



Cryptographic Algorithm Validation Testing.

Table of Contents

FIPS APPROVED ALGORITHMS.....	3
Symmetric Key.....	4
AES	5
TDES:.....	6
SKIPJACK / EES	7
Asymmetric Key	8
DSA.....	11
DSA-2.....	12
RSA:.....	14
ECDSA	15
Secure Hash Standard (SHS)	17
SHA	17
Random Number Generators (RNG).....	18
RNG	18
Deterministic Random Bit Generators (DRBG)	20
DRBG:.....	20
Key Management –Key Agreement Schemes and Key Confirmation (KAS).....	22
KAS FFC	22
KAS ECC	23
Message Authentication (MAC)	25
CMAC	25
CCM:.....	26
GCM:	27
HMAC:	28
NON FIPS APPROVED ALGORITHMS.....	29
Data Encryption Standard (DES)	29
Message Authentication Code (MAC)	29
Data (Message) Authentication Code (MAC) and Key Management Using ANSI X9.17.....	29
RC4.....	29
Blowfish	30
HOTP	30
CRC.....	30



FIPS Approved algorithms

Much of the information given here is derived from the Cryptographic Algorithm Validation Program pages at <http://csrc.nist.gov/groups/STM/cavp/index.html>. In the case of the following FIPS Approved and NIST recommended algorithm and security function testing the CAVP is the authoritative source. Any differences between this document and the CAVP web site are unintentional and the CAVP pages take precedence.

This document includes a checklist, facilitating requests conformance testing with atsec. Please mark those algorithms/functions and the modes that require testing.

Symmetric Key

Advanced Encryption Standard (AES), Triple-DES, and Skipjack Algorithms

Currently, there exist three FIPS-approved symmetric key algorithms for encryption: Advanced Encryption Standard (AES), Triple-DES, and Skipjack. AES is the FIPS-Approved symmetric encryption algorithm of choice.

- [FIPS 197](#), *Advanced Encryption Standard (AES)*, specifies the AES algorithm.
- *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, NIST [Special Publication 800-67](#), May 2004.
- *Recommendation for Block Cipher Modes of Operation, Methods and Techniques*, [Special Publication 800-38A](#), December 2001. Appendix E references Modes of Triple-DES.
- *Triple Data Encryption Algorithm Modes of Operation*, ANSI X9.52-1998. **Copies of X9.52-1998 may be obtained from X9**, a standards committee for the financial services industry. NIST does NOT have copies of the standard available for distribution.
- The [Skipjack algorithm](#) is referenced in [FIPS 185](#), *Escrowed Encryption Standard (EES)*, and a complete specification is available in [SKIPJACK and KEA Algorithm Specifications \(Version 2.0, 29 May 1998\)](#).

Testing Requirements:

Validation testing for AES, Triple-DES, and Skipjack algorithms are handled by the Cryptographic Algorithm Validation Program's ([CAVP](#)) [Atsec](#).

- AES tests are described in [The Advanced Encryption Standard Algorithm Validation Suite \(AESAVS\)](#).
- Triple-DES tests are described in NIST Special Publication [800-20](#), *Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures*. An additional test, the [Multi-block Message Text \(MMT\)](#), is also required.
- Skipjack tests are described in NIST Special Publication [800-17](#), *Modes of Operation Validation System (MOVS): Requirements and Procedures*. The validation test suite for Skipjack implementations using the encrypt state consists of the Modes test for the Encryption Process, the Variable Plaintext Known Answer Test and the Variable Key Known Answer Test for the Encryption Process. The validation test suite for Skipjack implementations using the decrypt state consists of the Modes test for the Decryption Process, the Variable Ciphertext Known Answer Test and the Variable Key Known Answer Test for the Decryption Process. NIST Special Publication 800-17 erroneously states the Inverse Permutation KAT for the Encryption Process and the Initial Permutation KAT for the Decryption Process are also required for Skipjack validation. This is not true.

Validation List:

NIST maintains validation lists for [AES](#), [Triple-DES](#), and [Skipjack](#). These lists identify the algorithm implementations which have been tested as correctly implementing the AES, Triple-DES, and Skipjack algorithms. Points of contact and implementation descriptions are also included.

