PRE-PUBLICATION
NIST Special Publication 800-38F
November, 2012

Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping

Morris Dworkin

National Institute of Standards and Technology

U.S. Department of Commerce

C O M P U T E R S E C U R I T Y

NIST Special Publication 800-38F
PRE-PUBLICATION

Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping

Morris Dworkin

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

November 2012

U.S. Department of Commerce
Rebecca M. Blank, Acting Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Director

Reports on Information Security Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security, and its collaborative activities with industry, government, and academic organizations.

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 800-38F
Natl. Inst. Stand. Technol. Spec. Publ. 800-38F, 30 pages (November 2012)
CODEN: NSPUE2

Acknowledgements

The author wishes to thank his colleagues who reviewed drafts of this publication and contributed to its development, especially Elaine Barker, Meltem Turan, Allen Roginsky, Lily Chen, Donghoon Chang, Sharon Keller, Tim Polk, John Kelsey, Ray Perlner, Tim Hall, and Souradyuti Paul. Heather Pearce worked extensively on early drafts of this publication. Phillip Rogaway kindly provided Figure 1 from his paper analyzing key-wrap modes. The author also gratefully acknowledges the comments from the public and private sectors to improve the quality of this publication.

Abstract

This publication describes cryptographic methods that are approved for "key wrapping," i.e., the protection of the confidentiality and integrity of cryptographic keys. In addition to describing existing methods, this publication specifies two new, deterministic authenticated-encryption modes of operation of the Advanced Encryption Standard (AES) algorithm: the AES Key Wrap (KW) mode and the AES Key Wrap With Padding (KWP) mode. An analogous mode with the Triple Data Encryption Algorithm (TDEA) as the underlying block cipher, called TKW, is also specified, to support legacy applications.

KEY WORDS: authenticated encryption; authentication; block cipher; computer security; confidentiality; cryptography; encryption; information security; key wrapping; mode of operation.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1 Purpose

This publication is the sixth part in a series of Recommendations regarding the modes of operation of block ciphers. The purpose of this part is to provide approved methods for key wrapping, i.e., the protection of cryptographic keys.

2 Authority

This publication has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems.

This recommendation has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright. (Attribution would be appreciated by NIST.)

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official.

Conformance testing for implementations of this Recommendation will be conducted within the framework of the Cryptographic Algorithm Validation Program (CAVP) and the Cryptographic Module Validation Program (CMVP). The requirements of this Recommendation are indicated by the word “shall.” Some of these requirements may be out-of-scope for CMVP or CAVP validation testing, and thus are the responsibility of entities using, implementing, installing, or configuring applications that incorporate this Recommendation.

3 Introduction

3.1 Overview

This Recommendation specifies a deterministic authenticated-encryption mode of operation of the Advanced Encryption Standard (AES) block cipher [3]. The mode is called AES Key Wrap, abbreviated as KW in this Recommendation. Although KW can be used in conjunction with any reversible padding scheme, a variant of KW with an internal padding scheme is also specified to promote interoperability. This variant is called AES Key Wrap With Padding, abbreviated as KWP. The analogue of KW with the Triple Data Encryption Algorithm (TDEA) [5] as the underlying block cipher is also specified, to support legacy applications. This analogue is called Triple DEA Key Wrap, abbreviated as TKW.

KW, KWP, and TKW were designed to protect the confidentiality and the authenticity/integrity of cryptographic keys. Each provides an option for protecting keys in a manner that is distinct from the methods that protect general data. Segregating keys from general data can provide an extra layer of protection.

Nevertheless, there is no requirement to protect cryptographic keys with a distinct cryptographic method. Previously approved authenticated-encryption modes—as well as combinations of an approved encryption mode with an approved authentication method—are approved for the protection of cryptographic keys, in addition to general data.

Similarly, KW, KWP, and TKW are each approved for the protection of general data, as well as cryptographic keys.

KW, KWP, and TKW are each robust in the sense that each bit of output can be expected to depend in a nontrivial fashion on each bit of input, even when the length of the input data is greater than one block. This property is achieved at the cost of a considerably lower throughput rate, compared to other authenticated-encryption modes, but the tradeoff may be appropriate for some key management applications. For example, a robust method may be desired when the length of the keys to be protected is greater than the block size of the underlying block cipher, or when the value of the protected data is very high.

3.2 Related Specifications

Earlier specifications of key-wrap algorithms[1] that are related to KW, KWP, and TKW are discussed in this subsection.

In 2001, NIST posted a document entitled “AES Key Wrap Specification” on NIST’s Computer Security Resource Center web site as an unofficial suggestion for the protection of cryptographic keys. That algorithm is essentially equivalent to KW as specified in this Recommendation.

In 2002, two industry groups published specifications of key-wrap algorithms that were based on the specification that NIST posted. First, the Internet Engineering Task Force (IETF) developed an essentially equivalent specification in Request For Comments (RFC) 3394 [11]. Second, the Telecommunications Industry Association published a protocol for Digital Radio Over-the-Air-Rekeying [1] containing a “Key Wrap Algorithm” that supported TDEA, in addition to the AES block cipher. Those algorithms are essentially equivalent to KW and TKW as specified in this Recommendation.

In 2008, Accredited Standards Committee X9, Inc., published a key-wrap standard for the financial services industry [2]. The variant of KW in that standard features a more general framework for formatting the input data, including a padding scheme, as well as an analogue with TDEA as the underlying block cipher.

[1] The term “algorithm” here indicates a high-level cryptographic technique that may encompass more than one computational procedure; for example, an "encryption algorithm" like TDEA or the AES algorithm has transformations for both encryption and decryption. This publication also contains ten numbered algorithms in the original sense of the word, i.e., as a list of instructions for executing a single computational procedure.

In 2009, a different padding scheme was specified in RFC 5649 [4], referencing elements of the specification in [11]. The resulting algorithm, called AES Key Wrap With Padding, is essentially equivalent to KWP as specified in this Recommendation.

The differences between KW, KWP, and TKW as specified in this Recommendation and their earlier specifications are described in Appendix B.1.

