CCMP AES
Counter CBC-MAC Protocol
Advanced Encryption Standard

ADI - AMD
ARM - DSP Group
LSI Logic ZSP
MIPS - TI

VOCAL Technologies, Ltd. software libraries include a complete range of ETSI / ITU / IEEE compliant algorithms, optimized for execution on ANSI C and leading DSP architectures (ADI, AMD-Alchemy, ARM, DSP Group, LSI Logic ZSP, MIPS and TI).

The CCMP protocol is based on AES encryption algorithm using the Counter Mode with CBC–MAC (CCM) mode of operation. The CCM mode combines Counter (CTR) mode privacy and Cipher Block Chaining Message Authentication Code (CBC-MAC) authentication. These modes have been used and studied for a long time, have well-understood cryptographic properties. They provide good security and performance in either hardware or software.

CCM is a generic authenticate-and-encrypt block cipher mode. CCM is only defined for use with 128-bit block ciphers, such as AES. For the generic CCM mode there are two parameter choices. The first choice is M, the size of the authentication field. 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 are 4, 6, 8, 10, 12, 14, and 16 octets. The second choice is L, the size of the length field. 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 of *L* range between 2 octets and 8 octets (the value L=1 is reserved). *M* Number of octets in authentication field 3 bits *(*M-2)/2; *L* Number of octets in length field 3 bits L-1.

http://www.vocal.com

CCMP AES Encapsulation:

Encapsulates a plaintext MPDU using the following steps:

- It first increments the Packet Number (PN), to obtain a fresh PN for each MPDU.
- The fields in the MAC header are used to construct the Additional Authentication Data (AAD).
- Construct CCM Nonce block (initialization vector) from the PN, A2 and the Priority of the MPDU.
- Encode the new PN and the KeyId into the 8 octet CCMP Header.
- Run CTR mode AES using the temporal key (TK), AAD, Nonce and MPDU data to form the ciphertext and MIC.
- The Encrypted MPDU is formed by concatenating the original MAC Header, the CCMP header, the Encrypted Data and the MIC.

The following figure 1 depicts the CCMP AES encapsulation process.

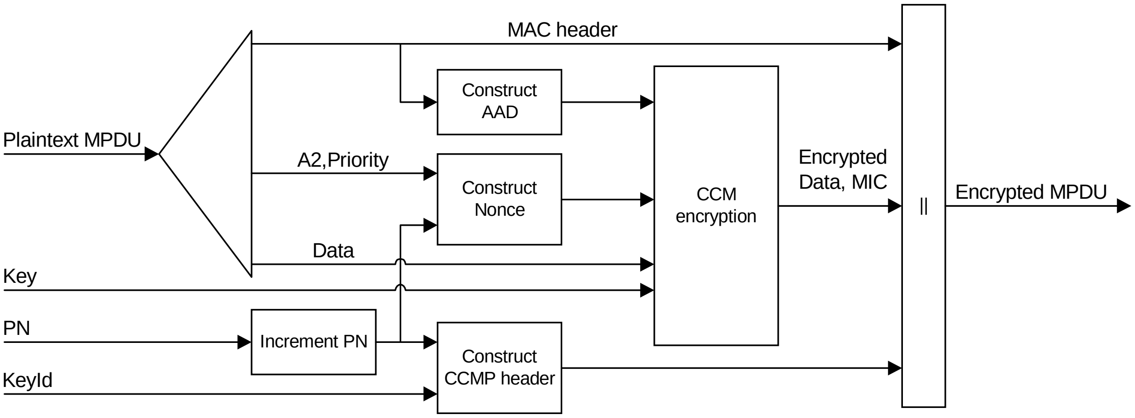


Figure 1. CCMP AES Encapsulation Process

VOCAL Technologies, Ltd.

