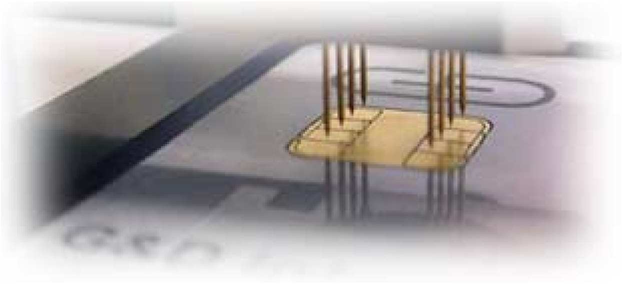
Giesecke & Devrient

STARCOS SPK 2.4 in ID-1Module

(Software Version - CP5WxSPKI24-01-3-S_V0330,
Hardware Version - P8WE 5032 M5.1)



FIPS 140-2 Non-Proprietary Security Policy

Level 2 Validation

Version 1.01
June 2002

Table of Contents

1. Introduction

1.1. Purpose

This is a non-proprietary Cryptographic Module Security Policy for Giesecke & Devrient (G&D) Smart Card Chip Operating System Standard Version with Public Key Extension 2.4 (STARCOS SPK 2.4) in ID-1 module. This security policy describes how STARCOS SPK 2.4 in ID-1 module meets the security requirements of FIPS 140-2 and how to run STARCOS SPK 2.4 in ID-1 module in a secure FIPS 140-2 approved mode. This policy was prepared as part of the Level 2 FIPS 140-2 validation of STARCOS SPK 2.4 in ID-1 module.

Throughout this document, the STARCOS SPK 2.4 in ID-1 module is referred to as the chip, the STARCOS SPK 2.4, and the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 — *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the NIST website at www.nist.gov/cmvp.

1.2. References

This document deals only with operations and capabilities of STARCOS SPK 2.4 in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on STARCOS SPK 2.4 from the following sources:

- Overview information of Giesecke & Devrient products and services can be found at: www.gdai.com
- For answers to technical or sales related questions, please refer to the contacts listed on the Giesecke & Devrient website at www.gdai.com

1.3. Document Organization

The Security Policy document is one document in a complete FIPS 140-2 Submission Package. In addition to this document, the complete Submission Package contains:

- Vendor Evidence document
- Finite State Machine Model
- G&D STARCOS SPK 2.1 and 2.4 Reference Manuals
- Other supporting documentation as additional references

This Security Policy and the other certification submission documentation were produced by Corsec Security, Inc. under contract to Giesecke & Devrient. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Submission Documentation is proprietary to Giesecke & Devrient and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Giesecke & Devrient.

2. STARCOS SPK 2.4

2.1 Overview

STARCOS was first developed in 1987 and has been evolving ever since. It is the market proven smart card operating system and workhorse for G&D smart card applications. STARCOS SPK is part of the STARCOS product line.

STARCOS SPK 2.4 was developed by G&D. Providing a complete set of International Organization for Standardization (ISO), Europay Mastercard and Visa (EMV) and proprietary enhanced commands, the STARCOS SPK 2.4 incorporates standards-based functionality along with its own optimized command set.

STARCOS SPK 2.4 for Public Key applications extends the STARCOS command set by providing support for public key cryptography and digital signatures in accordance with ISO and German Institute for Standardization (DIN) standards. It features a crypto co-processor for optimal RSA performance as well as Triple-DES co-processor for symmetric Triple-DES cryptography resistant to all known attacks.

STARCOS SPK 2.4 is based on the P8WE5032 smart card controller.

Some highlighted features of STARCOS SPK2.4 are:

- Commands are compatible to ISO 7816-8
- SHA-1 Hash algorithm
- Delivery PIN mechanism
- PKCS#11 client server authentication support
- PKCS#1 padding
- RSA up to 1024 bit for:
    - Digital signature generation and verification
    - Key generation
    - Asymmetric authentication including session key establishment for secure messaging
- DES (for legacy systems only) and Triple-DES Encryption
- DES-MAC and Triple-DES-MAC

2.2 Physical Security with Well-Defined Interfaces

STARCOS SPK 2.4 is defined as a single-chip module for FIPS 140-2 purposes. The physical form of the module is a single chip with attached faceplate embedded in a plastic card (25mm by 15mm). It is intended to meet overall FIPS 140-2 level 2 requirements (see Table 1).

Section	Section Title	Level
1	Cryptographic Module Specification	2
2	Cryptographic Module Ports and Interfaces	2
3	Roles, Services, and Authentication	2
4	Finite State Model	2
5	Physical Security	2
6	Operational Environment	N/A
7	Cryptographic Key Management	2
8	Electromagnetic Interference/Electromagnetic Compatibility	2
9	Self-tests	2
10	Design Assurance	2
11	Mitigation of Other Attacks	2

Table 1 – Intended Level Per FIPS 140-2 Section

The module is composed of a plastic card and a single chip micro-controller, coated in epoxy, with an attached faceplate. The chip contains the processor, Read Only Memory (ROM - 32 kilobytes), Random Access Memory (RAM - 256+2048 bytes), Electrically Erasable Programmable ROM (EEPROM - 32 kilobytes), co-processors, input/output (I/O), and timers. The power interface accepts voltages in the range of +5V +/-10%.

