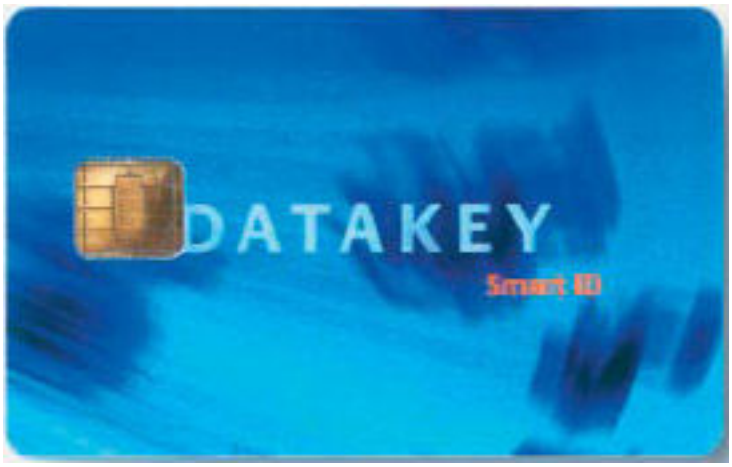
FIPS 140-2
SECURITY POLICY

MODEL 330G3 SMART CARD

Version 1.1



DATAKEY®

407 West Travelers Trail
Burnsville, MN 55337-2554
(612) 890-6850

Table of Contents

Introduction

1.1. Scope

This document describes the cryptographic module security policy for the Datakey Model 330G3 smart card (Hardware version: 1.0, Firmware Version 2.0). It contains specification of the security rules, under which the cryptographic module operates, including the security rules derived from the requirements of the FIPS 140-2 standard and the additional security rules imposed by the GSC-IS and Biometric authentication application executable (G3 EXF) Version 21 loaded on the 330 smart card.

1.2. Overview

The 330G3 cryptographic module is a smart card compliant with parts 1 through 4 of the ISO 7816 standard, which define the physical characteristics, contact arrangement /location, electrical characteristics, and interface protocol.

The chip platform operating system is based on Datakey Crypto Card Operating System (DKCCOS version 2.0) and the application extension G3 EXF. Together they manage all the low level resources, cryptographic algorithms implementation, object access control and applications life cycle.

The G3 EXF is a state of the art application designed and developed using Datakey's extensive experience in secure operating system development. When downloaded to the chip the G3 EXF provides the following high-level security services:

- Multiapplication secure storage and retrieval of objects and digital credentials.
- Authentication of the cardholder and the security officer.
- Supports on-token fingerprint verification using algorithms compatible with Precise BioMatch MOC.
- Secure secret key agreement features are included to allow for secure encrypted injection of the private fingerprint templates.
- Cryptographic services such as SHA-1, DES, Triple DES, RSA Sign/Verify, RSA Encrypt/Decrypt and DSA Sign/Verify with on board key generation.

1.3. Model 330G3 smart card architecture

- The architecture of the Model 330G3 smart card is different from the FIPS 140-1 Level 2 certified Model 330 smart card. Datakey designed the Model 330G3 to provide capabilities required for the FIPS 140-2 certification, the GSC-IS version 2.1 interoperability standard compliance and to provide strong biometric authentication support using on-token fingerprint verification.

1.4. Related Standards and Documents

CC	ISO 15408 – Information Technology – Security Techniques – Evaluation Criteria for IT Security (Hereafter referred to as Common Criteria or CC)
ISO 7816-1	ISO/IEC 7816-1 (1987): "Identification cards – Integrated circuit(s) cards with contacts, Part 1: Physical characteristics".
ISO 7816-2	ISO/IEC 7816-2 (1988): "Identification cards – Integrated circuit(s) cards with contacts, Part 2: Dimensions and locations of the contacts".
ISO 7816-3	ISO/IEC 7816-3 (1989): "Identification cards – Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols".
ISO 7816-4	ISO/IEC 7816-4: "Identification cards – Integrated circuit(s) cards with contacts, Part 4:
PKCS 1	PKCS #1: RSA Encryption Standard, Version 1.5, November 1993

2. Glossary

Authentication Data	Comprise the officer identifier, certificate, role and privileges.
Bond-out chips	Raw ICs, which have been mounted on a small board. Wire bonds are connected from the IC's input/output pads to the carrier, which has contacts on its reverse side. Bond-out chips are sometimes referred to as a module.
Card disablement	The IC function related to terminating all operations other than possibly some limited audit functions. Card disablement is permanent.
Card embedder	A manufacturer who assembles a card and integrated circuit.
Card holder	A person to whom a card has been legitimately issued (a user).
Card issuer	An institution, which issues cards to cardholders.
Card Operating System (COS)	Operating system developer specific code, written in the microprocessor's native or machine code.
Card reader	A machine capable of reading and/or writing to a card, such as magnetic stripe card or smart card.
Carrier	The holder in which an operational integrated circuit is placed. This is typically the thin, credit card sized piece of plastic that is known as a smart card.
DKCCOS	Datakey Cryptographic Card Operating System
Die	The semiconductor IC without any packaging or connections.
EEPROM	Electrically Erasable Programmable Read Only Memory. A non-volatile memory technology where data can be electrically erased and rewritten.
FAR	False Acceptance Rate
FRR	False Rejection Rate
Failure analysis	The compilation of techniques used by semiconductor development and testing labs to identify the operating problems in newly designed or modified integrated circuits. Such techniques include not only observation (to determine what is not functioning properly) but also modification of IC internal structure (to determine fixes).
First use indication	The IC function related to setting a specific audit bit indicating that the smart card is now in the issued, operational state and can be used for its intended function.
I&A	Identification and Authentication
IC	Integrated Circuit. Electronic component(s) contained on a single chip and designed to perform processing and/or memory functions.
ICC	Integrated Circuit Card. A card into which has been inserted one

