(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
H04L 9/20 (2006.01)
H04L 12/56 (2006.01)
H04L 9/18 (2006.01)

(11) 공개번호 10-2006-0091018
(43) 공개일자 2006년08월17일

(21) 출원번호 10-2005-0011731
(22) 출원일자 2005년02월12일

(71) 출원인 최준림
대구광역시 북구 산격동 1370 경북대학교

(72) 발명자 최준림
대구광역시 북구 산격동 1370 경북대학교

(74) 대리인 리앤목특허법인
이해영

심사청구 : 없음

(54) 무선 랜에서의 ＣＣＭＰ를 이용한 암호화, 복호화 장치

요약

본 발명은 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치를 개시한다.

본 발명에 의하면, 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치에 있어서, OSI 7 계층의 네트워크 계층에서 전달되는 패킷을 MAC 처리 크기로 나눈 MPDU(MAC Protocol Data Unit)으로 나누고, 나뉘어진 순서를 나타내는 패킷 번호를 인코딩하며, 패킷을 CCMP의 CBC 모드를 사용하여 MIC를 생성하고, 상위 8 바이트를 패킷의 하위에 붙이며, CCMP의 CBC의 시동 벡터(initialization vector)를 결정하는 TA(Transmitter MAC Address), DLEN (Data Length)과 PN (Packet Number)에 의해 결정되며, MIC가 부가된 데이터를 카운터 값이 PN 및 TA에 의해 결정되는 CCMP의 counter 모드를 사용하여 암호화하는 CCMP 암호화부 및 암호화되어 있는 패킷이 전달되면 그 패킷으로부터 PN과 DLEN을 추출하며, 추출된 PN으로부터 해당 패킷의 순서를 확인하여 순서가 맞는 패킷일 경우 CCMP counter 모드를 사용하여 암호화된 패캣을 복호화하며, 복호화된 데이터에서 실제의 평문(plaintext)와 MIC값을 분리하며, 분리된 평문(plaintext)은 CCMP의 CBC를 이용하여 MIC'를 얻고, MIC'과 MIC를 비교하여 그 값들이 서로 같으면 메시지의 무결성을 입증한 것으로 보고 상위 계층에 전달하는 CCMP 복호화부를 포함하여, 무선랜 환경에서 보안의 취약성을 보완하여 안전한 네트워크 환경을 제공할 수 있다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 본 발명에 따른 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치의 구성을 블록으로 도시한 것이다.

도 2는 TKIP의 암호화 과정을 도시한 것이다.

도 3은 무선랜 환경에서 CCMP를 적용하지 않은 경우와 적용한 경우의 패킷 전송 과정을 도시한 것이다.

도 4는 AES 암호화 과정을 도시한 것이다.

도 5는 AES 복호화 과정을 도시한 것이다.

도 6은 키 길이가 128비트일 때의 AES 키 생성을 위한 알고리즘을 도시한 것이다.

도 7 CBC(Cipher Block Chaining) 모드를 도시한 것으로, 각 암·복호 블럭이 서로 연결되는 구조를 도시한 것이다.

도 8은 CTR(counter) 모드를 도시한 것이다.

도 9는 CCMP의 encapsulation 연산 순서를 도시한 것이다.

도 10은 CCMP의 decapsulation 연산 순서를 나타내는 도면이다.

도 11은 CCMP 암호화(encapsulation) 과정의 흐름을 도시한 것이다.

도 12는 CCMP 복호화(decapsulation) 과정의 흐름을 도시한 것이다.

도 13은 CBC 모드를 이용한 MIC 생성과정을 도시한 것이다.

도 14는 counter 모드를 이용한 암호화(encapsulation) 과정을 도시한 것이다.

도 15는 counter 모드를 이용한 복호화(decapsulation) 과정을 도시한 것이다.

도 16은 Counter 모드에서 사용되는 16바이트 counter의 data format을 도시한 것이다.

도 17은 본 발명에 따라 IEEE 802.11i용 CCMP의 하드웨어 소프트웨어 모듈 설계의 구성을 도시한 것이다.

도 18은 본 발명의 전체적인 구성의 일 예를 도시한 것이다.

도 19는 본 발명에 따라 구성된 칩의 내부 구조를 도시한 것이다.

도 20은 본 발명에 따라 어드레스 모드, 데이터전송 모드, 읽기 대기 모드, 에러 모드간의 변이를 나타내는 상태도를 도시한 것이다.

도 21은 본 발명에 따른 CCMP의 전체 구조이다.

도 22는 본 발명에 따른 key scheduler 모듈을 나타낸다.

도 23은 본 발명에 따른 CBC 및 Counter 동작 모드 연산을 위한 블럭 구조를 도시한 것이다.

도 24는 본 발명에 따른 CCMP의 CBC 동작 모드를 하드웨어와 소프트웨어로 설계한 부분을 도시한 것이다.

도 25는 본 발명에 따른 CCMP의 Counter 동작 모드를 하드웨어와 소프트웨어로 설계한 부분을 도시한 것이다.

도 26은 본 발명에 따라 CCMP의 연산과정을 도시한 것이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 통신에 관한 것으로서, 무선랜 환경에서의 암호화를 통하여 보안을 제공할 수 있는 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치에 관한 것이다.

공항이나 대학, 일반기업 등 엔터프라이즈 환경을 기반으로 제한된 공간에서 제공되던 무선랜은 한국통신의 네스팟, 하나로통신의 AnyWay와 같은 공중망의 보급과 일반 소호(SOHO) 환경에의 적용 등으로 인하여 그 보급량이 급속도로 증가하였다. 그러나 성장에 따른 보안 문제가 심각히 대두되었는데, 현재 무선랜에서는 유선 동등 프라이버시(WEP : Wired Equivalent Privacy)라 불리는 선택적 암호화 기법이 사용되고 있다. WEP 보안 알고리즘은 취약한 보안성, 동적인 키 분배 방법이 없는 구조, 취약한 무결성 알고리즘의 사용 및 무선랜의 브로드 캐스팅 방식의 특성으로 인한 도청 등 취약성이 알려졌다.

이러한 보안 취약성을 보완하기 위해서 IEEE 802.11 task group I 에서 무선랜 보안을 증대시키기 위한 논의가 시작되어 표준화 작업을 진행해왔으며, 2004년 6월에 이르러 표준화 작업이 완료되었다.

IEEE 802.11i 표준에서는 기존 WEP이 가진 데이터 은닉성의 단점을 보완하기 위하여 장기적인 관점에서 보안 알고리즘 자체를 보안 강도가 높은 AES(Advanced Encryption Standard) 알고리즘으로 바꾸는 방식을 제안하고 있다. IEEE 802.11i에서는 기존 WEP 알고리즘을 소프트웨어적으로 보완한 TKIP(Temporal Key Integrity Protocol)와 AES 기반의 CCMP(Counter with CBC-MAC Potocol)를 data privacy mechanism으로 제공한다. CCMP는 AES를 기반으로 하는 모드 방식으로 표준에서는 기본으로 제공하도록 하고 있다.

그런데 아직 무선랜 방식에서 이와 같은 방식을 지원할 수 있는 CCMP 보안 회로 혹은 보안 장치는 없다는 문제가 있다.

발명이 이루고자 하는 기술적 과제

본 발명이 이루고자 하는 기술적인 과제는, 상기의 문제점들을 해결하기 위해, 무선랜 환경에서의 암호화를 통하여 보안을 제공할 수 있는 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치를 제공하는데 있다.

발명의 구성 및 작용

상기 기술적 과제를 해결하기 위한 본 발명에 의한, 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치는, OSI 7 계층의 네트워크 계층에서 전달되는 패킷을 MAC 처리 크기로 나눈 MPDU(MAC Protocol Data Unit)으로 나누고, 나뉘어진 순서를 나타내는 패킷 번호를 인코딩하며, 상기 패킷을 CCMP의 CBC 모드를 사용하여 MIC를 생성하고, 상위 8 바이트를 상기 패킷의 하위에 붙이며, CCMP의 CBC의 시동 벡터(initialization vector)를 결정하는 TA(Transmitter MAC Address), DLEN (Data Length)과 PN(Packet Number)에 의해 결정되며, MIC가 부가된 데이터를 카운터 값이 상기 PN 및 TA에 의해 결정되는 CCMP의 counter 모드를 사용하여 암호화하는 CCMP 암호화부; 및 암호화되어 있는 패킷이 전달되면 그 패킷으로부터 PN과 DLEN을 추출하며, 추출된 PN으로부터 해당 패킷의 순서를 확인하여 순서가 맞는 패킷일 경우 CCMP counter 모드를 사용하여 암호화된 패캣을 복호화하며, 복호화된 데이터에서 실제의 평문(plaintext)와 MIC값을 분리하며, 분리된 평문(plaintext)은 CCMP의 CBC를 이용하여 MIC'를 얻고, MIC'과 MIC를 비교하여 그 값들이 서로 같으면 메시지의 무결성을 입증한 것으로 보고 상위 계층에 전달하는 CCMP 복호화부;를 포함하는 것을 특징으로 한다.

도 1은 본 발명에 따른 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치의 구성을 블록으로 도시한 것이다.

이 장치는 패킷 데이터를 수신하여 CCMP암호화부(100)에서 암호화를 한 후, 이를 CCMP복호화부(110)에서 수신하여 복호화를 한 후, 무결성에 대한 검사까지 완료하여 다른 계층으로 전달하는 기능을 수행한다.

이하에서 첨부된 도면을 참조하여 본 발명의 바람직한 일 실시예를 상세히 설명한다.

본 발명의 기반이 될 수 있는 기술들에 대해 먼저 설명한다.

먼저 무선랜의 암호 시스템에 대해서 설명한다.

IEEE 802.11i 암호시스템

IEEE 802.11i 표준에서는 기존의 WEP 기반의 보안시스템을 새롭게 제안하는 시스템(RSN, Robust Security Network)과 구별하여 정의한다. 기존 시스템은 Pre-RSN (Robust Security Network)으로 정의하며 이때 사용되는 data privacy mechanism은 WEP를 기반으로 한다. 새롭게 제시되는 RSN은 TKIP, CCMP 두 가지의 data privacy mechanism을 제공한다.

- WEP (Wired Equivalent Privacy)

현재의 802.11b 표준은 WEP을 사용하여 무선 랜 구간의 보안이 이루어진다. WEP은 대칭키 알고리즘을 사용하기 때문에 자료의 암호화와 복호화를 처리할 때 동일한 키와 알고리즘을 사용한다. 액세스포인트의 서비스를 받는 모든 단말은 40비트 크기의 암호키를 공유하고 있으며 액세스포인트는 단말을 인증하기 위해서 랜덤 챌린지를 보내면, 단말은 40비트의 암호키와 24비트의 IV(Initial Vector)를 결합하여 RC4 암호화 알고리즘에 입력시켜 pseudo random 키 스트림을 생성하고, 이를 이용해 평문을 암호화하여 전송한다.

액세스포인트는 이를 복호화 하여 단말을 인증한다.