4 Definitions and Notation

4.1 Definitions

approved	FIPS-approved or NIST-recommended: an algorithm or technique that is either 1) specified in a FIPS or a NIST Recommendation, or 2) adopted in a FIPS or a NIST Recommendation.
authenticated-encryption function	A function that encrypts plaintext into ciphertext and provides a means for the associated authenticated-decryption function to verify the authenticity and, therefore, the integrity of the data.
authenticated-decryption function	A function that decrypts purported ciphertext into corresponding plaintext and verifies the authenticity and, therefore, the integrity of the data. The output is either the plaintext or an indication that the plaintext is not authentic.
authenticity	The property that data originated from its purported source. In the context of a key-wrap algorithm, the source of authentic data is an entity with access to an implementation of the authenticated-encryption function with the KEK.
bit	A binary digit: `0` or `1`.
bit string	A finite, ordered sequence of bits.
block	For a given block cipher, a bit string whose length is the block size of the block cipher.
block cipher	A parameterized family of permutations on bit strings of a fixed length; the parameter that determines the permutation is a bit string called the key.
block cipher mode of operation	An algorithm for the cryptographic transformation of data that is based on a block cipher.

block size	For a given block cipher and key, the fixed length of the input (or output) bit strings.
ciphertext	The confidential form of the plaintext that is the output of the authenticated-encryption function.
collision	In a given context, the equality of two values, usually out of a large number of possible values.
designated cipher function	As part of the choice of the underlying block cipher with a KEK, either the forward transformation or the inverse transformation.
exclusive-OR	The bitwise addition, modulo 2, of two bit strings of equal length.
forward transformation	The permutation of blocks that is determined by the choice of a block cipher and a key.
integrity check value	A fixed string that is prepended to the plaintext within the authenticated-encryption function of a key-wrap algorithm, in order to enable the verification of the integrity of the plaintext within the authenticated-decryption function.
inverse transformation	The inverse of the permutation of blocks that is determined by the choice of a block cipher and a key.
key-encryption key	The key for the underlying block cipher of KW, KWP, or TKW. May be called a key-wrapping key in other documents.
key-wrap algorithm	A deterministic, symmetric-key authenticated-encryption algorithm that is intended for the protection of cryptographic keys. Consists of two functions: authenticated encryption and authenticated decryption.
least significant bit(s)	The right-most bit(s) of a bit string.
most significant bit(s)	The left-most bit(s) of a bit string.
mode	See “block cipher mode of operation.”
octet	A string of eight bits. Often referred to as a byte.
plaintext	The input to the authenticated-encryption function.

prerequisite	A required input to an algorithm that has been established prior to the invocation of the algorithm.
semiblock	Given a block cipher, a bit string whose length is half of the block size.
semiblock string	For a given block size, a string that can be represented as the concatenation of semiblocks.
shall	Is required to. Requirements apply to conforming implementations.
should	Is recommended to.
unwrapping function	The inverse of the wrapping function.
valid length	A length for a plaintext or ciphertext input that is allowed for an implementation of the authenticated-encryption function or the authenticated-decryption function.
wrapping function	The keyed, length-preserving permutation that is applied to an enlarged form of the plaintext within the authenticated-encryption function to produce the ciphertext.

4.2 Acronyms

AES	Advanced Encryption Standard.
CAVP	Cryptographic Algorithm Validation Program.
CCM	Counter with Cipher Block Chaining Mode
CMVP	Cryptographic Module Validation Program.
FIPS	Federal Information Processing Standard.
FISMA	Federal Information Security Management Act.
GCM	Galois/Counter Mode
ICV	integrity check value.
IETF	Internet Engineering Task Force.

ITL	Information Technology Lab.
KW	AES Key Wrap.
KWP	AES Key Wrap with Padding.
KEK	key-encryption key.
MAC	message authentication code.
NIST	National Institute of Standards and Technology.
RFC	Request For Comment.
TDEA	Triple Data Encryption Algorithm.
TKW	TDEA Key Wrap.

4.3 Variables

C The ciphertext.

ICV1 The 64-bit default ICV for KW: `0xA6A6A6A6A6A6A6A6`.

ICV2 The 32-bit default ICV for KWP: `0xA65959A6`.

ICV3 The 32-bit default ICV for TKW: `0xA6A6A6A6`.

P The plaintext.

4.4 Operations and Functions

$\mathtt{0}^s$ The bit string that consists of *s* consecutive '0' bits.

$\mathrm{CIPH}_K(X)$ The output of the designated cipher function of the block cipher under the key *K* applied to the block *X*.

$\mathrm{CIPH}^{-1}_K(X)$ The output of the inverse of the designated cipher function of the block cipher under the key *K* applied to the block *X*.

int(*X*) The integer for which the bit string *X* is the binary representation.

len(*X*) The bit length of the bit string *X*.

$\text{LSB}_s(X)$	The bit string consisting of the s right-most bits of the bit string X.
$\text{MSB}_s(X)$	The bit string consisting of the s left-most bits of the bit string X.
$\text{TW}(S)$	The output of the wrapping function for TKW applied to the string S.
$\text{TW}^{-1}(C)$	The output of the unwrapping function for TKW applied to the string C.
$\text{W}(S)$	The output of the wrapping function for KW and KWP applied to the bit string S.
$\text{W}^{-1}(C)$	The output of the unwrapping function for KW and KWP applied to the bit string C.
$\lceil x \rceil$	The least integer that is not less than the real number x.
$[x]_s$	The binary representation of the non-negative integer x as a string of s bits, where $x < 2^s$.
$X \oplus Y$	The bitwise exclusive-OR of bit strings X and Y whose bit lengths are equal.
$X \parallel Y$	The concatenation of bit strings X and Y.
0x	The marker for the beginning of a hexadecimal representation of a bit string.

4.5 Examples of Basic Operations and Functions on Strings

In this publication, the `new courier` font indicates the '`0`' bit, the '`1`' bit, and any hexadecimal symbols: `0`, `1`, …, `9`, `A`, `B`, `C`, `D`, `E`, `F`.

The beginning of a hexadecimal representation of a string is marked by '0x.' For example, 0x`A659` = `1010011001011001`.

Given a real number x, the ceiling function, denoted $\lceil x \rceil$, is the least integer that is not less than x. For example, $\lceil 2.1 \rceil = 3$, and $\lceil 4 \rceil = 4$.

Given a positive integer s, $\texttt{0}^s$ denotes the string that consists of s '`0`' bits. For example, $\texttt{0}^8$ = `00000000`.

The concatenation operation on bit strings is denoted ||. For example, `001` || `10111` = `00110111`.

Given bit strings of equal length, the exclusive-OR (XOR) operation, denoted ⊕, specifies the addition, modulo 2, of the bits in corresponding bit positions. For example, `10011` ⊕ `10101` = `00110`.

