RSA BSAFE® CNG Cryptographic Primitives Library 1.0 FIPS 140-2 Validation Security Policy

This document is a non-proprietary security policy for the RSA BSAFE CNG Cryptographic Primitives Library 1.0 (BSAFE CNG Cryptographic Primitives Library). This document may be freely reproduced and distributed whole and intact including the copyright notice.

Contents:

1 Preface

This document is a non-proprietary security policy for the BSAFE CNG Cryptographic Primitives Library. For the remainder of this document, the BSAFE CNG Cryptographic Primitives Library will be referred to as the Module. This Module is contained in RSA BSAFE CNG 1.0 (BSAFE CNG) from RSA, the Security Division of EMC (RSA).

This security policy describes how the Module meets the security requirements of FIPS 140-2, and how to securely operate it. This policy is prepared as part of the Level 1 FIPS 140-2 validation of the Module. The Module provides the primitive provider functionality through the dynamic link library `BSAFEPrimitives.dll`.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 - Security Requirements for Cryptographic Modules) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available from this website: http://csrc.nist.gov/.

1.1 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 on BSAFE CNG and the entire RSA BSAFE product line is available at:

- http://www.rsa.com/ for information on the full line of products and services.
- http://www.rsa.com/node.aspx?id=1204 for an overview of the RSA BSAFE product range.

1.2 Document Organization

This non-proprietary Security Policy document is one document in the FIPS 140-2 validation submission package. With the exception of this document, the FIPS 140-2 validation submission documentation is EMC-proprietary and is releasable only under appropriate non-disclosure agreements. For access to the documentation, contact RSA.

This document explains the Module's features and functionality relevant to FIPS 140-2, and contains the following sections:

- This section, "Preface" on page 2 provides an overview and introduction to the Security Policy.
- "The Cryptographic Module" on page 3, describes the Module and how it meets the FIPS 140-2 requirements.
- "Secure Operation of the Module" on page 12, provides information on implementing the FIPS140 mode of operation.
- "Acronyms" on page 15, lists the definitions for the acronyms used in this document.

2 The Cryptographic Module

This section provides an overview of the Module, and contains the following topics:

2.1 Introduction

BSAFE CNG consists of two user-mode CNG Providers which are drop-in replacements for two Microsoft user-mode CNG providers: the "Microsoft Primitive Provider" and the "Microsoft SSL Protocol Provider".

The BSAFE CNG Primitive Provider (the BSAFE CNG Cryptographic Primitives Library, or Module), is a drop-in replacement for the Microsoft user-mode CNG provider. Applications written against the Microsoft CNG framework, that do not explicitly request a specific provider, will automatically use the BSAFE CNG cryptographic implementations without modification once the BSAFE CNG Primitive Provider is installed. It can be dynamically linked into applications by software developers to permit the use of general purpose FIPS 140-2 Level 1 compliant cryptography.

The BSAFE CNG SSL Protocol Provider uses the Module to provide Transport Layer Security cipher suites for the Microsoft Schannel provider. This CNG provider is described in detail in the *RSA BSAFE CNG Developers Guide* and is not discussed further in this document.

2.2 Module Characteristics

The Module is classified as a FIPS 140-2 multi-chip standalone module. As such, it is tested on particular operating systems and computer platforms. The cryptographic boundary includes the Module running on selected platforms that are running selected operating systems.

The Module is validated for all FIPS 140-2 Level 1 security requirements. It is packaged as a dynamic link library (`BSAFEPrimitives.dll`). In addition, the Module relies on the physical security provided by the host on which it runs.

The Module is tested on the following platforms:

- Microsoft® Windows® 7 x86
- Microsoft Windows 7 x86_64.

Compliance is maintained on platforms for which the binary executable remains unchanged. This includes (but is not limited to) Microsoft Windows 2008 R2 x86_64.