	or more ICs.
ID	Identity (also, a token asserting an identity)
Initialization	The process of writing specific information into Non-Volatile Memory during the early card life cycle.
IP	Internet Protocol
ISO	International Standards Organization
Life cycle identifiers	The specific identification of chip fabricator identifier, operating software identifier, chip module identifier, chip embedder identifier, initialiser identifier, initialization equipment identifier, personaliser identifier and personalization equipment identifier.
MOC	Match-On-Card
Modules	A functional assembly for use with other assemblies. These may be separate parts of an IC (CPU, Coprocessor, ROM, RAM, etc.), bond-out chips, or software components.
NIST	National Institute of Standards and Technologies
Non-volatile memory	A semiconductor memory that retains its content when power is removed. (i.e. ROM, EEPROM, FLASH).
Personalization	The process of writing specific information into the non-volatile memory in preparing the IC for issuance to users.
PIN	Personal Identification Number
Platform	A term representing an operational smart card system.
Post-issuance	The time period during which the smart card is in the hands of the cardholder. In some smart cards, additional functionality can be loaded into the smart card post-issuance.
PP	Protection Profile
RAM	Random Access Memory. A volatile, randomly accessible memory (used in the IC) that requires power to maintain data.
ROM	Read Only Memory. A non-volatile memory (used in the IC) that requires no power to maintain. ROM data is often contained in one of the numerous masks used during manufacture.
RSA	Rivest, Shamir, Adleman (encryption algorithm)
Security Officer	The administrator of the CM system. The security officer has in addition to the administrative privileges also all the privileges a registration officer can have
SHA	Secure Hash Algorithm
Smart card	A shaped piece of plastic or other carrier with a small computer chip embedded into it.
Terminal	The device used in conjunction with the CAD at the point of transaction.

3. Security Levels

The Datakey Model 330G3 meets all requirements for FIPS 140-2 level 2. Refer to the following table for individual security requirements:

Security Requirements	Certification Level
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services, and Authentication	2
Finite State Model	2
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	2
Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)	3
Self Tests	2
Design Assurance	2
Mitigation of Other Attacks	N/A

4. Cryptographic Module Specification

4.1. Cryptographic Boundary

The cryptographic boundary for the Datakey Model 330G3 Smart Card is the physical boundary of the card itself. Although the boundary could be defined as the physical micro-module containing the processor chip, the embedding of the micro-module within the plastic card provides no further cryptographic function. The card provides utility to the micro-module as a standalone cryptographic module component and enhances the tamper evidence of the cryptographic module.

The cryptographic module submitted for testing is a single-chip module combining the DKCCOS v2.0 operating system in ROM and the G3 EXF in EEPROM, together to form the cryptographic module that meets the FIPS 140-2 level 2 requirements.

This platform contains the following hardware components:

- A crypto co-processor optimized for public key cryptographic calculations
- A Triple DES (Data Encryption Standard) Co-processor.
- 32K User ROM.
- 2300 bytes RAM.
- High-reliability 32K EEPROM for both data storage and program execution.

4.2. Hardware Security Features

The following features are provided by the Philips P8WE5032 microcontroller.

- Power-up / power-down reset,
- Low / high supply voltage sensors,
- Low / high clock frequency sensors,
- Low / high temperature sensors,
- On-chip self test with signature technique,
- EEPROM erase / write timing independent from external clock,
- EEPROM erase /write operation controlled by hardware sequencer,
- Electronic fuses for safeguarded mode control,

These hardware security features were not tested as part of the FIPS 140-2 validation.

4.3. Physical structure

The die contains bonding pads for GND (ground), VCC (supply voltage), CLK (clock signal), RST (external reset signal), I/O 1 (used for half-duplex input / output communication), and I/O 2 (currently not used).

4.4. Fabrication Process

The die is attached with adhesive to the back of a film-based, ISO 7816-compliant contact substrate, such that the die bonding pads and the contact substrate bonding pads

are accessible. Wire bonds are made between the die bonding pads and the contact substrate bonding pads for GND, VCC, CLK, RST and I/O. A measured amount of encapsulant is flowed over the contact substrate, such that the die and wire bonds are contained within the encapsulant.

The micro-module is glued into a matching milled cavity in a plastic card, which complies with the dimensional and other physical requirements of the ISO 7816-1 standard. Typical customization of the card for the application and / or user enterprise may be accomplished prior to and subsequent to micro-module insertion.

Silk screen (or other mass printing) and magnetic stripe application are performed prior to micro-module insertion. Personalization printing (such as names, account numbers, and photographs) and writing of the magnetic stripe data and personal data on the chip are performed after micro-module insertion.

5. Cryptographic Module Ports and Interfaces

5.1. Physical Interface

Five electrical connections are made between the die and the contact substrate of the smart card:

- **VSS**, Ground (reference voltage).
- **VDD**, Power supply input.
- **RST**, External reset signal from the interface device (card read / write device)
- **CLK**, External clock (3.517 MHz).
- **I/O**, Input or output for serial data to / from the processor.