The smart card controller provides a number of security features, including voltage, temperature, and clock sensors. The chip is embedded in epoxy, which completely encapsulates the whole Integrated Circuit (IC). Only micro-wires connecting to the faceplate penetrate the epoxy, connecting to the faceplate interface of the module. Attempts to tamper with the module result in damage to the epoxy, the plastic card, or the metal faceplate (scratches, chips, dents, etc.).

The module has been tested for and meets applicable Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined in Subpart A of FCC Part 15.

There is only one physical interface to the module, the faceplate, which contains eight contacts, pinned as defined in ISO 7816-2. The eight contacts of the faceplate can be mapped to the FIPS 140-2 logical interfaces as depicted in Table 2.

Contact	Function	FIPS 140-2 Logical Interface
C1	Power supply	Power Interface
C2	Reset	Control Input Interface
C3	Clock	Control Input Interface
C4	Not connected	N/A
C5	Ground	Power Interface
C6	Not connected	N/A
C7	Input/Output for serial data	Data Input Interface, Data Output Interface, Control Input Interface, Status Output Interface
C8	Not connected	N/A

Table 2 - Contact to Function Mapping

As described in ISO 7816-2, when the module is first inserted into the reader (also referred to as the terminal), an RST is transmitted to contact C2. Power is applied via contact C1; C7 is set to reception mode; and the external clock is established via contact C3. The I/O interface (C7) has reception and transmission modes. The smart card reader sends commands to the module and the module transmits responses.

STARCOS SPK 2.4 is only capable of operating in response to commands sent from the reader in what is called a command-response pair. The reader sends an Application Protocol Data Unit (APDU) to the module and module responds with an APDU.

The APDU sent by the reader consists of a header and a body. The header contains a class byte differentiating between ISO defined command and private commands, an instruction byte containing the command code, and parameters relating to the command. The body contains any data that is needed for the command and, if necessary, the length of the expected data.

The response APDU transmitted by the module consists of a body and a trailer. The body contains any data that is returned in response to the command and the trailer contains the status message.

2.3 Software Security

The firmware for the STARCOS SPK 2.4 is written entirely in machine language specific to the underlying chip. It is loaded onto the module during manufacturing and does not allow for modification. An Error Detection Code (EDC) is calculated over the firmware during this installation and is checked at each power up. Attempts to modify the firmware require direct access to the IC and are prevented by the physical security described in section 2.2 above.

2.3.1 Command Structure

STARCOS SPK 2.4 provides a well-defined, static set of commands. A smart card reader sends these commands to the module and then responses are transmitted from the module to the reader. Only these commands are available to an operator, and only the faceplate interface may be used to access the module's functionality.

Internally, the receive/respond mechanism of the module is as follows (depicted in Figure 1). The transmission manager receives a command (along with the associated parameters for that command) from the reader. If this communication is encrypted or has a Message Authentication Code (MAC) calculated over it, then it is passed to the secure messaging component for processing. When accessing sensitive data, secure messaging should be enabled (see section 3 for more details). Next, the command is passed to the command interpreter and processed. File access operations are handed off to the file manager, which handles file access control and data operations. When processing of the command is completed, a response is prepared. If the response communication is to be encrypted or a MAC needs to be calculated over it, then the communication is passed through the secure messaging component for processing. Finally, the transmission manager transmits the response to the reader.

