National Information Assurance Partnership

TM

Common Criteria Evaluation and Validation Scheme Validation Report

IBM Global Security Kit (GSKit)

8.0.14

Report Number: CCEVS-VR-VID10394-2011

Dated: 2012-03-06

Version: 1.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6740
Fort George G. Meade, MD 20755-6740

ACKNOWLEDGEMENTS

Validation Team

Jim Brosey
Orion Security
Fort Meade, Maryland

Jandria S. Alexander
Aerospace
Fort Meade, Maryland

Vicky Ashby
The MITRE Corporation
McLean, Virginia

Evaluation Team

Alejandro Masino, Trang Huynh, Courtney Cavness
atsec Information Security Corporation
Austin, Texas

Table of Contents

1. EXECUTIVE SUMMARY

This report documents the NIAP validators' assessment of the evaluation of the IBM Global Security Kit (GSKit) 8.0.14. It presents the evaluation results, their justifications, and the conformance results. This validation report is not an endorsement of the information technology (IT) product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

The evaluation was performed by the atsec information security corporation, and was completed during January 2012. atsec information security corporation is an approved NIAP Common Criteria Testing Laboratory (CCTL). The evaluation was conducted in accordance with the requirements of the Common Criteria for Information Technology Security Evaluation, version 3.1. The information in this report is largely derived from the Evaluation Technical Report (ETR) and associated test report, both written by the CCTL. The evaluation determined the product to be Common Criteria Version 3.1 Revision 3, Part 2 extended, Part 3 conformant, and to meet the requirements of EAL4 augmented by ALC_FLR.1.

GSKit is a set of tools and C/C++programming interfaces that can be used to add secure channels using the TLS protocol to TCP/IP applications (products). It provides the cryptographic functions, the protocol implementation, and key generation and management functionality for this purpose.

GSKit is only for IBM internal use and is not offered for sale as a standalone product, i.e., it is designed for the use in other IBM products only.

GSKit encapsulates the IBM Crypto for C (ICC) cryptographic software module. ICC Version 8.0 has been validated under the Federal Information Processing Standard (FIPS) 140-2 for an overall Security level 1 with certificate number 1433. The module provides a variety of FIPS 140-2validated cryptographic algorithms, as well as some algorithms that are not standardized by the cryptographic algorithm validation program. The TOE encapsulates ICC for software-based cryptographic functions and key generation.

The validation team agrees that the CCTL presented appropriate rationales to support the Results of Evaluation IBM Global Security Kit 8.0.14 is complete and correct.

2. IDENTIFICATION

The Common Criteria Evaluation and Validation Scheme (CCEVS) is a National Security Agency (NSA) effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) using the Common Evaluation Methodology (CEM) for Evaluation Assurance Level (EAL) 1 through EAL 4 in accordance with National Voluntary

Laboratory Assessment Program (NVLAP) accreditation granted by the National Institute of Standards and Technology (NIST).

The NIAP Validation Body assigns validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desiring a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP's Validated Products List.

Table 1 provides information needed to completely identify the product, including:

- The Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated;
- The Security Target (ST), describing the security features, claims, and assurances of the product;
- The conformance result of the evaluation;
- The Protection Profile to which the product is conformant;
- The organizations and individuals participating in the evaluation.

Table 1: Evaluation Identifiers

Item	Identifier
Evaluation Scheme	United States NIAP Common Criteria Evaluation and Validation Scheme
Target of Evaluation	IBM Global Security Kit (GSKit) 8.0.14
Protection Profile	None.
Security Target	*IBM Global Security Kit (GSKit) 8.0.14 Security Target, Version 3.5*
Evaluation Technical Report	*Evaluation Technical Report for a Target of Evaluation* *IBM Global Security Kit version 8.0.14* *ETR Version 1.0 as of 2012-01-20*
Conformance Result	CC V3.1, Part 2 extended, Part 3 conformant, EAL 4 augmented by ALC_FLR.1
Sponsor	International Business Machines (IBM)
Developer	International Business Machines (IBM)
Evaluators	Alejandro Masino, Trang Huynh, Courtney Cavness atsec information security corporation
Validators	Jim Brosey Orion Jandria S. Alexander The Aerospace Corporation Vicky Ashby The MITRE Corporation

3. CLARIFICATION OF SCOPE

This section details the scope of the evaluation and describes the logical and physical boundaries of the TOE.

3.1. Physical Scope

The physical scope of the evaluated configuration consists of:

- Software:
  - IBM Global Security Kit (GSKit) 8.0.14.21
- Hardware
  - None.
- User documentation:
  - GSKCapiCmd User's Guide Version 8
  - Global Security Kit Common Criteria Mode Operating Guidance Version 8
  - IBM Global Security Kit Key Management for C Programmer's Guide Version 8
  - Global Security Kit Install and Packaging Guide Version 8
  - IBM Global Security Kit Secure Socket Layer for C Programmer's Guide Version 8
  - IBM Global Security Kit Certificate Validation and Trust Policy Design Version 8
  - Global Security Kit Delivery Procedure Additional Guidance 8

The Global Security Kit Common Criteria Mode Operating Guidance is the authoritative documentation that must be used in order to place the TOE into the evaluated configuration.

