



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0074042
(43) 공개일자 2019년06월27일

(51) 국제특허분류(Int. Cl.)
G06F 21/85 (2013.01) G06F 21/60 (2013.01)
G06F 21/72 (2013.01)
(52) CPC특허분류
G06F 21/85 (2013.01)
G06F 21/60 (2013.01)
(21) 출원번호 10-2017-0175397
(22) 출원일자 2017년12월19일
심사청구일자 2017년12월19일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
강진구
서울특별시 서초구 양재대로2길 109, 101동 1104호(우면동, 서초참누리에코리치)
박청대
대구광역시 동구 방촌로1길 109, 103동 307호(방촌동, 성호그린빌라트)
(74) 대리인
양성보

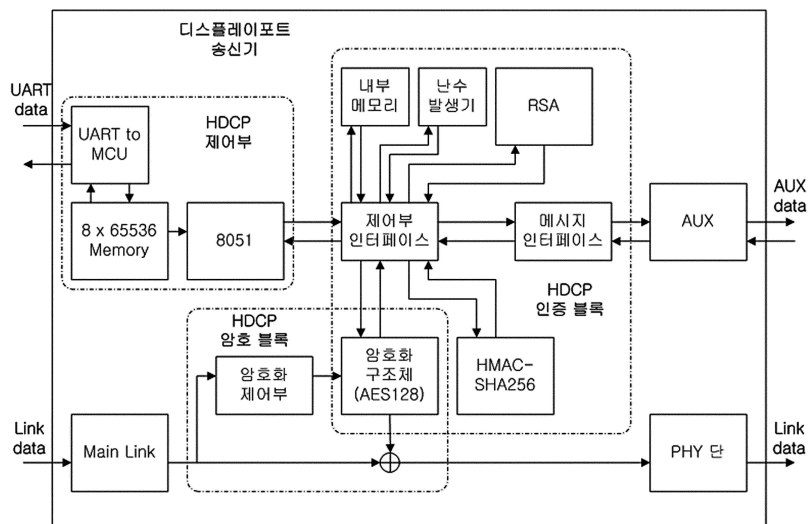
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 디스플레이포트용 HDCP 설계

(57) 요약

디스플레이포트용 HDCP 기술에 관한 것이다. 디스플레이포트용 HDCP 시스템은, 디스플레이 인터페이스인 디스플레이포트(DisplayPort)의 보조채널(AUX)에 연결되어 송신기 인터페이스와 수신기 인터페이스가 HDCP(High Definition Contents Protect)를 지원하는 기기인지 인증하고 디지털 데이터의 암호화와 복호화에 필요한 암호 키를 공유하는 인증 블록; 및 상기 디스플레이포트의 메인링크(Main Link)에 연결되어 상기 암호 키를 이용하여 상기 송신기 인터페이스에서 상기 디지털 데이터를 암호화하고 상기 수신기 인터페이스에서 상기 암호화된 디지털 데이터를 복호화하는 암호 블록을 포함하고, 상기 인증 블록을 제어하기 위한 제어부를 MCU(micro controller unit)로 구성하고, 상기 인증 블록은 상기 MCU로부터 제어 신호를 입력받고 상기 인증 블록의 상태 신호를 상기 MCU로 출력하는 제어부 인터페이스를 포함한다.

대표도



(52) CPC특허분류

G06F 21/72 (2013.01)

명세서

청구범위

청구항 1

디스플레이 인터페이스인 디스플레이포트(DisplayPort)의 보조채널(AUX)에 연결되어 송신기 인터페이스와 수신기 인터페이스가 HDCP(High Definition Contents Protect)를 지원하는 기기인지 인증하고 디지털 데이터의 암호화와 복호화에 필요한 암호 키를 공유하는 인증 블록; 및

상기 디스플레이포트의 메인링크(Main Link)에 연결되어 상기 암호 키를 이용하여 상기 송신기 인터페이스에서 상기 디지털 데이터를 암호화하고 상기 수신기 인터페이스에서 상기 암호화된 디지털 데이터를 복호화하는 암호 블록

을 포함하고,

상기 인증 블록을 제어하기 위한 제어부를 MCU(micro controller unit)로 구성하고,

상기 인증 블록은 상기 MCU로부터 제어 신호를 입력받고 상기 인증 블록의 상태 신호를 상기 MCU로 출력하는 제어부 인터페이스를 포함하는 것

을 특징으로 하는 디스플레이포트용 HDCP 시스템.

청구항 2

제1항에 있어서,

상기 MCU는 상기 디스플레이포트 외부에 연결되며 외부 메모리에 저장된 프로그램 코드에 의해 동작하는 것

을 특징으로 하는 디스플레이포트용 HDCP 시스템.

청구항 3

제1항에 있어서,

상기 제어부 인터페이스는 상기 MCU의 출력이 바뀌기 전까지 출력 신호를 고정하고 상기 MCU의 출력이 바뀌면 HDCP의 클록에 동기화된 펄스 신호를 생성하는 것

을 특징으로 하는 디스플레이포트용 HDCP 시스템.

청구항 4

제1항에 있어서,

상기 인증 블록은 메시지 인터페이스를 더 포함하고,

디스플레이포트 송신기용 HDCP의 메시지 인터페이스는 인증 과정에서 생성된 메시지를 AUX 메시지 형태로 변경하여 상기 디스플레이포트의 보조채널(AUX)로 전달하고,

디스플레이포트 수신기용 HDCP의 메시지 인터페이스는 인증 과정에서 생성된 메시지를 수신기의 메모리에 입력하도록 해당 메시지 데이터와 함께 메모리 제어 신호를 출력하는 것

을 특징으로 하는 디스플레이포트용 HDCP 시스템.

청구항 5

제1항에 있어서,

상기 암호 블록은 암호화 상태 신호를 감지하고 암호화 여부를 결정하는 암호화 제어부와, 암호 데이터열을 생성하는 암호화 구조체를 포함하고,

상기 암호화 제어부는 상기 메인링크의 컨트롤 심볼에 기초하여 상기 암호화 구조체의 동작을 제어하는 것

을 특징으로 하는 디스플레이포트용 HDCP 시스템.

청구항 6

제5항에 있어서,

상기 암호화 구조체는 상기 메인링크의 클럭에 따라 암호 데이터열을 생성하고 상기 암호 데이터열을 메인링크 데이터와 XOR 연산을 하는 동작을 하는 것

을 특징으로 하는 디스플레이포트용 HDCP 시스템.

발명의 설명

기술 분야

[0001] 아래의 설명은 데이터 암호화 기술에 관한 것이다.

배경 기술

[0002]최근 저작권과 관련하여 사회적 이슈가 대두되는 가운데 보안의 중요성이 부각되고 있다. 이에 따라 디지털 데이터의 암호화에 대한 요구가 많아지면서 일반 데이터뿐 아니라 HDMI(High Definition Multimedia Interface)나 디스플레이포트(DisplayPort)와 같은 디스플레이 인터페이스에도 데이터 보호가 요구되고 있다.

[0003]이러한 요구에 맞추어 데이터 암호화 표준인 HDCP(High Definition Contents Protect)가 발표되었다. HDCP를 필요로 하는 데이터는 HDCP가 지원되지 않는 기기로의 전송이 불가능하며 수신기의 인증여부를 판단하여 데이터를 전송하기 때문에 공격자에 대해서 안전하다. 또한 HDCP는 AES-CTR을 이용한 데이터 암호를 사용하고 있으며 이 암호는 데이터가 전송됨에 따라 바뀌기 때문에 외부에서 쉽게 공격할 수 없으며 비교적 고속의 암호화가 가능하여 고속 인터페이스에 적합하다. 또한 안정성 높은 암호 알고리즘인 RSA를 통해 키 교환을 하여 키 교환 도중의 공격에 대해서도 안전하다. 한 번 키 교환을 한 기기 사이에는 간략화 된 키 교환 방법을 사용함으로써 빠른 키 교환이 가능하며 올바르게 받은 수신기들의 정보를 공유하여 키 교환을 차단함으로써 키 교환 상의 위험을 최소화한다.

[0004]이렇듯 HDCP의 속도와 안전성으로 인해 HDCP를 지원하는 기기들이 요구되고 있다. HDCP를 지원할 수 있는 인터페이스들은 HDMI, 디스플레이포트, USB(Universal Serial Bus), 이더넷(Ethernet) 등이 있다. 이 중 디스플레이포트는 디지털 영상정보와 음성정보를 전달하는 디스플레이 인터페이스로 최근 증가하는 데이터 전송속도와 4K 이상의 초고해상도 디스플레이 수요에 따라 많은 PC와 모바일 기기에서 채택되고 있다.

발명의 내용

해결하려는 과제

[0005]HDCP 지원기기에 대한 수요를 만족하기 위해 디스플레이포트(DisplayPort)에서 전달되는 비디오, 오디오 데이터를 암호화하기 위한 HDCP(High Definition Contents Protect) 구조를 제공한다.

[0006]HDCP는 송신기 인터페이스와 수신기 인터페이스에 모두 존재하며 송신기에서 수신기로 전송되는 디지털 내용들을 암호화하여 인증되지 않은 기기가 송신기에서 전송되는 내용들을 알 수 없게 하는 데에 목적이 있다.

과제의 해결 수단

[0007]디스플레이 인터페이스인 디스플레이포트(DisplayPort)의 보조채널(AUX)에 연결되어 송신기 인터페이스와 수신기 인터페이스가 HDCP(High Definition Contents Protect)를 지원하는 기기인지 인증하고 디지털 데이터의 암호화와 복호화에 필요한 암호 키를 공유하는 인증 블록; 및 상기 디스플레이포트의 메인링크(Main Link)에 연결되어 상기 암호 키를 이용하여 상기 송신기 인터페이스에서 상기 디지털 데이터를 암호화하고 상기 수신기 인터페이스에서 상기 암호화된 디지털 데이터를 복호화하는 암호 블록을 포함하고, 상기 인증 블록을 제어하기 위한 제어부를 MCU(micro controller unit)로 구성하고, 상기 인증 블록은 상기 MCU로부터 제어 신호를 입력받고 상기 인증 블록의 상태 신호를 상기 MCU로 출력하는 제어부 인터페이스를 포함하는 것을 특징으로 하는 디스플레이포트용 HDCP 시스템을 제공한다.

[0008]일 측면에 따르면, 상기 MCU는 상기 디스플레이포트 외부에 연결되며 외부 메모리에 저장된 프로그램 코드에 의

해 동작한다.

[0009] 다른 측면에 따르면, 상기 제어부 인터페이스는 상기 MCU의 출력이 바뀌기 전까지 출력 신호를 고정하고 상기 MCU의 출력이 바뀌면 HDCP의 클럭에 동기화된 펄스 신호를 생성한다.

[0010] 또 다른 측면에 따르면, 상기 인증 블록은 메시지 인터페이스를 더 포함하고, 디스플레이포트 송신기용 HDCP의 메시지 인터페이스는 인증 과정에서 생성된 메시지를 AUX 메시지 형태로 변경하여 상기 디스플레이포트의 보조 채널(AUX)로 전달하고, 디스플레이포트 수신기용 HDCP의 메시지 인터페이스는 인증 과정에서 생성된 메시지를 수신기의 메모리에 입력하도록 해당 메시지 데이터와 함께 메모리 제어 신호를 출력한다.

[0011] 또 다른 측면에 따르면, 상기 암호 블록은 암호화 상태 신호를 감지하고 암호화 여부를 결정하는 암호화 제어부와, 암호 데이터열을 생성하는 암호화 구조체를 포함하고, 상기 암호화 제어부는 상기 메인링크의 컨트롤 심볼에 기초하여 상기 암호화 구조체의 동작을 제어한다.

[0012] 또 다른 측면에 따르면, 상기 암호화 구조체는 상기 메인링크의 클럭에 따라 암호 데이터열을 생성하고 상기 암호 데이터열을 메인링크 데이터와 XOR 연산을 하는 동작을 한다.

발명의 효과

[0013] 본 발명의 실시예들에 따르면, 제안된 HDCP는 인증 블록이 상태 머신이 아니라 MCU(Micro Controller Unit)로 인해 제어되어 동작의 유연성을 높일 수 있다. 또한, 이런 구조로 인해 설계 시 새로운 동작을 시험할 때마다 새롭게 디지털 회로를 합성할 필요 없이 실시간으로 프로그램코드를 변경하여 새로운 동작을 시험할 수 있다. 그리고, 향후 최신 HDCP 표준이 발표될 때 프로그램코드 변경을 통해 동작을 추가 및 개선할 수 있으며 프로그램 코드를 변경하면 디스플레이포트 외에 HDMI와 같은 디스플레이 인터페이스와도 호환할 수 있을 것이다.

[0014] 본 발명의 실시예들에 따르면, 제안된 암호화 구조체는 메인링크 데이터와 암호 데이터열이 서로 다른 클럭으로 생성됨으로 인해 XOR 연산 시 연산 딜레이로 인해 올바른 암호 데이터열이 메인링크 데이터와 연산되지 않는 문제를 해결하여 고속의 메인링크에서 안정적으로 데이터를 암호화 할 수 있다.

도면의 간단한 설명

[0015] 도 1은 본 발명에 따른 디스플레이포트용 HDCP 구조를 도시한 것이다.

도 2는 해시 메시지 인증 코드 알고리즘 블록도를 도시한 것이다.

도 3은 SHA-256 알고리즘 블록도를 도시한 것이다.

도 4는 SHA-256 압축 함수 블록도를 도시한 것이다.

도 5는 SHA-256 압축 함수 라운드 블록도를 도시한 것이다.

도 6은 128bit 고급 암호화 표준 알고리즘 블록도를 도시한 것이다.

도 7은 S-box의 내용을 도시한 것이다.

도 8은 행 이동의 동작을 도시한 것이다.

도 9 내지 도 10은 키 확장 블록도를 도시한 것이다.

도 11은 K_{in} 이 저장되어 있지 않은 수신기와의 인증 및 키 교환 과정을 도시한 것이다.

도 12는 K_{in} 이 저장되어 있는 수신기와의 인증 및 키 교환 과정을 도시한 것이다.

도 13은 수신기 인증서 서명 검증 과정을 도시한 것이다.

도 14는 지역성 검사 과정을 도시한 것이다.

도 15는 시스템 재개가능성 메시지 내용을 도시한 것이다.

도 16은 디스플레이포트용 HDCP 내 암호 블록의 구조를 도시한 것이다.

도 17은 암호화 구조체 블록도를 도시한 것이다.

도 18은 디스플레이포트 송신기용 HDCP 인증블록의 구조를 도시한 것이다.

도 19는 디스플레이포트 수신기용 HDCP 인증블록의 구조를 도시한 것이다.

도 20은 암호화 구조체의 구성을 도시한 것이다.

도 21은 암호화 상태 신호와 HDCP 암호 동작을 도시한 것이다.

도 22는 8051이 포함된 디스플레이포트 송신기용 HDCP 구조를 도시한 것이다.

도 23은 8051이 포함된 디스플레이포트 수신기용 HDCP 구조를 도시한 것이다.

도 24는 디스플레이포트 송신기용 HDCP 인증 블록의 제어 과정을 도시한 것이다.

도 25는 디스플레이포트 수신기용 HDCP 인증 블록의 제어 과정을 도시한 것이다.

도 26은 PC to MCU 직렬통신 블록의 상태도를 도시한 것이다.

발명을 실시하기 위한 구체적인 내용

[0016] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0018] 본 발명에서는 HDCP 지원기기에 대한 수요를 만족하기 위해 디스플레이포트 송신기 및 수신기 용 HDCP 구조를 제공한다. 디스플레이포트용 HDCP는 Verilog HDL을 이용해 설계되며 Digital Content Protection LLC에서 발표하는 규격 HDCP 2.2을 기준으로 한다.

