PERMIT/Gate Crypto Module Functional Specification for the TSCMP30 v1.00

Document Number: FIPSFS00

Author(s): Andrew Robison

Version: 1.6

Last Revised: 99.04.19

Total Pages: 10

TABLE OF CONTENTS

INTRODUCTION

About this Document

The Crypto Module (CM) is a set of code running on the PERMIT/Gate 4520 and 2520 designed to do DES encryption in a manner that meets the FIPS 140-1 standard (level 2). This document describes how the module fits into the rest of the software on the gateway, what duties it is responsible for performing, and what interactions can take place between this module and others.

About the Hardware

Timestep sells two products that the crypto module will operate on, the PERMIT/Gate 4520 and the PERMIT/Gate 2520. The hardware platform is identical in all respects except for a small number of components (the 4520 has a faster CPU, running at a higher bus frequency). See the hardware information binder for more details.

Module Purpose

The CM is intended to be used by the IPSEC layer to encrypt and decrypt traffic passing though the gate. When using this CM on a gateway, a different build of the software will be used. This software will function for the most part like the software on a normal gateway except that a few new interfaces must be used; no functionality has been removed. These changes will ensure that the module can operate in a manner secure enough to meet the FIPS 140-1 level 2 standard.

How it will effect the Gateway

The "new" crypto module is still responsible for the DES cryptography done on the gateway. The IPSEC tasks and services will continue use this module as if nothing was changed.

All of the changes surround how the gateway manages static SA's. The "old" way of specifying static SA's involved a file containing plaintext keys; the new CM layers this old system to so that the keys can be encrypted. Instead of specifying actual keys in the static SA file, key "numbers" will be entered; these numbers will index a table of keys that will be entered separately into a utility that will encrypt them into a signed file that will be uploaded to the gateway separately. This allows all of the existing mechanisms in place for managing static SA's to continue oblivious to the fact that they are no longer using "real" keys.

The CM itself will handle the translation of these key numbers to real keys. Once the key file is uploaded to the gateway the CM will verify the signature on the file and then load the file into memory. Whenever it is asked to perform a cryptographic operation, the CM will decrypt the required key and use it to perform the action requested.

One final detail to cover is how the key file is authenticated and decrypted. These operations make use of two new "master" keys: the "Master Authentication Key" and the "Master Decryption Key". These keys are DES keys and are entered into the gateway via the console. They are stored in the zeroizable NVRAM so that in the event of an enclosure compromise they will be cleared and the stored key table will be illegible. The key file itself is authenticated by a 32-bit DES MAC generated with the Master Authentication Key. Before loading a key file the module will verify this signature and in the event of failure, it will reject the file and return the module to the un-initialized state.

FUNCTIONS

Master Key Management

Two master keys will be used to safe-guard the key table, as described in the next section. These keys will be kept in the zeroizable RTC NVRAM and in the event of a detected enclosure compromise, this key will be erased. Functions will be provided for loading and clearing these master keys.

These keys are the only keys that are stored in plain-text (other keys may pass into the module in plain-text but will be encrypted immediately. The master keys will never be stored in RAM; once inside the crypto-module, they will exist only in the RTC and the VMS 110.

In order to change one of the master keys, the existing authentication master key must be entered by the crypto-officer. If the gate has been zeroized then the master authentication key is 00000000000000.

Key Table Management

All keys to be used for DES encryption and decryption will be stored in a table. This table will be stored in RAM, and each key will be encrypted with ECB DES with the Master Decryption Key.

Some of the keys in the table will be keys for static SA's; these keys will be loaded at startup in encrypted form (a function will be provided to clear and reload the table). Outside the module it is expected that the key table will be stored in encrypted form on some non-volatile storage medium.

When the key is loaded, it will be authenticated by the second master key, the Master Authentication Key. The file's 32 bit CBC DES MAC will calculated with this key.

Some of the keys in the table will be keys arrived at through ISAKMP. These keys, too, can be stored in the table in encrypted form (so that the internal workings of the module can be homogeneous). This will allow the crypto-module to also perform the DES cryptography for traffic not using Static SA's.

Encryption / Decryption

This module will be used to do DES and 3DES encryption of network traffic passing through the gateway. The IPSEC module will make calls to this crypto module passing it data to encode or decode. It will also supply all initialization vectors and will specify which key is to be used by the key's reference number (a form of index into the key table). The module will use the decryption master key to decrypt the specified key from the table so that it can perform the requested operation.

