(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년10월30일
(11) 등록번호 10-2172181
(24) 등록일자 2020년10월26일

(51) 국제특허분류(Int. Cl.)
H04L 9/06 (2006.01) *H04L 9/32* (2006.01)
(52) CPC특허분류
H04L 9/0618 (2013.01)
H04L 9/0643 (2013.01)
(21) 출원번호 10-2018-0145919
(22) 출원일자 2018년11월23일
심사청구일자 2018년11월23일
(65) 공개번호 10-2020-0067265
(43) 공개일자 2020년06월12일
(30) 우선권주장
62/728,078 2018년09월07일 미국(US)
(56) 선행기술조사문헌
KR101683051 B1*
KR1020130042440 A*
KR1020180007974 A*
Youtao Zhang, SENSS: Security Enhancement to Symmetric Shared Memory Multiprocessors, Proc. of HPCA 2005 (2005.)*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
김황남
서울특별시 서초구 서운로 104, 205동 503호
오승환
성남시 분당구 양현로 138, 809동 1602호
박성준
서울특별시 강남구 삼성로4길 17, 502동 1203호
(74) 대리인
심경식, 홍성욱

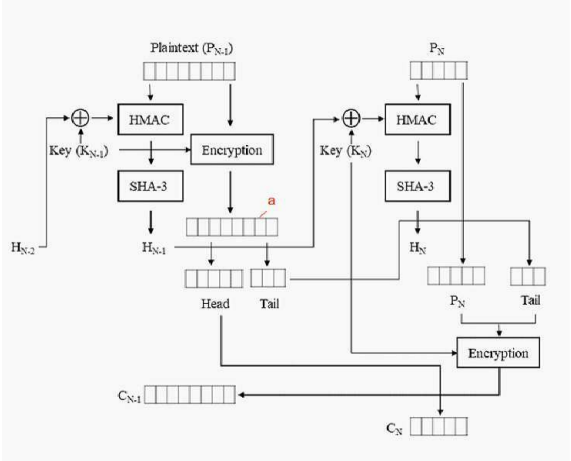
전체 청구항 수 : 총 8 항 심사관 : 양종필

(54) 발명의 명칭 **실시간 데이터 전송을 위한 블록 암호 장치 및 방법**

(57) 요 약

본 발명은 실시간 데이터 전송을 위한 블록 암호 장치 및 방법에 관한 것으로, 본 발명의 일 실시예에 따른 블록 암호 장치는, 서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각 평문(plaintext) 블록을 암호화하여 암호문 블록을 생성하는 블록 암호화부, 상기 블록 암호화부에서 암호화되는 현재 평문 블록의 암호화시 선택된 키 및 상기 현재 평문 블록 이전의 평문 블록에 의해 생성된 이전 메시지 인증 코드를 이용하여 메시지 인증 코드를 생성하는 메시지 인증부를 포함한다.

대 표 도 - 도3



(52) CPC특허분류
H04L 9/3242 (2013.01)
H04L 2209/20 (2013.01)
H04L 2209/38 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 NRF-2016M1B3A1A01937599
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 무인이동체 미래선도 핵심기술개발사업
연구과제명 다중 무인 이동체 기반 사이버-물리 융합공격 대응 기술 및 자율복원 통신/보안기술 개발
기 여 율 1/1
과제수행기관명 고려대학교 산학협력단
연구기간 2016.09.21 ~ 2019.03.31

명 세 서

청구범위

청구항 1

서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각 평문(plaintext) 블록을 암호화하여 암호문 블록을 생성하는 블록 암호화부; 및

상기 블록 암호화부에서 암호화되는 현재 평문 블록의 암호화시 선택된 키 및 상기 현재 평문 블록 이전의 평문 블록에 의해 생성된 이전 메시지 인증 코드를 이용하여 메시지 인증 코드를 생성하는 메시지 인증부; 및

하나의 패스워드를 해시 함수에 입력하여 얻은 제1 결과값 또는 서로 다른 종류의 해시 함수를 하나의 패스워드에 입력하여 얻은 제2 결과값에 기반하여 각 평문 블록을 암호화하기 위해 사용되는 서로 다른 길이의 키를 생성하고, 상기 생성된 키들의 순서를 정하여 패턴을 설정하는 키 생성부를 포함하고,

상기 암호문 블록이 기 설정된 크기 이상인 경우, 상기 암호문 블록을 기 설정된 크기에 해당하는 헤더(header)와 나머지를 테일(tail)로 나누고, 상기 헤더를 암호문 블록으로 생성하며, 상기 테일을 다음 순서의 평문 블록과 합하여 블록 암호화를 수행하고, 오류 확산 없는 실시간 전송을 위해 암호화 스틸링(stealing)을 이용해 추가적인 패딩 작업을 수행하지 않는 것을 특징으로 하는, 블록 암호 장치.