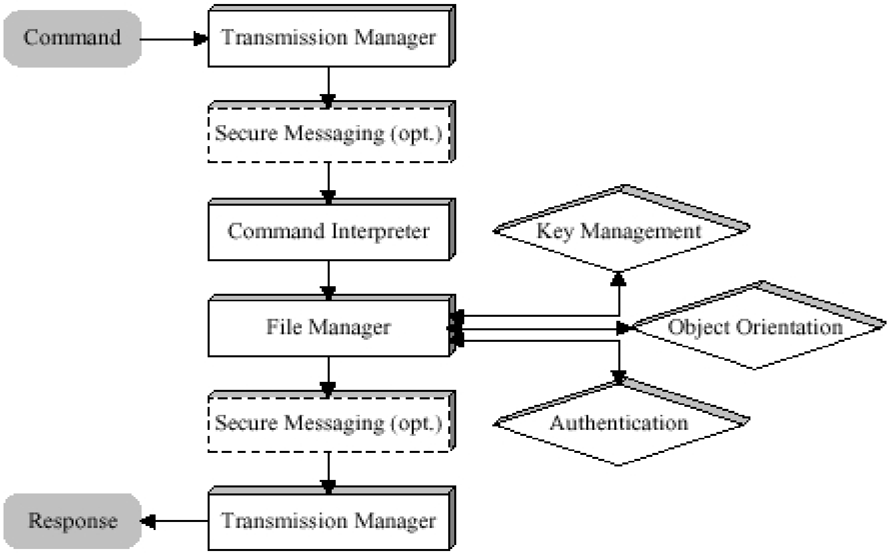


Figure 1 - Command Processing

2.3.2 File Structure

The file structure of STARCOS SPK 2.4 is based upon a root directory with sub-directories and files. The master file (MF) is the root directory of the file system and dedicated files (DFs) are the sub-directories of the MF. The MF and DFs can contain elementary files (EFs), Internal Secret Files (ISFs), and Internal Public Files (IPFs). Figure 2 depicts the file structure and Table 3 describes the files types supported by the module.

File Type	Function	Notes
Master File (MF)	Constitutes the root file system (i.e., is the root directory).	The MF has one ISF and one IPF under it.
Dedicated File (DF)	Stores all data for a particular branch of the file system (i.e., is a directory)	Each DF has one ISF and one IPF under it. A DF cannot contain another DF (i.e., the directory structure only goes one level deep).
Elementary File (EF)	Stores actual data	
Internal Secret File (ISF)	Stores secret/private keys and PINs	Special type of EF. It is implicitly created when the MF or DFs are created.
Internal Public File (IPF)	Stores public keys	Special type of EF

Table 3 - File Types

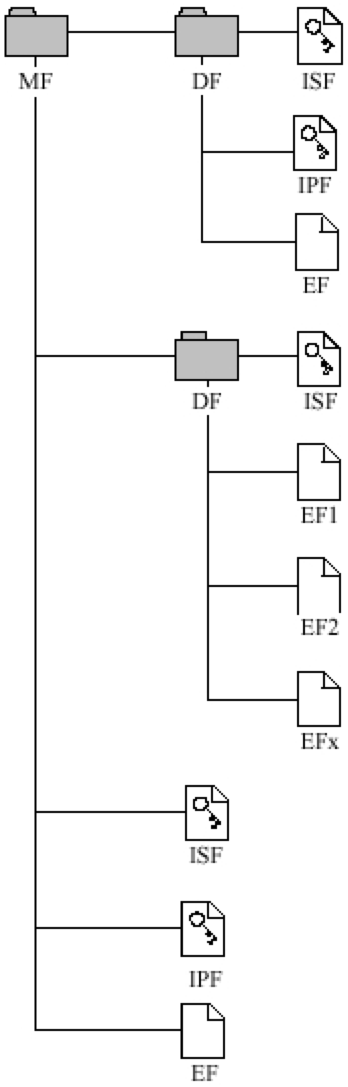


Figure 2 - File Hierarchy

2.3.3 Authentication Mechanisms

Authentication is performed using a variety of mechanisms based on keys. Additionally, PINs may be used to authenticate to the module. The key-based authentication schemes used by the module are described in Table 4.

Authentication Type	Description	Key Type Used	FIPS 140-2 Use
Internal Authentication	The module authenticates itself to the terminal.	Symmetric (DES/Triple-DES) or Asymmetric (RSA key pair)	N/A (except as part of mutual authentication described below)
External Authentication	The terminal authenticates itself to the module.	Symmetric or Asymmetric	Yes – Role-based of User
Mutual Authentication	The module authenticates itself to the terminal and vice versa.	Symmetric or Asymmetric	Yes – Role-based of User
Master-Slave Authentication	The module authenticates itself to another module (i.e., module to module authentication with one card functioning as the master and the other as the slave).	Symmetric Only	Yes – Role-based of User
Client-Server Authentication (PKCS#11-based)	The module authenticates itself as a client to a server.	Asymmetric Only	N/A

Table 4 - Key-Based Authentication Mechanisms

The following table provides rough estimates of the strengths of the module's authentication mechanisms.

Authentication Type	Strength
PINs	Dependent on length and content. Assuming length is equal to 8 and only integers between 0-9 were used without repetition, the probability of randomly guessing the correct sequence is 1 in 1,814,400.
Internal Authentication	This mechanism is as strong as either the RSA algorithm using a 512-1024 bit key pair, the DES algorithm with a 56 bit key, or the Triple-DES algorithm with a 56 or 112 bit key.
External Authentication	This mechanism is as strong as either the RSA algorithm using a 512-1024 bit key pair, the DES algorithm with a 56 bit key, or the Triple-DES algorithm with a 56 or 112 bit key.
Mutual Authentication	This mechanism is as strong as either the RSA algorithm using a 512-1024 bit key pair, the DES algorithm with a 56 bit key, or the Triple-DES algorithm with a 56 or 112 bit key.
Master-Slave Authentication	This mechanism is as strong as either the DES algorithm with a 56 bit key or the Triple-DES algorithm with a 56 or 112 bit key.
Client-Server Authentication (PKCS#11-based)	This mechanism is as strong as the RSA algorithm using a 512-1024 bit key pair.

