V-ONE™

Security for a Connected World

SmartPass FIPS Token

FIPS 140-1 Non-Proprietary Cryptographic Module Security Policy

Level 1 Validation

Oct. 1, 2001

Table of Contents

1 Introduction

1.1 Purpose

This document is the non-proprietary Cryptographic Module Security Policy for the V-ONE FIPS token (a virtual cryptographic authentication token). This Cryptographic Module Security Policy is part of the FIPS 140-1[1] documentation prepared by V-ONE for validation of the SmartPass FIPS token module. The SmartPass FIPS token provides robust security in a flexible software module, meeting all FIPS 140-1 Level 1 requirements. This security policy describes how the SmartPass FIPS token meets the FIPS 140-1 requirements, and how the SmartPass FIPS token is securely used within V-ONE products.

1.2 Audience

This document is intended for FIPS 140-1 testers, National Institute of Standards and Technology (NIST) and Communications Security Establishment (CSE) reviewers, and customers interested in the SmartPass FIPS token's functionality and compliance with FIPS 140-1. This security policy describes the SmartPass FIPS token using technical terminology associated with computer security and FIPS 140-1. Readers seeking additional information are referred to the following sources:

> For more detailed information about the SmartPass FIPS token and V-ONE's entire product line, please visit the V-ONE Web site at http://www.v-one.com.
>
> For more information about the FIPS 140-1 standard and validation program please visit the NIST Web site at http://csrc.nist.gov/cryptval.
>
> For answers to technical or sales related questions please refer to the contacts listed on the V-ONE Web site at http://www.v-one.com.

2 The SmartPass FIPS Token

The V-ONE SmartPass FIPS token is a software cryptographic module that provides key generation, storage, and exchange services. These services are modeled after those provided by popular smart cards, allowing seamless substitution of software or hardware tokens. Whether V-ONE customers require the high-security provided by hardware cryptography, or the lower cost that comes with software-only implementations, the SmartPass FIPS token allows all of V-ONE's VPN products the flexibility to support both.

[1] Federal Information Processing Standards Publication 140-1 -- *Security Requirements for Cryptographic Modules.* (FIPS 140-1)

2.1 SmartPass FIPS Token Functionality Overview

The SmartPass FIPS token uses DES (FIPS PUB 46-3: Data Encryption Standard, and Triple DES) encryption to store keys, data, and files in a single secure file on the hard disk of a personal computer, a floppy diskette, or a smart card. Whereas the software is referred to as the SmartPass FIPS Token, the encrypted storage file will henceforth be referred to as the SmartPass FIPS token file. The module maintains a set of 8 DES keys (created with FIPS 171 and ANSI X9.17 Appendix C compliant random number generation and DES key generation). These keys are stored with a single Crypto Officer password plus 7 separate user passwords. All passwords and DES keys are DES encrypted.

Each user of the SmartPass FIPS token can create encrypted key files suitable for storing cryptographic keys, sensitive data files, session keying information, or session data files. The SmartPass FIPS token allows users to specify read, write, and update access for each file for other users of the SmartPass FIPS token.

The SmartPass FIPS token will generate ephemeral session keys by accepting a challenge and generating a response and DES or 3DES (FIPS PUB 46-3) session key. SmartPass FIPS token users can then execute encrypted sessions using the session and store temporary session files encrypted with that key.

2.2 Module Interfaces

The SmartPass FIPS token is officially considered to be a multi-chip standalone module for FIPS 140-1. As such, the module must include a computer running an operating system (OS) and interfacing to the computer keyboard, mouse, screen, floppy drive, CD-ROM drive, speaker, microphone inputs, serial ports, parallel ports, and power plug. However, once information is processed through these physical interfaces, the SmartPass FIPS token software module provides a logical interface through an Application Programming Interface (API). This logical interface exposes services through API functions to other programs such as V-ONE's SmartGate, SmartGuard, and SmartWall.

Thus, there is a single API interface provided by the SmartPass FIPS token, which is further logically divided into data input, data output, control input, and status output interfaces. Since it is a software module, the SmartPass FIPS token does not provide a separate power or maintenance access interface beyond the power interface provided by the personal computer itself. Data input is represented by input parameters described in functions in the *SmartPass FIPS Token Software Design Overview Level 1 Validation*. Data output is logically provided through output parameters, control input is provided by the API function calls, and status output is provided by the return codes from each function.

2.3 Roles and Services

The SmartPass FIPS token supports two distinct roles using password authentication: Crypto Officer and User. There is a single Crypto Officer role for each SmartPass FIPS token with a single Crypto Officer PIN code. Similarly, there is a single User role for the SmartPass FIPS token, but there are seven separate User PIN codes.