Random Number Generation

The module will also provide a random number generating function. The module doesn't use the RNG for generation of initialization vectors - it would be up to the application to do this. This is outside of the cryptographic boundary and is not a validated interface nor function.

INTERFACES

Initialization

This module will provide a function call that will initialize the module.

IPSEC

The IPSEC module will interface with this module via function calls for encrypting and decrypting blocks of data with DES or 3DES. The functions will specify a key reference to be used with the key table and a block of data to work with.

ISAKMP

Functions will be provided for loading and clearing plain-text keys into the key table.

Crypto-officer / User Interface

Functions will be provided for loading and clearing the master keys and the Static Key Table. These functions will be called as part of the user-interface.

The crypto-officer has two interfaces: setting the master authentication key and setting the master decryption key. To set the master keys, the crypto-officer must enter the existing master authentication key as validation of identity.

ROLES

Crypto Officer

The Crypto Officer is a user authorized to set the master keys, and to load the static key table. Outside the CM, the Crypto Officer is also allowed to set up security associations so that the PERMIT/Gate has a policy for what to do with data passing through (whether to let it pass, to encrypt it, to decrypt it, or to block it).

The Crypto Officer is responsible for the initialization and configuration of the device and for selecting an appropriate mode of operation.

With respect to the CM, the Crypto Officer is allowed to:

Set the master keys (but not read the currently loaded keys).

Load a new static key table in encrypted form (but not read the currently loaded table).

User

A user is anyone who is not a crypto officer. A user can request encryption and decryption operations on data with keys from the Key Table. A user can load the key table from a key file previously signed (off-line outside the module) key file.

SECURITY POLICY

Crypto-Module Boundary

The crypto-module boundary is the physical enclosure. The significant components in the module are shown in the block diagram as shaded in yellow.

The logical boundary is the calling interface to the crypto module as defined in the include/crypto.h header file., which is based upon the C calling convention with arguments passed in the registers of the processor. For crypto-officer role actions, the interface between the user and the module occurs through the console UART interface. The final interface to the module is the hardware switches to the RTC which clear the master keys and cause the gate to reset.

Authentication Policy

The CM employs a role based authentication scheme.

The crypto officer role is established by initializing the module by setting both the authentication and encryption master keys.

For calls to set master keys, the crypto-officer will be authenticated by having him enter the master authentication key.

The user role is authenticated by the loading of the key table from a key file previously signed (off-line outside the module) key file.

Access Control Policy

Roles: Crypto Officer and User.

Services: Setting the Master Key, Loading the Key Table, performing cryptography on data, and power-on self-tests.

Security Relevant Data Items: The Master Authentication Key (plaintext), the Master Decryption Key (plaintext), the Key Table (encrypted) and authentication data.

Modes of Access: Write/Delete Master Key, Write/Delete Key Table, Read/Write Data.

Physical Security Policy

The two halves of the enclosure are fastened together using four Phillips screws. Each of the screws are covered by a frangible seal.

The module enclosure will be shut with tamper evident seals. The module enclose has a compromise detection switch that will zeroize the two master keys if the enclosure is opened.

Signs of physical security would be broken seals or missing master keys (the module being in the un-keyed state would be obvious by the module not being functioning, although a message would be printed to the console at startup).

Crypto Officer

Services that can be performed: Setting the Master Key (write only), and initializing the module (and performing power-on self-tests).

User

Services that can be performed: performing cryptography on data with keys from the Key Table (read/write), and Loading the Key Table (write only).

DOCUMENT REVISIONS

Version	Date	Person	Reason
1.0	Sept 1, 1998	G. Miner	Creation
1.1	Oct 1, 1998	G. Miner	Updated Design now requires two master keys.
1.2	Oct 15, 1998	G. Miner	Updated Added RNG function.
1.3	Oct 29, 1998	G. Miner	Updated Removed multi-mode idea.
1.4	Mar 18, 1999	A. Robison	Updates Added more description of boundary
1.5	Apr 13, 1999	A. Robison	Updates Added new crypto-officer authentication
1.6	Apr 19, 1999	A. Robison	Updates to fix missing bits for validation. New format for library.