Table 5 - Estimated Strength of Authentication Mechanisms

2.3.4 Communication Modes

Communications between the module and the reader can be encrypted or secured against tampering via a MAC. This secure communication is referred to as secure messaging and it comes in two forms.

- Authentic mode – a DES-MAC or Triple-DES-MAC is calculated over the packets in order to ensure they have not been modified.

- Combined mode – the packets are encrypted using DES or Triple-DES and then a DES-MAC or Triple-DES-MAC is calculated over the encrypted packet.

During either asymmetric or symmetric authentication, a key establishment is performed if secure messaging is enabled.

Access control mechanisms based on a state machine are configured by the Crypto-Officer and enforced by the STARCOS SPK 2.4.

A complete description of all commands and response codes can be found in the G&D Reference manuals for the SPK 2.1 and 2.4. Descriptions of the roles supported by the module and the services available to those roles are contained below in this document.

2.3.5 Access Control

The access control functionality of the module is governed by an access control state machine. When certain information (such as a PIN) is entered, the module can switch states for a particular DF or the MF. File access can be controlled on a per state basis, allowing certain authenticated operators different-types of access to different files from different states. This is configured by specifying access conditions (ACs) for a file.

The MF has a global state machine that is always active. Each DF has its own access control state machine that is only active when the DF is selected. Switching from one DF to another resets the state of the DF.

There are 16 states available for each access control state machine, ranging from state 0 to state 15. EFs, DFs, ISFs, and IPFs all allow for configuration of access rights based on the state the MF or currently selected DF is in.

Access control state machine transitions only occur with authentication commands. During authentication, the module checks the current state of the MF or the currently selected DF to ensure that access to the key being used for authentication is permitted. If it is, the module checks to see if secure messaging is required and activates it if required. The authentication data is then verified and the state of the MF or currently selected DF is changed. If any of these checks fails, the authentication command fails and the state is not changed.

Given the command-response nature of the module, transitions between states only occur in response to commands received from the terminal. The module itself has an overall state machine, while the individual DFs and the MF have state machines as well.

Further detail on the access control state machine mechanisms can be found in the reference manual for the STARCOS SPK 2.1 and 2.4.

2.4 Roles and Services

STARCOS SPK 2.4 performs role-based authentication. Operators authenticate using either PINs or keys. Once authenticated, an operator is able to access functionality/data based on the permissions granted by that authentication (see section 2.3.5 for more information about access control).

The STARCOS SPK 2.4 supports two roles, the Crypto-Officer role and the User role. The Crypto-Officer role is used to initialize the module by creating a file structure on the module. The User role is utilized to access this file structure and to use the services provided by the module after initialization. The module does not implement a maintenance role.

The following table summarizes the commands provided by the module.

Role	Service	Description	Keys Accessed	Type of Key Access
User	CHANGE REFERENCE DATA	Change current PIN value.	PIN	Write
User	COMPUTE SIGNATURE	Calculate an RSA signature on a hash of some data.	RSA private key	Read
Crypto-Officer/User	CREATE	Create files and directories on the module.	-	
User	*CRYPT*	Perform DES or Triple-DES encryption/decryption/calculation of MAC.	DES or Triple-DES key	Read
User	*DECREASE*	Decrease value of data stored in EF (e.g. a counter).	-	
User	*ENCIPHER/DECIPHER*	Perform RSA encryption/decryption (for key wrapping only).	RSA private/public key	Read
User	*ERASE OBJECT FILE*	Erase a file with object structure.	-	
User	*EXCHANGE CHALLENGE*	Exchange random data between module and terminal.	-	
User	*EXTERNAL AUTHENTICATE*	Authenticate terminal to module or module to module.	RSA private/public key, DES or Triple-DES key	Read
Crypto-Officer/User	*GENERATE PUBLIC KEY PAIR*	Generate an RSA key pair.	RSA public/private key pair	Write
Crypto-Officer/User	*GET CARD DATA*	Read serial number, version number of operating system, or chip configuration data from module.	-	
User	*GET CHALLENGE*	Get random data from module.	-	
User	*GET DATA*	Read data from Tag Value Length (TLV) object of EFs with object structure.	-	
Crypto-Officer/User	*GET RESPONSE*	Get response data.	-	
User	*HASH*	Hash some data.	-	
User	*INCREASE*	Increase value of data stored in EF (e.g. a counter).	-	
User	*INTERNAL AUTHENTICATE*	Authenticate module to terminal or module to module.	RSA private/public key, DES or Triple-DES key	Read
User	*KEY STATUS*	Get number of unsuccessful	Key header	Read

