Aruba VMC-TACT Series Virtual Controllers

with ArubaOS FIPS Firmware

Non-Proprietary Security Policy

FIPS 140-2 Level 1

Version 2.1
January 2017

www.arubanetworks.com

1344 Crossman Avenue
Sunnyvale, California 94089
Phone: 408.227.4500
Fax 408.227.4550

Contents

1 Preface

This security policy document can be copied and distributed freely.

1.1 Purpose of this Document

This release supplement provides information regarding the Aruba VMC-TACT Series Virtual Controllers with FIPS 140-2 Level 1 validation from Aruba Networks. The material in this supplement modifies the general Aruba firmware documentation included with this product and should be kept with your Aruba product documentation.

This supplement primarily covers the non-proprietary Cryptographic Module Security Policy for the Aruba VMC-TACT Series Virtual Controllers. This security policy describes how the module meets the security requirements of FIPS 140-2 Level 1 and how to place and maintain the module in a secure FIPS 140-2 mode. This policy was prepared as part of the FIPS 140-2 Level 1 validation of the product.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2, *Security Requirements for Cryptographic Modules*) details the U.S. 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) website at: **http://csrc.nist.gov/groups/STM/cmvp/index.html**
In addition, in this document, the Aruba VMC-TACT Series Virtual Controllers are referred to as the controller or the module.

1.2 Related Documents

The following items are part of the complete installation and operations documentation included with this product:

- *Aruba VMC-TACT Virtual Mobility Controller Installation Guide*
- *ArubaOS 6.4 User Guide*
- *ArubaOS 6.4 CLI Reference Guide*
- *ArubaOS 6.4 Quick Start Guide*
- *ArubaOS 6.4 Upgrade Guide*
- *Aruba AP Installation Guides*

1.2.1 Additional Product Information

More information is available from the following sources:

- The Aruba Networks Web-site contains information on the full line of products from Aruba Networks:
  http://www.arubanetworks.com
- The NIST Validated Modules Web-site contains contact information for answers to technical or sales-related questions for the product:
  http://csrc.nist.gov/groups/STM/cmvp/index.html

2 Overview

Aruba VMC-TACT Series Virtual Controllers includes four varieties (Aruba VMC-TACT-F1, Aruba VMC-TACT-USF1, Aruba VMC-TACT8-F1 and the Aruba VMC-TACT8-USF1) which are optimized for 802.11ac and mobile app delivery. For radio regulatory reasons (with respect to how the controller manages the Access Points), the module ending with -USF1 is to be sold in the US only. The module ending with -F1 is considered 'rest of the world' and must not be used for deployment in the United States. However, from a FIPS perspective, both -USF1 and -F1 models are identical and fully FIPS compliant. The product image (DVD, download, etc) is universal (there is only one). The country code and 8 or full version is set by license key, post-installation. So the customer buys the version by country, and the license makes it an F1 or a USF1 and an 8 or a 32 AP version by setting it when the license is installed.

Aruba VMC-TACT Series Virtual Controllers can support over 32,000 wireless devices and performs stateful firewall policy enforcement at speeds up to 40 Gbps – plenty of capacity for BYOD (Bring Your Own Device) and 802.11ac devices. Fully application-aware, the module prioritizes mobile apps based on user identity and offers exceptional scale for BYOD transactions and device densities.

New levels of visibility, delivered by Aruba AppRF on the module, allow IT to see applications by user, including top web-based applications like Facebook and Box.

The module also manages authentication, encryption, VPN connections, IPv4 and IPv6 services, the Aruba Policy Enforcement Firewall™ with AppRF Technology, Aruba Adaptive Radio Management™, and Aruba RFprotect™ spectrum analysis and wireless intrusion protection.

The module configurations validated during the cryptographic module testing included:

- The firmware version is ArubaOS VMC 6.4.2.0-1.3-FIPS
- The tested platform is a PacStar 451 SSV Small Server (processor Intel i7 running on VMWare ESXI 5.5)

Figure 1: *PacStar 451 SSV Small Server*

More information about PacStar 451 SSV Small Server can be found at **http://pacstar.com/wp-content/uploads/2014/05/PacStar_451_Data_Sheet_patent-020216.pdf**.

2.1 Cryptographic Module Boundaries

For FIPS 140-2 Level 1 validation, the module has been tested as a multi-chip standalone firmware module. The logical cryptographic boundary is defined as the libraries used for the crypto function. The physical boundary is the surface of the computer chassis.

