Security Policy

Ensuredmail (9/25/2000)
Revised 2/19/01

1.0 Introduction

This document defines the security rules under which this product operates. The rules are enforced by the use of software modules. The use of the software modules are mandatory and are called automatically while exercising the module.

1.1 Identification:

Name: Ensuredmail; Version: 1.0
Vendor: Ensuredmail, Inc.

1.2 Description

Ensuredmail is a cryptographic module that permits the encryption and decryption of web based mail messages and attachments, Outlook email messages and attachments and local files. The module is contained within a Java DLL. Interfaces are via Java Script and HTML pages. The module runs on a personal computer running MS Windows 95 or later with MS Internet Explorer version 4.0 (or later). The module was designed to provide security for the transfer of financial information, however it is available to anyone. The module is designed for a single user using a single operation. Multiple concurrent operations are not allowed. Batch operations are allowed for commercial users.

1.3 Basic Security Rules:

Access to (CSP) critical security parameters.

Access Matrix

Operator	Service	Role	Access to CSP
Any	Initialization	Crypto-Officer	None
Any (Sending)	Password Entry	Crypto-Officer	Password
Any (Sending)	Encryption	User	None
Any (Receiving)	Password Entry	Crypto-Officer	Password
Any (Receiving)	Decryption	User	None

1.4 Physical Security

The module is executed entirely with software running on a personal computer (PC). Physical security is limited to that provided by a commercial grade PC sold for home use. The embodiment is multi-chip standalone.

2.0 Design Concepts

The module is designed to implement a simple procedure that protects web based e-mail messages and attachments, Outlook email messages and attachments, and local files from low level security threats using only a user provided password. The module uses a triple DES encryption algorithm to encrypt the message. A session key is generated using a FIPS-approved algorithm. The session key is encrypted using a FIPS approved method (ANSI X9.17). Both the encrypted message and the encrypted key are sent to the recipient. The password is used to derive the decryption key. Ensuredmail is designed to the FIPS Pub 140-1 level 1 standard.

2.1 Cryptographic Boundary

The logical cryptographic boundary of the cryptographic module is the Java DLL. The source code that defines the cryptographic boundary is:

BlockCipher.java
Cipher.java
EmCryptoEngine.java
CryptoUtils.java
Des3Cipher.java
DesCipher.java
PRNG.java
ByteArrayObject.java
ProgressingFrame.java
progressingBar.java
gifPanel.java

The physical cryptographic boundary is the computer case.

2.2 Interfaces

Logical interfaces to the module are Java Script and HTML pages which ask the crypto-officer or user to input information and/or take a particular action. E-mail information, including attachments, as well as local files are captured automatically from the host (such as hotmail, yahoo, etc.) screen and placed in the

HTML window by emIEEngine.dll for processing by the module. All logical interfaces are implemented in software code. The decision to encrypt or decrypt is automatically made for the user by the software and is determined by whether a compose or read window is open in the web based e-mail webpage, or Outlook email client program. In addition, Ensuredmail protected files are identified by the .emx file extension which tells the program what operation is required (encryption or decryption) for local files. This limits the module to one and only one operation (state) at a time as both encrypt and decrypt can not be entered simultaneously.

Data input interface: The data input interface consists of a screen of HTML code that permit input of authentication data (password). The plaintext or ciphertext data is input automatically to the module from the web based HTML page, Outlook mail client window, or local files present on the users display by use of java script.

Data output interface: The data output interface consists of a screen of HTML code that permit output of plaintext or ciphertext. The plaintext or ciphertext data is input automatically to the module from the web based HTML page or Outlook client present on the users display by use of java script.

Control input interface: The control input interface is in the form of HTML code that permits the selection of a button to command the module to either encrypt or decrypt the message and /or attachments.

Status output interface: The status output interface is in the form of HTML code that permits the user to observe error status messages or status messages while the program is working. The messages are automatically presented when the proper conditions exist. User action is either not required or that of clicking the "OK" button after the status message is read. The actual status messages are outside the cryptographic boundary. Code that invokes the status display is contained within the cryptographic boundary or outside the cryptographic boundary.

Power and Maintenance Interface: No power or maintenance interface exists.

2.3 Errors

Several error conditions exist as detailed below. Whenever an error is encountered the output is inhibited.

Error conditions:
Initialize self-test: Upon initialization the module performs a software integrity check (using 3DES-MAC). If the software test fails the module will not initialize and no operations can be performed.

Incorrect Password entry (encryption): The module checks that password entry is correct by requesting the user to enter the password twice. If the second entry is not exactly identical to the first the user is notified of an incorrect password.

Incorrect password entry (decryption): The module checks that the password entry is correct. If not, two more tries are allowed. Upon completion of the third incorrect password attempt, the module terminates the session.

Unsupported web based e-mail site: The module notifies the user that the web site is not supported.

Invalid Format error: When Ensuredmail is invoked with an invalid file format, this error will display.

