Gesellschaft für sichere Mobile Kommunikation mbH
CryptoPhone Security Kernel

(Software Version: 2.0)

FIPS 140-2
Non-Proprietary Security Policy

Level 1 Validation

Document Version 0.6

Prepared for:

Gesellschaft für sichere Mobile Kommunikation mbH
Marienstrasse 11
10117 Berlin, Germany
Phone: +49 - (0)700 - 2797 8835
Fax: +49 - (0)700 - 27 97 83 29
http://www.cryptophone.de/

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050
Fax: (703) 267-6810
http://www.corsec.com/

Revision History

Version	Modification Date	Modified By	Description of Changes
0.1	2008-04-14	Darryl H. Johnson	Initial draft.
0.2	2008-05-16	Darryl H. Johnson	Made modifications per lab request.
0.3	2008-05-27	Darryl H. Johnson	Made modifications per lab request.
0.4	2008-09-04	Darryl H. Johnson	Added algorithm certificate numbers; corrected platform names and configurations; made minor modifications per lab request.
0.5	2008-12-20	Darryl H. Johnson	Addressed CMVP comments.
0.6	2009-02-06	Darryl H. Johnson	Addressed final CMVP comments.

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Gesellschaft für sichere Mobile Kommunikation mbH CryptoPhone Security Kernel. This Security Policy describes how the CryptoPhone Security Kernel meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp/index.html.

The CryptoPhone Security Kernel is referred to in this document as the kernel or the module.

1.2 References

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

- The GSMK website (http://www.cryptophone.com/) contains information on the full line of products from GSMK.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/index.html) contains contact information for answers to technical or sales-related questions for the module.

1.3 Document Organization

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

- Vendor Evidence document
- Finite State Machine
- Other supporting documentation as additional references

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

2 CryptoPhone Security Kernel

2.1 Overview

The GSMK CryptoPhone G10i Secure GSM Smartphone is a fully featured, quad-band flip phone provides high-level, end-to-end encrypted mobile communications anywhere in the world. It operates virtually around the globe, providing secure mobile connectivity. Phone features include two Liquid Crystal Displays (LCDs), voice dial and command, speaker phone and mute capability, a microphone, camera/camcorder, and built-in vibration. It runs the Microsoft® Windows Mobile™ 5.0 operating system (OS) with Internet Explorer, Pocket Outlook, calculator, and media player. It has a Universal Serial Bus (USB) port and is Bluetooth-capable. The CryptoPhone 300i runs on Windows Mobile 6.1, and provides the same functions and features on a dual tri-band PocketPC platform.

GSMK also offers their the CryptoPhone PSTN/1,product, a purpose-built encryption unit that attaches directly to a fixed-line public switched telephone network (PSTN) or plain old telephone service (POTS) telephone. The encryption unit runs Microsoft Windows XP Embedded (XPe) with Service Pack 2 (SP2).

The CryptoPhone Security Kernel version 2.0 is a software library that provides the cryptographic functionality for both platforms. The module is a dynamic link library (dll) that performs encryption of GSM[1] telephone calls using both Advanced Encryption Standard (AES) and Twofish encryption. Diffie-Hellman key agreement is employed to construct a master secret, and encryption keys are derived from that secret.

In FIPS 140-2 terminology, the CryptoPhone Security Kernel is a multi-chip standalone software module that meets the Level 1 FIPS 140-2 requirements. The module was tested and found to be compliant with FIPS 140-2 requirements on devices equipped with the following processor/operating system combinations:

- ARM9 w/ Windows Mobile 5.0
- ARM11 w/ Windows Mobile 6.1
- VIA C3 w/ Windows XP Embedded (SP2)

The following table details the security level achieved by the CryptoPhone Security Kernel in each of the eleven sections of FIPS 140-2.

Table 1 – Security Level Per FIPS 140-2 Section

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

[1] GSM – Global System for Mobile Communications

Section	Section Title	Level
11	Mitigation of Other Attacks	N/A

2.2 Cryptographic Boundary

2.2.1 Logical Cryptographic Boundary

Figure 1 below provides a logical block diagram of the module executing in memory and its surrounding components. The logical cryptographic boundary encapsulates the CryptoPhone Security Kernel only.

