Security Policy

MOTOROLA MESSAGING SERVER SERVER AND MOTOROLA MYMAIL™ DESKTOP PLUS

ENCRYPTION DLL CRYPTOGRAPHIC MODULE

MOTOROLA
intelligence everywhere™

REV 1.3, 10/2002

CONTENTS

MODULE OVERVIEW

The cryptographic boundary for Motorola Messaging Server (MMS) and *Motorola MyMail*™ Desktop Plus (MDP) is defined as the Encryption DLL program module. The program module is running on a Personal Computer running a WIN32 platform. This module is a software component installed by the MMS or MDP installation program.

MMS is a program application server intended for enterprise use. It allows applications on wireless devices to securely access enterprise data. A prime example of such an application is *Motorola MyMail*. The *Motorola MyMail* application on the wireless device communicates with the *Motorola MyMail* plug-in on the MMS forming an end-to-end link giving the subscriber full secure wireless access to his enterprise e-mail.

For those users where an enterprise solution would not be appropriate, MDP may be run on the user's workstation. This provides similar capabilities such as the ability to run *Motorola MyMail*, giving the user full secure wireless access to his enterprise e-mail.

SCOPE OF DOCUMENT

This document outlines the security policy for the Encryption DLL used in the Motorola Messaging Server (MMS) and *Motorola MyMail* Desktop Plus (MDP). MMS and MDP are solutions designed to enable secure client-server applications. The security policy addresses all of the applicable requirements of FIPS 140-1, includes an overview of the cryptographic module, and lists roles and services of the module and how they are related, the different types of security relevant data items (keys, key components), capabilities and protections.

TERMS AND DEFINITIONS

Term	Definition
ANSI X9.31	Standard of “Digital Signature Using Reversible Public Key Cryptography for the Financial Services Industry”
DAC	Data authentication code
DES	Data encryption standard
DESMAC	A type of integrity-checking (checksum) based on DES
DLL	Dynamic link libraries
MDP	*Motorola MyMail* Desktop Plus
MMS	Motorola Messaging Server
PRNG	Pseudo-random number generator
RC4	A stream cipher not approved under FIPS
SHA-1	Secure hash algorithm
SRDI	Security relevant data items
Subscriber	User, application
TDES	Triple-DES, a block cipher approved under FIPS

SECURITY LEVEL

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-1. The module embodiment is a multi-chip standalone module.

Security Requirements Section	Level
Cryptographic Module	1
Module Interfaces	1
Roles and Services	1
Finite State Machine	1
Physical Security	1
Software Security	3
Operating System Security	1
Key Management	3
Cryptographic Algorithms	1
EMI/EMC	3
Self Test	1

ROLES AND SERVICES

The cryptographic module supports the following roles:

Application Role (User Role): Other MMS/MDP program modules assume the application role when making calls into the Encryption DLL.

Crypto-Officer: This consists of the MMS or MDP install programs that are used to install and if necessary, uninstall the MMS application.

The Encryption DLL provides the following indirect services to the User Role:

- **FMSInit:** An initialization function that processes all power-up self-tests and initializes the log file.
- **FMSTerm:** A function that stops the logging before unloading the DLL.
- **DecryptPreProc:** A function that looks at each incoming message, determines if it is encrypted, and decrypts it using the TDES algorithm.
- **DecryptAction:** A function that performs a custom action to process errors that are identified from the key exchange protocol message within the DecryptPreProc service. If an error has occurred, this function will handle the event.
- **EncryptPostProc:** A function that encrypts a block of data using the TDES algorithm.

The Encryption DLL provides the following direct services to the User Role:

- **FMSInitEncryption:** An initialization function that processes all power-up self-tests and initializes the log file.
- **FMSTermEncryption:** A function that stops the logging before unloading the DLL.
- **EncryptFSEncode:** A function that provides access to the TDES encryption algorithm.
- **EncryptFSDecode:** A function that provides access to the TDES encryption algorithm.
- **EncryptFSGetAppString:** A function that parses the data to determine if the message is an encrypted data packet, and if so, retrieves the clear text application string available in the data header.
- **EncryptFSParseKeyMessage:** A function that takes an encrypted key from the key exchange protocol message, decrypts it, potentially verifies the sender using a password parameter, and encrypts the data before storing it in the database.
- **EncryptFSBuildKeyMessage:** A function that performs a look up within the database for the given subscriber. If a subscriber key is not found a new one will be generated before building a key exchange protocol message. If a key is found, it will use the existing key before building a key exchange protocol message.
- **EncryptFree:** A function that releases memory allocated and returned by another service.
- **SelfTest:** A function that performs all self-tests on demand and provides a status to the user.

SECURITY RULES

1. The module does not support a maintenance role.
2. The module supports a Bypass state.
3. The module provides status upon completion of a function call (service) with the exception of the FMSTerm and FMSTermEncryption.
4. The chips are of production-grade quality, which include standard passivation techniques.
5. The module is implemented for use on a production-grade multi-chip general purpose personal computer as defined by FIPS.
6. The module supports the following FIPS approved cryptographic algorithms:
   - TDES
   - DESMAC
   - SHA-1
   - PRNG per ANSI X9.31

   Note: *The module also supports the RC4 algorithm; this algorithm is not used in FIPS mode.*
7. The module is entirely written using a high level language, C and C++.
8. The operating system supported by the module is the Microsoft Windows 2000 SP1 (MMS/MDP), WindowsNT 4.0 SP4-6a (MMS), WindowsNT 4.0 SP3-6a (MDP), Windows98 (MDP), Windows98 SE (MDP) and WindowsME (MDP).
9. The cryptographic software/firmware is installed only as executable code.
10. The cryptographic module is limited to a single user at a time, which is enforced by the Operating System when configured for single user mode.
11. Use of the cryptographic module is dedicated to the cryptographic process during the time the cryptographic process is in use.
12. The module does not distribute cryptographic keys in plaintext form.
13. The module provides a mechanism to ensure that keys are associated with the correct entity.
14. The module performs a continuous random number generator test.

DEFINITION OF SECURITY RELEVANT DATA ITEMS (SRDI)

The following are the cryptographic keys that are contained in the module:

- **Server Key:** This is a TDES key used to encrypt cryptographic keys.
- **Root Key:** This is a TDES key used to encrypt the Subscriber key before transport.
- **Storage Key:** This is a TDES key used to encrypt the Subscriber key before storage.
- **PRNG Key:** This is the key used during the PRNG process per ANSI X9.31.
- **Subscriber Key:** This is a TDES key used to encrypt data.
- **DAC Key:** This is a DESMAC key used to authenticate the Software/Firmware power-up self-test.

The following lists other SRDIs that are contained in the module:

- **Server Password:** This is used to authenticate a key exchange protocol message from the device.
- **Key Password:** This is used to access keys contained in the module.
- **Application ID:** This is used to associate a specific key to a specific application on the device.
- **Key Index:** This is used to allow an application to support multiple keys.