RSA LABORATORIES

PKCS #11 v2.20 Amendment 3 - Draft 1

PKCS#11 Mechanisms for SHA 224 and AES in Counter Mode

RSA Laboratories

August 30, 2006

Editor's note: This is the first draft of this amendment. Comments and feedback are welcome and should be sent to the Cryptoki mailing list (Cryptoki@rsasecurity.com) or the PKCS editor (pkcs-editor@rsasecurity.com)

Table of Contents

List of Tables

1 Introduction

This document is an amendment to PKCS #11 v2.20 [1] and describes extensions to PKCS #11 to support the SHA-224 described in [2].

2 Definitions

SHA-224 The Secure Hash Algorithm with a 224-bit message digest, as defined in RFC 3874.

3 Mechanisms

The following table shows, for the mechanisms defined in this document, their support by different cryptographic operations. For any particular token, of course, a particular operation may well support only a subset of the mechanisms listed. There is also no guarantee that a token that supports one mechanism for some operation supports any other mechanism for any other operation (or even supports that same mechanism for any other operation).

Table 1, Mechanisms vs. Functions

Mechanism	Functions						
	Encrypt & Decrypt	Sign & Verify	SR & VR[1]	Digest	Gen. Key/ Key Pair	Wrap & Unwrap	Derive
CKM_SHA224				✓			
CKM_SHA224_HMAC		✓					
CKM_SHA224_HMAC_GENERAL		✓					
CKM_SHA224_RSA_PKCS		✓					
CKM_SHA224_RSA_PKCS_PSS		✓					
CKM_SHA224_KEY_DERIVATION							✓
CKM_AES_CTR	✓					✓	

The remainder of this section will present in detail the mechanisms and the parameters which are supplied to them.

3.1 RSA additional variants

For completeness and consistency with all the other SHA variants the following additions have been made to include the SHA-224 variant of these mechanisms.

3.1.1 Definitions

Mechanisms:

CKM_SHA224_RSA_PKCS


```
CKM_SHA224_RSA_PKCS_PSS
```

3.1.2 PKCS #1 RSA OAEP mechanism parameters

The following table lists the added MGF functions.

Table 2, PKCS #1 Mask Generation Functions

Source Identifier	Value
CKG_MGF1_SHA224	0x00000005

3.1.3 PKCS #1 v1.5 RSA signature with SHA-224

The PKCS #1 v1.5 RSA signature with SHA-224 mechanism, denoted **CKM_SHA224_RSA_PKCS** perform similarly as the other **CKM_SHA*X*_RSA_PKCS** mechanisms but using the SHA-224 hash functions.

3.1.4 PKCS #1 RSA PSS signature with SHA-224

The PKCS #1 RSA PSS signature with SHA-224 mechanism, denoted **CKM_SHA224_RSA_PKCS_PSS**, perform similarly as the other **CKM_SHA*X*_RSA_PSS** mechanisms but using the SHA-224, hash functions.

3.2 SHA-224

3.2.1 Definitions

Mechanisms:

```
CKM_SHA224
CKM_SHA224_HMAC
CKM_SHA224_HMAC_GENERAL
CKM_SHA224_KEY_DERIVATION
```

3.2.2 SHA-224 digest

The SHA-224 mechanism, denoted **CKM_SHA224**, is a mechanism for message digesting, following the Secure Hash Algorithm with a 224-bit message digest defined in [2].

It does not have a parameter.

Constraints on the length of input and output data are summarized in the following table. For single-part digesting, the data and the digest may begin at the same location in memory.

Table 3, SHA-224: Data Length

Function	Input length	Digest length
C_Digest	any	28

3.2.3 General-length SHA-224-HMAC

The general-length SHA-224-HMAC mechanism, denoted **CKM_SHA224_HMAC_GENERAL**, is the same as the general-length SHA-1-HMAC mechanism except that it uses the HMAC construction based on the SHA-224 hash function and length of the output should be in the range 0-28 The keys it uses are generic secret keys. FIPS-198 compliant tokens may require the key length to be at least 14 bytes; that is, half the size of the SHA-224 hash output.

It has a parameter, a **CK_MAC_GENERAL_PARAMS**, which holds the length in bytes of the desired output. This length should be in the range 0-28 (the output size of SHA-224 is 28 bytes). FIPS-198 compliant tokens may constrain the output length to be at least 4 or 14 (half the maximum length). Signatures (MACs) produced by this mechanism will be taken from the start of the full 28 byte HMAC output.

Table 4, General-length SHA-224-HMAC: Key And Data Length

Function	Key type	Data length	Signature length
C_Sign	generic secret	Any	0-28, depending on parameters
C_Verify	generic secret	Any	0-28, depending on parameters

3.2.4 SHA-224-HMAC

The SHA-224-HMAC mechanism, denoted **CKM_SHA224_HMAC**, is a special case of the general-length SHA-224-HMAC mechanism.

It has no parameter, and always produces an output of length 28.

3.2.5 SHA-224 key derivation

SHA-224 key derivation, denoted **CKM_SHA224_KEY_DERIVATION**, is the same as the SHA-1 key derivation mechanism in Section 12.21.5, except that it uses the SHA-224 hash function and the relevant length is 28 bytes.