WEP은 802.11i에서 Pre-RSN으로 구분하여 정의되며 기존 시스템과의 호환을 위하여 사용된다. WEP 방식의 보안 문제점은 WEP 키와 IV 사이즈가 작고, 모두에게 알려진 공유키를 사용하며, 암호 알고리즘(RC4)과 무결성 알고리즘(CRC-32)이 근본적으로 보안에 취약하다는 사실에 있다. 이에 IEEE 802.11i에서는 이러한 문제를 해결하는 방법으로 두 가지 접근 방식을 채택하고 있다. 하나는 장기적인 관점에서 알고리즘 자체를 보안 강도가 높은 알고리즘(CCMP-AES)으로 바꾸는 것이고, 또 다른 하나는 단기적인 관점에서 앞에서 기술한 보안상의 문제점을 소프트웨어적으로 개선하는 TKIP 방식이다.

- TKIP (Temporal Key Integrity Protocol)

TKIP는 기존의 WEP-RC4 보안의 문제점을 소프트웨어적으로 개선하여 단말과 액세스포인트에 패치하여 사용할 수 있도록 함으로써 이미 배치되어 사용중인 무선랜의 보안 문제점을 해결하자는 취지에서 개발된 보안 프로토콜이다.

도 2는 TKIP의 암호화 과정을 도시한 것이다. Key-mixing 함수는 가입자의 per-packet키를 생성하는 함수이다. 키생성 절차는 2단계로 이루어지는데 1단계에서는 TKIP Encryption key와 단말의 MAC 주소를 이용하여 Temporary TA key (TTAK)를 생성하고, 2단계에서는 128비트 TTAK와 16비트 sequence counter를 이용하여 128비트 WEP seed를 생성한다. 이렇게 생성된 키는 802.11b의 WEP 암호화를 위한 105비트 RC4키와 24비트 WEP IV로 활용된다. Michael 함수는 기존의 WEP이 가진 메시지 인증방식의 문제점을 개선하기 위하여 새롭게 제안한 메시지 인증 함수로서 64비트 TKIP MIC 키, 출발지/목적지 MAC 주소와 MSDU 메시지를 결합하여 인증 코드를 생성한다. 이렇게 생성된 코드는 key-mixing 함수에 의해서 생성된 WEP seed로 암호화하여 전송된다.

- CCMP (Counter-mode privacy / CBC-MAC Protocol)

CCMP는 메세지의 은닉성과 무결성을 위하여 두 가지 mode operation을 각각 사용하는 프로토콜이다. 메시지의 은닉성을 위하여 counter 모드 알고리즘을 사용하며, 메시지의 무결성 검증을 위하여 CBC 모드를 사용한다. CCMP에 대한 자세한 설명은 다시 언급하기로 한다.

- 무선 랜 환경에서의 IEEE 802.11i CCMP의 적용

기존의 무선 랜 환경에서 네트워크 상에서 전달되는 패킷들에 보안 알고리즘을 적용시키지 않을 경우 원치않는 사용자에게 도청되거나 변형 될 수 있다. 이에 IEEE 802.11i에서 정의하고 있는 CCMP를 사용할 경우 무선 데이터들에 대해 AES 암호화 알고리즘의 CBC 모드와 Counter 모드를 적용시킴으로서 데이터의 은닉성과 무결성을 보장할수 있게 된다. CCMP는 MAC Layer 층에서 전체 패킷에 대해 적용되고 그 결과로서 추가적인 RSN 헤더(8 Bytes), MIC (8 Bytes) 추가에 따른 패킷 확장이 일어나게 된다.

도 3은 무선랜 환경에서 CCMP를 적용하지 않은 경우와 적용한 경우의 패킷 전송 과정을 도시한 것이다.

이하는 AES 알고리즘을 설명한다.

AES 알고리즘 개요

AES(Advanced Encryption Standard)는 차세대 대칭키 블럭 암호 알고리즘의 표준으로서 128비트 평문이 사용되며, 128비트, 192비트, 256비트의 가변 키 길이를 지원하는 알고리즘이다. AES는 복호화 연산시 암호화 연산의 역 연산을 사용하는 non-feistel 구조를 갖는다.

AES 세부 연산과정

AES는 연산을 위해 입력 블럭과 키 블럭을 state 형태로 정의하며 byte 단위의 array로 표현된다. AES 연산과정은 이러한 state 형태를 바탕으로 수행되며 도 4의 AES 암호화 과정 및 도 5의 AES의 복호화 과정에 나타나 있듯이 초기 라운드에서 AddRoundKey 연산을 수행 후, SubBytes, ShiftRows, MixColumns, AddRoundKey의 4가지 연산으로 이루어진 라운드 연산을 반복적으로 수행하여 암·복호 연산을 하게된다. 최종 라운드에서는 MixColumns 연산을 제외한 최종 라운드 연산을 수행한다.

각 라운드는 다음과 같은 4가지의 변환으로 이루어진다.

1) SubBytes는 연산시 각 byte는 state의 다른 byte 값에 독립적으로 치환한다.

2) ShiftRows는 state의 값을 변경시키지 않고, state들의 위치를 교환한다.

3) Mixcolumns은 state의 값을 column에 대해 GF(2^8)에서 정의되는 행렬 곱셈을 연산한다.

4) AddRoundKey는 각 state에 대해 생성된 라운드 Key를 exclusive-or 연산을 수행한다.

키 생성

AES에서는 매 라운드 연산에서 사용되는 AddRoundKey의 exclusive-or 연산을 위하여 각 라운드 키를 생성해야 한다. 키 생성 알고리즘은 키 사이즈에 따라 다른 생성 알고리즘을 정의하고 있다. 키 생성은 입력된 초기 키 값들을 이용하여 연산을 수행하게 된다. 키 생성과정은 입력키를 S-box로 치환을 하는 SubWord(), cyclic permutation 수행하는 RotWord(), GF(2^8)에서의 상수를 exclusive-or operation를 수행하는 Rcon(i)을 거쳐 키를 생성한다.

다음은 키 길이가 128비트일 때의 AES 키 생성을 위한 의사코드이다.

Key Scheduling Algorithm

function KeyExpansion (Key[4*Nk], W[Nb*(Nr+ 1)])

Step 1. for i = 0 to Nk /* 초기화 */

W[i] = (Key[4*i], Key[4*i+ 1], Key[4*i+ 2], Key[4*i+ 3])

Step 2. for i = Nk to Nb*(Nr+ 1)

2a. temp = W[i-1]

2b. if (i%Nk == 0) then /* 나머지 0일때 */

{ temp = ByteSub(Rot(temp) ^ Rcon[i / Nk] }

2c. W[i] = W[i - Nk] ^ temp /* W[i-Nk]와 XOR */

2d. return W[i]

도 6은 키 길이가 128비트일 때의 AES 키 생성을 위한 알고리즘을 도시한 것이다. 각 블록은 32비트의 word 단위이다. 따라서 128비트 round key값은 4개의로 word로 나뉘게 되고, 이 값들은 다음 4개의 word를 결정하게 되는 구조를 갖게 된다.

다음은 CCMP 모드에 대해 설명한다.

CCMP(Counter with CBC-MAC Protocol)의 개요

CCMP는 대칭키 블럭 암호 알고리즘의 무결성과 은닉성을 동시에 보장하기 위하여 NIST(National Institute of Standards and Technology)에서 새롭게 제시한 모드 방식이다. CCMP는 현재 NIST에서 제공하고 있는 표준 모드들 중에서 CBC(Cipher Block Chaining) 모드를 이용하여 메시지의 무결성을 위한 MAC(Message Authentication Code)를 생성하게 되고, Counter 모드를 이용하여 메시지의 은닉성을 위한 암·복호 연산을 수행한다. 또한 CCMP는 암호 연산만으로 구현이 가능하다는 장점을 가지고 있다. CCMP는 128비트 블럭 연산만을 정의하고 있다.

표준 동작 모드에서의 CBC와 Counter 모드

블럭 암호알고리즘의 경우 정해진 블럭 사이즈의 암호 연산을 수행하게 된다. AES의 경우 128비트 (16바이트)를 수행하게 되는데 이는 일반적인 메시지의 경우 128비트를 여러 번 반복하여 수행해야 한다. 여러 개의 블럭 단위 연산을 연관시켜 암호화시키는 것이 동작모드(modes of operation)이다.

NIST(National Institute of Standards and Technology)는 DES를 사용할 때부터 표준 동작모드를 정해두었으며, AES의 채택 후에 "NIST Special Publication 800-38A"의 문서를 통하여 5가지의 표준 동작모드를 제안하고 있다.

5가지의 동작모드는 ECB(Electirc Codebook), CBC(Cipher Block Chaining), CFB(Cipher Feedback), OFB(Output Feedback), CTR(Counter) 이다. 이 중에서 CCMP는 데이터 은닉화를 위하여 Counter 모드를, 데이터 무결성을 위하여 CBC를 사용한다.

CBC mode

도 7 CBC(Cipher Block Chaining) 모드를 보여주는 도면으로, 각 암·복호 블럭이 서로 연결되는 구조이다. CBC mode는 IV(initialization vector)를 필요로 하며 IV는 입력 평문과 exclusive-or된 후 암호화되고, 이 결과 값이 다시 다음의 암호 연산에 IV로 사용된다. CBC mode는 암호화 시에는 병렬 처리가 불가능 하지만 복호화 시에는 병렬 처리가 가능하다. CBC mode가 암·복호 연산에 사용될 경우 암·복호 연산기가 모두 필요하지만 CCMP에서 MIC 생성에 사용될 때에는 암호 연산기만을 필요로 한다.

Couner mode

CTR(Counter) mode는 AES가 차세대 블럭 암호 알고리즘으로 채택되면서 새롭게 추가된 모드이다. 도 8은 CTR (counter) 모드를 보여주는 도면으로, 암호 연산기만 필요로 하는 구조이며 입력으로 counter를 필요로 한다. Counter 값은 매 블럭마다 정해진 규격에 의해 증가되는 값이며, 입력된 counter 값은 블럭 암호 알고리즘의 암호 연산을 거치게 된다. 이 때 생성된 pseudo random number와 해당 plain text와 exclusive-or 연산된 값이 암호화 혹은 복호화 된 값이 된다.

IEEE 802.11i에서 CCMP의 사용

IEEE 802.11i 표준에서는 CCMP를 반드시 구현해야 하는 프로토콜로 규정하고 있으며 AES 알고리즘을 기반으로 한다. 무선랜에서는 AES의 키 사이즈를 128비트로 정의한다. CCMP를 채택한 것은 특허에 관련된 장애물이 없으며 이미 오랫동안 검증된 알고리즘이라는 점 때문이다. CCMP는 마지막 블록 사이즈가 16바이트가 되지 않을 때 0으로 padding 시켜 처리한다.

CCMP 암복호 연산과정

일반적인 MSDU에 CCMP를 적용하게 되면 8 Bytes의 RSN Header와 8 Bytes의 MIC가 추가적으로 확장하게 된다. 우선 CBC 모드에서 MIC가 생성이 된 후 전체 message가 counter 모드에 의해 암호화 되게 된다. 도 9는 CCMP의 encapsulation 연산 순서를, 도 10은 CCMP의 decapsulation 연산 순서를 나타내는 도면이다.