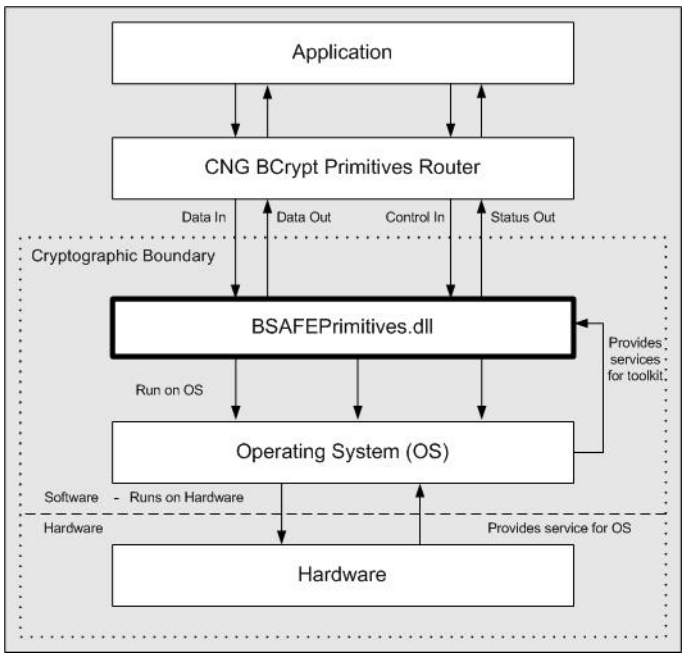
2.3 Module Interfaces

The Module is evaluated as a multi-chip, standalone module. The physical cryptographic boundary of the Module is the general-purpose computer system, which encloses the hardware running the Module. The physical interfaces for the Module consist of the keyboard, mouse, monitor, CD-ROM drive, floppy drive, serial ports, USB ports, COM ports, and network adapter(s).

The logical boundary of the cryptographic Module is the library file `BSAFEPrimitives.dll`. The underlying logical interface to the Module is the API, documented in the *RSA BSAFE CNG Developers Guide*. The Module provides for Control Input through the API calls. Data Input and Output are provided in the variables passed with the API calls, and Status Output is provided through the returns and error codes that are documented for each call.

This is illustrated in the following diagram.

Figure 1 BSAFE CNG Logical Interfaces



2.4 Roles, Services and Authentication

The Module meets all FIPS 140-2 Level 1 requirements for Roles and Services, supporting both a Crypto Officer role and a Crypto User role. As allowed by FIPS 140-2, BSAFE CNG does not require user identification or authentication for these roles.

The Crypto Officer role is assumed when the Module is being installed. Installation of the Module is the only action the Crypto Officer role can perform. No services are available to the Crypto Officer role after installation of the Module.

After installation of the Module, the Crypto User role is assumed. All services implemented by the Module are available to the Crypto User role.

Table 1 Services

Services	
BCryptCloseAlgorithmProvider	BCryptCreateHash
BCryptDecrypt	BCryptDeriveKey
BCryptDestroyHash	BCryptDestroyKey
BCryptDestroySecret	BCryptDuplicateHash
BCryptDuplicateKey	BCryptEncrypt
BCryptExportKey	BCryptFinalizeKeyPair
BCryptFinishHash	BCryptFreeBuffer
BCryptGenerateKeyPair	BCryptGenerateSymmetricKey
BCryptGenRandom	BCryptGetProperty
BCryptHashData	BCryptImportKey
BCryptImportKeyPair	BCryptOpenAlgorithmProvider
BCryptSecretAgreement	BCryptSetProperty
BCryptSignHash	BCryptVerifySignature

2.5 Cryptographic Key Management

2.5.1 Key Generation

The Module supports the generation of the DSA, RSA, Diffie-Hellman (DH) and Elliptic Curve Cryptography (ECC) public and private keys. The Module employs the HMAC Deterministic Random Bit Generator (HMAC DRBG SP 800-90), as well as the Dual Elliptic Curve Deterministic Random Bit Generator (EC DRBG SP 800-90) for generating asymmetric keys used in algorithms such as RSA, DSA, DH, and ECC.

2.5.2 Key Protection/Zeroization

All key data resides in internally allocated data structures and can be output only using the `BCryptExportKey()` API. The operating system protects memory and process space from unauthorized access.

A key is destroyed and its memory location zeroized when the application calls `BCryptDestroyKey()` or `BCryptDestroySecret()` on the key's key handle.

2.5.3 Key Access

An authorized operator of the Module has access to all key data created during a BSAFE CNG operation.

The following table lists the services provided by the Module with the type of access to keys or Critical Security Parameters (CSPs).

