

Specifying Cryptographic Requirements in Security Targets
CCEVS Guidance
Version 1.0
5 January 2007

Purpose: This paper provides guidance on how cryptographic protocols are to be specified in Security Targets.

Background: There has been a dramatic increase in the number of products being submitted to CCEVS for evaluation that employ cryptographic protocols to perform or support security functionality. There are standards that describe what these cryptographic protocols are to provide, and in some cases how that functionality is to be provided, however many of the standards are ambiguous, or contain options that may or may not be implemented. Vendors also implement variations in their product that are not contained in a given standard. It is critical that cryptographic protocols are specified clearly in the appropriate requirements language because the requirements form the basis for analysis and testing of the design and implementation.

Specifying FCS_CKM requirements:

When cryptography is implemented within the TOE and it is responsible for fulfilling any of the TOE Security Functional Requirements (SFRs) (e.g., digital signature to provide source authentication), then the ST is required to describe such functionality.

When referencing a standard, the SFR should contain the level of detail necessary in order to properly define the cryptography according to the given standard.

FCS_CKM_SYM_EXP.1 gives such an example for the symmetric key establishment.

FCS_CKM_SYM_EXP.1 Cryptographic Key Establishment for AES symmetric keys

FCS_CKM_SYM_EXP.1.1 – The cryptomodule shall provide the following FIPS-approved cryptographic key establishment using Discrete Logarithm Key Agreement that meets the following:

- a) The cryptomodule shall provide the capability to act as the initiator or responder (that is, act as Party U or Party V as defined in the standard) to agree on cryptographic keys of all sizes using the [**selection: dhStatic, dhEphem, dhOneFlow, dhHybrid1, dhHybrid2, dhHybridOneFlow, MQV1, MQV2**] key agreement scheme where domain parameter p is a prime of [**assignment: size of prime “p” in number of bits that is 3072 or greater**] and domain parameter q is a prime of [**assignment: size of prime “q” in number of bits that is 256 or greater**], and that conforms with ANSI X9.42, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography.

Application Note: It should be noted that the actual key size of the symmetric key agreed to when using this scheme will be a function of the algorithm that will be using the key, as specified in FCS_COP_EXP.2.

In the selection in paragraph a), one or more of the schemes should be chosen by the ST author, based on what schemes the TOE implements. Note that the requirement is for the cryptomodule to be able to act as either party (as detailed in the standard) for the chosen scheme(s).

The two assignments are used to specify the number of bits used for the domain parameters p and q (which are primes). The requirement above indicates that p must be a prime of at least 3072 bits, while q must be a prime of at least 256 bits. The ST author should fill in the appropriate number of bits based on the implementation. This applies if the implementation generates its own domain parameters, or if it obtains the domain parameters in some other way (e.g., hard-coded, obtained from an outside authority). Appendix A provides a list of algorithms, their appropriate standards and possible key sizes. The specific standards should be referenced, when appropriate.

- b) The cryptomodule shall conform to the standard using a FIPS-approved MAC function, a FIPS-approved Random Number generation function, and a FIPS-approved Hashing function.

Application Note: It should be noted that this portion of the SFR mandates that the functionality be present. The appropriate COP requirement must also be included, which includes the specifics of the hashing or random number generation. Appendix A includes information regarding which requirements need to be addressed when a given algorithm is used.

- c) The choices and options used in conforming to the key agreement scheme(s) are as follows: **[assignment: options that the cryptomodule implements when implementing the selected key agreement schemes, including options for any prerequisite or dependant functions (e.g., domain parameter generation and validation).];**

Application Note: In the X9.42 standard there are several sections that are marked “optional”, or where a choice is given. Choices are, for example, how the domain parameters are obtained (generated or obtained from some other entity). Another example is the key derivation function that is implemented. ST authors should use the assignment to provide sufficient information so that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the key agreement schemes implemented. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment. Other examples include security attributes such as Implicit/Explicit Key Authentication, Known-Key Security, and Unknown Key-share Resilience.

Another example are the standards that contain options that must be reflected in an SFR are those used in specifying digital signature generation and verification, specifically ANSI X9.31, and ANSI X9.62. This example demonstrates the expectations for specifying the behavior for both rDSA and ECDSA. An ST does not have to include both methods, this simply illustrates what one would specify for either of the methods.

