iKey 2032 Security Policy

iKey 2032 Authentication Token

Original: January 26, 2001
Updated: November 20, 2006

SafeNet, Inc.
4690 Millennium Drive
Belcamp, MD 21017

Table of Contents **Page**

iKey 2032 Security Policy

1.0 Abstract

The SafeNet iKey 2032 RoHS compliant USB-based security token (Hardware Version 909-25001, Firmware Version 0.6) contains the SafeNet Model 330 Smartcard chip. This chip performs all the cryptographic logic functions for the iKey 2032 security token. The SafeNet Model 330 Smartcard, containing the same chip, was awarded FIPS (Federal Information Processing Standard) 140-1 certification, Security Level 2, in May of 2000 (reference certificate #94). The Smartcard chip contains an eight-bit microcontroller, a math accelerator for public key cryptographic functions, a hardware accelerator for Triple DES symmetric encryption, a ROM for containing the token operating system, a RAM for I/O registers, and a programmable memory.

PIN or pass phrase authentication provides role-based access control and protects the cryptographic functions of the token, such as: public key pair generation (RSA and DSS), digital signature generation (RSA and DSS), unwrapping of session encryption keys, user authentication, symmetric encryption/decryption (DES and TDES), and on-line authentication (mutual challenge/response).

The SafeNet Crypto Card Operating System (DKCCOS v2.0) is the proprietary operating system used on the Model 330 Smartcard chip and housed within the iKey 2032 token.

2.0 SafeNet iKey 2032 Security Solution

The SafeNet iKey 2032 USB-based security solution provides cost-effective and easy-to-use control for multiple applications and network services, protects Virtual Private Networks (VPN's), and controls intranet, extranet, and Internet access.

The SafeNet iKey 2032 security token contains one processor and a few other associated integrated circuits and discrete components. The processor is the SafeNet 330 Model Smartcard chip (FIPS140-1 validation certificate No. 94). This component is responsible for performing all cryptographic logic functions within the iKey 2032 token. The Hardware Block Diagram in Figure 1 (next page) illustrates the architecture of the iKey 2032 security token.

The Smartcard chip has an eight-bit microcontroller, a math accelerator for public key cryptographic functions, a hardware accelerator for Triple DES symmetric encryption, a ROM for containing the token operating system, a RAM for I / O registers, and a programmable memory.

The hardware token is built as a printed circuit board within a plastic housing, with the USB connector at one end. This housing covers the Smartcard chip, which is mounted on the circuit board with several non-cryptographic components.

The token end-user is authorized to perform sensitive functions in the token via PIN or pass phrase authentication. These functions include digital signature generation and unwrapping of session encryption keys, i.e., those functions that require the use of the closely guarded private key of the user's public key pair. The enterprise Security Officer (SO) may also authenticate himself/herself to the token with a separate PIN or pass phrase in order to perform sensitive operations such as the initial entry of the end-user's PIN, pass phrase, or encryption keys.

The token is capable of performing a variety of cryptographic functions, including public key pair generation (RSA and DSS), digital signature generation (RSA and DSS), unwrapping of session encryption keys, user authentication, and on-line authentication (mutual challenge/response).

The SafeNet Crypto Card Operating System (DKCCOS v2.0) is the proprietary operating system used on the Model 330 Smartcard chip and housed within the iKey 2032 token. Although this operating system is unchangeably embedded in the ROM of the processor chip, it does incorporate a feature in the form of downloadable/executable files (EXFs) that allows for the augmentation and extension of the operating system. It should be noted that there are currently no FIPS approved EXFs and that downloading of EXFs is not currently supported.

With a jump table arrangement, the operating system can accommodate patches or changes on major functions or commands. It also allows the addition of new or custom algorithms. The use of EXFs as an adversarial attack channel is prevented by signing the EXF file in the SafeNet Signing Facility, and allowing it to execute in the token only if the signature is verified in the token with the signing facility public key embedded in ROM.

