Everyplace™ Wireless Gateway Cryptographic Module

Version 1.6

Security Policy

Document Version 1.11

Table of Contents

Table of Tables

Table of Figures

Document Control Information

Change history:

Version	Date	By Whom	Description
1.1	5/24/2002	Henry Welborn	Initial Draft
1.2	6/4/2002	Henry Welborn	Updates to Level 2 compliance
1.3	6/17/2002	Henry Welborn	Updates suggested by EWA
1.4	7/17/2002	Henry Welborn	Updates suggested by IBM legal, hardware configurations, key transport
1.5	9/16/2002	Henry Welborn	Updates to Windows CE tested configuration
1.6	9/24/2002	Henry Welborn	Updates to PRNG specification
1.7	11/6/2002	Dave MacFarlane	Updates for CMVP submission
1.8	11/20/2002	Henry Welborn	Add AIX 5.2 Support
1.9	04/04/2003	Henry Welborn	Updates for CMVP comments
1.10	05/14/2003	Henry Welborn	Updates for CMVP comments
1.11	05/22/2003	Henry Welborn	Updates for CMVP comments

Introduction

The IBM® Everyplace™ Wireless Gateway Cryptographic Module is a FIPS PUB 140-2 validated software cryptographic module providing encryption and other cryptographic services for the IBM® Everyplace™ Wireless Gateway for Multiplatforms.

The IBM® Everyplace™ Wireless Gateway for Multiplatforms is a distributed, scalable, multipurpose UNIX® communications platform that can support optimized, security-enhanced data access by both Wireless Application Protocol (WAP) clients and non-WAP clients over a wide range of international wireless network technologies, as well as local area (LAN) and wide area (WAN) wire line networks.

All the cryptographic services of the IBM® Everyplace™ Wireless Gateway (hereafter referred to as the Wireless Gateway) are implemented by the IBM® Everyplace™ Wireless Gateway Cryptographic Module (hereafter referred to as the Cryptographic Module), which is dynamically linked into the Wireless Gateway product code.

Components of the Wireless Gateway using a validated version of the Cryptographic Module are in compliance with the requirements of FIPS 140-2. The following IBM Everyplace Wireless Gateway components utilize the Cryptographic Module and are therefore (FIPS) 140-2 [1] compliant components[1]:

- Wireless Gateway for Solaris
- Wireless Gateway for AIX
- Wireless Client for Windows®
- Wireless Client for Windows CE

[1] Although the Wireless Gatekeeper for AIX®, Solaris. Linux, and Windows, and the Wireless Client for Palm OS and QNX Neutrino are considered components of the Everyplace Wireless Gateway, they do not utilize the cryptographic module described in this document and thus do not meet FIPS 140-2 requirements.

The Wireless Gateway provides these FIPS 140-2 compliant services[2]:

- Mobile access services

This document focuses on the features and security policies provided by the Cryptographic Module,, and describes how the module complies with FIPS 140-2 requirements. For more information on the Everyplace Wireless Gateway, refer to http://www.ibm.com/pvc.

[2] Although the Wireless Gateway also provides a WAP version 1.2.1 proxy and messaging services, they do not utilize the cryptographic module described in this document and thus do not meet FIPS 140-2 requirements.

Operation of the Cryptographic Module

The cryptographic module must be utilized as described herein to maintain its FIPS 140-2 validation. It is the responsibility of the host application and its administrators to install and employ the services of the cryptographic module in a FIPS 140-2 compliant manner.

The module is available as a software shared library on multiple platforms (*.dll* for Windows/Windows CE and *.so* for Solaris/AIX). The platforms used for FIPS 140-2 conformance testing are outlined in the *Cryptographic Module Specification* section of this document. The module must be used in one of the specified operating environments or an acceptable equivalent to remain FIPS 140-2 compliant.

An application utilizes the module through the interfaces specified in the *Cryptographic Module Interfaces* section of this document. A list of all services provided through these interfaces may be found in the *Cryptographic Module Roles and Services* section of this document.

The module requires authenticated access to perform cryptographic and diagnostic functions. The module provides two operator roles:

- Crypto Officer
- User

An application must login to the module to enable the appropriate services. The roles govern which of the services are available for operator use. The *Cryptographic Module Roles and Services* section of this document details the identification and authentication, and access control policies of the module.