3.2. Logical Scope

The description of the security features of the product are described in further details in section 4 of this document. In summary, these functions are:

- Secure channels

The TOE allows consuming applications to implement TLS functionality. (The SSL protocol versions supported by GSKit are not available in the evaluated configuration). GSKit supports both TLS client and server functionality.

The TLS functionality is offered via an API (called SSL API) for TLSv1, 1.1, and 1.2 (as defined in [TLSv1], [TLSv1.1], and [TLSv1.2], with [TLS_AES] support for TLSv1 and 1.1, and with certain extensions from [RFC6066]).

The TOE supports a wide variety of cipher suites specified in [ST] for the encryption of payload, enforces server authentication, and optionally can enforce client authentication; including certificate validation with optional revocation checking via CRLs and/or OCSP.

The TOE implements Certificate Validation and Revocation Checking conformant with [RFC5280]. The TOE can query OCSP responders in the operational environment using OCSP as defined in [RFC2560] and [RFC6277], or using the lightweight OCSP profile defined in [RFC5019]. The TOE can also retrieve and validate X.509 version 1 and 2 CRLs from the operational environment using LDAP, flat files or HTTP.

- Key and certificate generation and management

The TOE implements both a key management API and command line interface (CLI) to generate keys and certificate requests, and manage (import, export, define the trust status, etc.) keys and certificates. Key data (i.e. keys, certificates, and related information) is stored in a so-called keystore, a file stored in the operational environment. The TOE ensures by cryptographic means that the integrity and, where appropriate, the confidentiality of the data stored in the keystore is protected. Alternatively, PKCS#11 devices or MSCAPI/MSCNG cryptographic service providers can be used for key and certificate storage and for performing cryptographic primitives.

- Self-tests

The cryptographic module within GSKit implements self-tests as required by [FIPS140-2].

4. SECURITY POLICY

4.1. Secure Channel

The TOE offers a secure channel for the confidentiality and integrity protection of data transmitted over that channel.

The secure channel functionality of the TOE is available through the TOE's SSL API. The operational environment is responsible to limit access to this API to the users and roles in the operational environment authorized to use those functions. When demanded by the product that integrates the TOE, the secure channel is established by GSKit using the TLS protocol. Where the standards defining the TLS protocol suites leave several options, the TOE does not always provide all possible options. These restrictions are described as follows.

The SSL protocol has been excluded from the TOE as mandated by the Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program. The use of only TLS is ensured by initializing the TOE in FIPS mode. This has to be done by the operational environment.

The TLS connections support mandatory server authentication and optional (configurable as mandatory) client authentication. Total anonymity mode as defined for TLS, meaning that not even the server of the connection has to authenticate, cannot be used in the evaluated configuration.

For authentication, X.509 [X.509] certificates are used; version 1, 2, and 3 certificates are supported for root and end user certificates, and version 3 certificates are supported for intermediate CA certificates.

The following ciphersuites are supported by the TOE (see chapter 1.3.2.2 for further details on cryptographic algorithms):

a) 3DES-based:
   - TLS_RSA_WITH_3DES_EDE_CBC_SHA = { 0x00,0x0A } [TLSv1]
   - TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA = { 0xC0, 0x08 } [RFC4492]
   - TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA = {0xC0, 0x12} [RFC4492]

b) AES CBC- and GSM-based (RSA):
   - TLS_RSA_WITH_AES_128_CBC_SHA = { 0x00, 0x2F } [RFC3268]
   - TLS_RSA_WITH_AES_256_CBC_SHA = { 0x00, 0x35 } [RFC3268]
   - TLS_RSA_WITH_AES_128_CBC_SHA256 = { 0x00, 0x3C } [TLSv1.2]
   - TLS_RSA_WITH_AES_256_CBC_SHA256 = { 0x00, 0x3D } [TLSv1.2]
   - TLS_RSA_WITH_AES_128_GCM_SHA256 = { 0x00, 0x9C } [RFC5288]
   - TLS_RSA_WITH_AES_256_GCM_SHA384 = { 0x00, 0x9D } [RFC5288]
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA = { 0xC0, 0x09 } [RFC4492]
   - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA = { 0xC0, 0x0A } [RFC4492]
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 = {0xC0,0x23} [RFC5289]
   - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 = {0xC0,0x24} [RFC5289]

c) AES CBC- and GCM- based (ECDHE):
   - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 = {0xC0,0x2F} [RFC5289]
   - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 = {0xC0,0x27} [RFC5289]
   - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 = {0xC0,0x30} [RFC5289]
   - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 = {0xC0,0x28} [RFC5289]

d) Suite B compliant profile cipher suites for TLS 1.2 (per [RFC6460]:
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 = {0xC0,0x2B} [RFC5289]
   - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 = {0xC0,0x2C} [RFC5289]

If no common ciphersuite is found, no TLS connection is set up, i.e. a session with the ciphersuite SSL_NULL_WITH_NULL_NULL is not supported.