Given a bit string X, the bit length of X is denoted len(X). For example, len(`00010`) = 5.

Given a bit string X and a non-negative integer s such that len(X) ≥ s, the functions $LSB_s(X)$ and $MSB_s(X)$ return the s least significant (right-most) bits and the s most significant (left-most) bits, respectively, of X. For example, LSB_3(`111011010`) = `010`, and MSB_4(`111011010`) = `1110`.

Given a positive integer s and a non-negative integer x that is less than 2^s, the integer-to-string function, denoted $[x]_s$, is the binary representation of x as a string of bit length s with the least significant bit on the right. For example, for the (base 10) integer 39, the binary representation (base 2) is `100111`, so $[39]_8$ = `00100111`.

Given a (non-empty) bit string X, the string-to-integer function, denoted int(X), is the integer x such that $[x]_{len(X)} = X$. In other words, int(X) is the non-negative integer less than $2^{len(X)}$ whose binary representation is X. For example, int(`00011010`) = 26.

5 Preliminaries

5.1 The Underlying Block Cipher and Key

The transformations of the variants of KW feature a block cipher as the main component; thus, each variant is a mode of operation (mode, for short) of the block cipher. The key for the underlying block cipher is called the key encryption key (KEK), denoted K.

For any given KEK, the underlying block cipher of the mode is a permutation, i.e., an invertible transformation on bit strings of a fixed length. In this publication, the transformation is called the forward transformation, and its inverse is called the inverse transformation. The strings are called blocks, and the length of a block is called the block size. As part of the choice of the underlying block cipher with the KEK, either the forward transformation or the inverse transformation is specified as the designated cipher function, denoted $CIPH_K$. The inverse of $CIPH_K$ is denoted $CIPH^{-1}_K$.

For example, the forward and inverse transformations for the AES block cipher—called "cipher" and "inverse cipher" in [3]—are informally known as the AES encryption and AES decryption functions, respectively. If the designated cipher function for a key-wrap algorithm is chosen to be the AES decryption function, then $CIPH^{-1}_K$ will be the AES encryption function.

For KW and KWP, the underlying block cipher shall be approved, and the block size shall be 128 bits. Currently, the AES block cipher, with key lengths of 128, 192, or 256 bits, is the only block cipher that fits this profile. For TKW, the underlying block cipher is specified to be TDEA, and the block size is therefore 64 bits; the KEK for TKW may have any length for which TDEA is approved; see [8].

The length of the KEK affects the security of the algorithms against brute force search, but this length will not be explicitly indicated in the specifications. Methods for generating cryptographic keys are discussed in [9]; the goal is to select the keys uniformly at random, i.e., for each possible key to occur with equal probability.

The KEK shall be secret, i.e., disclosed only to parties that are authorized to know the protected information. Compliance with this requirement is the responsibility of the entities using, implementing, installing, or configuring applications that incorporate this Recommendation. The management of KEKs is outside the scope of this publication.

5.2 The Authenticated-Encryption and Authenticated-Decryption Functions

For a given KEK and block cipher, KW, KWP, and TKW each comprise two related functions: authenticated encryption and authenticated decryption. The authenticated-encryption function takes an input string, called the plaintext, denoted *P*, and returns a longer output string, called the ciphertext, denoted *C*. The authenticated-encryption function expands the data so that only a small fraction of all possible strings of any given length can be ciphertexts.

The authenticated-decryption function takes an input string, called the purported ciphertext, and returns either 1) an output string or 2) a special symbol, denoted FAIL. In the first case, the output string is the unique plaintext that corresponds to the purported ciphertext, so the ciphertext should be regarded as authentic; the nature of the resulting assurance is described in Appendix A.3.

For KW, the authenticated-encryption function and the authenticated-decryption function are denoted KW-AE and KW-AD; for KWP, the functions are denoted KWP-AE and KWP-AD; for TKW, the functions are denoted TKW-AE and TKW-AD.

Note that, although the KEK, *K*, is a parameter for each of these six functions, in the specifications of these functions in this Recommendation, the KEK is considered to be a prerequisite, i.e., an input that has been established prior to the invocation of the function, and is omitted from the notation. Similarly, the choice of the block cipher for the KW and the KWP functions and the designation of CIPH_K are prerequisites that are omitted from the notation.

The authenticated-encryption and authenticated-decryption functions of KW and KWP are based on a keyed transformation, called the wrapping function, denoted W, and its inverse, called the unwrapping function, denoted W^{-1}. The analogous keyed transformations for TKW are denoted TW and TW^{-1}.

Within the authenticated-encryption function, the wrapping function is applied to an enlarged plaintext string to produce the ciphertext. Each key-wrap variant enlarges the plaintext by prepending a fixed string called the integrity check value (ICV); for KWP-AE, the enlarged plaintext also includes a 32-bit encoding of the octet length of the plaintext and possibly some "zero" octets as padding.

In each key-wrap variant, the authenticated-decryption function applies the unwrapping function to the purported ciphertext and then verifies whether the output string is the result of properly enlarging a plaintext string.

A useful unit of length for describing these functions is half the block size, i.e., 64 bits for KW and KWP, and 32 bits for TKW. A bit string of this length is called a semiblock, and, for a non-

negative integer *n*, the term "*n* semiblocks" means a bit string that can be represented as the concatenation of *n* semiblocks , or, alternatively, a semiblock string of length *n*.

For each key-wrap variant, the wrapping function and the unwrapping function are invoked on three or more semiblocks, although in principle, the definition could be extended to inputs of two semiblocks. The length of the output is the same as the length of the input for both functions.

KW-AE and TKW-AE are defined on two or more semiblocks. For KWP-AE, the domain of possible inputs is extended to nonempty octet strings. The upper bounds on the lengths of the inputs to these functions are discussed in Section 5.3.1.

KW-AD and TKW-AD are defined on three or more semiblocks, and KWP-AD is defined on two or more semiblocks.

5.3 Limits on Data Length

Mandatory limits on plaintext lengths for each key-wrap variant, and the corresponding limits on ciphertext lengths, are described in Sec. 5.3.1. Additional, implementation-specific limits on the data lengths are discussed in Sec. 5.3.2.