[0019] 도 1은 디스플레이포트용 HDCP의 구조를 도시한 것이다.

[0020] HDCP는 HDMI, 디스플레이포트, USB 등의 데이터 통신 인터페이스에서 전송 데이터를 암호화 및 복호화하는 블록이다. 도 1을 참조하면, 디스플레이포트용 HDCP는 디스플레이포트의 보조채널(AUX)에 연결되어 기기인증 및 키교환을 하는 인증 블록(Authentication Block)과, 디스플레이포트의 메인링크(Main Link)에 연결되어 전송 데이터를 암호화 및 복호화하는 암호 블록(Cipher Block)으로 나뉘어서 동작한다.

[0021] 인증 블록(Authentication Block)

[0022] 송신기의 HDCP에서는 암호화가 필요한 데이터를 보내기 전에 보조채널을 통해 수신기와 통신하여 데이터를 보내도 되는 기기인지 판단하고 데이터 암호에 필요한 키를 교환한다. HDCP 내 인증블록은 이때 필요한 알고리즘 블록과 제어부로 구성된다. 인증단계는 인증 및 키 교환(AKE), 지역성 검사(Locality Check), 세션 키 교환(SKE) 3단계로 나뉜다. 이때 RSA, HMAC-SHA256, AES 총 3가지 암호 알고리즘이 사용된다.

[0023] (1) RSA 암호 알고리즘

[0024] RSA 암호 알고리즘은 HDCP 내 암호 알고리즘 중 가장 많은 시간이 소요되면서 안전성이 가장 높은 알고리즘으로, 송신기가 수신기의 인증서와 시스템 재개가능성 메시지(SRM)의 인증여부를 판단할 때, 그리고 처음 통신하는 송신기와 수신기 사이에서 인증 및 키 교환(AKE) 단계에서 송신기가 마스터키를 생성하고 수신기로 보낼 때 사용된다.

[0025] RSA는 모듈로 연산의 성질을 이용하여 암호화하는 키와 복호화하는 키가 서로 다른 비대칭 키 교환 암호 알고리즘이다. 이때 기존 데이터와 키, 암호화된 데이터의 관계는 관계식 1과 같다.

[0026] [관계식 1]

$$c = m^e \bmod n$$

$$m = c^d \bmod n$$

[0027] 여기서 m은 기존 데이터이고 d는 비밀 키, e와 n은 공개 키, c는 암호화된 데이터이다. 암호화하는 키인 e와 n은 외부로 공개되어 데이터 송신기에서 이 키로 데이터를 암호화할 수 있으며 복호화하는 키인 d는 데이터 수신기의 외부로 공개되지 않기 때문에 제 3자가 암호화된 데이터를 복구할 수 없다.

[0029] 관계식 2와 같은 관계를 e와 d와 n을 정하면 어떤 m과 c에 대해서도 관계식 1이 성립될 수 있다.

[0030] [관계식 2]

$n = pq$ ($p \neq q$), p 와 q 는 모두 소수(prime number)

$\Phi(n) = (p-1)(q-1)$

$\gcd(\Phi(n), e) = 1$, $e < \Phi(n)$

$de \equiv 1 \pmod{\Phi(n)}$

[0031]

[0032] 위 식이 성립됨을 증명하면 다음과 같다.

[0033] [관계식 3]

$c^d \equiv m^{ed} \equiv m^{k(p-1)(q-1)+1}$

$m^{k(p-1)(q-1)+1} \equiv (m^{p-1})^{k(q-1)} m \equiv 1^{k(q-1)} m \pmod{p}$

$1^{k(q-1)} m \pmod{p} \equiv m \pmod{p} \equiv m \pmod{n} \because n = pq$

[0034]

[0035] (2) 해시 메시지 인증 코드(HMAC-SHA256)

[0036] 해시 메시지 인증 코드는 인증 및 키 교환(AKE) 단계와 지역성 검사(Locality Check) 단계에서 수신기가 송신기의 메시지를 올바르게 받고 키와 난수를 제대로 복원 했는지를 판단하기 위해 사용된다.

[0037] 데이터의 적합성을 판단하기 위해서 부수적으로 전송하는 작은 데이터의 코드를 생성하기 위해 사용되며 임의의 길이의 데이터를 암호화키를 이용하여 정해진 길이의 코드로 변환하는 알고리즘이다. 송신기와 수신기는 동일한 암호화키를 공유하며 일반적으로 수신기는 이 알고리즘을 이용하여 전송된 데이터로부터 코드를 추출하고 이 코드가 송신기에서 보낸 코드와 동일한지 판단하여 적합한 송신기가 적합한 암호화키를 가진 상대인지 판단한다. HDCP에서는 이와 다르게 수신기에서 송신기로부터 받은 데이터를 이용해 코드를 생성한 후 송신기가 수신기로부터 코드를 읽고 송신기가 생성한 코드와 비교하여 적합한 암호화키를 가졌는지 판단한다.

[0038] 임의의 길이의 데이터를 정해진 길이의 코드로 압축시키는 해시 함수 알고리즘을 이용하여 사용하는 해시 함수 X에 따라 HMAC-X로 명명된다. HDCP에서는 해시함수 SHA256을 이용하는 HMAC-SHA256을 사용한다. 해시 메시지 인증코드 알고리즘은 도 2와 같다.

[0039] 도 2를 참조하면, K+는 256bits의 입력키의 뒤에 256bits만큼 '0'을 채워 512bits가 되게 확장시킨 값이고 ipad는 8bits 숫자 0x36을 64번 반복한 512bits 값, opad는 8bits 숫자 0x5c를 64번 반복한 512bits 값이다. IV는 256bits 길이의 초기 해시 값으로, 3부터 시작하여 가장 작은 소수 8개를 각각 제공근한 값들의 소수점 자리 수 32bits를 모은 것이다.

[0040] K+와 ipad를 XOR 연산한 값을 Si라 하고 이 값의 뒤에 확장된 입력 데이터를 붙여서 해시 함수의 입력으로 넣으면 256bits 길이의 H(Si||M)이 생성된다.

[0041] K+와 opad를 XOR 연산한 값을 So라 하고 이 값의 뒤에 H(Si||M)를 512bits가 되도록 확장된 값을 붙여서 해시 함수의 입력으로 넣어 출력코드 HMAC(K,M)을 생성한다.

[0042] 이때 쓰이는 해시함수 SHA256의 알고리즘은 도 3과 같다.

[0043] 입력 데이터는 그 길이가 512bits의 배수가 되도록 확장(padding)되어 해시 함수의 입력으로 들어가는데 그 규칙은 수학식 1과 같다.

[0044] [수학식 1]

$L + 1 + k = 448 \pmod{512}$ (L은 입력 데이터의 bit 수)

[0045]

[0046] 입력 데이터의 마지막 bit 다음 bit는 '1'로 하고 다음 수식을 만족하는 k만큼 '0'을 그 뒤에 붙인다. 그 후 원래 입력 데이터의 bit 수를 나타내는 64bits 숫자를 붙여서 512bits의 배수가 되도록 한다. SHA256에 사용되는

압축함수(Compression Function)의 내용은 도 4 내지 도 5와 같다.

여기서 sum은 산술연산 덧셈이고 다른 논리 연산들의 정의는 다음과 같다.

$$\begin{aligned}\sigma_0(W) &= RotR7(W) \oplus RotR18(W) \oplus ShR3(W) \\ \sigma_1(W) &= RotR17(W) \oplus RotR19(W) \oplus ShR10(W) \\ \Sigma_1(e) &= RotR6(e) \oplus RotR18(e) \oplus RotR25(e) \\ \Sigma_0(a) &= RotR2(a) \oplus RotR13(a) \oplus RotR22(a) \\ Ch(e, f, g) &= (e \cdot f) \oplus (\bar{e} \cdot g) \\ Maj(a, b, c) &= (a \cdot b) \oplus (a \cdot c) \oplus (b \cdot c)\end{aligned}$$

여기서 RotRx(W)는 x-bit만큼 W를 오른쪽으로 회전시키는 연산이고 ShRx(W)는 x-bit만큼 W를 오른쪽으로 이동시키고 왼쪽은 '0'으로 채우는 연산이다.

(3) 고급 암호화 표준(AES)

고급 암호화 표준(AES)는 HDCP의 인증블록과 암호블록 모두 사용되며 특히 암호블록에서는 카운터모드(AES-CTR)로 사용하여 매 128bit마다 새로운 암호데이터가 나오게 한다. 인증블록에서는 송신기와 수신기가 교환한 난수를 이용해 인증에 필요한 키를 생성하는 데에 사용된다.

데이터 암호화 표준(DES)을 대체하여 미국 표준 기술 연구소(NIST)가 제정한 암호화 방식으로 송신기와 수신기가 같은 암호화키를 공유하는 대칭키 암호 알고리즘이다. 128bit의 입력 데이터와 출력 데이터를 가지며 이용하는 키 길이에 따라 AES-128, AES-192, AES-256으로 나누어진다. 각 뒤에 붙은 숫자가 이용하는 키의 비트 수이며 HDCP에서는 128bit 길이의 키를 이용하는 AES-128을 사용한다.

AES는 128bit의 입력 데이터를 4*4의 데이터 행렬로 취급하여 처리하고 이 행렬에 4가지 연산을 반복함으로써 데이터를 암호화 한다. 전체 암호화 알고리즘은 도 6과 같다.

Add Round Key(라운드 키 덧셈)은 확장키와 데이터를 XOR 연산을 하는 블록이며 Substitute Bytes(바이트 치환)은 입력 값을 바이트 단위로 S-box의 해당 바이트로 치환하는 블록, Shift Rows(행 이동)은 행을 왼쪽으로 이동시키는 블록, Mix Columns(열 혼합)은 열의 모든 바이트를 특정 연산과정을 통해 섞는 블록이다. S-box의 내용은 도 7, 행 이동(Shift Rows)의 동작은 도 8과 같고 열 혼합(Mix Columns)의 연산은 다음과 같다.

$$\begin{bmatrix} S'_{0,c} \\ S'_{0,c} \\ S'_{0,c} \\ S'_{0,c} \end{bmatrix} = \begin{bmatrix} 02 & 02 & 02 & 02 \\ 01 & 01 & 01 & 01 \\ 01 & 01 & 01 & 01 \\ 03 & 03 & 03 & 03 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{0,c} \\ S_{0,c} \\ S_{0,c} \end{bmatrix}$$

확장될 열의 번호가 4의 배수라면 도 9와 같은 연산으로 확장되고 그렇지 않은 경우도 10과 같은 연산으로 확장된다.

(4) 인증 및 키 교환(AKE)

디스플레이포트 송신기와 수신기가 연결되고 전송데이터가 HDCP 지원이 필요하면 송신기와 수신기는 보조 채널(AUX)을 통해 인증 및 키 교환을 시작한다.

도 11은 Km이 저장되어 있지 않은 수신기와의 인증 및 키 교환 과정을, 도 12는 Km이 저장되어 있는 수신기와의 인증 및 키 교환 과정을 도시한 것이다.

첫 번째 순서로 송신기가 64bits 크기의 난수 R_{tx} 를 생성한다. 그 후 송신기의 정보를 담고 있는 24bits 값 TxCaps와 함께 R_{tx} 를 수신기로 AUX를 통해 전송한다. 수신기에서는 이 메시지(여기서 송신기와 수신기가 AUX를 통해 교환하는 데이터를 메시지라 한다.)를 읽으면 64bits 크기의 난수 R_{rx} 생성하고 수신기 인증서와 수신기의

정보를 담고 있는 24bits 값 R_{xCaps} 를 R_{rx} 와 함께 송신기로 전송한다.

[0061] 송신기에서는 수신기로 메시지를 보내고부터 100ms 뒤에 수신기로부터 메시지를 읽고 수신기의 ID가 수신기 ID 리스트에 있는지 확인한다. 만약 수신기가 리스트에 있다면 해당 ID의 K_m , m , $E_{kh}(K_m)$ 을 읽는다. 만약 수신기가 리스트에 없다면 인증서의 서명을 검증한다. 인증서의 내용은 표 1과 같고, 서명 검증 과정은 도 13과 같다.

표 1

이름	크기(bits)	Bit position	기능
Receiver ID	40	4175:4136	수신기 고유의 ID이며 20개의 1과 20개의 0으로 구성되어 있다.
Receiver Public Key	1048	4135:3088	HDPC 수신기의 고유한 RSA 공개키이며 흔히 $k_{pub_{rx}}$ 로 명명한다. 첫 1024bits는 n 이며 다음 24bits는 e 이다.
Reserved2	4	3087:3084	0x0 혹은 0x1의 값을 가진다.
Reserved1	12	3083:3072	0x000값을 가진다.
DCP LLC Signature	3072	3071:0	위 인증서의 내용을 보안 서명한 값이다. PKCS #1 V2.1: RSA Cryptography Standard에 정의된 RSASSA-PKCS1-v1_5 서명법칙을 따른다.

[0063] 인증서의 하위 3072bits는 인증서의 서명이며 이 서명은 인증서의 상위 1104bits를 해시 함수 계산한 값을 DCP LLC의 개인키로 RSA 암호화 한 값이다. 따라서 올바른 인증서 서명이라면 DCP LLC의 공개키로 RSA 복호화 했을 때 그 하위 256bits 값이 인증서의 상위 1104bits를 해시 함수 계산한 값과 동일하게 된다. 올바른 인증서 서명이 아니라면 인증과정을 중단하고 인증시작 전 으로 돌아간다.

[0064] 서명 검증이 성공하면 128 bits 크기의 난수 K_m 을 생성하고 수신기의 공개키를 이용하여 RSA 암호화하고 암호화된 $K_m(E_{k_{pub}}(K_m))$ 을 수신기로 전송한다. 또한 K_m 을 입력키로 하여 AES를 통해 128bits 키 $dkey0$, $dkey1$ 을 생성한다. 이것을 입력키로 하여 HMAC-SHA256을 통해 256bits 값 H 를 생성한다.

[0065] 수신기에서는 $E_{k_{pub}}(K_m)$ 을 읽은 뒤 수신기의 개인키로 복호화하여 K_m 을 구하고 송신기와 같은 방법으로 $dkey0$, $dkey1$ 을 생성한다. 이 것을 입력키로 하여 HMAC-SHA256을 통해 256bits 값 H' 을 생성하고 이 값을 송신기로 전송한다. 그 뒤 수신기의 개인키를 이용하여 해시함수 SHA256을 통해 k_h 를 생성하고 이 것을 입력키로 하여 m 을 AES 연산을 한 값을 K_m 과 XOR 연산하여 $E_{kh}(K_m)$ 을 생성한다. 여기서 m 은 R_{tx} 뒤에 R_{rx} 을 붙인 128bits 수이다.

[0066] 송신기는 $E_{kh}(K_m)$ 을 전송한 뒤 수신기의 ID가 시스템 재개가능성 메시지에 포함되어 있는지 확인하고 만약 포함되어 있다면 인증과정을 중단하고 인증시작 전으로 돌아간다. 만약 포함되어 있지 않다면 수신기에서 H' 을 생성하기까지 대기했다가 H' 을 읽고 난 뒤 자신이 생성한 H 와 같은지 비교한다. 만약 같다면 수신기가 $E_{kh}(K_m)$ 을 생성할 때까지 대기했다가 $E_{kh}(K_m)$ 을 읽는다. 그 뒤 수신기의 ID와 K_m , m , $E_{kh}(K_m)$ 을 수신기 ID 리스트에 저장하고 지역성 검사를 시작한다. 만약 H 와 H' 이 같지 않거나 대기 시간이 초과되면 인증과정을 중단하고 인증시작 전으로 돌아간다.