The check that a certificate is valid and has not been revoked can be done with the help of CRLs or OCSP. CRLs have to be conformant to X.509 [X.509], version 1 and 2 CRLs are supported. The use of CRLs is not mandatory but checking the revocation status for certificates is recommended in the TLS standards. If CRLs are used, the environment has to provide an LDAP server, or an HTTP server, or the CRL via a flat file available on the underlying system. Current and correct CRLs must be provided by the environment. If CRLs are used and no CRL can be retrieved, the validation check will fail and the certificate's revocation status will be considered undetermined.

As an alternate means to CRL validation, the TOE can use OCSP as defined in [RFC2560] (and augmented by [RFC6277]) or [RFC5019] (lightweight OCSP profile) to obtain the revocation status of a certificate. The TOE can be configured to use CRL checking as a fall-back if no valid OCSP response can be obtained from the operational environment.

The use of CRL checking and/or OCSP responses in CC mode is optional.

The TOE furthermore supports the following TLS extensions:

- Server Name Indication [RFC6066]
- Certificate Status Request [RFC6066]
- Signature and Hash Algorithms as part of the TLS 1.2 specification [TLSv1.2]

GSKit supports compression as specified in [RFC3749].

Please note: Traditionally and when operated outside the evaluated configuration, GSKit supports versions of SSL in addition to TLS. This is the reason why – in the guidance and this ST – certain interfaces and functions offered by the TOE, such as the SSL API and the SSL environment initialization function, carry the name "SSL" instead of TLS, although their operation in the evaluated configuration (in "CC mode") is nevertheless restricted to the TLS protocol versions.

4.2. Cryptographic Operations

The TOE offers generation of symmetric keys, generation of asymmetric key pairs, symmetric encryption/decryption, asymmetric encryption/decryption, generation/verification of digital signatures, data authentication, secure message digest algorithms, and random number generation. These cryptographic services are provided by the ICC component and are used by the TOE for TLS.

The ICC module that is part of the TOE has been FIPS 140-2 level 1 [FIPS140-2] validated under Certificate No. 1433 [ICC1433]. A subset of its cryptographic algorithms used within the TOE is FIPS Approved and/or NIST-allowed;

The TOE uses only FIPS-Approved and/or NIST-allowed cryptographic algorithms for the cipherspec part of the TLS cipher suites and uses non-FIPS approved algorithms for other functions, like RSA key exchange for a TLS session, when operated in compliance with this Security Target.

The TOE's DH and ECDH key establishment mechanisms, as well as MD5 and MD2 are not FIPS 140-2 [FIPS140-2] approved. DH and ECDH key establishment, and MD5, are required for key exchange during the setup of TLS sessions with corresponding cipher suites. MD5 and MD2 are used for the verification of certificates that have not been generated by the TOE. None of these functions are part of the cipherspecs of the TLS cipher suites.

The use of only FIPS-Approved and/or NIST-allowed cryptographic algorithms for the cipherspecs of TLS cipher suites (which leads to the supported cipher suites listed in [ST] chapter 1.3.2.1) is assured by initializing the TOE in FIPS mode. This initialization has to be done by the operational environment.

The TOE then only provides the cipher suites listed above and initializes its ICC component also in FIPS mode which assures that only the FIPS approved random number generator is used.

Alternatively to using the cryptographic mechanisms implemented in the TOE, the TOE is able to use cryptographic modules in the operational environment via PKCS#11 and, on Windows, hardware or software cryptographic service providers via the MSCAPI/MSCNG, for cryptographic primitives. MSCNG is available only on Windows versions starting with Windows Vista and Server 2008, and is currently only used by GSKit (if configured) to perform ECDSA operations and store related keys and certificates.

4.3. Self-tests

GSKit offers self-tests for the ICC component: some of ICC's cryptographic functions and the integrity of ICC can be tested. The self-tests have been analyzed as part of the FIPS 140-2 level 1 [FIPS140-2] validation.

The ICC self-tests are automatically executed upon the first call to the SSL environment initialization command of the TOE. This command (function) must be called by the application that deploys the TOE to initialize the environment before setting up a trusted channel. If the self-tests fail, the SSL environment initialization will fail and the user will be notified.

Furthermore, the TOE checks the integrity of the trust status of certificates and of certificate request data when accessing data in the keystore and will notify the user of errors if these checks fail.