5.3.1 Mandatory Limits

The plaintext for KWP shall be an octet string, and KWP-AE is only defined when the length of the plaintext is less than 2^{32} octets, i.e., 2^{29} semiblocks. KW can accept longer inputs; nevertheless, the plaintext for KW-AE shall be limited to fewer than 2^{54} semiblocks. The plaintext for TKW-AE shall be limited to fewer than 2^{28} semiblocks. Consequently, the ciphertext for KW-AD is 2^{54} or fewer semiblocks, and the ciphertext for TKW-AD is 2^{28} or fewer semiblocks. The motivation for these restrictions is discussed in Appendix A.4. Along with the minimum length requirements from Sec. 5.2, this information is summarized in Table 1:

Table 1: Summary of Limits on Data Length

Algorithm	Plaintext	Ciphertext	Reason for Upper Bound
KW	2 to 2^{54}-1 semiblocks	3 to 2^{54} semiblocks	requirement—see Appendix A.4
KWP	1 to 2^{32}-1 octets	2 to 2^{29} semiblocks	undefined on other lengths
TKW	2 to 2^{28}-1 semiblocks	3 to 2^{28} semiblocks	requirement—see Appendix A.4

Compliance with these requirements is the responsibility of the entities using, implementing, installing, or configuring applications that incorporate this Recommendation.

5.3.2 Implementation-Specific Limits

Implementations of authenticated-encryption/decryption are not required to accept plaintext/ciphertext inputs with every length that is described in Table 1. A length that is allowed

for a given implementation is called a valid length. The definition of a set of valid lengths, within the limits that are specified in Table 1, is a prerequisite for both the authenticated-encryption function and the authenticated-decryption function. Different sets of valid lengths may be defined for different KEKs.

Ideally, for a given choice of KEK and designated cipher function, the set of valid lengths for the authenticated-encryption function should correspond to the set of valid lengths for the authenticated-decryption function. If not, interoperability may be affected; in particular, the authenticated-decryption function might not accept a legitimate ciphertext as input on the basis of its length.

The manner in which the validity of the lengths of the inputs to the authenticated-encryption and authenticated-decryption functions is enforced is outside the scope of this Recommendation.

5.4 Limits on the Number of Invocations

There is no requirement to limit the number of invocations for KW-AE or KWP-AE, or for TKW with three-key TDEA as the underlying block cipher. There is a requirement in [8] that, in order to keep the "restricted" status for TKW with two-key TDEA as the underlying block cipher, the number of invocations of TKW-AE shall not exceed 2^{20} for a given KEK; the restricted status is described in that publication.

Considerations for limiting the number of invocations of the authenticated-decryption function are discussed in Appendix A.3.

6 Specifications of KW and KWP

6.1 W and W^{-1}

Algorithm 1 below specifies the wrapping function, W, for KW-AE (see Sec. 6.2) and KWP-AE (see Sec. 6.3), using the same KEK and designated cipher function.

Algorithm 1: W(S)

Prerequisites:
KEK, K, for an approved, 128-bit block cipher;
designated cipher function, CIPH_K.

Input:
a string, S, of n semiblocks, for some integer $n \geq 3$.

Steps:

1. Initialize variables.
    a) Let $s = 6(n-1)$.
    b) Let $S_1, S_2, \ldots, S_n$ be the semiblocks such that $S = S_1 \| S_2 \| \ldots \| S_n$.
    c) Let $A^0 = S_1$.
    d) For $i = 2, \ldots, n$: let $R_i^0 = S_i$.

2. Calculate the intermediate values. For $t = 1, \ldots, s$, update the variables as follows:
   a) $A^t = \text{MSB}_{64}(\text{CIPH}_K(A^{t-1} \| R_2^{t-1})) \oplus [t]_{64}$;
   b) For $i = 2, \ldots, n\text{-}1$: $R_i^t = R_{i+1}^{t-1}$;
   c) $R_n^t = \text{LSB}_{64}(\text{CIPH}_K(A^{t-1} \| R_2^{t-1}))$.
3. Output the results:
   a) Let $C_1 = A^s$.
   b) For $i = 2, \ldots, n$: $C_i = R_i^s$.
   c) Return $C_1 \| C_2 \| \ldots \| C_n$.

Figure 1 illustrates the wrapping function applied to four semiblocks, i.e., $W(S_1 \| S_2 \| S_3 \| S_4) = C_1 \| C_2 \| C_3 \| C_4$. Each "wire" carries a semiblock, and each of the eighteen numbered rectangles represents an invocation of the underlying block cipher with the KEK. On the left side of these rectangles, the input block's most significant 64 bits enter the top wire, while on the right side of these rectangles, the output block's most significant 64 bits exit the bottom wire; this convention reduces the number of wire crossings.

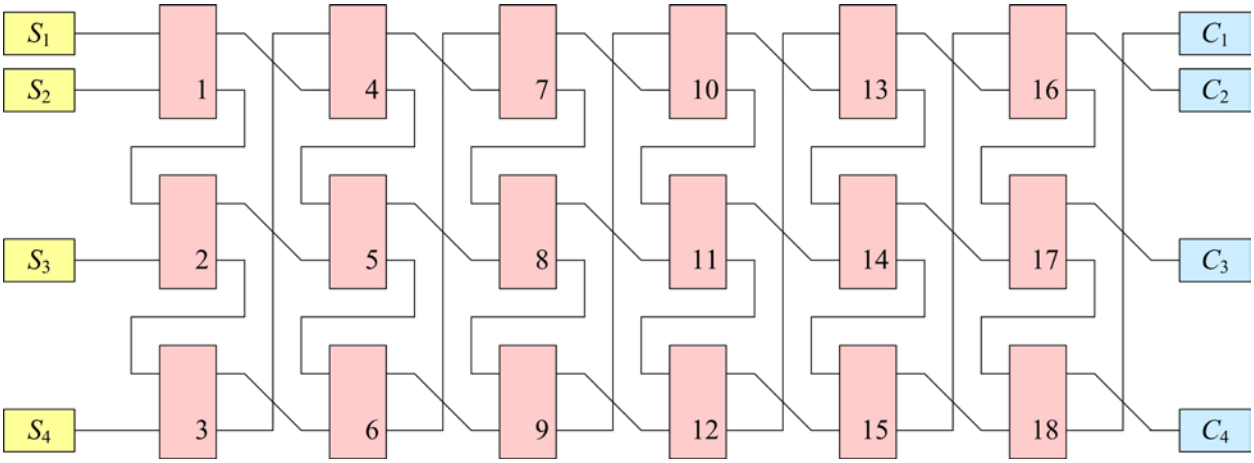

Figure 1: Illustration of the wrapping function, W

Figure 2 illustrates the assignment of intermediate values within Step 2 of Algorithm 1. The dashed lines indicate the assignments of new values to the n semiblock variables. The variable that indexes the iterations, t, increases from 1 to $6(n\text{-}1)$.

