IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)	
Title	**Draft D15 Annex B Security Implementation AES-CCM**	
Date Submitted	[14 January, 2003]	
Source	[Rene Struik] [Certicom Corporation] [5520 Explorer Drive, 4th floor, Mississauga, ON Canada L4W 5L1]	Voice: [905-501-6083] Fax: [905-507-4230] E-mail: [rstruik@certicom.com]
Re:	[]	
Abstract	[This document provides replacement text that supports a single AES-CCM specification across 802.11 WLANs and 802.15 WPANs, which completely conforms to the NIST submission of the AES-CCM mode]	
Purpose	[Assist sponsor ballot comment resolution for the Draft D15 for the IEEE 802.15.3 WPAN.]	
Notice	This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15.	

Editorial note:
The text below provides complete replacement text for Annex B.1, as currently specified in Draft D15 of the 802.15.3 specification, in support of a single AES-CCM specification across 802.11 WLANs and 802.15 WPANs, which completely conforms to the NIST submission of the AES-CCM mode.

Adopting this replacement settles the following sponsor ballot comments:
#332, #334, #335, #336, #337.
If we add at the end of B.1 that integers are represented as bit strings in most-significant-bit first order, we also solve sponsor ballot comment #333.

Annex B Security implementation

B.1 CCM mode

CCM {xref CCM specification} is a generic combined encryption and authentication block cipher mode. CCM is only defined for use with block ciphers with a 128-bit block size, such as AES. The CCM ideas can easily be extended to other block sizes, but this will require further definitions.

For the CCM mode there are two parameter choices to be made. The first choice is *M*, the size of the authentication field, in octets. The choice of the value for *M* involves a trade-off between message expansion and the probability that an attacker can undetectably modify a message. Valid values for *M* are the integers 4, 6, 8, 10, 12, 14, and 16. The second choice is *L*, the size of the length field, in octets. This value requires a trade-off between the maximum message size and the size of the Nonce. Different applications require different trade-offs, so *L* is a parameter. Valid values for *L* are the integers 2, 3, ..., 8 (the value *L*=1 is reserved). The parameters *L* and *M* are encoded as shown in Table {xref} B.1.

Table B.1—Parameters of CCM mode

Name	Description	Field Size	Encoding of field
M	Number of octets in authentication field	3 bits	(*M*-2)/2
L	Number of octets in length field	3 bits	*L*-1

Throughout this specification, the representation of integers as bit strings or octet strings shall be fixed. All integers shall be represented as octet strings in most-significant-octet first order.

B.1.1 Inputs

To encrypt a message using the CCM mode, the sender must provide the following information:

- An encryption key *K* suitable for the block cipher.
- A nonce *N* of 15-*L* octets. Within the scope of any encryption key *K*, the nonce value shall be unique. That is, the set of nonce values used with any given key shall not contain any duplicate values. Using the same nonce for two different messages encrypted with the same key destroys the security properties of this mode.
- The message *m*, consisting of a string of *l(m)* octets where $0 \le l(m) < 2^{8L}$. The length restriction ensures that *l(m)* can be encoded in a field of *L* octets.
- Additional authenticated data *a*, consisting of a string of *l(a)* octets where $0 \le l(a) < 2^{64}$. This additional data is authenticated but not encrypted, and is not included in the output of this mode. It can be used to authenticate plaintext headers, or contextual information that affects the interpretation of

the message. Users who do not wish to authenticate additional data can provide a string of length zero.

Table B.2—Inputs for CCM

Name	Description	Field Size	Encoding of field
K	Block cipher key	Depends on block cipher	String of octets.
N	Nonce	15-*L* octets	Not specified
m	Message to be encrypted and sent	*l(m)* octets	String of octets.
a	Additional authenticated data	*l(a)* octets	String of octets.

B.1.2 Authentication

The first step is to compute the authentication field *T*. This is done using the CBC-MAC mechanism, truncating the output to the appropriate size. We first define a sequence of blocks B_0, B_1, ... B_n and then apply CBC-MAC to these blocks.

The first block B_0 is formatted as follows:.

