbers generated by a DRBG and entropy source on the badge's applications processor (which is outside the module's logical boundary).*

2.4.3 Authentication

The module supports role-based authentication. As the module's only operator, the calling application explicitly assumes the CO or User role by passing the appropriate password to the `FIPS_module_mode_set()` function. The password values are specified at build-time and have a minimum length of 16 characters. Any attempt to authenticate with an invalid password will result in an immediate and permanent failure condition, rendering the module unable to enter the FIPS mode of operation, even with subsequent use of a correct password.

Authentication data is loaded into the module during the module build process and otherwise cannot be accessed.

The minimum password length is 16 characters, and each character can be any one of the 256 characters in the ASCII character set. For each attempt to use the authentication mechanism, the probability of a random successful attempt will succeed or a false acceptance will occur is 1/256^16, which is less than 1/10^6. The module permanently disables further authentication attempts after a single failure.

2.5 Physical Security

The VCM is a multiple-chip standalone firmware module. The module runs on a Vocera C1000 Badge consisting of production-grade components that include standard passivation techniques. The module is entirely contained within the hard plastic badge enclosure, which blocks physical access to the module.

2.6 Operational Environment

The module does not provide a general-purpose OS to the user. The module executes on a Vocera C1000 Badge with an NXP i.MX 6 applications processor running Vocera's proprietary BadgeOS 6.0.

The module is not intended to operate on any platform other than the Vocera C1000 Badge, and the module does not provide operators with any means to modify software/firmware components or to load and execute software/firmware that was not included as part of the validation of the module. This constitutes a non-modifiable operational environment.

[32] ECC CDH – Elliptic Curve Cryptography Co-Factor Diffie-Hellman

2.7 Cryptographic Key Management

The module supports the CSPs listed below in Table 6.

Table 6 – Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type / Length	Generation / Input	Output	Storage	Zeroization	Use
AES key	128, 192, 256-bit AES CBC/CTR/ECB/CCM key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; Remove power	Encryption and decryption
AES CMAC key	128-bit AES CMAC key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; Remove power	MAC generation and verification
AES GCM key	128, 192, 256-bit AES GCM key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; Remove power	Encryption and decryption
AES GCM IV[33]	96-bit value	Constructed at its entirety internally deterministically[34]	Never exits the module	Not persistently stored	Unload module; Remove power	Initialization vector for AES GCM
HMAC key	160, 224, 256, 384, or 512-bit HMAC key	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; Remove power	Message authentication with SHA
ECDSA public key	All FIPS-Approved P/B/K-curves	Input via API call parameter	Output in plaintext	Not persistently stored	Unload module; Remove power	Signature verification
RSA private key	2048, 3072, 4096-bit RSA key	Input via API call parameter	Output in plaintext	Not persistently stored	Unload module; Remove power	Decryption
RSA public key	2048, 3072, 4096-bit RSA key	Input via API call parameter	Output in plaintext	Not persistently stored	Unload module; Remove power	Encryption
	1024, 1536, 2048, 3072-bit RSA key	Input via API call parameter	Output in plaintext	Not persistently stored	Unload module; Remove power	Signature verification
ECC CDH private component	All FIPS-Approved P/B/K-curves	Input via API call parameter	Output in plaintext	Not persistently stored	Unload module; Remove power	Shared secret generation
ECC CDH public component	All FIPS-Approved P/B/K-curves	Input via API call parameter	Output in plaintext	Not persistently stored	Unload module; Remove power	Shared secret generation
TLS pre-master secret	384-bit value	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; Remove power	Derivation of the TLS master secret
TLS master secret	384-bit shared secret	Derived internally using the TLS pre-master secret via TLS KDF	Never exits the module	Not persistently stored	Unload module; Remove power	Derivation of the TLS session key and TLS integrity key
TLS session key	128-bit AES key	Derived internally using the TLS master secret via TLS KDF	Output in plaintext	Not persistently stored	Unload module; Remove power	Encryption and decryption of TLS session packets

[33] IV – Initialization Vector
[34] The IV construction method follows section 8.2.1 of *NIST SP 800-38D*.

CSP	CSP Type / Length	Generation / Input	Output	Storage	Zeroization	Use
TLS integrity key	160-bit (minimum) HMAC key	Derived internally using the TLS master secret via TLS KDF	Output in plaintext	Not persistently stored	Unload module; Remove power	Authentication of TLS session packets
SRTP master key	128/192/256-bit shared secret	Input via API call parameter	Never exits the module	Not persistently stored	Unload module; Remove power	Derivation of the SRTP session key and SRTP integrity key
SRTP session key	128/192/256-bit AES-CTR or 128/256-bit AES GCM key	Derived internally using the SRTP master key via SRTP KDF	Output in plaintext	Not persistently stored	Unload module; Remove power	Encryption and decryption of SRTP session packets
SRTP integrity key	160-bit (minimum) HMAC key	Derived internally using the SRTP master key via SRTP KDF	Output in plaintext	Not persistently stored	Unload module; Remove power	Authentication of SRTP session packets
CO password	20 bytes based on the 16-character (minimum) string	HMAC SHA-1 digest of the password is loaded into the module during the module build process	Never exits the module	Not persistently stored	Not needed since it is protected stored by an Approved Algorithm (#A2059).	Crypto Officer authentication
User password	20 bytes based on the 16-character (minimum) string	HMAC SHA-1 digest of the password is loaded into the module during the module build process	Never exits the module	Not persistently stored	Not needed since it is protected stored by an Approved Algorithm (#A2059).	User authentication

The module provides AES-GCM services to a calling application in support of TLS 1.2 communications and supports acceptable GCM cipher suites from section 3.3.1 of *NIST SP 800-52rev2*.