CCMP encapsulation / decapsulation

도 11은 CCMP 암호화(encapsulation) 과정의 흐름을 도시한 것이다. 상위 layer (OSI 7 layer에서 제 3계층인 network layer)에서 내려온 packet은 MAC 처리 크기(1～2312 bytes)로 나누게 된다. 이렇게 나뉜 data는 MPDU(MAC Protocol Data Unit)라고 부르고, 나뉘어진 순서를 나타내는 packet number를 encoding 한다. 이 packet은 CBC 모드를 사용하여 MIC를 생성하게 되고 상위 8 바이트가 packet의 하위에 붙여진다. CBC는 initialization vector를 필요로 하는데 이 값은 TA (Transmitter MAC Address), DLEN (Data Length)과 PN(Packet Number)에 의해 결정된다. 이렇게 packet에 MIC가 붙은 data를 counter 모드를 사용하여 암호화하게 된다. 이 때 사용되는 counter값은 PN과 TA에 의해 결정된다.

도 12는 CCMP 복호화(decapsulation) 과정의 흐름을 도시한 것이다. 암호화(encapsulation)되어 있는 packet이 전달되면 PN과 DLEN을 추출한다. 먼저 PN을 보고 해당 packet의 순서를 확인한다. 순서가 맞는 packet일 경우 counter 모드를 사용하여 복호화 한다. 복호화된 data는 실제 plaintext와 MIC값을 분리한다. 분리된 plaintext는 CBC를 이용하여 MIC'를 얻게 되고 MIC'과 MIC를 비교하여 값이 같다면 메시지의 무결성이 입증되어 상위 layer에 전달하게 된다.

CBC 모드를 이용한 MIC 생성과정

도 13은 CBC 모드를 이용한 MIC 생성과정을 도시한 것이다.

CBC 암호화를 위해 RSN header 및 frame header 정보들과 plaintext를 이용한다. 먼저 헤더정보들을 이용하여 암호화를 수행한 후 16바이트 단위로 나뉘어진 plaintext를 CBC를 이용하여 계속 암호화시킨다. 마지막 plaintext의 경우 16바이트가 되지 않을 경우 0을 하위에 padding 시켜 16바이트를 채워 암호화 연산을 수행한다. 이 결과 출력되는 16바이트 결과 값 중 상위 8바이트 값을 CBC의 MIC값으로 채택하여 이 값을 원래의 plaintext 하위에 붙인다.

수신측에서 decapsulation 과정을 수행할 때에는 위와 동일한 순서로 CBC 암호 연산을 수행한다. 다만, 마지막 MIC값을 하위에 붙이는 것이 아니라 원래 전달된 MIC 값과의 비교를 통해서 메시지의 무결성을 검증하는데 사용하는 것이 다른 점이다. 또한 CBC의 일반적인 암호화의 경우 CBC를 위한 초기 벡터 입력이 필요하지만, 무선랜에서 CBC 모드의 경우 초기 벡터 생성은 아래와 같은 MIC_IV의 암호화 값이 사용된다. 따라서 CBC의 내부 값을 0으로 둔 상태에서 입력을 받아 암호화 한 것과 동일한 연산이므로 초기 벡터 입력을 따로 하지 않고 사용 가능하다.

Counter 모드를 이용한 암복호 연산과정

도 14는 counter 모드를 이용한 암호화(encapsulation) 과정을 도시한 것이다.

각 data는 16바이트 단위로 나뉘어 진다. 각 블럭별로 사용되는 counter 값은 하나의 MSDU에 대하여 값이 변하며 한번의 초기 counter 값의 update 이후에 내부적으로 counter 값을 1씩 증가시키면서 연산을 수행한다.

이렇게 생성된 각 블럭마다의 counter 값은 AES를 이용하여 암호화되며 이 값은 해당 data 16바이트와 exclusive-or 연산을 거쳐 최종 16바이트의 암호화 값이 결정된다. 마지막 data의 경우 16바이트가 되지 않을 때 하위에 0을 padding 하여 암호를 수행하며 유효 바이트 길이 만큼만 취하고 나머지는 버리게 된다. 또한 MIC 암호화 시에는 counter 값은 최하위 2바이트 값이 0으로 채워져 사용된다.

이러한 counter의 입력 format은 이하에서 다시 설명될 것이다.

도 15는 counter 모드를 이용한 복호화(decapsulation) 과정을 도시한 것이다. Decapsulation 과정은 encapsulation 과정과 거의 동일하다. 단, MIC의 경우 전송된 MIC를 복호화 한 값과의 비교를 통하여 data의 무결성을 검증한다.

Counter data format

도 16은 Counter 모드에서 사용되는 16바이트 counter의 data format을 도시한 것이다. 상위 첫 바이트는 flag 값으로서 항상 16진수 1을 갖는다. 두 번째 바이트는 QoS_TC 값으로서 헤더정보에서 값을 가져온다. 3번째부터 8번째 바이트는 A2 값이며 이 역시 헤더 정보로부터 가져온다. 다음 6바이트 값은 PN(Packet Number) 값이며 RSN 헤더로부터 가져온다. 단, RSN 헤더에는 PN0부터 PN5까지의 순서로 나열되어 있으나 counter에서는 이 값을 역순을 취하여 갖는다. 맨 하위 2바이트 값은 실제 카운팅에 사용되는데, 첫 16바이트 블럭 암호화 시에 사용되는 값은 1이며 그 이후로 각각 1씩 증가시켜 사용한다. 마지막 MIC의 암호 시에는 이 2바이트 값을 0으로 하여 counter 모드 암호화를 수행한다. 이러한 counter의 format으로부터 알 수 있는 사실은 하나의 MSDU가 전송될 때 PN의 변화가 생기며 이것은 counter 값의 update를 필요로 함을 알 수 있다. 단, 하위 2바이트 값은 매 16바이트마다 1씩 변하므로 내부적으로 1씩 증가시켜 사용할 수 있다.

다음은 본 발명을 실제로 구현하기 위한 하드웨어적인 구성 및 소프트웨어적인 구성과 그 동작에 대한 설명이다.

본 발명에서 제시한 IEEE 802.11i용 CCMP의 하드웨어 소프트웨어 모듈 설계의 구성은 도 17과 같다.

하드웨어 부분은 Verilog HDL로 AES 암호화기, AES 키스케쥴러, CBC 모드 코어 및 Counter 모드 코어를 설계할 수 있으며, ModelSim과 같은 프로그램을 통해 시뮬레이션 과정을 거쳐 검증을 한다.

소프트웨어 부분은 CCMP 코어를 구동하는 C code와 주변 장치(UART)를 위한 driver code 등을 ADS compiler로 컴파일하고, AXD Debugger를 이용하여 소프트웨어의 동작유무를 확인한다.

최종적으로 QuartusⅡ 툴과 같은 프로그램을 사용하여 하드웨어와 소프트웨어 모듈을 통합하여 컴파일을 한다. 컴파일이 끝난 뒤 생성되는 플래쉬 다운로드 파일인 hex 파일을 ARM922T와 40만 게이트 FPGA가 내장되어 있는 Excalibur칩에 다운로드하여 하드웨어와 소프트웨어가 통합 설계된 IEEE 802.11i CCMP의 검증을 수행한다. 이때 디버깅을 위해 Multi-ICE를 이용한다.

상기와 같이 상용의 프로그램, 소자들을 이용해서 본 발명을 구현할 수 있다. 이와 다른 방법을 통해서 본 발명을 구현할 수 있다는 것은 자명한 사실이다.

또한 본 발명에서는 AES 암호 알고리즘과 CBC 및 Counter 동작 모드를 하드웨어 IP로 구현하였고, CCMP 구현을 위해 전체적인 동작 모드의 제어 및 CBC 모드의 IV(Initial Vector) 및 Counter 모드의 초기 Counter값 생성을 위해 ARM 코어 기반의 소프트웨어 모듈로 설계한다. 도 18은 이와 같은 본 발명의 전체적인 구성의 일 예를 도시한 것이다.

상기에 언급한 본 발명의 구현을 위해 사용한 상용칩인 Excalibur Chip을 사용하여 구현한 내부 구조를 설명한다.

도 19는 본 발명에 따라 구성된 칩의 내부 구조를 도시한 것이다.

FPGA 영역에 CCMP 코어를 구현하였고, AMBA(advanced microcontroller bus architecture) 버스 전송 표준을 따르기 위해 FSM(Finite State Machine) 모듈을 설계하여 AMBA 버스와 CCMP-AES 코어간의 인터페이스를 제공하였다. FSM에는 기본적인 어드레스 모드, 데이터전송 모드, 읽기 대기 모드, 에러 모드의 4가지 동작 모드를 가지도록 설계한다.

도 20은 이와 같은 어드레스 모드, 데이터전송 모드, 읽기 대기 모드, 에러 모드간의 변이를 나타내는 상태도를 도시한 것이다.

다음은 CCMP를 구현하기 위한 설명이다.

도 21은 본 발명에 따른 CCMP의 전체 구조이다. IEEE 802.11i CCMP에서는 AES의 암호화 연산만을 필요로 하므로 전체 하드웨어는 AES의 암호화기와 라운키 생성을 위한 키 스케쥴러 , CBC 레지스터, Counter 생성기 그리고 AMBA 버스와의 인터페이스 처리 부분으로 구성된다.

IEEE 802.11i CCMP에 있어서 무선 단말과 억세스 포인트 간에 사용하는 비밀키는 하나의 무선 단말이 억세스 포인트의 beacon frame 수신 영역에 들어올 때 최초 한번만 주고 받는다. 그러므로 전체 회로의 성능 향상을 위해 AES 키 스케쥴러는 on-fly 방식이 아닌 별도의 모듈로 구현하고 AES 암호화시에 각각의 라운키를 넘겨받아 연산을 수행한다. 전체 하드웨어는 AMBA 버스를 통해 컨트롤 신호들과 데이터들을 주고받으면서 CCMP 연산을 수행한다.

AES 암호기의 세부 알고리즘 구현에 대해 설명한다.

SubBytes

SubBytes는 8비트 입력에 대하여 S-Box라는 치환 테이블에 의한 값의 변화이며 이 값은 AES 표준문서에 정의되어 있다. 이러한 치환은 각 8비트별로 256ㅅ8비트 ROM을 사용하여 구현되어지며 따라서 128비트 data를 동시에 처리하기 위해서 암호기 및 복호기에 각각 16개의 256ㅅ8비트 ROM이 사용된다.

ShiftRows

ShiftRows는 AES를 4ㅅ4의 행렬로 표현하여 이를 정해진 값만큼 바이트 단위로 shift 시키는 연산이며 이는 하드웨어적으로 data path 신호를 정해진 위치로 배선하여 처리할 수 있다.

MixColumns

MixColums 연산은 column 단위의 입력 32비트를 AES 표준 문서에 정의된 행렬 연산과의 곱셈을 하여 32비트 출력 값을 얻게 되어 있으며, 곱셈은 shift와 exclusive-or 연산으로 구현하였다.

AddRoundKey

AddRoundKey는 라운드 키가 저장된 레지스터 파일로부터 128비트 값을 읽어서 전체 128비트 데이터를 bitwise exclusive-or 연산을 수행하는 것으로 구현된다.