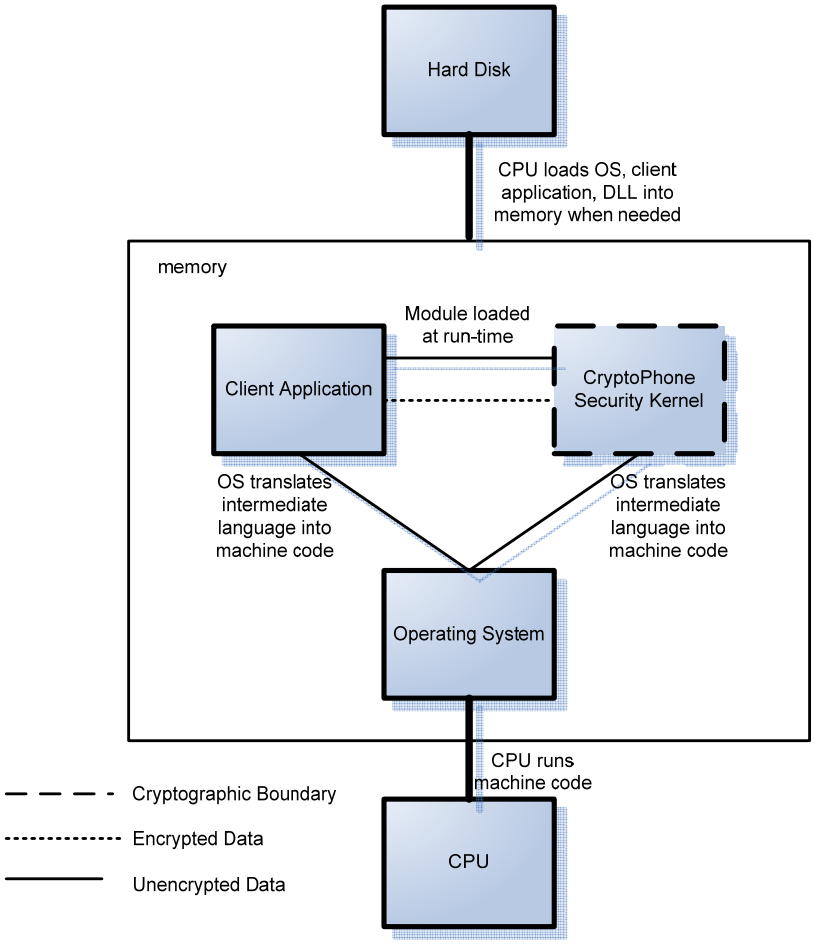

Figure 1 – CryptoPhone Security Kernel Logical Block Diagram

2.2.2 Physical Cryptographic Boundary

Although this is a software module, it is protected by the physical enclosure of the host device. Currently, the module executes on a standard Smartphone, PocketPC, or purpose-built fixed-line encryption unit; thus, the physical cryptographic boundary is the case of the host device. The casings are hard opaque plastic enclosures. Each physical device consists of motherboard circuits, the central processing unit (CPU), random access memory (RAM), read-only memory (ROM), and other hardware components included in the devices. Figure 2 provides a block diagram for a standard Smartphone, illustrating the various components, connections, and information flows (the dashed line surrounding the various components makes up the module's physical cryptographic boundary). Figure 3 provides the same for a standard PocketPC.