Table B.3—First authentication block B_0

Octet no:	0	1 ... 15-*L*	16-*L* ... 15
Con-tents:	Flags	Nonce *N*	*l(m)*

~~The value *l(m)* is encoded in most-significant-octet first order.~~

Within the first block B_0, the Flags field is formatted as follows:

Table B.4—Authentication flags octet

Bit no:	7	6	5	4	3	2	1	0
Con-tents:	Reserved	Adata	*M*			*L*		

The Reserved bit is reserved for future expansions and shall be set to zero. The Adata bit is set to zero if *l(a)*=0, and set to one if *l(a)*>0. The *L* field and the *M* field are the 3-bit representations of the integer encodings of the CCM mode parameters *L* and *M*, as specified in Table {xref} B.1.

If *l(a)*>0 (as indicated by the Adata field), then one or more blocks of authentication data are added. These blocks contain *l(a)* and *a* encoded in a reversible manner. We first construct the string that encodes *l(a)*.

If $0 < l(a) < 2^{16}-2^8$ then the length field is encoded as the two-octet encoding of $l(a)$ ~~in most-significant-octet first order.~~

If $2^{16}-2^8 \le l(a) < 2^{32}$ then the length field is encoded as six octets consisting of the right-concatenation of the octets 0xff, 0xfe, and the four-octet encoding of $l(a)$ ~~in most-significant-octet-first order.~~

If $2^{32} \le l(a) < 2^{64}$ then the length field is encoded as ten octets consisting of the right-concatenation of the octets 0xff, 0xff, and the eight-octet encoding of $l(a)$ ~~in most-significant-octet-first order.~~

This is summarized in the following table. ~~Note that all fields are interpreted in most-significant-octet first order).~~

Table B.5—Length encoding for additional authentication data

First two octets	Followed by	Comment
0x0000		Reserved
two-octet $l(a)$		For $0 < l(a) < 2^{16} - 2^8$
0xFF00 ... 0xFFFD		Reserved
0xFFFE	four-octet $l(a)$	For $2^{16} - 2^8 \le l(a) < 2^{32}$
0xFFFF	eight-octet $l(a)$	For $2^{32} \le l(a) < 2^{64}$

The blocks encoding a are formed by right-concatenating the octet string that encodes $l(a)$ with a itself, and splitting the result into 16-octet blocks, right-padding the last block with zeroes if necessary. These blocks are right-appended to the first block B_0.

After the (optional) additional authentication blocks have been added to the first block B, we right-concatenate the message blocks hereto. The message blocks are formed by splitting the message m into 16-octet blocks, right-padding the last block with zeroes if necessary. Note that if the message m is the empty string, no blocks are added in this step.

The result of these two steps is the sequence of 16-octet blocks $B_0, B_1, ..., B_n$.

The CBC-MAC is now computed as follows:

$$X_1 := E(K, B_0);$$

$$X_{i+1} := E(K, X_i \oplus B_i) \quad \text{for } i=1, ... , n,$$

where E() is the block cipher encryption function ~~and T is the MAC value~~. The 16-octet string X_{n+1} is the resulting CBC-MAC value.

The authentication tag T is the result of truncating X_{n+1} to the leftmost M octets hereof.

B.1.3 Encryption

To encrypt the message data we use the counter (CTR) mode. We first define the key stream blocks by

$$S_i := E(K, A_i) \text{ for } i=0, 1, 2, \ldots \text{, where}$$

the encryption blocks A_i are formatted as shown in Table {xref} B.6.

Table B.6—Encryption blocks A_i

Octet no:	0	1 ... 15-*L*	16-*L* ... 15
Con-tents:	Flags	Nonce *N*	Counter *i*

~~where *i* is encoded in most-significant-octet first order.~~

Within each encryption block A_i, the Flags field is formatted as shown in Table {xref} B.7.

Table B.7—Encryption flags octet

Bit no:	7	6	5	4	3	2	1	0
Con-tents:	Reserved	Reserved	0			*L*		

The nonce *N* in Table {xref} B.6 shall be formatted in the same way as the corresponding field in Table {xref} B.3.