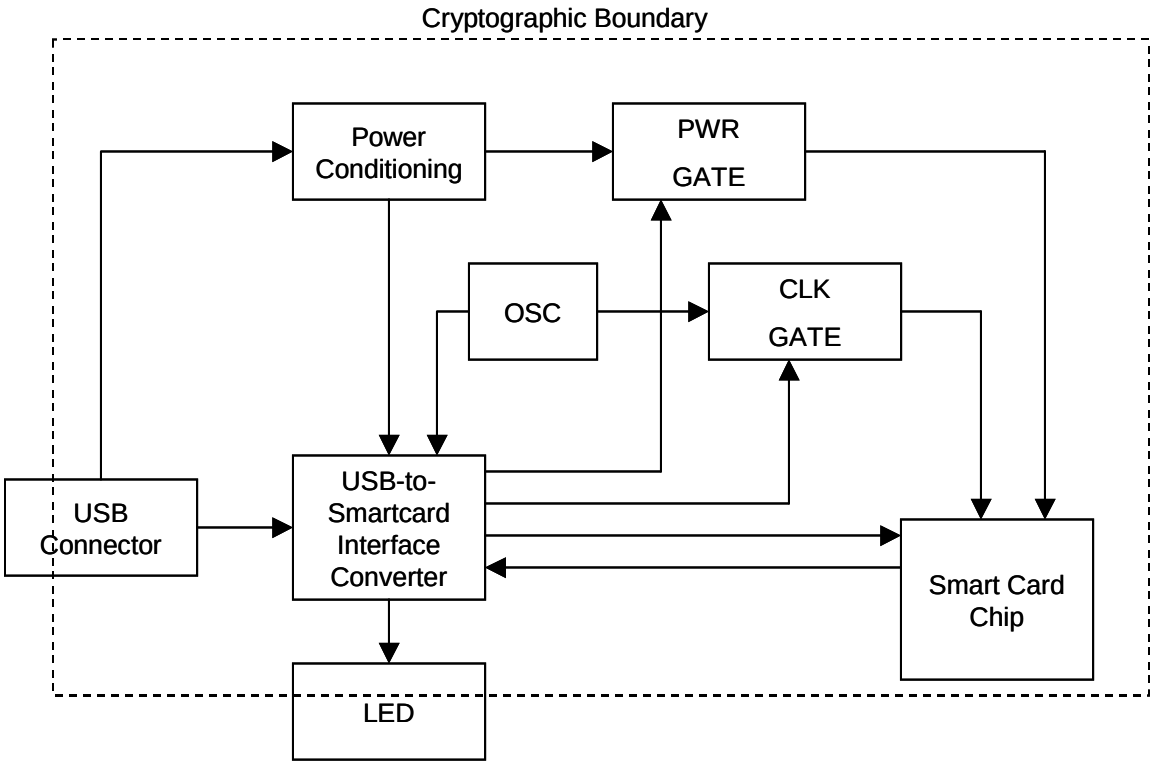

Figure 1. Hardware Block Diagram

The SafeNet iKey 2032 security token uses the same cryptographic interface provider as that used with the SafeNet Model 330 Smartcard, entitled the iKey2000 Series Software. This product combines the products previously marketed as the SignaSURE™ CIP (Cryptoki Interface Provider) and a CSP (Cryptographic Service Provider) for Microsoft Crypto API applications. The product is layered and can be called from either of two APIs.

It allows CAPI-compliant applications to automatically upgrade to token-based information security. The CryptoAPI is a high-level interface between those applications and a standards-based, core cryptographic functionality, and is the foundation technology for the Microsoft Security Framework. This top layer is internally connected to the lower layer, which also has its own API - the Cryptoki API (defined by PKCS #11).

The Cryptoki interface is a medium-level, hardware-independent token interface developed by RSA Data Security, Inc. and its licensees. In the SafeNet iKey2000 Series Software, cryptographic calls from the application program to the API and its extensions are translated and passed either to the token or to the software security module embedded in the product for execution.