Table 2 Key and CSP Access

Service	Key or CSP	Type of Access
Encryption and decryption	AES key Triple DES key RSA public key RSA private key	Read/Execute
Digital signature and verification	DSA public key DSA private key RSA public key RSA private key ECDSA public key ECDSA private key	Read/Execute
Hashing	None	N/A
MAC	HMAC with SHA-1 and SHA-2 keys	Read/Execute

Table 2 Key and CSP Access (continued)

Service	Key or CSP	Type of Access
Random number generation	RNG seeds (FIPS 186-2), RNG keys (FIPS 186-2) HMAC DRBG entropy, HMAC DRBG V value, HMAC DRBG key and HMAC DRBG init_seed EC DRBG entropy, EC DRBG S value, and EC DRBG init_seed	Read/Write/Execute
Key establishment primitives	RSA public key RSA private key DH public key DH private key EC public key EC private key	Read/Execute
Key generation	AES key Triple DES key DSA public key DSA private key EC public key EC private key DH public key DH private key RSA public key RSA private key HMAC with SHA-1 and SHA-2 keys	Write
Self-test	Hard-coded (AES key, Triple DES key, RSA public key, RSA private key, DSA public key, DSA private key, EC public key, EC private key, HMAC with SHA-1 and SHA-2 keys)	Read/Execute

2.5.4 Key Storage

The Module does not provide long-term cryptographic key storage. Storage of keys is the responsibility of the user of the Module. The following table shows how the storage of keys and CSPs are handled. The Crypto User and Crypto Officer roles have equal and complete access to all keys and CSPs.

Table 3 Key and CSP Storage

Key Type	Storage
AES key	In volatile memory only (plaintext)
Triple DES key	In volatile memory only (plaintext)
HMAC with SHA-1 and SHA-2 keys	In volatile memory only (plaintext)
EC public key	In volatile memory only (plaintext)
EC private key	In volatile memory only (plaintext)
DH public key	In volatile memory only (plaintext)
DH private key	In volatile memory only (plaintext)
RSA public key	In volatile memory only (plaintext)
RSA private key	In volatile memory only (plaintext)
DSA public key	In volatile memory only (plaintext)
DSA private key	In volatile memory only (plaintext)
RNG seeds (FIPS 186-2)	In volatile memory only (plaintext)
RNG keys (FIPS 186-2)	In volatile memory only (plaintext)
EC DRBG entropy	In volatile memory only (plaintext)
EC DRBG S value	In volatile memory only (plaintext)
EC DRBG init_seed	In volatile memory only (plaintext)
HMAC DRBG entropy	In volatile memory only (plaintext)
HMAC DRBG V value	In volatile memory only (plaintext)
HMAC DRBG key	In volatile memory only (plaintext)
HMAC DRBG init_seed	In volatile memory only (plaintext)

2.6 Cryptographic Algorithms

The Module meets FIPS 140-2 requirements by requiring applications to only use algorithms listed in the table below.

Table 4 BSAFE CNG FIPS-approved Algorithms

Algorithm	Validation Certificate
AES (CBC, CFB, ECB, and GCM modes)	1598
Triple DES (CBC, CFB, and ECB modes)	1044
Triple DES (two key) (CBC, CFB, and ECB modes) 112 bits	1044
RSA signing	780
DSA	493
ECDSA	196
SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512	1410
HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512	935
FIPS 186-2 RNG	855
HMAC SHA-1 RNG, HMAC SHA-256 RNG, HMAC SHA-512 RNG, and Dual EC DRBG	77

The following are the non-FIPS 140-approved algorithms allowed in FIPS mode provided by the BSAFE CNG Cryptographic Primitives Library:

- RSA encryption and decryption used for key transport (key wrapping; key establishment methodology provides 80, 112, or 128 bits of encryption strength).
- Diffie-Hellman
- Elliptic Curve Diffie-Hellman.

The following are the non-FIPS 140-approved algorithms provided by the BSAFE CNG Cryptographic Primitives Library:

- DES (CBC and ECB modes) – 56 bits
- DESX (CBC, CFB, and ECB modes) – 184 bits
- RC2 (CBC, CFB, and ECB modes) – 1 to 1024 bits
- RC4 – 8 to 2048 bits
- MD2
- MD4
- MD5

- HMAC-MD2
- HMAC-MD4
- HMAC-MD5.