[0067] 송신기가 수신기의 ID를 초기에 수신기 ID 리스트에 가지고 있는 경우 해당 ID의 K_m , m , $E_{kh}(K_m)$ 을 읽고 $E_{kh}(K_m)$ 과 m 을 수신기로 전송한다. 수신기는 이 $E_{kh}(K_m)$ 와 m 을 이용하여 AES를 통해 K_m 을 구하고 K_m 을 입력키로 하여 AES를 통해 128bits 키 $dkey0$, $dkey1$ 을 생성한다. 이 것을 입력키로 하여 HMAC-SHA256을 통해 256bits 값 H' 을 생성하고 이 값을 송신기로 전송한다.

[0068] 송신기는 $E_{kh}(K_m)$ 과 m 을 전송한 뒤 수신기의 ID가 시스템 재개가능성 메시지에 포함되어 있는지 확인하고 만약 포함되어 있다면 인증과정을 중단하고 인증시작 전으로 돌아간다. 만약 포함되어 있지 않다면 수신기에서 H' 을 생성하기까지 대기했다가 H' 을 읽고 난 뒤 자신이 생성한 H 와 같은지 비교한다. 만약 같다면 지역성 검사를 시작하고 만약 H 와 H' 이 같지 않거나 대기 시간이 초과되면 인증과정을 중단하고 인증시작 전으로 돌아간다.

$$\begin{aligned}
 E_{k_{pub}}(K_m) &= RSA(K_m, K_{pub}) \\
 dkey0 &= AES(R_{tx} \parallel (R_{rx} \oplus 0x0000000000000000), K_m) \\
 dkey1 &= AES(R_{tx} \parallel (R_{rx} \oplus 0x0000000000000001), K_m) \\
 H(H) &= HMAC-SHA256(R_{tx} \parallel RxCaps \parallel TxCaps, K_d) \\
 K_d &= dkey0 \parallel dkey1 \\
 k_h &= SHA256(K_{privrx}) \\
 K_{privrx} &= p \parallel q \parallel d \pmod{p-1} \parallel d \pmod{q-1} \parallel q^{-1} \pmod{p}
 \end{aligned}$$

$$E_{kh}(K_m) = K_m \oplus AES(m, k_h)$$

인증 및 키 교환 과정에서 암호 알고리즘의 입력과 결과는 위와 같다. 여기서 왼쪽 항은 "알고리즘의 결과값", 오른쪽 항은 "알고리즘(입력값, 입력키)"이다.

(5) 지역성 검사(Locality Check)

도 14는 지역성 검사 과정을 도시한 것이다.

도 14를 참조하면, 인증 및 키 교환이 성공하면 송신기는 64bits 크기의 난수 R_n 을 생성하고 수신기로 전송한다. 그 뒤 R_n 을 입력으로 하여 HMAC-SHA256을 통해 해시코드 L 을 생성한다. 수신기는 송신기로 전송받은 R_n 으로 송신기와 같은 방법을 통해 해시코드 L' 을 생성한다. 송신기는 R_n 을 전송시간으로부터 20ms 뒤에 수신기로부터 L' 을 읽고 L 과 비교하여 같은지를 판단한다.

위 과정에서 알 수 있듯이 송신기에서 R_n 을 보내고 짧은 시간 안에 수신기로부터 정상적인 L' 을 읽을 수 있는지를 판단하여 송신기와 수신기가 서로 적절한 거리 안에 존재하는지를 판단한다. 만약 L 과 L' 이 같다면 지역성 검사가 성공한 것이고 세션 키 교환을 시작한다. 하지만 L 과 L' 이 같지 않다면 인증과정을 중단하고 인증시작 전으로 돌아간다.

여기서 L 과 L' 은 수학식 2를 통해 계산된다.

[수학식 2]

$$L(L') = HMAC-SHA256(R_n, k_d[63:0] \oplus R_{rx})$$

왼쪽 항은 "알고리즘의 결과값", 오른쪽 항은 "알고리즘(입력값, 입력키)"이며 [63:0]은 하위 63비트에서 0까지의 값을 말한다.

(6) 세션 키 교환(SKE)

지역성 검사가 성공하면 송신기는 64bits 크기의 난수 R_{iv} 와 128bits 크기의 세션 키 K_s 를 생성한다. 그리고 $dkey0$ 과 같은 방법으로 $dkey2$ 를 생성하고 $dkey2$ 를 통해 K_s 를 암호화한다. 암호화된 K_s 를 R_{iv} 와 함께 수신기로 보내고 수신기는 암호화된 K_s 를 송신기와 같은 방법으로 복호화하여 K_s 를 복원한다.

$$dkey2 = AES(R_{tx} \parallel (R_{rx} \oplus 0x0000000000000002), K_m)$$

$$E_{dkey}(K_s) = K_s \oplus (dkey2[63:0] \oplus R_{rx})$$

인증 및 키 교환 과정에서 암호 알고리즘의 입력과 결과는 위와 같다. 여기서 왼쪽 항은 "알고리즘의 결과값", 오른쪽 항은 "알고리즘(입력값, 입력키)"이며 [63:0]은 하위 63비트에서 0까지의 값을 말한다.

(7) 재개가능성 검사

- [0084] 이론적으로 HDCP가 지원되는 수신기는 절대 개인키가 공개되지 않으며 신뢰성을 가지지만 간혹 개인키가 공개되거나 올바르게 않은 HDCP 수신기가 존재할 수도 있다. 이런 수신기와의 통신을 막기 위해서 DCP LLC에서 제공하는 시스템 재개가능성 메시지가 존재하며 도 15와 같이 구성되어 있다.
- [0085] HDCP 송신기는 시스템 재개가능성 메시지를 가지고 있으며 인증과정 중에 수신기가 재개가능성 메시지의 revocation list에 있는지 확인하고 만약 있다면 올바르게 않은 HDCP 수신기이기 때문에 인증을 중단한다. 이 시스템 재개가능성 메시지를 읽을 때에는 이 메시지가 올바른지 판단해야 하기 때문에 메시지 하단의 DCP LLC 서명과 메시지 내용을 검증하는 과정을 거친다. 그 과정은 AKE에서 수신기 인증서를 검증하는 과정과 일치하다.
- [0086] (8) DPCP 입출력
- [0087] 디스플레이포트 수신기는 보조채널(AUX)에 DPCD 메모리가 존재하여 송신기의 상태와 수신기의 상태를 저장하고 서로 메시지를 교환할 수 있다. 송신기가 AUX를 통해 수신기의 DPCD에 메시지를 쓰고 읽는 과정으로 메시지 교환을 하며 각 메시지마다 저장 위치가 지정되어 있다.
- [0088] (9) CP IRQ 인터럽트
- [0089] 인증과정 중 송신기는 수신기로부터 최소 3번, 최대 4번 메시지를 읽어 오는데 그 중 <AKE_Send_Cert>와 <LC_Send_L_prime>은 송신기가 메시지 전송 후 일정시간 뒤에 일방적으로 DPCD에서 읽고 <AKE_Send_H_prime>과 <AKE_Send_Pairing_Info>의 경우 수신기에서 IRQ 인터럽트를 발생시키고 나면 송신기가 DPCD에서 읽는다.
- [0090] 디스플레이포트 수신기는 자신의 상태를 디스플레이포트 송신기에게 알리기 위해 핫플러그디텍트(HPD) 라인을 짧은 시간동안 그라운드로 다운시켰다가 올리는 IRQ 인터럽트라는 방법을 사용하는데 그 중 HDCP가 필요한 부분에서 사용하는 것을 CP IRQ 인터럽트라 한다. 송신기는 IRQ 인터럽트를 감지하면 DPCD 메모리의 RxStatus 바이트를 읽고 바이트 값에 따라서 적절한 행동을 한다. 수신기는 송신기가 RxStatus 바이트를 읽고 난 뒤 RxStatus 바이트를 0으로 바꾼다.
- [0091] CP IRQ 인터럽트는 인증과정이 끝난 이후에 데이터 완전성이 좋지 않은 경우 발생되어 인증과정이 재시작할 수 있도록 한다.
- [0092] 암호 블록(Cipher Block)
- [0093] 도 16은 디스플레이포트용 HDCP 내 암호 블록의 구조를 도시한 것이다.
- [0094] 인증 블록에 의해 모든 인증과정이 끝나면 암호 블록에서 세션 키 K_S 를 이용해 암호 데이터열을 만들고 전송 데이터와 XOR 하여 암호화 한다. 이 과정에서 AES 암호 알고리즘이 사용된다. 암호 블록은 암호화 상태 신호를 감지하고 암호화 여부를 결정하는 암호화 제어부와 암호 데이터열을 생성하는 암호화 구조체로 구성되어 있다. 수신기의 암호화 제어부는 암호화 상태 신호를 감지하는 기능 이외에도 데이터 완전성을 검사하여 완전성 결핍 신호를 생성하는 기능도 한다.
- [0095] 디스플레이포트의 메인 링크(Main Link)는 데이터를 전송하는 라인 수에 따라 1-lane, 2-lane, 4-lane 이 있고 이 라인 수에 따라 매 링크클록마다 XOR 되는 전송 데이터의 크기가 달라진다. 예로 1-lane에서 데이터가 전송되면 매 링크클록마다 8bits의 암호 데이터가 전송 데이터와 XOR 된다. 이 경우 AES는 16번의 링크클록마다 하나의 암호 데이터열을 생성한다. 전송 데이터가 8bits의 암호 데이터와 XOR되면 다음 전송 데이터는 암호 데이터열의 다음 8bits의 데이터와 XOR 된다.
- [0096] (1) 암호화 구조체
- [0097] 암호화 구조체는 AES의 카운터모드(AES-CTR)를 사용하며 원리는 다음과 같다.
- [0098] 카운터모드는 초기 입력을 받고 암호 알고리즘의 출력이 생성될 때마다 입력 값에 1을 더하고 다음 출력을 생성하는 구조이다. 알고리즘 출력 값인 암호 데이터열과 원본 데이터를 XOR 연산하여 암호화 한다. 이런 식으로 원본 데이터와 XOR 연산되는 데이터를 계속 바꿈으로써 암호키와 원본 데이터를 알기 어렵게 한다. AES-CTR은 이 카운터모드의 동작원리를 따르는데 그때의 암호 알고리즘이 AES인 구조이다.
- [0099] 도 17은 암호화 구조체 블록도를 도시한 것이다.
- [0100] 도 17을 참조하면, 암호화 구조체는 AES의 키로 K_S 와 128bits 크기의 상수 1c128을 XOR 연산한 값을 사용하며 입력으로 R_{iv} 뒤에 64bits 크기의 카운터 수를 붙인 값을 사용한다. 이 카운터 수는 HDCP 인증과정이 끝난 직후 0

으로 초기화 되고 암호 데이터열이 생성될 때마다 1씩 증가한다. 암호 데이터열 생성주기는 링크클록에 의존하며 메인 링크의 레인 수가 많아지면 비례하여 짧아진다.

[0101] (2) 암호화 상태 신호

[0102] HDCP 인증블록에서 인증과정을 모두 마치면 디스플레이포트 송신기의 SR 생성 블록에서 SR을 CPSR로 바꾼다. 여기서 SR은 메인 링크로 전송되는 컨트롤 데이터로 Enhanced Framing Mode인 경우 0x1c7c7c1c(SR,BS,BS,SR)의 값을 가진다. CPSR은 0x1c3c3c1c(SR,CP,CP,SR)의 값을 가지며 HDCP 암호 블록의 암호화 제어부는 이 값을 감지하여 인증과정이 모두 마쳤음을 인지한다. CPSR이 전송되면 그 다음 메인 링크 데이터는 CPSR 대신 SR이 전송될 때까지 암호화된다. SR이 전송되더라도 암호화 구조체의 암호 데이터열과 카운터 수는 초기화 되지 않고 기존의 값을 계속 유지한다. CPSR이 다시 전송되면 기존의 값부터 시작하여 암호화를 계속 진행한다.

[0103] (3) 데이터 안전성 검사

[0104] HDCP는 암호화 방법으로 카운터모드를 사용하기 때문에 송신기와 수신기의 카운터가 조금만 달라도 암호화된 데이터가 복원되지 않는다. 이런 문제를 해결하기 위해 HDCP 수신기의 암호 블록은 메인 링크의 데이터 완전성을 검사해야 한다. 일정 크기의 데이터가 전송될 때까지 연속적으로 데이터가 제대로 복원되지 않으면 송신기와 수신기의 카운터가 맞지 않는 것으로 판단하고 CP IRQ 인터럽트를 통해 HDCP 송신기에게 이 상황을 알린다. HDCP 송신기는 이 신호를 받으면 새로 인증과정을 시작하고 기존의 키 값들과 카운터를 초기화한다.

[0105] HDCP 수신기는 복원된 VB-ID 바이트의 Bit 5 값을 읽고 이 비트가 일정한 시퀀스(0x531F)와 연속적으로 다른 값을 띠면 링크 완전성 검사 실패로 판단한다. 2프레임 동안 연속적으로 링크 완전성 검사 실패가 되면 데이터 완전성 검사 실패로 판단하고 암호화 불가 상태로 바뀐다. 이때 CP IRQ 인터럽트가 발생하게 된다.

[0106] VB-ID 바이트의 구성은 표 2와 같다.

표 2

이름	Bit Field	정의
VerticalBlanking_Flag	0	이 비트는 마지막 active line의 끝에서 1이 되고 vertical blanking period 동안 1을 유지한다. 이 비트는 video stream이 없을 때에도 1이 된다.
FieldID_Flag	1	이 비트는 top field의 마지막 active line 바로 뒤에 0이 된다. bottom field의 마지막 active line 바로 뒤에 1이 된다.
Interlace_Flag	2	이 비트는 main stream이 interlaced video인 경우 1이 된다. non-interlaced video이거나 video가 없으면 0이 된다.
NoVideoStream_Flag	3	이 비트는 이전의 BS가 video stream이 전송되지 않을 때 추가된 경우 일 때 1이 된다. 이 비트가 1이면 Mvid 7:0 값은 반드시 "don't care"이 되어야 한다.
AudioMute_Flag	4	이 비트는 audio가 mute 되었을 때 1이 된다.
HDCP SYNC DETECT	5	HDCP 동작에 따라 값을 가지며 특정 시퀀스의 비트 값을 가진다.
RESERVED	7:6	모두 0이다.

[0108] 이하에서는 디스플레이포트 용 HDCP의 설계에 대해 설명한다. 더불어 인증 블록의 제어부를 상태 머신에서 8051 MCU로 변경했을 때 이점과 FPGA를 이용해 설계 시 PC에서 8051 MCU로 프로그램 코드를 전송하기 위한 PC to MCU 모듈의 설계에 대해 설명한다.

[0109] 인증 블록 설계

[0110] 디스플레이포트 송신기 용 HDCP의 인증 블록은 도 18과 같이 설계하며 다음과 같이 동작한다. 제어부 인터페이스가 제어부로부터 동작신호를 받고 동작신호에 따라 인증 블록 내 다른 세부 블록들을 제어한다. 또한 각 알고리즘에서 나온 값들을 연산하고 현재 인증단계에 따라 메시지로 묶는다. 각 알고리즘 블록은 각각 키와 입력을 받아 출력 값을 생성하고 저장한다. 내부 메모리는 수신기 ID 리스트(RxID_list)와 시스템 재개가능성 메시지(SRM)을 저장하고 있으며 수신기 ID 리스트에 대한 읽기/쓰기 신호와 SRM에 대한 읽기/쓰기 신호를 개별적으로 받는다. 난수 생성기는 발진기를 이용하여 128bits 길이의 난수를 생성한다. 메시지 인터페이스는 생성된 메시지를 AUX 패킷으로 변경하여 디스플레이포트의 AUX 블록 내 링크 폴리시 메이커로 출력한다.