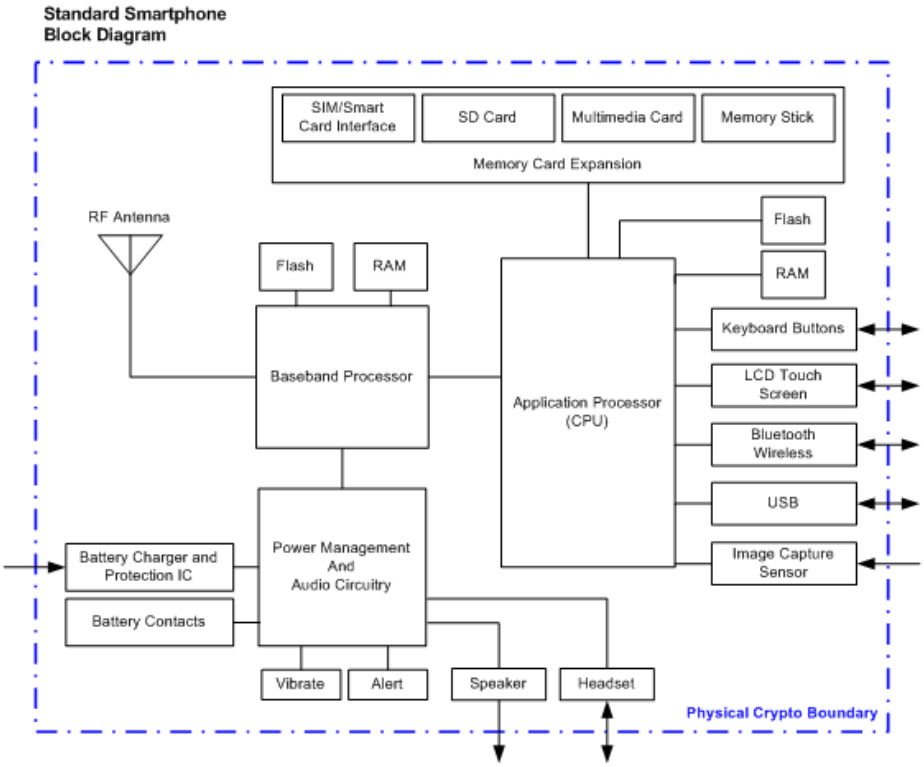

Figure 2 – Standard Smartphone Block Diagram

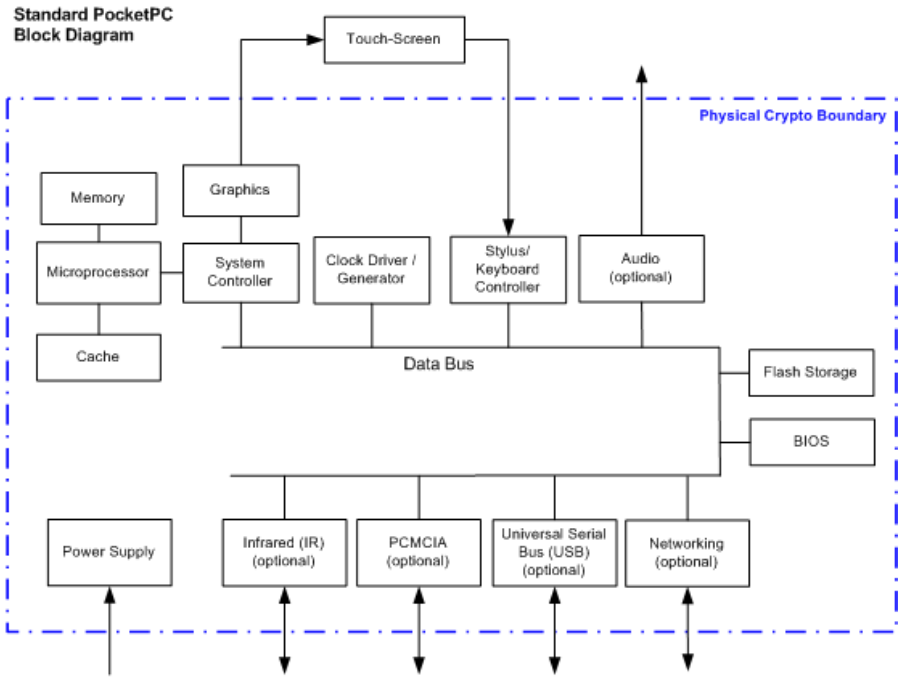

Figure 3 – Standard PocketPC Block Diagram

The CryptoPhone PSTN/1 fixed-line platform is essentially a general-purpose computer utilizing a modem and custom audio and phone line interfacing hardware. Figure 4 provides a block diagram for the device (the dashed line surrounding the various components makes up the module's physical cryptographic boundary).

