Q1 Labs

Cryptographic Security Kernel

Software Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 1.0

Prepared for:

Q1 Labs
890 Winter Street, Suite 230
Waltham, MA 02451
United Stated of America

Phone: +1 781 250-5800
http://www.q1labs.com/

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Memorial Hwy, Suite 220
Fairfax, VA 22033
United States America

Phone: +1 703 267-6050
http://www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

This section introduces the non-proprietary Security Policy for the Cryptographic Security Kernel (CSK) by Q1 Labs.

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Cryptographic Security Kernel from Q1 Labs. This Security Policy describes how the Cryptographic Security Kernel meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which 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) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The module is referred to in this document as the Cryptographic Security Kernel, the CSK, the cryptographic module, 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 Q1 Labs website (http://www.q1labs.com) contains information on the full line of solutions from Q1 Labs.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical 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 Model document
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Q1 Labs. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to Q1 Labs and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Q1 Labs.

2 Cryptographic Security Kernel

This section describes the Cryptographic Security Kernel (CSK) by Q1 Labs.

2.1 Overview

Q1 Labs Cryptographic Security Kernel is multi-algorithm library providing general-purpose cryptographic services. The purpose of the module is to provide a single API for cryptographic functionality that can provide centralized control over FIPS-Approved mode status, provide availability of only FIPS-Approved algorithms or vendor-affirmed implementations of non FIPS-Approved algorithms, and provide for centralized logging and reporting of the cryptographic engine.

The CSK is used by the several product families within the Q1 Labs product line as the underlying cryptographic provider. A typical usage for the module is to provide the core cryptographic services for communications between devices, for secure UI interactions, and for secure remote management functions.

The Cryptographic Security Kernel is validated at the following FIPS 140-2 Section levels shown in Table 1.

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[1]
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC[2]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A
14	Cryptographic Module Security Policy	1

[1] N/A - Not Applicable
[2] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

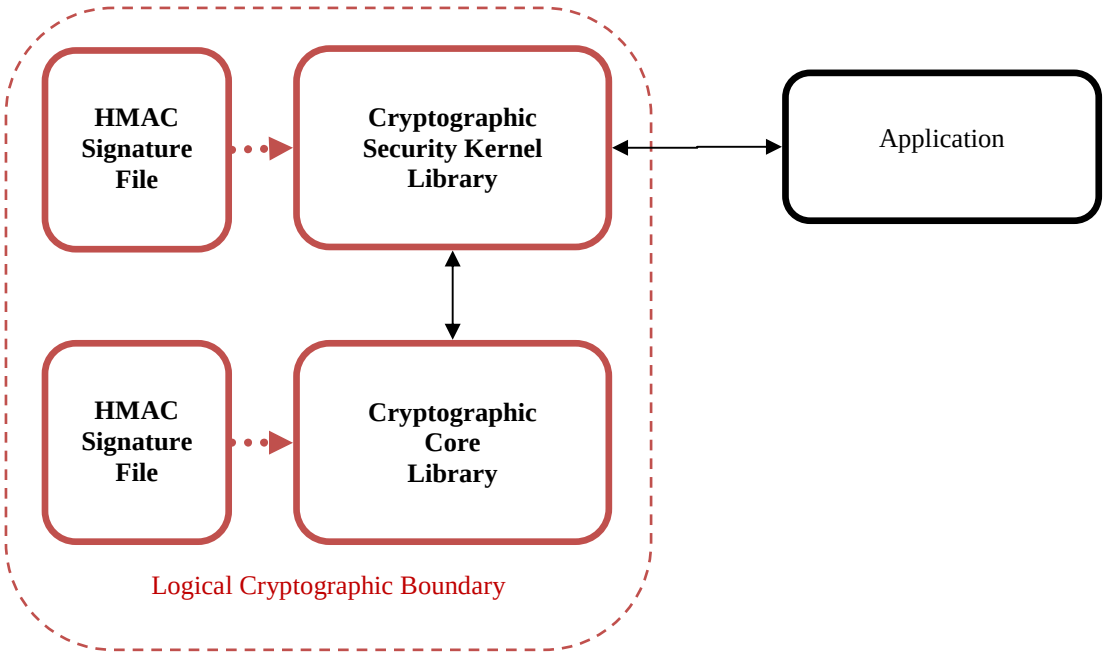
2.2 Module Specification

The Cryptographic Security Kernel is a software-only module that operates within a multi-chip standalone embodiment, such as a General Purpose Computer (GPC). The overall security level of the module is 1. The cryptographic boundary of the Cryptographic Security Kernel includes the following components as depicted in Figure 1.