The module follows the TLS 1.2 protocol GCM IV generation method (technique #1) from section A.5 of the *FIPS 140-2 Implementation Guidance*. In compliance with *RFC 5288*, the 96-bit AES-GCM IV consists of 32-bit name field and a 64-bit counter field. If the counter field exhausts the maximum number of possible values for a given key, then the calling application must trigger a new handshake to establish a new encryption key.

2.8 EMI / EMC

The module is a firmware module whose target platform is a Vocera C1000 Badge, which is considered a radio device. The Vocera C1000 Badge was independently tested by Intertek Testing Services NA, Inc. (accredited by the A2LA under certificate number 1455.01) and was awarded FCC ID QGZC1000.

2.9 Self-Tests

Cryptographic self-tests are performed automatically by the module when the module is first powered up and loaded into memory. Additionally, these tests can be performed on demand via removal and re-insertion of the Badge battery or by unloading and reloading the module for execution. 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 the following FIPS-required self-tests automatically at module power-up:

- Firmware Integrity Check using HMAC SHA-1
- Known Answer Tests (KATs)
  - AES-ECB encrypt KAT (128-bit)
  - AES-ECB decrypt KAT (128-bit)
  - AES-CCM encrypt KAT (192-bit)
  - AES-CCM decrypt KAT (192-bit)
  - AES-GCM encrypt KAT (256-bit)
  - AES-GCM decrypt KAT (256-bit)
  - AES-CMAC generation KAT (128/192/256-bit)
  - AES-CMAC verification KAT (128/192/256-bit)
  - HMAC KATs for SHA2-224, SHA2-256, SHA2-384, and SHA2-512
  - ECDSA PCT[35] (P-224/K-233 curves)
  - RSA signature verification KAT (PSS, 2048-bit)
  - ECC CDH Primitive “Z” Computation KAT

The module’s HMAC SHA-1 power-up integrity test fully tests the SHA-1 implementation. Thus, as allowed per section 9.3 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, no independent KATs are needed for the module’s SHA-1 implementation.

The module’s HMAC SHA2-224, HMAC SHA2-256, HMAC SHA2-384, and HMAC SHA2-512 power-up KATs fully test the SHA2-224, SHA2-256, SHA2-384, and SHA2-512 implementations. 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-2 implementations.

The testing of AES-ECB sufficiently tests both the forward and inverse cipher functions. Thus, as allowed per section 9.4 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, explicit forward/inverse cipher testing for the other supported modes of AES are not required.

[35] PCT – Pairwise Consistency Test

The module includes a power-up signature verification KAT for the PSS scheme of RSA, but it supports both the PSS and PKCS1-v1_5 schemes. As allowed per section 9.4 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, no independent KAT for the PKCS1-v1_5 scheme is required.

2.9.2 Conditional Self-Tests

The module includes no conditional self-tests.

2.9.3 Critical Functions Self-Tests

The module includes no critical functions self-tests.

2.9.4 Self-Test Failure Handling

If any self-test fails during module power-up, the module will enter a critical error state. An internal global error flag `fips_selftest_fail` is set and subsequently tested to prevent the invocation of any cryptographic function calls. The module can only recover from the critical error state by unloading and reloading the module or power-cycling the Badge and passing all power-up self-tests.

If this recovery method does not result in the successful execution of the power-up self-tests, then the module will not be able to resume normal operations, and the CO should contact Vocera Communications, Inc. for assistance.

If a power-up self-test fails when performed on-demand via API call, the module will enter a soft error state and provide a failure status indication to the calling application. Provision of the status indicator will automatically clear the error state. It is the responsibility of the calling application to take proper action in response to the status indication.

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 sections below describe how to ensure that the module is operating in its validated configuration. **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 C1000 Badge application firmware. The application firmware is pre-installed onto the Badge at the factory hardware 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 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 Badge must be configured to support the use of the module. The CO must enable FIPS support on the Vocera C1000 Badge by updating a badge configuration file called “badge.properties”. This update is accomplished using the following utilities:

- Badge Properties Editor (BPE) – for settting values for badge properties so the Vocera badges can connect to the wireless network. .
- Badge Configuration Utility (BCU) – for downloading the badge properties, as well as any firmware upgrades, to Vocera badges.