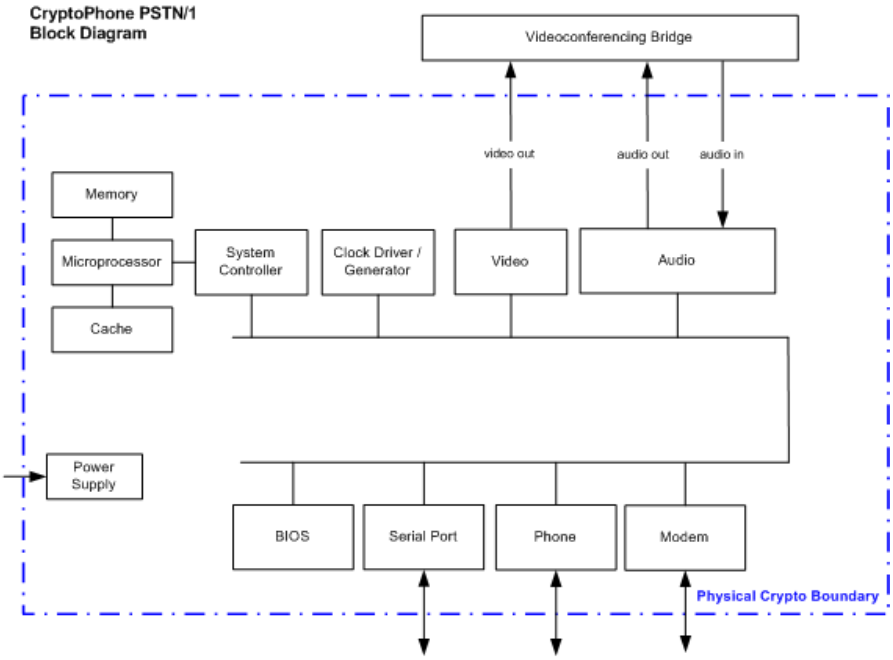

Figure 4 – CryptoPhone PSTN/1 Block Diagram

2.3 Module Interfaces

The CryptoPhone Security Kernel's physical ports are those provided by the host device. Table 2 below provides a list of the physical interfaces.

Table 2 – List of Physical Interfaces

Physical Interfaces		
Mobile Platforms		
Keypad	Microphone	Display
Antenna	Vibrator	Speaker
Headset jack	Power connection	
Fixed-line Platform		
Phone jack	Line jack	Power button
Power connection	Video jack	Audio (in/out) jacks

A mapping of the FIPS 140-2 logical interfaces, the physical interfaces, and the module can be found in Table 3 below.

Table 3 – Logical, Physical, and Module Interface Mapping

FIPS 140-2 Interface	Physical Interface Mapping	Module Mapping
Data Input Interface	Keypad, microphone, display, headset jack, antenna (mobile); phone jack, line jack, audio (in) jack (fixed-line)	Function calls that accept, as their arguments, data to be used or processed by the module.
Data Output Interface	Headset jack, display, speaker, antenna (mobile); phone jack, line jack, video jack, audio (out) jack (fixed-line)	Arguments for a function that specify where the result of the function is stored.
Control Input Interface	Keypad, display (mobile); power button (fixed-line)	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output Interface	Display, speaker, vibrator, headset jack (mobile); phone jack, line jack (fixed-line)	Return values for function calls
Power Interface	Power connection	N/A

2.4 Roles and Services

Two roles are supported by the module (as required by FIPS 140-2) that operators may assume: a Crypto Officer role and a User role. The operator of the module may implicitly assume either of the roles; the role assumed is based on the operation performed, and does not depend upon any explicit authentication mechanism.

Because this is a software module, services are provided in the form of function calls to the module's Application Programming Interface (API). Both of the roles and their respective responsibilities and services are described below.