Specifying FCS_COP requirements:

**Cryptographic Operation (Digital Signature Generation/Verification)
(FCS_COP_EXP.3)**

FCS_COP_EXP.3.1 – A FIPS-validated cryptomodule shall perform digital signature generation and verification using the FIPS-Approved Security Functions that meet the FIPS 140-2 standard [*selection*]:

- *rDSA*

Application Note: This top-level selection indicates that the digital signatures will be calculated using the rDSA algorithm specified in X9.31-1998, as implemented in a FIPS-validated cryptomodule.

- a) The cryptomodule shall implement rDSA with a modulus size of [ST assignment: *size of modulus “n” in number of bits that is 2048 bits or greater*] in a manner that conforms to ANSI X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).

Application Note: The ST author should fill in the assignment with the number of bits the module uses for its moduli. Note that in order to meet the requirement moduli must be at least 2048 bits.

- b) The choices and options used in conforming to the X9.31-1998 are as follows: **[ST assignment: options that the cryptomodule implements when implementing the signature generation and validation functions, including options for any prerequisite or dependant functions (e.g., key generation)];**

Application Note: In the X9.31-1998 standard there are several sections that are marked “optional”, or where a choice is given. For instance, the public verification exponent “e” can be fixed or randomly generated. Another instance is that the procedure in section 4.1.2.1 can be followed to generate the primes p and q, or another procedure followed as long as the primes generated meet the conditions in section 4.1.2. The goal of the assignment is to provide sufficient information such that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the rDSA implementation. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment.

- *ECDSA*

Application Note: This top-level selection indicates that the digital signatures will be calculated using the ECDSA algorithm specified in X9.62-1998, as implemented in a FIPS-validated cryptomodule.

- a) The FIPS-validated cryptomodule shall implement ECDSA where the order of the base point is a [ST assignment: *size of the order of the base point “n” in number of bits that is 256 or greater*]-bit value, and where the algorithm conforms with ANSI X9.62-1998, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).

Application Note: The assignment is used to specify the number of bits used for the domain parameter n, which is the order of the base point of the curve chosen (the standard uses “n” to denote this value). The requirement above indicates that n must be at least a 256-bit value. The ST author should fill in the appropriate number of bits based on the implementation. This applies if the implementation generates its own domain parameters, or if it obtains the domain parameters in some other way (e.g., hard-coded, obtained from an outside authority).

- b) The choices and options used in conforming to X9.62-1998 are as follows:
[ST assignment: options that the TSF implements when implementing the signature generation and validation functions, including options for any prerequisite or dependant functions (e.g., domain parameter generation and validation)].

Application Note: In the X9.62-1998 standard there are several sections that are marked “optional”, or where a choice is given. Choices are, for example, in the domain parameter generation and validation section (Section 5.1) where domain parameters can be generated over F_p or over F_2^m . Public Key validation is an example of an optional part of the standard. ST authors should use the assignment to provide sufficient information such that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the key transport schemes implemented. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment.

The purpose of requiring this level of specificity in an ST is not to regurgitate what exists in the applicable standard, rather it is to make it explicitly clear what the developer of the TOE has implemented with respect to the standard. This then forms the basis for the evaluation activities, and provides end users information to provide them the ability to determine if the TOE is appropriate for their environment. In addition to the FIPS approved protocols, CCEVS will maintain a list of “other approved” cryptographic protocols, and their associated standards. The reason for this is CCEVS does not want to validate products that contain protocols that are inherently vulnerable (e.g., SSHv1).

In the cases where the standards are not freely available (e.g., ANSI standards), the CCTL is required to provide CCEVS with a copy of the standard for oversight purposes.

Specifying within the Security Environment:

When the TOE depends upon the IT environment to provide cryptographic functionality, the ST must state that it depends on the environment for such functionality and that it meets a given standard.

Specifying within the TSS:

Since the SFR is intended to define the implementation of the cryptographic algorithm(s), the TSS must define other areas of importance to the CCEVS. These areas include specification regarding how the cryptographic protocols are used to protect information within the TOE and what information is being protected. The TSS must also define how the cryptographic module and protocol implementation is protected, along with protection of the keys. Furthermore, if cryptography is used within the TOE but is not used for security related functionality, the ST should not make any claims as to the provided cryptographic functionality.