4.4. Key Management

GSKit provides a local command line interface and an API for key generation and management of asymmetric key pairs as well as generation and management of certificates, including the storage of any such key material. GSKit uses a native keystore (also referred to as "CMS keystore") to store key data, or can alternatively access external hardware or software key stores via PKCS#11, PKCS#12, and (on Windows) the MSCAPI and MSCNG. For the generation of certificates, certificate requests conformant to PKCS#10 are used.

GSKit's native CMS key stores come in different versions:

- CMS V3 is deprecated and not supported in the evaluated configuration
- CMS V4 is a proprietary format
- CMS V5 is, in fact, implemented according to the PKCS#12 standard

For key and certificate management, and when initializing an SSL environment, GSKit always requires the user to specify the keystore (or cryptographic module) to be used. As a result, multiple keystores and modules can exist in the operational environment, allowing the user to specify which one is being used by GSKit. GSKit will always operate on only one keystore or module at a time.

GSKit supports import- and export of private elliptic curve keys per [RFC5915].

4.5. Protection of the stored data

For CMS keystore contents, GSKit implements password protection for confidentiality and integrity. The password must be provided before any access to the keystore database. Each certificate, certificate/private key pair, or certificate request data is stored in one so-called data record.

Both in CMS V4 and V5, GSKit uses TDEA (Triple DES) and HMAC SHA-1 to provide the protection (pbeWithSHAAnd3KeyTripleDESCBC). CMS V4 only encrypts and hashes keys in this fashion, an adds an additional HMAC-SHA-1 hash that is generated over all records, using the user-supplied password as the HMAC Key. This hash is then stored in the keystore, in a special record that exists once per keystore file, a so-called header record. CMS V5 encrypts and hashes all records in the keystore individually, and does not apply an additional overall hash to the keystore.

The CMS keystore password protection does not limit unauthorized users from reading and writing the keystore file. GSKit relies on the underlying operating system for these protections, but access to sensitive data is effectively controlled because all sensitive data in the keystore is encrypted, all records hashed, and the index to all records is hashed. This ensures that any modification to the file is detectable. If tampering is detected, GSKit will deny access to the keystore (the behavior is similar to receipt of an incorrect password).

The protection of data stored in non-native key stores is provided by the respective keystore, i.e. the operational environment.

If a cryptographic module in the operational environment requires a PIN or password, GSKit can temporarily store the user-supplied PIN in memory during run-time and hand it down to the module when required. This is not, however, a requirement. It is possible that a cryptographic module provides its own trusted path for user authentication, in which case an “unlocked” module may be available to GSKit without GSKit having to provide a credential to the module. GSKit therefore does not require its users to provide a PIN or password for the use of external modules.

5. ASSUMPTIONS

The evaluation makes the following assumptions on the TOE environment and personnel managing the TOE:

- It is assumed that the operational environment will provide mechanisms to audit all security-relevant TSF actions initiated via the TOE’s APIs or performed on keystores through the command line.

- It is assumed that the TOE and its operational environment will be operated in a configuration compliant with the FIPS validated configuration of the ICC component as described in the FIPS 140-2 Level 1 Security Policy for ICC [ICCSEC].

- Those responsible for the administration of the TOE and the operational environment are competent and trustworthy individuals, capable of managing the TOE and the operational environment and the security of the information it contains.

- Those who integrate the TOE as part of a larger product or system are assumed to follow the guidance provided with the TOE with respect to the configuration of the TOE, the use of TOE functions and interfaces and the protection of the TOE code and data. They are trustworthy and do not try to subvert or bypass TOE security functions.

- The operational environment provides public key infrastructure services not supplied by the TOE (such as, registration authorities, processing of certificate requests by a certification authority, issuance of CRLs, operation of OCSP responders, etc.), as needed for the intended usage of the application integrating the TOE, in a reliable and trustworthy fashion.

6. ARCHITECTURAL INFORMATION

GSKit is a set of tools and C/C++ programming interfaces that can be used to add secure channels using the TLS protocol to TCP/IP applications (products). It provides the cryptographic functions, the protocol implementation, and key generation and management functionality for this purpose.

GSKit encapsulates the IBM Crypto for C (ICC) cryptographic software module. ICC Version 8.0 has been validated under the Federal Information Processing Standard (FIPS) 140-2 [FIPS140-2] for an overall Security level 1 with certificate number 1433. The module provides a variety of FIPS 140-2validated cryptographic algorithms, as well as some algorithms that are not standardized by the cryptographic algorithm validation program. The TOE encapsulates ICC for software-based cryptographic functions and key generation.

The following interfaces are offered to the user:

- SSL API: this is the main API of the product. For the evaluated configuration of the TOE, only secure connections using TLS are supported (this is enforced by operating the TOE in FIPS mode).