Other Information:

- [AES Known Answer Test \(KAT\) Vectors](#) - This file provides an electronic version of the KAT vectors that can be used to informally verify the correctness of an AES algorithm implementation, using the Known Answer Test (KAT) described in [The Advanced Encryption Standard Algorithm Validation Suite \(AESAVS\)](#). **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).** >

- [Triple-DES Sample Vectors](#) - This file provides sample vectors that can be used to informally verify the correctness of a Triple-DES implementation, using the Monte Carlo Tests described in NIST Special Publication [800-20](#). **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

AES:					
	ECB	128 Bits	Encrypt,	Decrypt	
		192 Bits	Encrypt,	Decrypt	
		256 Bits	Encrypt,	Decrypt	
	CBC	128 Bits	Encrypt,	Decrypt	
		192 Bits	Encrypt,	Decrypt	
		256 Bits	Encrypt,	Decrypt	
	OFB	128 Bits	Encrypt,	Decrypt	
		192 Bits	Encrypt,	Decrypt	
		256 Bits	Encrypt,	Decrypt	
	CFB 1	128 Bits	Encrypt,	Decrypt	
		192 Bits	Encrypt,	Decrypt	
		256 Bits	Encrypt,	Decrypt	
	CFB 8	128 Bits	Encrypt,	Decrypt	
		192 Bits	Encrypt,	Decrypt	
		256 Bits	Encrypt,	Decrypt	
	CFB 128	128 Bits	Encrypt,	Decrypt	
		192 Bits	Encrypt,	Decrypt	
		256 Bits	Encrypt,	Decrypt	
	CTR ¹	with state options	128 bits	Encrypt	
		with state options	192 bits	Encrypt	
with state options		256 bits	Encrypt		
with counter source options		Internal	External		

¹ CTR mode for AES requires a source code / design review by atsec

TDES:			
	ECB	with state options	Encrypt, Decrypt
		with keying options	K1, K2, K3 independent
			K1 = K3, K2 independent
	CBC	with state options	Encrypt, Decrypt
		with keying options	K1, K2, K3 independent
			K1 = K3, K2 independent
	CBC-1	with state options	Encrypt, Decrypt
		with keying options	K1, K2, K3 independent
			K1 = K3, K2 independent
	CFB	with state options	1 Bit Encrypt, Decrypt
			8 Bits Encrypt, Decrypt
			64 Bits Encrypt, Decrypt
		with keying options	K1, K2, K3 independent
			K1 = K3, K2 independent
			K1 = K2 = K3
	CFB-P	with state options	1 Bit Encrypt, Decrypt
			8 Bits Encrypt, Decrypt
			64 Bits Encrypt, Decrypt
		with keying options	K1, K2, K3 independent
			K1 = K3, K2 independent
			K1 = K2 = K3
OFB	with state options	Encrypt, Decrypt	
	with keying options	K1, K2, K3 independent	
		K1 = K3, K2 independent	
OFB-1	with state options	Encrypt, Decrypt	
	with keying options	K1, K2, K3 independent	
	with keying options	K1 = K3, K2 independent	
	with keying options	K1 = K2 = K3	
CTR	with state options	Encrypt	
	with counter source options	Internal	
		External	

SKIPJACK / EES:				
	ECB	128 Bits	Encrypt	
			Decrypt	
		192 Bits	Encrypt	
			Decrypt	
		256 Bits	Encrypt	
			Decrypt	
	CBC	128 Bits	Encrypt	
			Decrypt	
		192 Bits	Encrypt	
			Decrypt	
		256 Bits	Encrypt	
			Decrypt	
	OFB	128 Bits	Encrypt	
			Decrypt	
		192 Bits	Encrypt	
			Decrypt	
		256 Bits	Encrypt	
			Decrypt	
	CFB 1	128 Bits	Encrypt	
			Decrypt	
		192 Bits	Encrypt	
			Decrypt	
		256 Bits	Encrypt	
			Decrypt	
CFB 8	128 Bits	Encrypt		
		Decrypt		
	192 Bits	Encrypt		
		Decrypt		
	256 Bits	Encrypt		
		Decrypt		
CFB 128	128 Bits	Encrypt		
		Decrypt		
	192 Bits	Encrypt		
		Decrypt		
	256 Bits	Encrypt		
		Decrypt		
CTR	with state options 128 bits	Encrypt		
	with state options 192 bits	Encrypt		
	with state options 256 bits	Encrypt		
	with counter source options	Internal	External	

Asymmetric Key

FIPS 186-3 Digital Signature Standard (DSS) (DSA2, RSA2, and ECDSA2 algorithms)

On June 10, 2009, NIST [announced](#) the adoption of [FIPS 186-3](#), *Digital Signature Standard (DSS)*, which is a revision of FIPS 186-2. The FIPS specifies three techniques for the generation and verification of digital signatures:

- **Digital Signature Algorithm (Referred to as DSA2),**
- **RSA (as specified in ANSI X9.31 (Referred to as RSA2)), and**
- **Elliptic Curve DSA (Referred to as ECDSA2; as specified in ANSI X9.62).**

FIPS 186-3 incorporates the following changes:

General:

- Specifies the use of all hash functions specified provided in FIPS 180-3, rather than just SHA-1,
- Provides requirements for obtaining assurances of domain parameter validity (DSA2 and ECDSA2 only), public key validity, and private key possession,
- References SP 800-57 for guidance on key management, including the key sizes and security strengths to be used,
- Provides guidance on domain parameter and key pair management,
- References SP 800-90 for random number generation, rather than including RNGs in the Standard, either explicitly or by reference to ANSI Standards,
- Provides more guidance on the use of RNGs to generate key pairs,
- Provides revised primality test guidance.

DSA2:

- Specifies larger key sizes,
- Replaces the domain parameter generation routine with new methods,
- Includes explicit methods for the validation of domain parameters,

RSA2:

- Approves the use of both ANSI X9.31 and PKCS #1, and provides guidance for their use,
- Provides multiple explicit methods for the generation of key pairs,
- Limits the key sizes and provides criteria for the generation of key pairs to be used for Federal government use.

ECDSA2:

- Although the Recommended Elliptic Curves continue to be included in FIPS 186-3 (as they were in FIPS 186-2), FIPS 186-3 allows the generation of alternative curves, using methods specified in ANS X9.62.

Copies of the **ANSI X9.31** and **ANSI X9.62** standards are available from [X9](#), a standards committee accredited by the American National Standards Institute (ANSI). NIST does NOT have copies of these standards available for distribution.

All three digital signature techniques in FIPS 186-3 make use of the Secure Hash Algorithms specified in FIPS 180-3 dated October 2008, *Secure Hash Standard (SHS)* accessible via the hashing section of this webpage.

FIPS 186-2 and FIPS 186-3 are currently the *only* FIPS standards that contain Approved methods for digital signatures.

Testing Requirements:

atsec can test for conformance to the algorithm specifications in FIPS 186-3 for the DSA2 algorithm, with the exception of the generation and validation of provably prime domain parameters p and q and canonical generation and validation of domain parameter g . These methods, along with FIPS 186-3 ECDSA2 and RSA2, will require vendor affirmation until validation testing is available in the CAVS tool. Refer to I.G.1.15 **CAVP Requirements for Vendor Affirmation of FIPS 186-3 Digital Signature Standard** for details.



The algorithm validation testing requirements for FIPS 186-3 DSA2 are specified in: [Digital Signature Algorithm Validation System \(DSA2VS\)](#)

Additional testing note: For the Domain Parameter Generation and Verification, and the Signature Generation and Verification functions, the underlying SHA algorithm must be validated as part of the DSA2 validation. In addition, Signature Generation and Key Pair Generation require the RNG/DRBG algorithm to be validated as well.

FIPS 186-2 Digital Signature Standard (DSS) (DSA, RSA, and ECDSA algorithms)

On February 15, 2000, NIST [announced](#) the approval of [FIPS 186-2 with Change Notice 1 dated October 5, 2001](#), *Digital Signature Standard (DSS)*, which supersedes FIPS 186-1. This standard specifies three FIPS-approved algorithms for generating and verifying digital signatures:

- **Digital Signature Algorithm (DSA),**
- **RSA (as specified in ANSI X9.31), and**
- **Elliptic Curve DSA (ECDSA; as specified in ANSI X9.62).**

New items in the DSS include:

- the approval of Elliptic Curve DSA (ECDSA) as specified in ANSI X9.62,
- a list of recommended elliptic curves for Federal Government use (see Appendix 6 of [FIPS 186-2 with Change Notice 1 dated October 5, 2001](#)), and
- an [allowance for the continued acquisition of implementations of PKCS#1 for a transition period of eighteen \(18\) months](#).

Copies of the **ANSI X9.31** and **ANSI X9.62** standards are available from [X9](#), a standards committee accredited by the American National Standards Institute (ANSI). NIST does NOT have copies of these standards available for distribution.

All three digital signature techniques in FIPS 186-2 (with Change Notice 1 dated October 5, 2001) make use of the Secure Hash Algorithms specified in FIPS 180-3 dated October 2008, *Secure Hash Standard (SHS)* accessible via the hashing section of this webpage.

DSA, RSA, and ECDSA are currently the *only* FIPS-approved methods for digital signatures.