The module provides for the protection of sensitive data, such as keys or Critical Security Parameters (CSPs). Information on key protection is outlined in the C*ryptographic Module Key Management* section. When the module is initialized, it verifies its own integrity, and confirms that the algorithms are functioning correctly. The *Cryptographic Module Self-Tests* section details the self tests performed by the module.

Cryptographic Module Specification

The services of the cryptographic module software library are accessible for C and C++ language programs through an application programming interface (API). The available API functions are listed in the *Cryptographic Module Roles and Services* section. Usage guidelines and details of the API function are available in the *IBM Everyplace Wireless Gateway Cryptographic Module API* document.

The module has been tested and validated to the following FIPS 140-2 security levels:

	Microsoft® Windows Microsoft Windows CE	Trusted Solaris 8 AIX 5L V5.2
Overall	**Security Level 1**	**Security Level 2**
Cryptographic Module Specification	Security Level 1	Security Level 2
Cryptographic Module Ports and Interfaces	Security Level 1	Security Level 2
Roles, Services, and Authentication	Security Level 2	Security Level 2
Finite State Model	Security Level 1	Security Level 2
Physical Security	N/A	N/A
Operational Environment	Security Level 1	Security Level 2
Cryptographic Key Management	Security Level 1	Security Level 2
EMI/EMC	Security Level 3	Security Level 2
Self-Tests	Security Level 1	Security Level 2
Design Assurance	Security Level 2	Security Level 2
Mitigation of Other Attacks	N/A	N/A

Table 1 - Module FIPS 140-2 Security Levels

The operating environments used to test the module are outlined as follows:

Hardware Platform	Test Hardware Specifications	Operating System
Sun SPARC Server	SunBlade 1000 2 X UltraSPARC III, 750 MHz 1 GB RAM	Trusted Sun Solaris Version 8 04/01 (Evaluated to Common Criteria EAL4) [7]
IBM Compatible PC with an Intel® Pentium® (or greater) or compatible processor	Acer Power 4300 Pentium III, 650 MHz 184 MB RAM	Microsoft Windows 2000 Professional SP2
Windows CE handheld device with either an ARM, MIPS, SH3, or compatible processor	Compaq iPAQ 3950 Intel X-Scale™ PXA250, 400MHz 64 MB Memory	Microsoft Pocket PC 2002 (Microsoft Pocket PC Version 3.0.11171)
IBM eServer pSeries	IBM pSeries 660 Model 6H1 64-bit RS64 IV, 600-750 MHz 1024 MB Memory	AIX Version 5L 5.2 (Evaluated to Common Criteria EAL4+) [8]

Table 2 - Module Tested Configurations

As outlined in G.5 of the Implementation Guidance for FIPS 140-2 [9], the module maintains its compliance for other operating environments on a general-purpose computer (GPC), provided:

- For Security Level 1, the GPC uses the specified single user operating system/mode specified on the validation certificate, or another compatible single user operating system;
- For Security Level 2, the GPC incorporates the specified Common Criteria evaluated (or equivalent) operating system/mode/operational settings or another compatible evaluated Common Criteria (or equivalent) operating system with like mode and operational settings; and
- The software of the cryptographic module does not require modification when ported (platform-specific configuration settings are excluded).

The Everyplace Wireless Gateway Cryptographic Module for Windows was conformance tested on the Microsoft Windows 2000 Professional with Service Pack 2 operating system. The software module maintains FIPS 140-2 compliance when running on the Microsoft Windows 95, Microsoft Windows 98, Microsoft Windows Me, Microsoft Windows NT®, and Microsoft Windows XP operating systems in a single-user mode.

The Everyplace Wireless Gateway Cryptographic Module for Windows CE was conformance tested on the Microsoft Pocket PC 2002 operating system. The software module maintains FIPS 140-2 compliance when running on other Microsoft Windows CE based operating systems, including Microsoft Windows CE .NET, Microsoft Handheld PC 2000, Microsoft Pocket PC, and Microsoft Handheld PC Professional Edition.