Key scheduler의 구현

키 스케줄러는 초기 키 입력을 이용하여 각 라운드별 키를 만드는 key scheduling 모듈과 라운드 키 값을 저장하는 레지스터로 구성된다. 도 22는 본 발명에 따른 key scheduler 모듈을 나타낸다. 128비트 키 값은 각각 4개의 32비트 값으로 나뉘어 지고 최하위 32비트 값은 ROT, S-box, Rcon 연산을 통해 값이 생성된다. ROT는 byte 단위의 cyclic permutation을 수행하는 연산이고 S-box는 바이트 단위의 1:1 치환연산이며 Rcon은 GF(2^8)에서의 상수를 exclusive-or하는 연산이다. 이렇게 연산된 최하위 32비트 값은 각각 다음 32비트 값을 결정하기 위하여 서로 exclusive-or 연산을 수행하게 되고, 이렇게 생성된 라운드 키 값은 해당 레지스터에 저장된다. 그리고 현재 라운드에 생성된 라운드 키 값은 다음 라운드 키 값 생성에 같은 방법으로 사용된다.

CBC 및 Counter 동작 모드의 구현

도 23은 본 발명에 따른 CBC 및 Counter 동작 모드 연산을 위한 블럭 구조를 도시한 것이다. 전체 구조는 data path를 결정하고 CBC 및 Counter 동작 모드 연산의 컨트롤 신호들을 생성하는 Control 블럭과 MIC 생성을 위해 CBC 모드 수행시 이전 단계의 값을 저장하고 다음 블럭에 출력하는 CBC register, Counter 모드를 이용하여 암호화 연산에 필요한 counter 값을 생성하고 AES 결과값과 exclusive-or 연산을 수행하는 CTR generator로 크게 나뉘어진다.

CCMP에서 CBC 모드는 MIC 생성을 위해 사용되어지며 매 블럭마다의 출력값을 내보내지 않고 CBC register에 해당 값을 저장하였다가 한번의 출력으로 MIC 생성을 수행한다. 또한 counter 모드를 이용한 암호화 연산 시에 매 블럭마다 해당 counter 값을 필요로 하는데 무선랜에서는 counter 16바이트 중 14바이트 값을 MSDU 전송 시 한번 update 시키고 하위 2바이트는 내부적으로 증가시켜 사용할 수 있으므로, 매 MSDU 전송시 한번의 counter 값의 update를 수행한 후 내부적으로 이 값을 생성하는 구조를 갖는다.

CCMP 연산 구조

도 24는 본 발명에 따라 CCMP의 CBC 동작 모드를 하드웨어와 소프트웨어로 설계한 부분을 도시한 것이다. CBC 모드 연산을 위한 전체적인 구성은 AES 코어 및 CBC 레지스터를 하드웨어 IP로 구현하고, 나머지 부분은 모두 소프트웨어로 연산을 하는 구조로 이루어져 있다.

CCMP의 CBC 동작을 위해서는 802.11 헤더와 IEEE 802.11i에서 새롭게 정의한 RSN 헤더로부터 해당 부분의 정보를 추출하여 IV(Initial Vector) 및 AAD(Additional Authenticated Data)를 생성하는 과정이 필요하다. 이러한 초기화 과정은 소프트웨어로 처리된 뒤 IV, AAD 및 평문(Plain text)을 하드웨어로 구성된 AES 코어로 넘겨서 CBC 동작 연산을 하게 된다. CBC 동작은 이전 블록의 AES 결과 값이 다음 블록의 IV가 되어 입력 메시지와 exclusive-or 연산을 하고 다시 이 값이 AES 연산을 거친 뒤 그 다음 블록의 IV로 입력되는 반복 구조를 가지고 있다.

그리고 마지막 블록이 128비트가 되지 않을 경우 나머지 비트 정보들을 0으로 padding 처리를 해주어야 한다.

전체 연산과정이 끝나면 마지막 AES 암호화 연산 결과값의 상위 8바이트를 CCMP를 위한 MIC(Message Integrity Code)값으로 취한다. CBC 모드가 종료되고 나면 전체 메시지와 MIC의 암호화를 위한 Counter 모드가 실행된다.

도 25는 본 발명에 따라 CCMP의 Counter 동작 모드를 하드웨어와 소프트웨어로 설계한 부분을 도시한 것이다.

전체 메시지와 CBC 모드로 생성된 MIC 값을 Counter 모드를 통하여 암호화한다. Counter 모드는 입력 메시지를 128비트 단위로 나눈뒤 각 부분을 암호화된 counter 값과 exclusive-or 연산을 취하게 된다. 이 때 사용되는 카운터 값은 IEEE 802.11i 표준에서 정의하고 있으며 802.11 헤더와 RSN 헤더로부터 생성된다. 생성된 counter 값은 각 블록별로 1씩 증가하며 암호화 연산을 진행한다. 전체 Counter 동작 모듈은 counter 생성 부분과 AES 암호화 연산 부분, exclusive-or 연산 부분으로 구성된다.

도 26은 본 발명에 따라 CCMP의 연산과정을 도시한 것이다.

최종적인 CCMP의 암호화 연산 순서는 아래와 같다.

1. AES 키 update

2. CBC 모드를 통한 MIC 값 생성을 위해 헤더 및 메시지를 16바이트 단위 로 입력 후 암호화 수행

3. CBC 모드의 최종 출력으로부터 MIC 8바이트 값을 생성

4. Counter 모드 암호화를 위해 초기 counter값을 생성

5. 메시지 값을 16바이트 단위로 나누어 counter 모드 암호화 수행

6. MIC 값의 암호화 연산. 이 때 counter 값의 하위 2바이트는 0이 된다.

CCMP 모드를 이용한 복호화는 전체를 복호화 시킨 후 MIC 생성을 하여 이전 값과의 비교를 통해 무결성을 검증하게 되므로 아래의 순서를 갖게된다.

1. AES 키 update

2. Counter 모드 복호화를 위해 초기 counter값을 입력

3. 메시지 값을 16바이트 단위로 나누어 counter 모드 복호화 수행

4. 수신된 메시지의 MIC 복호화 연산.

5. CBC를 위하여 헤더 및 메시지를 16바이트 단위로 입력 후 암호화 수행

6. CBC의 최종 출력으로부터 MIC 8바이트 값을 얻음. 이 값을 전달되어 온 MIC 값과 비교를 통해 무결성을 검증한다.

본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 본 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 상기의 설명에 포함된 예들은 본 발명에 대한 이해를 위해 도입된 것이며, 이 예들은 본

발명의 사상과 범위를 한정하지 않는다. 상기의 예들 외에도 본 발명에 따른 다양한 실시 태양이 가능하다는 것은, 본 발명이 속한 기술 분야에 통상의 지식을 가진 사람에게는 자명할 것이다. 본 발명의 범위는 전술한 설명이 아니라 청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

또한 본 발명에 따른 상기의 각 단계는 일반적인 프로그래밍 기법을 이용하여 소프트웨어적으로 또는 하드웨어적으로 다양하게 구현할 수 있다는 것은 이 분야에 통상의 기술을 가진 자라면 용이하게 알 수 있는 것이다.

그리고 본 발명의 일부 단계들은, 또한, 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, CD-RW, 자기 테이프, 플로피디스크, HDD, 광 디스크, 광자기 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드로 저장되고 실행될 수 있다.

발명의 효과

본 발명에 의하면, 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치에 있어서, OSI 7 계층의 네트워크 계층에서 전달되는 패킷을 MAC 처리 크기로 나눈 MPDU(MAC Protocol Data Unit)으로 나누고, 나뉘어진 순서를 나타내는 패킷 번호를 인코딩하며, 패킷을 CCMP의 CBC 모드를 사용하여 MIC를 생성하고, 상위 8 바이트를 패킷의 하위에 붙이며, CCMP의 CBC의 시동 벡터(initialization vector)를 결정하는 TA(Transmitter MAC Address), DLEN (Data Length)과 PN (Packet Number)에 의해 결정되며, MIC가 부가된 데이터를 카운터 값이 PN 및 TA에 의해 결정되는 CCMP의 counter 모드를 사용하여 암호화하는 CCMP 암호화부 및 암호화되어 있는 패킷이 전달되면 그 패킷으로부터 PN과 DLEN을 추출하며, 추출된 PN으로부터 해당 패킷의 순서를 확인하여 순서가 맞는 패킷일 경우 CCMP counter 모드를 사용하여 암호화된 패캣을 복호화하며, 복호화된 데이터에서 실제의 평문(plaintext)와 MIC값을 분리하며, 분리된 평문(plaintext)은 CCMP의 CBC를 이용하여 MIC'를 얻고, MIC'과 MIC를 비교하여 그 값들이 서로 같으면 메시지의 무결성을 입증한 것으로 보고 상위 계층에 전달하는 CCMP 복호화부를 포함하여, 무선랜 환경에서 보안의 취약성을 보완하여 안전한 네트워크 환경을 제공할 수 있다.

(57) 청구의 범위

청구항 1.

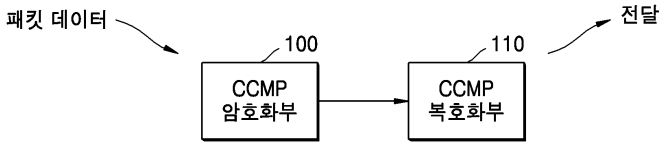
무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치에 있어서,

OSI 7 계층의 네트워크 계층에서 전달되는 패킷을 MAC 처리 크기로 나눈 MPDU(MAC Protocol Data Unit)으로 나누고, 나뉘어진 순서를 나타내는 패킷 번호를 인코딩하며, 상기 패킷을 CCMP의 CBC 모드를 사용하여 MIC를 생성하고, 상위 8 바이트를 상기 패킷의 하위에 붙이며, CCMP의 CBC의 시동 벡터(initialization vector)를 결정하는 TA(Transmitter MAC Address), DLEN (Data Length)과 PN(Packet Number)에 의해 결정되며, MIC가 부가된 데이터를 카운터 값이 상기 PN 및 TA에 의해 결정되는 CCMP의 counter 모드를 사용하여 암호화하는 CCMP 암호화부; 및

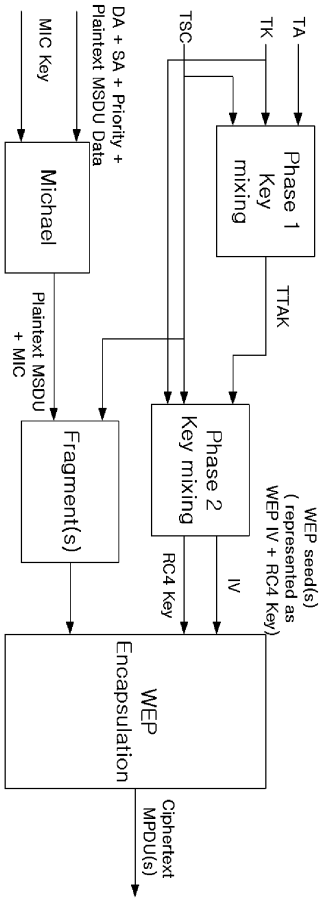
암호화되어 있는 패킷이 전달되면 그 패킷으로부터 PN과 DLEN을 추출하며, 추출된 PN으로부터 해당 패킷의 순서를 확인하여 순서가 맞는 패킷일 경우 CCMP counter 모드를 사용하여 암호화된 패캣을 복호화하며, 복호화된 데이터에서 실제의 평문(plaintext)와 MIC값을 분리하며, 분리된 평문(plaintext)은 CCMP의 CBC를 이용하여 MIC'를 얻고, MIC'과 MIC를 비교하여 그 값들이 서로 같으면 메시지의 무결성을 입증한 것으로 보고 상위 계층에 전달하는 CCMP 복호화부;를 포함하는 것을 특징으로 하는 무선 랜에서의 CCMP를 이용한 암호화, 복호화 장치.

