# *VME Crypto Engine Security Policy*

Document *Version 1.4*

## *Meganet Corporation*

12/1/07

## TABLE OF CONTENTS

**1. MODULE OVERVIEW ..... 3**

**2. SECURITY LEVEL ..... 4**

**3. MODES OF OPERATION ..... 4**

**4. PORTS AND INTERFACES ..... 5**

**5. ROLES AND SERVICES ..... 5**

**6. CRITICAL SECURITY PARAMETERS ..... 9**

**7. SECURITY RULES ..... 10**

**8. PHYSICAL SECURITY ..... 12**

**9. OPERATIONAL ENVIRONMENT ..... 12**

**10. MITIGATION OF OTHER ATTACKS POLICY ..... 12**

**11. DEFINITIONS AND ACRONYMS ..... 13**

# 1. Module Overview

The VME Crypto Engine (SW Version 4.4.0.0/M145) is a software implementation that runs on a general purpose PC. The embodiment of the module, according to the definition found in Section 4.5 of the FIPS 140-2, is a multi-chip standalone. The primary purpose for this device is to provide cryptographic services for applications running on the system on which it is installed. The VME Crypto Engine includes the following software components in the logical boundary:

VME04B32.DLL

MSCAPI, version 5.1.2600.1029, FIPS 140-1 Certificate #238

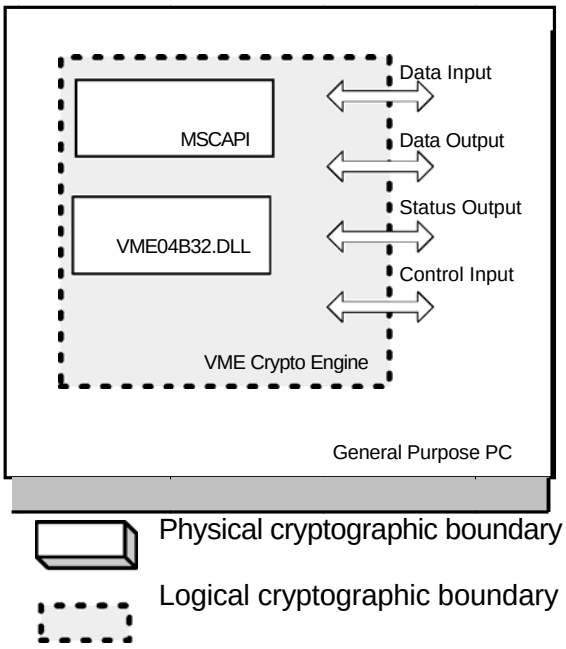


**Figure 1 – Image of the Cryptographic Module**

The VME Crypto Engine was operationally tested on the following platforms:

Intel PentiumM with the following Operating Systems (the following platforms must utilize MSCAPI versions 5.1.2518.0 or 5.1.2600.1029 to be FIPS 140-2 compliant)

:

- Windows 98, Second Edition
- Windows ME Build 4.90.3000

- Windows NT 4.0 Workstation SP 6
- Windows NT 4.0 Server SP 6
- Windows 2000 Professional SP4
- Windows 2000 Server SP 4
- Windows 2000 Advanced Server SP 4
- Windows XP Home Edition SP 1
- Windows XP Professional SP 1
- Windows Server 2003 Enterprise Edition

The platforms above were officially tested and certified as of the date of the issuance of the FIPS 140-2 certificate on 1/14/2005 by Infogard Laboratories.

The VME Crypto Engine was operationally tested and certified on the following additional platforms by Meganet Corporation as of 12/1/07:

- Windows XP Home Edition SP 2
- Windows XP Professional SP 2
- Windows XP x64
- Windows XP Itanium
- Windows Server 2003 Professional Edition SP2
- Windows Server 2003 Enterprise Edition SP2
- Windows Server 2003 x64
- Windows Server 2003 Itanium
- Windows Vista Ultimate
- Windows Vista Ultimate x64
- Windows Server 2008 RC0
- Windows Server 2008 RC0 x64
- Windows Server 2008 RC0 Itanium

# 2. Security Level

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2.