2.4.1 Crypto Officer Role

The Crypto Officer has access to services that allow the execution of self-tests and status monitoring. Descriptions of the services (along with the inputs, outputs, critical security parameters (CSPs) and type of access for each) available to the Crypto Officer role are provided in the table below.

Table 4 – Mapping of Crypto Officer's Services to Inputs, Outputs, CSPs, and Type of Access

Service	Description	Input	Output	CSP and Type of Access
getStatus	Get power-up self-test status	API call	Power-up self-test status indicator	None
runSelfTest	Run power-up self-test	API call	Status	Software integrity test key - read

2.4.2 User Role

The User role has the ability to establish secure call connections with another similarly-equipped endpoint. Descriptions of the services available to the User role are provided in the table below.

Table 5 – Mapping of User's Services to Inputs, Outputs, CSPs, and Type of Access

Service	Description	Input	Output	CSP and Type of Access
AES::encrypt	Encrypt a block of data	API call, including plaintext data block	Encrypted data block	Symmetric keys - read

Service	Description	Input	Output	CSP and Type of Access
AES::decrypt	Decrypt a block of data	API call, including ciphertext data block	Decrypted data block	Symmetric keys - read
CBC<cipher>::decrypt	Decrypts a block of data using the specified algorithm	API call, including ciphertext data block	Decrypted data	Symmetric keys - read
CCM<cipher>::generate_encrypt	Encrypts and generates MAC for a block of data	API call, including input data	Encrypted data block with MAC	Symmetric keys - read
CCM<cipher>::decrypt_verify	Decrypts and verifies MAC for a block of data	API call, including input data	Decrypted, verifies data block	Symmetric keys - read
CTR<cipher>::encrypt	Encrypts a block of data using the specified algorithm	API call, including plaintext data block	Encrypted data block	Symmetric keys - read
CTR<cipher>::decrypt	Decrypts a block of data using the specified algorithm	API call, including ciphertext data block	Decrypted data block	Symmetric keys - read
Sha256::add	Processes additional input data	API call, including additional message data	None	None
Sha256::final	Compute SHA-256 hash for current message	API call, including message data	SHA-256 hash of input data	None
Sha384::add	Processes additional input data	API call, including additional message data	None	None
Sha384::final	Compute SHA-384 hash for current message	API call, including message data	SHA-384 hash of input data	None
Sha512::add	Processes additional input data	API call, including additional message data	None	None
Sha512::final	Compute SHA-512 hash for current message	API call, including message data	SHA-512 hash of input data	None
X931prng::random	Generates random value	API call, including input data	New random value; success/failure indicator	PRNG seed – read PRNG seed key – read
X931prng::seed	Provides PRNG seeding from collected entropy	API call, including entropy	New seed	PRNG seed – read/write
dh::serialise	Serialize a DH object	API call, including input data	Serialized API object; success/failure indicator	None

Service	Description	Input	Output	CSP and Type of Access
dh::generateKeyPair	Generates a DH public/private key pair	API call, including input data	New DH key pair; success/failure status	DH keys – write
dh::calculateSharedSecret	Calculates shared secret	API call, including input data	New shared secret; success/failure indicator	DH keys – read Shared secrete - write

2.4.3 Non-FIPS-Approved Services

The module also provides for encryption and decryption services using the non-FIPS-Approved algorithm Twofish. Those services are listed in Table 6 below.

Table 6 –Non-FIPS-Approved Services

Service	Description
Twofish::encrypt	Encrypt a block of data
Twofish::decrypt	Decrypt a block of data

If any of the services referenced in Table 6 are used, the module is not considered to be operating in the FIPS-Approved mode of operation.

2.5 Physical Security

The CryptoPhone Security Kernel is a multi-chip standalone cryptographic module. It is a software module and does not implement any physical security mechanisms.

Although the module consists entirely of software, the FIPS 140-2 tested platform is a standard Smartphone or PocketPC. Both devices have received FCC Grants of Equipment Authorization as Intentional Radiators, meeting Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for Intentional Radiators as defined in Subpart C of FCC Part 15.