도면

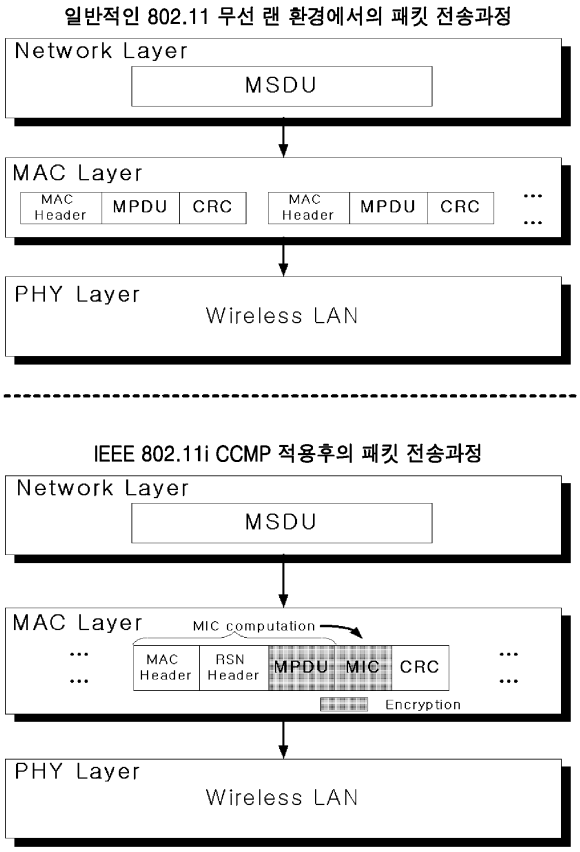
도면1


패킷 데이터
100
CCMP
암호화부
110
CCMP
복호화부
전달

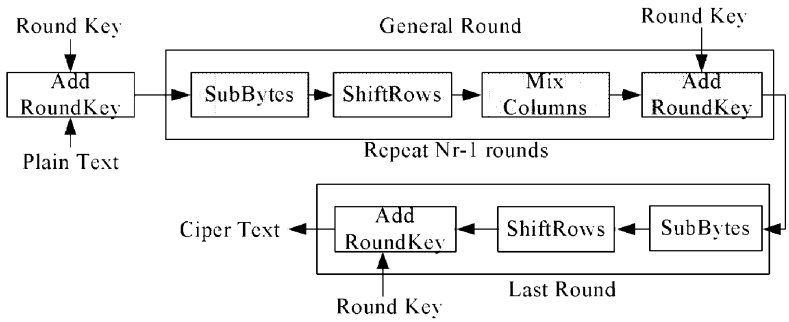
도면2


MIC Key
DA + SA + Priority +
Plaintext MSDU Data
TSC
TK
TA
Michael
Phase 1
Key
mixing
TTAK
Plaintext MSDU
+ MIC
Fragment(s)
Phase 2
Key mixing
WEP seed(s)
(represented as
WEP IV + RC4 Key)
RC4 Key
IV
WEP
Encapsulation
Ciphertext
MPDU(s)

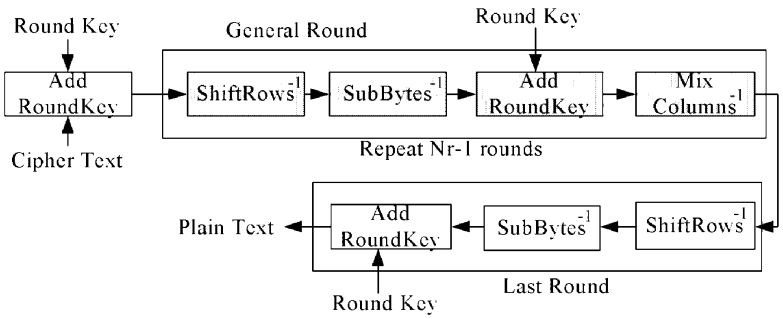
도면3


일반적인 802.11 무선 랜 환경에서의 패킷 전송과정
Network Layer
MSDU
MAC Layer
MAC Header
MPDU
CRC
MAC Header
MPDU
CRC
PHY Layer
Wireless LAN
IEEE 802.11i CCMP 적용후의 패킷 전송과정
Network Layer
MSDU
MAC Layer
MIC computation
MAC Header
RSN Header
MPDU
MIC
CRC
Encryption
PHY Layer
Wireless LAN

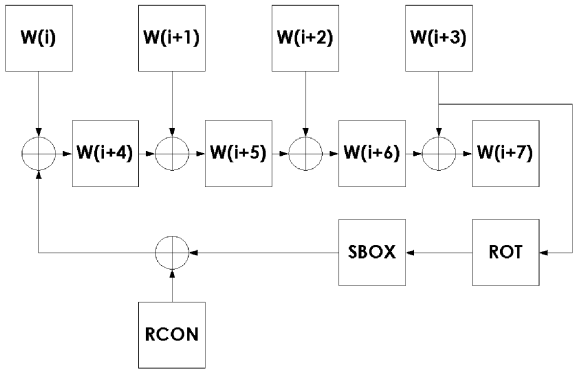
도면4


Round Key
General Round
Round Key
Add RoundKey
SubBytes
ShiftRows
Mix Columns
Add RoundKey
Plain Text
Repeat Nr-1 rounds
Ciper Text
Add RoundKey
ShiftRows
SubBytes
Last Round
Round Key

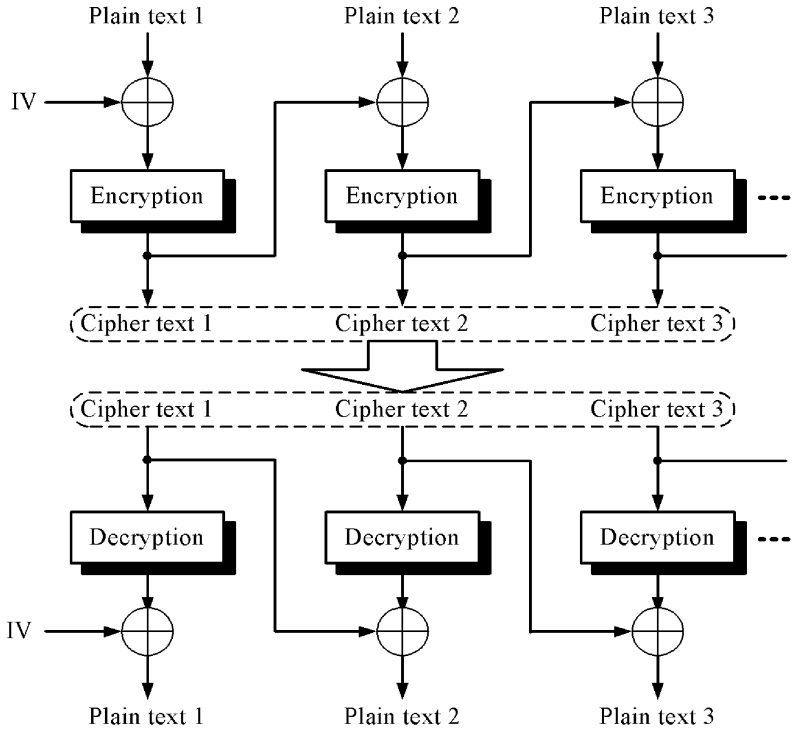
도면5


Round Key
General Round
Round Key
Add RoundKey
ShiftRows-1
SubBytes-1
Add RoundKey
Mix Columns-1
Cipher Text
Repeat Nr-1 rounds
Plain Text
Add RoundKey
SubBytes-1
ShiftRows-1
Last Round
Round Key

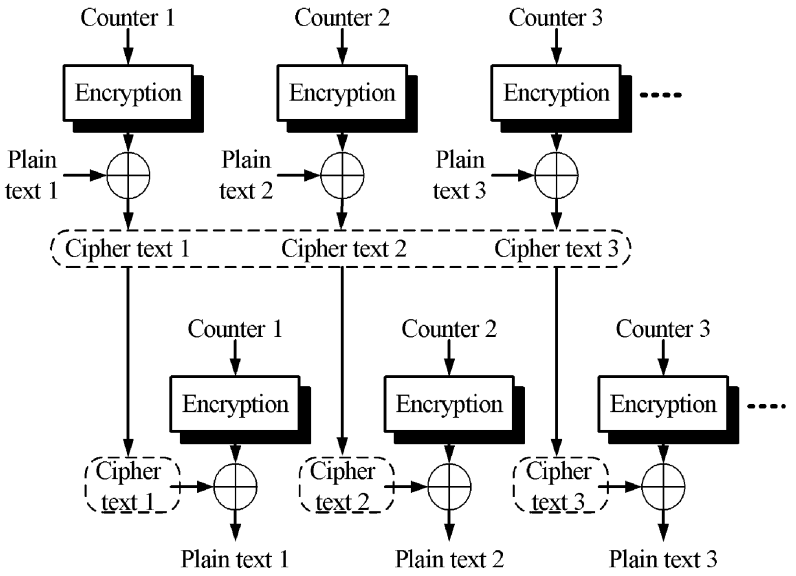
도면6


W(i)
W(i+1)
W(i+2)
W(i+3)
W(i+4)
W(i+5)
W(i+6)
W(i+7)
SBOX
ROT
RCON

도면7

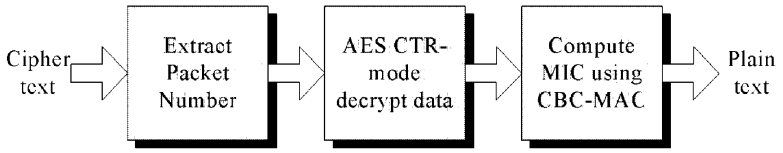

Plain text 1
Plain text 2
Plain text 3
IV
Encryption
Encryption
Encryption
Cipher text 1
Cipher text 2
Cipher text 3
Cipher text 1
Cipher text 2
Cipher text 3
Decryption
Decryption
Decryption
IV
Plain text 1
Plain text 2
Plain text 3