The above five electronic signals are in full compliance with ISO 7816-3. The VPP (programming voltage) contact is not used because the EEPROM of the chip contains its own internal programming voltage generation. Also the I/O 2 and I/O 3 ports, which are present on the chip, are not connected to the contact substrate.

Reference: ISO 7816 Part 3 Identification Cards – Integrated Circuit(s) Cards with Contacts – Electronic Signals and Transmission Protocols

5.2. Logical Interface

The Datakey DKCCOS operating system controls the logical interface thru a well-defined set of Application Protocol Data Unit (APDU) commands. It manages the secure object storage system, interface protocol and parameters, interprets and executes external commands. The G3 EXF provides the functions required for FIPS 140-2 Level 2 compliance and provides an interface thru a set of APDUs that are compliant with the GSC-IS Interoperability standard and adds the following functionality:

- Each sub-directory (DF) can contain its own UserPIN and SOPIN. The DF must be selected first prior to executing either VerifyCHV or ChangeCHV/PIN command.
- DeleteFile command will remove a whole sub-directory and the files it may contain.
- Supports on-token fingerprint verification using algorithms compatible with Precise BioMatch MOC.
- Secure secret key agreement features using Diffie Hellman are included to allow for secure DES encrypted injection of the private fingerprint templates.
- This EXF can only be removed by executing the Recycle command.

The APDU communication protocol defines the following four logical interfaces as per the FIPS 140-2 standard as follows:

a) Data Input interface: The input data field of the command APDU comprises the data input interface of the module. All input parameters can only be passed through this interface.
b) Data Output interface: The output data field of the response APDU comprises the data output interface of the module. All output data can only be passed through this interface.

c) Control Input interface: The command APDU header consisting of the CLA, INS, P1, P2 and LC bytes comprises the control input interface. All control parameters for module execution can only be passed through this interface
d) Status Output interface: The status words SW1 and SW2 of the response APDU comprise the status output interface. All error codes and output indicators are output through this interface.

References: 1- DKCCOS v2.0 Interface Control Document, June 22, 1999.

2- GSA APDU Set and Multiple PIN support interface Control Document Addendum to DKCCOS v2 Interface Control Document Revision 2.0, July 14, 2003.

3- GSA APDU Set, Biometrics, and Multiple PIN support interface Control Document Addendum to DKCCOS v2 Interface Control Document Revision 3.0, March 1, 2004.

6. Roles, Services, and Authentication

6.1. Roles

The Datakey Model 330G3 smart card provides two roles, the Security Officer (SO) and the User role. Each sub-directory (DF) can also implement its own SO and User role. The Security Officer is tantamount to the Crypto-officer in FIPS 140-2 terminology. Each role is assigned various services. Please see the following table for a list of all services available to a particular role in FIPS mode of operation. When an operator is in a particular role the module state is set to indicate this. For example: When the operator is authenticated as a SO the corresponding module state is SO_AUTH, when the operator is authenticated as a User the corresponding module state is USER_AUTH and when no operator is authenticated the module state is IDLE

State / Command	Reset	Error	Unformatted	Idle	User authenticated	SO authenticated
Get Status		x	x	x	x	x
SoftReset		x	x	x	x	x
Format			x			
GenerateRandomNumber				x	x	x
LoadEXF				x	x	x
ReadBinary				x	x	x
Recycle				x	x	x
SelectFile				x	x	x
SHA1				x	x	x
Verify				x	x	x
UpdateBinary				x	x	x
WriteBinary				x	x	x
CreateFile					x	x
Crypt					x	
DeleteFile					x	x
DH/DSAGenerateKey					x	
DHKeyAgreement					x	
DHDESKeyAgreement					x	
DSASign					x	
DSAVerify					x	
EndSession					x	x
GenerateDESKey					x	x
RSADecrypt					x	
RSAEncrypt					x	
RSAGenerateKey					x	
RSASign					x	
RSAVerify					x	
UpdatePIN					x	x
ChangeCHV/PIN					x	x

State / Command	Reset	Error	Unformatted	Idle	User authenticated	SO authenticated
InternalAuthenticate					x	x
ChangeConfiguration						x
PBCreateTemplate					x	x
PBGetPublicTemplate				x	x	x
PBVerify				x	x	

6.1.1. Security Officer Role

The Security Officer (SO) role is responsible for configuring the card by changing the configuration file settings (specifying which algorithms are allowed by the card, which keys may be generated, and who may generate keys), setting up the User's PIN and unblock User PINs. It is the SO's responsibility to ensure that the configuration file settings are set so that the module is in FIPS mode (see FIPS Mode section).

6.1.2. 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 the access (that the Security Officer has) to the card configuration functions.

Additionally, several unauthenticated services are available. These services are listed in the table above. The services in the Idle (unauthenticated) state only provide general card status and do not provide access to cryptographic services or objects on the card.

The Datakey Model 330G3 smart card implements a method of restricting access to data and objects based upon the role authenticated. Each data or key object is stored in a file, and each file has associated security permissions (nibbles) that are set during its creation. The security permissions determine whether the Security Officer (SO), User, anyone (Always), or no one (Never) has access to read, write, update, execute (use a key), or delete the file.