The following is the vendor-affirmed security method provided by the BSAFE CNG Cryptographic Primitives Library:

- FIPS 186-3 RSA.

2.7 Self-tests

The Module performs power-up and conditional self-tests to ensure proper operation.

A conditional self-test failure does NOT disable the Module.

2.7.1 Power-up Self-tests

The following Known Answer Tests (KAT) FIPS 140-2 required power-up self-tests are implemented in `BSAFEPrimitives.dll` when `DllMain` is called by the operating system.

- HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512
- SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512
- Triple DES encryption/decryption (ECB)
- Triple DES encryption/decryption (CBC)
- Triple DES encryption/decryption (CFB 8-bit)
- Triple DES two key encryption/decryption (ECB)
- Triple DES two key encryption/decryption (CBC)
- Triple DES two key encryption/decryption (CFB 8-bit)
- AES-128, AES-192, and AES-256 encryption/decryption (ECB)
- AES-128, AES-192, and AES-256 encryption/decryption (CBC)
- AES-128, AES-192, and AES-256 encryption/decryption (CFB 8-bit)
- AES-128, AES-192, AES-256 encryption/decryption (CCM)
- AES-128, AES-192, AES-256 encryption/decryption (GCM)
- DSA signing/verification (1024-bit key)
- RSA signing/verification (2048-bit key)
- ECDSA signing/verification (P256, P384, and P521 curves)
- HMAC-SHA1-DRBG, HMAC-SHA256-DRBG, and HMAC-SHA512-DRBG
- FIPS 186-2 DSA RNG
- SP 800-90 Dual ECDRBG (P256, P384, and P521 curves)
- Software/firmware integrity check.

In all cases for any failure of a power-up self-test, `BSAFEPrimitives.dll` `DllMain` fails to return the value `TRUE` to the operating system. The only way to recover from the failure of a power-up self-test is to attempt to reload the `BSAFEPrimitives.dll`, which will rerun the self-tests, and will only succeed if the self-tests pass.

2.7.2 Conditional Self-tests

`BSAFEPrimitives.dll` performs pair-wise consistency checks upon each invocation of RSA, DSA, ECDSA, DH, and ECDH key-pair generation and import as defined in FIPS 140-2.

`BSAFEPrimitives.dll` also performs a continuous RNG test on each of the implemented RNGs as defined in FIPS 140-2. Additionally, health testing is performed on each of the implemented DRBGs as defined in SP 800-90.

2.7.3 Mitigation of Other Attacks

RSA key operations implement blinding by default. Blinding is a reversible way of modifying the input data, so as to make the RSA operation immune to timing attacks.

Blinding has no effect on the algorithm other than to mitigate attacks on the algorithm.

3 Secure Operation of the Module

The Crypto Officer role must configure the Module using one of the two procedures described below to place the Module in a FIPS 140 approved mode. This allows the Crypto User role to operate the Module in a FIPS 140 approved mode of operation.

1. One of the following `DWORD` registry values is set to 1:
   - `HKLM\SYSTEM\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy\Enabled`
   - `HKLM\SYSTEM\CurrentControlSet\Policies\Microsoft\Cryptography\Configuration\SelfTestAlgorithms`

   and

   The following `DWORD` registry value is set to 3 `[CNG_DELEGATE_TO_MS]`:

   `HKLM\Software\RSA Security Inc.\RSA BSAFE CNG\SelfTestMode`
2. The following `DWORD` registry value is set to 2 `[CNG_TEST_MODE_FIPS]`:

   `HKLM\Software\RSA Security Inc.\RSA BSAFE CNG\SelfTestMode`

The following guidance must be followed to achieve a FIPS140 mode of operation.

3.1 Crypto User Guidance

This section provides guidance to the Module user to ensure that the Module is used in a FIPS 140-2 compliant way.

3.1.1 Crypto User Guidance on Algorithms

To operate the Module in a FIPS 140 approved mode of operation, the Crypto User shall only use the algorithms approved for use in a FIPS 140 mode of operation, as listed in Table 4, “BSAFE CNG FIPS-approved Algorithms” on page 9.