- [0111] 디스플레이포트 수신기 용 HDCP의 인증 블록은 도 19와 같이 설계한다. 송신기 용 HDCP 인증 블록과는 내부 메모리를 제외하면 대칭적인 구조이며 송신기 용 HDCP 인증 블록과 다르게 복호용 RSA 알고리즘 블록이 사용된다. 또한 메시지 인터페이스는 생성된 메시지를 DPCD 메모리에 쓸 수 있도록 메시지 데이터와 함께 DPCD 메모리 제어 신호를 출력한다.
- [0112] (1) AES 블록
- [0113] AES 블록은 인증 블록과 암호 블록 모두 사용하고 양 블록에서 모두 동일한 AES128을 사용하고 암호 블록에서 데이터 암호화를 하는 중에 인증 블록은 AES를 사용하는 일이 없기 때문에 하나의 AES 블록이 존재하고 이 것을 양 블록이 모두 사용한다.
- [0114] 암호 블록에서 AES 블록은 고속의 메인 링크 클럭에 따라 동작해야 하기 때문에 설계 시 데이터 처리량이 가장 중요시된다. 높은 데이터 처리량을 보장하기 위해 AES 블록은 하나의 로직 블록으로 설계된다. 즉 입력 클럭 없이 입력값과 키만 입력하면 출력값을 생성한다. 이 경우 매 라운드에 해당하는 연산 블록들이 모두 필요하기 때문에 한 라운드에 해당하는 연산 블록만 만들고 매 라운드의 연산 값을 저장하는 방법에 비해 자원 많이 사용하는 단점이 있다. 하지만 후자의 경우 FPGA 보드로 설계하였을 때 처리량이 디스플레이포트의 메인 링크 처리량에 못 미쳐서 암호화가 안 되는 문제가 발생한다. 이 외에 매 라운드에 해당하는 연산 블록들을 모두 만들고 매 라운드마다 연산 값을 저장하는 방법도 있지만 이 경우 필요이상의 높은 처리량을 가지고 많은 자원을 사용하는 문제가 있다. 따라서 AES 블록은 위의 방법으로 설계되고 출력값은 AES 블록의 후단에서 따로 클럭에 따라 저장된다.
- [0115] (2) HMAC-SHA256 블록
- [0116] HDCP 인증과정 중에는 HMAC의 일부로서가 아닌 SHA256의 동작만이 필요한 때(수신기 인증서의 서명 검증, 시스템 재개가능성 메시지 서명 검증, 수신기에서 k_h 를 생성할 때)가 있기 때문에 HMAC-SHA256 블록은 이 여부에 따라 SHA256 모드, HMAC 모드로 나뉘어서 동작하도록 설계한다. SHA256 모드의 경우 SHA256의 입력을 외부에서 직접 받도록 하였고 출력도 외부로 직접 나가게 된다. HMAC 모드인 경우 SHA256의 입출력은 HMAC-SHA256 블록 내부와 연결되어 HMAC의 해시함수로 동작한다. HDCP 인증과정 중에서 HMAC의 입력으로 들어가는 값은 그 길이가 AKE인 경우 112bits, Locality Check인 경우 64bits로 고정되어 있기 때문에 확장된 메시지의 하위 비트는 인증 과정에 따라 고정되어 있고 해시함수가 반복되는 경우는 항상 두 번이기 때문에 이 값들은 상수로 고정한다. SHA256이 외부로 입출력이 연결되는 경우 외부로부터 반복횟수도 입력받도록 하여 입력 메시지의 길이를 판단하는 기능을 생략한다.
- [0117] (3) 제어부 인터페이스
- [0118] 제어부 인터페이스는 디스플레이포트 외부의 8051 MCU와 연결되어 제어신호를 받고 상태신호를 출력하도록 설계한다. 제어신호는 인증 블록의 세부 블록들의 동작을 시작하는 starting signal과 현재 인증단계를 변경하는 state signal과 전송해야 할 메시지의 종류를 나타내는 message ID signal 등으로 구성되고 상태신호는 세부 블록들의 동작 완료를 나타내는 done signal과 인증동작 중 비교하는 값들(예. H와 H')의 일치를 나타내는 verifying signal 등으로 구성된다.
- [0119] MCU와 HDCP는 동작하는 클럭이 다르기 때문에 제어부 인터페이스는 MCU와 통신하기 위해서 MCU의 출력이 바뀌기 전까지 출력신호를 고정하고 MCU의 출력이 바뀌면 HDCP의 클럭에 동기화된 펄스 신호를 만들어 인증 블록 내 세부 블록이 제어되도록 설계한다.
- [0120] (4) 메시지 인터페이스
- [0121] 디스플레이포트 송신기 용 HDCP의 메시지 인터페이스는 송신기 인증 블록으로부터 1024bits 길이의 메시지를 입력으로 받아 현재 인증단계와 메시지 ID에 따라 접근할 DPCD 메모리의 주소를 정하고 주소값과 데이터, 메시지 길이를 디스플레이포트의 AUX 내에 있는 링크 폴리시 메이커로 출력하도록 설계한다. 링크 폴리시 메이커는 AUX에서 수신기로 전송할 메시지를 패키징하고 수신기로부터 전송된 패킷을 언패키징하는 역할을 하는 블록이다. 메시지 인터페이스에서 패키징, 언패키징을 하도록 설계하여 HDCP 메시지를 교환할 때에는 링크 폴리시 메이커가 단순히 메시지 인터페이스와 외부로 연결하는 역할을 하도록 한다.
- [0122] 디스플레이포트 수신기 용 HDCP의 메시지 인터페이스는 수신기 인증 블록으로부터 256bits 길이의 메시지를 입력받아 현재 인증단계와 메시지 ID에 따라 해당 DPCD 메모리 주소에 메시지를 입력하도록 한다. 또한 송신기로부터 메시지가 전송되어 DPCD 메모리에 저장되면 DPCD 메모리에서 메시지 인터페이스로 메시지 도착 신호가 전

송되어 그때부터 메시지 인터페이스가 DPCD 메모리로부터 해당 데이터를 읽도록 한다. 송신기에서 전송되는 메시지는 여러 데이터를 포함하는 경우가 많기 때문에 메시지 도착 신호는 송신기로부터 전송되는 마지막 데이터가 저장되었을 때 발생되도록 한다.

[0123] (5) 수신기 정보 블록

[0124] 디스플레이포트 송신기 용 HDCP의 인증 블록에는 수신기 정보와 시스템 재개가능성 메시지가 저장되어 있는 내부 메모리가 있으며 이 두 내용은 각각 개별적인 읽기/쓰기 동작을 가지도록 설계된다. 즉 수신기 정보를 저장하는 신호가 들어오면 별다른 주소지정이 없어도 수신기 정보를 저장하는 구간에 주소를 할당하여 현재 수신기 정보 목록 마지막에 새로운 수신기 정보를 저장하는 형태이다. 하나의 수신기 정보를 저장할 때 수신기 ID, K_m , $E_{kh}(K_m)$, m 순서대로 저장을 하며 수신기 ID의 저장여부를 판단할 때에는 저장된 수신기 ID와 현재 수신기 ID를 비트 하나씩 바로 비교하여 같지 않은 경우 빠르게 다음 수신기 ID를 비교 할 수 있도록 설계한다. 또한 메모리의 첫 주소에는 저장된 수신기 ID의 개수를 저장하고 이 개수만큼만 비교하도록 하여 필요이상의 회수로 비교하는 일이 없도록 한다.

[0125] (6) 시스템 재개가능성 메시지 블록

[0126] 시스템 재개가능성 메시지를 저장하기 위해서 규정상 최소 5kB의 용량이 필요하기 때문에 디스플레이포트 송신기 용 HDCP 인증 블록의 내부 메모리의 하위 5kB 부분을 시스템 재개가능성 메시지 블록 영역으로 설정한다. 또한 수신기 정보 블록과 마찬가지로 revocation list에 수신기 ID가 포함되어 있는지 판단할 때에는 저장된 수신기 ID와 현재 수신기 ID를 비트 하나씩 바로 비교하여 같지 않은 경우 빠르게 다음 수신기 ID를 비교 할 수 있도록 한다.

[0127] 암호 블록 설계

[0128] HDCP의 암호 블록은 암호화 제어부와 암호화 구조체 두 개로 나누어 구성되고 디스플레이포트 송신기의 메인 링크(Main Link)단에서 SR inserter 뒤, Scrambler 전에 위치하여 scramble전에 암호화 되도록 한다. 수신기의 경우 Descrambler 뒤, stream unpacker 전에 위치한다.

[0129] (1) 암호화 제어부

[0130] 인증 블록에서 세션 키 교환 단계를 완료하면 Encryption Enable signal을 디스플레이포트의 SR inserter에 보내고 SR inserter는 이것을 감지하여 SR 대신 CPSR을 생성한다. HDCP의 암호화 제어부는 메인 링크의 컨트롤 심볼을 읽어서 CPSR인지 아닌지를 판단하고 만약 CPSR이면 Cipher Enable signal을 1로 세팅한다. Cipher Enable signal이 1인 상태에서는 링크클럭 카운터가 동작해서 링크클럭을 카운트한다. 레인에 따라 정해진 숫자만큼(1-lane인 경우 15, 2-lane인 경우 7, 4-lane인 경우 3) 카운터 값이 올라가면 다시 0으로 초기화되고 이 카운터 값은 암호화 구조체에 들어가서 AES 결과를 저장하는 클럭을 생성한다. 또한 컨트롤 심볼에서 SR이 있으면 Cipher Enable signal을 0으로 세팅하고 링크클럭 카운터는 초기화되지 않은 상태에서 카운트를 중지한다. 인증 블록으로부터 Reauthentication signal이 들어오면 암호 블록을 모두 초기화한다. 또한 컨트롤 심볼은 암호화하면 안 되기 때문에 데이터 출력부분에 MUX가 존재하여 XOR 연산된 데이터와 원본 데이터 둘 중의 하나가 나갈 수 있도록 한다.

[0131] (2) 암호화 구조체

[0132] 도 20은 암호화 구조체의 구성을 도시한 것이다. AES128 블록의 입력에 Counter와 출력에 256bit FIFO가 연결된 형태로 메인링크의 클럭에 따라 암호 데이터열을 생성하고 이 데이터열을 메인링크 데이터와 XOR 연산을 하는 동작을 한다.

[0133] 인증 블록으로부터 Encryption Enable signal이 들어오면 암호화 구조체는 Cipher Enable 상태가 되고 링크클럭을 분주하여 AES 결과를 저장하는 클럭을 생성한다. 이 결과는 256bits 길이의 FIFO에 저장하고 이 FIFO가 다 차면 암호화 제어부로부터 읽는 링크클럭 카운터 값으로 AES 결과를 저장하는 클럭을 생성한다. 이렇게 같은 속도를 가지는 두 개의 클럭을 이용하는 이유는 인증이 끝나고 데이터를 암호화하기 전에 여분의 암호 데이터열을 생성하기 위해서이다. 즉 링크클럭 카운터가 동작하기 전에 미리 inputCTR = 0과 inputCTR = 1에 해당하는 암호 데이터열을 생성하고 FIFO에 저장한다.

[0134] 암호 데이터열은 저장될 때는 128bits가 저장되고 사용될 때는 8bits씩 사용되기 때문에 FIFO 쓰기 클럭과 읽기 클럭이 다르게 된다. 따라서 암호 데이터열을 미리 생성하지 않으면 새로운 암호 데이터열과 링크 데이터를 XOR

연산하기 전에 연산할 암호 데이터열이 제대로 저장되지 않는 이슈가 발생한다. 이런 상황을 방지하기 위해 미리 2개의 암호 데이터열을 생성하는 방법을 사용한다.

[0135] (3) 데이터 완전성 검사 블록

[0136] 디스플레이포트 수신기 용 HDCP 암호 블록은 데이터 완전성 검사를 해야 하는데 이것은 하나의 상태 머신으로 동작이 가능하다. 상태 머신의 동작은 도 21과 같으며 VB-ID의 bit 5가 현재 링크 검증 패턴과 같은지 판단하고 그렇지 않으면 패턴 오류 카운트를 한다. 두 번의 패턴이 지나갈 때 동안 모두 패턴 오류가 되면 에러 디렉트 상태가 되고 다음 프레임에서도 같은 검사를 진행한다. 2번째 프레임에서도 위와 같은 패턴 오류가 발생하면 완전성 오류로 판단하고 인증 블록에 재인증 요청 신호를 보내고 암호화 비사용 상태가 된다. 인증 블록은 재인증 요청 신호를 받으면 DPCD 메모리의 RxStatus 값을 변경하고 CP IRQ 인터럽트를 발생시킨다. 만약 데이터 완전성 검사 도중 SR 데이터가 들어오면 바로 암호화 비사용 상태가 된다.

[0137] MCU 및 직렬통신 블록 설계

[0138] 도 22는 8051이 포함된 디스플레이포트 송신용 HDCP 구조를 도시한 것이고, 도 23은 8051이 포함된 디스플레이포트 수신용 HDCP 구조를 도시한 것이다.

[0139] 인증 블록은 암호 알고리즘 블록(AES128, HMAC-SHA256, RSA)와 난수 발생기, 메시지 인터페이스, 제어부, 내부 메모리(송신기)로 이루어져있다. 암호 블록은 암호화 제어부와 암호화 구조체로 이루어져있다. 메시지 인터페이스는 인증과정에 따라 인증 블록에서 생성되는 난수, 암호 키 등을 AUX 메시지 형태로 만들어 디스플레이포트 AUX로 전달한다. RSA, AES128, HMAC-SHA256은 각 암호 알고리즘에 맞추어 입력과 암호 키를 이용해 암호화된 출력을 생성한다. 난수 발생기는 Oscillator에서 발생하는 noise를 AES128로 암호화한 값을 출력한다. 암호화 제어부는 메인링크의 컨트롤 심볼을 읽어서 암호화 구조체의 동작을 제어한다.

[0140] HDCP 인증 블록은 상태 머신으로 동작하며 인증 블록 내부에 상태 머신 블록을 설계하여 다른 세부 블록을 제어하는 형태로 설계할 수 있다. 이때 상태 머신 블록을 외부 MCU로 대체할 수 있다. 즉 상태 머신 블록에서 제어 신호가 나오는 것이 아니라 외부 MCU에서 제어 신호가 나오며 HDCP에서는 제어부 인터페이스가 존재하여 이 제어 신호를 받는 형태로 설계할 수 있다.

[0141] 이렇게 설계하면 첫째, 인증 블록의 동작을 변경하는 데에 용이하다. 상태 머신 블록으로 제어하는 경우 인증 블록의 제어동작이 고정되기 때문에 후에 프로토콜이 바뀌었을 경우 동작을 변경하기가 어렵다. 하지만 MCU에서 제어를 하게 되면 MCU의 프로그램코드를 변경함으로써 손쉽게 동작을 변경할 수 있다. 둘째, 설계 시 동작 설계가 용이해진다. 상태 머신 블록으로 설계하고 동작을 확인하기 위해서는 매번 합성하고 FPGA 보드 테스트를 해야 한다. MCU로 제어하는 경우 MCU와 HDCP를 같이 합성해서 FPGA 보드에 올린 뒤 MCU의 프로그램코드만 별도로 업로드 하기 때문에 실시간으로 동작을 설계하고 확인할 수 있다. 셋째, 실제 디스플레이포트의 경우 다른 디지털 인터페이스와 함께 제어되는 경우가 많기 때문에 실제로 MCU로 인해 제어된다. HDCP의 설계 시에도 제어부를 MCU로 설계함으로써 실제 디스플레이포트 환경에 최적화하기가 용이하다. MCU는 공개소스인 8051 MCU 소스를 변경하였으며 MCU에 PC to MCU 블록이 포함된 구조로 설계한다. 이 MCU가 디스플레이포트의 AUX 블록에 연결되는 구조로 전체 시스템을 구현한다.