The security nibble definitions including the definitions of each type of access are described below:

Security Nibble Structure					
Byte 1		**Byte 2**		**Byte 3**	
SN1	**SN2**	**SN3**	**SN4**	**SN5**	**SN6**
Read	Update	Write	Delete	RFU	RFU

Security Nibble Definition	
Role	**Value**
Never	0x0
SO	0x4
User	0x2
Always	0x1

(Permissions may be combined in the UNIX style. For example, 0x7 gives object file access to the SO, the User, and to anyone in the Idle state.)

Security Nibble Type Definition	
Nibble Type	**Definition**
Read	File may be read by entity with ReadBinary command
Update	File data may be written by the entity up to the high water mark[1] with WriteBinary command
Write	File data may be written by the entity at the high water mark with WriteBinary command
Delete	File may be deleted by the entity with DeleteFile command
Execute	File may be used as a key file
RFU	Reserved for future use

[1] The high water mark is the length of the data that is currently present in a file

6.2. Services

6.2.1. DKCCOS Services

Please refer to the ICD document for detailed information about each function including the required inputs and expected outputs

PIN and Fingerprint Management services

- **ChangeChvPin**: Updates the PIN of the given type if the given current PIN is valid for the currently authenticated entity in accordance to the GSA interoperability specification.
- **EndSession**: Ends the current authenticated session, returning the card to the idle state.
- **UpdatePIN**: Updates the PIN of the given type if the given current PIN is valid for the currently authenticated entity.
- **Verify**: Hashes the given data (operator pin) and compares the result with the value stored in the card's pin object container. If comparison is successful the module state is set to indicate successful authentication of the operator.
- **PBCreateTemplate**: Creates the fingerprint template file on the token, optionally decrypts the contents of the file as it comes in, and writes the initialized data to the file.
- **PBGetPublicTemplate**: Returns the public template data structure that is contained within the fingerprint template file on the token. The fingerprint template file must have been previously created with the PBCreateTemplate command.
- **PBVerify**: Performs a match between a private fingerprint template and test data obtained by reading a fingerprint

Cryptographic services

- **DH/DSAGenerateKey:** Generates a private exponent of the given length and computes the public/private key pairs for use by the Diffie-Hellman algorithm, the DSA algorithm, or both.
- **DHKeyAgreement:** Completes a Diffie-Hellman key negotiation with the given DH public key y, the DH private key x in the given file, and the DH parameter p: negotiated_key = y^x mod p. The negotiated key is written to the command response field.
- **DSASign:** Performs a DSA signature on the given data with the private key and parameters in the given files. The returned signature consists of two 20-bytes values, r and s.
- **DSAVerify:** Performs a DSA signature verification on the given data with the public key and parameters in the given files. A return value of OK indicates the signature verified correctly, while a return value of Authentication Failed indicates the signature did not verify correctly.
- **Crypt**: Performs a DES/Triple DES symmetric key encryption/decryption on the given data.

- **GenerateDESKey**: Generates a single DES key (eight bytes) or a two key Triple DES key (sixteen bytes) from the on-card FIPS 186-2 pseudo random number generator
- **GenerateRandomNumber**: Creates a random number of the given size, using the FIPS 186-2 (Appendix 3.1) compliant pseudo random number generator.
- **RSAEncrypt:** Encrypts the given plaintext with the public RSA key in the given file. If the Format Type is RSAPKCS1v1_0 the given plaintext is formatted according to the given PKCS #1 version type before being encrypted.
- **RSADecrypt**: Decrypts the given ciphertext with the private RSA exchange key (or exchange/signature key) in the given file.
- **RSAGenerateKey**: Generates an RSA key pair into the given private key and public key files.
- **RSASign**: Performs RSA PKCS #1 (version 1.5) signature on the given data with the private RSA signature key (or exchange/signature key) in the given file.
- **RSAVerify:** Performs an RSA verify operation on the given signature and hash with the public RSA key in the given file. If the Format Type is RSAPKCS1, the result is unformatted according to PKCS #1 before being compared to the hash.
- **SHA1**: Initiates, continues, or completes a SHA-1 hash of the given data.

Secure storage services

- **CreateFile**: Creates an empty file of the given type.
- **DeleteFile**: Deletes references to a given file.
- **ReadBinary**: Returns the requested amount of data (at the given offset) from the active file.
- **Recycle**: Deletes all files and zeroes all allocated buffer space.
- **SelectFile**: Makes the given file the active file, to be used by subsequent commands.
- **UpdateBinary**: Overwrites data in the active EF, at the given offset, with data given in the command.
- **WriteBinary**: Writes the given data (at the given offset) to the active file.

General Services

- **ChangeConfiguration**: Updates the current configuration data in the configuration file.
- **GetStatus**: Returns the current status of the card. The remaining file space gives the number of bytes available for creating new files (16-bit number, MSB first). The configuration data has the same format as the Configuration File.
- **Format**: During initialization of the card, this command is required to initialize containers storage system of a card and sets the initial SO PIN . This is a pre-issuance command and is not available once the card has been issued to the end user (Security Officer and User roles).
- **InternalAuthenticate**: This command computes authentication data, using the challenge data sent in the command and the key in the currently selected file.

6.3. Authentication

6.3.1. User Authentication

During fingerprint enrollment, one of four ways to authenticate to User State on the token can be selected:

- Fingerprint -or- PIN: either fingerprint or PIN authentication can be used.
- Fingerprint only: the PIN may not be used to authenticate to User State.
- PIN only: a fingerprint cannot be used to authenticate to User State.
- Fingerprint -and- PIN: both fingerprint and PIN entry are required consecutively to authenticate to User State.