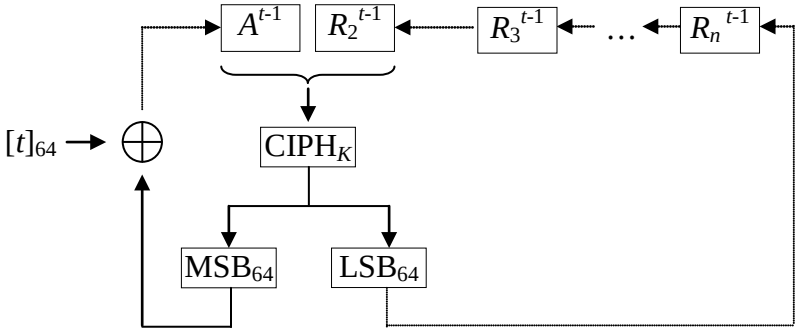

Figure 2: Illustration of an iteration within Step 2 of Algorithm 1

Algorithm 2 specifies the unwrapping function, W^{-1}, for KW-AD (see Sec. 6.2) and KWP-AD (see Sec. 6.3), with a given block cipher and KEK.

Algorithm 2: $W^{-1}(C)$

Prerequisites:
KEK, *K*, for an approved, 128-bit block cipher;
inverse of the designated cipher function, $CIPH^{-1}_K$.

Input:
a string, *C*, of *n* semiblocks, for some integer $n \geq 3$.

Steps:

1. Initialize the variables.
   a) Let $s = 6(n-1)$.
   b) Let $C_1, C_2, \ldots, C_n$ be the semiblocks such that $C = C_1 \| C_2 \| \ldots \| C_n$.
   c) Let $A^s = C_1$.
   d) For $i = 2, \ldots, n$: let $R_i^s = C_i$.
2. Calculate the intermediate values. For $t = s, s-1, \ldots, 1$, update the variables as follows:
   a) $A^{t-1} = MSB_{64}(CIPH^{-1}_K((A^t \oplus [t]_{64}) \| R_n^t))$;
   b) $R_2^{t-1} = LSB_{64}(CIPH^{-1}_K((A^t \oplus [t]_{64}) \| R_n^t))$;
   c) For $i = 2, \ldots, n-1$, $R_{i+1}^{t-1} = R_i^t$.
3. Output the results:
   a) Let $S_1 = A^0$.
   b) For $i = 2, \ldots, n$: $S_i = R_i^0$.
   c) Return $S_1 \| S_2 \| \ldots \| S_n$.

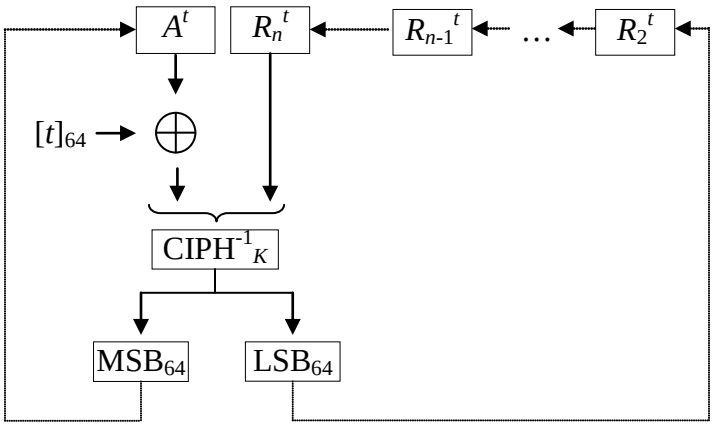

Figure 3: Illustration of an iteration within Step 2 of Algorithm 2

Figure 3 illustrates the assignment of intermediate values within Step 2 of Algorithm 2. The dashed lines indicate the assignments of new values to the *n* semiblock variables. The variable that indexes the iterations, *t*, decreases from $6(n-1)$ to 1. The input to the inverse cipher function is the concatenation of the two semiblocks that are indicated by the incoming arrows.

6.2 KW

Algorithm 3 specifies the authenticated-encryption function for KW for a given block cipher and KEK. The wrapping function, W, specified in Algorithm 1 above, is invoked in Step 3 with the same block cipher and KEK as prerequisites.

Algorithm 3: KW-AE(*P*)

Prerequisites:
KEK, *K*, for an approved, 128-bit block cipher;
designated cipher function, $CIPH_K$;
definition of valid plaintext lengths.

Input:
plaintext *P*, with valid length.

Output:
ciphertext C.

Steps:
1. Let *ICV*1 = 0xA6A6A6A6A6A6A6A6.
2. Let $S = ICV1 \,||\, P$.
3. Return $C = W(S)$.

Algorithm 4 specifies the authenticated-decryption function for KW for a given block cipher and KEK. The unwrapping function, W^{-1}, specified in Algorithm 2 above, is invoked in Step 4 with the same block cipher and KEK as prerequisites.

Algorithm 4: KW-AD(*C*)

Prerequisites:
KEK, *K*, for an approved, 128-bit block cipher;
inverse of the designated cipher function, $CIPH^{-1}_K$;
definition of valid ciphertext lengths.

Input:
purported ciphertext, *C*, with valid length.

Output:
Plaintext *P* or indication of inauthenticity, *FAIL*.

Steps:
1. Let *ICV*1 = 0xA6A6A6A6A6A6A6A6.
2. Let $S = W^{-1}(C)$.
3. If $MSB_{64}(S) \neq ICV1$, then return FAIL and stop.
4. Return $P = LSB_{64(n-1)}(S)$.

6.3 KWP

Algorithm 5 specifies the authenticated-encryption function for KWP for a given block cipher and KEK. The wrapping function, W, specified in Algorithm 1 above, is invoked in Step 5 with the same block cipher and KEK as prerequisites.

Algorithm 5: KWP-AE(*P*)

Prerequisites:
KEK, *K*, for an approved, 128-bit block cipher;
designated cipher function, CIPH_K;
definition of valid plaintext lengths.

Input:
plaintext *P*, with valid length.

Output:
ciphertext C.