- Key management API and command line interface (CLI): through this API and the CLI, key and certificate generation and management functionality (like validation or deletion of certificates) can be accessed. A product that uses the TOE shall access the keystore through these interfaces only.

The TOE encapsulates ICC for software-based cryptographic functions and key generation.

Certificate validation may be done with the help of OCSP or CRLs. If CRLs are to be used for certificate validation, LDAP or HTTP resources or a flat file containing the revocation list must be provided by the operational environment. Likewise, for the usage of OCSP, an OCSP responder must be provided by the environment.

Alternatively to using the cryptographic and key storage mechanisms implemented in the TOE, the TOE is able to use cryptographic modules in the operational environment via PKCS#11 and, on Windows, via the MSCAPI/MSCNG. Such modules must be FIPS 140-1 or -2 validated. Key management tools other than the CLI, e.g., iKeyman, are not part of the TOE.

7. PRODUCT TESTING

7.1. Sponsor Testing

7.1.1. TOE test configuration

The developer's test environment for the TOE comprises the 14 combinations of testing platforms and TOE binaries listed in [ST] and [TEP] as part of the evaluated configuration. In addition, the developer provided an LDAP server and an OCSP responder in the operational environment. In general tests executed against the API follow the steps outlined in the guidance to operate the TOE in its evaluated configuration, as appropriate.

7.1.2. Testing approach

The developer employs a test strategy where basic function testing is executed in an automated fashion after the build process. This testing serves as build verification testing and is not necessarily performed in the evaluated configuration. In addition, and as primary testing to achieve test coverage and depth for the evaluation, the developer maintains and uses automated function verification tests (FVT). While partially derived from the build verification test suites, those tests allow execution in the evaluated configuration as described above. Tests are primarily implemented using the TOE's APIs – the developer was able to demonstrate that the command line interface provided by the TOE is merely a wrapper for the Key Management API and sufficiently covered by tests for the API. Direct test coverage is achieved for the complete TSFI, subsystem internal interfaces are tested primarily indirectly and to a smaller extend directly. The cryptographic algorithms that comprise the ICC crypto module have been FIPS 140-2 validated, therefore this validation provides appropriate test coverage.

In addition, the developer successfully exercised the NIST PKITS test suite on the TOE.

7.1.3. Testing results

The overall test results provided by the developer match the expected test results and are able to demonstrate that the security functions and interfaces behave as designed and as documented in the evaluation evidence.

7.1.4. Test coverage & depth

The developer provides tests for all TSF interfaces, all TSF aspects identified in the TOE Summary Specification and all Security Functional Requirements as defined in the [ST]. The amount of testing performed by the developer for external and internal interfaces of the TOE is sufficient.

7.2. Evaluator Testing

7.2.1. TOE test configuration

Functional test

The test environment used for the independent testing activities consisted of the systems shown in the table below:

ID	Platform	Architecture	Hostname/ IP address	TOE binaries	Purpose
WinXP	Microsoft Windows XP SP3	Intel 64-bit		Windows 32-bit	Testing suite calibration.
Win2008	Microsoft Windows Server 2008 SP2	VMWare ESXI 64-bit	gsk8test.win2008 10.4.1.109	Windows64-bit	TOE testing
RHEL5	Red Hat EnterpriseLinux 5	VMWare ESXI 64-bit	gsk8test.rhel 10.4.1.101	Linux64-bit	TOE testing
Ubuntu	Ubuntu Linux	VMWare ESXI 64-bit	gsk8test.helper 10.4.1.104	Not applicable	Helper system

The test configuration for the machines used for TOE testing matches the supported platforms specified in the [ST], and was chosen to test both 32-bit and 64-bit versions of the TOE running on Linux and Windows operating systems.

The test cases on the TOE were executed using the evaluated configuration, as specified in the [ST] and the [GDO], as appropriate. Restrictions to the underlying operating system as mandated by the ST for the operational environment were not applied, since tests were performed in an isolated environment and those configurations did not affect the operations of the TOE in any manner.

The operational environment used for testing also included a helper system (Ubuntu) providing:

- An OpenLDAP server for the provision of CRLs using the LDAP.
- A Web server for the provision of CRLs through HTTP/HTTPS.
- The OCSP server implemented in OpenSSL for the provision of OCSP request services.
- OpenSSL 1.0.0e as an alternative implementation of SSL/TLSv1 for interoperability testing.

Penetration test

The TOE was used in the same configuration as for independent evaluator testing. Tests were conducted by exercising external interfaces of the TOE, direct manipulation of internal files used by the TOE to store TSF data, and code analysis.

7.2.2. Evaluator tests performed

Functional test

Subset size chosen

All security functions defined in the TOE summary specification were addressed. Emphasis was made on the security functionality aspects related to key management, TLS secure channel establishment, enforcement of constraints imposed by the TOE in CC and FIPS mode, secure state preservation, and interoperability with third party products. Since all cryptographic algorithms supplemented by the TOE and used in FIPS mode have already been FIPS-validated, the independent test only focused on the correct integration of the ICC module in the TOE.