The requirements for using the approved algorithms in a FIPS 140 mode of operation are as follows:

- The bit length for a DSA key pair must be 1024 bits.
- RNGs must be seeded with values of at least 160 bits in length.
- Bit lengths for an RSA key pair must be 1024, 2048 or 3072.
- Bit lengths for Diffie-Hellman key agreement must be 1024 or 2048 bits.
- Diffie-Hellman shared secret provides between 80 bits and 112 bits of encryption strength.
- The bit lengths for an HMAC key must be one half of the block size.
- For RSASSA-PSS, the length of the salt (`sLen`) shall be `0<=sLen<=hlen` where `hlen` is the length of the hash function output block.

For more information on the algorithm strength and keysize, see the *RSA BSAFE CNG Release Notes*. Users should take care to zeroize CSPs when they are no longer needed.

3.1.2 Crypto User Guidance on Obtaining Assurances for Digital Signature Applications

The Module supports FIPS 186-3 Digital Signature Standard for ECDSA and RSA. The following gives an overview of the assurances required by FIPS 186-3.

NIST Special Publication 800-89: "Recommendation for Obtaining Assurances for Digital Signature Applications" provides the methods to obtain these assurances.

The tables below describe the FIPS 186-3 requirements for signatories and verifiers and the corresponding Module capabilities and recommendations.

Table 5 Signatory Requirements

FIPS 186-3 Requirement	Module Capabilities and Recommendations
Obtain appropriate ECSDSA parameters when using ECDSA.	ECDSA uses the NIST recommended curves P256, P384, and P521.
Obtain assurance of the validity of those parameters.	The Module provides APIs to validate DSA parameters for probable primes as described in FIPS 186-3.
Obtain a digital signature key pair that is generated as specified for the appropriate digital signature algorithm.	The Module generates the digital signature key pair according to the required standards. A FIPS-approved RNG such as HMAC DRBG or Dual ECDRBG is used to generate the key pair.
Obtain assurance of the validity of the public key.	Public key is validated upon import (`BCryptImportKeyPair()`) according to NIST Special Publication 800-89.
Obtain assurance that the signatory actually possesses the associated private key.	The Module verifies the signature created using the private key, but all other assurances are outside the scope of the Module.

Table 6 Verifier Requirements

FIPS 186-3 Requirement	Module Capabilities and Recommendations
Obtain assurance of the signatory's claimed identity.	The Module verifies the signature created using the private key, but all other assurances are outside the scope of the Module.
Obtain assurance of the validity of the domain parameters for ECDSA.	ECDSA uses the NIST recommended curves P256, P384, and P521.
Obtain assurance of the validity of the public key.	Public key is validated upon import (`BCryptImportKeyPair()`) according to NIST Special Publication 800-89.
Obtain assurance that the claimed signatory actually possessed the private key that was used to generate the digital signature at the time that the signature was generated.	Outside the scope of the Module.

Note: For more details on the requirements, see the FIPS 186-3 and NIST Special Publication 800-89.

3.2 Crypto Officer Guidance

The Crypto Officer is responsible for installing the Module. For instructions on how to install BSAFE CNG, see the *RSA BSAFE CNG Installation Guide*.

When operating the Module after installation, the Crypto Officer must follow the Crypto User guidance requirements detailed in "Crypto User Guidance" on page 12.

3.3 Operating the Cryptographic Module

The Module does not enforce the FIPS140 mode of operation. Both FIPS and non-FIPS algorithms are available to the operators. The operators must ensure that they follow the Security Policy guidelines.

3.4 Startup Self-tests

All KATs are executed on Module start-up, which occurs on first use. If any KAT fails, the library fails to load.

3.5 Default Random Number Generator

The Module provides a default RNG, which is the Dual EC DRBG, using a P256 curve and SHA-256.

4 Acronyms

This table lists the definitions for the acronyms used in the BSAFE CNG Cryptographic Primitives Library.