도면8

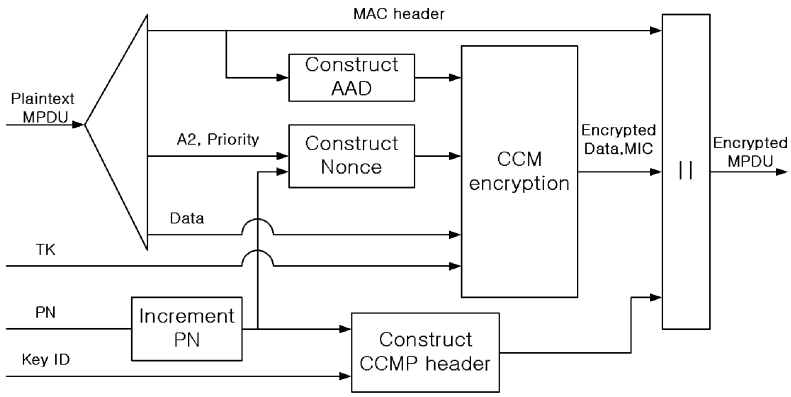


도면9

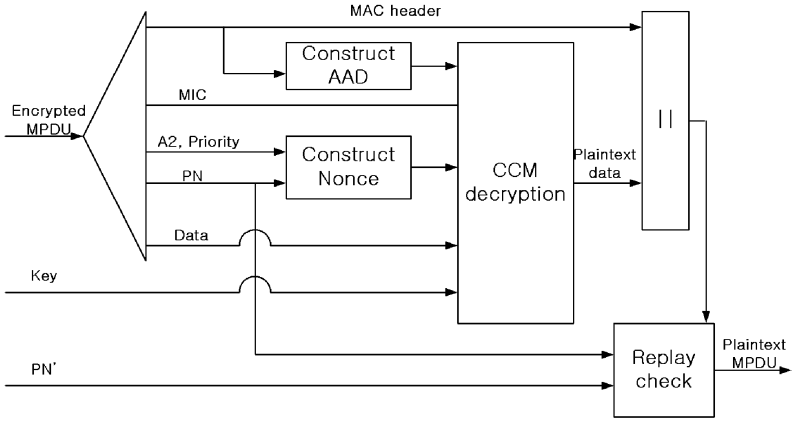
도면10



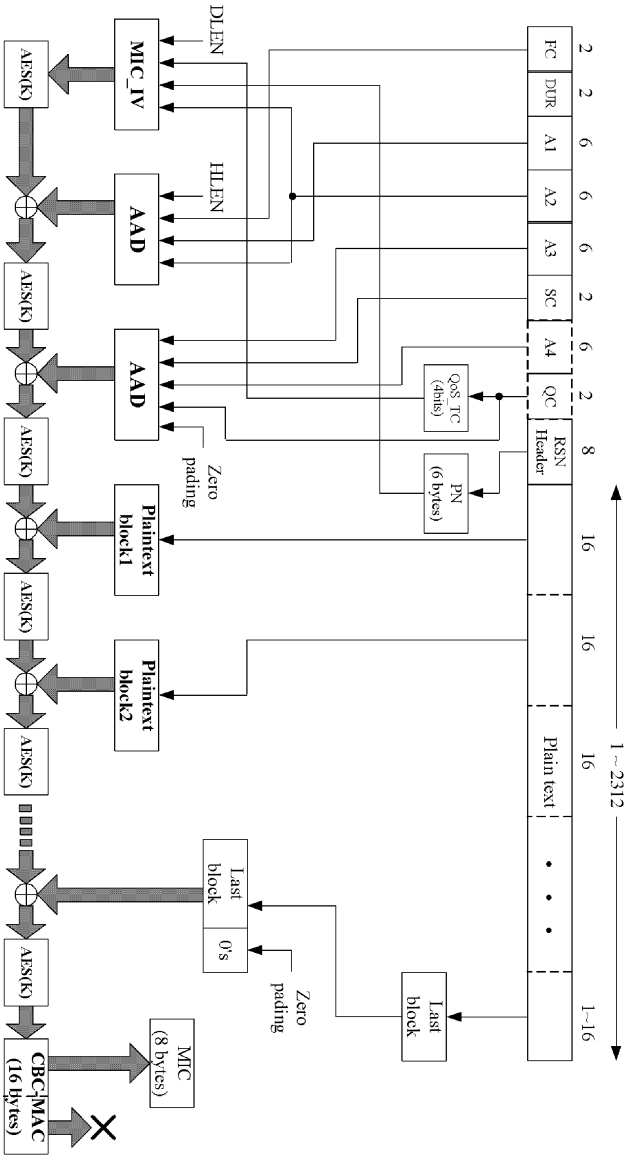
도면11



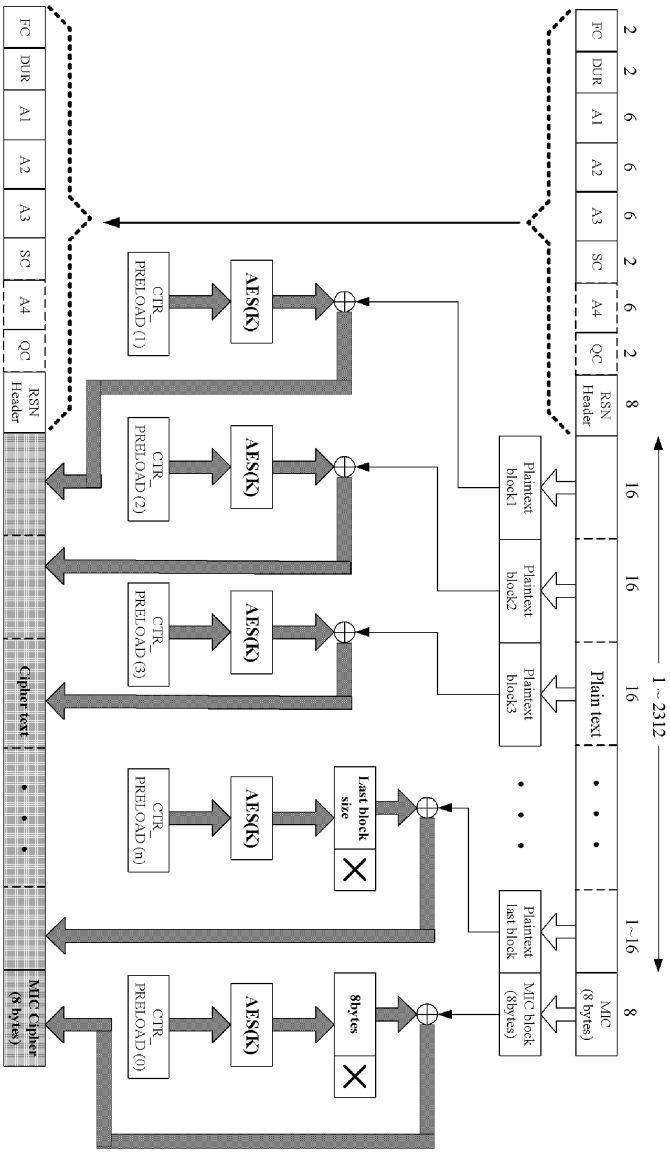
도면12


MAC header
Construct AAD
MIC
Encrypted MPDU
A2, Priority
PN
Construct Nonce
CCM decryption
Plaintext data
Data
Key
PN'
Replay check
Plaintext MPDU

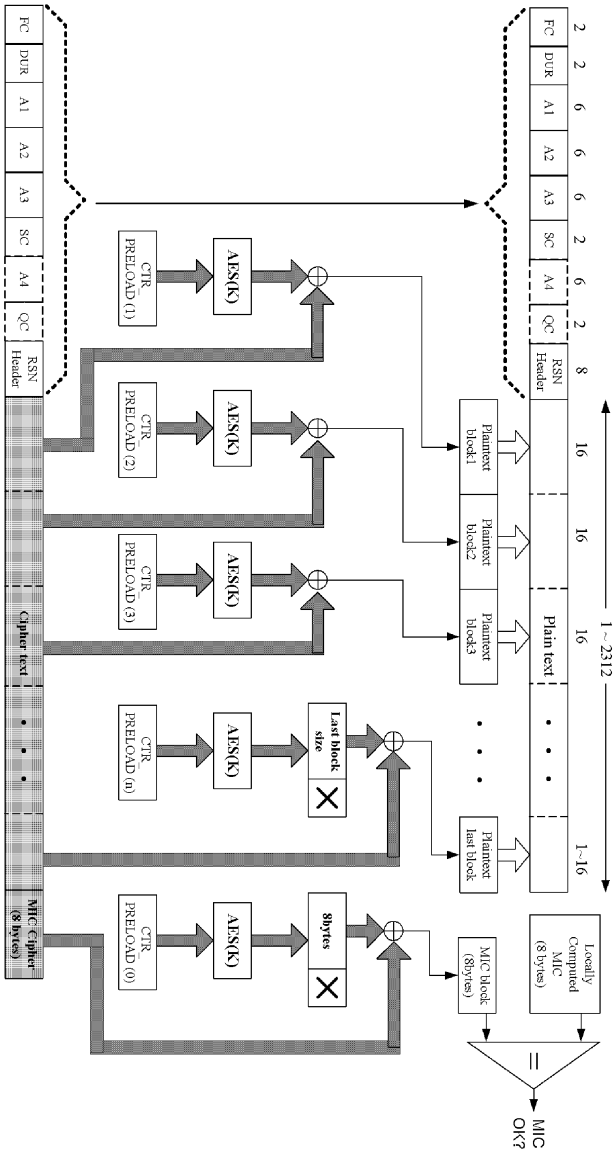
도면13


DLEN
HLEN
MIC_IV
AAD
AAD
Plaintext block1
Plaintext block2
Zero pading
AES(K)
FC
DUR
A1
A2
A3
SC
A4
QC
RSN Header
Plain text
QoS TC (4bits)
PN (6 bytes)
Last block
0's
Zero pading
MIC (8 bytes)
CBC-MAC (16 Bytes)
1~2312
1~16

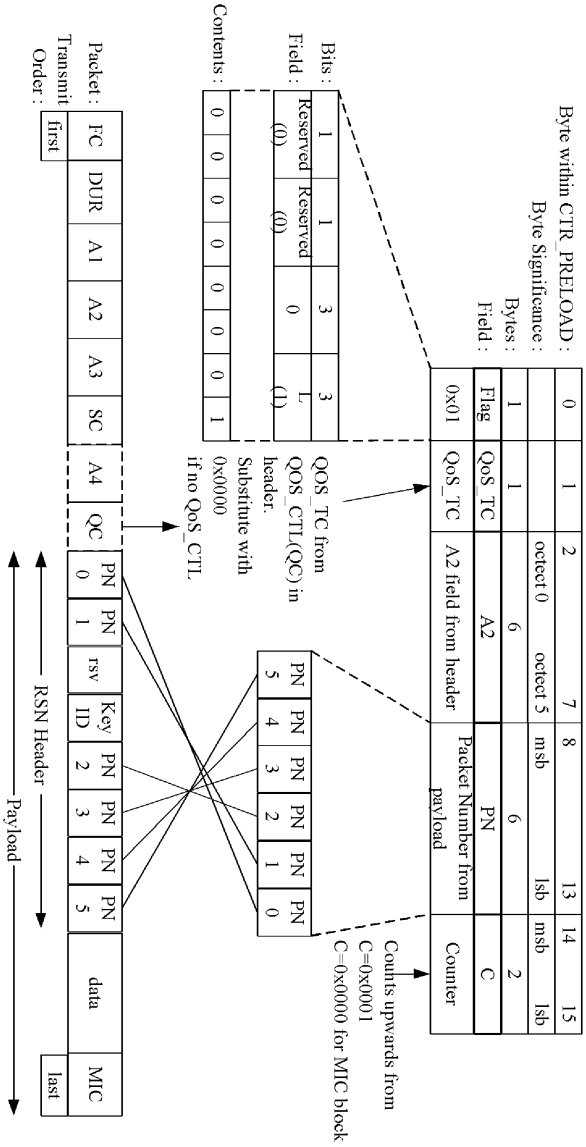
도면14


FC
DUR
A1
A2
A3
SC
A4
QC
RSN Header
2
2
6
6
6
2
6
2
8
16
16
16
1 ~ 2312
1~16
8
Plaintext block1
Plaintext block2
Plaintext block3
Plain text
Plaintext last block
MIC block (8bytes)
MIC (8 bytes)
CTR_ PRELOAD (1)
CTR_ PRELOAD (2)
CTR_ PRELOAD (3)
CTR_ PRELOAD (m)
CTR_ PRELOAD (0)
AES(K)
Last block size
8bytes
Cipher text
MIC Cipher (8 bytes)