3.3 AES Counter Mode

3.3.1 Definitions

Mechanisms:

```
CKM_AES_CTR
```

3.3.2 AES Counter Mode mechanism parameters

♦ **CK_ AES_CTR _PARAMS; CK_ AES_CTR _PARAMS_PTR**

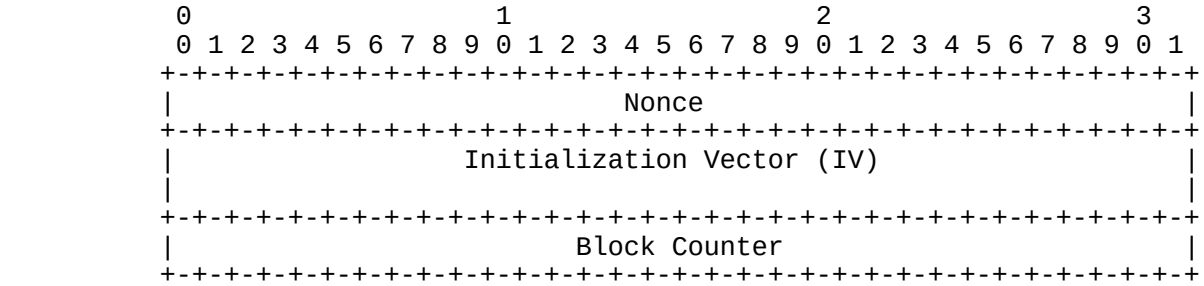
CK_ AES_CTR _PARAMS is a structure that provides the parameters to the **CKM_AES_CTR** mechanism. It is defined as follows:

```
typedef struct CK_AES_CTR_PARAMS {
        CK_ULONG ulCounterBits;
        CK_BYTE cb[16];
} CK_AES_CTR_PARAMS;
```

The fields of the structure have the following meanings:

ulCounterBits specifies the number of bits in the counter block (cb) that shall be incremented. This number shall be such that $0 < ulCounterBits \leq 128$. For any values outside this range the mechanism shall return **CKR_MECHANISM_PARAM_INVALID**.

cb counterblock. It's up to the caller to initialize all of the bits in the counter block including the counter bits. The counter bits are the least significant bits of the counter block (*cb*). They are a big-endian value usually starting with 1. The rest of *cb* is for the nonce, and maybe an optional IV. E.g. as defined in RFC 3686 [4]:



This construction permits each packet to consist of up to: 2^{32}-1 blocks = 68,719,476,720 octets.

CK_ AES_CTR _PARAMS_PTR is a pointer to a **CK_ AES_CTR _PARAMS**.

3.3.3 AES Counter Mode Encryption / Decryption

Generic AES counter mode is described in NIST Special Publication 800-38A [3], and in RFC 3686 [4]. These describe encryption using a counter block which may include a nonce to guarantee uniqueness of the counter block. Since the nonce is not incremented, the mechanism parameter must specify the number of counter bits in the counter block.

The block counter is incremented by 1 after each block of plaintext is processed. There is no support for any other increment functions in this mechanism.

If an attempt to encrypt/decrypt is made which will cause an overflow of the counter block's counter bits to be used then the mechanism shall return **CKR_DATA_LEN_RANGE**. Note that the mechanism should allow the final post increment of the counter to overflow (if it implements it this way) but not allow any further processing after this point. E.g. if *ulCounterBits* = 2 and the counter bits start as 1 then only 3 blocks of data can be processed.

A. Manifest constants

The following definitions can be found in the appropriate header file.

```
#define CKM_SHA224                      0x00000255
#define CKM_SHA224_HMAC                 0x00000256
#define CKM_SHA224_HMAC_GENERAL         0x00000257
#define CKM_SHA224_RSA_PKCS             0x00000046
#define CKM_SHA224_RSA_PKCS_PSS         0x00000047
#define CKM_SHA224_KEY_DERIVATION       0x00000396

#define CKM_AES_CTR                     0x00001086

#define CKG_MGF1_SHA224                 0x00000005
```

B. Intellectual property considerations

RSA Security makes no patent claims on the general constructions described in this document, although specific underlying techniques may be covered.

C. References

[1] RSA Laboratories. *PKCS #11: Cryptographic Token Interface Standard*. Version 2.20, June 2004. URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf

[2] R. Housley. "*A 224-bit One-way Hash Function: SHA-224*," IETF RFC 3874, September 2004. URL: http://ietf.org/rfc/rfc3874.txt

[3] NIST Special Publication 800-38A. "*Recommendation for Block Cipher Modes of Operation*."

[4] R. Housley. "*Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)*," IETF RFC 3686, January 2004. URL: http://ietf.org/rfc/rfc3686.txt.

D. About PKCS

The *Public Key Cryptography Standards* are documents produced by RSA Security in cooperation with secure systems developers for the purpose of simplifying integration and management of accelerating the deployment of public-key cryptography and strong authentication technology into secure applications, and to enhance the user experience of these technologies.

RSA Security plans further development of the PKCS series through mailing list discussions and occasional workshops, and suggestions for improvement are welcome. Results may also be submitted to standards forums. For more information, contact:

PKCS Editor
RSA Security
174 Middlesex Turnpike
Bedford, MA 01730 USA
pkcss-editor@rsasecurity.com
http://www.rsasecurity.com/rsalabs/