Testing Requirements:

atsec can test for conformance to the algorithm specifications in FIPS 186-2 (with Change Notice 1 dated October 5, 2001). Algorithm specifications included in this standard are the DSA, the RSA and the ECDSA algorithms. In addition, NIST can test for conformance to two other versions of the RSA algorithm specified in *PKCS#1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 2002*.

The testing requirements are specified in:

[Digital Signature Algorithm Validation System \(DSAVS\)](#)

Additional testing note: For the Domain Parameter Generation and Verification, and the Signature Generation and Verification functions, the underlying SHA-1 algorithm must be validated as part of the DSA validation. In a future release, the other SHA algorithms will be supported.

[RSA Validation System \(RSAVS\)](#)

Beginning September 28, 2006: Validation testing for RSA algorithm implementations of the RSASSA-PKCS1-v1_5, as specified in *Public Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard-2002*, and the RSA X9.31 algorithms include additional testing to assure the encoded message EM and the intermediate integer IR are in the correct formats. This testing verifies that an implementation under

test (IUT) does not contain a potential implementation design that could introduce a vulnerability in these algorithms. This testing has been added to the Signature Verification validation test described in the RSAVS document. No modification to this document was necessary to add this feature. Below in the Test Vectors section, there are test vectors available to informally test if this vulnerability exists in an implementation.

For all validated cryptographic modules that incorporate RSA, the CMVP and CAVP strongly suggest re-testing of the RSA algorithmic implementations to determine if the vulnerability is present.

If new CAVP testing is performed and the vulnerability is determined not to be present, the CMTL can submit the new test results to the CAVP along with a letter indicating that the implementation passed the RSA testing in CAVS5.2 and the vulnerability is not present. The letter should request that a new algorithm certificate be printed to replace the already issued certificate referencing the new version of CAVS. Please indicate the already issued certificate number. This letter should be included in the zip file along with the other files. Note that the certificate number will not change. Only the reference to the version of the CAVS tool and the signatory date will be changed. (Note the validation request will be submitted using already established procedures.)

If CAVP testing is performed and the vulnerability is discovered, the following revalidation process shall be followed:

- The algorithm implementation is changed to remove the vulnerability resulting in a different version number,
- Submit the new test results to the CAVP for the new version of the implementation. A new algorithm certificate will be issued for the new version of the implementation. The certificate will reference CAVS5.2.

Additional testing note: For the RSA functions, all underlying SHA algorithm(s) supported by the RSA implementation must be validated as part of the RSA validation.

[Elliptic Curve Digital Signature Algorithm \(ECDSA\) Validation System \(ECDSAVS\)](#)

Additional testing note: For the Signature Generation and Verification functions, the underlying SHA-1 algorithm must be validated as part of the ECDSA validation. In a future release, the other SHA algorithms will be supported.

Validation Listings:

NIST maintains the current DSA, ECDSA, and RSA Validation Lists.

- [DSA Validation List](#)
- [RSA Validation List](#)
- [ECDSA Validation List](#)

Test Vectors:

These files provide an electronic version of the test vectors that can be used to informally verify the correctness of the algorithm implementation using the associated validation system document (DSAVS, ECDSAVS, or RSAVS). **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

- [DSA Test Vectors](#)
- [RSA Sample Vectors](#)
 - [RSA SigVer PKCS1.5 Vulnerability Test Vectors](#)
 - [RSA SigVer X9.31 Vulnerability Test Vectors](#)



- [ECDSA Test Vectors](#)

Other Information:

Elliptic curves recommended for Federal Government use are specified in Appendix 6 of [FIPS 186-2 with Change Notice 1 dated October 5, 2001](#). They are also listed separately: [PDF](#) and [Word](#).

DSA:		
	PQG Gen	1024
	PQG Ver	1024
	Key Pair Gen	1024
	Sig Gen	1024
	Sig Ver	1024

DSA-2:			
	PQG Gen L=1024, N=160	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	PQG Gen L=2048, N=224	SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	PQG Gen L=2048, N=256	SHA-256	
		SHA-384	
		SHA-512	
PQG Gen L=3072, N=256	SHA-256		
	SHA-384		
	SHA-512		
	PQG Ver L=1024, N=160	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	PQG Ver L=2048, N=224	SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	PQG Ver L=2048, N=256	SHA-256	
		SHA-384	
		SHA-512	
	PQG Ver L=3072, N=256	SHA-256	
		SHA-384	
		SHA-512	
		Key Pair L=1024, N=160	SHA-1
SHA-224			
SHA-256			
SHA-384			
SHA-512			
Key Pair L=2048, N=224		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
Key Pair L=2048, N=256		SHA-256	
		SHA-384	