[0142] (1) 송신기용 HDCP 제어

[0143] 디스플레이 송신기 용 HDCP 인증 블록은 블록의 상태를 MCU로 출력하고 MCU로부터 제어신호를 받는다. 따라서 제어부 인터페이스를 MCU의 입출력 핀에 연결하여 각 신호를 주고받을 수 있도록 한다. MCU는 8bit의 입력 핀과 8bit의 출력 핀을 가진다. MCU의 제어 동작은 도 24와 같다.

[0144] (2) 수신기용 HDCP 제어

[0145] 디스플레이 수신기 용 HDCP 인증 블록은 송신기 용 인증 블록과 마찬가지로 블록의 상태를 MCU로 출력하고 MCU로부터 제어신호를 받는다. 따라서 송신기 용 인증 블록과 마찬가지로 제어부 인터페이스를 MCU의 입출력 핀에 연결하여 각 신호를 주고받을 수 있도록 한다. MCU는 8bit의 입력 핀과 8bit의 출력 핀을 가진다. MCU의 제어 동작은 도 25와 같다.

[0146] (3) PC to MCU 직렬통신 블록

[0147] 제안하는 HDCP에는 8051 MCU가 디스플레이포트 인터페이스 외부에서 연결되며 8051 MCU는 외부메모리에 저장된 프로그램코드에 의해 동작한다. 인증 블록내의 제어부 인터페이스가 8051 MCU의 PIN으로 연결되어 있으며 인증

블록의 상태를 상태 신호로 만들어 8051 PIN IN으로 입력하고 8051 PIN OUT으로부터 제어 신호를 받아서 인증 블록 내 세부 블록들을 제어하는 신호를 생성한다. UART통신 블록이 존재하여 PC와 UART통신을 통해 프로그램코드를 받으며 이 것을 8x65536 메모리에 저장한다. 8x65536 메모리에 저장된 값을 MCU가 프로그램코드로 인식하여 그에 따라 동작한다.

[0148] FPGA에 합성된 MCU 블록에 프로그램코드를 업로드 하기 위해서는 MCU 블록의 ROM에 프로그램코드를 써야 하기 때문에 PC로부터 FPGA로 프로그램코드에 해당하는 데이터가 전송되어야 한다. 또한 전송된 데이터를 MCU의 ROM에 저장하는 별도의 블록이 필요하다.

[0149] 프로그램코드의 작성은 노트패드와 무료 컴파일러인 SDCC를 사용한다. SDCC는 프리웨어로써, 간단한 리포트를 제공하고 c파일을 hex파일로 컴파일 해주고 hex파일은 ihx파일로 변환해주는 기능을 한다. OpenCore에서 제공하는 Rom_Maker 프로그램으로 ihx파일을 OP코드로 변환하여 프로그램코드로 사용한다.

[0150] PC로부터 FPGA로 프로그램코드를 전송하기 위해서 FPGA의 UART 통신 모듈을 사용한다. 또 별도의 메모리를 생성하고 UART 통신으로 받은 데이터를 자동으로 이 메모리에 저장하는, PC to MCU 직렬통신 블록을 설계하여 전송된 데이터를 저장하도록 한다. 이 메모리를 MCU 블록의 ROM으로 사용하여 결과적으로 PC로부터 MCU로 프로그램코드를 업로드 할 수 있도록 한다. 이때 PC to MCU 직렬통신 블록의 상태 머신은 도 26과 같다.

[0151] 따라서, 제안된 HDCP는 인증 블록이 상태 머신이 아니라 MCU로 인해 제어되어 동작의 유연성을 높일 수 있다. 또한 이런 구조로 인해 설계 시 새로운 동작을 시험할 때마다 새롭게 디지털 회로를 합성할 필요 없이 실시간으로 프로그램코드를 변경하여 새로운 동작을 시험할 수 있다. 그리고 향후 최신 HDCP 표준이 발표될 때 프로그램코드 변경을 통해 동작을 추가 및 개선할 수 있으며 프로그램 코드를 변경하면 디스플레이포트 외에 HDMI와 같은 디스플레이 인터페이스와도 호환할 수 있을 것이다. 또한 제안된 암호화 구조체는 메인링크 데이터와 암호 데이터열이 서로 다른 클럭으로 생성됨으로 인해 XOR 연산 시 연산 딜레이로 인해 올바른 암호 데이터열이 메인링크 데이터와 연산되지 않는 문제를 해결하여 고속의 메인링크에서 안정적으로 데이터를 암호화 할 수 있다.

[0152] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0153] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0154] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 이때, 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수 개의 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 어플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어

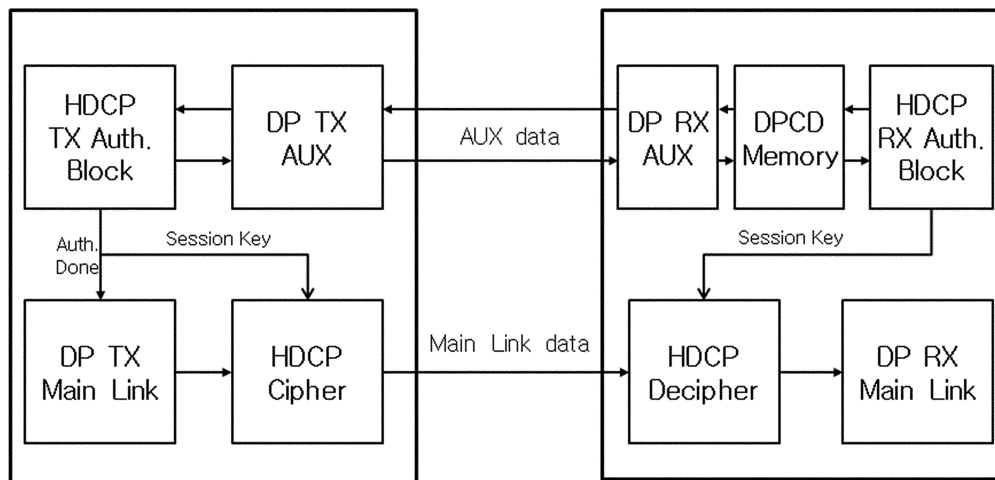
를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다.

[0155] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

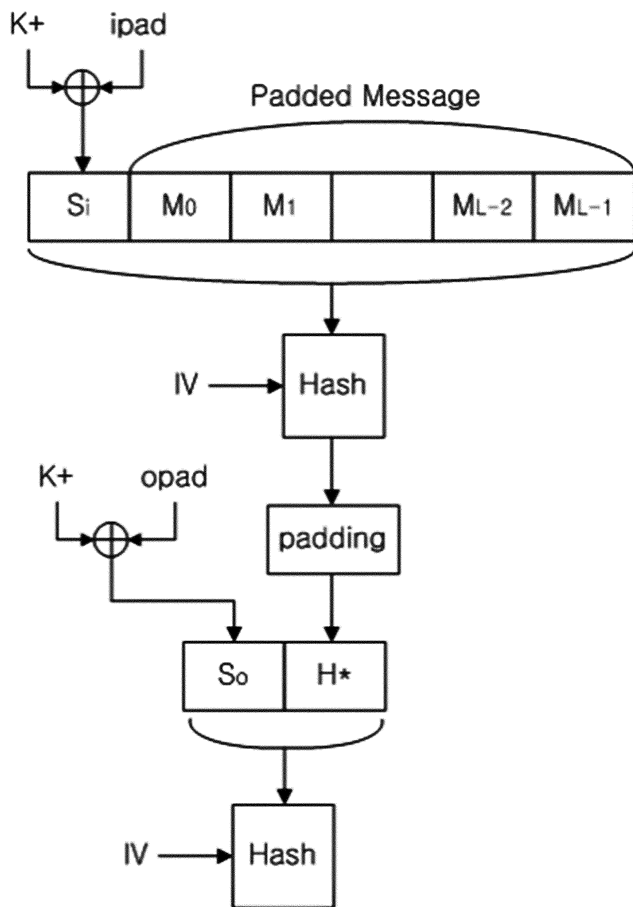
[0156] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

