Tumbleweed Secure Mail Security Kernel

FIPS 140-1 Non-Proprietary Security Policy

Version 1.1

Level 1 Validation

April, 2002

© Copyright 2002 Tumbleweed Communications Corp.
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

Table of Contents

1 INTRODUCTION

1.1 Purpose

This is the non-proprietary FIPS 140-1 security policy for the Tumbleweed Secure Mail Security Kernel v5.0. This Security Policy details the secure operation of the Secure Mail Security Kernel as required in Federal Information Processing Standards Publication 140-1 (FIPS 140-1) as published by the National Institute of Standards and Technology (NIST) of the United States Department of Commerce.

1.2 References

For more information on the Tumbleweed Secure Mail product, the Secure Mail Security Kernel, or Tumbleweed's entire product line, please visit www.tumbleweed.com.

For more information on FIPS 140-1 and validation process, please visit www.FIPS140-1.com.

For more information on NIST and the cryptographic module validation program, please visit csrc.nist.gov/cryptval

1.3 Document Organization

The Security Policy document is one document in complete FIPS 140-1 Submission Package. In addition to this document, the complete Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Module Software Listing
- Other supporting documentation as additional references

The remainder of this document (Section 2) outlines the functionality of the module and gives high-level details on how it meets the requirements of FIPS 140-1. This Security Policy and other Certification Submission Documentation were produced by Corsec Security, Inc. under contract to Tumbleweed Communications Corporation (Tumbleweed). With the exception of this Non-Proprietary Security Policy, the FIPS 140-1 Certification Submission Documentation is Tumbleweed-proprietary and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Tumbleweed.

2 Tumbleweed Secure Mail

The Tumbleweed Secure Mail is a suite of software designed to allow organizations to apply security beyond the firewall and secure e-mail and Web traffic. Secure Mail provides a complete set of secure e-mail solutions, including virus scanning, content control, access control, encryption, and authentication. Secure Mail integrates with existing e-mail networks and adds secure communications using S-MIME and cryptography.

Flexible administration capabilities allow Secure Mail users to secure e-mail configuration with custom policies across an entire enterprise network. Tumbleweed Secure Mail allows administrators and policy-makers to define and enforce security policies to ensure the safe and efficient use of corporate e-mail systems. Policies can apply to virus scanning, content control, access control, encryption, and digital signature policies universally across the enterprise.

A number of subcomponents of Tumbleweed Secure Mail provide specific functionality. Secure Mail Content Manager allows content filtering of e-mail messages and attachments according to set policies. E-mail is scanned for specific words or content that constitutes policy violations. Secure Mail can then block, quarantine, archive, defer delivery, send to additional recipients, or redirect to alternate recipients. Content filtering can be combined with Access controls to provide robust protection against Spam and junk mail, and can be used to block e-mail viruses, such as Melissa.

With Secure Mail Access Manager, companies can set policies that restrict e-mail from certain senders or to certain recipients. Secure Mail Virus Manager uses integrated server-based anti-virus software from Network Associates to detect and optionally clean or strip infected attachments in both incoming and outgoing messages. Secure Mail Format Manager can strip or rewrite addresses in message headers, protecting your network from potential hackers. The Secure Mail Web Filter monitors and controls submissions to Internet message boards and Web-based e-mail services, and includes countermeasures to detect malicious mobile code (Java and ActiveX applets), and control downloads of inappropriate files and software.

Secure Mail can also enact policies for secure communications by automatically re-routing sensitive e-mail traffic to Tumbleweed's solution for secure online messaging, Tumbleweed Secure Messenger. The integrated Secure Mail/Secure Messenger solution automatically redirects sensitive traffic to Tumbleweed Secure Messenger - no end user action is required to send groupware-generated messages securely over the Internet.

2.1 The Secure Mail Security Kernel

The Tumbleweed Secure Mail uses a set of cryptographic functionality called the Secure Mail Security Kernel. The Tumbleweed Secure Mail Security Kernel exposes application programming interface (API) calls to the other portions of Secure Mail. These APIs allow each sub-component of Secure Mail to access the same robust, tested security services. These services are also made available to other non-Secure Mail applications through the Secure Messenger Toolkit (SMT), which includes the Secure Mail Security Kernel as its Secure Messenger Engine.

The Cryptographic boundary for the Secure Mail Security Kernel encompasses the software contained within the MMSSMT.DLL. However, for the FIPS 140-1 evaluation, the Secure Mail Security Kernel is considered to be a multi-chip standalone cryptographic module running on a standard personal computer (PC) running the Windows NT operating system in single-user mode. The Secure Mail Security Kernel can also be run on other Windows and UNIX platforms.