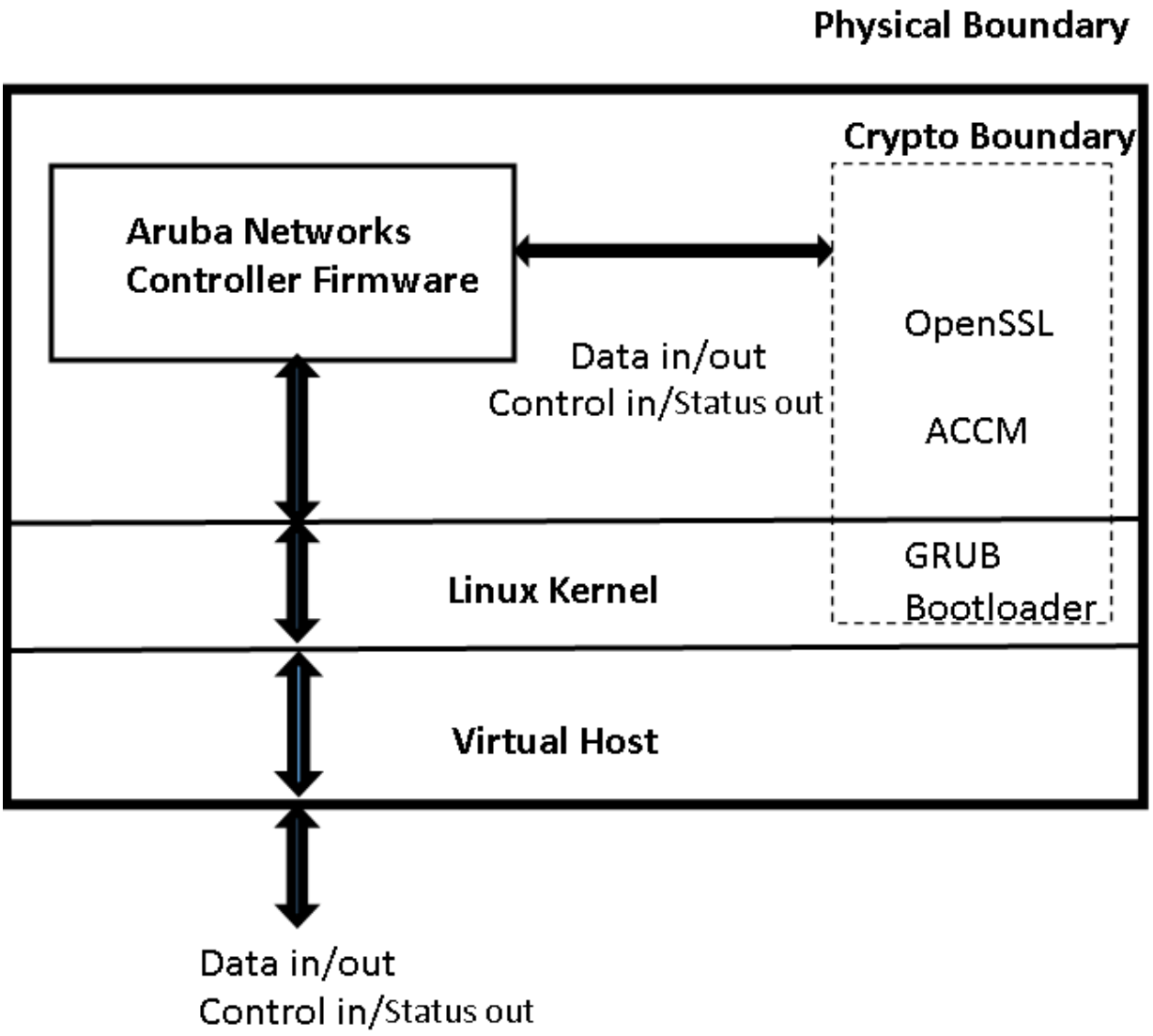

Figure 2: *Functional Block Diagram of the System Component Stack*

2.2 Intended Level of Security

The module is intended to meet overall FIPS 140-2 Level 1 requirements as shown in Table 1.

Table 1 *Intended Level of Security*

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	2
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A
Overall	**Overall module validation level**	**1**

3 Physical Security

The module is a firmware module. It must be run on a production grade platform (such as a standard commercially made PC, laptop, server, etc) to meet requirements from FIPS 140-2 level 1.

4 Operational Environment

The operational environment is limited and non-modifiable. The module was tested on PacStar 451 SSV Small Server (Intel i7 processor running on VMWare ESXI 5.5). The platform used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B.

5 Logical Interfaces

Interfaces on the module can be categorized as the following FIPS 140-2 logical interfaces:

- Data Input Interface
- Data Output Interface
- Control Input interface
- Status Output Interface
- Power Interface

All of these physical interfaces are separated into logical interfaces defined by FIPS 140-2, as described in the following table.

Table 2 *FIPS 140-2 Logical Interfaces*

FIPS 140-2 Logical Interface	Module Virtual Interface	Physical Interface
Data Input Interface	Virtual Ethernet Ports	• Host Platform Ethernet ports • Host Platform Keyboard and Mouse
Data Output Interface	Virtual Ethernet Ports	• Host Platform Ethernet Ports
Control Input Interface	Virtual Control Input Ports	• Host Platform Ethernet Ports • Host Platform

Table 2 *FIPS 140-2 Logical Interfaces*

		Keyboard and Mouse
Status Output Interface	Virtual Status Output Port	• Host Platform Video Monitor
Power Interface	N/A	• Host PC Power Interface

Data input and output, control input, status output, and power interface are defined as follows:

- Data input and output are the packets that use the firewall, VPN, and routing functionality of the modules.
- Control input consists of virtual control inputs for power and reset through the power and reset interface. It also consists of all of the data that is entered into the controller while using the Host interfaces.
- Status output consists of the status indicators displayed through the status data that is output from the module while using the Host management interfaces, and the log file.
- The hosts console indicates the virtual state such as power-up (or rebooting), utilization level, and activation state. The log file records the results of self-tests, configuration errors, and monitoring data.
- A power supply is used by the virtualization host.