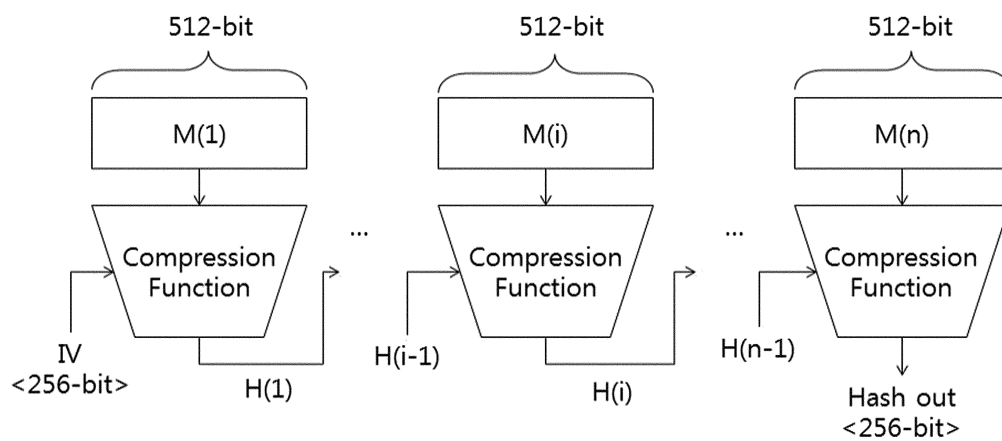
도면1



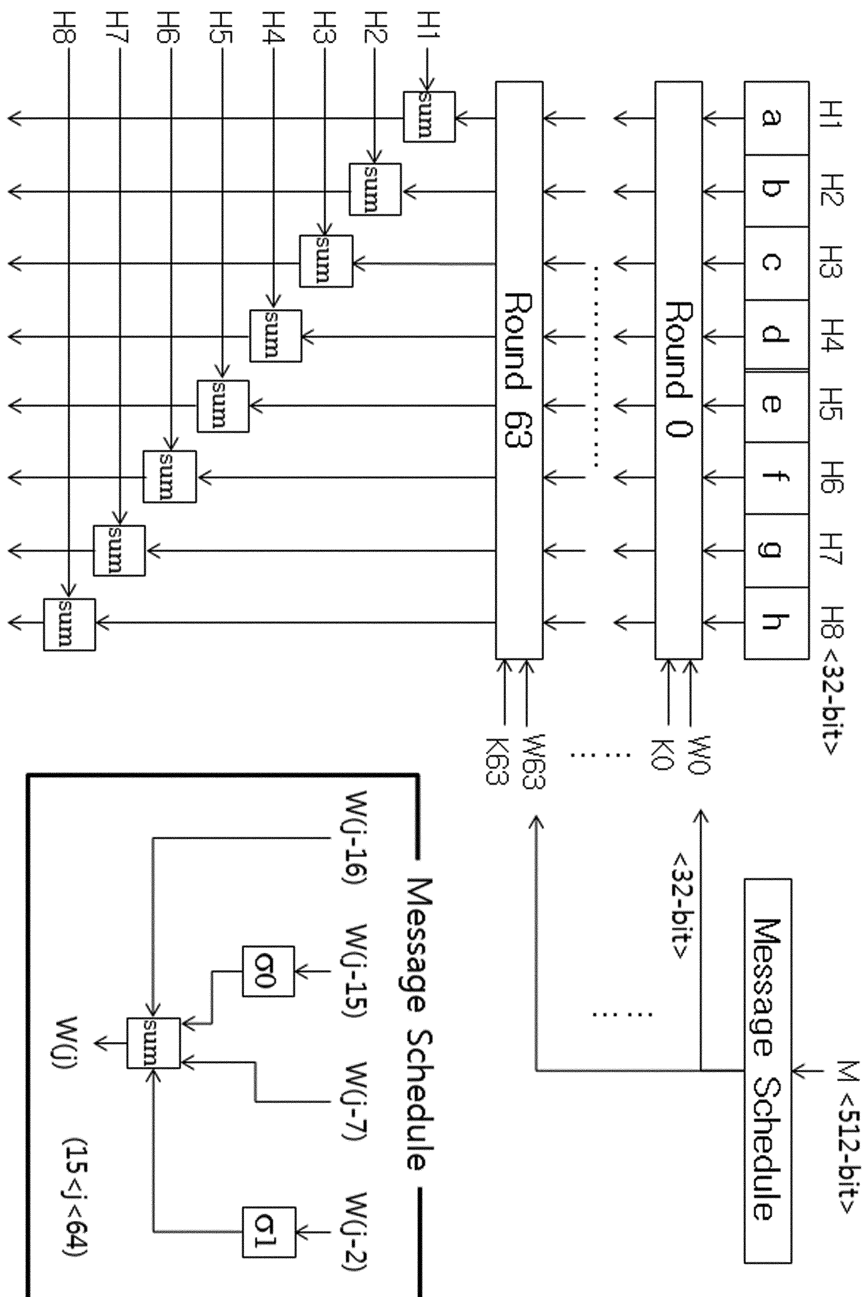
도면2



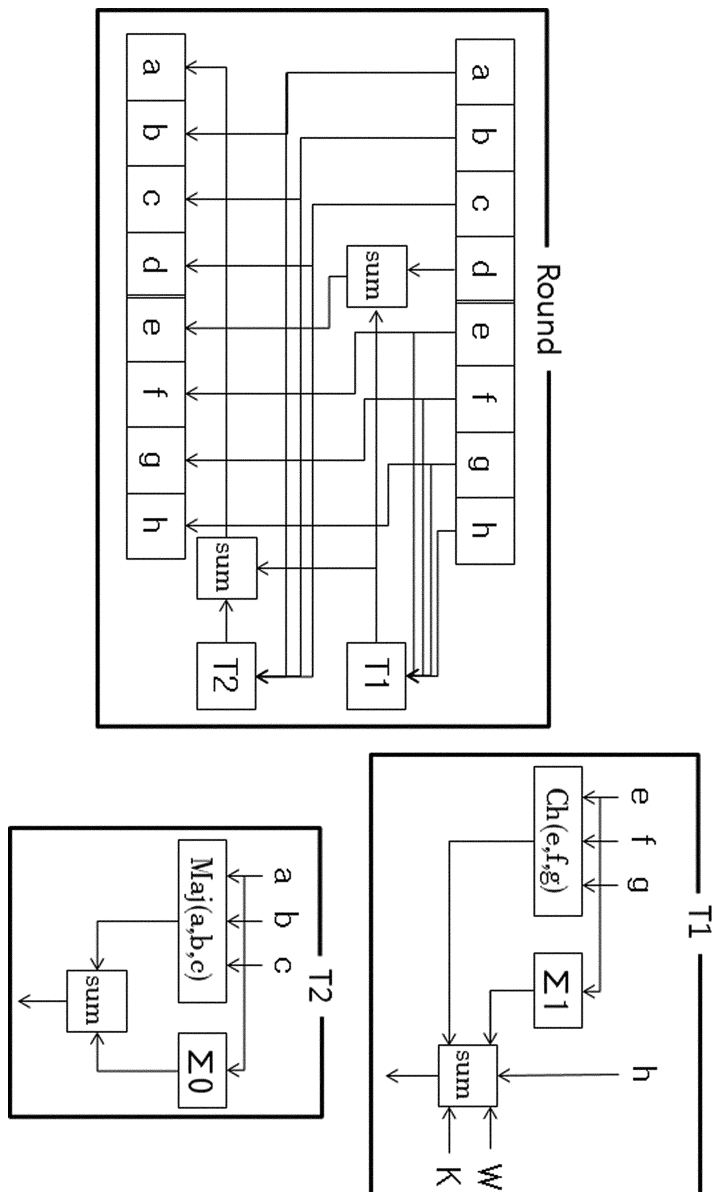
도면3



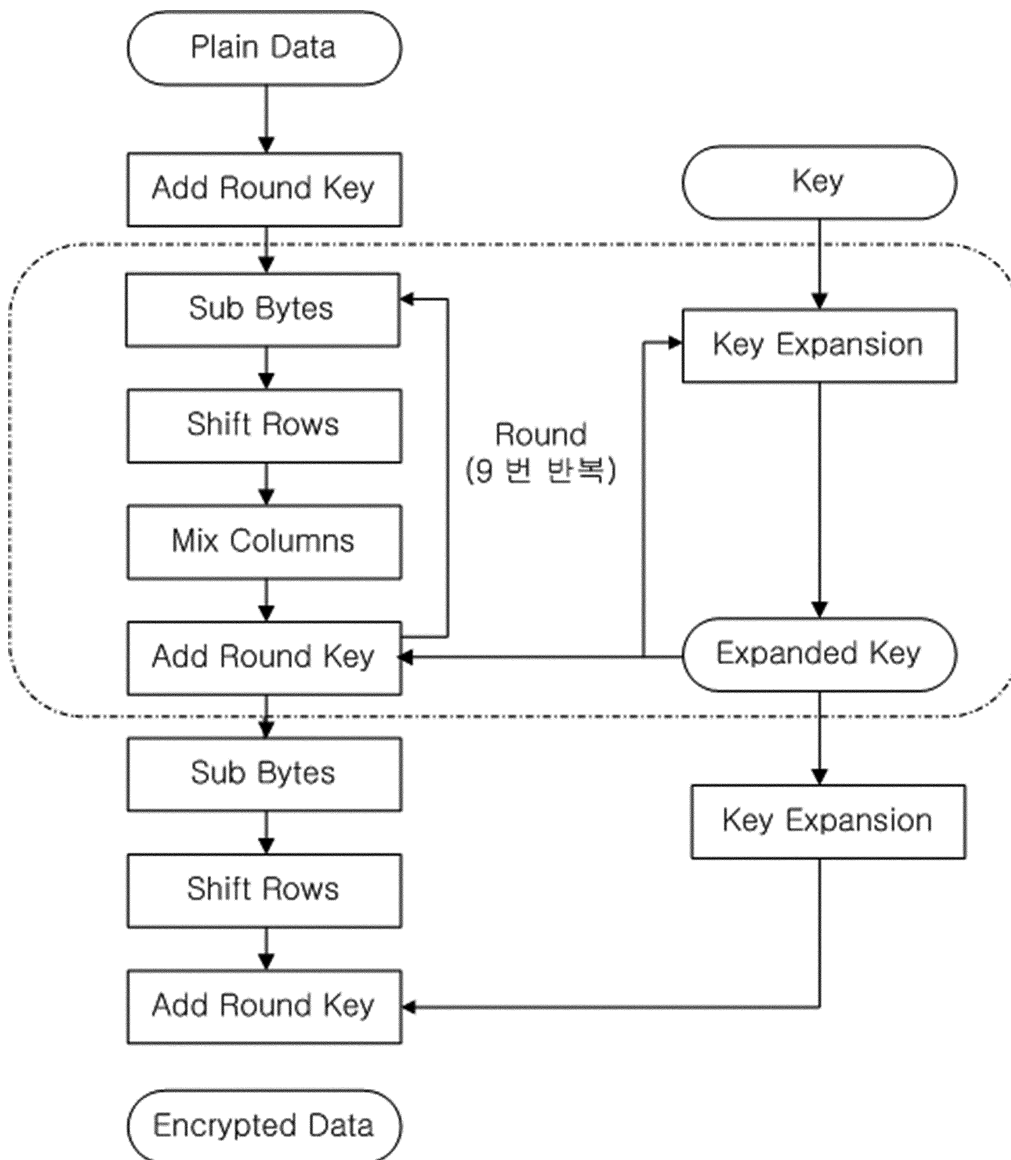
도면4



도면5



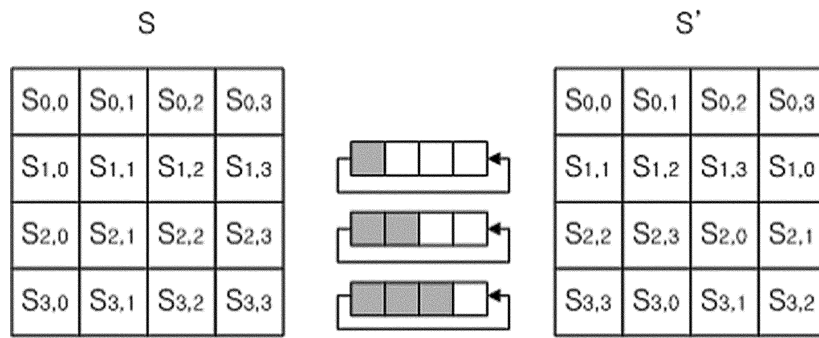
도면6



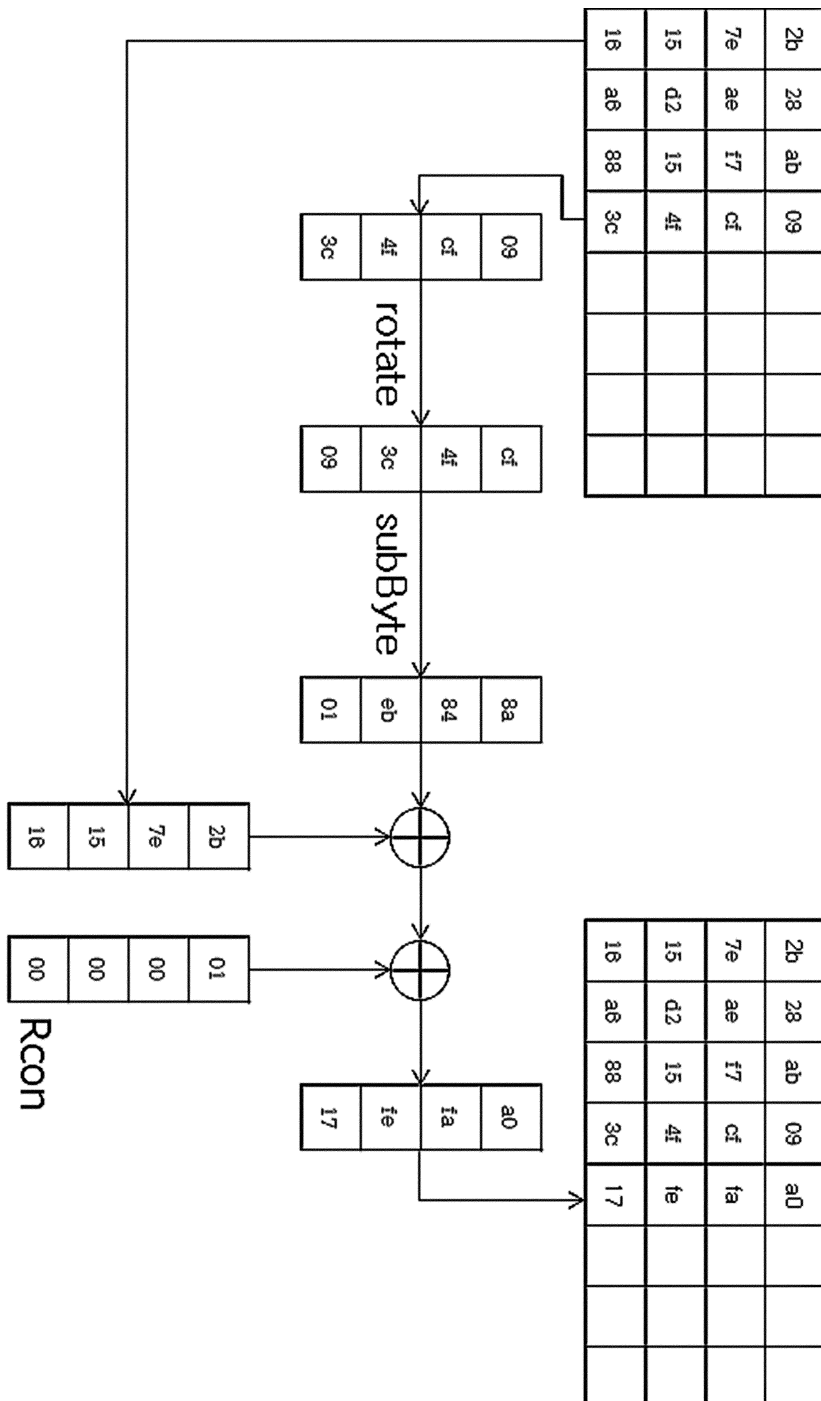
도면7

0x XY		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	84	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