The following table summarizes the test cases that were performed during the independent testing:

Test case	**Description**
Developer Tests	Functional Verification Tests (FVT) in the CMS, SSL and KM subsystems.
TLS interoperability	Verify interoperability between the TOE and openssl with certificates created by the TOE.
X.509 interoperability	Exchange of certificates and establishment of TLS sessions considering: • Certificates created by openssl in TOE. • Certificates generated in TOE and signed by openssl. • Certificates generated and signed by TOE. • Certificates generated and signed by TOE using MSCAPI/MSCNG certificate stores.
SSL environment variables	Verification that the GSK_PROTOCOL_TLSV10 variable environment does not make any effect in CC mode.
TLS cipher suite verification	Verification of the TLS session: TLS cipher suite verification • CC mode ON, all TLSv10, TLSv11 and TLSv12 allowed ciphersuites • CC mode OFF, all SSLv2, SSLv3, TLSv10, TLSv11 and TLSv12allowed cipher suites • All possible combinations of certificates (EC, RSA) and ciphersuites
Key data zeroization verification	Verifies the zeroization of the secure socket handle.
Secure state on failure verification	Verify the TOE cannot be started after the ICC library is corrupted.
SSLv2 and SSL v3.0	Verifies these protocols are disabled in CC mode, including interoperability with the Firefox browser.

negotiation	
Keystore integrity checking	Verifies that the TOE detects corruption of the keystores.
MD5 certificate signing requests	Verifies the TOE does not use MD5 for signing certificate requests generated by openssl.
TLS buffer overflows	Verifies that the TOE can correctly handle malformed certificates in the following cases: • Acting as a client in a TLS session. • Acting as a server in a TLS session. • Importing certificates into keystores.
OCSP responder	Verifies that the TOE can handle properly: • Valid and revoked certificates. • Unexpected response error messages. • Unexpected length response messages to cause buffer overflows.

Selection criteria
Tests were devised for functional areas that triggered the evaluation team's attention during evaluation of the ST and the design and guidance documentation, and in cases where the evaluation team decided that the assurance provided by this evaluation would benefit from augmentation of the test coverage and depth provided by the developer.

Security functions tested
Tests covered a broad range of security functionality, especially key management, keystore integrity, TLS connection establishment including path validation and error handling, and correct handling of self-test errors received from ICC subsystem. All TSFI were addressed adequately by tests.

Developer tests performed

The sample strategy applied to the repetition of developer tests was to cover a wide range of developer tests to gain additional confidence in the testing conducted by the developer. This approach resulted in the decision to repeat all automated FVT tests implemented by the developer, ensuring broad coverage while limiting the efforts to an appropriate level. The tests covered all security functions provided by the TOE and exercised the two API interfaces and the SSL/TLS network interface.

Penetration test

Vulnerability assessment

To obtain a candidate set of vulnerabilities, the evaluators performed the following actions.

1. First, the evaluator took into account any findings raised while evaluating other parts of the evidence such as the Security Target, guidance, design, and test documentation that could be considered potential vulnerabilities. No such findings were noted.

2. Second, the evaluators searched the following public sources

    - National Vulnerability Database with key words such as gskit, icc, tls, ssl, ocsp, crl, x.509, ans.1, pkcs, mscapi, mscng, cipher suites, certificate, openssl, cryptoapi.

    - Securnia Advisory and Vulnerability Database with key words such as gskit, icc, tls, ssl, ocsp, crl, x.509, ans.1, pkcs, mscapi, mscng, cipher suites, certificates.

3. Third, the evaluator reviewed several papers and information available on the Internet regarding vulnerabilities in the SSL and TLS protocols.

4. Together with the RFCs referenced by other parts of the evaluation evidence (i.e., security target, high-level design, and low-level design), the evaluator also considered the NIST News & Events page for any relevant announcements of security/cryptography-related.

5. Lastly, the evaluator considered the vulnerability analysis provided by the developer. This document is maintained by the developer which shows the vulnerabilities discovered on similar products (e.g., OpenSSL), and how the developer have addressed it in the TOE.

The evaluators then gathered a set of potential vulnerabilities which resulted from the above actions. The evaluator analyzed each of the potential vulnerabilities to determine whether they could apply to the TOE. The criteria used for this analysis are described in detail in the evaluation report of vulnerability assessment. Based on this analysis, the evaluator derived a list of potential vulnerabilities as candidates for penetration test which was performed as part of the CCTL's independent test. Penetration tests were executed against aspects of security channel (buffer overflows in certificate interpretations and OCSP responses, acceptance of SSLv2 and SSLv3 connections, and OCSP replay attacks) and key management, (generation of certificates with unsupported hashing algorithms, keystore integrity). The non-existence of other potential vulnerabilities in the security functions security channel, key management, secure state was determined by code analysis as an alternative to penetration testing.