When a token is first sensed in a USB port by the iKey2000 Series Software, all public objects of the token are cached in software, so that any operations involving those objects can be performed in software. Operations involving sensitive objects are required to be performed within the token.

The iKey 2000 Series Software uses SafeNet's iKey 2032 token in a public key infrastructure to process and store private cryptographic data. When these critical operations are performed within the hardware token, a much higher level of information security is achieved than that which can be provided by software-only systems.

The iKey 2032 token is capable of performing all private, public, and secret key functions within the token. However, performance of all cryptographic functions within the token is not always practical or desirable. When used in conjunction with the iKey2000 Series Software, all private key functions are performed within the Smartcard chip, including generation of digital signatures. All public key functions are performed within the security module embedded within the iKey2000 Series Software. Secret key functions are performed both within the Smartcard chip and within the iKey software, depending upon the application requesting the secret key function.

3.0 Comprehensive Security Services

The SafeNet iKey 2032 security token applies both symmetric key and public key cryptography. The token provides the following security services:

- **User Authentication**
  User authentication is a process by which an individual may identify himself/herself as the proper owner or user of a device such as a smart token. Typically, a secure token is capable of executing minimal, non-sensitive functions until it is placed into an authenticated state as a result of this activating process.

  Two-factor authentication (based on possession of the smart card chip + knowledge of a secret pass phrase) may be accomplished with the secure smart card chip either by the enterprise Security Officer (SO) or by the end user.

  "Role-based access" is accomplished by means of security nibbles associated with all files in the smart card chip, providing read/update/write/delete/execute access permission to anyone, the enterprise SO, the user, or to no one, depending on the sensitivity of the information contained in the file.

  Under DKCCOS, the PIN file contains either a 20-byte PIN (padded if necessary) or the SHA.1 hash of the pass phrase. The latter provides a higher level of security in the authentication process because the hash of the pass phrase can be initialized to the smart card chip without the need for the owner to disclose the pass phrase itself. This method is used as the transport key during the early life cycle of the smart card chip while the processor module/token passes through a chain of custodians.

- **On-line Authentication**

  On-line authentication is a process by which a mutual authentication can be affected between the iKey 2032 and the on-line database of the iKey 2032 issuing system. This type of authentication normally takes the form of a cryptographically based challenge-response protocol.

  ANSI X9.17 describes a DES-based protocol that utilizes a key shared between the token and the issuing system. A challenge in the form of a random number is passed in one direction, and when received is encrypted by the shared key and the result is returned as the response. The response is then decrypted with the shared key and compared with the original random number. The procedure can then be reversed. This protocol can be implemented using the iKey 2032 security token. This function is not implemented as a method of authenticating a user to the iKey 2032. FIPS Publication 196 describes an equivalent challenge-response protocol based on the use of digital signatures. The iKey 2032 security token can also perform this protocol. As with ANSI X9.17, this method of authentication is not implemented to authenticate a user to the iKey 2032 token.

- **Hybrid Cryptosystem**

  The SafeNet iKey 2032 security token utilizes a hybrid cryptosystem, that is, a cryptographic system made up of both symmetric key and public key cryptographic algorithms. This is done in order to benefit from the strongest features of both types of algorithms in combination.

  For example, symmetric key systems such as DES are very fast and computationally efficient in encrypting / decrypting large data files, and are widely used to protect the integrity of transmitted documents and files independent of encryption. The disadvantages of symmetric key systems are:

  1) They require that the communicating parties share cryptographic keys that must be kept secret. Key management for large symmetric key-only systems, comprising the creation, transportation, storage, recovery, and revocation/destruction of keys, is complex and expensive.

  2) Effective non-repudiation of the source of messages and files cannot be built with symmetric key systems.

  Public key cryptography is complementary to symmetric key cryptography, and the two are used in combination to provide a full set of cryptographic functions.

  The most commonly used public key algorithms in use today, RSA and DSS, are computationally intensive, and are therefore limited in use to digital signatures and key exchange/key agreement. (DSS was designed only for digital signatures). Execution time with these algorithms are held within reasonable ranges through the constraint of operations to short data digest (hash) lengths or key length data blocks.