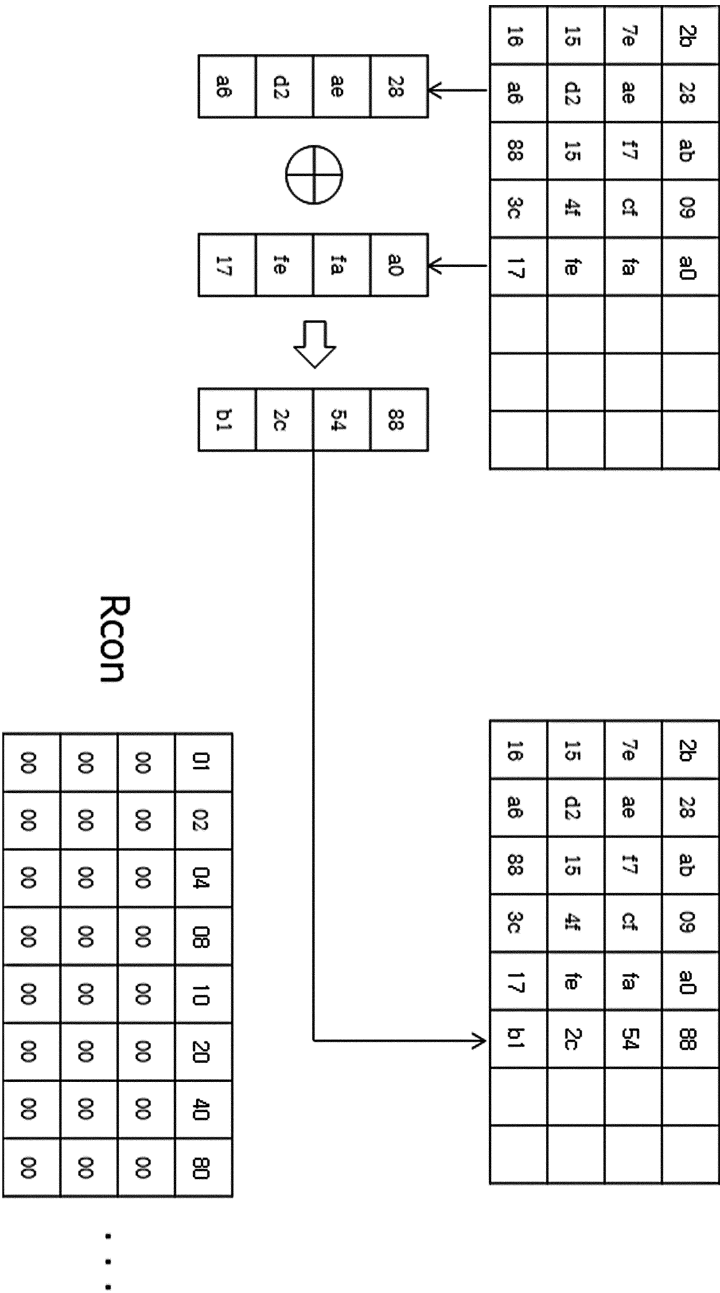
도면8



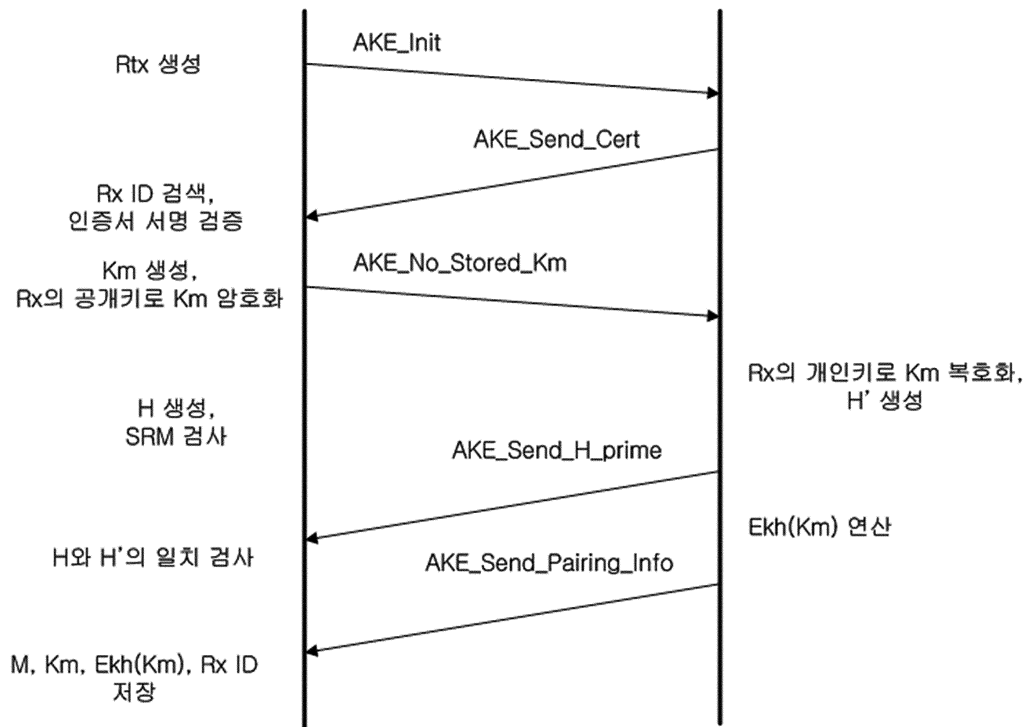
도면9



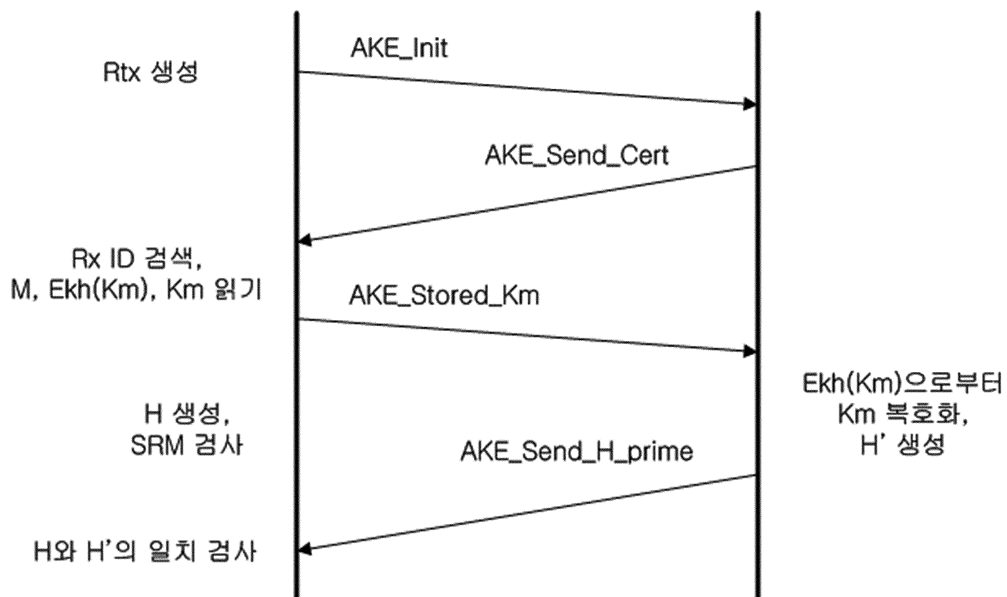
도면10



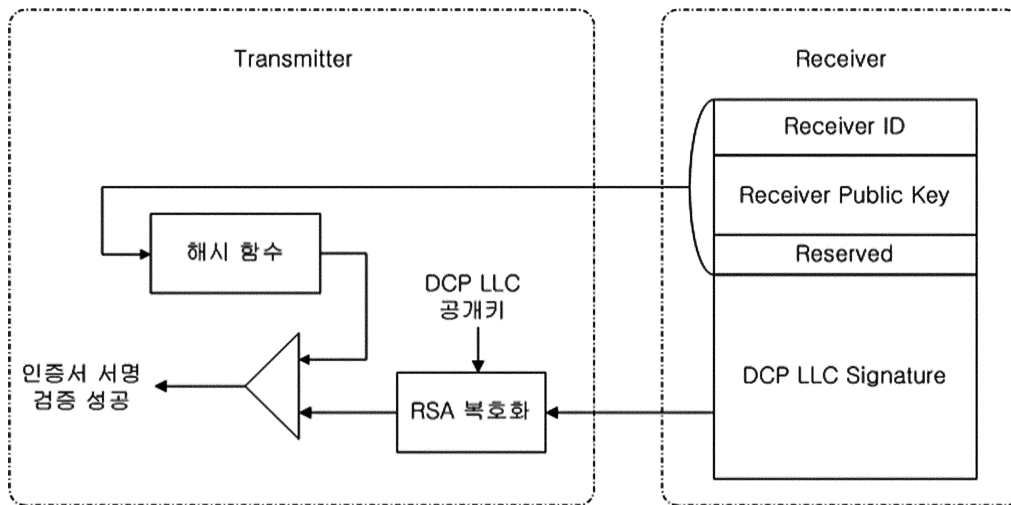
도면11



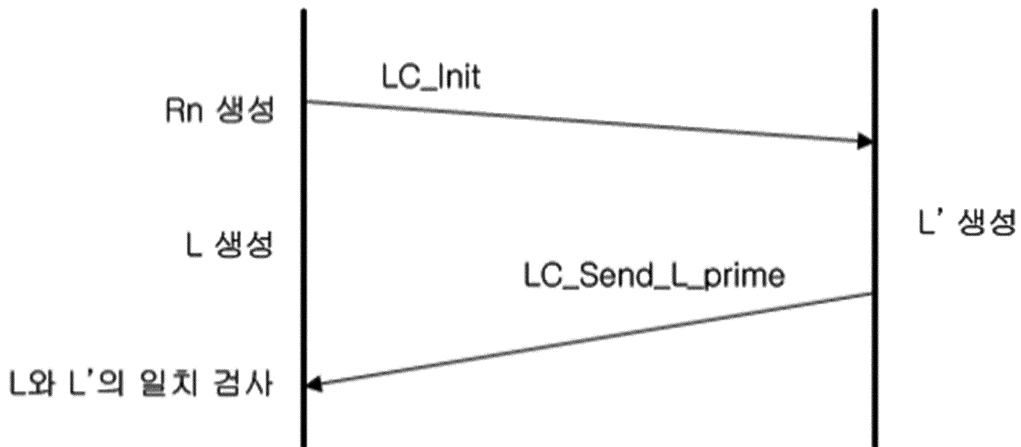
도면12



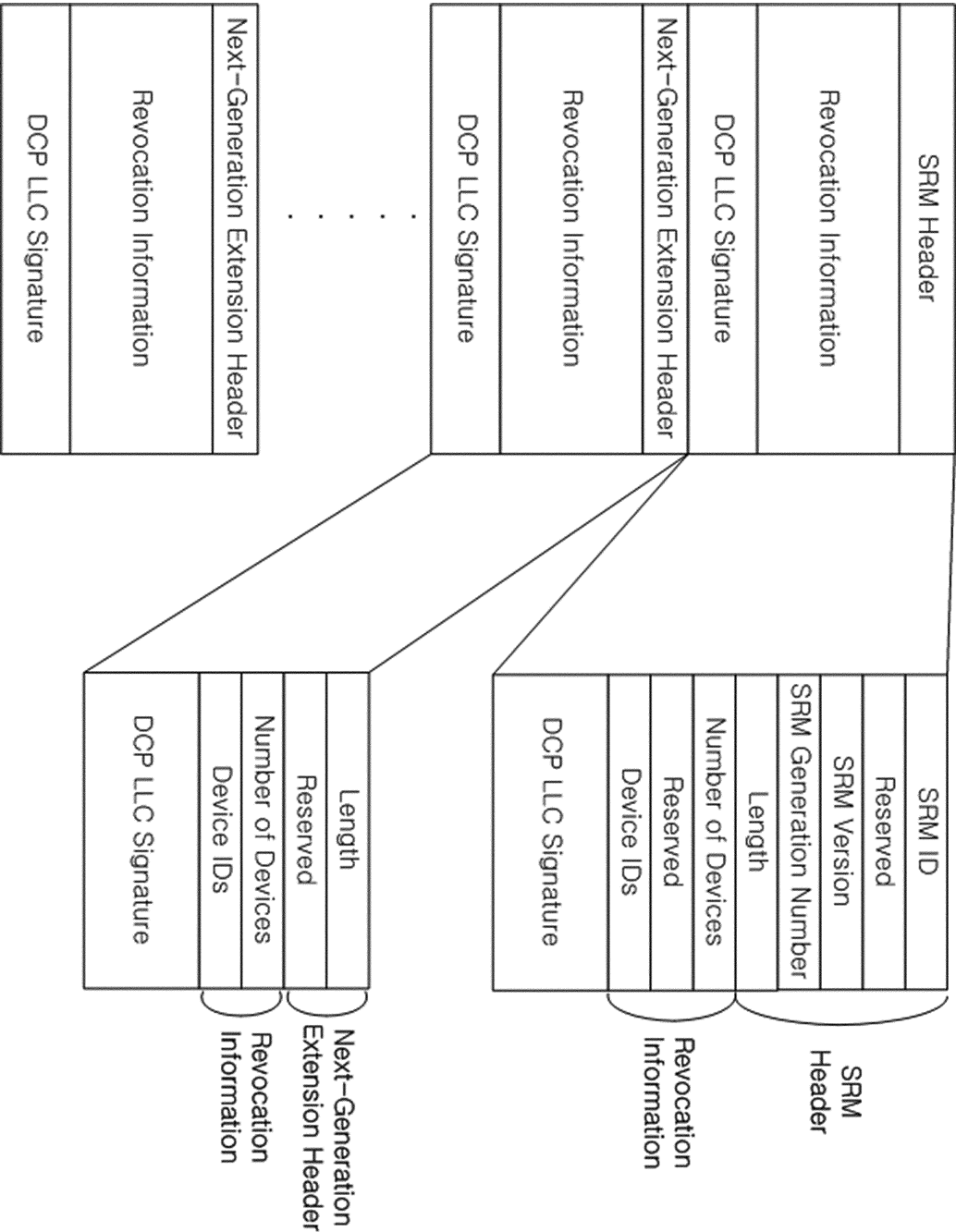
도면13



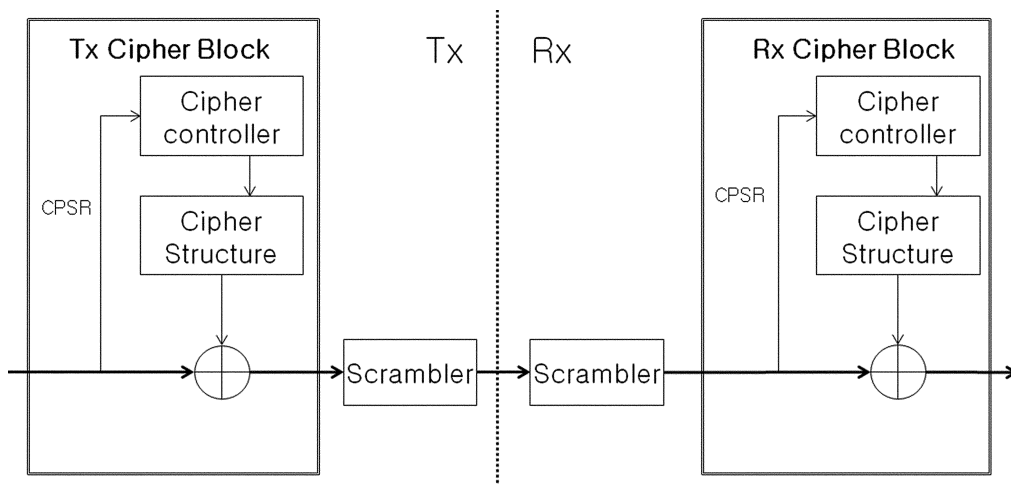
도면14



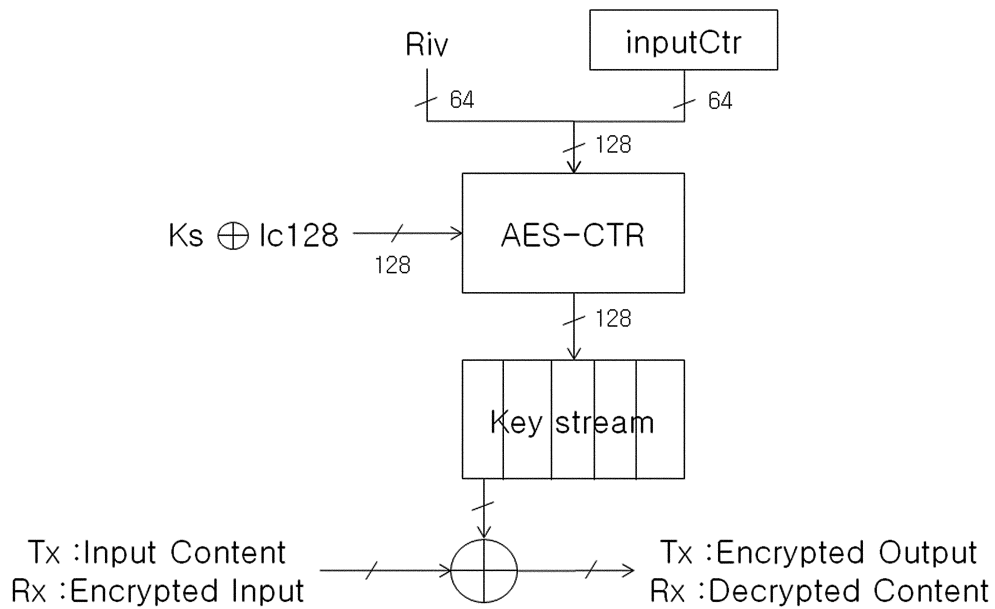
도면15