Term	Description
AES	Advanced Encryption Standard. A fast block cipher with a 128-bit block, and keys of lengths 128, 192, and 256 bits. Replaces DES as the US symmetric encryption standard.
API	Application Programming Interface.
Attack	Either a successful or unsuccessful attempt at breaking part or all of a cryptosystem. Various attack types include an algebraic attack, birthday attack, brute force attack, chosen ciphertext attack, chosen plaintext attack, differential cryptanalysis, known plaintext attack, linear cryptanalysis, and middle person attack.
CBC	Cipher Block Chaining. A mode of encryption in which each ciphertext depends upon all previous ciphertexts. Changing the Initialization Vector (IV) alters the ciphertext produced by successive encryptions of an identical plaintext.
CFB	Cipher Feedback. A mode of encryption that produces a stream of ciphertext bits rather than a succession of blocks. In other respects, it has similar properties to the CBC mode of operation.
CRNG	Continuous Random Number Generation.
CSP	Critical Security Parameter.
DES	Data Encryption Standard. A symmetric encryption algorithm with a 56-bit key. See also Triple DES.
Diffie-Hellman	The Diffie-Hellman asymmetric key exchange algorithm. There are many variants, but typically two entities exchange some public information (for example, public keys or random values) and combines them with their own private keys to generate a shared session key. As private keys are not transmitted, eavesdroppers are not privy to all of the information that composes the session key.
DSA	Digital Signature Algorithm. An asymmetric algorithm for creating digital signatures.
DRBG	Deterministic Random Bit Generator.
Dual EC DRBG	Dual Elliptic Curve Deterministic Random Bit Generator.
EC	Elliptic Curve.
ECB	Electronic Codebook. A mode of encryption that divides a message into blocks and encrypts each block separately.
ECC	Elliptic Curve Cryptography.
ECDSA	Elliptic Curve Digital Signature Algorithm.

Term	Description
Encryption	The transformation of plaintext into an apparently less readable form (called ciphertext) through a mathematical process. The ciphertext can be read by anyone who has the key that decrypts (undoes the encryption) the ciphertext.
FIPS	Federal Information Processing Standards.
GCM	Galois/Counter Mode. A mode of encryption that combines the Counter mode of encryption with Galois field multiplication for authentication.
GMAC	Galois Message Authentication Code. An authentication only variant of GCM.
HMAC	Keyed-Hashing for Message Authentication Code.
HMAC DRBG	HMAC Deterministic Random Bit Generator.
IV	Initialization Vector. Used as a seed value for an encryption operation.
KAT	Known Answer Test.
Key	A string of bits used in cryptography, allowing people to encrypt and decrypt data. Can be used to perform other mathematical operations as well. Given a cipher, a key determines the mapping of the plaintext to the ciphertext. The types of keys include distributed key, private key, public key, secret key, session key, shared key, subkey, symmetric key, and weak key.
MD5	A secure hash algorithm created by Ron Rivest. MD5 hashes an arbitrary-length input into a 16-byte digest.
NIST	National Institute of Standards and Technology. A division of the US Department of Commerce (formerly known as the NBS) which produces security and cryptography-related standards.
OS	Operating System.
PC	Personal Computer.
PDA	Personal Digital Assistant.
PPC	PowerPC.
privacy	The state or quality of being secluded from the view or presence of others.
private key	The secret key in public key cryptography. Primarily used for decryption but also used for encryption with digital signatures.
PRNG	Pseudo-random Number Generator.
RC2	Block cipher developed by Ron Rivest as an alternative to the DES. It has a block size of 64 bits and a variable key size. It is a legacy cipher and RC5 should be used in preference.
RC4	Symmetric algorithm designed by Ron Rivest using variable length keys (usually 40-bit or 128-bit).

Term	Description
RC5	Block cipher designed by Ron Rivest. It is parameterizable in its word size, key length, and number of rounds. Typical use involves a block size of 64 bits, a key size of 128 bits, and either 16 or 20 iterations of its round function.
RNG	Random Number Generator.
RSA	Public key (asymmetric) algorithm providing the ability to encrypt data and create and verify digital signatures. RSA stands for Rivest, Shamir, and Adleman, the developers of the RSA public key cryptosystem.
SHA	Secure Hash Algorithm. An algorithm that creates a unique hash value for each possible input. SHA takes an arbitrary input that is hashed into a 160-bit digest.
SHA-1	A revision to SHA to correct a weakness. It produces 160-bit digests. SHA-1 takes an arbitrary input that is hashed into a 20-byte digest.
SHA-2	The NIST-mandated successor to SHA-1, to complement the Advanced Encryption Standard. It is a family of hash algorithms (SHA-224, SHA-256, SHA-384 and SHA-512) that produce digests of 224, 256, 384 and 512 bits respectively.
Triple DES	A variant of DES that uses three 56-bit keys.