**Table 1 - Module Security Level Specification**

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Module Ports and Interfaces | 1 |
| Roles, Services and Authentication | 2 |
| Finite State Model | 1 |
| Physical Security | N/A |
| Operational Environment | 1 |
| Cryptographic Key Management | 1 |
| EMI/EMC | 3 |
| Self-Tests | 1 |
| Design Assurance | 1 |
| Mitigation of Other Attacks | N/A |

# 3. Modes of Operation

### *Approved mode of operation*

The module only operates in FIPS mode, and supports the following FIPS Approved algorithms:

- 3DES-ECB
- 3DES-CBC
- AES-ECB (256-bit)
- AES-CBC (256-bit)
- SHA-1 for hashing
- RSA with 1024-bit keys for digital signature generation and verification
- RSA with 1024-bit keys, PKCS#1 for key wrapping (vendor affirmed).
- FIPS 186-2, Appendix 3.1 with the SHA-1 based 'G' function for random number generation.

The module supports the following Non-Approved FIPS algorithms:

- VME: VME is a Proprietary Encryption Algorithm. This algorithm is not used to fulfill any of the FIPS 140-2 requirements.

# 4. Ports and Interfaces

The cryptographic module provides the following logical interfaces:

- Data input
- Data output
- Control input
- Status output
- Power input

The VME Crypto Engine is a software implementation that functions on a general purpose PC. The logical interfaces are provided by the module's API calls. The physical ports are those found on a general purpose PC.

# 5. Roles and Services

### *Authentication*

The VME Crypto Engine supports role-based authentication for authorized roles. The Crypto-Officer and User roles explicitly assume an authorized role by entering the correct password for that role. The application associates the User password with the username 'User', and the CO password with the Username 'Administrator'. The module takes a SHA-1 hash of the authentication information and stores it to perform a comparison when the operator attempts to assume the authorized role again. The Senior Recovery Officer (SRO) and Junior Recovery Officer (JRO) explicitly assume an authorized role by possession of the correct private RSA key for that role. The authentication information used for the Senior and Junior Recovery Officers should be kept on a removable storage device.

### *Assumption of roles*

The cryptographic module supports four distinct operator roles:

- Cryptographic Officer Role
- User Role
- Junior Recovery Officer
- Senior Recovery Officer

**Table 2 - Roles and Required Identification and Authentication**

| Role | Type of Authentication | Authentication Data |
|---|---|---|
| User | Role-based operator authentication | Password |
| Cryptographic Officer | Role-based operator authentication | Password |
| Junior Recovery Officer | Role-based operator authentication | Possession of private RSA key |
| Senior Recovery Officer | Role-based operator authentication | Possession of private RSA key |

***Strength of Authentication***

For authorized roles to authenticate to the module using role-based authentication, the module imposes a minimum of 6 password characters, and estimates a minimum of 64 possible alphanumeric characters. The module allows for 3 authentication attempts before terminating, and reinitializing.

For the Crypto-Officer and the User roles, the associated false acceptance or random access rate is less than one in 1,000,000 as 6^64 is greater than 1,000,000.

The associated probability of a successful random attempt during a one-minute period is less than one in 100,000 as each termination cycle takes approximately 5 seconds, therefore only 20 attempts of 3 times each, a total of 60 authentication attempts per minute are possible.

**Table 3 – Strengths of Authentication Mechanisms**

| Authentication Mechanism | Strength of Mechanism |
|---|---|
| Crypto-Officer, User: Role-based authentication using a password. | The probability that a random attempt will succeed or a false acceptance will occur is 1/6^64 which is smaller than 1/1,000,000.<br><br>The probability of a random attempt successfully authenticating to the module |

| | |
|---|---|
| | within one minute is also 1/6^64 which is less than 1/100,000. |
| SRO, JRO: Role-based authentication using a private 1024-bit RSA key. | The probability that a random attempt will succeed or a false acceptance will occur is 1/2 ^ 80 which is smaller than 1/1,000,000.<br><br>The probability of a random attempt successfully authenticating to the module within one minute is also 1/2^80 which is less than 1/100,000. |

***The following services are available to each role:***

**Crypto Officer: the Crypto Officer is in charge of maintaining the application functionality and visual looks.**

- Select Recipient's Public Key: This service presents the operator with a list of valid recipients with RSA Public Keys, of which the operator may select one. Once selected, this recipient will remain selected for the remainder of the session or until the operator uses this service to select a different one.

- Encrypt Data: The Crypto-Officer has the ability to encrypt files and messages using TDES or AES.

- Sign Data: The Crypto-Officer can sign his messages with RSA using any x509 compatible certificate.

- Decrypt Data: The Crypto-Officer has the ability to decrypt files and messages using TDES or AES.