- Q1 Labs Cryptographic Security Kernel Library
- Q1 Labs Cryptographic Core Library
- HMAC signature files generated for each library file (using the HMAC-SHA-256 algorithm)

The module supports only a FIPS-Approved mode. If the FIPS-Approved mode cannot be entered, due to a failure of software integrity checks or power-up self tests, the module will enter an error state and refuse to service cryptographic requests.

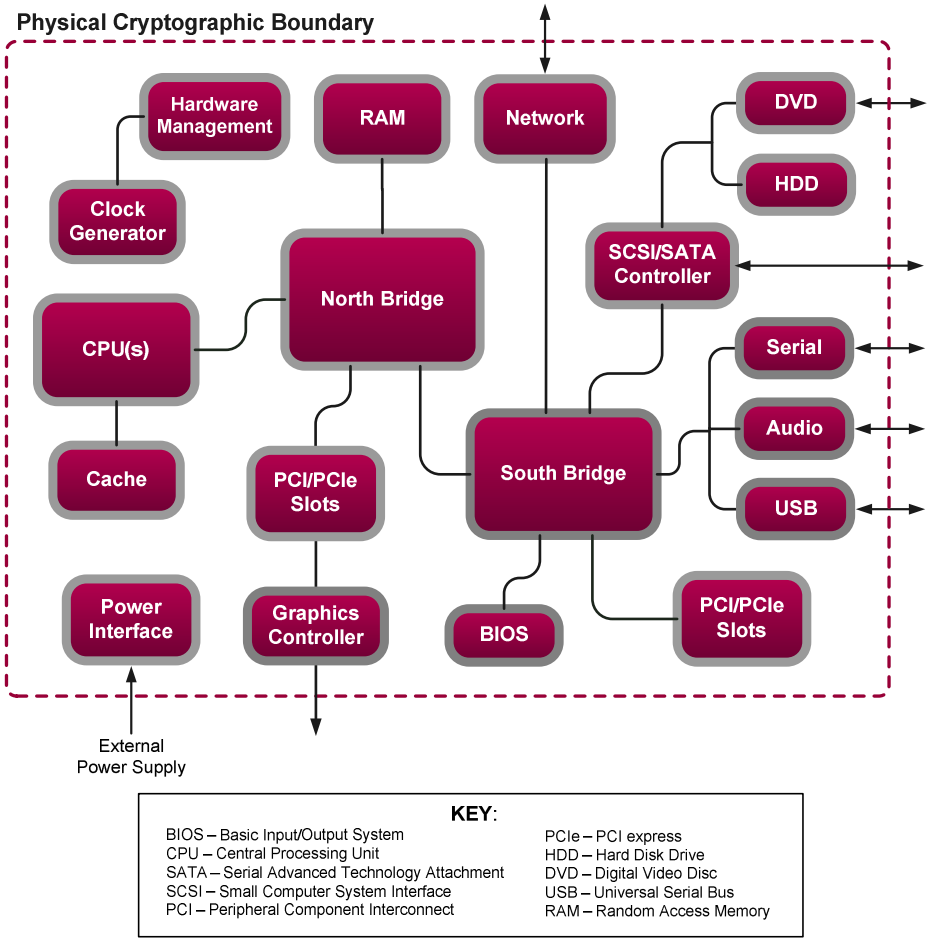
Figure 1 – FIPS 140-2 Logical Block Diagram



2.3 Module Interfaces

The module's interfaces are provided by the logical application programming interface (API), which provides the data input, data output, control input, and status output logical interfaces defined by FIPS 140-2. The module is installed on a GPC with physical ports consistent with that of a GPC as depicted in Figure 2.

Figure 2 – FIPS 140-2 GPC Block Diagram



The mapping of logical interfaces to the physical ports of the GPC is provided in Table 2 below.