The cardholder must execute the Verify command with the correct PIN (an 8-20 byte secret) or PBVerify with the correct fingerprint verification data to transition the card state to User authenticated state. In this state, the User can access services provided by DKCCOS that require User authentication.

To discourage an attacker from guessing the SO PIN, DKCCOS maintains a count of the number of consecutive Verify (User) and UpdatePIN (User) attempts remaining (the limit is established by configuration file) due to an incorrect PIN. This count is maintained in nonvolatile memory. When this count reaches zero, the User PIN will be blocked, the uninitialized PIN error response will be returned to the host, and the card will enter the Error state. If allowed by the configuration file, the SO may update the User PIN in order to re-enable the User PIN.

As for fingerprint, the number of allowed bad fingerprint authentication attempts is set when the fingerprint template is enrolled on the smart card. It can have one of the following values:

- 0: There is no limit to the number of bad attempts.
- 1: Only one bad fingerprint attempt is allowed. Two bad attempts will lock the fingerprint template.
- 2 through 63: 2 though 63 consecutive bad attempts are allowed.

The relatively high maximum limit (63) compared to the maximum smart card bad PIN limit (15) reflects the reality that a user is more likely to inadvertently mismatch on a fingerprint than a PIN.

The count of bad fingerprint authentication attempts is kept internally on the smart card, and is independent of the internal bad PIN counter. It is incremented with every bad fingerprint logon attempt, regardless of which fingerprint is used. Switching fingers does not clear the count. The count of bad fingerprint authentication attempts is cleared with every successful fingerprint logon.

When the internal count of bad fingerprint authentication attempts exceeds the maximum value set at enrollment, logon via the fingerprint template is locked. Once locked, no

fingerprint can be used to log on until a new fingerprint template is enrolled onto the smart card.

A flag can be set during enrollment to lock this parameter. If locked, the maximum bad fingerprint limit is fixed and cannot be changed during future enrollments. Once the lock flag is set, it cannot be cleared during re-enrollment. It can only be cleared by deleting the entire fingerprint template via either the Recycle command or by deleting the cryptoki directory.

6.3.2. Security Officer Authentication

The Security Officer must execute the Verify command with the correct PIN to transition the card state to SO authenticated state. In this state the SO can access services provided by the DKCCOS that require SO authentication.

Additionally to discourage an attacker from guessing the SO PIN, DKCCOS maintains a count of the number of consecutive Verify (SO) and UpdatePIN (SO) attempts remaining (the limit is established by configuration file) due to an incorrect PIN. This count is maintained in nonvolatile memory. When this count zero, the SO PIN is disabled, the command response is Uninitialized PIN and the card enters the Error state. The SO PIN can be reset to the default SO PIN by issuing the Recycle APDU that restores the backup SO PIN to the SO PIN file. Every successful Verify (SO) and UpdatePIN (SO) will reset the failed attempts count to zero.

The following table summarizes the type of authentication and strength of mechanism for each role.

Role	Authentication	Strength of Mechanism
User	PIN	8-20 bytes
SO	PIN	8-20 bytes
User	Fingerprint	1 in 1,000,000

The PINs can be considered Security Relevant Data Items (SRDI). However the module does not store any actual PINs in EEPROM. Only the SHA-1 hash of PIN value is stored in the User or SO PIN file, which cannot be read or written except by using the UpdatePIN command by an authenticated user that writes the hash of the new PIN to the PIN file. There is also a default SO PIN hash value stored in ROM, which is used to authenticate to the card for the first time.

The following table summarizes the SRDIs available to each role.

Role	SRDI	Type of access
Idle	Configuration file settings	Read
User	Internally generated secret and private keys	Usage
	User loaded secret and private keys	Usage/Write
	Internally generated public keys	Usage/Read/Write
	User loaded public keys	Usage/Read/Write
	Configuration file settings	Read
	User PIN	Update
	Private Fingerprint Template	Write

SO	Configuration file settings User PIN SO PIN Private Fingerprint Template	Read/Update Update Update Write

6.4. Configuration file

The 330G3 has a configuration file (with file ID FF00). The configuration settings determine the overall security rules employed by the module. Only the SO is allowed to modify the configuration file settings by calling the ChangeConfiguration APDU. The first 20 bytes of data in this command contain the new configuration data, which is used to update the data in the configuration file. The second 20 bytes of data in this command contain the new configuration mask, which is used to update the mask in the configuration file:

The configuration file has two 20-byte sections - the first is the configuration data, and the second is a bit mask that specifies which bits of the first section may be changed by the SO. A '1' in a particular mask location indicates that the SO may change the corresponding bit in the configuration data, while a '0' indicates that the corresponding bit in the configuration data can not be changed.