The module was also tested on a purpose-built fixed-line telephone encryption unit. The encryption unit uses the RJModem™ embedded modem by Multi-Tech Systems, Inc. The modem has been tested and found to comply with FCC requirements for a Class B digital device, and currently holds an FCC Code of Federal Regulations Title 47 (CFR47) Part 15 (Class B) Certification. The modem vendor could provide more information on the modem's certification.

2.6 Operational Environment

The CryptoPhone Security Kernel operates on devices equipped with the following processor/operating system combinations:

- ARM9 w/ Windows Mobile 5.0
- ARM11 w/ Windows Mobile 6.1
- VIA C3 w/ Windows XP Embedded (SP2)

For FIPS 140-2 compliance, these are considered to be single user operating systems. As such, all keys, intermediate values, and other CSPs remain only in the process space of the operator using the module. The operating systems use their native memory management mechanisms to ensure that outside processes cannot access the process space used by the module.

2.7 Cryptographic Key Management

The module uses the following FIPS-Approved software algorithm implementations:

- Advanced Encryption Standard (AES) (256 bits) – Electronic Codebook (ECB), Cipher Block Chaining (CBC), 128-bit Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) – FIPS 197 (certificate #849)
- Counter with Cipher Block Chaining-Message Authentication Code (CCM) using AES – SP 800-38C (certificate #849)
- Secure Hash Algorithm (SHA-256, SHA-384, and SHA-512) – FIPS 180-2 (certificate #841)
- Pseudorandom Number Generator – General-purpose software implementation of American National Standards Institute (ANSI) X9.31 Appendix A.2.4 (certificate #485)
- Keyed-Hash Message Authentication Code (HMAC) using SHA-256, SHA-384, and SHA-512 (certificate #467)

The module uses the following FIPS-Allowed algorithm(s) when running in an Approved mode of operation:

- Diffie-Hellman (key agreement; key establishment methodology provides 149 bits of encryption strength)

The module also supports the following non-FIPS-Approved algorithm(s):

- Twofish (256 bits) – ECB, CBC, 128-bit CFB, OFB, CTR
- CCM with Twofish

The following table lists all cryptographic keys, key components, and CSPs used by the module.

Table 7 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
Software Integrity Test Key	512-bit HMAC key	Stored in the file *cryptokern_dll.mac*	Never exits the module	Stored in persistent memory in plaintext	When the module is uninstalled	To check the integrity of the crypto module
PRNG seed	256-bit seed	Generated externally and input by the calling application	Never exits the module	Plaintext in volatile memory	By uninstalling the module or power cycle	Generate random numbers
PRNG seed key	128-bit seed key	Generated externally and input by the calling application	Never exits the module	Plaintext in volatile memory	By uninstalling the module or power cycle	Generate random numbers
Diffie-Hellman public key	Public key	Generated internally from ANSI X9.31 A.2.4 PRNG or input in plaintext	Exits the module in plaintext	Plaintext in volatile memory	After shared secret is established	To establish shared secret
Diffie-Hellman private key	Private key	Generated internally from Diffie-Hellman public key	Never exits the module	Plaintext in volatile memory	After shared secret is established	To establish shared secret
Shared secret	4096-bit shared secret	Established during Diffie-Hellman key exchange	Never exits the module	Plaintext in volatile memory	After symmetric key is established or by power cycle	To generate AES key, Twofish key, and session code
AES symmetric key	256-bit encryption key	Derived from shared secret	Never exits the module	Plaintext in volatile memory	When call is completed	To produce 128-bit keystream for encryption and decryption
Session code	256-bit session code	Derived from shared secret	Never exits the module	Plaintext in volatile memory	When call is completed	To generate 6-character readout code
Readout code	6-character session verification code	Derived from session code	Never exits the module	Plaintext in volatile memory	When call is completed	To verify endpoints in call

2.8 Self-Tests

The module performs the following self-tests at power-up:

- Software integrity check: Verifying the integrity of the module using a Message Authentication Code produced from a 512-bit keyed hash of the module (HMAC-SHA-512).
- AES Known Answer Test (KAT): Verifying the correct operation of the AES algorithm implementation.
- SHA KAT: Verifying the correct operation of the SHA-256, SHA-384, and SHA-512 algorithm implementations.
- HMAC KAT: Verifying the correct operation of the HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 algorithm implementations.
- PRNG KAT: Verifying the correct operation of the PRNG implementation.