Table 2 – FIPS 140-2 Logical Interface Mappings

FIPS 140-2 Logical Interface	Port/Interface Name	Module Interface
Data Input	Network, DVD, SCSI/SATA Controller, Serial, USB	Arguments for API calls that contain data to be used or processed by the module
Data Output	Network, SCSI/SATA Controller, Serial, USB, Graphics Controller	Arguments for API calls that contain or point to where the result of the function is stored
Control Input	Network, DVD, Serial, USB	API Function calls and parameters that initiate and control the operation of the module
Status Output	Network, Audio, Graphics Controller	Return values from API function calls and error messages
Power	Power Interface	N/A

2.4 Roles and Services

The Cryptographic Security Kernel is a software only module that provides an API for applications to perform general cryptography. As such, authentication is not provided by the module, though it may be enforced by the calling applications. The module has been designed to comply with FIPS 140-2 Level 1 requirements only, which does not require authentication. The module supports two roles in the module that operators may assume: a Crypto-Officer role and a User role, which are implicitly assumed.

2.4.1 Crypto-Officer Role

The Crypto-Officer role has the ability to query the module for status information and to force the module to perform startup self-tests.

Please note that the keys and critical security parameters (CSPs) listed in the table indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within a FIPS-Approved or Allowed security function or authentication mechanism.

Table 3 – Crypto-Officer Services

Service	Description	Input	Output	CSP and Type of Access
Get Version	Queries the module for the software version currently operating	None	Status	None
Get Status	Queries the module for the current operating status (Operational or Failed)	None	Status	None
Log Status	Queries the module for the current operating status and outputs to the logging facilities	None	None	None
Perform Self-Test	Forces the module to perform Known Answer Tests (KATs) for all appropriate algorithms and update the module error state	None	Status	None

2.4.2 User Role

The User role has the ability to perform basic cryptographic operations. Descriptions of the services available to the User role are provided in Table 4 below.

Table 4 – User Services

Service	Description	Input	Output	CSP and Type of Access
Generate random number (ANSI X9.31)	Returns the specified number of random bits to calling application	API call parameters	Status, random bits	ANSI X9.31 RNG seed – RWX ANSI X9.31 seed key – RX
Generate keyed hash (HMAC)	Compute and return a message authentication code using HMAC-SHAx	API call parameters, key, message	Status, hash	HMAC key – RX

Service	Description	Input	Output	CSP and Type of Access
Zeroize key	Zeroizes and de-allocates memory containing sensitive data	API call parameters	Status	AES key – W TDES key – W HMAC key – W RSA private/public key – W DH[3] components – W
Symmetric encryption	Encrypt plaintext using supplied key and algorithm specification (TDES or AES)	API call parameters, key, plaintext	Status, ciphertext	AES key – RX TDES key – RX
Symmetric decryption	Decrypt ciphertext using supplied key and algorithm specification (TDES or AES)	API call parameters, key, ciphertext	Status, plaintext	AES key – RX TDES key – RX
Generate asymmetric key pair	Generate and return an RSA asymmetric key pair	API call parameters	Status, key pair	RSA private/public key – W
RSA encryption	Encrypt plaintext using RSA public key (used for key transport)	API call parameters, key, plaintext	Status, ciphertext	RSA public key – RX
RSA decryption	Decrypt ciphertext using RSA private key (used for key transport)	API call parameters, key, ciphertext	Status, plaintext	RSA private key – RX
DH key agreement	Perform key agreement using Diffie-Hellman algorithm	API call parameter	Status, key components	DH components – W
Signature Generation	Generate a signature for the supplied message using the specified key and algorithm (RSA)	API call parameters, key, message	Status, signature	RSA private key – RX,
Signature Verification	Verify the signature on the supplied message using the specified key and algorithm (RSA)	API call parameters, key, signature, message	Status	RSA public key – RX

2.5 Physical Security

The Cryptographic Security Kernel is a software only module, which operates on a multi-chip standalone device, such as a GPC. As such, it does not include physical security mechanisms and the FIPS 140-2 requirements for physical security are not applicable.

2.6 Operational Environment

The module was validated with FIPS 140-2 requirements on each of the following operating system platforms:

- Red Hat Enterprise Linux (RHEL) 5.7
- CentOS 5.7

[3] DH – Diffie-Hellman

The operating system must be configured for single user mode for FIPS 140-2 compliance (see section 3 for guidance).

All cryptographic keys and CSPs are under the control of the OS or calling applications, which is responsible for protection of the CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to CSPs through its well-defined APIs. The module performs a Software Integrity Test using the HMAC-SHA-256 algorithm.