The configuration data of the file has the following format

Data	Length in bytes
RSA Exchange enable/size	1
RSA Signature enable/size	1
DSA enable/size	1
DH enable/size	2
Public key formatting	1
Crypto command enable	2
Symmetric key enable	1
SO Authentication	1
User Authentication	1
EXF enable	2
Idle Allow	2
Export Control	1
Hardware Control	1
RFU	3
Configuration Mask	20

The meaning of each type of byte settings is explained in the tables below:
The top nibbles of the RSA enable/size bytes indicates the maximum allowable modulus size, The bottom nibble of the RSA enable/size bytes is defined below:

Bottom Nibble				
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Meaning**
0	x	x	x	No user readable private keys
1	x	x	x	User readable private keys allowed
x	0	x	x	User cannot generate keys
x	1	x	x	User may generate keys
x	x	0	x	SO cannot load keys

x	x	1	x	SO may load keys
x	x	x	0	User cannot load keys
x	x	x	1	User may load keys

The second byte of the DH enable/size field is the size of the maximum allowable private exponent, in bytes

Public Key Formatting	
Bit 7	RFU
Bit6	0 - DH/DSA keys are not allowed 1 - DH/DSA keys are allowed
Bit 5	0 - DH Wrapped Key Disabled 1 - DH Wrapped Key Enabled
Bit 4	0 - DH Raw Disabled 1 - DH Raw Enabled
Bit 3	RFU
Bit 2	0 - RSA Wrapped Key Disabled 1 - RSA Wrapped Key Enabled
Bit 1	0 - RSA Raw Disabled 1 - RSA Raw Enabled
Bit 0	0 - RSA PKCS1 Disabled 1 - RSA PKCS1 Enabled

The two Crypto Command Enable bytes control the availability of the crypto-related commands. A zero indicates the command is not available, while a one indicates that the command is available.

Crypto Command Enable - First Byte	
Bit 7	Crypt
Bit 6	DH/DSAGenerateKey
Bit 5	DHKeyAgreement
Bit 4	DSASign
Bit 3	DSAVerify
Bit 2	GenerateDESKey
Bit 1	RSADecrypt
Bit 0	RSAEncrypt

Crypto Command Enable - Second Byte	
Bit 7	RSAGenerateKey
Bit 6	RSASign
Bit 5	RSAVerify
Bits 4:0	RFU

The upper nibble in the Symmetric Key Enable byte controls the use of single and Triple DES (Crypt and GenerateDESKey commands) and the use of the ECB and CBC DES modes (Crypt command). The bottom nibble has the same definition as the bottom nibble of the public key enable/size bytes.

Symmetric Key Enable	
Bit 7	0 - Two Key Triple DES Disabled 1 - Two Key Triple DES Enabled
Bit 6	0 - Single DES Disabled 1 - Single DES Enabled
Bit 5	0 - ECB mode Disabled 1 - ECB mode Enabled
Bit 4	0 - CBC mode Disabled 1 - CBC mode Enabled

SO Authentication	
Bits 7:6	RFU
Bit 5	Update PIN (without current PIN)
Bit 4	0 - Recycle command writes hash of default PIN phrase to SO PIN file 1 - SO PIN data is not changed
Bits 3-0	Maximum consecutive Verify(SO) failures

User Authentication	
Bit 7	RFU
Bit 6	PIN file may be written in Idle state
Bit 5	Update PIN (without current PIN)
Bit 4	0 - SO may not update User PIN 1 - SO may update User PIN
Bits 3:0	Maximum consecutive Verify(User) failures

The SO Authentication and User Authentication fields are followed by a two-byte EXF field. The two-byte EXF field is interpreted as a 16-bit (MSB first) count that is one less than the minimum acceptable EXF (EXF's are identified with a 16 bit count). A value of zero indicates all EXF's will be accepted. A value of 0xFFFF indicates no EXF's will be accepted.

The two Idle Allow bytes determine if certain commands are allowed in the Idle state (a zero indicates the command is not allowed in Idle, while a one indicates the command is allowed in Idle):

Idle Allow - First Byte	
Bit 7	CreateFile
Bit 6	Crypt
Bit 5	DH/DSAGenerateKey
Bit 4	DHKeyAgreement
Bit 3	DSASign
Bit 2	DSAVerify
Bit 1	GenerateDESKey
Bit 0	RSADecrypt

Idle Allow - Second Byte	
Bit 7	RSAEncrypt
Bit 6	RSAGenerateKey
Bit 5	RSASign
Bit 4	RSAVerify
Bits 3:0	RFU

The definition of the Export Control byte is controlled by Datakey. DKCCOS does not use this byte.

The Hardware Control byte is defined as follows:

Hardware Control	
Bit 7	CPU Clock Doubling Enable
Bits 6:0	RFU

When the CPU Clock Doubling Enable bit is changed in the configuration file, it does not take effect until the next power cycle or reset.

The three bytes of RFU space are reserved for use by EXF's. All bits and bytes labeled RFU are by default set to zero.

6.5. FIPS mode:

For FIPS Mode, the configuration file should have the following values:

Field	Value	Meaning
RSA Exchange enable/size	0x77	RSA Exchange enabled to 2048 bits
RSA Signature enable/size	0x77	RSA Signature enabled to 2048 bits
DSA enable/size	0x37	DSA enabled to 1024 bits
DH enable/size	0x77, 0xC0	DH Enabled to 2048 bits
Public key formatting	0x01	Only PKCS #1 formatting enable for RSA
Crypto command enable	0xFF, 0xFF	All commands are enabled
Symmetric key enable	0xF7	DES, Triple DES enabled
SO Authentication	0x2A	Recycle replaces SO PIN with default value, SO may not UpdatePIN without current PIN SO PIN count is 10
User Authentication	0x6A	SO may update User PIN, User may not UpdatePIN without current PIN User PIN count is 10
EXF enable	0xFF, 0xFF	EXF's disabled
Idle Allow	0x00, 0x00	Commands not allowed in Idle state
Export Control	0x00	RFU
Hardware Control	0xFF	CPU clock doubling disabled
RFU	0x00, 0x00, 0x00	RFU

It is the SO responsibility to ensure the bit settings are set as per the table above when module is in FIPS Approved mode. Also the bit-mask in the configuration file must be set so that the configuration settings cannot be changed once set. An operator can use the GetStatus command to determine whether the module is in the Approved FIPS 140-2 mode by matching the configuration file settings to the values mentioned in the table above.