  Public key cryptosystems are based on the existence of a pair of keys for each user or entity; these two keys are mathematically related such that a short data block encrypted by one of the two keys can only be decrypted with the other. One of the keys, the private key is securely held and used by its owner, but it is never disclosed , even to himself. The other key, the public key, is broadly distributed to anyone who wishes to communicate securely with its owner.

  These characteristics of a public key cryptosystem are ideal for processing digital signatures, and for exchanging symmetric encryption keys with individuals and organizations with whom no previous key sharing relationship had existed.

  <u>Digital Signatures</u>

  The digital signature is intended to provide the same (or higher) level of confidence and trust when placed on an electronic document as a physical handwritten signature on a printed document. The content of the message, file or document to be “signed” is first hashed, or digested, by a secure one-way function to a small (e.g., 20-byte) hash value. A digital

signature is produced by processing the hash value by the highly-protected private key of the originator of the message, file or document. Because the private signature key exists in only one secure repository where it is used, the property of non-repudiation can confidently be attributed to any message, file or document accompanied by a digital signature produced by that private key. The digital signature, to be of any value, needs to be verified by the receiver. The receiver needs only the sender's public key and the same non-secret hashing function.

Symmetric Key Exchange

The other extremely useful function that public key cryptography brings to a hybrid cryptosystem is the ability to provide the two parties involved in an encrypted exchange with the symmetric key used to encrypt the message, file or document being transmitted.

A random number is generated in the sender's cryptographic module and this random number is used as the one-time session encryption key to encrypt the item to be transmitted. The sender's cryptographic module also encrypts the random number with the intended receiver's public key (from a directory) and appends the result to the encrypted item to be transmitted. The intended receiver is the only one who has the intended receiver's private key, and alone can recover the session key, and then decrypt the transmission.

Diffie-Hellman Key Agreement is another algorithm that is used to develop a symmetric session key between two parties. It requires parameter contributions from both parties which can then be combined mathematically by either (or both) parties to derive the session key.

- **Random Number Generation**

A SHA.1-based pseudorandom number generator is implemented within the Model 330 smart card chip, housed within the iKey 2032 security token. A hardware random number generator (HW RNG) is included within the Philips P8WE5032 crypto-processor (Philips provides the original chip wafer for the smart card chip). The HW RNG is used to add entropy to the FIPS 186-1 pseudorandom number generator built into DKCCOS v2.0.

The pseudorandom number generator built into DKCCOS maintains 16 bytes of secret internal state. When a pseudorandom number is needed, this secret internal state and the output of the HW RNG are hashed with a one-way function to produce a 20-byte hash, four bytes of which are used as pseudorandom output, while the remaining 16 bytes become the new secret internal state. Additional uncertainty can be introduced through an additional entropy file each time a pseudo RN is requested. The contents of this file are hashed along with the secret internal state during the update process.

- **Data Encryption / Decryption**

Data encryption and decryption, as applied to messages, files, and documents, is performed with symmetric key algorithms.

The most commonly used symmetric key algorithm is the Data Encryption Standard (DES). It is block cipher algorithm, which uses a 56-bit key and 8-bit parity, and can operate in a number of modes, including electronic code book (ECB) and Cipher Block Chaining (CBC).

Stronger versions of DES include triple-DES and DESX. Triple-DES performs three 56-bit operations (encrypt / decrypt / encrypt) using either a double-length or a triple-length key. DESX uses a 56-bit DES operation both preceded and followed by white masking.

Other commercially-available and proprietary symmetric key encryption algorithms, such as RC2, RC4, CAST and IDEA, are in common use. Any of these algorithms can be added to the iKey 2032 security token via an EXF.

The software security module contained within the iKey 2032 can perform DES in the ECB and CBC modes, DESX, triple-DES, RC2, RC4 and RC5. Although data encryption within the token is infrequently requested, requests for these functions are supported by the iKey 2032.