The module performs the following conditional self-tests:

- Continuous Random Number Generator (RNG) Test: Verifying that the Approved RNG does not repeatedly generate a constant value.

The module also performs the following critical functions tests at power-up:

- Twofish KAT: Verifying the correct operation of the Twofish algorithm implementation.
- Diffie-Hellman Test: Verifying the correctness of the shared secret calculation in the Diffie-Hellman algorithm implementation.

The module will start its services only after all the self-tests have passed. If the self-tests have not passed, it enters an error state and logs the failure. All error conditions can be cleared by restarting the module.

2.9 Design Assurance

GSMK uses Subversion to provide source code control. The Subversion tool is used for software version control, code sharing, and build management. Subversion also keeps track of what versions of files were used for each release and what combinations were used in builds.

Additionally, Microsoft Visual SourceSafe 6.0 is used to provide configuration management for the CryptoPhone Security Kernel's FIPS documentation. This software provides access control, versioning, and logging.

2.10 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

3 Secure Operation

The CryptoPhone Security Kernel meets Level 1 requirements for FIPS 140-2. The sections below describe how to ensure that the module is operating securely.

3.1 Initial Setup

The CryptoPhone Security Kernel requires no set-up. When the module is powered up, it runs the power-on self-tests. If the power-up self-tests pass, the module is deemed to be operating in FIPS mode.

3.1.1 Installation

The module runs on a standard Smartphone and PocketPC, as well as a purpose-built outboard encryption unit. These devices are delivered to the customer from the factory with the module pre-installed, and require no further actions from the customer in order for the module to execute as documented.

3.1.2 Management

No specific management activities are required to ensure that the module runs securely.

3.1.3 Zeroization

With the exception of the software integrity test key, all keys are created “on the fly” and zeroized as soon as they have served their purpose. The Diffie-Hellman keys generated by the PRNG are zeroized after the shared secret is computed. The shared secret is zeroized after generation of the AES and Twofish symmetric keys, which in turn are zeroized after the call is completed. The software integrity test key is zeroized at the factory by uninstalling the module or installing a new module.

3.2 Crypto Officer Guidance

The Crypto Officer can initiate the execution of self-tests, and can access the module’s status reporting capability. Self-tests can be initiated at any time by power cycling the module. Status is reported automatically at the completion of the self-test execution.

3.3 User Guidance

The User accesses the module’s cryptographic functionality. The User must not attempt to modify the configuration of the module as established by the Crypto Officer, nor should a User reveal any of the CSPs used by the module to other parties.

The cryptographic functionality of the module (i.e. the collection of User role services listed in Table 5 above) is accessed and exercised by the User each time the User makes a secure call using a suitably-equipped device.

4 Acronyms

Table 8 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CBC	Cipher-Block Chaining
CCM	Counter with Cipher-Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CP	CryptoPhone
CPU	Central Processing Unit
CSP	Critical Security Parameter
CTR	Counter
ECB	Electronic Codebook
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
GSM	Global System for Mobile Communications
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
LCD	Liquid Crystal Display
NIST	National Institute of Standards and Technology
OFB	Output Feedback
OS	Operating System
PKE	Public Key Exchange
POTS	Plain Old Telephone Service
PRNG	Pseudorandom Number Generator
PSTN	Public Switched Telephone Network
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read-Only Memory
SHA	Secure Hash Algorithm
SP2	Service Pack 2
USB	Universal Serial Bus
XOR	Exclusive-Or

Acronym	Definition
XPe	Microsoft Windows XP Embedded