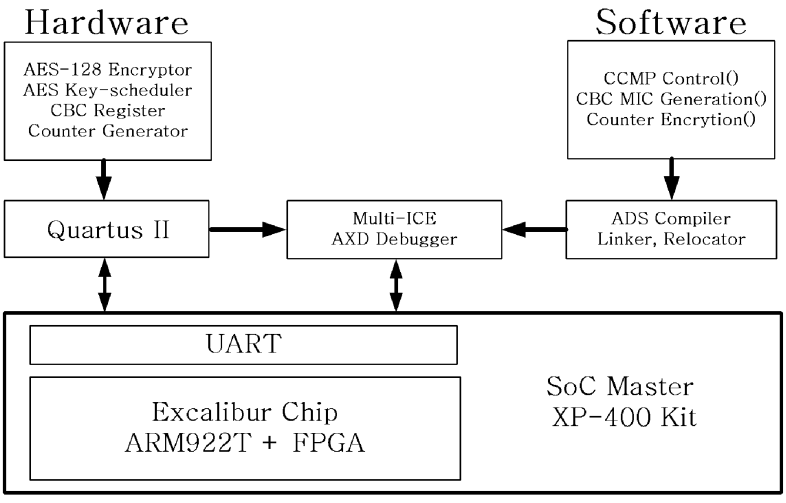
도면15


FC
DUR
A1
A2
A3
SC
A4
QC
RSN Header
Cipher text
MIC Cipher (8 bytes)
CTR PRELOAD (1)
CTR PRELOAD (2)
CTR PRELOAD (3)
CTR PRELOAD (n)
CTR PRELOAD (0)
AES(K)
Last block size
8bytes
Plaintext block1
Plaintext block2
Plaintext block3
Plaintext last block
Plain text
1~16
1 ~ 2312
Locally Computed MIC (8 bytes)
MIC block (8bytes)
=
MIC OK?

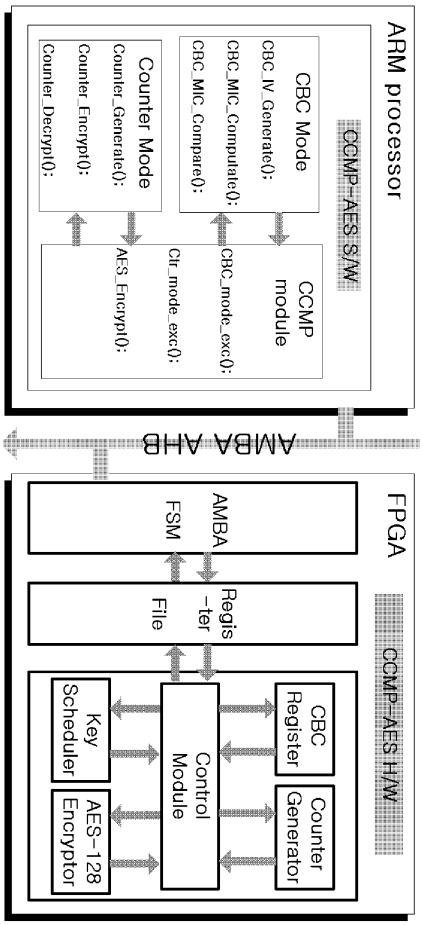
도면16


Byte within CTR_PRELOAD :
Byte Significance :
Bytes :
Field :
Flag
QoS_TC
A2 field from header
Packet Number from payload
Counter
0x01
QoS TC
A2
PN
C
octect 0
octect 5
msb
lsb
msb
lsb
Bits :
Field :
Reserved (0)
Reserved (0)
L (1)
Contents :
QOS_TC from QOS_CTL(QC) in header.
Substitute with 0x0000 if no QoS_CTL
Counts upwards from C=0x0001
C=0x0000 for MIC block
Packet :
Transmit Order :
FC
DUR
A1
A2
A3
SC
A4
QC
PN 0
PN 1
rsv
Key ID
PN 2
PN 3
PN 4
PN 5
data
MIC
first
last
RSN Header
Payload

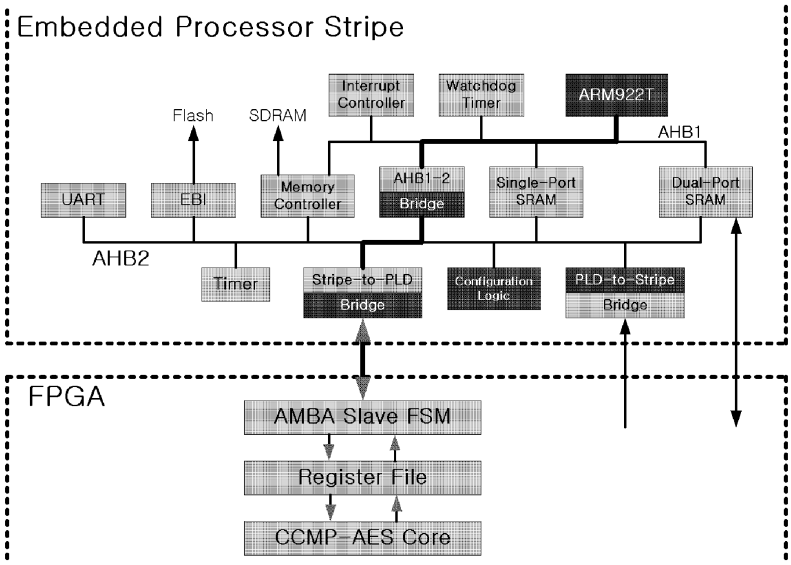
도면17


Hardware
Software
AES-128 Encryptor
AES Key-scheduler
CBC Register
Counter Generator
CCMP Control()
CBC MIC Generation()
Counter Encrytion()
Quartus II
Multi-ICE
AXD Debugger
ADS Compiler
Linker, Relocator
UART
Excalibur Chip
ARM922T + FPGA
SoC Master
XP-400 Kit

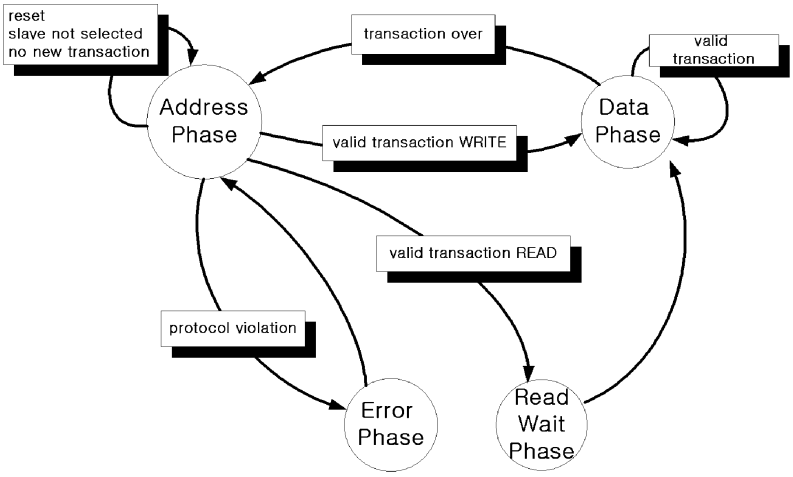
도면18


ARM processor
CCMP-AES S/W
CBC Mode
CBC_IV_Generate();
CBC_MIC_Computate();
CBC_MIC_Compare();
Counter Mode
Counter_Generate();
Counter_Encrypt();
Counter_Decrypt();
CCMP module
CBC_mode_exc();
Ctr_mode_exc();
AES_Encrypt();
AMBA AHB
FPGA
AMBA FSM
Regis-ter File
CCMP-AES H/W
Key Scheduler
AES-128 Encryptor
Control Module
CBC Register
Counter Generator

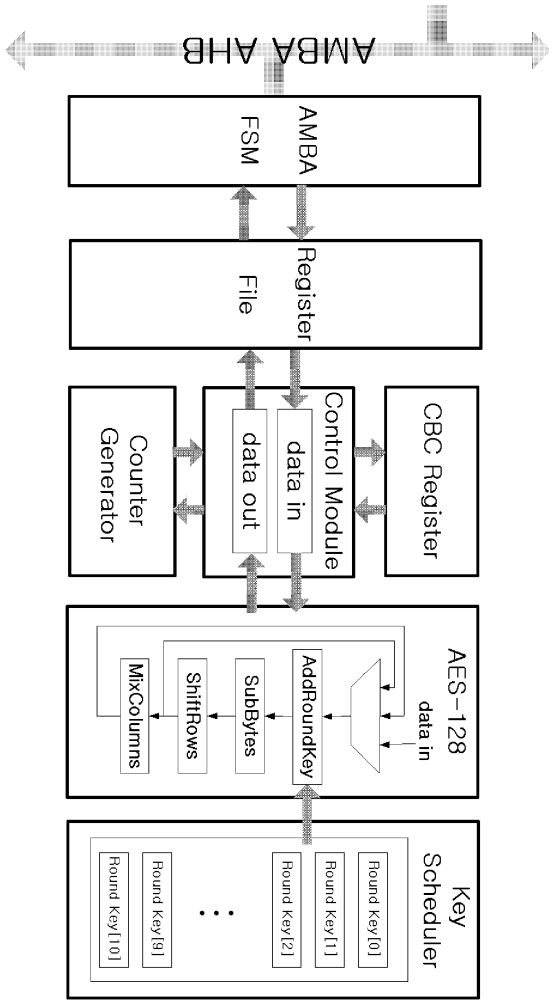
도면19


Embedded Processor Stripe
Interrupt Controller
Watchdog Timer
ARM922T
Flash
SDRAM
AHB1
UART
EBI
Memory Controller
AHB1-2 Bridge
Single-Port SRAM
Dual-Port SRAM
AHB2
Timer
Stripe-to-PLD Bridge
Configuration Logic
PLD-to-Stripe Bridge
FPGA
AMBA Slave FSM
Register File
CCMP-AES Core

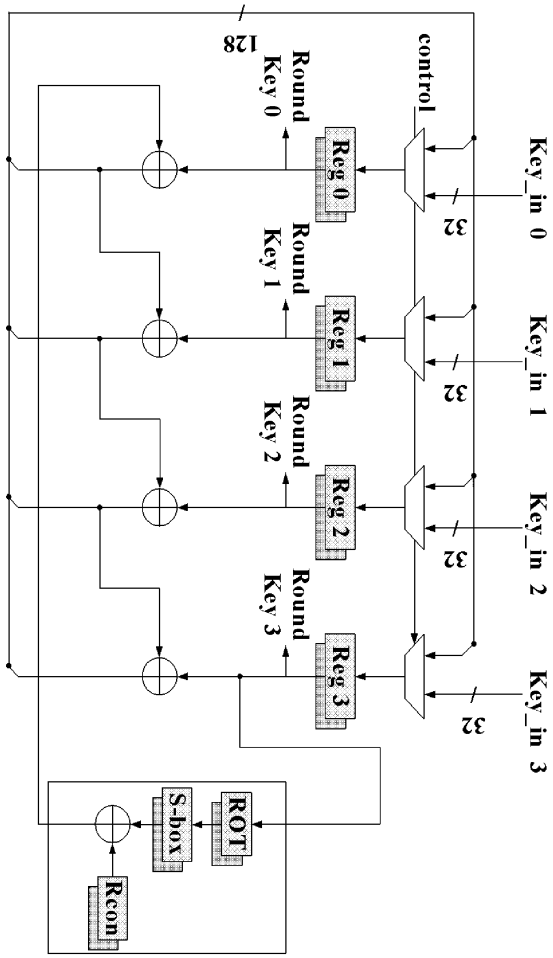
도면20


reset
slave not selected
no new transaction
transaction over
valid transaction
Address Phase
Data Phase
valid transaction WRITE
valid transaction READ
protocol violation
Error Phase
Read Wait Phase