Appendix A: Cryptographic Algorithm Specification

	Security Service	Algorithm Standard	Key Generation Standard	Mode of Operation	Key Size in Bits	Other Required Algorithms
Digital Signature						
RSA	Digital Signature	ANSI X9.31; PKCS # 1, V1.5; PKCS #1 PSS;	ANSI X9.31 ANSI X9.80	Not Applicable	1024, 2048, 3072....	Must having a secure hashing algorithm
DSA	Digital Signature	FIPS 186-2	FIPS 186-2	Not Applicable	1024 + 256*i, where i>=0	Must use SHA-1 or other SHA
ECDSA	Digital Signature	FIPS 186-2 ANSI X9.62	ANSI X9.62	Not Applicable	160, 224, 256, 384, 512	Must use SHA-1 or other SHA
Key Establishment						
RSA	Key Encryption (Key Transfer)	ANSI X9.44; PKCS # 1, V1.5; PKCS #1 OAEP;	ANSI X9.31 ANSI X9.80	Not Applicable	1024, 2048, 3072....	Generally a data encryption algorithm
Diffie Hellman (DH)	Key Exchange (key Agreement; Key Calculation; Key Establishment)	SP 800-56 ANSI X9.42	SP 800-53 ANSI X9.42 FIPS 186-3	dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow dhStatic, MQV1, and MQV2	1024, 2048, 3072	Generally a data encryption algorithm
ECDH	Key Exchange (key Agreement; Key Calculation; Key Establishment)	SP 800-56 ANSI X9.63	SP 800-53 ANSI X9.63 FIPS 186-3	Full Unified Model, Ephemeral Unified Model, One-Pass Unified Model, One-Pass Diffie-Hellman, Static Unified Model, Full MQV and One-Pass MQV	160, 224, 256, 384, 512	Generally a data encryption algorithm
Hash						
SHA-1	Hashing	FIPS 180-2	Not Applicable	Not Applicable	Not Applicable	None

	Security Service	Algorithm Standard	Key Generation Standard	Mode of Operation	Key Size in Bits	Other Required Algorithms
SHA-224	Hashing	FIPS 180	Not Applicable	Not Applicable	Not Applicable	None
SHA-256	Hashing	FIPS 180	Not Applicable	Not Applicable	Not Applicable	None
SHA-384	Hashing	FIPS 180	Not Applicable	Not Applicable	Not Applicable	None
SHA-512	Hashing	FIPS 180	Not Applicable	Not Applicable	Not Applicable	None
MAC						
HMAC	Data Integrity; Source Authentication	FIPS 198 RFC 2104	FIPS 140-2 Annex C	Not Applicable	80 for SHA-1 112 for SHA-224 128 for SHA-256 192 for SHA-384 256 for SHA-512	Generally, there is a key encryption or key exchange algorithm
TDES-MAC	Data Integrity; Source Authentication	SP 800-38B	FIPS 46-3	Not Applicable	168	Generally, there is a key encryption or key exchange algorithm
AES-MAC	Data Integrity; Source Authentication	SP 800-38B	FIPS 197	Not Applicable	128, 192, 256	Generally, there is a key encryption or key exchange algorithm
TDES-CCM ¹	Data Integrity; Source Authentication	SP 800-38C	FIPS 46-3	Not Applicable	168	Generally, there is a key encryption or key exchange algorithm
AES-CCM ²	Data Integrity; Source Authentication	SP 800-38C	FIPS 197	Not Applicable	128, 192, 256	Generally, there is a key encryption or key exchange algorithm
Encryption						
TDES	Confidentiality (data	FIPS 46-3	FIPS 46-3	CBC, ECB, OFB,	168	Generally, there is a

¹ Also provides data confidentiality.

² Also provides data confidentiality

	Security Service	Algorithm Standard	Key Generation Standard	Mode of Operation	Key Size in Bits	Other Required Algorithms
	encryption)			CFB, Counter		key encryption or key exchange algorithm
AES	Confidentiality (data encryption)	FIPS 197	FIPS 197	CBC, ECB, OFB, CFB, CCM, Counter	128, 192, 256	Generally, there is a key encryption or key exchange algorithm
Skipjack	Confidentiality (data encryption)	Skipjack	Skipjack	CBC, ECB, OFB, CFB, Counter	80	Generally, there is a key encryption or key exchange algorithm
RNG						
PRNG (Deterministic)	Random Number	FIPS 140-2 Annex C	Not Applicable	Must specify which of the methods listed in the Annex is used	Size and entropy of seed must be equal to or greater than the symmetric key the PRNG is used to generate. For DSA, DH, ECDSA, ECDH must be equal to or greater than the private value. For RSA seed must be 100 bits or more.	Generally there is another algorithm that uses the PRNG
RNG (non-deterministic)	Random Number	ANSI X9.82	Not Applicable	Not Applicable	.	Generally there is another algorithm that uses the RNG