7.2.3. Summary of Evaluator Test Results

Functional test

The test results demonstrate that the TOE behaves as expected and that the developer tests conducted were repeatable and contribute to this demonstration.

Penetration test

The TOE shows no exploitable vulnerabilities in its intended environment.

8. DOCUMENTATION

8.1. Product Guidance

The guidance documentation examined during the course of the evaluation and delivered with the TOE is as follows:

- GSKCapiCmd User's Guide Version 8
- Global Security Kit Common Criteria Mode Operating Guidance Version 8
- IBM Global Security Kit Key Management for C Programmer's Guide Version 8
- Global Security Kit Install and Packaging Guide Version 8
- IBM Global Security Kit Secure Socket Layer for C Programmer's Guide Version 8
- IBM Global Security Kit Certificate Validation and Trust Policy Design Version 8
- Global Security Kit Delivery Procedure Additional Guidance 8

8.2. Evaluation Evidence

The following tables identify the additional documentation submitted as evaluation evidence by the vendor. With the exception of the Security Target, these documents may be proprietary and not available to the general public.

Design Documentation	**Version**	**Date**
GSKit doxygen documentation		2011-12-20
GSKit Functional Specification and Partial Representation Correspondence	1.6	2011-08-23
GSKit High Level Design and Partial Representation Correspondence	1.8	2011-08-23
IBM Crypto for C (ICC) Version 8.0.0 Design Document		2010-04-21
IBM Crypto for C (ICC) Version 8.0.0 FIPS 140-2 Non-Proprietary Security Policy	1.2	2010-12-06
GSKit doxygen Low level design		2011-08-05
Certificate Validation and Trust Policy Design	8	2011-07-19
GSKit V8 Security Architecture	0.2	2011-03-01

Configuration Management Documentation	**Version**	**Date**
GSKit Authority List		2010-08-12
GSKit Authority List of Permissions		2005-02-14
Bugzilla samples (3)		2010-05-26
CMVC 5.0 InfoCenter		2003-12-19

GSKit_SSL Change History	1.77.15.16	2011-09-01
GSK List of TOE Superusers		2010-08-10

Delivery and Operation Documentation	Version	Date
IBM Global Security Kit Key Management for C Programmer's Guide	8	2011-07-19
IBM Global Security Kit Secure Socket Layer for C Programmer's Guide	8	2011-12-20
IBM Global Security Kit GSKCapiCmd User's Guide	8	2011-11-28
IBM Global Security Kit Common Criteria Mode Operating Guidance	8	2012-01-11

Lifecycle Documentation	Version	Date
Alternative Site Visit Plan	-	2004-10-26
ATT Network Client – Overview	25	2010-08-12
Global Security Kit Build Environment	8	2011-01-18
Main page from BuildmasterWiki		2010-05-08
ADL Business Process Management – Business Process Reporting Procedure	Rev. 1	2010-05-26
Build Process GSKit		2010-07-22
buildinfl.h		2004-09-21
Workplace Security Walk-Arounds		2010-08-01
GSKit Final CI List		2012-01-10
Corporate Security Incident reporting		2011-08-03
GSKit Security Kit Delivery Procedure Additional Guidance	8	2010-08-08
IBM Global Security Kit Install and Packaging Guide	8	2011-02-28
Photo Session		2004-11-25
Information Technology Security Standards	8.0	2010-07-15
Security and Use Standards for IBM Employees	11.0	2009-06-01
Site Visit Report: Interview of the development team Checklist 4		2004-11-19
GSKit L3 Defect Lifecycle Procedure	Rev.2	2006-05-05

GSKit Level 3 Support Department Operations Manual (DOM)	Rev.1	2006-05-12
Global Security Kit Life Cycle Support	1.7	2011-08-08
Corporate Instruction LEG 116		2010-01-21
Corporate classification and control of IBM information – Frequently Asked Questions		2009-04-15
GSKit CI List non-Gold Coast development		2011-11-21
Physical security Check / Assets Confirmation		2011-07-01
Various IBM procedures: TIV-PROCD-0061, TIV-PROCD-0060, TIV-PROCD-0068 TIV-PROCD-0057, TIV-PROCD-0067		2011-07-01
Security Asset and Risk Management (SARM)	5.0	2010-04-12
IBM Security Manual	57	2009-08-17
Software Group IPD Guide	Rev. 10	2010-07-19
IT Tools User ID management		2006-08-16
IBM XL C/C++ Enterprise Edition for AIZ, v9.0 Compiler Reference	9.0	2010-09-23
IBM XL C/C++ Enterprise Edition for AIZ, v9.0 Language Reference	9.0	2010-09-23