		attempts at authenticating with a particular key.	only	
Crypto-Officer/User	*LOCK FILE*	Lock an EF so that read/write is not permitted.	-	
Crypto-Officer/User	*LOCK KEY*	Lock a key permanently so that read/write access is not permitted.	Key header only	Write
Crypto-Officer/User	*MANAGE SECURITY ENVIRONMENT*	Configure templates to be accessed by other commands.	-	
User	*MUTUAL AUTHENTICATE*	Authenticate terminal to module and module to terminal.	RSA private/public key, DES or Triple-DES key	Read
Crypto-Officer/User	*PUT DATA*	Write data from Tag Value Length (TLV) object of EFs with object structure.	-	
User	*READ BINARY*	Read data from an EF with transparent structure.	-	
User	*READ PUBLIC KEY*	Read public key or key record signature.	RSA public key	Read
User	*READ RECORD*	Read a record/element from an EF with linear fixed, cyclic, or compute structure.	-	
User	*RESET RETRY COUNTER*	Reset authentication failure counter on a PIN.	PIN header only	Write
Crypto-Officer/User	*REGISTER DF*	Allocates physical memory for a DF.	-	
Crypto-Officer/User	*SELECT FILE*	Activate an existing file or file level.	-	
Crypto-Officer/User	*TERMINATE CARD USAGE*	Zeroize all keys and make card inoperable.	All	Write (zeroize)
User	*UPDATE BINARY*	Write data to EF with transparent structure.	-	
User	*UPDATE RECORD*	Write a record/element from an EF with linear fixed, cyclic, or compute structure.	-	
User	*VERIFY*	Authenticate with a PIN.	PIN	Read
User	*VERIFY AND CHANGE*	Authenticate with a PIN and then change that PIN.	PIN	Read/Write
User	*VERIFY CERTIFICATE*	Verify signature on certificate for a particular public key.	RSA public key	Read
User	*VERIFY SIGNATURE*	Verify an RSA signature.	RSA private key	Read
Crypto-Officer/User	*WRITE KEY*	Create/import/overwrite a key.	RSA private key, DES or Triple-DES key	Write

Table 6 - Roles, Services, and Service Descriptions

2.4.1 Crypto-Officer Role

The Crypto-Officer role is responsible for initializing, distributing, and destroying the module. The Crypto-Officer will only use the module before it is initialized and, for initialization, the Crypto-Officer authenticates using a default PIN referred to as the Creation key. The Creation key is established by G&D during the manufacturing of the module.

During initialization of the module, the Crypto-Officer must first create the MF for the module. The creation of the MF is performed by issuing a “Create” command to the module with the proper parameters. One of these parameters is the Creation key established by manufacturing.

After the module verifies the Creation key is correct, it creates the MF. The parameters specified with the “Create” command include the access control permissions for the MF (i.e., establishing PINs, requiring encrypted communications, etc.). Although not required, after creation of the MF, the Crypto-Officer is able to create DFs, EFs, ISFs, and IPFs.

Once creation of the module’s initial file structure is finished, the Crypto-Officer ends the initialization process. During completion of the initialization, the Crypto-Officer has the ability to lock the file structure as it is or to allow other operators to create new files under the MF. If the Crypto-Officer locks the file structure, then the DFs and access control permissions are no longer modifiable. If the Crypto-Officer leaves the file structure open, then additional DFs may be added by the Crypto-Officer/User and the associated permissions can be configured.

The Crypto-Officer functionality includes:

- Initialization of the STARCOS SPK 2.4
    - Creation of the MF (with associated ISF)
        - Creation of keys and PINs
        - Configuration of access control
    - Creation of DFs (with associated ISFs), IPFs, and EFs
        - Creation of keys and PINs
        - Configuration of access control
- RSA key generation
- Distribution of the module
- Key zeroization
- Termination of the module

- Running the self-tests
- Querying status information

2.4.2 User Role

The User authenticates to the module using either a PIN or a key, as configured by the Crypto-Officer during initialization. Once authenticated, the User has access to commands and data as determined by the permissions associated with authenticating to that key or PIN.

Since the module uses a state machine-based access control mechanism, by authenticating to a particular PIN or key, the module potentially transitions to a new access control state. From this state, certain commands (such as writing to a particular EF) will be allowed while others will be denied. The states for access control are defined for the particular DF being accessed and for the MF.

The User will have varying levels of privilege depending on the access control governing the PIN or key used for authentication. These permissions are configured by Users or the Crypto-Officer, depending on how the initial file structure was created. For example, if the Crypto-Officer left the file structure open for the addition of DF's, then a User creating a DF would also be able to configure the permissions for that DF.