The Everyplace Wireless Gateway Cryptographic Module for Solaris and AIX was tested and FIPS 140-2 validated on the Common Criteria evaluated Solaris and AIX operating systems as noted in Table 2. The software module maintains FIPS 140-2 Level 2 compliance only when running on these operating systems in the evaluated configurations as specified in the applicable Common Criteria certification reports, or acceptable equivalents.

The cryptographic module supports FIPS 186-2 Appendix 3.1 for Approved random number generation, as well as the following validated Approved algorithms:

Type	**Algorithm**	**Specification**
Symmetric Cipher	AES (ECB, CBC) (Cert. #36)	FIPS 197 [2]
	DES (ECB, CBC) (Cert. #191) Triple DES (ECB, CBC) (Cert. #142)	FIPS 46-3 (single DES for legacy purposes only) [3]
Message Digest	SHA-1 (Cert. #127)	FIPS 180-2 [4]
Digital Signature	DSA (Cert. #74)	FIPS 186-2 [5]

Table 3 - Module Approved Algorithms

In addition, the module supports the following non-approved algorithms:

Type	**Algorithm**	**Specification**
Random Number Generation	Universal Software Based True Random Number Generator	Available upon request from IBM. Patented by IBM, EC Pat. No. EP1081591A2, U.S. pat. pend.
Key Wrapping	AES	AES Key Wrap Specification [6]

Table 4 - Module Non-approved Algorithms

Cryptographic Module Ports and Interfaces

The cryptographic module is a software module implemented as a shared library file (.dll on Windows/WindowsCE and .so on Solaris/AIX) that executes on a general purpose computer system. The physical cryptographic boundary is defined at the perimeter of the computer system enclosure on which the cryptographic module is to be executed, and includes all the hardware components within the enclosure. The cryptographic module interfaces with the Central Processing Unit (CPU) of the respective platform. The RAM and hard disk found on the computer are memory devices that store and execute the cryptographic module and its data. The logical or software cryptographic boundary is defined as the shared library which constitutes the software component of the cryptographic module.

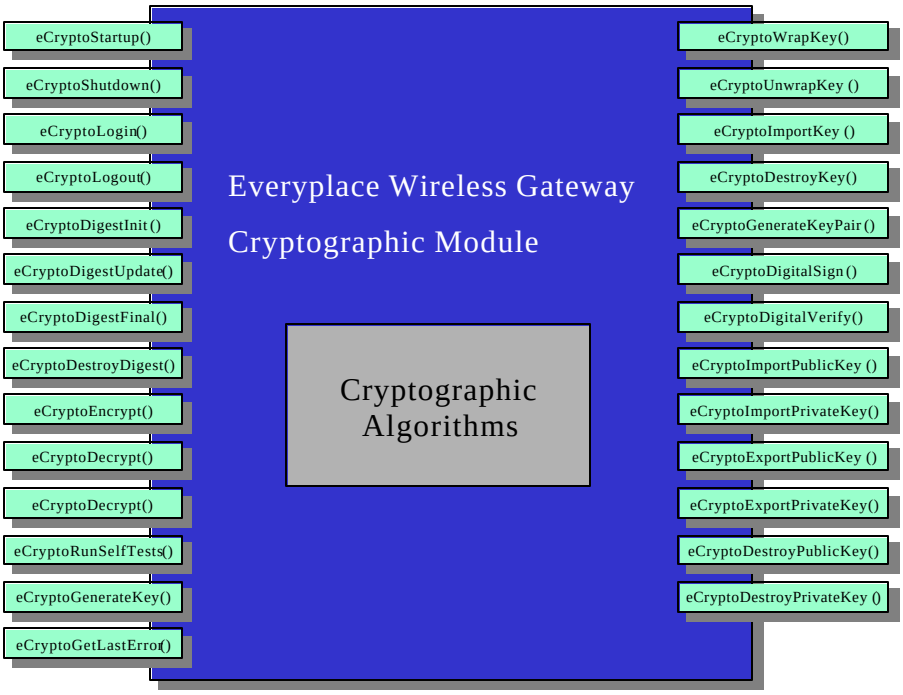

Figure 1 – Everyplace Wireless Gateway Cryptographic Module

The cryptographic module is classified as a “multi-chip standalone module”. The module’s physical ports consist of those found as part of the computer’s hardware, such as the keyboard, mouse, disk drive, CD drive, network adapters, serial and USB ports, monitor, speakers, etc. The module’s logical interfaces are provided through the documented API.