2.7 Cryptographic Key Management

The module implements the FIPS-Approved algorithms listed in Table 5 below.

Table 5 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
AES[4] 128/192/256 in ECB[5]/CBC[6]/CFB[7]/OFB[8] modes	#1907
Triple-DES[9] (TDES) 128/192 in TECB/TCBC/TCFB64/TOFB modes	#1239
RSA[10] (X9.31, PSS, PKCS#1.5) for signing, signature generation, and verification – 1024- , 2048- , and 3072-bit	#978
SHA[11]-256, SHA-384, SHA-512	#1674
HMAC-SHA-256, HMAC-SHA-512	#1144
NIST-Recommended ANSI X9.31 Deterministic Random Number Generator (DRNG) using AES	#1001

Additionally, the module utilizes the following non FIPS-Approved algorithm implementation:

- RSA with 1024- to 4096-bit keys (key wrapping; key establishment methodology provides between 80 and 150 bits of encryption strength)
- Diffie-Hellman (key agreement; key establishment methodology provides 80 bits of encryption strength)
- MD5[12] within TLS[13] only

Please see NIST document SP800-131A for guidance regarding the use of non FIPS-Approved algorithms.

[4] AES - Advanced Encryption Standard
[5] ECB - Electronic Code Book
[6] CBC - Cipher-Block Chaining
[7] CFB - Cipher Feedback
[8] OFB - Output Feedback
[9] DES - Data Encryption Standard
[10] RSA - Rivest, Shamir and Adleman
[11] SHA - Secure Hash Algorithm
[12] MD5 - Message-Digest algorithm 5
[13] TLS – Transport Layer Security

The module supports the critical security parameters (CSPs) listed below in Table 6.

Table 6 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP/Key Type	Generation / Input	Output	Storage	Zeroization	Use
AES Keys	AES128, 192, 256 bit keys	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot.	Keys are passed as arguments to the module API for cryptographic processing.
TDES Keys	TDES 128, 192 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot.	Keys are passed as arguments to the module API for cryptographic processing.
RSA private key	RSA 1024, 2048, 3072 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Signature generation, decryption
		Internally generated	API call parameter			Used by host application
RSA Public Key	RSA 1024, 2048, 3072 bit key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Signature verification, encryption
		Internally generated	API call parameter			Used by host application
DH Public Components	DH 1024 bit key	Internally generated	API call parameter	Plaintext in volatile memory	By API call, power cycle, host reboot	Used by host application
DH Private Components	DH 160 bit key	Internally generated	API call parameter	Plaintext in volatile memory	By API call, power cycle, host reboot	Used by host application
DRNG Seed Value	128 bit random value	Imported from dev/urandom	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Generate random number
DRNG Seed Key	AES 256 bit key	Imported from dev/urandom	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Generate random number
HMAC Key	HMAC key	API call parameter	Never	Plaintext in volatile memory	By API call, power cycle, host reboot	Message Authentication
Software Integrity Keys (HMAC)	HMAC key	Never	Never	On host system as plaintext file	By uninstalling the module	Used to perform the software integrity test at power-on.

2.7.1 Key Generation

The module uses an ANSI X9.31 Appendix A.2.4 DRNG implementation to generate cryptographic keys. This DRNG is FIPS-Approved as shown in Annex C to FIPS PUB 140-2.

2.7.2 Key Entry and Output

The cryptographic module itself does not support key entry or key output from its physical boundary. However, keys are passed to the module as parameters from the applications resident on the host platform via the exposed APIs. Similarly, keys and CSPs exit the module in plaintext via the well-defined exported APIs.

2.7.3 Key/CSP Storage and Zeroization

Symmetric, asymmetric, and HMAC keys are either provided by or delivered to the calling process, and are subsequently destroyed by the module at the completion of the API call. Keys and CSPs stored in RAM can be zeroized by a power cycle or a host system reboot. The X9.31 DRNG seed and seed key are initialized by the module at power-up and remain stored in RAM until the module is uninitialized by a host system reboot or power cycle. The HMAC keys that are used to verify the integrity of the module during power-on self tests are stored in files residing on the host GPC.

2.8 EMI/EMC

The Cryptographic Security Kernel is a software module. Therefore, the only electromagnetic interference produced is that of the host platform on which the module resides and executes. FIPS 140-2 requires that the host systems on which FIPS 140-2 testing is performed meet the Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. However, all systems sold in the United States must meet these applicable FCC requirements.