The User functionality includes:

- Reading, writing, and the deleting the contents of EFs
- Reading and writing to the contents of ISFs and IPFs
- Accessing cryptographic functionality (authentication, secure messaging, encryption/decryption, digital signatures, hashing, MACs)
- Creation of DFs (with associated ISFs), IPFs, and EFs
    - Creation of keys and PINs
    - Configuration of access control
- RSA key generation
- Key zeroization
- Termination of the module
- Running the self-tests

- Querying status information

2.5 Cryptographic Key Management

The following table summarizes the module's critical security parameters (CSPs):

Key	Key type	Storage	Use
Creation key	PIN	Non-volatile memory	Authenticate Crypto-Officer during initialization of the module.
Session keys	56 bits DES keys, 56 bit or 112 bit Triple-DES keys	Volatile memory	Encrypt communication between terminal and module and calculate MACs on communications.
User asymmetric keys	512 bit - 1024 bit (multiple of 8) RSA keys	Non-volatile memory	Digital signing and key establishment.
User symmetric keys	56 bit DES keys, 56 or 112 bits Triple-DES keys	Non-volatile memory	Encrypt/decrypt data, MACs, and authentication.
PINS	PIN	Non-volatile memory	Authentication.
PUK	PIN	Non-volatile memory	Authentication to unlock a locked PIN.
X9.17 PRNG keys	112 bit Triple-DES keys	Non-volatile memory	Used by X9.17 PRNG.

Table 7 - Summary of the Module's CSPs

Each module ships with a Creation key established by the manufacturer. This creation key is used as a PIN to authenticate the Crypto-Officer during initialization of the module.

During G&D initialization, the Triple-DES key is loaded onto the module for use by the X9.17 pseudo-random number generator (PRNG) and the initial seed for the PRNG is generated.

The STARCOS SPK 2.4 has ISFs that contain secret keys (DES, Triple-DES), private keys (RSA), and PINs/PUKs. Each ISF can hold multiple keys and is protected by a checksum (Generalized Hamming Code – GHC) calculated over the header/data of the ISF. ISFs are not readable by external operators and it is not possible to export a secret key from the module. Cryptographic operations are performed internal to the module using the proper commands and keys.

The access controls on keys follow the access control state machine outlined above. Permissions are granted based on the access control state the module is in for a particular DF or the MF. Keys also have properties defined at creation establishing what they can be used for. For instance, keys can be defined to only be used for encryption or authentication.

Keys can be updated in or added to an ISF if permission to do so is granted. Additionally, PINs can also be changed or added. Other than using the terminate card usage command, it is not possible to delete an ISF, though if a key is writeable, it can be overwritten.

IPFs store public keys (RSA). Multiple public keys can be stored in an IPF, and these public keys are readable by authorized operators. Public keys can be loaded into the module raw or in certificates, signed by a CA.

The module provides the capability to generate RSA keys and DES/Triple-DES session keys (through key establishment mechanisms). Other than session keys generated via a key establishment mechanism, there is no internal DES or Triple-DES key generation.

Besides session keys, all DES and Triple-DES keys are generated externally and then loaded onto the module. When loading these keys, the module is able to employ secure messaging to encrypt the communication.

The module can use user symmetric keys or session keys to secure communications when secure messaging is configured for a particular file. Session keys are generated using a symmetric or asymmetric key establishment mechanism. For both key establishment mechanisms, random data generated by both the module and the reader is used to construct a session key. Session keys are destroyed when the session is terminated or a new key establishment is initiated.

All keys can be zeroized by issuing the terminate card usage command.

2.6 Standards-Based Cryptography

The STARCOS SPK 2.4 module implements strong, standards-based cryptography. It includes the following FIPS-approved algorithms:

Data Encryption:

- Data Encryption Standard (DES) in CBC mode (for legacy use only) – as per NIST’s FIPS PUB 46-3
- Triple-DES (TDES) in CBC mode – as per NIST’s FIPS PUB 46-3

Data Hashing:

- Secure Hash Algorithm (SHA-1) – as per NIST's FIPS PUB 180-1

Data Integrity:

- DES-MAC – as per ANSI X9.19
- Triple-DES-MAC – as per ANSI X9.19
- RSA Digital Signatures – as per PKCS #1

Key establishment:

- Asymmetric key-based key establishment – as per ISO 9796-2
- Symmetric key-based key establishment – as per DIN Vornorm 66291 Teil 1

Pseudo Random Number Generation:

- PRNG based on ANSI X9.31 (Appendix A – A.2.4)

2.7 Self-Tests

The STARCOS SPK 2.4 runs startup and conditional self-tests to verify that it is functioning properly. These startup self-tests are performed either immediately when the module is powered on (POSTs) or before the module processes the first command it receives after a Reset (other startup self-tests). Conditional self-tests are executed whenever specific conditions are met. The self-tests include:

POSTs: When the module is powered on, it performs basic hardware checks and initializations to ensure proper functioning. If these checks are completed successfully, the POSTs are passed. Otherwise, it is failed.
Software Integrity Tests: The module checks the integrity of its firmware using a GHC. If the GHC verifies, the test is passed. Otherwise, it is failed.
Cryptographic Algorithm KATs: Known Answer Tests (KATs) are run at power-up for DES and Triple DES encryption/decryption, X9.31 PRNG random data generation, and SHA-1 hashing.