		SHA-512	
	Key Pair L=3072, N=256	SHA-256	
		SHA-384	
		SHA-512	
	Sig Gen L=1024, N=160	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Gen L=2048, N=224	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Gen L=2048, N=256	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Gen L=3072, N=256	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Ver L=1024, N=160	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Ver L=2048, N=224	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Ver L=2048, N=256	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	Sig Ver L=3072, N=256	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	

RSA:				
	GenKey9.31	1024		
		1536		
		2048		
		3072		
		4096		
	Public Key Value:	3		
		17		
		65537		
	SigGen9.31 Modulus size	1024		
		1536		
		2048		
		3072		
		4096		
	SigGen9.31 Algorithm	SHA-1		
		SHA 256		
		SHA 384		
		SHA 512		
	SigGenPKCS1.5 Modulus Sizes	1024		
		1536		
		2048		
		3072		
		4096		
	SigGenPKCS1.5 Algorithms	SHA-1		
		SHA-224		
		SHA-256		
		SHA-384		
		SHA-512		
	SigGenPSS Modulus Sizes	1024		
		1536		
		2048		
		3072		
		4096		
	SigGenPSS Algorithms	SHA-1		
SHA-224				
SHA-256				
SHA-384				
SHA-512				
SigVer9.31 Modulus Sizes	1024			
	1536			
	2048			

		3072	
		4096	
	SigVer9.31 Supported Algorithms	SHA-1	
		SHA-256	
		SHA-384	
		SHA-512	
	SigVerPKCS1.5 Modulus Sizes	1024	
		1536	
		2048	
		3072	
		4096	
	SigVerPKCS1.5 Supported Algorithms	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
	SigVerPSS Modulus Sizes	1024	
		1536	
		2048	
		3072	
		4096	
	SigVerPSS Supported Algorithms	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	

ECDSA		
	KeyPair	
	PKV	
	SigGen	
	SigVer	



Secure Hash Standard (SHS)

(SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 algorithms)

The [Secure Hash Algorithms](#) (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512) are specified in [FIPS 180-3 dated October 2008](#), *Secure Hash Standard (SHS)*.

Testing Requirements:

atsec can test for conformance to the SHA algorithms in FIPS 180-3. The testing requirements for these algorithms can be found in the document titled [The Secure Hash Algorithm Validation System \(SHAVS\)](#).

Validation List:

NIST maintains the current [SHA Validation List](#).

Test Vectors:

SHA Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of a SHA algorithm implementation using the SHAVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

- [SHA Test Vectors for Hashing Bit-Oriented Messages](#)
- [SHA Test Vectors for Hashing Byte-Oriented Messages](#)

SHA		
	SHA-1 Byte Only Option	
	SHA-224 Byte Only Option	
	SHA-256 Byte Only Option	
	SHA-384 Byte Only Option	
	SHA-512 Byte Only Option	

Random Number Generators (RNG)

The algorithms for generating approved random numbers are referenced in [FIPS 140-2 Annex C](#).

Testing Requirements:

atsec can test for conformance to the following RNG algorithms that are referenced in FIPS 140-2 Annex C:

- [FIPS 186-2 with Change Notice 1 dated October 5, 2001](#) (Appendix 3.1 and 3.2)
- **ANSI X9.31** (Appendix A.2.4) - *Using 2-Key Single DES*
- [NIST-Recommended Random Number Generator based on ANSI X9.31 Appendix A.2.4 using the 3-Key Triple DES and AES algorithms](#)
- **ANSI X9.62** (Appendix A.4).
Copies of the ANSI X9.31 and ANSI X9.62 standards are available from [X9](#), a standards committee accredited by the American National Standards Institute (ANSI). NIST does NOT have copies of these standards available for distribution.

The testing requirements for these algorithms can be found in the document titled [The Random Number Generator Validation System \(RNGVS\)](#).

Validation List:

NIST maintains the current [RNG Validation List](#).