Steps:
1. Let *ICV2* = 0x`A65959A6`.
2. Let *padlen* = $8 \cdot \lceil \text{len}(P)/64 \rceil - \text{len}(P)/8$.
3. Let *PAD* = $\mathtt{0}^{8padlen}$.
4. Let $S = ICV2 \,||\, [\text{len}(P)/8]_{32} \,||\, P \,||\, PAD$.
5. If $\text{len}(P) \leq 64$, then return $C = \text{CIPH}_K(S)$; if $\text{len}(P) > 64$, then return $C = \text{W}(S)$.

Algorithm 6 specifies the authenticated-decryption function for KWP for a given block cipher and KEK. The unwrapping function, W^{-1}, specified in Algorithm 2 above, is invoked in Step 4 with the same block cipher and KEK as prerequisites.

Algorithm 6: KWP-AD(*C*)

Prerequisites:
KEK, *K*, for an approved, 128-bit block cipher;
inverse of the designated cipher function, CIPH^{-1}_K;
definition of valid ciphertext lengths.

Input:
purported ciphertext, *C*, with valid length.

Output:
Plaintext *P* or indication of inauthenticity, *FAIL*.

Steps:
1. Let *n* be the number of semiblocks in *C*.
2. Let *ICV2* = 0x`A65959A6`.

3. If $n = 2$, then let $S = \text{CIPH}^{-1}_K(C)$; if $n > 2$, then let $S = \text{W}^{-1}(C)$.
4. If $\text{MSB}_{32}(S) \neq ICV2$, then return FAIL and stop.
5. Let $Plen = \text{int}(\text{LSB}_{32}(\text{MSB}_{64}(S)))$.
6. Let $padlen = 8(n\text{-}1)\text{-}Plen$.
7. If $padlen < 0$ or $padlen > 7$, then return FAIL and stop.
8. If $\text{LSB}_{8padlen}(S) \neq \mathtt{0}^{8padlen}$, then return FAIL and stop.
9. Return $P = \text{MSB}_{8Plen}(\text{LSB}_{64(n\text{-}1)}(S))$.

7 Specification of TKW

7.1 TW and TW^{-1}

Algorithm 7 specifies the wrapping function, TW, for the authenticated-encryption function of TKW (see Sec. 7.2) with a given KEK, *K*.

Algorithm 7: TW(*S*)

Prerequisites:
KEK, *K*, for TDEA;
designated cipher function, CIPH_K.

Input:
a string, *S*, of *n* semiblocks, for some integer $n \geq 3$.

Steps:
1. Initialize the variables.
    a) Let $s = 6(n\text{-}1)$.
    b) Let $S_1, S_2, \ldots, S_n$ be the semiblocks such that $S = S_1 \| S_2 \| \ldots \| S_n$.
    c) Let $A^0 = S_1$.
    d) For $i = 2, \ldots, n$, let $R_i^0 = S_i$.
2. Calculate the intermediate values. For $t = 1, \ldots, s$, update the variables as follows:
    a) $A^t = \text{MSB}_{32}(\text{CIPH}_K(A^{t-1} \| R_2^{t-1})) \oplus [t]_{32}$;
    b) For $i = 2, \ldots, n\text{-}1$: $R_i^t = R_{i+1}^{t-1}$;
    c) $R_n^t = \text{LSB}_{32}(\text{CIPH}_K(A^{t-1} \| R_2^{t-1}))$.
3. Output the results:
    a) Let $C_1 = A^s$.
    b) For $i = 2, \ldots, n$: $C_i = R_i^s$.
    c) Return $C_1 \| C_2 \| \ldots \| C_n$.

Algorithm 8 specifies the unwrapping function, TW^{-1}, for the authenticated-decryption function of TKW (see Sec. 7.2) with a given KEK, *K*.

Algorithm 8: $\text{TW}^{-1}(C)$

Prerequisites:
KEK, *K*, for TDEA;

inverse of the designated cipher function, CIPH^{-1}_K.

Input:
a semiblock string, *C*, with length, *n*, for an integer $n \geq 3$.

Steps:

1. Initialize the variables.
   a) Let $s = 6(n-1)$.
   b) Let $C_1, C_2, \ldots, C_n$ be the semiblocks such that $C = C_1 \| C_2 \| \ldots \| C_n$.
   c) Set $A^s = C_1$.
   d) For $i = 2, \ldots, n$: $R_i^s = C_i$.
2. Calculate the intermediate values. For $t = s, s-1, \ldots, 1$, update the variables as follows:
   a) $A^{t-1} = \text{MSB}_{32}(\text{CIPH}^{-1}_K((A^t \oplus [t]_{32}) \| R_n^t))$;
   b) $R_2^{t-1} = \text{LSB}_{32}(\text{CIPH}^{-1}_K((A^t \oplus [t]_{32}) \| R_n^t))$;
   c) For $i = 2, \ldots, n-1$: $R_{i+1}^{t-1} = R_i^t$.
3. Output the results:
   a) Let $S_1 = A^0$.
   b) For $i = 2, \ldots, n$: $S_i = \text{R}_i^0$.
   c) Return $S_1 \| S_2 \| \ldots \| S_n$.

7.2 TKW

Algorithm 9 specifies the authenticated-encryption function for TKW for a given TDEA key. The wrapping function, TW, specified in Algorithm 7 above, is invoked in Step 3 with the same key as a prerequisite.

Algorithm 9: TKW-AE(*P*)

Prerequisites:
KEK, *K*, for TDEA;
designated cipher function, CIPH_K;
definition of valid plaintext lengths.

Input:
plaintext *P*, with valid length.

Output:
ciphertext C.

Steps:

1. Let *ICV*3 = 0xA6A6A6A6.
2. Let $S = ICV3 \| P$.
3. Return $C = \text{TW}(S)$.

Algorithm 10 specifies the authenticated-decryption function for TKW for a given TDEA key. The unwrapping function, TW^{-1}, specified in Algorithm 8 above, is invoked in Step 4 with the same key as a prerequisite.

Algorithm 10: TKW-AD(*C*)

Prerequisites:
KEK, *K*, for TDEA;
inverse of the designated cipher function, $CIPH^{-1}_{K}$;
definition of valid ciphertext lengths.

Input:
purported ciphertext, *C*, with valid length.

Output:
plaintext *P* or indication of inauthenticity, *FAIL*.

Steps:
1. Let *n* be the number of semiblocks in *C*.
2. Let *ICV*3 = 0xA6A6A6A6.
3. Let $S = TW^{-1}(C)$.
4. If $MSB_{32}(S) \neq ICV3$, then return FAIL and stop.
5. Return $P = LSB_{32(n-1)}(S)$.