Each of the FIPS 140-2 defined logical interfaces are implemented as follows:

- Data Input Interface – data passed into the module with the API function calls
- Data Output Interface – data returned from the module with the API function calls
- Control Input Interface – the API function calls executed by the module
- Status Output Interface – return and error codes returned from the module with the API function calls

Cryptographic Module Roles and Services

The cryptographic module implements both a Crypto Officer and a User role, and employs a role-based authentication process. Only one role may be active at a time and concurrent operators are not permitted.

The authentication process is implemented using X.509 DSA/SHA-1 certificates signed by the root 1024-bit DSA private key. A root certificate containing the corresponding public key is inserted into the cryptographic module at compilation time. Two X.509 authentication certificates (also 1024-bit DSA), signed by the root private key, are provided to the module operator for use during the authentication process for the two supported roles, Crypto Officer and User.

When the operator logs into the cryptographic module, the corresponding authentication certificate must accompany the role identifier to validate the authentication. The use of roles is explicitly enforced by the cryptographic module.

Only one role may be active in the module at a time. When a transition is made from a User role to a Crypto Officer role or vice versa, all data objects and key/CSP values within the User's or Crypto Officer's process space are actively zeroized. Each time an operator changes roles, they must authenticate to that role, regardless of any previous authentication.

The Crypto Officer role is responsible for initiating a subset of the power-up self tests - the cryptographic algorithm self test diagnostics (complete self-tests are performed automatically at power-up), and does not have the ability to perform operational cryptographic functions. Attempts to execute unauthorized APIs will result in an error code being returned.

The User role is responsible for generating and entering keys, and performing cryptographic functions on data. The User role cannot initiate the cryptographic algorithm self test diagnostics. Attempts to execute unauthorized APIs will result in an error code being returned.

The module does provide some services for which an authenticated role is not required. For example, no login is required to startup the module, as well as to perform the complete power-up self tests.

Complete Crypto Officer and User guidance information is available in the *IBM Everyplace Wireless Gateway Cryptographic Module API* document.

The module services are accessible from C and C++ language programs through an Application Program Interface (API). The available services are:

Services	**Description**	**Role (CO, User, or No Role)**
Power-up self tests	Perform integrity and cryptographic algorithm power-up self tests by loading or rebooting the module	CO, User, or No Role
eCryptoStartup	Initializes the module, resetting all internal information.	No Role
eCryptoShutdown	Closes the module interface, resetting all internal information and zeroizing keys/CSPs.	CO, User, or No Role
eCryptoLogin	Begins a module operator's session.	CO, User, or No Role
eCryptoLogout	Ends a module operator's session and zeroizes all keys/CSPs.	CO or User
eCryptoDigestInit	Initializes a message digest for use.	User
eCryptoDigestUpdate	Updates a message digest context with data.	User
eCryptoDigestFinal	Outputs the hashed contents of the digest context.	User
eCryptoDestroyDigest	Deletes a digest context.	User
eCryptoEncrypt	Encrypts a data buffer.	User
eCryptoDecrypt	Decrypts a data buffer.	User
eCryptoRunSelfTests	Performs a subset of the power-up self-tests – the known answer tests (KATs) for the module's cryptographic algorithms	CO

eCryptoGenerateKey	Creates a symmetric key.	User
eCryptoWrapKey	Outputs a symmetric key.	User
eCryptoUnwrapKey	Inputs a symmetric key.	User
eCryptoImportKey	Inputs a plain text symmetric key.	User
eCryptoDestroyKey	Deletes a symmetric key.	User
eCryptoGetLastError	Returns the status reported from the last API call.	All
eCryptoGenerateKeyPair	Generates a public key/private key pair.	User
eCryptoDigitalSign	Computes the signature of a data buffer.	User
eCryptoDigitalVerify	Verifies the signature of a data buffer.	User
eCryptoImportPublicKey	Forms a public key object from data.	User
eCryptoImportPrivateKey	Forms a private key object from data.	User
eCryptoExportPublicKey	Exports a public key object.	User
eCryptoExportPrivateKey	Exports a private key object.	User
eCryptoDestroyPublicKey	Deletes a public key.	User
eCryptoDestroyPrivateKey	Deletes a private key.	User