DES-CBC KAT
Triple-DES-CBC KAT
PRNG KAT
SHA-1 KAT

Startup RSA Pairwise Consistency Check: The module performs a sign/verify with an RSA key pair during startup to verify the proper functioning of the RSA implementation. If the signature verifies, the test is passed. Otherwise, it is failed.
Continuous Random Number Generator Test: This test is run to detect failure of the module's random number generator.
Conditional RSA Pairwise Consistency Check: After generating an RSA key pair, the module performs a sign/verify with that key pair to

ensure that the key pair is correct.

Conditional RSA Encryption/Decryption Pairwise Consistency Check: After generating an RSA key pair, the module performs an encryption/decryption with that key pair to ensure that the key pair is correct.

File Header Integrity Check: A check value calculated over the header of the file, which contains the permissions for the file, is verified. If that fails, the file is considered corrupt and access is denied**.**

If any of the power up self-tests or the continuous RNG test fail, the module will halt all operations until it is reset, indicating to the smart card reader via a timeout in the protocol that an error has occurred. If the conditional RSA pairwise consistency check or the file header integrity check fail, the command that required either of those tests fails and the module outputs an error message. If all of the power up self-tests or the conditional tests are passed, the module continues with its normal operations and outputs a command status message, indicating that the module is functioning properly.

2.8 Mitigation of Attacks

The module implements countermeasures for three attacks commonly used against smart cards: simple power analysis (SPA), differential power analysis (DPA), and timing analysis. These attacks work by monitoring the power consumption (SPA, DPA) or timing of operations during cryptographic processing in order to gain information about sensitive content, such as secret keys.

The module's IC has a co-processor for performing DES and Triple-DES operations. This co-processor was specifically designed by Philips Semiconductor to counter SPA, DPA, and timing analysis attacks. G&D and an independent third party conducted testing of the module's DES and Triple-DES processing for resistance to these attacks and found that no information was leaked during this processing via these attacks.

The module's RSA implementation has been hardened against SPA, DPA, and timing analysis using a variety of techniques. For timing analysis, the timing of the RSA implementation does not correlate to the inputs to the implementation. To counter SPA, conditional jumps based on the exponent and squares were avoided. Randomization of the base and exponent is employed to counter DPA. G&D and an independent third party conducted testing of the module's RSA processing for resistance to these attacks and found that no information was leaked during this processing via these attacks.

3. FIPS 140-2 OPERATION OF THE STARCOS SPK 2.4

The STARCOS SPK 2.4 module has a few minor configuration restrictions to ensure a FIPS 140-2 approved mode of operation. If these configuration restrictions are followed, the module will operate in compliance with FIPS 140-2.

In order to activate role-based authentication, the module's files must be configured to require authentication before allowing access to their contents, with one exception. Keys/PINS being used to authenticate an unauthenticated operator have to allow the necessary access for conducting the authentication. At a minimum, this may be achieved by configuring a global PIN for the MF and having access to the contents of the files on the module require that this PIN have already been entered.

When configuring a key for use during authentication, the mechanism must be set to either asymmetric external or asymmetric mutual, symmetric external, symmetric mutual, or master-slave (see Table 3). All PINs must be 8 bytes in length. Additionally, the Crypto-Officer or User must configure a limit on the number of failed attempts to authenticate with a key or PIN (the maximum setting supported by the module is 14).

If an RSA key pair (both the public and private components) is generated and stored within a module, this key pair must be configured for internal generation and storage only. By doing so, the only way to update the key pair is for the module to generate a new key pair and update both the stored private and public key pairs. In addition, The RSA key pair must be configured for signature generation/verification and/or authentication only.

When accessing security sensitive data or keys, secure messaging must be turned on. Since the data is required to be encrypted, secure messaging has to be set to combined mode. Both asymmetric and symmetric key-based key establishment mechanisms may be used.

Key zeroization requires the module's access control state machine for the MF to be in the maximum (highest access level) state (15) and secure messaging to be enabled. In order to provide this functionality, at least one authentication mechanism (PIN or key) must transition the MF access control state machine to this state and, while in this state, at least one key or key pair should allow secure messaging to be enabled.

While not required, it is recommended that keys not be used for multiple purposes. For example, a Triple-DES key used for authentication should not also be used to data encryption.

3.1 Crypto-Officer Guidance

The Crypto-Officer has three key responsibilities: initialization of the module, distribution of the module, and destruction of the module. As part of these responsibilities, the Crypto-Officer should track the life-cycle of each module, from receipt from the manufacturer to destruction by the Crypto-Officer.