Custom Product Design Division
200 John James Audubon Parkway
Buffalo, New York 14228
716-688-4675
http://www.vocal.com

CCMP employs the AES encryption algorithm using the CCM mode of operation. The CCM mode combines Counter Mode (CTR) for confidentiality and Cipher Block Chaining Message Authentication Code (CBC-MAC) for authentication and integrity. The Advanced Encryption Algorithm (AES) algorithm is defined in FIPS PUB 197. All processing uses AES with a 128 bit key and a 128 bit block size. CCM is a generic mode that can be used with any block oriented encryption algorithm. CCMP must use the AES algorithm with with a 128 bit key and 128 bit block size. CCM provides other parameters (K, M and L) that must have the values: K=16, M=8 and L=2. CCM requires a fresh temporal key (TK) for every session. CCM also requires a unique nonce value for each frame protected by a given TK, and CCMP uses a 48-bit packet number (PN) for this purpose. Reuse of a packet number (PN) with the same TK voids all security guarantees.

CCMP AES MIC Computation:

- CCMP uses AES in the CBC-MAC mode to compute a MIC for the MPDU.
  The input to this algorithm is:
  1. The plaintext MPDU.
  2. The Initial Block for this MPDU.
  3. The temporal key.
- The output of the algorithm is a MIC value. This can be appended to the MPDU on transmit, and compared with a received MIC at the receiver.
- The algorithm first encrypts the Initial Block to produce the CBC mode Initialization vector (IV). Next it computes the CBC-MAC over the IEEE 802.11 header length (Hlen), selected parts of the IEEE 802.11 MPDU header, and the plaintext MPDU data.

The following figure 2 depicts the MIC calculation process.