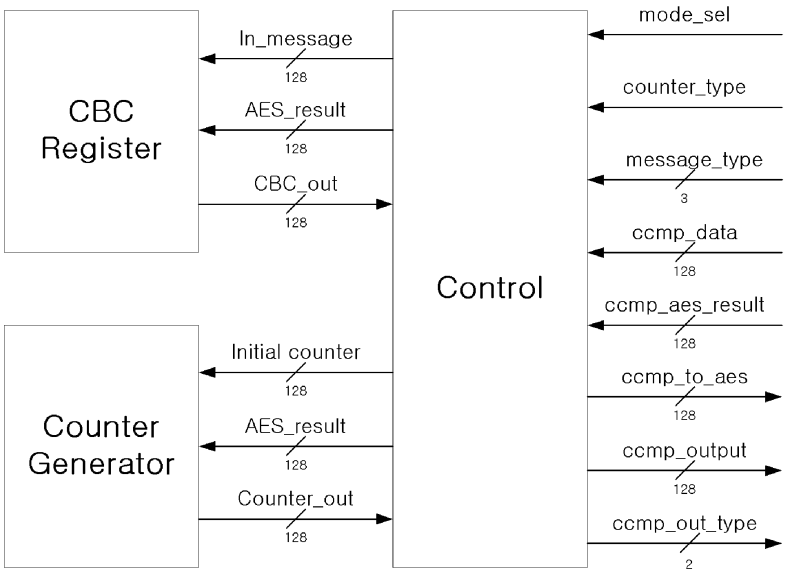
도면21


AMBA AHB
AMBA FSM
Register File
Counter Generator
Control Module
data in
data out
CBC Register
AES-128
data in
AddRoundKey
SubBytes
ShiftRows
MixColumns
Key Scheduler
Round Key[0]
Round Key[1]
Round Key[2]
Round Key[9]
Round Key[10]

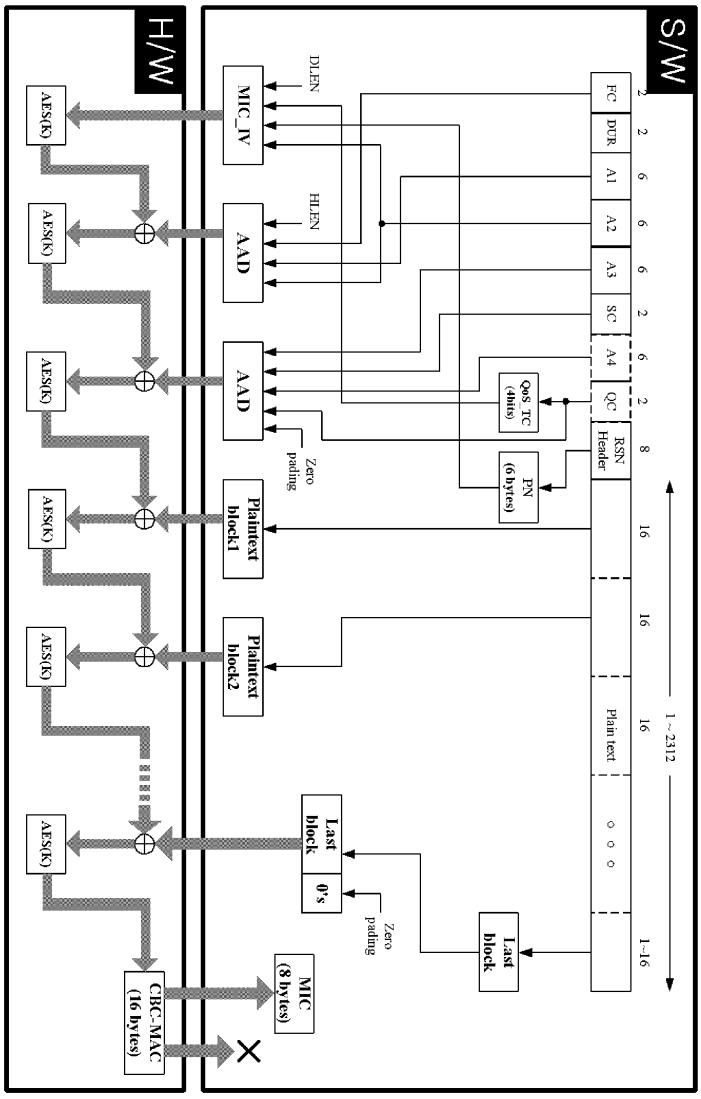
도면22


128
Round Key 0
Round Key 1
Round Key 2
Round Key 3
control
Reg 0
Reg 1
Reg 2
Reg 3
32
Key_in 0
Key_in 1
Key_in 2
Key_in 3
S-box
ROT
Rcon

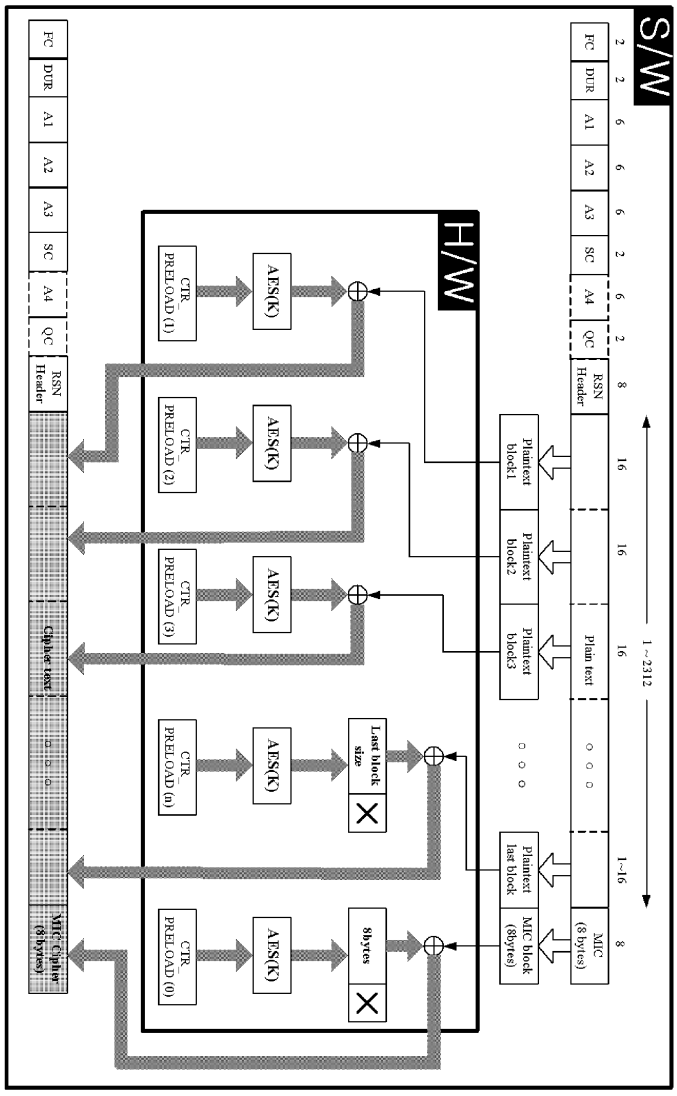
도면23


CBC Register
In_message
128
AES_result
128
CBC_out
128
Counter Generator
Initial counter
128
AES_result
128
Counter_out
128
Control
mode_sel
counter_type
message_type
3
ccmp_data
128
ccmp_aes_result
128
ccmp_to_aes
128
ccmp_output
128
ccmp_out_type
2

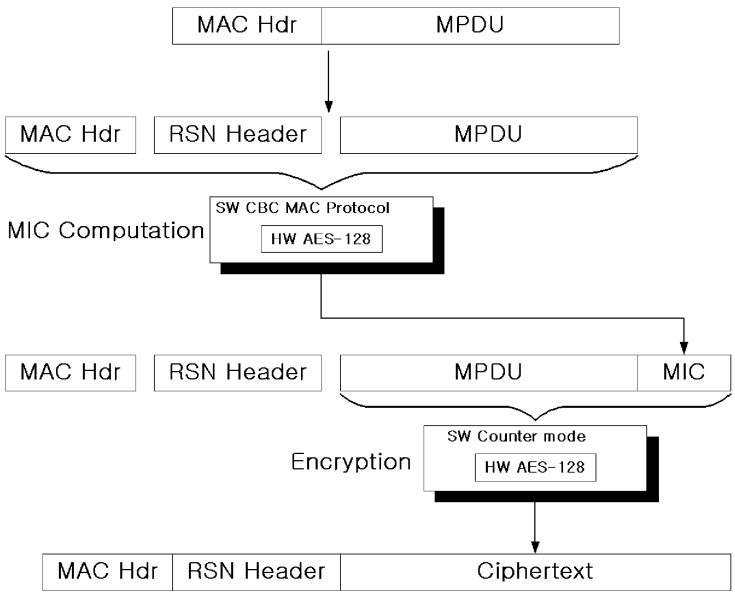
도면24


S/W
H/W
FC
DUR
A1
A2
A3
SC
A4
QC
RSN Header
Plain text
2
2
6
6
6
2
6
2
8
16
16
16
1~2312
1~16
MIC_IV
AAD
AAD
Plaintext block1
Plaintext block2
Last block
0's
Last block
DLEN
HLEN
Zero pading
Zero pading
QoS TC (4bits)
PN (6 bytes)
AES(K)
AES(K)
AES(K)
AES(K)
AES(K)
AES(K)
MIC (8 bytes)
CBC-MAC (16 bytes)

도면25


S/W
H/W
FC
DUR
A1
A2
A3
SC
A4
QC
RSN Header
Plaintext block1
Plaintext block2
Plaintext block3
Plain text
Plaintext last block
MIC block (8bytes)
MIC (8 bytes)
Cipher text
MIC Cipher (8 bytes)
CTR_ PRELOAD (1)
CTR_ PRELOAD (2)
CTR_ PRELOAD (3)
CTR_ PRELOAD (n)
CTR_ PRELOAD (0)
AES(K)
Last block size
8bytes
1 ~ 2312
1~16

도면26


MAC Hdr
MPDU
MAC Hdr
RSN Header
MPDU
MIC Computation
SW CBC MAC Protocol
HW AES-128
MAC Hdr
RSN Header
MPDU
MIC
Encryption
SW Counter mode
HW AES-128
MAC Hdr
RSN Header
Ciphertext