The Reserved bits are reserved for future expansions and shall be set to zero. Bit 6 corresponds to the Adata bit in the B_0 block, but as this bit is not used here, it is reserved and shall be set to zero. The bits 3, 4, and 5 are set to 0. This ensures that all the *A* blocks are distinct from the B_0 blocks that are actually used, as those have a non-zero encoding of *M* in this position. Bits 0, 1, and 2 contain *L*, using the same encoding as in B_0.

The encrypted message is the result of XORing the octets of the message *m*, in order, with the leftmost *l(m)* octets of the right-concatenation of S_1, S_2, S_3, ... Note that S_0 is not used to encrypt the message; it is used to encrypt the authentication field *T* instead.

The encrypted authentication value *U* is the result of XORing the octets of the authentication field *T* determined in Subclause {xref} B.1.2 with the leftmost *M* octets of the key stream block S_0, i.e.,

$$U := T \oplus \text{leftmost-}M\text{-octets}(S_0)$$

B.1.4 Output

The encrypted and authenticated message *c* that is the output of the CCM mode computation is the right-concatenation of the encrypted message and the encrypted authentication value *U*.

B.1.5 Decryption

To decrypt a message the following information is required:

— The encryption key *K*.
— The nonce *N*.
— The additional authenticated data *a*.
— The encrypted and authenticated message *c*.

Decryption starts by recomputing the key stream to recover the message *m* and the authentication field *T*. The message and additional authentication data is then used to recompute the CBC-MAC value and check *T*.

If the *T* value is not correct, the receiver shall not reveal any information except for the fact that *T* is incorrect. In particular, the receiver shall not reveal the decrypted message, the value *T*, or any other information.

B.1.6 Restrictions

All implementations shall limit the total amount of data that is encrypted with a single key. The sender shall ensure that the total number of block cipher encryption operations in the CBC-MAC and encryption together shall not exceed 2^{61}. (This allows close to 2^{64} octets to be encrypted and authenticated using CCM, which should be more than enough for most applications.) Receivers that do not expect to decrypt the same message twice may also implement this limit.

The recipient shall verify the (truncated) CBC-MAC before releasing any information such as the plaintext. If the CBC-MAC verification fails, the receiver shall destroy all information, except for the fact that the CBC-MAC verification failed.

B.1.7 List of symbols

Table {xref} B.8 provides a list of the symbols used for the above specification of CCM.

Table B.8—List of symbols

Name	Description	Size	Comment
a	Additional authenticated data	*l(a)* octets	Use empty string if not desired.
A_i	Counter block to generate key stream	16 octets	Contains block counter, nonce, and flags.
B_i	Input block for CBC-MAC	16 octets	Together encode *N*, *L*, *M*, *m*, and *a* uniquely.
c	Ciphertext	*l(m)* + *M* octets	Includes the encrypted MAC
K	Block cipher key	Depends on block cipher	At least 128 bits, preferably 256 bits.
L	Number of octets in length field	3 bits	Integer values 1, 2, ..., 8, encoded in 3 bits as *L*-1.
m	Message to be encrypted and sent	*l(m)* octets	Subject to $0 \le l(m) < 2^{8L}$
M	Number of octets in authentication field	3 bits	Values 4, 6, 8, 10, 12, 14, 16. Encoded value is (*M*-2)/2

Table B.8—List of symbols

N	Nonce	15-L octets	Nonce shall never be repeated for same key.
S_i	Block of the encryption key stream	16 octets	Use S_0, S_1, S_2, ... to encrypt m and T.
T	Unencrypted authentication field	M octets	Result of CBC-MAC calculation.
U	Encrypted authentication field	M octets	Appended to the message after encryption
X_i	Intermediate value of CBC-MAC	16 octets	

Editorial note: Add the following two informative references to the draft:

R. Housley, D. Whiting, N. Ferguson, Counter with CBC-MAC (CCM), submitted to N.I.S.T., June 3, 2002. Available from http://csrc.nist.gov/encryption/modes/proposedmodes/.

J. Jonsson, On the Security of CTR + CBC-MAC, NIST Mode of Operation - Additional CCM Documentation. Available from http://csrc.nist.gov/encryption/modes/proposedmodes/.