The module distinguishes between different forms of data, control, and status traffic over the network ports by analyzing the packets header information and contents.

6 Roles and Services

The module supports role-based authentication, meeting level 2 requirements. There are two roles in the module that operators may assume: a Crypto Officer role and a User role. The Administrator maps to the Crypto-Officer role and the client Users map to the User role.

6.1 Crypto Officer Role

The Crypto Officer role has the ability to configure, manage, and monitor the module. Three management interfaces can be used for this purpose:

- SSHv2 CLI

  The Crypto Officer can use the CLI to perform non-security-sensitive and security-sensitive monitoring and configuration. The CLI can be accessed remotely by using the SSHv2 secured management session over the Ethernet ports or locally over the serial port.

- Web Interface

  The Crypto Officer can use the Web Interface as an alternative to the CLI. The Web Interface provides a highly intuitive, graphical interface for a comprehensive set of controller management tools. The Web Interface can be accessed from a TLS-enabled Web browser using HTTPS (HTTP with Secure Socket Layer) on logical port 4343.

- SNMP v3

  The Crypto Officer can also use SNMPv3 to remotely perform non-security-sensitive monitoring and use ‘get’ and ‘getnext’ commands.

See the table below for descriptions of the services available to the Crypto Officer role.

Table 3 *Crypto-Officer Services*

Service	Description	Input	Output	CSP/Algorithm Access (please see table 9 below for details)
SSHv2	Provide authenticated and encrypted remote management sessions while using the CLI	SSHv2 key agreement parameters, SSH inputs, and data	SSHv2 outputs and data	27, 28 (read/write/delete)
SNMPv3	Provides ability to query management information	SNMPv3 requests	SNMPv3 responses	32, 33, 34 (read/write/delete)
IKEv1/IKEv2-IPSec	Provide authenticated and encrypted remote management sessions to access the CLI functionality	IKEv1/IKEv2 inputs and data; IPSec inputs, commands, and data	IKEv1/IKEv2 outputs, status, and data; IPSec outputs, status, and data	1,19 (read) 6,7,8 9,10,11 (read/write/delete) 20, 21, 22, 23, 24, 25 and 26 (read/delete)
Configuring Network Management	Create management Users and set their password and privilege level; configure the SNMP agent	Commands and configuration data	Status of commands and configuration data	1,32, 33 (read) 34 (delete)

Table 3 *Crypto-Officer Services*

Configuring Module Platform	Define the platform subsystem firmware of the module by entering Bootrom Monitor Mode, File System, fault report, message logging, and other platform related commands	Commands and configuration data	Status of commands and configuration data	None
Configuring the module	Define synchronization features for module	Commands and configuration data	Status of commands and configuration data	None
Configuring Internet Protocol	Set IP functionality	Commands and configuration data	Status of commands and configuration data	None
Configuring Quality of Service (QoS)	Configure QOS values for module	Commands and configuration data	Status of commands and configuration data	None
Configuring VPN	Configure Public Key Infrastructure (PKI); configure the Internet Key Exchange (IKEv1/IKEv2) Security Protocol; configure the IPSec protocol	Commands and configuration data	Status of commands and configuration data	1,19 (read) 15,16, 17, 18(read) 19, 20, 21, 22, 23, 24,25 and 26 (delete)
Configuring DHCP	Configure DHCP on module	Commands and configuration data	Status of commands and configuration data	None
Configuring Security	Define security features for module, including Access List, Authentication, Authorization and Accounting (AAA), and firewall functionality	Commands and configuration data	Status of commands and configuration data	12, 13, 14 (read/write/delete) 1 (read)
Manage Certificates	Install, rename, and delete X.509 certificates	Commands and configuration data; Certificates and keys	Status of certificates, commands, and configuration	15, 16, 17,18 (write/delete)
HTTPS over TLS	Secure browser connection over Transport Layer Security acting as a Crypto Officer service (web management interface)	TLS inputs, commands, and data	TLS outputs, status, and data	6,7,8, 29, 30 and 31 (read/write/delete), 4,5 (read/write) 2.3 (read)