3.1.1 Initialization

The Crypto-Officer receives the module from the manufacturer via a secure delivery mechanism. This is customer contract -dependent and might be a direct exchange from the manufacturer to the Crypto-Officer, but typically it involves shipping the module in a box containing a tamper-evident bag. Inside this box is a sealed bag containing the module(s) for use by the Crypto-Officer.

The Crypto-Officer should examine this bag for evidence of tampering before proceeding to use the module(s). Tamper-evidence includes tears, scratches, and other irregularities in the packaging.

The Crypto-Officer should maintain possession of the module at all times. The module should be checked regularly for signs of tamper-evidence (prying marks, strange scratches, and other signs listed in section 2.2). If the module is lost, it should be assumed compromised.

Once the Crypto-Officer has possession of the module, the Crypto-Officer must initialize the module for use by the User role. The Crypto-Officer authenticates to the module using the Creation key that is transmitted to the module as one of the parameters to begin creation of the file structure on the module. The actual value of the Creation key is discussed in the G&D user manuals for the module.

The Crypto-Officer will proceed to create the file structure on the module, establish access control mechanisms for the module, and to enter other data into the module. The Crypto-Officer must follow the configuration instructions at the beginning of this section in order to ensure operation of this module in a FIPS approved mode. Once creation of the file structure has been completed, the Crypto-Officer ends the initialization and the module’s configuration is put into effect.

At this point, the Crypto-Officer is ready to distribute the module to a User. The Crypto-Officer will no longer use the module, but the Crypto-Officer is responsible for distribution and destruction of the module.

3.1.2 Distribution

The Crypto-Officer should distribute the module to the User in a secure manner. The mechanisms of secure delivery can mirror those for delivery from the manufacturer (i.e. direct passing of the module from the Crypto-Officer to the User or shipping of the module from the Crypto-Officer to the User). If the module is shipped to the User, it is secured using some form of tamper-evident medium (e.g., a box with tamper-evident labels, a tamper-evident bag, etc.). The Crypto-Officer should confirm receipt of the module with the User.

The Crypto-Officer should keep track of the distribution of the module. As part of tracking the life of the module, the provision of the module to a User should be recorded. When the module is returned to the Crypto-Officer, this should be recorded as well. Any issues brought up by the User about the module should also be recorded.

During distribution of a module to a User, the Crypto-Officer should convey in a secure manner to the User the configuration of the module, including the necessary authentication information and access control settings.

When the User is finished with the module, it should be returned to the Crypto-Officer via secure delivery. The mechanisms of the delivery can follow what was used to deliver the module to the User.

If the module consistently malfunctions or otherwise repeatedly enters an error state, the Crypto-Officer should zeroize the module and optionally return it to the manufacturer.

3.1.3 Destruction

When a module's usage has been completed, the module should be zeroized by the Crypto-Officer in order to wipe all sensitive data and render the module inoperable. Additionally, the Crypto-Officer could destroy the module.

3.2 User Guidance

The User must check the tamper-evident medium for any indication of tampering. If tamper-evidence is located, the Crypto-Officer should be notified and the module should not be used.

The User is able to use the module as defined above in the description of the User role. Depending on the permissions defined by the Crypto-Officer, the User may be able to add DFs to the module, in which case the User must follow the rules at the beginning of the section for operating the module in a FIPS approved mode.

The User should maintain possession of the module at all times. The module should be checked regularly for signs of tamper-evidence (prying marks, strange scratches, etc.). If the module is lost, it should be assumed compromised.

If the module consistently malfunctions or otherwise repeatedly enters an error state, the User should return it to the Crypto-Officer. Additionally, if the User zeroizes the module, it should be returned to the Crypto-Officer.

Anything suspicious with the module should be reported to the Crypto-Officer immediately.

4. ACRONYMS

AC	Access Conditions
ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
CBC	Cipher-Block Chaining
CSP	Critical Security Parameters
DES	Data Encryption Standard
DF	Directory File
DIN	German Institute for Standardization
DPA	Differential Power Analysis
EDC	Error Detection Code
EF	Elementary Files
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EMV	Europay Mastercard and Visa
EEPROM	Electrically Erasable Programmable ROM
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
G&D	Giesecke & Devrient
GHC	Generalized Hamming Code
I/O	Input/Output
IC	Integrated Circuit
IPF	Internal Public File
ISF	Internal Secret File
ISO	International Organization for Standardization
KAT	Known Answer Test
MAC	Message Authentication Code
MF	Master File
N/A	Not Applicable
NIST	National Institute of Standards and Technology
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standards
PRNG	Pseudo Random Number Generator
PUK	Personal Unblocking Key
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest Shamir and Adleman
RST	Reset
SHA	Secure Hash Algorithm
SPA	Simple Power Analysis
STARCOS SPK 2.4	Smart Card Chip Operating System Standard Version with Public Key Extension 2.4