Test Vectors:

RNG Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of an RNG algorithm implementation using the RNGVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[RNG Test Vectors](#)

RNG			
	FIPS 186	RNG Test: General Purpose RNG	
	FIPS 186	RNG Test: Regular 186 RNG	
	FIPS 186	RNG Generator: X - Original	
	FIPS 186	RNG Generator: X – Change Notice	
	FIPS 186	RNG Generator: K - Original	
	FIPS 186	RNG Generator: X – Change Notice	
	FIPS 186	RNG G Function SHA-1	
	FIPS 186	RNG G Function DES	
	FIPS 186	RNG Seed-Key Byte Size min length range 20-64	
	FIPS 186	RNG Seed-Key Byte Size min length range 20-64	
	ANSI 9.62	curve P-192	
		curve P-224	
		curve P-256	
		curve P-384	
		curve P-521	

		curve K-163	
		curve P-233	
		curve P-283	
		curve P-409	
		curve P-571	
		curve B-163	
		curve B-233	
		curve B-283	
		curve B-409	
		curve B-571	
	ANSI 9.62	G function SHA-1	
		G function DES	
	ANSI 9.62	Seed-Key Byte Size Min Length in the range of [20, 64]	
		Seed-Key Byte Size Max Length in the range of [20, 64]	
	ANSI 9.31	2 Key Triple DES	
		3 Key Triple DES	
		AES with Key Size 128	
		AES with Key Size 192	
		AES with Key Size 256	

Deterministic Random Bit Generators (DRBG)

[SP 800-90 Recommendation for Random Number Generation Using Deterministic Random Bit Generators \(Revised March 2007\)](#) specifies mechanisms for the generation of random bits using deterministic methods. There are four mechanisms discussed in this Special Publication. These mechanisms are based on either hash functions (Hash_DRBG, HMAC_DRBG), block cipher algorithms using Counter mode (CTR_DRBG) or number theoretic (Dual_EC_DRBG) problems.

Testing Requirements:

atsec can test for conformance to the DRBG algorithms in Special Publication 800-90. The testing requirements for this algorithm can be found in the document titled [The DRBG Validation System \(DRBGVS\)](#).

Additional testing note: Each of the mechanisms containing underlying algorithms which must be validated as part of the DRBG validation. For HASH_DRBG, the SHA algorithm(s) must be tested. For HMAC_DRBG, the HMAC algorithm must be tested. For the block cipher algorithms using Counter mode CTR_DRBG, a NIST-Approved symmetric key algorithm using Counter mode, must be validated as part of the CMAC validation. Currently, NIST approves both the AES and TDES algorithms for use with DRBG. For Dual_EC_DRBG, the ECDSA Key Generation function and the SHA algorithm must be tested. The ECDSA Key Generation function tests the point multiplication function used in the Dual_EC_DRBG..

Validation List:

NIST maintains the current [DRBG Validation List](#).

Test Vectors:

DRBG Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of a DRBG algorithm implementation using the DRBGVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[DRBG Test Vectors](#) DRBG Test Vectors In this zip file, there are 4 text files with NIST SP 800-90 DRBG testvectors: *HASH_DRBG.txt*, *HMAC_DRBG.txt*, *CTR_DRBG.txt*, and *Dual_EC_DRBG.txt*.

DRBG:			
	Hash DRBG	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
		Prediction Resistance Supported	
		Reseed not implemented	
	HMAC DRBG	SHA-1	
		SHA-224	
		SHA-256	
		SHA-384	
		SHA-512	
		Prediction Resistance Supported	
		Reseed not implemented	
	CTR DRBG	3 key TDEA	
		AES 128	
		AES 192	

		AES 256	
		Derivation function	
		No derivation function	
		Prediction Resistance Supported	
		Reseed not implemented	
	Dual EC DRBG	P256: SHA-1	
		P256: SHA-224	
		P256: SHA-256	
		P256: SHA-384	
		P256: SHA-512	
		P384: SHA-224	
		P384: SHA-256	
		P384: SHA-384	
		P384: SHA-512	
		P521: SHA-256	
		P521: SHA-384	
		P521: SHA-512	
		Prediction Resistance Supported	
		Reseed not implemented	

Key Management –Key Agreement Schemes and Key Confirmation (KAS)

[SP 800-56A Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography \(Revised March 2007\)](#) specifies key establishment schemes based on standards developed by the Accredited Standards Committee (ASC) X9, Inc.: ANS X9.42 (Agreement of Symmetric Keys Using Discrete Logarithm Cryptography) and ANS X9.63 (Key Agreement and Key Transport Using Elliptic Curve Cryptography).

Testing Requirements:

atsec can test for conformance to the Key Agreement Schemes (KAS) and Key Confirmation algorithms specified in Special Publication 800-56A. The testing requirements for this algorithm can be found in the document titled [The KAS Validation System \(KASVS\)](#). **Additional testing note:** The KASVS validation process requires additional prerequisite testing of the underlying DSA and/or ECDSA algorithm for both domain parameter generation and key pair generation (determined by which type of cryptography is supported), the supported SHA algorithm(s), supported MAC algorithm(s) (CCM, CMAC, and/or HMAC), and the supported RNG and/or DRBG algorithm(s).