2.9 Self-Tests

This section describes the power-up and conditional self-tests performed by the module.

2.9.1 Power-Up Self-Tests

The Cryptographic Security Kernel performs the following self-tests at power-up:

- Software integrity checks (HMAC SHA-256) over each component of the module.
- Known Answer Tests (KATs)
  - AES
  - Triple-DES
  - RSA
  - HMAC SHA-256
  - HMAC SHA-512
  - ANSI X9.31 DRNG

2.9.2 Conditional Self-Tests

The Cryptographic Security Kernel performs the following conditional self-tests:

- Continuous DRNG Test
- RSA Pairwise Consistency Check

2.10 Design Assurance

Q1 Labs uses SVN as their software configuration management tool. It is used to manage the module's source code and configuration files. Q1 Labs uses Bugzilla to track all changes made to a project during its evolution, including the project requirements and task assignments.

2.11 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 Cryptographic Security Kernel meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-Approved mode of operation.

The Crypto-Officer role is responsible for installing the module as part of its host application. The cryptographic module is installed and always operates in a FIPS-Approved mode of operation. The cryptographic module implements a software integrity test that consists of an HMAC-SHA-256 computed over each of the libraries. During the power-up self-tests phase, the signatures are verified over the stored CSK Module instance. If the stored signatures are verified, then the test is passed. Otherwise, the test is failed and the module enters an error state where no cryptographic functionality is allowed.

3.1 Initial Setup

The module components must be installed within the same directory.

On the host operating system, pre-linking of library modules must be disabled. The following command sequence, when executed at the console while logged in as user 'root', will disable pre-linking:

1. cd /etc/sysconfig/prelink
2. ./prelink –u –a

3.2 Secure Management

This section provides guidance which ensures that the module is always operated in a secure configuration.

3.2.1 Initialization

FIPS 140-2 mandates that a software cryptographic module at Security Level 1 shall be restricted to a single operator mode of operation. Prior to installing the module, the Crypto-Officer must ensure that the Linux operating system environment is in single-user mode.

During initialization of applications that require use of the Cryptographic Security Kernel, the calling application(s) must call the 'QCrypto_Init()' function to initialize the module. Failure to properly initialize the module will result in the module operating in an unsupported configuration outside of the scope of its FIPS validation.

3.2.2 Management

The Crypto-Officer should monitor the module's status regularly and make sure only the services listed in Table 3 and Table 4 are being used. If any irregular activity is noticed or the module is consistently reporting errors, then Q1 Labs customer support should be contacted.

3.2.3 Zeroization

The module does not provide for persistent storage of cryptographic keys. The calling applications are responsible for key management, protection, and zeroization. Thus, zeroization is not required by the module.

3.3 User Guidance

Only the module's cryptographic functionalities are available to the User. Users are responsible to use only the services that are listed in Table 4. Although the User does not have any ability to modify the configuration of the module, they should report to the Crypto-Officer if any irregular activity is noticed.

The User must not modify the configuration of the module as established by the Crypto-Officer.

4 Acronyms

This section describes the acronyms.

Table 7 – Acronyms

Acronym	Definition
API	Application Programming Interface
ATM	Automated Teller Machine
BIOS	Basic Input/Output System
CBC	Cipher Block Chaining
CDR	Call Detail Record
CFB	Cipher Feedback
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CTR	Counter
DRNG	Deterministic Random Number Generator
DSA	Digital Signature Algorithm
DVD	Digital Video Disc
ECB	Electronic Codebook
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
GUI	Graphical User Interface
HDD	Hard Disk Drive
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
LAN	Local Area Network
N/A	Not Applicable
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OFB	Output Feedback
OS	Operating System

Acronym	Definition
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PEFS	Private Encryption File System
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
RNG	Random Number Generator
RSA	Rivest Shamir and Adleman
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
SHA	Secure Hash Algorithm
SKV	Secure Key Vault
SLS	Scalable Log Server
TDES	Triple Data Encryption Standard
TLS	Transport Layer Security
USB	Universal Serial Bus
WAN	Wide Area Network

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Hwy, Suite 220
Fairfax, VA 22033
United Stated of America

Phone: +1 703 267-6050
Email: info@corsec.com
http://www.corsec.com