For biometric verification, in order to be in a FIPS Approved mode, the SO or User enrolling the fingerprint template on the card (for authentication of Users) should ensure that Fingerprint Authentication Mode is set to one of the following:

1) Fingerprint and PIN
2) PIN only
3) Fingerprint only: The False Acceptance Rate (FAR) value must be set to 1 in 1,000,000 during enrollment using the Datakey CIP utilities. The number of allowed Bad Fingerprint authentication attempts must be set to 10.
4) Fingerprint or PIN: The FAR value in this case must be set to 1 in 1,000,000 during enrollment using the Datakey CIP utilities. The number of allowed Bad Fingerprint authentication attempts must be set to 10.

7. Finite State Model

The state diagram for the 330G3 complies with, and in some instances is dictated by, the ISO 7816 standards for smart cards. The card's state diagram consists of several states as indicated in the DKCCOS ICD document.
The loading of the G3 EXF and the settings of the default SO PIN takes place in the pre-issuance phase at the Datakey facility using an HSM. The configuration file is then changed to disallow further loading of EXFs.

8. Physical Security

The Datakey Model 330G3 Smart Card is a single-chip cryptographic module. It is designed to meet FIPS 140-2 level 3 requirements for physical security.

The 330G3 IC is a production quality IC. It meets commercial-grade specifications for power, temperature, reliability, and shock / vibration. It uses standard passivation techniques for the entire chip.

In addition to the passivation material, a coating covers the chip. The epoxy material fills a reservoir constructed around the die and wire bonds. The epoxy used on the back of the chip is resistant to solvents that are commonly available.

9. Cryptographic Key Management

DKCCOS supports the use of public key cryptography in two primary functions - digital signatures and key management (exchange / agreement) for symmetric encryption keys. While a single RSA key may be used for both functions, best practices recommend using separate key pairs.

Key generation:
The DKCCOS generates the following types of keys:

- 8 byte single DES
- 16 byte Triple DES
- 512- 2048-bit RSA public and private keys
- 512- 1024-bit DSA public and private keys

The module uses the FIPS 186-2 Appendix 3.1 and 3.3 compliant (SHA-1 based) PRNG to generate keys. A non-Approved non-deterministic hardware RNG is used to provide additional entropy for the seed value used in the PRNG process. The 20-byte seed value is read from the User Entropy File stored on the card.

Only an authenticated User can generate keys on the card. No internally generated secret or private keys can be read, written or updated in the FIPS mode. The DKCCOS employs checks to make sure that the key generation functions GenerateDESKey, DH/DSAGenerateKey and RSAGenerateKey will not write to key files that are writeable/updateable/readable by anyone (including the User himself).

Key entry and output:

The card allows only an authenticated User to perform cryptographic functions in FIPS mode. Thus only the operator owning the keys can use them for cryptographic purposes. . All secret and private key file security nibbles are checked during cryptographic operations to ensure that they are not updateable or readable by anyone. Secret and private keys may be loaded in key files and are protected by the security nibbles set by the User. Internally generated keys cannot be read or modified.

The user may load keys in key files in plaintext form. Keys can be protected against unauthorized modification, substitution and disclosure by setting the appropriate key file security nibbles.

Key Storage:
The keys are stored in plaintext form in the card file system in a key file in EEPROM. The User may also load keys in appropriate key files. These keys are also stored in plaintext. As seen from Section 7 the User can set file permissions using three bytes of security nibbles during file object creation to allow only the authenticated Users to read, modify the public key value.

Key Zeroization:

A key file can be deleted/zeroized by issuing the DeleteFile command. Additionally the Recycle command can be used to destroy all files including all module keys and PIN files.

10. Cryptographic algorithms:

The module only supports the following FIPS-approved algorithms:

- DES[2] (ECB, CBC modes) (Cert. #88)
- Triple DES (ECB, CBC modes-2key Triple DES) (Cert. #236)
- SHA-1 (byte-oriented) (DSA/SHA Cert. #35)
- DSA (512-1024 bit modulus) (DSA/SHA Cert. #35)
- RSA (512-2048 bit modulus) (PKCS#1, vendor-affirmed)

Additionally the module provides a FIPS 186-2 Appendix 3.1 compliant PRNG.

The module also provides the following non-Approved algorithms:

- RSA encryption/decryption for key exchange
- Diffie Hellman key agreement

[2] DES should only be used in legacy systems

11. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The Model 330J module conforms to the EMI/EMC requirements specified in FCC Part 15, Subpart J, Class B (i.e., for home use).

The Model 330J cryptographic smartcard is a single chip and is a passive device. The reader/writer device shall supply operating power and oscillation frequency to the smartcard.

Refer to FCC certificate included in this documentation package labeled

TEST RESULT SUMMARY
FCC PART 15 SUBPART B
Class A Limit
Test report number M6095 and M6095.1

Lab Name: TUV PRODUCT SERVICES Inc.
Location: Taylors Falls MN USA

12. Self Tests

The POST self test functions test the following critical systems: the RAM (internal and external), the FAME coprocessor, the file system integrity, and the checksum on the contents of the ROM. If any of these tests fail, the card will enter the Error state. The cause of the error can be determined with the GetStatus command.