Validation List:

NIST maintains the current [KAS Validation List](#).

Test Vectors:

KAS Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of a key agreement scheme and key confirmation algorithm implementation using the KASVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[KAS Test Vectors](#) See the KASVS document for an explanation of the files.

KAS FFC			
	dhHybrid1	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	
	MVQ1	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	
	MVQ2	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	

	dhStatic	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	
	dhEphem	initiator	
		responder	
	dhOneflow	initiator	
		responder	
	dhHybridOneflow	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	

KAS ECC			
	Full Unified	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	
	Ephemeral Unified	initiator	
		responder	
	Full MVQ	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	
	One Pass Unified	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	
	One Pass MVQ	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	

		Key confirmation: bilateral	
		responder	
	One Pass DH	initiator	
		responder	
		Key confirmation	
	Static Unified	initiator	
		responder	
		Key confirmation: provider	
		Key confirmation: recipient	
		Key confirmation: unilateral	
		Key confirmation: bilateral	

Message Authentication (MAC)

Block Cipher-based MAC Algorithm (CMAC)

The CMAC algorithm is specified in [Special Publication 800-38B](#) dated May 2005, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. CMAC can be considered a mode of operation of the block cipher because it is based on an approved symmetric key block cipher, such as the Advanced Encryption Standard (AES) algorithm currently specified in Federal Information Processing Standard (FIPS) Pub. 197. CMAC is also an approved mode of the Triple Data Encryption Algorithm (TDEA).

Testing Requirements:

atsec can test for conformance to the CMAC algorithm in Special Publication 800-38B. The testing requirements for this algorithm can be found in the document titled [The CMAC Validation System \(CMACVS\)](#).

Additional testing note: The underlying NIST-Approved symmetric key algorithm must be validated as part of the CMAC validation. Currently, NIST approves both the AES and TDES algorithms for use with CMAC.

Validation List:

NIST maintains the current CMAC Validations. CMAC Validations are included on the validation list of its approved symmetric key block cipher -- therefore it is included on either the [AES Validation List](#) or the [TDES Validation List](#).

Test Vectors:

CMAC Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of a CMAC algorithm implementation using the CMACVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[CMAC Test Vectors](#)

CMAC:			
	Generate with AES	AES 128	
		AES 192	
		AES 256	
	Verify with AES	AES 128	
		AES 192	
		AES 256	
	Generate with TDES	2-Key TDES	
		3-Key TDES	
	Verify with TDES	2-Key TDES	
3-Key TDES			



Counter with Cipher Block Chaining-Message Authentication Code (CCM)

The Counter with Cipher Block Chaining-Message Authentication Code (CCM) is specified in [Special Publication 800-38C](#) dated May, 2004, Counter with Cipher Block Chaining-Message Authentication Code (CCM). CCM is based on an approved symmetric key block cipher algorithm whose block size is 128 bits, such as the Advanced Encryption Standard (AES) algorithm currently specified in Federal Information Processing Standard (FIPS) Pub. 197 [2]; thus, CCM cannot be used with the Triple Data Encryption Algorithm [3], whose block size is 64 bits. Currently the only NIST-Approved 128 bit symmetric key algorithm is AES.

Testing Requirements:

atsec can test for conformance to the CCM algorithm in Special Publication 800-38C. The testing requirements for this algorithm can be found in the document titled [The Counter with Cipher Block Chaining-Message Authentication Code \(CCM\) Validation System \(CCMVS\)](#). Additional testing note: The underlying NIST-Approved 128 bit symmetric key algorithm must be validated as part of the CCM validation. Currently, the only 128 bit symmetric key algorithm approved by NIST is AES.

Validation List:

NIST maintains the current CCM Validations. CCM Validations are included on the validation list of its approved symmetric key block cipher whose block size is 128 bits-- therefore it is included on the [AES Validation List](#). NIST maintains the original [CCM Validation List](#) for historical purposes. The information contained on the CCM Validation List has been duplicated in the AES Validation List.