Table 3 *Crypto-Officer Services*

Status Function	Cryptographic officer may use CLI "show" commands or view WebUI via TLS to view the controller configuration, routing tables, and active sessions; view health, temperature, memory status, voltage, and packet statistics; review accounting logs, and view physical interface status	Commands and configuration data	Status of commands and configurations	None
IPSec tunnel establishment for RADIUS protection	Provided authenticated/encrypted channel to RADIUS server	IKEv1/IKEv2 inputs and data; IPSec inputs, commands, and data	IKEv1/IKEv2 outputs, status, and data; IPSec outputs, status, and data	12 and 19 (read/write/delete) 20, 21, 22, 23, 24, 25 and 26 (write/delete) 1(read) 4,5 (read/write), 2.3 (read)
Self-Test	Perform FIPS start-up tests on demand	None	Error messages logged if a failure occurs	None
Configuring Bypass Operation	Configure bypass operation on the module	Commands and configuration data	Status of commands and configuration data	None
Updating Firmware	Updating firmware on the module	Commands and configuration data	Status of commands and configuration data	1, 39 (read)
Configuring Online Certificate Status Protocol (OCSP) Responder	Configuring OCSP responder functionality	OCSP inputs, commands, and data	OCSP outputs, status, and data	27, 28, 29, 30 (read)
Configuring Control Plane Security (CPSec)	Configuring Control Plane Security mode to protect communication with APs using IPSec and issue self signed certificates to APs	Commands and configuration data, IKEv1/IKEv2 inputs and data; IPSec inputs, commands, and data	Status of commands, IKEv1/ IKEv2 outputs, status, and data; IPSec outputs, status, and data	12 and 19 (read/write/delete) 20, 21, 23, 22, 24, 25 and 26 (write/delete)

Table 3 *Crypto-Officer Services*

			and configuration data, self signed certificates	1(read)4,5 (read/write), 2.3 (read)
Zeroization	The cryptographic keys stored in SDRAM memory can be zeroized by rebooting the module. The cryptographic keys (IKEv1 Pre-shared key and 802.11i Pre-Shared Key) stored in the flash can be zeroized by using command 'ap wipe out flash' or by overwriting with a new secret. The 'no' command in the CLI can be used to zeroize IKE, IPsec and CA CSPs. Please See CLI guide for details.The other keys/CSPs (KEK, RSA/ECDSA public key/private key and certificate) stored in Flash memory can be zeroized by using command 'write erase all.	Command	Progress information	All CSPs will be destroyed.

6.2 User Role

Table 4 below lists the services available to User role:

Table 4 *User Service*

Service	Description	Input	Output	CSP Access (please see table 9 below for CSP details)
IKEv1/IKEv2-IPSec	Access the module's IPSec services in order to secure network traffic	IPSec inputs, commands, and data	IPSec outputs, status, and data	6,7,8, 9,10,11 (read, write, delete) 15,16,17,18 (read) 20, 21, 22, 23, 24, 25 and 26 (read/delete) 4,5 (read/write), 2.3 (read)
HTTPS over TLS	Access the module's TLS services in order to secure network traffic	TLS inputs, commands, and data	TLS outputs, status, and data	6,7,8, 9, 10, 11. 29, 30, 31 (read/write/delete) 4,5 (read/write), 2.3 (read)

EAP-TLS termination	Provide EAP-TLS termination	EAP-TLS inputs, commands and data	EAP-TLS outputs, status and data	6,7,8, 29, 30, 31 (read/delete), 4,5 (read/write) 2.3 (read)
802.11i Shared Key Mode	Access the module's 802.11i services in order to secure network traffic	802.11i inputs, commands and data	802.11i outputs, status and data	35, 36, 37 and 38 (create/read/delete) 4,5 (read/write)
802.11i with EAP-TLS	Access the module's 802.11i services in order to secure network traffic	802.11i inputs, commands and data	802.11i outputs, status, and data	15,16,17,18 (read) 35, 36, 37 and 38 (read/delete) 4,5 (read/write)

6.3 Authentication Mechanisms

The module supports role-based authentication. Role-based authentication is performed before the Crypto Officer enters privileged mode using admin password via Web Interface or SSHv2 or by entering enable command and password in console. Role-based authentication is also performed for User authentication.

This includes password and RSA/ECDSA-based authentication mechanisms. The strength of each authentication mechanism is described below.

Table 5 *Estimated Strength of Authentication Mechanisms*

Authentication Type	Role	Strength
Password-based authentication	Crypto Officer	Passwords are required to be a minimum of eight characters and a maximum of 64 with a minimum of one letter and one number. Given these restrictions, the probability of randomly guessing the correct sequence is one (1) in 3,608,347,333,959,680 (this calculation is based on the assumption that the typical standard American QWERTY computer keyboard has 10 Integer digits, 52 alphabetic characters, and 32 special characters providing 94 characters to choose from in total. The calculation should be 94^8 (Total number of 8-digit passwords) – 84^8 (Total number of 8-digit passwords without numbers) – 42^8 (Total number of 8-digit passwords without letters) + 32^8 (Total number of 8-digit passwords without letters or numbers, added since it's double-counted in the previous two subtractions) = 3,608,347,333,959,680). At optimal network conditions (assuming 1ms round-trip latency), an attacker would only get 60,000 guesses per minute. Therefore the associated probability of a successful random attempt during a one-minute period is 60,000/3,608,347,333,959,680, which is less than 1 in 100,000 required by FIPS 140-2.