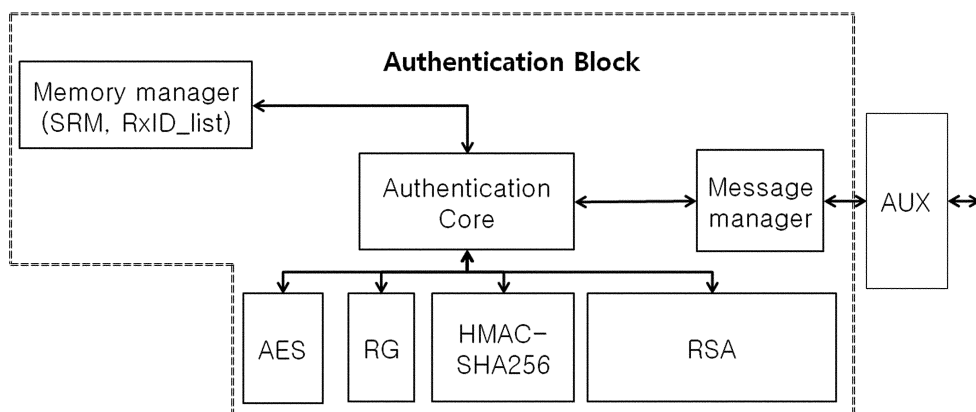
도면16



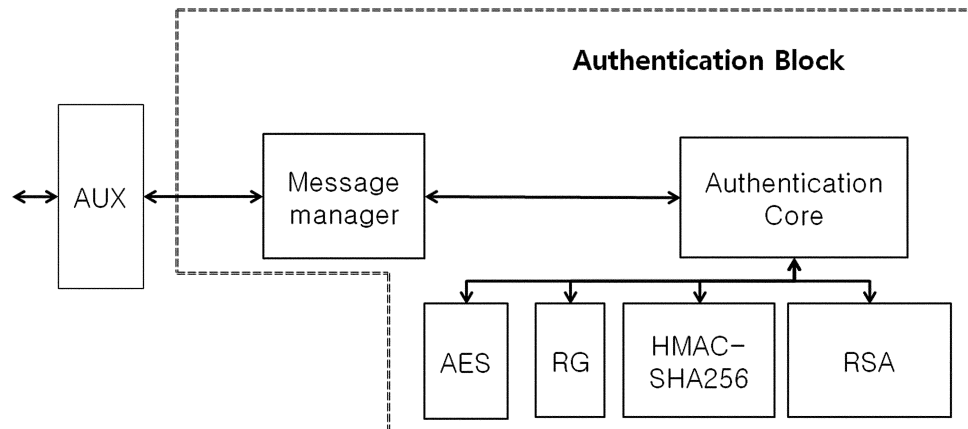
도면17



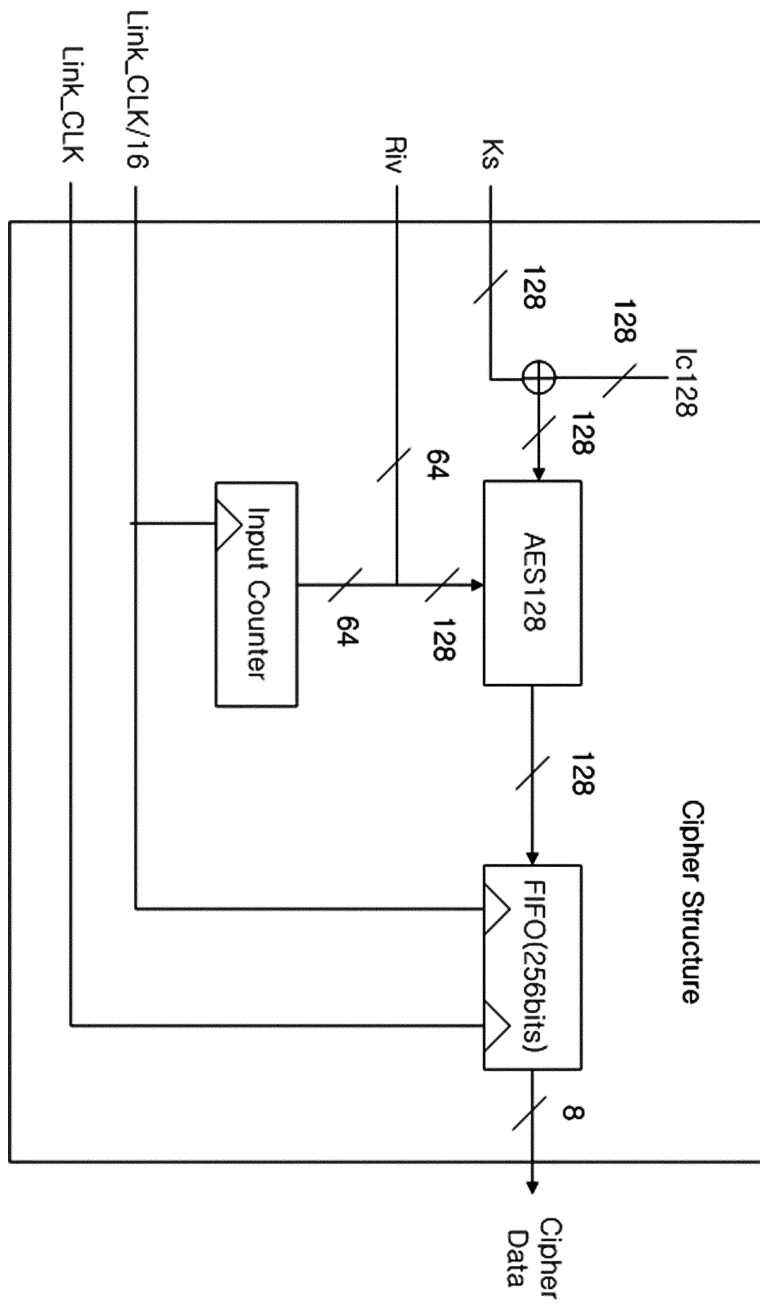
도면18



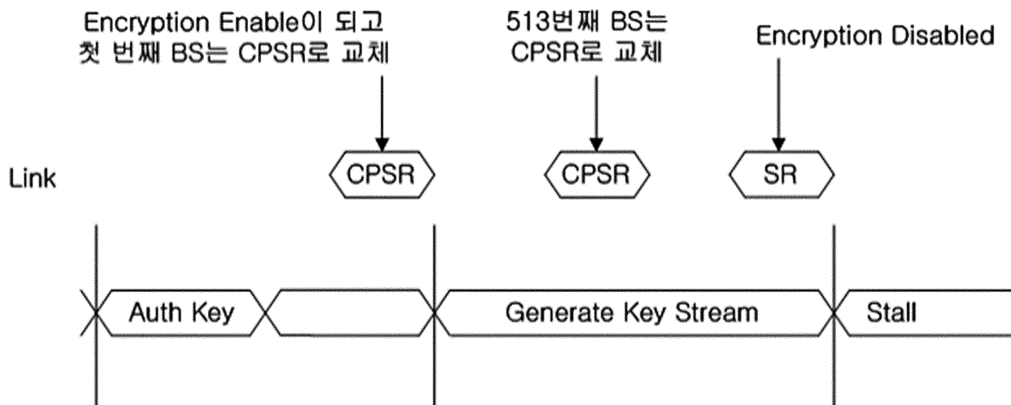
도면19



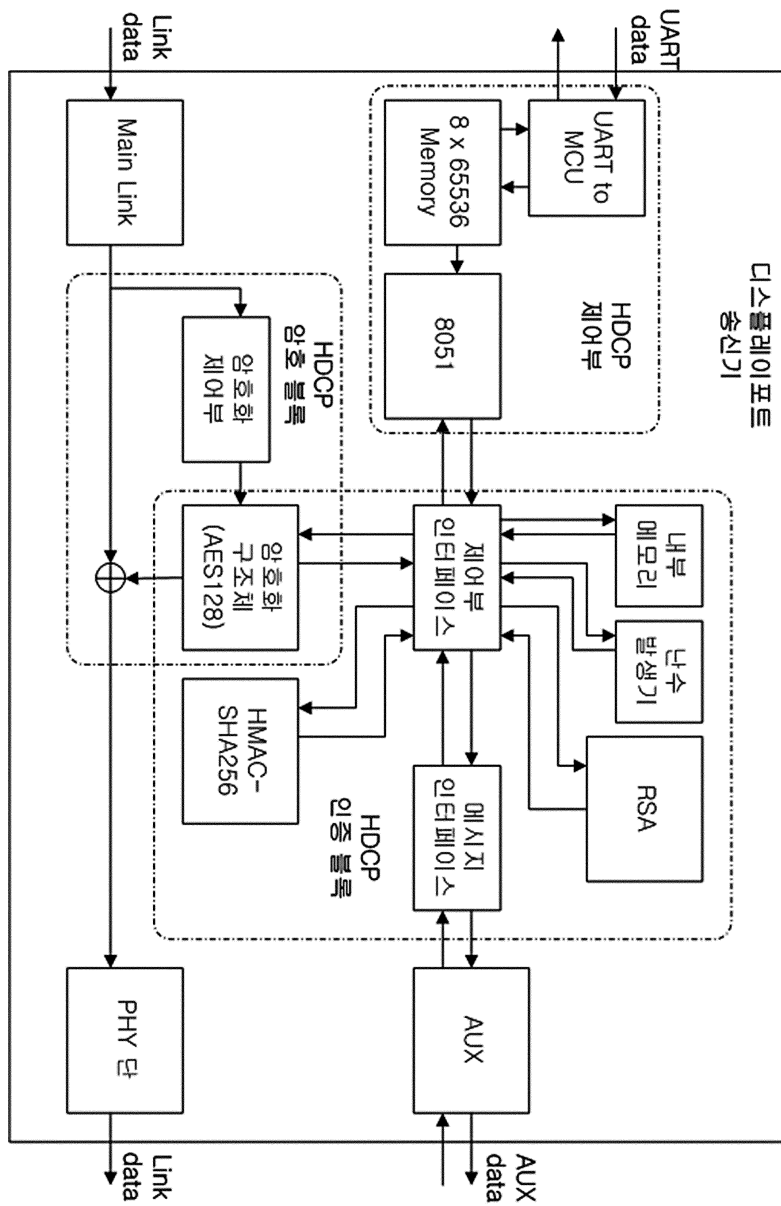
도면20



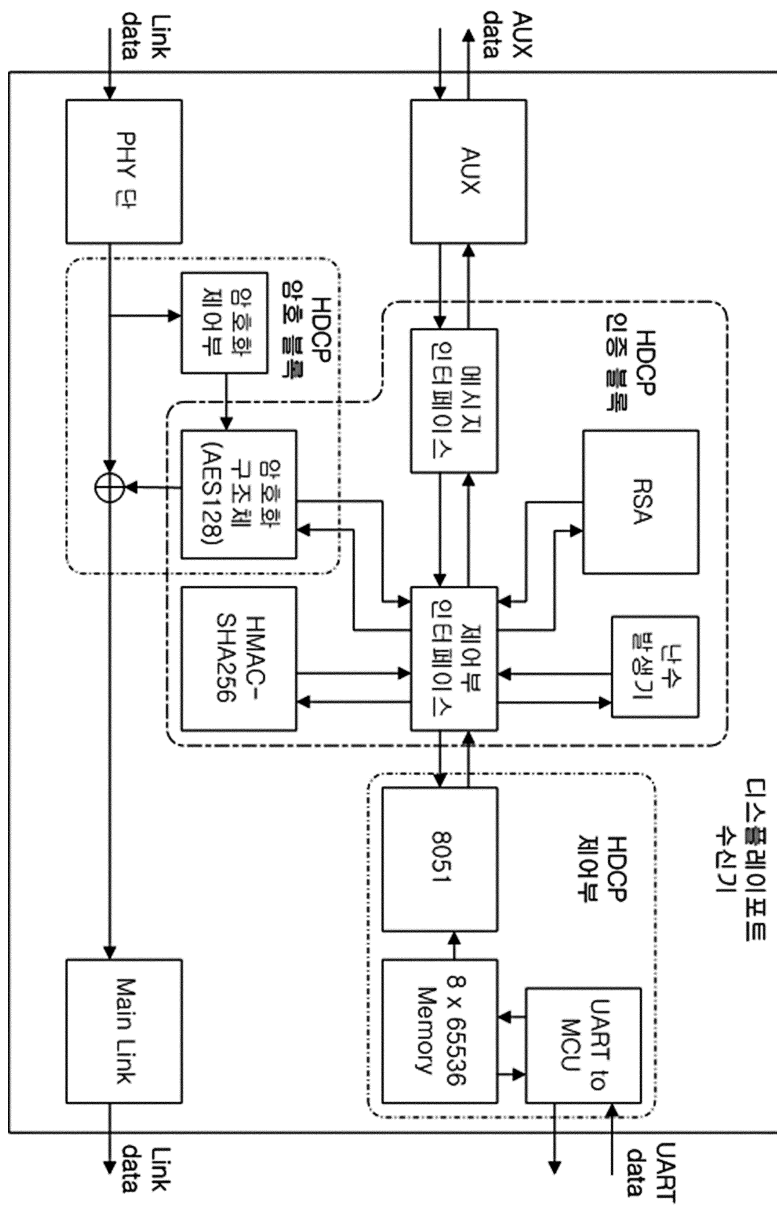
도면21



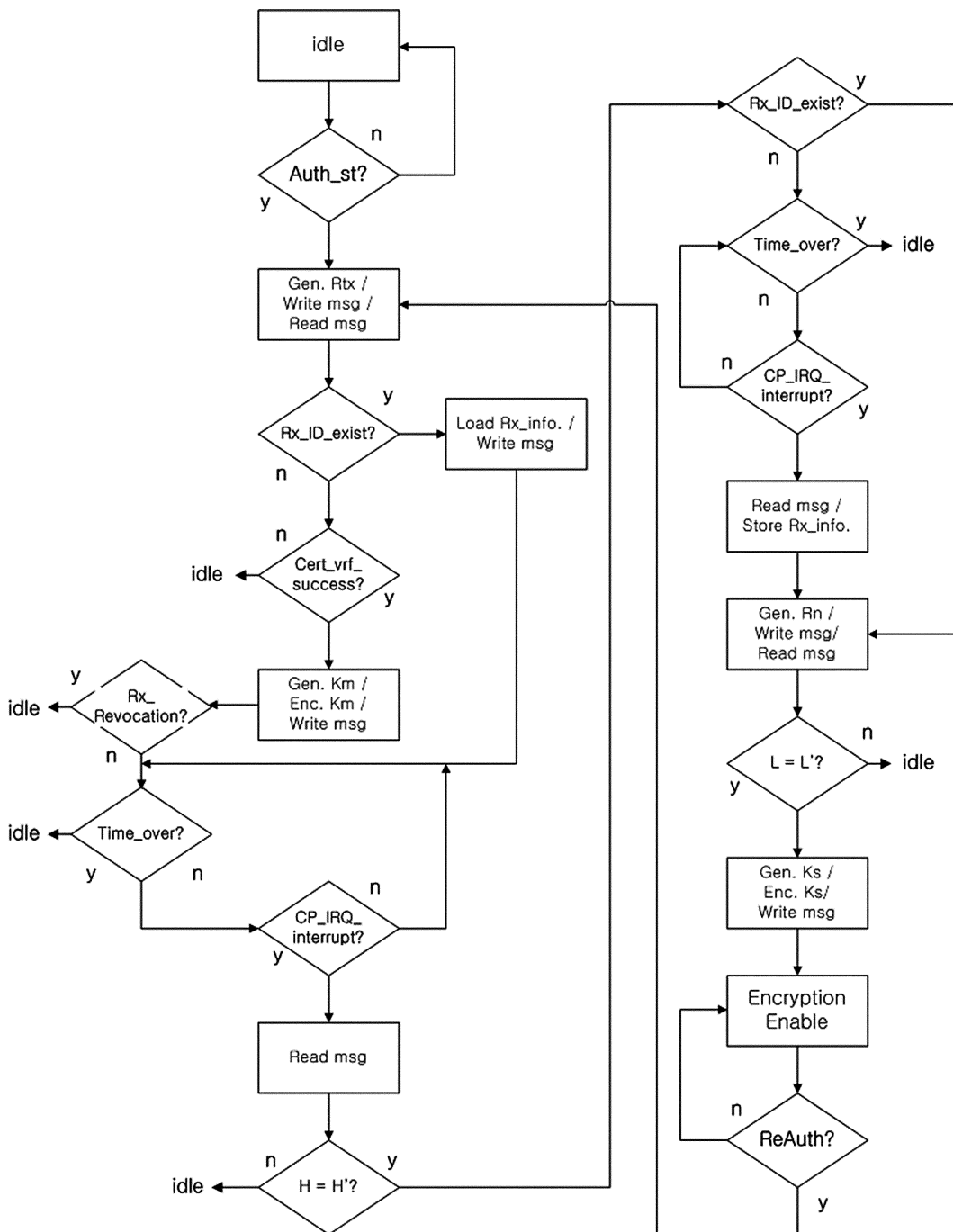
도면22



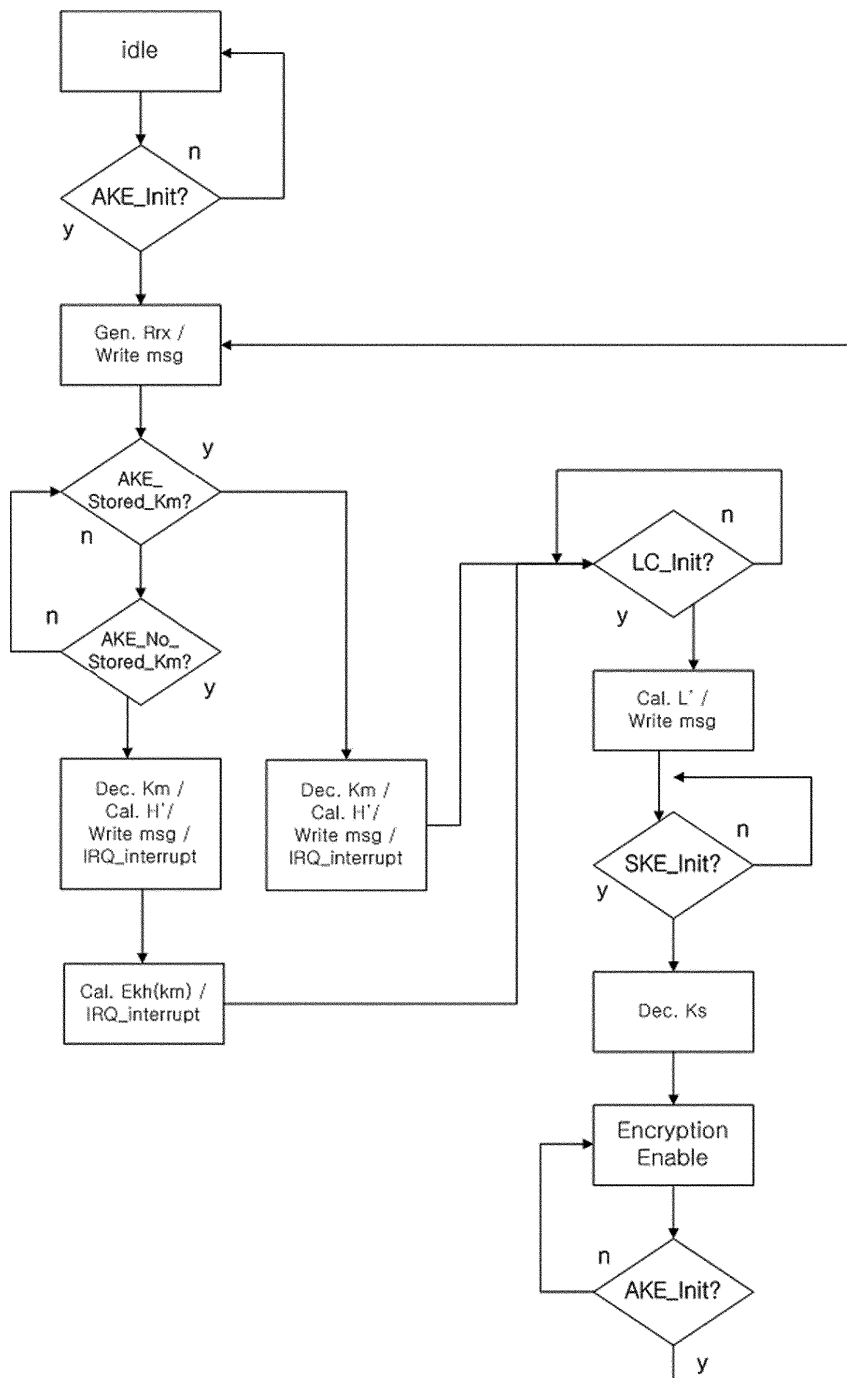
도면23



도면24



도면25



도면26