2.2 Operating System Configuration

The Secure Mail Security Kernel was evaluated against all level one FIPS 140-1 requirements. The module is installed only as executable code, includes cryptographic self-integrity checks using FIPS-approved algorithms, and limits access to the executable to authorized users or processes. The Secure Mail Security Kernel code is written in C and C++ and meets all FIPS 140-1 software security requirements. The module software design corresponds to the Finite State Machine (FSM) Model described in the Tumbleweed proprietary document: "*Tumbleweed Secure Mail Security Kernel FIPS 140-1 Finite State Machine*".

The Secure Mail Security Kernel should be securely installed by an administrator of the Windows NT machine, with permissions to access the module files and directory limited to appropriate accounts. Using Windows NT to provide this type of controlled access requires the use of the NTFS file system, and not FAT or FAT32. The operating system should be configured only in single user mode. The administer should limit access to the module to only users approved to operate the module, and limit access to user data created and stored in SQL tables. Access to SQL tables should be limited to the particular user and the module itself. The Windows NT User Manager for Domains can be used to define users and groups for whom access permissions may be defined. The file and folder security properties can then be defined to specify access for the Secure Mail Security Kernel files.

2.3 The Secure Mail Security Kernel API

The Secure Mail Security Kernel provides a number of services, and supports two distinct roles. These two roles provide services to configure User information (which is referred to as the Crypto-Officer role), and to exercise routine User services (which is referred to as the User role). All of these services are accessed through Applications Programming Interface (API) function calls described in more detail in the following sections. FIPS 140-1 does not require authentication for access to services under each role at level 1; however, the Secure Mail Security Kernel provides some optional User identification capabilities.

Since the Secure Mail Security Kernel is considered a multi-chip standalone module, the physical interfaces consist of the keyboard, mouse, monitor, serial ports, network adapters, etc. However, the actual interfaces to the module are the logical inputs through the API and the Graphical User Interfaces that can be optionally called through the APIs. This includes fifty active User and Crypto-Officer function calls, and forty two inactive function calls (provided for backwards compatibility purposes). The function calls are divided below into Crypto-Officer functions and User functions.

2.3.1 User API Calls:

The User services fall into three groups of API calls. The SMT Message group provides message manipulation routines, the SMT Session Group provides functions to choose a User and set up an active User session, and the SMT Util Group provides string conversion utilities.

SMT Session Group (User Services)	
SmtSession_Start	Start an SMT session
SmtSession_End	End an SMT session
SmtSession_CertManager	Launch the certificate manager
SmtSession_PickCert	Allow selection of a certificate from a list
SmtSession_ViewCert	Display a certificate (by SMTID_CERT)
SmtSession_GetCertStatus	Gets the status of a certificate
SmtSession_SetProps	Set the session properties
SmtSession_GetProps	Get the session properties
SmtSession_DisplayProps	Display the session properties to the user
SmtSession_GetMaxPageCount	(Win32) Propsheet extension support
SmtSession_GetPages	(Win32) Propsheet extension support
SmtSession_FreePages	(Win32) Propsheet extension support
SmtSession_DetermineBestSigningCertificate	Determines the best signing certificate
SmtSession_ExportCert	Exports the signing certificate
SmtSession_GenerateProxyCertificate	Generates proxy certificate
SmtSession_GetAllUserCertificates	Gets signing and proxy certificates
SmtSession_GetCertDetails	Get certificate details
SmtSession_GetCertDisplayName	Retrieve Certificate and display name
SmtSession_GetProxyCertID	Get proxy Certificate Identification
SmtSession_Lock	Checks for invalid parameters
SmtSession_IsCertificateSuitableForProxyParent	Check that Certificate is suitable for Proxy
SmtSession_Unlock	Retrieve Session Identification
SMT Message Group (User Services)	
SmtMsg_Copy	Copy contents of one message into another.
SmtMsg_Create	Create a new message object
SmtMsg_Destroy	Destroy an existing message object
SmtMsg_SetProps	Set the message security properties
SmtMsg_GetProps	Get the message security properties
SmtMsg_SetSender	Set the sender (properties) of a message
SmtMsg_GetSender	Get the sender (properties) of a message
SmtMsg_AddRecip	Add a recipient to a message
SmtMsg_DeleteRecip	Delete a recipient from a message
SmtMsg_BeginEnumRecip	Begin enumerating the recipients of a message
SmtMsg_EnumRecip	Retrieve the next recipient of a message
SmtMsg_EndEnumRecip	End enumerating recipients of a message
SmtMsg_SetRecipProps	Set the properties for a recipient
SmtMsg_GetRecipProps	Get the properties of a recipient
SmtMsg_ClearPlainBody	Clear the plaintext body of a message
SmtMsg_ClearCipherBody	Clear the ciphertext body of a message
SmtMsg_BeginWritePlainBody	Begin writing the plaintext body of a message
SmtMsg_BeginWriteCipherBody	Begin writing the ciphertext body of a message
SmtMsg_WriteBody	Write to an open body property of a message
SmtMsg_EndWriteBody	End writing to a body property of a message
SmtMsg_BeginReadPlainBody	Begin reading the plaintext body of a message
SmtMsg_BeginReadCipherBody	Begin reading the ciphertext body of a message
SmtMsg_ReadBody	Read from an open body property of a message