Known answer tests are also performed for the following cryptographic algorithms: SHA-1, DES, Triple DES, DSA Sign, DSA Verify, RSADecrypt, RSAEncrypt, RSASign, and RSA Verify. If any of these tests fail, the card will enter the Error state.

If a validated EXF has been loaded, its checksum is checked against a value stored in EEPROM. The EXF will also be examined to see if any of its functions are enabled as POST functions. If any EXF POST functions are found, they will be executed and any failure will cause the card to enter the Error state.

The G3 EXF Power On Self Test (POST) function tests the following critical systems:
Known Answer Test (KAT) of the PRNG.

The pairwise consistency test is performed each time a key pair is generated. If the key pair generated is used for Exchange only then only an RSADecrypt/RSAEncrypt is performed using the key pair. If the key pair generated is to be used only for digital signatures then simply an RSASign/RSAVerify is performed or if it is a DSA then a DSASign/DSAVerify is performed. If any of these tests fail, the card will transition to error state. No data is output from the module.

The **Continuous random number generator** test is performed each time a block of random data is requested from the FIPS 186-2 compliant PRNG.

If this test fails, the card will enter the Error state. The cause of the error can be determined with the GetStatus command. The operator must then issue the Recycle command to bring the module back to an operational state.

The module also performs a firmware load test by doing an RSA signature verification when an EXF is loaded on the card. In case this test fails the EXF is invalidated and removed from EEPROM

13. Security Guidance

13.1 Development Errors and Oversights

The primary object of concern relative to development errors and oversights is the DKCCOS source code, and to a lesser extent the supporting software in the IT environment. The DKCCOS source code is the primary concern because once the code has been embedded into the ROM of the smart card, it is not easy to change.

The steps taken during development of DKCCOS to minimize errors and oversights were:

- use of cryptographic experts for both high-level and code design,
- frequent discussion among design group members of functional and performance objectives / specifications,
- frequent design reviews of design approaches,
- special code reviews with third party cryptographic experts,
- development of extensive test scripts,
- use of emulation equipment to evaluate OS code with the target IC,
- extensive testing of pilot production ICs with OS embedded in ROM, and
- extensive evaluation in customer trials.

There are currently no known errors or faults in the DKCCOS Version 2 code.

13.2 User Guidance

When or before the card is issued, the end-user should be made aware that the card is an extension of the user's ID and is capable of generating a digital signature for the user, which is as valid and legal as a written signature on a paper document. For this reason the user should also understand that he /she should keep the card on their person or under lock and key when not in use, and to protect their secret pass phrase from observation when logging on.

At the time the card is issued, an initial user pass phrase is in the card. The issuer should be urged to immediately change the initial pass phrase to one which the user can easily remember but one which others cannot easily guess.

These things seem to be simple enough to remember, but in fact require some personal discipline on the part of the user. *Lapses in this discipline can lead to the use of an authorized user's card by an unauthorized user.*

Users must also ensure not to use RSA encryption and decryption for protecting data in order to be an Approved mode. RSA encrypt/decrypt should only be used for key exchange.

Users must ensure that a new seed value is written to the User Entropy File before invoking key generation.

13.3 Protection against Unauthorized Users

If an unauthorized user can gain access to an authorized user's card **and** pass phrase, the imposter can act in every sense with all the capability as the true owner of the card. *This usurpation of the user's identity, at best would be very embarrassing, and at worst extremely costly, to the authorized user.*

Other than the above, there are no known vulnerabilities that can come from the end-user's lack of application knowledge or carelessness.

13.4 Security Officer Guidance

The issuing organization's senior security administrator may be responsible for ensuring that cards are issued with a card configuration file as well as subject / object / operation attributes in accordance with organization security policy, and the administrator is normally assumed to be trustworthy in fulfilling this responsibility.

If this assumption is valid and the administrator is well-trained and competent, there are no known vulnerabilities added as a result of the card issuing process.

However, if the security administrator is motivated maliciously, compromise of application security is possible.

There are a number of card configuration bits that are used to implement elements of the organization / application security policy. Some examples of malicious intent would be:

- The policy may state that the administrator cannot unlock 'locked user cards'. If the administrator sets this bit contrary to policy, he / she can collect the 'locked cards', unlock and reissue them to unauthorized users, or use them personally for his own gain.

- The policy may state that cryptographic commands may not be processed from the Idle State. Setting this bit contrary to policy would allow unauthorized user with a card but no pass phrase to perform crypto operations.

To protect against such vulnerabilities and to ensure that the module is in an Approved mode of operation as defined in FIPS 140-2, the Security Officer must follow the following rules:

1) The Security Officer must set the configuration settings as per the table in Section 6.5. Also the bit-mask in the configuration file must be set so that the configuration settings cannot be changed once set
2) The Security Officer or user enrolling the fingerprint template on the card for biometric verification must ensure that the FIPS 140-2 strength of authentication requirements are met by configuring the biometric enrollment settings correctly (See Page 25).
3) The Security Officer must change the default SO PIN as soon as the card is in possession of the SO and must not communicate his/her PIN to any entity.
4) When the Recycle command is issued the EXF file is deleted. This command must only be used when the card is destroyed.
5) The SO must unblock User PIN only for legitimate Users.

13.5 Potential Loss of Secure State

There are no known outsider scenarios or act-of-nature failures that leave the card in an insecure state, in which further attacks could more easily compromise the system security.