Password-based authentication (802.11i Pre-shared secret)	User	Same authentication mechanism strength as CO role above.
Password-based authentication (User Password)	User	Same authentication mechanism strength as CO role above.
Password-based authentication (IKEv1)	User	A 64 ASCII (128 HEX) character pre-shared string is randomly chosen by the administrator. It may consist of upper and lower case alphabetic characters, numeric characters and 32 special characters. The probability of randomly guessing the correct sequence is one (1) in 94^64. This calculation is based on the assumption that the typical standard American QWERTY computer keyboard has 10 Integer digits, 52 alphabetic characters, and 32 special characters providing 94 characters to choose from in total. Therefore, the associated probability of a successful random attempt is approximately 1 in 94^64, which is less than 1 in 1,000,000 required by FIPS 140-2. At optimal network conditions (assuming 1ms round-trip latency), an attacker would only get 60,000 guesses per minute. Therefore the associated probability of a successful random attempt during a one-minute period is 60,000/94^64, which is less than 1 in 100,000 required by FIPS 140-2.
RSA-based authentication (IKEv1, IKEv2 and TLS)	User	The module supports 2048-bit RSA key authentication during IKEv1, IKEv2 and TLS. RSA 2048 bit keys correspond to 112 bits of security. Assuming the low end of that range, the associated probability of a successful random attempt is 1 in 2^112, which is less than 1 in 1,000,000 required by FIPS 140-2. At optimal network conditions (assuming 1ms round-trip latency), an attacker would only get 60,000 guesses per minute. Therefore the associated probability of a successful random attempt during a one-minute period is 60,000/2^112, which is less than 1 in 100,000 required by FIPS 140-2.
ECDSA-based authentication IKEv1, IKEv2 and TLS	User	ECDSA signing and verification is used to authenticate to the module during IKEv1/IKEv2. Both P-256 and P-384 curves are supported. ECDSA P-256 provides 128 bits of equivalent security, and P-384 provides 192 bits of equivalent security. Assuming the low end of that range, the associated probability of a successful random attempt during a one-minute period is 1 in 2^128, which is less than 1 in 1,000,000 required by FIPS 140-2. At optimal network conditions (assuming 1ms round-trip latency), an attacker would only get 60,000 guesses per minute. Therefore the associated probability of a successful random attempt during a one-minute period is 60,000/2^128, which is less than 1 in 100,000 required by FIPS 140-2.

6.4 Cryptographic Algorithms and Key Management

6.4.1 Implemented Algorithms

The module contains the following cryptographic algorithm implementations/crypto libraries to implement the different FIPS approved cryptographic algorithms that will be used for the corresponding security services supported by the module in FIPS mode:

- ArubaOS OpenSSL library algorithm implementation
- ArubaOS Crypto library algorithm implementation
- AOS VMC GRUB Bootloader library algorithm implementation.

Note that not all algorithm modes that appear on the module's CAVP certificates are utilized by the module, and the table below lists only the algorithm modes that are utilized by the module.

The module supports the following cryptographic implementations.

- ArubaOS OpenSSL library implements the following FIPS-approved algorithms:

Table 6 *Cryptographic Algorithms implemented by ArubaOS OpenSSL library*

Algorithms	Algorithm Certificates	Used and Tested during the Power Up Self-Tests
AES	#3778	• CBC (128, 192 and 256 bits) • CFB128 (128 bits) • AES CCM (128 bits)
CVL (SP800-135)	#718	• IKEv1, TLS, SSH and SNMP KDF
DRBG	#1044	• AES-256 CTR_DRBG
ECDSA	#813	• FIPS 186-4, PKG: CURVES(P-256 P-384); • SigGen: CURVES(P-256: (SHA-256, 384) P-384: (SHA-256, 384); • SigVer: CURVES(P-256: (SHA-256, 384) P-384: (SHA-256, 384)
HMAC	#2474	• HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512. Please note that the module performs the Power On Self-Test to HMAC-SHA512, but not use it in other security services at this time.

KBKDF (SP800-108)	#80	• CTR_Mode
RSA	#1945	• 186-4KEY(gen); 2048 • ALG[RSASSA-PKCS1_V1_5] SIG(gen) (2048 SHA(256 , 384 , 512)); SIG(Ver) (2048 SHA(1, 256 , 384)) NOTE: Only SHA-256 used for signing in this module
SHS	#3145	• SHA-1, SHA-256, SHA-384 and SHA-512. Please note that the module performs the Power On Self-Test to SHA-512, but not use it in other security services at this time.
Triple-DES	#2099	• TCBC (3-key Triple-DES)

- ArubaOS Common Cryptographic library implementation supports the following FIPS Approved Algorithms:

Table 7 *Cryptographic Algorithms implemented by ArubaOS Common Cryptographic library*

Algorithms	Algorithm Certificates	Used and Tested during the Power Up Self-Tests
AES	#3845	• CBC (128, 192 and 256 bits), • AES GCM (128 and 256 bits)[1]
CVL	#734	• SP800-135 IKEv2 and TLS KDF
ECDSA	#830	• FIPS 186-4, PKG: CURVES(P-256 P-384); • SigGen: CURVES(P-256: (SHA-256, 384) P-384: (SHA-256, 384); • SigVer: CURVES(P-256: (SHA-256, 384) P-384: (SHA-256, 384)
HMAC	#2494	• HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512.Please note that the module performs the

[1] The IV is generated internally using section 8.2.2 (as listed in Cert. #3845).