SmtMsg_EndReadBody	End reading from a body property of a message
SmtMsg_EncryptAndSign	Encrypt and/or sign a message
SmtMsg_DecryptAndVerify	Decrypt and/or verify a message signature
SMT Util Group (User Services)	
SmtUtil_GetEncAlgorithmStr	Converts an encryption algorithm ID to a string
SmtUtil_GetSigAlgorithmStr	Converts a signature algorithm ID to a string

2.3.2 Crypto Officer API Calls:

The Crypto Officer calls allow the creation, deletion and modification of Users and their associated information. This includes user certificates, private keys, addresses, trusts, and other user configuration data. These may be accessed with the calls in the SMT User Group, and the optional GUIs that these functions will present. In addition, there is a set of compatibility functions which may be called but do no processing in this version of the Secure Mail Security Kernel. These functions return an error value when called in order to indicate that they are only provided for compatibility purposes.

SMT User Group (Crypto Officer Services)	
SmtUser_Add	Add a user
SmtUser_GetProps	Set the properties of a user
SmtUser_BeginEnum	Begin enumerating users
SmtUser_Enum	Retrieve the next user
SmtUser_EndEnum	End enumerating users
SMT Compatibility Group (Crypto Officer Services)	
SmtUser_Delete	Delete a user
SmtUser_SetProps	Get the properties of a user
SmtUser_DisplayProps	Display user properties
SmtUser_DisplayList	Display a list of users, returning selected user
SmtUser_DisplayListWithHelp	Display users with help, returning selected
SmtAddress_Add	Add an entry to the address book
SmtAddress_Delete	Delete an entry from the address book
SmtAddress_SetProps	Set the properties of an address book entry
SmtAddress_GetProps	Get the properties of an address book entry
SmtAddress_DisplayProps	Display address book entry properties to user
SmtAddress_BeginEnum	Begin enumerating address book entries
SmtAddress_Enum	Retrieve the next address book entry
SmtAddress_EndEnum	End enumerating address book entries
SmtAddress_DisplayRecipList	Displays the recipient list for a message object
SmtAddress_DisplayAddrBook	Displays the SMT address book
SmtMsgProp_Display	Display message security properties to user
SmtMsgProp_FreePages	(Win32) Propsheet extension support
SmtMsgProp_GetMaxPageCount	(Win32) Propsheet extension support
SmtMsgProp_GetPages	(Win32) Propsheet extension support
SmtMsg_ImportCerts	Import Session Identification Certificates
SmtMsg_Publish	Publish Session Identification Certificates
SmtSession_AddCertificateSources	Add the sources of certificates
SmtSession_ClearPassphrase	Clear the private key passphrase
SmtSession_Config	Configure Session
SmtSession_DoAbout	Display the SMT about box
SmtSession_DoHelp	Provide help for a given context
SmtSession_GenerateKey	Generate a key pair for the current user
SmtSession_GetCurrentUserAddressKey	Retrieve Current Address Key

SmtSession_GetCurrentUserSigningCertificate	Retrieve Current Signing Certificate properties
SmtSession_RegisterCallbacks	Register Session ID, UI Parameters, Flags, and Callbacks
SmtSession_Version	Retrieve Session Version
SmtUtil_CheckVersion	Performs version checking on SMT
SmtUtil_GetCertStatusStr	Converts a certificate status value to a string
SmtUtil_GetRandomData	Fills a buffer with random data
SmtUtil_GetSupportedEncAlgorithms	Retrieve supported Encryption algorithms
SmtUtil_GetSupportedSigAlgorithms	Retrieve supported Signature algorithms
SmtUtil_WipeFile	Overwrites and deletes a file

A complete description of each of these functions, including inputs and outputs is provided in the *Tumbleweed MMS SMT Security Kernel Programmers Reference.*

2.3.3 Accessing Services

The operators of the module implicitly assume the Crypto Officer or User role when they access the appropriate API functions for the module. However, before accessing functions, the operators must create a message or session, and identify themselves to the device as a particular operator. The evaluated version of the module expects that a single operator is created and associated with a particular e-mail address. Each user has a set of properties and preferences associated with them that are used in the generation of messages. Users are selected after the module has been started and an active session created.