2.3.1 Authentication

Both the Crypto Officer and users authenticate to the SmartPass FIPS token by providing a PIN with the CheckCode function. Each Crypto Officer and User PIN code is actually an 8 byte case insensitive password. Currently, only the Crypto Officer PIN also known as the format code (PIN code 0) and the access code (PIN code 1) are used. Due to the sensitivity of the access code, tokens created in SmartPass 4.2 or later have a case-sensitive access code of 4-16 characters (see section 2.3.4).

2.3.2 Services

The SmartPass FIPS token offers the following API functions to implement the services that were described in section 2.1:

CreateNew	**Clear**	**LoadCode**
GetSerialNumber	**GetCurrentFileInfo**	**GetCurrentFileNumber**
LoadReader	**UnloadReader**	**OpenReader**
CloseReader	**CheckCode**	**UpdateCode**
SelectFile	**MakeFile**	**EraseFile**
ReadFile	**WriteFile**	**UpdateFile**
LockFile		
SessionInit	**SessionFinal**	
WriteFileCipher	**UpdateFileCipher**	**ReadFileCipher**

2.3.3 Crypto Officer Role

The role of the Crypto Officer includes creation and destruction of the SmartPass FIPS token, and initialization of the SmartPass FIPS token including User PIN codes. The Crypto Officer does this using the *CreateNew* function, which initializes a blank SmartPass FIPS token with default PIN codes. The Crypto Officer then uses *LoadCode* to change the default PIN codes to values that will be given to the Users. At any time subsequent to this, the Crypto Officer (and only the Crypto Officer) may run the *Clear* function to destroy the SmartPass FIPS token, overwriting the encrypted keys in the SmartPass FIPS token file with zeros. Note: The functionality for formatting/clearing the token is only present in the Win32 client; in the WinCE, Mac, and Unix client one simply deletes the token file and then creates a new token file.

The following are Crypto Officer functions:

CreateNew **Clear** **LoadCode**

2.3.4 User Role

The SmartPass FIPS token User role includes creation, storage, and reading of encrypted files on the SmartPass FIPS token, as well as symmetric key session functions. The first action of a user is usually to change the PIN code assigned by the Crypto Officer using the *UpdateCode* function. Subsequent to this, a user can exercise the following API functions under the User role:

GetSerialNumber **GetCurrentFileInfo** **GetCurrentFileNumber**

LoadReader **UnloadReader** **OpenReader**
CloseReader **CheckCode** **UpdateCode**

SelectFile **MakeFile** **EraseFile**
ReadFile **WriteFile** **UpdateFile**
LockFile

SessionInit **SessionFinal**
WriteFileCipher **UpdateFileCipher** **ReadFileCipher**

As stated previously only user PIN code 1, the access code, is currently used. Due to the sensitivity of the access code, tokens created in SmartPass 4.2 or later have a case-sensitive access code. Also, the SmartPass administrator may configure the client to change the minimum length (default 4, range 4-16) and place restrictions on the content (e.g., require a mixture of both upper and lowercase characters plus numbers, etc.) to ensure the user does not select an access code that can be easily guessed.

2.4 Finite State Machine

The SmartPass FIPS token is designed around a Finite State Machine (FSM) which is detailed in a V-ONE proprietary document (*SmartPass FIPS Token FIPS 140-1 Proprietary Finite State Machine – Level 1 Validation*). Parties interested in reviewing this document should contact V-ONE through the sources listed in section 1.2.

2.5 Physical Security

The SmartPass FIPS token exists as a shared DLL under Windows or as part of a statically linked application on Mac and Unix. SmartPass runs on Windows 95, 98, 98 SE, ME, NT, 2000, CE, Mac OS, Linux, and Solaris operating systems. The operating system must be configured for single-user mode. However, for FIPS 140-1 purposes, the module was evaluated against Level 1 FIPS 140-1 physical security requirements when running on a standard Intel-compatible personal computer with the Windows 98 operating system. This platform meets all Level 1 FIPS 140-1 physical security requirements, providing a multi-chip standalone module with production grade equipment, standard passivation, and a strong enclosure.

Standard installations of Windows and Mac OS are single-user; however, Linux and Solaris must be configured for single-user mode. On Linux and Solaris, the administrator must delete or disable all accounts except for root and one normal user. To ensure only one user can be logged in at a time, both accounts must only allow console access logins and there must not be any remote server services running (e.g., telnet or rlogin server daemon). Services that only allow local access connections (e.g., the SmartPass proxies) or client applications that perform outgoing connections (e.g., telnet client, ftp client, web browser, etc.) are allowed. The root account will be used for installing/uninstalling software and creating/administrating the user account; the user account will be used for running applications (this is equivalent to the roles of the Administrator and User accounts on WinNT/2000). SmartPass can only be installed/uninstalled by the root user but any user can run SmartPass.