- **Message Authentication**

  Message authentication are cryptographically-based algorithms for protecting the content integrity of messages, files and documents. Message authentication can be implemented in symmetric key or public key systems.

  A DES-based message authentication method defined in ANSI X9.9 is commonly used to protect wholesale banking funds transfers from alteration during transmission. It uses DES in CBC mode with feedback to generate a Message Authentication Code (MAC) that is appended to the message. It can be used with or without encryption.

  Message authentication with public key cryptography uses digital signatures. Successful digital signature verification by the receiver provides proof that no message alteration could have occurred.

  The SafeNet iKey 2032 security token can perform either method.

- **Public Key Functions and Public Key Pair Generation**

  The SafeNet iKey 2032 security token supports the use of public key cryptography in two primary processes: digital signatures and key exchange for symmetric encryption keys.

  While a single RSA key may be used for both functions, there is a strong rationale behind the trend to use separate keys:

  - The digital signature is an individual, personal and private entity, and the private key that is used to generate digital signatures must be safeguarded such that it can never be used by anyone other than the owner in an authenticated environment. If this level of safeguarding is not afforded the signature private key, then the masquerading of one

individual by another would be possible, thereby compromising the property of non-repudiation.

It is an absolute requirement by many user establishments that the private signature key be generated *within a secure token, by the token end-user* (who is also the owner of the key token), thus providing the token end-user with the total confidence that no other person has ever seen the key. It is also required that the end user cannot know the private key in order to use it, and that the secure token operating system will not permit the private key to be exported from the token.

This treatment of the signature private key provides no means for backup, escrow or recovery. If the end-user loses his/her token, or if the token becomes locked because of multiple erroneous PIN entries, then that signature key pair is dead and any certificates based on it must be revoked.

- It is broadly accepted that the public key pair used for session encryption key exchange needs to be recoverable by an authorized key recovery agent. This requirement is normally imposed by the user enterprise in order to protect availability to its intellectual property, and may additionally be imposed by the government for law enforcement reasons.

  The iKey 2032 security token provides for the injection of public key pairs that are generated outside the token in order to accommodate secure backup procedures. The backup must be done prior to key injection; otherwise it is not possible to access the private key. The iKey 2032 may be configured to allow either the enterprise security officer and/or the end user to inject keys, and to allow the end user to generate keys in the iKey 2032.

  The SafeNet iKey 2032 security token provides for RSA key pair generation in software.

• **Digital Signature Generation and Verification**

The need for secure generation of digital signatures was addressed in the previous section. However, digital signature verification is normally performed in software because it is a non-sensitive public key operation.

The cryptographic module within the iKey 2032, paired with the SafeNet cryptographic interface, is able to perform hashing functions using SHA.1 and to perform digital signature functions in DSS and RSA. When digital signature generation is performed within the token, the hash function is first performed in software.

The iKey 2032 can perform SHA.1 hashing, but for performance reasons is used predominantly with EXF signature verification, random number generation and for pass phrase authentication. In addition, digital signature verification can be performed on the token if desired.

- **Session Encryption Key Exchange**

  The process performed by the sender involves encryption of the random session key ("wrapping") with the public key of the intended receiver. Because this is a public key function, it is normally done in software. Key exchange based on RSA or Diffie-Hellman Key Agreement can be performed within the iKey 2032 security token.

  The process performed by the receiver involves decryption of the random session key ("unwrapping") with the private key of the intended receiver, and is therefore done within the intended receiver's iKey 2032 token.

  An enterprise that chooses to absorb the performance degradation and perform data encryption and decryption within the token, may also perform key wrapping in the iKey 2032 token.

4.0 SafeNet iKey 2032 Security Token Early Life Cycle

The early life cycle that the token and its processor device experiences is significant from the standpoint of the protection that is provided against adversarial scenarios, such as the loading of "Trojan horse routines" and interception of shipments/counterfeit issue. The phases of the life cycle described here are typical for all iKey tokens, but variations in the latter phases may occur. This process is representative of the one implemented for the iKey 2032 security token.