- Verify Sender's Signature: The user has the ability to verify the RSA signature of the sender as long as it was signed with an x509 compliant certificate.

- Generate Symmetric Encryption Keys: The Crypto-Officer has the ability to generate symmetric encryption keys using the AES and TDES algorithms.

- Load symmetric encryption keys: The Crypto-Officer has the ability to load symmetric TDES or AES keys.

- Delete symmetric encryption keys: The Crypto-Officer has the ability to zeroize symmetric TDES or AES encryption keys.

- Add other operator's public keys to repository: The Crypto-Officer can add additional RSA public keys to their repository.

- Show Status: Obtain the current status of the module.
- Perform Self-tests: The module will initiate the required power-up self-tests.
- Log-out: The Crypto-Officer must log out at the end of the session. Once logged out, the crypto officer will have to login again to use the system again.

**User: the User are able to encrypt and decrypt, data, and perform other application features.**

- Select Recipient's Public Key
- Encrypt Data
- Sign Data
- Decrypt Data
- Verify Sender's Signature
- Add Other Operators Public Keys to Repository
- Show Status
- Perform Self-tests
- Log-out:

**Junior Recovery Officer: the Recovery Officers are able to recover authentication information if needed. Both Officers are needed to recover the information.**

- Recover Authentication information: The authentication information is encrypted with a 256-bit AES key, which is then encrypted with two 1024-bit RSA keys, one from the Junior Recovery Officer and one from the Senior Recovery Officer. Either Recovery Officer has the ability to extract the information using the appropriate Private RSA Key of both the Senior and Junior Recovery Officers.
- Log Out

**Senior Recovery Officer:**

- Recover Authentication information
- Log Out

# 6. Critical Security Parameters

### *Definition of Critical Security Parameters (CSPs)*

The following are CSPs contained in the module:

- AES Session Key – 256-bit key used for encryption of data.
- AES Authentication Key – 256-bit key used to encrypt the authentication data.
- TDES Session Key – 3 Key used for encryption of data.
- RSA Private Wrapping Key – 1024-bit key used for key wrapping.
- RSA Private Signing Key – 1024-bit key used for signing.
- RSA Private Senior Recovery Key – 1024-bit key used for encryption of AES Authentication Key.
- RSA Private Junior Recovery Key – 1024-bit key used for encryption of AES Authentication Key.
- DRNG state

### *Definition of Public Keys:*

The following are the public keys contained in the module:

- RSA Public Wrapping key – 1024-bit key used for key wrapping.
- RSA Public Signing key – 1024-bit key used for signing.
- RSA Public Senior Recovery key – 1024-bit key used for encryption of AES Authentication Key.
- RSA Public Junior Recovery key – 1024-bit key used for encryption of AES Authentication Key..

### *Definition of CSPs Modes of Access*

Table 6 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as follows:

- Generate: Generate TDES, AES, or RSA keys
- Use: Use a TDES, AES, or RSA key for encryption, decryption, signing, verification, key wrapping.
- Update: Update a TDES, AES or RSA key.
- Zeroize: Zeroize a CSP.

**Table 4 – CSP Access Rights within Roles & Services**

| Role | | | | Service | CSP Mode of Access |
|---|---|---|---|---|---|
| **C.O.** | **User** | **S.R.O.** | **J.R.O.** | | |
| X | X | | | Select Recipient's Public Key | |
| X | X | | | Encrypt Data | Use – AES Session key<br>Use – TDES Session key |
| X | X | | | Decrypt Data | Use – AES Session key<br>Use – TDES Session key |
| X | X | | | Sign Data | Use – RSA Private Signing key |
| X | X | | | Verify Sender's Signature | Use – RSA Private Signing key |
| X | | | | Generate symmetric encryption keys | Generate – AES Session key<br>Generate – TDES Session key<br>Use and Update - DRNG State |
| X | | | | Update symmetric encryption keys | Update – AES Session key<br>Update – TDES Session key |
| X | | | | Delete symmetric encryption keys | Zeroize – AES Session key<br>Zeroize – TDES Session key |
| X | X | | | Add other users public keys to repository | |
| X | X | | | Show Status | |
| X | X | | | Perform Self-tests | Use – SHA-1 hash to verify software integrity |
| X | X | X | X | Log-out | |
| | | X | X | Recover Key | Use – AES Session key<br>Use – TDES Session key<br>Use – RSA Recovery Key |

Note: Crypto-Officer (CO), Senior Recovery Officer (SRO), Junior Recovery Officer (JRO)

# 7. Security Rules

This section documents the security rules enforced by the cryptographic module to implement the security requirements of a FIPS 140-2 Level 1 module.