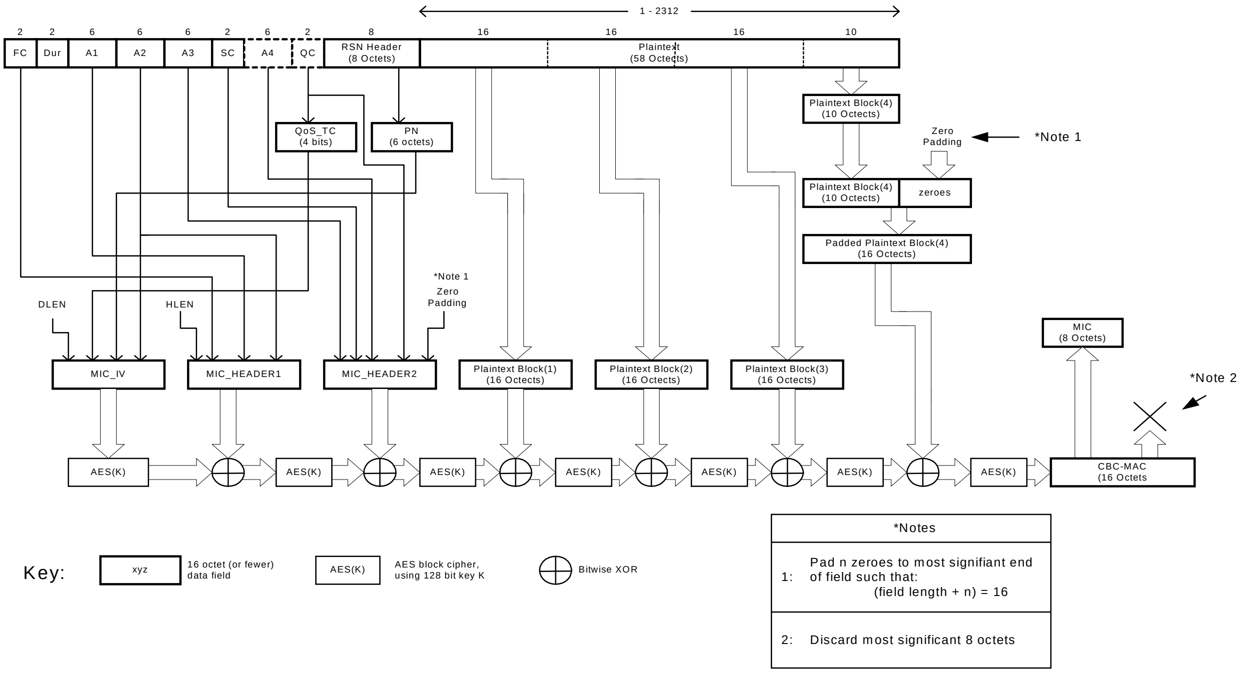
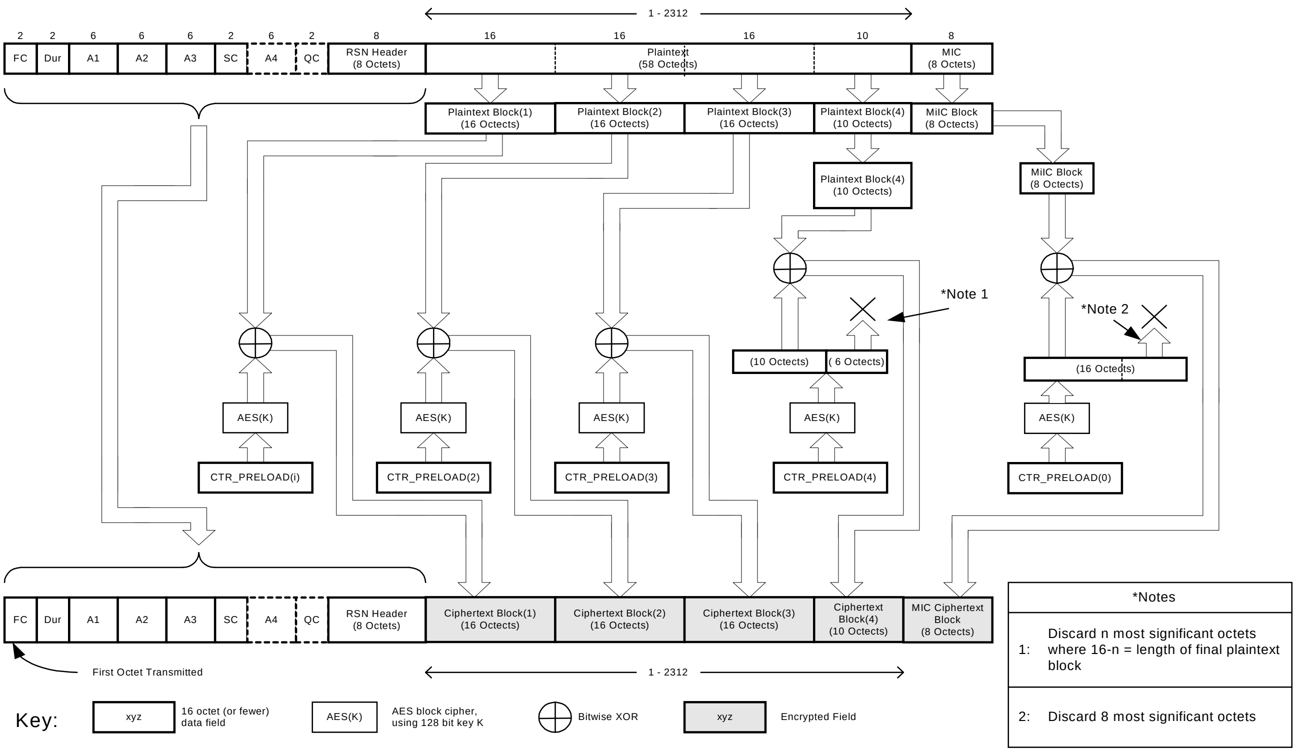


Figure 2. MIC calculation

CCMP CTR-mode encryption:

- CCMP uses AES in Counter Mode to encrypt and decrypt the MPDU data and MIC. The input to this algorithm is:
  1. The MPDU data field, with MIC appended. On transmission, the data field with MIC is plaintext, while on reception bother are ciphertext.
  2. The Counter for this MPDU
  3. The temporal key.
- The CTR Preload contains one flag byte, one byte of QoS information, a six bytes address field, a six byte packet number and a two byte counter.
- The output of the algorithm is an encrypted MPDU data field on transmit and a decrypted MPDU data field with MIC on reception.

The following figure 3 depicts the encryption process.