Token Manufacturing Process

It all begins with the microelectronics manufacturer (Philips), who produces the processor chip with the DKCCOS operating system contained in its ROM. Also embedded in the ROM at the time of manufacture are:

- A fabrication key which contains the SHA.1 hash of a pass phrase that the next custodian of the device or token will use to gain authenticated access. The pass phrase is known initially by SafeNet.

- The SHA.1 hash of the public key associated with the SafeNet EXF signing facility.

- The code necessary to perform the Power-On Self Test (POST) of all major processor functions.

All processor chips are tested at the wafer level (prior to dicing into chips).

The microelectronics manufacturer supplies raw die, used to produce the iKey 2032 token.

A SafeNet subtractor is the second custodian of the device. At this phase, the raw die is fabricated into the circuit board within the iKey 2032 token.

Authenticated access is obtained by entering the secret pass phrase. The token file system is formatted, and the hash of the next custodian's pass phrase is entered.

At this phase, user enterprise batch data is entered into the token or printed on the surface of the token by SafeNet.

Authenticated access is obtained by entering the current custodian's pass phrase. Files such as the ATR (answer-to-reset) File, DKIS File, User Entropy File and the Token Configuration File are created and initialized. The hash of the next custodian's pass phrase is entered. Normally this would be the user enterprise security officer.

It is conceivable that this phase would be performed by a certified initialization center and that the user enterprise would contract with the initialization center to enter user specific data (name, account number/employee number, expiration date, PINs/pass phrases, cryptographic keys and photograph) in/on the token in addition to the batch data described above.

User Enterprise Security Officer

Tokens are delivered directly from the initialization center to the end users, and functions are performed by the SO between the security workstation and the user's client workstation.

Authenticated access to a token is obtained by entering the security officer's pass phrase. After entering user-specific data, one of the SO's first tasks is to write the enterprise override conditions to the Configuration File defaults, as required and permitted.

Setting up the token for user authentication can be handled in two ways:

- Receive from the user the hash of his/her pass phrase (if the user has the ability to provide it) and enter the hash into the token.

- Generate a random initial user PIN and enter it into the token, and subsequently deliver the initial PIN to the user via some secure procedure.

The User's Confidentiality public key pair is generated on the SO system (or directly on the token), backed up as required for key recovery, and injected into the token.

The token is then delivered to the end user.

End User

Authenticated access to the token is achieved either by entering the predetermined pass phrase, or by entering the initial user PIN communicated by the SO, and immediately changing it.

The user then generates the user's signature key pair within the token by invoking the function in the application supported by the iKey2000 Series Software. The user then follows enterprise procedures for obtaining digital certificates.

An EXF may be installed or removed during this process, transparent to the user and under control of the iKey2000 Series Software.

5.0 Key Management

The iKey 2032 token provides the following types of keys:

- Single DES
- 2key Triple DES
- RSA public and private keys
- DH/DSA keys

The Security Officer Role is able to generate a DES key. The SO Role may also load an EXF, which will use an RSA public key stored in ROM to verify the signature. Access to the EXF public key is controlled by the LoadEXF command. The User Role is able to generate and access any key type allowed within the token. The User Role may also load an EXF thereby providing access to the EXF public key. In FIPS mode, there is no ability to generate or access cryptographic keys from an unauthenticated mode.

Each of these keys is stored in plaintext in the token's non-volatile memory (EEPROM), and access to each key is governed by the file permissions. The keys may be zeroized at any time by calling the Recycle command. This command will delete all files in the EEPROM except for the SHA-1 protected Configuration file, ATR file, and the SO PIN file.

Additionally, this subject is addressed in the sections of this document labeled:

- Hybrid Cryptosystem
- Public Key Functions and Public Key Pair Generation
- Session Encryption Key Exchange

6.0 PIN / Pass Phrase Management

This subject is addressed in the sections of this document labeled:

- User Authentication
- SafeNet iKey 2032 Security Token Early Life Cycle

7.0 Roles and Services

The iKey 2032 security token provides two roles: the Security Officer (SO) and the User. The Security Officer is tantamount to the Crypto-officer in FIPS 140-1 terminology. Each role is assigned various services.