2.6 Software Security

The SmartPass FIPS token software is written in C and C++ according to the documentation in *SmartPass FIPS Token Software Design Overview Level 1 Validation.* This document is proprietary, and parties wishing to review it should contact V-ONE through the sources listed in section 1.2.

2.7 Operating System Security

On Microsoft’s Windows 95, 98, 98 SE, ME, NT, 2000, and CE platforms, the SmartPass FIPS token software is implemented as a single shared loadable module (dynamic link library [DLL]) and is always distributed as part of a self-extracting executable to discourage unauthorized modification. On Mac OS, Linux, and Solaris it is statically linked to the SmartPass application. The operating system must be configured for single-user mode (see section 2.5). Additionally, a cryptographic mechanism is used within the module to ensure that the code has not been accidentally or maliciously modified from its evaluated configuration (see section 2.11).

The evaluated platform for the module is Windows 98. Since on this platform the SmartPass FIPS token is implemented as a DLL, the API functions can only be called one at a time to access the SmartPass FIPS token file. As depicted in Figure 1, the LLVCAT.DLL is loaded into a code segment in memory, with a separate data segment and a pointer to the physical SmartPass FIPS token file.

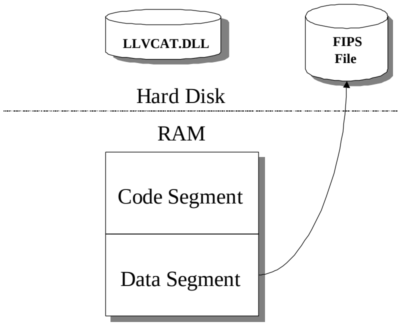

Figure 1 – Loading of LLVCAT.DLL into RAM

The LLVCAT.DLL runs on the Windows operating system. Windows 95, 98, 98 SE, ME, and CE have been defined for FIPS purposes to be a single-user operating system; Windows NT and 2000 must be configured in single-user mode. If while running SmartPass a user enters the Control Panel and double-clicks on the V-ONE FIPS token icon then the computer will load a second copy of the DLL, as depicted in Figure 2, with a separate data segment. The second SmartPass FIPS token module will apply the same access controls to the physical SmartPass FIPS file, maintaining the integrity of the SmartPass FIPS token.

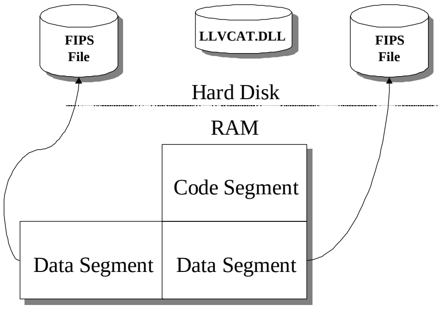

Figure 2 – Single use of an individual SmartPass FIPS file

Note: On WinCE the user does not load a second copy of the DLL (called SPCEFIPS.DLL instead of LLVCAT.DLL) via the Control Panel to format the token, edit the token (i.e., delete servers or modify a server’s info), or change the access code. On WinCE, Mac OS, Linux, and Solaris the user cannot format the token but instead deletes it and creates a new one. On WinCE editing the token and changing the access code is done through the SmartPass application. On Mac OS editing the token is not supported and changing the access code is done through the SmartPass application. On Linux and Solaris editing the token is not supported and changing the access code is done through the On-Line Registration (OLR) application.

2.8 Cryptographic Key Management

The SmartPass FIPS token uses three classes of cryptographic keys: SmartPass FIPS token keys, password-based encryption (PBE) keys and ephemeral session keys. All three types of keys are generated or managed differently as described below.

When a Crypto Officer first creates a SmartPass FIPS token, 8 DES keys (SmartPass FIPS token keys) are created using the ANSI X9.17 random number generator described in section 2.11.3. SmartPass FIPS token users will employ these keys for encryption and decryption of the virtual files within the SmartPass FIPS token physical file. 8 PBE DES keys are generated from the Crypto Officer PIN code and the 7 User PIN codes. Each PBE DES key is generated by applying the SHA-1 hashing algorithm and XOR folding to each PIN. The PIN codes and the 8 copies of these 8 DES keys are then encrypted using the PBE DES keys. The encrypted PIN codes and encrypted token keys are stored in the PIN code records in the physical SmartPass FIPS token file. The PBE DES keys are not stored but are regenerated whenever they are required. Users successfully authenticating as described in section 2.3 can decrypt a set of the module DES keys by providing the correct PIN. Currently only the access code (i.e., User PIN code 1) can access the data stored in the FIPS token file; the 7 extra copies of the 8 DES keys encrypted using PBE DES keys generated from the Crypto Officer PIN code and 6 other User PIN codes are overwritten with random data.