8 Conformance

An implementation may claim conformance to one or more of the following six functions: KW-AE, KW-AD, KWP-AE, KWP-AD, TKW-AE, and TKW-AD. TKW-AE and TKW-AD are approved to support legacy systems but should not be used for new applications.

The associated wrapping and unwrapping functions, W, W^{-1}, TW, and TW^{-1}, are not approved for use independently of these six functions.

Implementations of the authenticated-encryption and authenticated-decryption functions may further restrict the lengths of the plaintext and ciphertext given in Table 1, as discussed in Sec. 5.3.2, as long as at least one length is supported. Such restrictions may affect interoperability.

For every algorithm that is specified in this Recommendation, a conforming implementation may replace the given set of steps with any mathematically equivalent set of steps. In other words, different procedures that produce the correct output for any input are permitted.

Appendix A: Some Security Considerations

A.1 Equality of Plaintexts

Each key-wrap algorithm in this Recommendation is deterministic: for a given designated cipher function and KEK, any invocation of the authenticated-encryption function on a given plaintext produces the same ciphertext. It follows that any pair of ciphertexts reveals whether the corresponding plaintexts are equal.

Therefore, ideally, for a given designated cipher function and KEK, the authenticated-encryption function should be invoked only once on each plaintext. If multiple invocations are necessary on the same data for a system, then one method for ensuring that the ciphertexts are different would be to prepend the data with a fixed-length nonce before invoking the authenticated-encryption function. Upon authenticated decryption, the nonce would be discarded.

A.2 Implied Strength of Protected Keys

The disclosure of a KEK potentially compromises any data (i.e., any key) that the KEK protects. Therefore, the cryptographic strength of the protected key is limited implicitly by the resistance of the KEK to brute force search, and this resistance is limited by the length of the KEK. To maintain the expected level of assurance, the generation and management of the KEK should be at least as strong cryptographically as any key that it protects.

A.3 Authentication Assurance

The expansion of the plaintext within the authenticated-encryption function provides the mechanism whereby assurance of the authenticity of the data can be obtained when the authenticated-decryption function is invoked. The nature of this assurance depends on the output of the authenticated-decryption function:

- If the output is a plaintext, i.e., not FAIL, then the design of the mode provides strong, but not absolute, assurance of the authenticity of the data, i.e., that the ciphertext was generated by an invocation of the authenticated-encryption function on the plaintext. The authenticity implies the integrity of the ciphertext and resulting plaintext, i.e., that they were not altered, intentionally or unintentionally, after the generation of the ciphertext.

- If the output is *FAIL*, then it is certain that the ciphertext is not authentic.

In the first case, the assurance is not absolute because forgeries are possible, in principle. In other words, an adversary, without access to the key or to an implementation of the authenticated-encryption function, may be able to produce a genuine ciphertext, for example, by a lucky guess.

In particular, if the adversary chooses a string at random with a valid ciphertext length, the probability that the string will be a genuine ciphertext is exactly 1 in 2^{64} for KW, and approximately 1 in 2^{64} for KWP. The probability that a randomly chosen ciphertext will appear to be genuine for TKW is greater, 1 in 2^{32}, so TKW is significantly more vulnerable to forgeries.

For this reason, TKW is not recommended for new applications.

Given repeated attempts, of course, the adversary can increase the probability that a randomly-generated string will eventually be accepted as a valid ciphertext. The system or protocol that implements the authenticated-decryption function should monitor and, if necessary, limit the number of unsuccessful verification attempts for each key.

A.4 Forgeries of Extremely Long Messages

The motivation for the limits on the length of the plaintext in Sec. 5.3 is the following observation on the unwrapping function, due to John Kelsey of NIST: if S is extremely long, about 2^{67} semiblocks, and if T and U are semiblock strings of equal length, it is likely that

$$\mathrm{MSB}_{64d}(\mathrm{W}^{-1}(S||T)) = \mathrm{MSB}_{64d}(\mathrm{W}^{-1}(S||U)) \quad (1)$$

for some positive integer d.

Equation 1 will hold if six pairs of suitable intermediate values coincide within the two invocations of the unwrapping function in the equation. Each pair of coinciding values is called a collision.

In particular, let S be a semiblock string of length m, and let T and U be distinct semiblocks. Let $A^1, A^2, \ldots, A^{6m}$ be the semiblocks defined within Algorithm 2 for $\mathrm{W}^{-1}(S||T)$, and let $B^1, B^2, \ldots, B^{6m}$ be the corresponding semiblocks for $\mathrm{W}^{-1}(S||U)$. Let $i_1, i_2, \ldots, i_6$ be indices that satisfy the following "collision conditions":

$0 < i_1 < m$,	$A^{i1} = B^{i1}$,
$i_1+m < i_2$,	$A^{i2} = B^{i2}$,
$i_2+m < i_3$,	$A^{i3} = B^{i3}$,
$i_3+m < i_4$,	$A^{i4} = B^{i4}$,
$i_4+m < i_5$,	$A^{i5} = B^{i5}$,
$i_5+m < i_6 \leq 6m$, and	$A^{i6} = B^{i6}$.

These conditions imply that collisions will occur at many other intermediate values, e.g., $A^j = B^j$ for each index j such that $5m < j < i_6$, or $4m < j < i_5$, etc; collisions will also occur at these indices for the R values that are defined in Algorithm 2. For i_1, these colliding R values are part of the output of the unwrapping function, as described in Equation 1 with $d = i_1$.

The first semiblock of the output of the unwrapping function determines whether a purported ciphertext will pass the integrity check within the authenticated-decryption function. Therefore, if $S||T$ is a given ciphertext for KW, then the modified ciphertext $S||U$ will also pass the integrity check, provided that the collision conditions are satisfied. In other words, if six suitable collisions occur, $S||U$ will be a successful forgery. Moreover, the first i_1-1 semiblocks of the resulting plaintext will be identical to the plaintext from which $S||T$ was generated.

For KWP, $S||U$ would also be a successful forgery if the collision conditions are satisfied and if no padding octets were appended to the plaintext during the generation of $S||T$.

For TKW, an analogue of the above analysis applies, adapted to the smaller semiblock size.

The length of the ciphertext affects the probability that some set of indices will satisfy the collision conditions. One can estimate this probability for a fixed value of m by modeling some of the individual collisions as independent events that occur with probability 2^{-64}**.** In particular, beginning at index 6m, one considers the probability of individual collisions as follows:

- At index j, if $A^j \neq B^j$, or if $j < m$, one next considers the index j-1.
- At index j, if $A^j = B^j$ and $j > m$, one next considers the index j-m-1.