Anti-Forwarding Error: If a user tries to open an email that was forwarded to them and the original sender requested anti-forwarding, the anti-forward message will display and not allow the user to open the message even if they have the password.

Password less than 6 digits error.
No message or attachments to be encrypted error.
Send to error in Outlook CW version
Attachments or files already chosen error.
No attachments or files to be removed error.

2.4 Processors

The code is executed on the local processor. The executable code resides on the local users hard disk drive.

3.0 Roles

The module supports only two roles; that of crypto-officer and user. The same individual performs both roles. Since key is generated internally, the role of the crypto-officer is limited to that of initializing the module and entering a password. The password is hashed using the SHA-1 algorithm and the output is input into a FIPS approved KEK algorithm.. All other operations are assigned to the user.

Roles and Services

Role	Function	Authorized Services
Crypto-officer	Initialization	Show Status
Crypto-officer	Key management	Password entry
User	Cryptographic operations	Encryption/Decryption
User	Self test (automatic)	Software integrity check
User	Self test (automatic)	PRNG conditional test

4.0 Services

Both encryption and decryption services are provided by the module. Key is internally generated. The encrypted key is sent with the message to permit the recipient to decrypt the message. Knowledge of the password is necessary to decrypt the key.

Ensuredmail Service Matrix

Service	Input	Output	Role
Encryption	Plaintext Mail &/or Attachment, or local file(s)	Enciphered Mail &/or Attachment, or local file(s)	User
Decryption	Enciphered Mail &/or Attachment, or enciphered local file(s)	Plaintext Mail &/or Attachment, or local file(s)	User
Show Status	User Interaction with interface screens	Module Status Message	User
Password Entry	User Assigned Password	Session Keys	Crypto-Officer
Show Status	Password (incorrect)	Incorrect Password Message	Crypto-officer

5.0 Access Control

Access to the decrypted message is available only to a recipient having knowledge of the password used in the original encryption. Only when the caller software is authorized, the Java DLL can be utilized. By this way, any unauthorized software/hacker/developer cannot access the Ensuredmail cryptographic module.

Identification and Authentication Table

Role	Type of Authentication	Type of Identification	Strength

User	None	None	N/A
Crypto-officer	None	Password	Weak[1]

The operator or user has no access to the critical security parameters (key generation or plaintext key) while performing a service. By design the recipient has access to the originators password used in the encryption process. The password must be exchanged via other channels such as spoken word, postal mail, etc. Access to the module is restricted to one user at a time by the software.

The Java DLL will only be loaded if the caller software has the cryptographic module's password (only known to Ensuredmail, Inc.). By this way, any other software/hacker/developer (other than Ensuredmail developers) cannot access the Ensuredmail cryptographic module's API.

6.0 Key Management

The module produces a 128 bit secret key generated by a FIPS-approved random number generator. This secret key is the session key and is used to encrypt the message. A hash of the password is used to generate a KEK through a FIPS approved process (X9.17 appendix C). The KEK is used to encrypt the session key. The encrypted session key is then bundled with the encrypted message and sent to the recipient. The recipient must be in possession of the password to decrypt the message. A hash of the password followed by the X9.17 process is used to decrypt a known pattern contained in the message. If the decryption of the known pattern is successful, then the entire message is decrypted. If not, the recipient is allowed two additional tries at a correct password before the module terminates. The password is passed between the sender and recipient via some external means that is managed under a separate security policy.

Key is generated for each session and is not stored. Key is erased from dynamic memory after each use, upon termination of the module. Key is not available to the user during module operation.

[1]The strength of the password is orders of magnitude stronger than the 4-digit PINs currently used in most banking applications, as it consists of a minimum of 6 and may consist of as many characters and/or punctuation symbols as the Windows text control can hold (relatively unlimited).

6.1 Cryptographic Algorithms

Table of Algorithms

Algorithm	Cert#	Use
ANSI X9.17	N/A	Random Number Generation
3DES	(TBD)	Encryption/Decryption
3DES-MAC	(TBD)	Software Integrity Test
SHA-1	(TBD)	Password Hash

7.0 Tests

7.1 Self Tests

7.1.1 Software/Firmware Test

The module tests the software at each use by comparing the 3DES-MAC of the software with a value previously stored. The algorithm is the 3DES-MAC.

7.1.2 Conditional Random Number Generator Test

The module uses a failure to constant value random number generator test. Two random numbers are produced each time the module is invoked. The numbers are compared to determine if they are the same. If they are different the test is passed. If not the module terminates.

7.1.3 Triple DES Known Answer Test

The module uses the triple DES algorithm for encryption and decryption. A known answer test is run each time the algorithm is invoked to ensure the algorithm software is functioning properly.

7.2 Manual Key Entry Test

The module uses a manual key test to validate the correct entry of the password (encrypt side). The operator is asked to enter the password twice. The values are compared and if identical the password is accepted. If not, the operator is asked to try again.