In addition to these types of keys, users can create ephemeral DES and 3DES session keys using a simple challenge-response protocol. When a SmartPass FIPS token user receives a challenge, it can be provided to the SmartPass FIPS token, which will initiate a session by creating a response and generating a session DES or 3DES key. The response can be transmitted to the creator of the challenge, and the SmartPass FIPS token can subsequently use the session key to encrypt files for temporary storage of information. Session keys are ephemeral and are destroyed when the session is terminated.

All permanent keys in the SmartPass FIPS token are stored in encrypted form, and are destroyed (overwritten by zeros) when the Crypto Officer issues a clear command to the module.

2.9 Cryptographic Algorithms

SmartPass communicates with the SmartGate server using DES and/or 3DES (depending on how the SmartGate server is configured) using data from the FIPS token. The FIPS token uses DES for encryption of stored data and integrity checks; it uses SHA-1 for message digests in password-based encryption, random number generation, etc.

The V-ONE SHA-1 module implements the Secure Hashing Algorithm (SHA-1), and has been validated as conforming to Federal Information Processing Standard Publication (FIPS PUB) 180-1, *Secure Hash Standard (SHS)*. V-ONE has been issued a certificate signed by the National Institute of Standards and Technology (NIST) and the Communications Security Establishment (CSE), and is listed on the official validation Web site (http://csrc.nist.gov/cryptval/dss/dsaval.htm Cert #10).

The V-ONE DES module implements the Data Encryption Standard (DES) and has been validated as conforming to Federal Information Processing Standard Publication 46-3, *Data Encryption Standard (DES)*. The V-ONE DES module was validated by NIST using the Monte Carlo test described in NBS Special Publication 500-20. V-ONE has been issued a certificate signed by the National Institute of Standards and Technology (NIST), July 25, 1994. This certification predates the numbering of DES certificates, which began in 1996 and thus is not listed on the official validation Web site (http://csrc.nist.gov/cryptval/des/desval.htm).

The V-ONE 3DES module implements the Data Encryption Standard (DES) and has been validated as conforming to Federal Information Processing Standard Publication 46-3, *Data Encryption Standard (DES), Triple DES*. The V-ONE 3DES module was validated by NIST using the Monte Carlo test described in NBS Special Publication 800-20. V-ONE has been issued a certificate signed by the National Institute of Standards and Technology (NIST) and the Communications Security Establishment (CSE), and is listed on the official validation Web site (http://csrc.nist.gov/cryptval/des/tripledesval.html Cert #46).

The SmartPass FIPS token ensures correct operation of both the V-ONE SHA-1 module and V-ONE DES/3DES module using cryptographic algorithm self-tests described in section 2.11.

2.10 EMI/EMC

Although the SmartPass FIPS token consists entirely of software, the FIPS 140-1 evaluated platform is run on a standard PC, Mac, or Sparc that has been tested for and meets applicable Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business or home use as defined in Subpart J of FCC Part 15.

2.11 Self Tests

The SmartPass FIPS token includes several self-tests to ensure the integrity and correct operation of the module. These include the following:

- SmartPass FIPS token Module Software Test
- SHA-1 Cryptographic Algorithm Known Answer Test
- DES Encrypt Cryptographic Algorithm Known Answer Test
- DES Decrypt Cryptographic Algorithm Known Answer Test
- 3DES Encrypt Cryptographic Algorithm Known Answer Test
- 3DES Decrypt Cryptographic Algorithm Known Answer Test
- Continuous Random Number Generator Test

2.11.1 SmartPass FIPS token Module Software Test

The SmartPass FIPS token software module performs a self-integrity check automatically every time it is loaded. The module computes a DES Data Authentication Code (DAC) over the entire module as per FIPS PUB 113, and compares the result to a separately stored version of the DAC. Should the SmartPass FIPS token module software be corrupt or has been tampered with, the SmartPass FIPS token Module Software Test will fail, alerting the user to the problem and will refuse to load the module.

On Mac OS, Linux, and Solaris the module is statically linked to the SmartPass application; therefore, the DAC is calculated on the entire SmartPass application.

2.11.2 Known Answer Tests

The SmartPass FIPS token software automatically performs known answer tests of all cryptographic algorithms during module startup. This includes SHA-1 hashing a known block and comparing against the stored answer, DES/3DES encryption of a known block and comparison against the expected ciphertext, and DES/3DES decryption of a known block and comparison against the expected plaintext.

2.11.3 Continuous Random Number Generator Test

The SmartPass FIPS token incorporates the V-ONE random number generator (RNG). This RNG is compliant with American National Standards Institute (ANSI) X9.17 Appendix C random number and DES key generation. The RNG also incorporates a continuous random number generation test, which compares current blocks of data to previous blocks to prevent against failure of the random number generator to a constant value.