		Power On Self-Test to HMAC-SHA512, but not use it in other security services at this time.
RSA	#1964	• 186-4KEY(gen); 2048 • ALG[RSASSA-PKCS1_V1_5] SIG(gen) (2048 SHA(256 , 384)); SIG(Ver) (2048 SHA(1 , 256, 384))
SHS	#3167	• SHA-1, SHA-256, SHA-384 and SHA-512. Please note that the module performs the Power On Self-Test to SHA-512, but not use it in other security services at this time.
Triple-DES	#2118	• TCBC (3-key Triple-DES)

- AOS VMC GRUB Bootloader library implements the following FIPS-approved algorithms:

Table 8 – *Cryptographic Algorithms implemented by AOS VMC GRUB Bootloader library*

Algorithms	Algorithm Certificates	Used and Tested during the Power Up Self-Tests
RSA	#2082	• FIPS186-4: ALG[RSASSA-PKCS1_V1_5] SIG(Ver) (2048 SHA(1,256))
SHS	#3338	• SHA-1 • SHA-256

Non-FIPS Approved but Allowed Cryptographic Algorithms

- Diffie-Hellman (key agreement; key establishment methodology provides 112 bits of encryption strength)
- EC Diffie-Hellman (key agreement; key establishment methodology provides 128 or 192 bits of encryption strength)
- NDRNG
- RSA (key wrapping; key establishment methodology provides 112 bits of encryption strength)

NOTE: IKEv1, IKEv2, TLS, SSH and SNMP protocols have not been reviewed or tested by the CAVP and CMVP.

6.5 Critical Security Parameters

The following are the Critical Security Parameters (CSPs) used in the module.

Table 9 *CSPs/Keys Used in the module*

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
General Keys/CSPs					
1	Key Encryption Key (KEK)	Triple-DES (192 bits)	Hardcoded during manufacturing. Used only to protect keys stored in the flash, not for key transport.	Stored in Flash memory (plaintext).	Zeroized by using command 'write erase all'.
2	DRBG entropy input	SP800-90a CTR_DRBG (512 bits)	Entropy inputs to the DRBG function used to construct the DRBG seed. 64 bytes are gotten from the entropy source on each call by any service that requires a random number. Testing estimates 505.26 bits of entropy are returned in the 512 bit string.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
3	DRBG seed	SP800-90a CTR_DRBG (384-bits)	Input to the DRBG that determines the internal state of the DRBG. Generated using DRBG derivation function that includes the entropy input from the entropy source, , by any service that requires a random number	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
4	DRBG Key	SP800-90a CTR_DRBG (256 bits)	This is the DRBG key used for SP800-90a CTR_DRBG.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module

Table 9 *CSPs/Keys Used in the module*

5	DRBG V	SP800-90a CTR_DRBG V (128 bits)	Internal V value used as part of SP800-90a CTR_DRBG	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
6	Diffie-Hellman private key	Diffie-Hellman Group 14 (224 bits)	Generated internally by calling FIPS approved DRBG (cert #1044) during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
7	Diffie-Hellman public key	Diffie-Hellman Group 14 (2048 bits)	Generated internally by calling FIPS approved DRBG (cert #1044) during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
8	Diffie-Hellman shared secret	Diffie-Hellman Group 14 (2048 bits)	Established during Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
9	EC Diffie-Hellman private key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally by calling FIPS approved DRBG (cert #1044) during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
10	EC Diffie-Hellman public key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally by calling FIPS approved DRBG (cert #1044) during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

Table 9 *CSPs/Keys Used in the module*

11	EC Diffie-Hellman shared secret	EC Diffie-Hellman (Curves: P-256 or P-384)	Established during EC Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
12	RADIUS server shared secret	8-128 characters shared secret	Entered by CO role. Used for RADIUS server authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
13	Enable secret	8-64 characters password	Entered by CO role. Used for CO role authentication.	Stored in Flash memory (ciphertext) encrypted with KEK	Zeroized by using command 'write erase all' or by overwriting with a new secret
14	User Password	8-64 characters password	Entered by CO role. Used for User role authentication.	Stored in Flash memory (ciphertext) encrypted with KEK	Zeroized by using command 'write erase all' or by overwriting with a new secret
15	RSA Private Key	RSA 2048 bit private key	This key is generated by calling FIPS approved DRBG (cert #1044) in the module. Used for IKEv1, IKEv2, TLS, OCSP (signing OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all'
16	RSA public key	RSA 2048 bits public key	This key is generated by calling FIPS approved DRBG (cert #1044) in the module. Used for IKEv1, IKEv2, TLS, OCSP (verifying OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all'
17	ECDSA Private Key	ECDSA suite B P-256 and P-384 curves	This key is generated by calling FIPS approved DRBG (cert #1044) in the module. Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all'