1. The cryptographic module shall provide four distinct operator roles. The authorized roles are the User, the Crypto-Officer, the Junior Recovery Officer, and the Senior Recovery Officer roles.

2. The authorized operators will authenticate to the module using role-based authentication. The User and Crypto Officer roles enter a password to explicitly authenticate to the module. The Junior and Senior Recovery Officers explicitly authenticate to the module with knowledge of the correct Private RSA Key.
3. The cryptographic module gains security functionality from the Microsoft Enhanced Cryptographic Provider (MS CAPI RSAENH) module, FIPS 140-1 Certificate #238. The module utilizes MS CAPI's implementations of RSA for key wrapping, signing, and verification, SHA-1 hashing, and key generation based on the FIPS 186-2 Appendix 3.1 with the SHA-based 'G' function.
4. The cryptographic module shall encrypt message traffic using the AES and TDES algorithms.
5. The cryptographic module shall perform the following self-tests:

   A. Power up Self-Tests:

      1. Cryptographic algorithm tests:
         a. TDES ECB and CBC encrypt/decrypt KATs
         b. AES 256 ECB and CBC encrypt/decrypt KATs
         c. DRNG KAT
         d. SHA-1 KAT
         e. RSA test
      2. SHA-1 hash and RSA Signature Verification Software Integrity Test

   B. Conditional Self-Tests:

      1. Continuous Random Number Generator test – performed on DRNG
      2. RSA pair wise consistency test – performed on RSA key pairs generated
      3. Module installation authentication test – 1024-bit RSA signature verification when the module is initially installed.

6. At any time the cryptographic module is in an idle state, the operator shall be capable of commanding the module to perform the power-up self-test.
7. Prior to each use, the DRNG shall be tested using the conditional test specified in FIPS 140-2 §4.9.2.
8. The data output interface shall be inhibited during key generation, self-tests, zeroization, and error states.
9. Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
10. The module shall not support concurrent operators.

11. The AES and TDES Session Keys are output by the module wrapped with a 1024-bit RSA key. The 1024-bit RSA keys used to wrap both the 256-bit AES and TDES Session Keys have a cryptographic strength of approximately 2^80; therefore, the 256-bit AES keys, and 168-bit TDES keys offer a cryptographic strength of approximately 2^80.
12. The RSA Private Senior and Junior Recovery keys are manually established and entered into the module in plaintext.
13. When the module has not been placed in an authorized role, an operator will not have access to any security functionality.
14. The AES and TDES keys stored in plaintext within the module are zeroized when the operator executes the “Delete symmetric keys” service. The RSA Private Keys are zeroized when the MSCAPI function “CryptAcquireContext” with the “CRYPT_DELETE_KEYSET” flag is called by the operator.

# 8. Physical Security

Section 5 of the FIPS 140-2 standard does not apply as this is a software module that operates on a general purpose PC.

# 9. Operational Environment

Section 6 of the FIPS 140-2 standard applies to this module when configured in single operator mode of use.

# 10. Mitigation of Other Attacks Policy

The module has not been designed to mitigate against any specific attacks.

# 11. Definitions and Acronyms

| | |
|---|---|
| AES | Advanced Encryption Standard |
| TDES | Triple Data Encryption Standard |
| SHA-1 | Secure Hash Algorithm |
| RSA | Rivest, Shamir and Adelman. Asymmetric encryption algorithm |
| MS-CAPI | Microsoft CryptoAPI. Used for additional cryptographic functionality. |
| VME | Virtual Matrix Encryption |
| FIPS | Federal Information Processing Standards |
| PC | Personal Computer |
| DLL | Dynamic Link Library |
| ECB | Electronic Code Book |
| CBC | Cipher Block Chaining |
| EMI | Electro Magnetic Interference |
| EMC | Electro Magnetic Compatibility |
| PKCS #1 | Public Key Cryptography System |
| DRNG | Deterministic Random Number Generator |
| API | Application Programmer Interface |
| CO | Crypto Officer |
| SRO | Senior Recovery Officer |
| JRO | Junior Recovery Officer |
| CSP | Critical Security Parameter |