Test Vectors:

CCM Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of a CCM algorithm implementation using the CCMVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[CCM Test Vectors](#)

CCM:				
	AES Key size	128		
		192		
		256		
	Associated Data Length Range [0, 32]			
	Payload Length [0, 32]			
	Nonce Length	7		
		8		
		9		
		10		
		11		
		12		
		13		
	Tag Length	4		
		6		
		8		
		10		
		12		
16				



Galois/Counter Mode (GCM) and GMAC

The Galois/Counter Mode (GCM) and GMAC is specified in [Special Publication 800-38D](#) dated November, 2007, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. GCM is based on an approved symmetric key block cipher algorithm whose block size is 128 bits, such as the Advanced Encryption Standard (AES) algorithm currently specified in Federal Information Processing Standard (FIPS) Pub. 197 [2]; thus, GCM cannot be used with the Triple Data Encryption Algorithm [3], whose block size is 64 bits. Currently the only NIST-Approved 128 bit symmetric key algorithm is AES.

Testing Requirements:

atsec can test for conformance to the GCM and GMAC algorithms in Special Publication 800-38D. The testing requirements for this algorithm can be found in the document titled [The Galois/Counter Mode \(GCM\) and GMAC Validation System \(GCMVS\)](#). Additional testing note: The underlying NIST-Approved 128 bit symmetric key algorithm must be validated as part of the GCM validation. Currently, the only 128 bit symmetric key algorithm approved by NIST is AES.

Validation List:

NIST maintains the current GCM Validations. GCM Validations are included on the validation list of its approved symmetric key block cipher whose block size is 128 bits-- therefore it is included on the [AES Validation List](#).

Test Vectors:

GCM Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of a GCM algorithm implementation using the GCMVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[GCM Test Vectors](#)

GCM:					
	Mode	Encrypt			
		Decrypt			
	Key size	128			
		192			
		256			
	Tag length	128			
		120			
		112			
		104			
		96			
		64			
		32			
			96 bit IV supported		
			Other IV		



Keyed-Hash Message Authentication Code (HMAC)

The Keyed-Hash Message Authentication Code (HMAC) is specified in [FIPS 198](#) dated March 6, 2002, Keyed-Hash Message Authentication Code (HMAC). This algorithm utilizes the Secure Hash Algorithms as an underlying primitive.

Testing Requirements:

atsec can test for conformance to the HMAC algorithm in FIPS 198. The testing requirements for these algorithms can be found in the document titled [The Keyed-Hash Message Authentication Code \(HMAC\) Validation System \(HMACVS\)](#). Additional testing note: All underlying SHA algorithm(s) supported by the HMAC implementation must be validated as part of the HMAC validation.

Validation List:

NIST maintains the current [HMAC Validation List](#).

Test Vectors:

HMAC Test Vectors - These files provide an electronic version of the test vectors that can be used to informally verify the correctness of an HMAC algorithm implementation using the HMACVS. **However, use of these vectors does not take the place of validation obtained through the Cryptographic Algorithm Validation Program (CAVP).**

[HMAC Test Vectors](#)

HMAC:	
	SHA-1
	SHA-224
	SHA-256
	SHA-384
	SHA-512



Non FIPS Approved algorithms

atsec can also test the following algorithms as a service to customers still specifying these algorithms and who want to ensure their implementation correctness, but these **are not validated or certified by the NIST CAVP**.

Data Encryption Standard (DES)

FIPS 46-3, Data Encryption Standard (DES), was withdrawn May 19, 2005 because the cryptographic algorithm no longer provided the security that is needed to protect Federal government information. DES is no longer an Approved algorithm. The DES Algorithm Validation Webpage is still accessible via the Validations List webpage, for historical purposes only.

	ECB	Encrypt, Decrypt
	CBC	Encrypt, Decrypt
	CFB	1 Bit Encrypt Decrypt
		8 Bits Encrypt Decrypt
		64 Bits Encrypt Decrypt
	OFB	Encrypt Decrypt

Message Authentication Code (MAC)

The MAC Validation System (MVS) tested for compliance with [FIPS 113](#), Computer Data Authentication. A list of [validated products](#) is maintained by the Security Technology Group.

MAC	Contact atsec for details
------------	----------------------------------

Data (Message) Authentication Code (MAC) and Key Management Using ANSI X9.17

The automated conformance tests for FIPS 113 and 171 are no longer operational. Currently, if a FIPS 140-1 or FIPS 140-2 cryptographic module implements either of these two standards, the [CMT testing laboratories](#) perform some testing that these FIPS requirements are implemented correctly in the cryptographic module.

The Key Management Validation System (KMVS) tested for compliance with FIPS 171, Key Management Using ANSI X9.17. A list of [validated products](#) is maintained by the Security Technology Group.

FIPS 113	Contact atsec for details
FIPS 171	Contact atsec for details

RC4	Contact atsec for details
------------	----------------------------------



Blowfish		Contact atsec for details	
HOTP		Contact atsec for details	
CRC		Contact atsec for details	