1 - 2312
2 2 6 6 6 2 6 2 8 16 16 16 10 8
FC
Dur
A1
A2
A3
SC
A4
QC
RSN Header (8 Octets)
Plaintext (58 Octets)
MIC (8 Octets)
Plaintext Block(1) (16 Octects)
Plaintext Block(2) (16 Octects)
Plaintext Block(3) (16 Octects)
Plaintext Block(4) (10 Octects)
MilC Block (8 Octects)
Plaintext Block(4) (10 Octects)
MilC Block (8 Octects)
*Note 1
*Note 2
(10 Octects)
(6 Octects)
(16 Octects)
AES(K)
AES(K)
AES(K)
AES(K)
AES(K)
CTR_PRELOAD(i)
CTR_PRELOAD(2)
CTR_PRELOAD(3)
CTR_PRELOAD(4)
CTR_PRELOAD(0)
FC
Dur
A1
A2
A3
SC
A4
QC
RSN Header (8 Octets)
Ciphertext Block(1) (16 Octects)
Ciphertext Block(2) (16 Octects)
Ciphertext Block(3) (16 Octects)
Ciphertext Block(4) (10 Octects)
MIC Ciphertext Block (8 Octects)
First Octet Transmitted
1 - 2312
Key:
xyz
16 octet (or fewer) data field
AES(K)
AES block cipher, using 128 bit key K
Bitwise XOR
xyz
Encrypted Field
*Notes
1: Discard n most significant octets where 16-n = length of final plaintext block
2: Discard 8 most significant octets

Figure 3. CTR-Mode Encryption

CCMP AES De-capsulation Process:

- CCMP (AES-CTR and CBC-MAC) requires only AES encryption operations and not AES decryption operations. The decapsulation process succeeds when the calculated MIC matches the MIC value received in the Encrypted MPDU.
- The following figure 4 shows the CCMP decapsulation process.

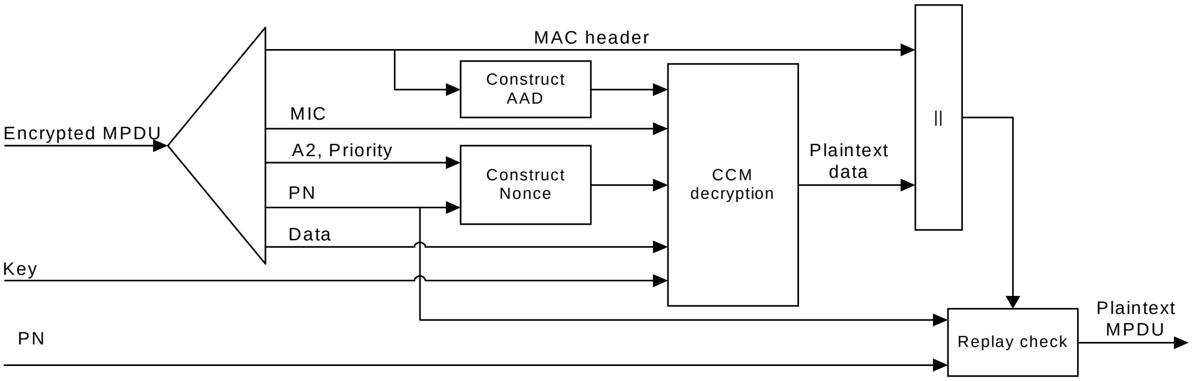


Figure 4. CCMP De-capsulation

CCMP AES Performance:

The following table 1 summarizes the number of MIPS required to encode 1 megabit of user data using CCMP with 64-bit Co-Processor:

Implementation	128-bit key	Gates
CCMP with 64-bit Co-Processor	355.46875	19843

Table 1

Performance depends on the speed of the block cipher implementation. Encrypting and authenticating the empty message, without any additional authentication data, requires two block cipher encryption operations. For each block of additional authentication data one additional block cipher encryption operation is required (if you include the length encoding). Each message block requires two block cipher encryption operations. The worst-case situation is when both the message and the additional authentication data are a single octet. In this case, CCM requires five block cipher encryption operations. In hardware, for large packets, the speed achievable for CCM is roughly the same as that achievable with the CBC encryption mode.

CCM results in the minimal possible message expansion; the only bits added are the authentication bits. Both the CCM encryption and CCM decryption operations require only the block cipher encryption function. In AES, the encryption and decryption algorithms have some significant differences. Thus, using only the encrypt operation can lead to a significant savings in code size or hardware size. In hardware, CCM can compute the message authentication code and perform encryption in a single pass. That is, the implementation does not have to complete calculation of the message authentication code before encryption can begin.