Table 9 *CSPs/Keys Used in the module*

18	ECDSA Public Key	ECDSA suite B P-256 and P-384 curves	This key is generated by calling FIPS approved DRBG (cert #1044) in the module. Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all'.
IPSec/IKE					
19	IKEv1 Pre-shared secret	Shared secret (64 ASCII or 128 HEX characters)	Entered by CO role. Used for IKEv1 peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
20	skeyid	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was established via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving other keys in IKE protocol implementation.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.
21	skeyid_d	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving IKE session authentication key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
22	SKEYSEED	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv2) and it will be used for deriving IKE session authentication key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

Table 9 *CSPs/Keys Used in the module*

23	IKE session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	The IKE session (IKE Phase I) authentication key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IKEv1/IKEv2 payload integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
24	IKE session encryption key	Triple-DES (192 bits) /AES (128/192/256 bits)	The IKE session (IKE Phase I) encrypt key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IKE payload protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
25	IPSec session encryption key	Triple-DES (192 bits) / AES and AES-GCM (128/256 bits) NOTE: 192 bit CAVS tested, but not used.	The IPsec (IKE phase II) encryption key. This key is derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IPSec traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
26	IPSec session authentication key	HMAC-SHA-1 (160 bits)	The IPsec (IKE Phase II) authentication key. This key is derived via using the KDF defined in SP800-135 KDF (IKEv1/IKEv2). Used for IPSec traffics integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
SSHv2					
27	SSHv2 session key	AES (128/192/256 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used for SSHv2 traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

Table 9 *CSPs/Keys Used in the module*

28	SSHv2 session authentication key	HMAC-SHA-1 (160-bit)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used for SSHv2 traffics integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
TLS					
29	TLS pre-master secret	48 bytes secret	This key is transferred into the module, protected by TLS RSA public key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
30	TLS session encryption key	AES 128/192/256 bits	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used for TLS traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
31	TLS session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used for TLS traffic integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
SNMPv3					
32	SNMPv3 authentication password	8-64 characters password	Entered by CO role. User for SNMPv3 authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
33	SNMPv3 engine ID	8-64 characters password	Entered by CO role. A unique string used to identify the SNMP engine.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret

Table 9 *CSPs/Keys Used in the module*

34	SNMPv3 session key	AES-CFB key (128 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SNMPv3). Used for SNMPv3 traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
802.11i					
35	802.11i Pre-shared secret	Shared secret (8-64 characters)	Entered by CO role. Used for 802.11i client/server authentication	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
36	802.11i Pair-Wise Master key (PMK)	Shared secret (256 bits)	The PMK is transferred to the module, protected by IPSec secure tunnel. Used to derive the Pairwise Transient Key (PTK) for 802.11i communications.	Stored in SDRAM (plaintext).	Zeroized by rebooting the module
37	802.11i Pairwise Transient Key (PTK)	Shared secret (512 bits)	This key is used to derive 802.11i session key by using the KDF defined in SP800-108.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
38	802.11i session key	AES-CCM (128 bits)	Derived during 802.11i 4-way handshake by using the KDF defined in SP800-108.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
Factory Key					
39	Factory CA Public Key	RSA (2048 bits)	This is RSA public key. Loaded into the module during manufacturing. Used for Firmware verification.	Stored in Flash encrypted with KEK	Zeroized by using command 'ap wipe out flash'

6.6 Self-Tests

The module performs Power On Self-Tests on power up. In addition, the module also performs Conditional tests after being configured into the FIPS mode. In the event any self-test fails, the module will enter an error state, log the error, and reboot automatically.

The module performs the following POSTs (Power On Self-Tests):

- ArubaOS OpenSSL library
  - AES encrypt KAT
  - AES decrypt KAT
  - AES-CCM encrypt KAT
  - AES-CCM decrypt KAT
  - DRBG KAT (plus all required health checks specified in SP800-90a, section 11.3)
  - ECDSA Pairwise Consistency Test
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KATs
  - RSA sign KAT
  - RSA verify KAT
  - SHS (SHA1, SHA256, SHA384 and SHA512) KATs
  - Triple-DES encrypt KAT
  - Triple-DES decrypt KAT

- ArubaOS Common Cryptographic library
  - AES encrypt KAT
  - AES decrypt KAT
  - AES-GCM encrypt KAT
  - AES-GCM decrypt KAT
  - ECDSA Pairwise Consistency Test
  - HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KATs
  - RSA sign KAT
  - RSA verify KAT
  - SHA (SHA1, SHA256, SHA384 and SHA512) KATs
  - Triple-DES encrypt KAT
  - Triple-DES decrypt KAT

- AOS VMC GRUB Bootloader library
  - Firmware Integrity Test: RSA PKCS#1 v1.5 (2048 bits) signature verification with SHA-1

The module performs the following Conditional Tests:

- ArubaOS OpenSSL library
  - Bypass Tests (Wired Bypass Test and Wireless Bypass Test)
  - CRNG Test on Approved DRBG
  - ECDSA Pairwise Consistency Test
  - RSA Pairwise Consistency Test
  - Firmware Load Test- RSA PKCS#1 v1.5 (2048 bits) signature verification with SHA-256

- ArubaOS Common Cryptographic library
  - ECDSA Pairwise Consistency Test
  - RSA Pairwise Consistency Test
- AOS VMC GRUB Bootloader library
  - Firmware Load Test - RSA PKCS#1 v1.5 (2048 bits) signature verification with SHA-256
- CRNG Test for NDRNG

Self-test results are logged in a log file. Upon successful completion of the power-up self tests, the module logs a KATS: passed message into a log file. Confirm the file update by checking the associated time of the file.