This procedure will consider exactly m indices; if it identifies collisions at six or more indices, then the first six collisions will satisfy the collision conditions. Conversely, if there are six indices that satisfy the collision conditions, then this procedure will identify collisions at six or more indices. Consequently, for KW, the probability, P, that the forgery attack succeeds in this model is

$$P = 1 - \sum_{i=0}^{5} \binom{m}{i} \left(\frac{1}{2^{64}}\right)^i \left(\frac{2^{64}-1}{2^{64}}\right)^{m-i} = \sum_{i=6}^{m} \binom{m}{i} \left(\frac{1}{2^{64}}\right)^i \left(\frac{2^{64}-1}{2^{64}}\right)^{m-i}.$$

If $2^{64}/m$ is sufficiently large, then only the first term is significant, so that

$$P \approx \frac{1}{720} \cdot \left(\frac{m}{2^{64}}\right)^6.$$

Consequently, if $m < 2^{54}$, as required in Sec. 5.3.1, then $P < 2^{-64}$.

For TKW, the analogous conclusion is that if $m < 2^{28}$, as required in Sec. 5.3.1, then $P < 2^{-32}$.

A.5 Additional Analysis

Within [10], Rogaway and Shrimpton provide an analysis of AESKW as specified in [2], much of which also applies to the key-wrap variants in this Recommendation. Among other criticisms, the authors emphasize the lack of a proof that the underlying structure of KW meets the goal of deterministic authenticated encryption that they formalize in the paper. Nevertheless, the authors expect that the AES Key Wrap achieves this property, possibly even in a particularly strong manner, i.e., with “beyond-birthday-phenomenon security.”

Appendix B: Related Algorithms

B.1 Technical Differences With Earlier Specifications of Key-Wrap Algorithms

Two features that are unique to this Recommendation are: 1) the maximum plaintext lengths for KW and TKW in Sec 5.3.1, and 2) the explicit option for implementation-specific subsets of valid lengths for KW, KWP, and TKW in Sec. 5.3.2. In particular, KW-AE, KWP-AE, and TKW-AE are undefined on plaintexts of invalid length, while KW-AD, KWP-AD, and TKW-AD check whether the length of the purported ciphertext is valid.

This Recommendation also differs slightly in its support of underlying block ciphers. The AES Key Wrap, both in the original "AES Key Wrap Specification" that was posted on NIST's Computer Security Resource Center web site and also in [11], is defined only for the AES block cipher; however, KW and KWP will also support any 128-bit block cipher that is approved in the future. The "Key Wrap" algorithm in [1] supports a wider choice of block ciphers, encompassing a variant that is equivalent to TKW, but also allowing TDEA with key lengths that are not supported in TKW.

Otherwise, the specification of KW in this Recommendation is equivalent to the original "AES Key Wrap Specification" and to the specification in [11], and almost equivalent to the "Key Wrap" specification in [1]. The specification in [1] supports the appending of random padding bits to plaintexts that are not semiblock strings, when the length of plaintext is fixed. This amounts to a reversible padding scheme that is different than the padding scheme defined in [4] and later adopted for KWP.

In [2], an analogue of KWP and an analogue of TKW are defined. The wrapping and unwrapping functions for these analogues are equivalent to W, W^{-1}, TW, and TW^{-1}, but the formatting of the plaintext is different, so implementations of the key-wrap algorithms in [2] cannot be compliant with this Recommendation.

B.2 Comparison of Functionality with Other Authentication Methods

The authentication assurance that KW-AD, KWP-AD, and TKW-AD each provide is described in Appendix A.3 above; that assurance is similar to the authentication assurance that other methods provide via the verification of an authentication tag, i.e., a digital signature or a message authentication code (MAC). Therefore, a digital signature or a MAC is not necessary to authenticate the data that is protected by a key-wrap algorithm.

However, other authentication methods differ from KW, KWP, and TKW in at least three properties: 1) Digital signatures provide non-repudiation. 2) Many authenticated-encryption algorithms, including the Galois/Counter Mode (GCM) [6] and the Counter with Cipher Block Chaining Mode (CCM) [5], provide an efficient means of authenticating associated data that is not confidential, such as the routing information in a networking protocol, or the description of the usage of the protected key. 3) Digital signatures on ciphertexts or MACs on ciphertexts can

be verified without decrypting the ciphertext.[2] By contrast, for KW, KWP, and TKW, there are no separate authentication tags: instead, the information that is necessary to verify the authenticity of the data is embedded in all of the ciphertext bits. Consequently, for these three algorithms, the authenticity of the data cannot be verified without invoking the authenticated-decryption function.

[2] This statement applies to the internal MAC of GCM, but not to the internal MAC of CCM, which is generated on the plaintext.

References

[1] ANSI/TIA-102.AACA-1-2002: *Project 25 – Digital Radio Over-the-Air-Rekeying (OTAR) Protocol: Addendum 1 – Key Management Security Requirements for Type 3 Block Encryption Algorithms*, Telecommunications Industry Association, November, 2002.

[2] ANS X9.102-2008, *Symmetric Key Cryptography For the Financial Services Industry—Wrapping of Keys and Associated Data*, Accredited Standards Committee X9, Inc., June, 2008.

[3] Federal Information Processing Standards (FIPS) Publication 197, *Advanced Encryption Standard*, U.S. DoC/NIST, November 26, 2001.

[4] R. Housley and M. Dworkin, *Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm*, RFC 5649, August, 2009.

[5] NIST Special Publication 800-38C: *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, May, 2004.

[6] NIST Special Publication 800-38D: *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode and GMAC*, November, 2007.

[7] NIST Special Publication 800-67, Revision 1: *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, January, 2012.

[8] NIST Special Publication 800-131A: *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January, 2011.

[9] NIST Special Publication 800-133: *Recommendation for Cryptographic Key Generation*, draft, August 2011.

[10] P. Rogaway and T. Shrimpton, *Deterministic Authenticated-Encryption: A Provable-Security Treatment of the Keywrap Problem*, EUROCRYPT 2006. LNCS vol. 4004, 373-390, Springer, 2006.

[11] J. Schaad and R. Housley, *Advanced Encryption Standard (AES) Key Wrap Algorithm*, RFC 3394, September, 2002.