2.3.4 Security Relevant Data Items

There are a number of security relevant data items present within the module. The services described above manage access to or use of these data items.

Security Relevant Data Item	**Purpose**
Symmetric Keys	Used for encrypting and decrypting messages
Private Keys	Used for key exchange or digital signature
Public Keys	Used for key exchange or digital signature verification

2.4 Physical Security

The Secure Mail Security Kernel is a software module and was tested on the Windows NT operating system as configured in single-user mode. The Secure Mail Security Kernel module can also be operated upon other Windows-compatible platforms, but was not tested upon these platforms. The module was tested against FIPS 140-1 requirements on a standard Intel platform Personal Computer (PC) that meets all FIPS 140-1 level 1 physical requirements. This includes providing production grade equipment, standard passivation of components, and FCC certification against electromagnetic interference and compatibility.

2.5 Cryptographic Algorithms and Keys

The Secure Mail Security Kernel provides a number of cryptographic algorithms, including both FIPS-approved algorithms and non-FIPS approved algorithms. All FIPS-approved algorithms have been separately validated as meeting the applicable FIPS standards. To operate the module in a FIPS approved manner, the non-FIPS-approved algorithms should not be used. The following algorithms are provided:

FIPS-approved algorithms:

- Data Encryption Standard (DES) (FIPS PUB 46-3) using Cipher Block Chaining (CBC) mode as defined in NIST Special Publication 800-17
- Triple DES (FIPS PUB 46-3) using CBC mode as defined in NIST Special Publication 800-20
- Digital Signature Standard (DSS) Digital Signature Algorithm (DSA) (FIPS 186-2)
- Secure Hashing Algorithm (SHA-1) (FIPS 180-1)
- RSA Digital Signatures (RSA) (FIPS PUB 186-2) using Public Key Cryptographic Standard (PKCS) #1 formats. This includes signature generation and verification using SHA-1 hashing and RSA encryption.

Other algorithms:

- MD5 with RSA Encryption Digital Signatures (Using PKCS#1)
- MD2 with RSA Encryption Digital Signatures (Using PKCS#1)
- MD5 Message Digests
- MD2 Message Digests
- Key Exchange using RSA Public/Private Encryption/Decryption
- RC2 CBC (40, 64, 128, 255 bit) Symmetric Encryption/Decryption
- RC5 (40, 64, 128 bit) Symmetric Encryption/Decryption

Symmetric keys are dynamically created by the module for encryption of messages, and are zeroized from memory when encryption or decryption operations are complete. Symmetric keys are encrypted using public key technology as defined in S/MIME specifications for secure transmission to recipients.

There are public and private keys managed in the module and associated with each user. Public keys, included in public key certificates, and private keys are stored in SQL tables. All of these keys can be replaced or deleted at the user's discretion.

2.6 Self-Tests

In order to prevent any secure data being released, it is important to test the cryptographic components of a security module to insure all components are functioning correctly. The Secure Mail Security Kernel consists of software modules running on a Windows NT operating system in single-user mode and includes an array of self-tests which are run during startup and periodically during operations. The self-tests run at power-up include cryptographic known answer test (KAT) on the FIPS-approved algorithms (DES, 3DES, SHA-1) and on all other algorithms supported by the module. Also performed at startup are software integrity tests using

a DSA digital signature of the module and several other critical functions tests. Other tests are run periodically or conditionally such the continuous random number generator test and a pairwise consistency check is run when an RSA or DSA key pair is generated.

3 FIPS Mode

The Tumbleweed Secure Mail Security Kernel requires no special configuration settings to be operated in a FIPS 140-2 approved mode of operation. There are however installation and usage requirements to operate the module properly.

1. The Windows NT operating system should be configured in single-user mode.
2. Only those algorithms listed in section 2.5 as being FIPS-approved algorithms should be utilized by the user. All other algorithms are not FIPS approved and may not be used.