In the event of the KATs failure, the log file records a message. The following are typical example messages depending on the algorithm being tested:

- `AES256 HMAC-SHA1 hash failed`
- `AES256 Encrypt failed`
- `AES256 Decrypt Failed`
- `3DES HMAC-SHA1 hash failed`
- `3DES Encrypt failed`
- `3DES Decrypt Failed`
- `AESCCM Encrypt Failed`

This text is followed by this message:

```
The POST Test failed!!!!
Rebooting…
```

6.7 Alternating Bypass State

The module implements an alternating bypass state when:

- a port is configured in trusted mode to provide unauthenticated services
- a configuration provides wireless access without encryption

The alternating bypass status can be identified by retrieving the port configuration or the wireless network configuration.

7 Installing the Module

This chapter covers the installation of the VMC-TACT Controllers with FIPS 140-2 Level 1 validation. The Crypto Officer is responsible for ensuring that the following procedures are used to install the module properly.

This chapter covers the following installation topics:

- Requirements for the module components
- Selecting a proper environment for the module
- Install the module on the hypervisor server
- Power on the module using virtual machine management client

7.1 Pre-Installation Checklist

You will need the following during installation:

- Aruba VMC-TACT Controller components (host server, VM Host SW and Aruba VMC-TACT installation disk).
- Cool, non-condensing air 0 to 40 ºC (32 to 104 ºF). May require air conditioning.
- Management Station (PC) with 10/100 Mbps Ethernet port and virtual machine management client software.

7.1.1 Product Examination

The units are shipped to the Crypto Officer in factory-sealed boxes using trusted commercial carrier shipping companies. The Crypto Officer should examine the carton for evidence of tampering. Tamper-evidence includes tears, scratches, and other irregularities in the packaging.

7.1.2 Package Contents

The product carton should include the following:

- Aruba VMC-TACT Series Virtual Controller CD
- Aruba User Documentation CD

8 Ongoing Management

The Aruba VMC-TACT Controllers meet FIPS 140-2 Level 1 requirements. The information below describes how to keep the controller in FIPS-approved mode of operation. The Crypto Officer must ensure that the controller is kept in a FIPS-approved mode of operation.

8.1 Crypto Officer Management

The Crypto Officer must ensure that the controller is always operating in a FIPS-approved mode of operation. This can be achieved by ensuring the following:

- The admin role must be root.
- Passwords must be at least eight characters long.
- VPN services can only be provided by IPsec or L2TP over IPsec.
- Access to the controller Web Interface is permitted only using HTTPS over a TLS tunnel. Basic HTTP and HTTPS over SSL are not permitted.
- Only SNMP read-only may be enabled.
- Only FIPS-approved algorithms can be used for cryptographic services (such as HTTPS, L2, AES-CBC, SSH, and IKEv1/IKEv2-IPSec), which include AES, Triple-DES, SHA-1, HMAC SHA-1, and RSA signature and verification.
- TFTP can only be used to load backup and restore files. These files are: Configuration files (system setup configuration), the WMS database (radio network configuration), and log files. (FTP and TFTP over IPsec can be used to transfer configuration files.)
- The controller logs must be monitored. If a strange activity is found, the Crypto Officer should take the controller off line and investigate.
- The ‘no’ command in the CLI can be used to zeroize IKE, IPsec and CA CSPs. Please See CLI guide for details.

8.2 User Guidance

The User accesses the controller VPN functionality as an IPsec client. The user can also access the controller 802.11i functionality as an 802.11 client. Although outside the boundary of the controller, the User should be directed to be careful not to provide authentication information and session keys to others parties.

8.3 Setup and Configuration

The Aruba VMC-TACT Controllers meet FIPS 140-2 Level 1 requirements. The sections below detail the FIPS-approved mode of operation.

8.4 Setting Up Your Virtual Controller

To set up your controller:

1. Make sure that the module is not connected to any device on your network.
2. Boot up the module.
3. Connect your PC or workstation to a physical port mapped to the module interface.

For further details, see the ArubaOS 6.4 Quick Start Guide.

8.5 Enabling FIPS Mode

- The Module operates in the FIPS Approved mode. No action needs to be taken to put the module in FIPS Approved mode. Taking the module out of FIPS Approved mode is not an Aruba approved mode of operation and is not supported.

To verify FIPS mode, issue the command "`show fips`," This should always return the affirmative result.