Table 5 - Module Roles and Services

Cryptographic Module Key Management

The module supports four main types of cryptographic keys and CSPs for use by the User role to perform cryptographic operations:

- Data Encrypting Keys (DEKs) perform general encryption/decryption of data, and are DES (legacy purposes only), 3DES (128 or 192 bits) or AES (128, 192, or 256 bits) symmetric keys
- Key Encrypting Keys (KEKs) protect (wrap/unwrap) DEKs only, and are AES (128, 192, or 256 bits) symmetric keys
- DSA private and public keys perform digital signing/verification of data, and are DSA (512, 768, or 1024 bit) asymmetric keys
- Seed and seed keys are used to initialize the Approved RNG. These values are not directly accessible by operators of the module.

Key Entry/Output

DEKs are entered and output encrypted in an AES-wrapped form, as defined in the draft AES Key Wrap Specification [6]. This method provides confidentiality for DEKs outside the module's logical boundary, which in accordance with FIPS 140-2 requirements at Security Levels 1 and 2, may be entered/output in plain text form.

Key Encrypting Keys (KEKs) are entered into the module in plain text and are restricted for use only to wrap or unwrap the DEKs. The KEKs reside in the same process space as the cryptographic module (within the module's logical cryptographic boundary), and the host application must protect the KEKs prior to their entry into the module.

Public DSA keys are entered or output as an ASN.1 SubjectPublicKeyInfo structure, as defined in RFC2459. Private DSA keys are entered or output as a PKCS#8 structure in plain text.

Key Generation

DEKs and DSA parameters, along with public and private keys, are generated using the Approved random number generation algorithm as defined in FIPS 186-2 Appendix 3.1.

IBM has invented a scheme to generate true randomness on a wide range of computer systems. The patented scheme, called the Universal Software Based True Random Number Generator, utilizes random events influenced by concurrent activities in the system (e.g. interrupts, process scheduling, etc). The run time of the algorithm will vary depending of the state of the system at the time of seed generation, and will be dependent on the type of system. The Universal Software Based True Random Number Generator is used to create a true random seed and seed key that are used in the Approved RNG algorithm.

Key Storage

The cryptographic module does not provide persistent storage of operational cryptographic keys (DEKs, KEKs, and DSA keys). All operational cryptographic keys are stored in computer memory only for the lifetime of the module operator session, and are zeroized when the module is shut down or the operator logs out.

However, three special purpose keys are permanently stored in the module:

- Integrity symmetric key and integrity public key – Used for the module integrity self-test at power-up
- Root public key – The module embeds a DSA public key within a X.509 root certificate that is used to verify the User and Crypto Officer authentication certificates when an operator logs into the module.

Key Protection

The management and allocation of memory is the responsibility of the operating system. Each instance of the cryptographic module is self-contained within a process space, and has access only to its own cryptographic keys. Since only the User role has access to cryptographic operations, all operational symmetric and asymmetric keys are associated with the User role.

Key Zeroization

Explicitly identified keys are zeroized when an operator uses the eCryptoDestroyKey, eCryptoDestroyPrivateKey, or eCryptoDestroyPublicKey API calls.

All operational cryptographic secret and private keys and CSPs are zeroized when an operator:

- Logs out of the module, using the eCryptoLogout API call
- Shuts Down the module using the eCryptoShutdown API call
- Powers off the module by unloading it from memory

The following table lists which services have access to keys and CSPs, and the type of access provided by each service:

Service	Key or CSP Accessed	Type of Access
eCryptoShutdown	All DEKs, KEKs, DSA keys, and seeds	Delete
eCryptoLogin	Root public key	Read
eCryptoLogout	All DEKs, KEKs, DSA keys, and seeds	Delete
eCryptoEncrypt	DEK	Read
eCryptoDecrypt	DEK	Read
eCryptoGenerateKey	DEK and seed/seed key	Write
eCryptoWrapKey	KEK and DEK	Read KEK, Read DEK
eCryptoUnwrapKey	KEK and DEK	Read KEK, Write DEK
eCryptoImportKey	KEK	Write
eCryptoDestroyKey	KEK or DEK	Delete
eCryptoGenerateKeyPair	DSA key pair, and seed/seed key	Write
eCryptoDigitalSign	DSA private key	Read
eCryptoDigitalVerify	DSA public key	Read
eCryptoImportPublicKey	DSA public key	Write
eCryptoImportPrivateKey	DSA private key.	Write
eCryptoExportPublicKey	DSA public key.	Read
eCryptoExportPrivateKey	DSA private key	Read
eCryptoDestroyPublicKey	DSA public key	Delete
eCryptoDestroyPrivateKey	DSA private key	Delete

Table 6 - Access to Module Keys and CSPs

Cryptographic Module Self-Tests

When an application loads the cryptographic module into its process space, an initialization routine is called by the operating system before control is handed to the application. This initialization routine automatically executes the power up self tests to ensure the integrity of the module image and correct operation of the cryptographic algorithms.

The first power-up self-test that is executed is the module integrity self-test. The integrity of the module is verified by performing a DSA digital signature verification of the module's file. The signature block used for the integrity self-test contains the module signature and the corresponding integrity public key for signature verification. For additional security, the integrity public key is encrypted with an Approved algorithm (AES), and is decrypted with the integrity symmetric key prior to signature verification. The self test will pass and module initialization will succeed if the signature is valid. If the signature is invalid, the module transitions to an unrecoverable error state and must be unloaded from computer memory and reset. Continuous failure of this test indicates that the file image is corrupted.

Other power-up self-tests include known answer tests for the SHA-1, DES, Triple DES, and AES cryptographic algorithms, and a pairwise consistency test of the DSA algorithm. Should any cryptographic algorithm self-test fail, the module will also transition to an unrecoverable error state and must be reset.

All power-up self tests can be initiated on demand by resetting or power cycling the module (unloading and re-loading the shared library). In addition, the module performs a subset of the power-up self tests on command by the Crypto Officer role - the cryptographic algorithm self tests.

Conditional self tests are performed when symmetric or asymmetric keys are generated. These tests include a continuous random number generator test and pair-wise consistency tests of the generated DSA key pairs.

Cryptographic Module Physical Security

The cryptographic module is implemented completely in software thus the physical security requirements of FIPS 140-2 are not applicable.

References

[1] National Institute of Standards and Technology. May 2001. *Security Requirements for Cryptographic Modules.* Federal Information Processing Standards Publication 140-2.

[2] National Institute of Standards and Technology. November 2001. *Advanced Encryption Standard (AES).* Federal Information Processing Standards Publication 197.

[3] National Institute of Standards and Technology. October 1999. *Data Encryption Standard (DES).* Federal Information Processing Standards Publication 46-3.

[4] National Institute of Standards and Technology. August 2002. *Secure Hash Standard (SHS).* Federal Information Processing Standards Publication 180-2.

[5] National Institute of Standards and Technology. January 2000. *Digital Signature Standard (DSS).* Federal Information Processing Standards Publication 186-2.

[6] National Institute of Standards and Technology. November 2001. *AES Key Wrap Specification (draft).* URL: http://csrc.nist.gov/encryption/kms/key-wrap.pdf

[7] COMMON CRITERIA CERTIFICATION REPORT No. P170, Sun Microsystems, Inc. Trusted Solaris™ Version 8 4/01, UK IT Security Evaluation and Certification Scheme, Cheltenham, Glos GL52 5UF United Kingdom.
URL http://www.commoncriteria.org/certRpt/sun_trusted_CR.pdf

[8] BSI-DSZ-CC-0194-2002 for AIX 5L for POWER V5.2, Program Number 5765-E62 from IBM Corporation, Bundesamt für Sicherheit in der Informationstechnik, Godesberger Allee 185-189 - D-53175 Bonn, Germany.
URL http://www.commoncriteria.org/certRpt/BSI_CR.pdf

[9] National Institute of Standards and Technology and Communications Security Establishment. March 2003. *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*

Notices

AIX, Everyplace, and IBM are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.

Pentium and X-Scale are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.