These utilities are installed on Vocera Voice Servers and on standalone configuration computers. To enable FIPS support on the C1000 Badge, the CO shall perform the following tasks (if performing initial badge configuration, use the BPE on the configuration computer):

1. Locate and double-click the Vocera BPE Launcher icon on the desktop the first time. For subsequent logins, access the Vocera BPE Launcher using the URL http://127.0.0.1:8011/#!/, where 127.0.0.1 is the localhost, and 8011 is the Voice Server IP port for BPE.
2. The BPE user interface appears.
3. Select **C1000**, under “Badges”.
4. Select the **Security Settings** on the C1000 configuration page and check the **Enable FIPS** checkbox.

5. Click one of the following:
    a. **Submit** – Allows you to submit the changes.
    b. **Discard Changes** – Allows you to discard the changes and re-enter the badge properties.
6. The Badge Properties Editor creates a badge.properties text file under \vocera\config. Badge properties can now be uploaded to the Badge.
7. Restart the server or BCU. The badge.properties file will be automatically updated upon server restart.

To see the status of the FIPS Mode The badge operator must open a web browser on a Laptop or PC and access the diagnsotic web page by typing in <Badge IP Address>:5000, and then go to the "Network Information" page. Under "Client Information", “FIPS Mode” should display that it is set to “On” without operator intervention. The version will display as “Vocera Cryptographic Module 6.0”.

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). When the badge is configured to enable FIPS operation, the module will begin initialization with the verification of the operator’s password. The invocation of the DEP will pass the CO or User password to the module via the `FIPS_module_mode_set()` function.

If the operator successfully authenticates, the module then immediately runs the power-up self-tests. Execution of these self-tests requires no action from the operator. If the power-up self-tests complete successfully, the module is deemed to be operating in a FIPS-Approved mode of operation. The following status message is then logged in the log file:

```
“crypto: Power on self test passed”
```

Failure of any of the initialization actions will result in a failure of the module to load for execution.

Note that once the “Enable FIPS” checkbox is selected, the selectable options for several other security settings will be limited to only those that are supported by the FIPS-Approved cryptographic methods. Further information regarding badge configuration and the badge.properties file is available in the *Vocera Device Configuration Guide* on Vocera’s online Support Portal.

3.2 Operator Guidance

The following sections provide guidance to module operators for the correct and secure operation of the module. As it relates to this module, the calling application (acting as the module’s sole authorized operator) takes on the role of both Crypto Officer and User.

3.2.1 Crypto Officer Guidance

Subsequent to the module’s successful initialization into FIPS Approved mode of operation, no further management activities are required to ensure that the module runs securely; once initialized, the module only executes in a FIPS-Approved mode of operation. However, if any irregular activity is noticed or the module is consistently reporting errors, then Vocera Customer Support should be contacted.

3.2.2 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 C1000 Badge, please refer to the *Vocera Badge User Guide*. The document can be found in Vocera's online Support Portal.

3.2.3 General Operator Guidance

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

- As a firmware module offering no physical storage media within the logical boundary, the module does not store CSPs 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 volatile memory can be zeroized by removing the Badge's battery or unloading the module from memory. Any persistent key storage occurs outside the module's logical boundary.

- To determine the module's operational status, the `FIPS_mode()` API can be used. A non-zero return value indicates that the module is running in its FIPS-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 the 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 `FIPS_selftests()` API to execute the power-up tests on-demand. A return value of "1" indicates that all self-tests completed successfully; a "0" indicates a test failure. Upon receiving a failure indication, the calling application shall unload and reload the module to trigger a full re-initialization of the module.

3.3 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.

- In the event that power to the module is lost and subsequently restored, the calling application must ensure that any AES-GCM keys used for encryption or decryption are re-distributed.

4. Acronyms

Table 7 provides definitions for the acronyms used in this document.

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CCM	Counter with Cipher Block Chaining – Message Authentication Code
CFR	Code of Federal Regulations
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CSP	Critical Security Parameter
CVL	Component Validation List
DTLS	Datagram Transport Layer Security
DND	Do Not Disturb
DRBG	Deterministic Random Bit Generator
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol over Local Area Network
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECC CDH	Elliptic Curve Cryptography Co-Factor Diffie-Hellman
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
FOM	FIPS Object Module
GCM	Galois/Counter Mode
HMAC	Hash-based Message Authentication Code
IP	Internet Protocol
KAS-SSC	Key Agreement Scheme - Shared Secret Computation
KAT	Known Answer Test
KDF	Key Derivation Function

Acronym	Definition
LAN	Local Area Network
MPE	Media Processing Engine
N/A	Not Applicable
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology
OS	Operating System
PBX	Private Branch Exchange
PCT	Pairwise Consistency Test
PEAP	Protected Extensible Authentication Protocol
PKCS1-v1_5	Public Key Cryptography Standard #1 version 1.5
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Adleman
SDRAM	Synchronous Dynamic Random Access Memory
SHA	Secure Hash Algorithm
SP	Special Publication
SRTP	Secure Real-Time Transport Protocol
TLS	Transport Layer Security
VG6CM	Vocera Gen6 Cryptographic Module
WLAN	Wireless Local Area Network

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