Security Officer Role

The SO role is responsible for configuring the token (specifying which algorithms are allowed on the token, which keys may be generated, who may generate keys, etc.) and setting up the User's password. Specifically, the SO is allotted the following services:

- ChangeConfiguration
- CreateFile
- DeleteFile
- EndSession
- GenerateDESKey
- GenerateRandomNumber
- GetStatus
- LoadEXF
- ReadBinary
- Recycle
- SelectFile
- SHA1
- UpdatePIN
- WriteBinary

User Role

The User role is essentially the end user and thus has access to all of the cryptographic functions of the module, but does not have access (that the Security Officer has) to the token configuration functions. Specifically, the User is provided the following services:

- CreateFile
- Crypt
- DeleteFile
- DH/DSAGenerateKey
- DHKeyAgreement
- DSASign
- DSAVerify
- EndSession
- Format
- GenerateDESKey
- GenerateRandomNumber
- GetStatus
- LoadEXF
- ReadBinary
- Recycle
- RSADecrypt
- RSAEncrypt
- RSASign
- RSAVerify
- SelectFile
- SHA1
- UpdatePIN
- WriteBinary

Additionally, two command are available without authentication: Verify and GetStatus. These commands only provide general token status and do not provide access to cryptographic services or objects within the token. Furthermore, the SafeNet iKey 2032 security token implements a method of restricting access to data and objects based upon the role authenticated. Each data or key object is stored as a file, and each file has an associated security nibble in the file header. The security nibble determines whether the SO, User, anyone, or no one has access to read, write, update, execute (use a key), or delete the file. This subject is addressed in the "SafeNet iKey 2032 Security Token Early Life Cycle" section of this document and in the section that follows.

8.0 User Enterprise Security Policy

Enterprise security policy varies widely depending on the type of business involved and the value of intellectual property and other assets addressed by the policy. Because the SafeNet iKey 2032 solution is token-based, it follows that some policy issues are dictated by how the token is designed and how it is configured. This is best illustrated with some specific examples:

- Associated with all files in the token are security nibbles that assign access permissions relative to read, update, write, delete and execute to values of never, SO, user, and anyone.

Cryptoki (PKCS #11) is a commonly-used cryptographic token API. When a token is first sensed under this API, all public objects are cached in host PC software, so that any operations involving those objects can be performed at host processing speed in software. In all modes, including the FIPS mode, the token operating system was designed to perform private key operations within the token only if the private key file has only "never" access permissions.

- The Configuration File in the token allows for the enabling/disabling of cryptographic algorithms, algorithm modes, and key lengths. Some of these bits are important for tokens to be exported. Other bits in the Configuration File relate to the enterprise security policy such as:
  - Is a "would-be security officer" allowed to issue a recycle command to a token in his/her possession, and become the de-facto SO, as opposed to being required to enter the pre-designated pass phrase?
  - Is the SO authorized to update a user's PIN on a locked token?

Each of the bits in the Configuration File has a companion bit in a mask, which if set to 0 prevents any further change to the configuration bit during the life of the token, and if set to 1 allows override by the enterprise SO. The mask bit can also be changed from 1 to 0, but not from 0 to 1.

9.0 Rules of Operation

To operate the token in a FIPS-compliant manner, the following rules should be observed:

- Only FIPS approved EXFs should be loaded while in FIPS mode.
- Use of the RSA algorithm should be limited to signature generation/verification (PKCS#1) and key exchange. RSA should not be used for encryption/decryption while in FIPS mode.

10.0 Physical Security

The SafeNet iKey 2032 security token has special physical security mechanisms to allow the end user to detect attempts to tamper with the token. Specifically, the user should check for tamper evidence such as scratches or cuts in, or melting of, the plastic housing of the iKey 2032 token.

If tamper evidence is detected, the user should discontinue use of the token and contact his or her Security Officer.