Test Documentation	**Version**	**Date**
Test Verification Test Logs		2011-12-22
GSKit Test case source code		2011-07-19
GSKit Test Plan		2011-08-05
GSKit Test case scripts for FVT		2011-07-19
Developer test report summary		2011-08-02
GSKit Independent Testing Plan	1.6	2012-01-12
GSKit Independent Test Suite		2011-01-18

Security Target	**Version**	**Date**
IBM Global Security Kit Version (GSKit) 8.0.14 Security Target	3.5	2011-12-06

9. RESULTS OF THE EVALUATION[1]

The evaluation team determined the product to be CC Part 2 extended, CC Part 3 conformant, and to meet the requirements of EAL 4 augmented by ALC_FLR.1. In short, the product satisfies the security technical requirements specified in IBM Global Security Kit (GSKit) 8.0.14 Security Target.

10. VALIDATOR COMMENTS

11. SECURITY TARGET

IBM Global Security Kit Version (GSKit) 8.0.14 Security Target, version 3.5, 2011-12-06

12. LIST OF ACRONYMS

AES	Advanced Encryption Standard
API	Application Programming Interface
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Evaluation Testing Laboratory
CEM	Common Evaluation Methodology
CLR	Certificate Revocation List
CMS	Certificate Management System
DES	Data Encryption Standard
DH	Diffie-Hellman
EAL	Evaluation Assurance Level
ECDH	Elliptic curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithms

[1] The terminology in this section is defined in CC Interpretation 008, specifying new language for CC Part 1, section/Clause 5.4.

ETR	Evaluation Technical Report
FIPS	Federal Information Processing Standard
FVT	Functional Verification Test
GSKit	Global Security Kit
HMAC	Hash-Based Message Authenticaiton
HTTP	Hypertext Transfer Protocol
ID	Identifier
ICC	IBM Crypto for C
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
MSCAPI	Microsoft CryptoAPI
MSCNG	Microsoft Cryptographic API
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards & Technology
NSA	National Security Agency
NVLAP	National Voluntary Laboratory Assessment Program
OCSP	Online Certificate Status Protocol
PKCS	Public-Key Cryptography Standard
PIN	Personal Identification Number
PP	Protection Profile
RFC	Request for Comments
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer

ST	Security Target
TCP/IP	Transmission Control Protocol/Internet Protocol
TDEA	Triple Data Encryption Standard
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functions
TSFI	TOE Security Function Interface

13. BIBLIOGRAPHY

Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, Version 3.1.

Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, Version 3.1.

Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, Version 3.1.

Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, Version 3.1.

Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology, Version 3.1.

[ST]	IBM Global Security Kit 8.0.14 Security Target, v3.5, 2011-12-06
[TEP]	GSKIT Test Plan, 2011-08-25
[GDO]	IBM Global Security Kit Common Criteria Mode Operating Guidance Version 8, 2012-01-11
[ICC1433]	FIPS 140-2 Validation Certificate No. 1433. IBM Crypto for C by IBM Corporation (When operated in FIPS mode). Obtained on 2010-12-02 from http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140crt/140crt1433.pdf.
[ICC1433]	IBM Crypto for C (ICC) Version 8.0.0 FIPS 140-2 Non-Proprietary Security Policy, version 1.2, September 30, 2010. Obtained on 2010-12-02 from http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1433.pdf
[FIPS140-2]	FIPS PUB 140-2: SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES, Issued May 25, 2001, including CHANGE NOTICES (12-03-2002)

[X.509] ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8: INFORMATION TECHNOLOGY OPEN SYSTEMS INTERCONNECTION - THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS

[TLSv1] T. Dierks, C.Allen: The TLS Protocol Version 1.0; RFC 2246, January 1999.

[TLSv1.1] RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1. April 2006.

[TLSv1.2] RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. August 2008.

[TLS_AES] P. Chown: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS); RFC 3268, June 2002..

[RFC2560] RFC 2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. June 1999

[RFC3268] RFC 3268: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS). June 2002.

[RFC3280] RFC 3280: Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL), obsoletes RFC 2459, April 2002.

[RFC3749] RFC 3749: Transport Layer Security Protocol Compression Methods. May 2004.

[RFC4492] RFC 4492: Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS). May 2006.

[RFC5019] RFC 5019: The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments. September 2007.

[RFC5280] RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. May 2008.

[RFC5288] RFC 5288: AES Galois Counter Mode (GCM) Cipher Suites for TLS. August 2008.

[RFC5289] RFC 5289: TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM). August 2008.

[RFC5915] RFC 5915: Elliptic Curve Private Key Structure. June 2010.

[RFC6066] RFC 6066: TLS Extension Definitions. January 2011.

[RFC6277] RFC 6277: Online Certificate Status Protocol Algorithm Agility. June 2011.

[RFC6460] RFC 6460: Suite B Profile for Transport Layer Security (TLS). September 2011.