청구항 2

제1항에 있어서,

평문 메시지를 서로 다른 비트 길이를 갖는 복수의 순서화된 평문 블록으로 나누는 입력 처리부를 더 포함하는, 블록 암호 장치.

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 메시지 인증부는,

상기 선택된 키와 상기 이전 메시지 인증 코드를 배타적 논리합(XOR) 연산하는 배타적 논리합 연산모듈;

상기 현재 평문 블록과 상기 배타적 논리합 연산모듈에서 연산된 값을 제1 해시 알고리즘으로 암호화하는 제1 해시모듈; 및

상기 제1 해시모듈에서 암호화된 값을 제2 해시 알고리즘으로 암호화하여 현재 평문 블록에 대한 메시지 인증 코드를 생성하는 제2 해시모듈을 포함하는 것을 특징으로 하는 블록 암호 장치.

청구항 5

제1항에 있어서,

상기 현재 평문 블록에 대한 암호문 블록과 메시지 인증 코드를 수신장치로 전송하는 통신부를 더 포함하되,

상기 통신부는 암호문 블록마다 카운터값을 함께 전송하는 것을 특징으로 하는 블록 암호 장치.

청구항 6

제1항에 있어서,

상기 키 생성부는 네트워크 환경과 데이터 안정성을 고려하여 상기 패턴을 설정하는 것을 특징으로 하는 블록 암호 장치.

청구항 7

제1항에 있어서,

핸드쉐이크 과정을 통하여 서로 다른 길이를 갖는 키들의 순서가 설정된 패턴, 공개키, 사전 마스터 비밀키 중 적어도 하나를 수신장치와 공유하도록 하는 상호 인증부를 더 포함하는, 블록 암호 장치.

청구항 8

블록 암호 장치가 평문 메시지를 블록 암호하는 방법에 있어서,

하나의 패스워드를 해시 함수에 입력하여 얻은 제1 결과값 또는 서로 다른 종류의 해시 함수를 하나의 패스워드에 입력하여 얻은 제2 결과값에 기반하여 각 평문 블록을 암호화하기 위해 사용되는 서로 다른 키를 생성하고, 생성된 서로 다른 키들의 순서를 정하여 패턴을 설정하는 단계;

핸드쉐이크 과정을 통하여 상기 패턴, 공개키, 사전 마스터 비밀키 중 적어도 하나를 수신장치와 공유하는 단계;

상기 평문 메시지를 복수의 평문 블록으로 나누고, 각 평문 블록에 대해 상기 패턴의 순서에 따라 키를 선택하여 암호문 블록과 메시지 인증 코드를 각각 생성하는 단계;

상기 암호문 블록이 기 설정된 크기 이상인 경우, 상기 암호문 블록을 기 설정된 크기에 해당하는 헤더(header)와 테일(tail)로 나누는 단계;

상기 헤더를 암호문 블록으로 생성하며, 상기 테일을 다음 순서의 평문 블록과 합하여 블록 암호화를 수행하는 단계; 및

오류 확산 없는 실시간 전송을 위해 암호화 스틸링(stealing)을 수행하는 단계를 포함하는, 블록 암호 방법.

청구항 9

제8항에 있어서,

각 평문 블록에 대응하는 암호문 블록과 메시지 인증 코드를 수신장치로 전송하는 단계를 더 포함하는 블록 암호 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 실시간 데이터 전송을 위한 블록 암호 장치 및 방법에 관한 것으로, 더욱 상세하게는 서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각 평문블록을 암호화함으로써, 공격자가 송신자로부터 전송된 암호문을 교체하거나 위조하는 경우를 수신자가 검증가능하게 하는 실시간 데이터 전송을 위한 블록 암호 장치 및 방법에 관한 것이다.

배 경 기 술

[0003] 암호화 기법은 그 자체로서는 큰 의미를 가지지는 못하지만 서로 간의 통신하는 상황에서 주고받는 메시지가 개인 정보를 담고 있는 중요한 데이터라면 데이터가 가지는 정보가 노출되지 않도록 하는 것이 중요하기 때문에 암호화 기법 기술이 필요하다. 이러한 암호화 기법에는 대칭키 암호화 기법과 비대칭키 암호화 기법이 있다. 일반적으로 대칭키 암호화 기법은 구조적인 복잡도에 기반으로 하며 키 길이가 비대칭키 암호화 기법의 키 길이보다 짧기 때문에 빠른 암호화/복호화 수행이 가능하다. 반면에 비대칭키 암호화 기법은 수학적인 복잡도(이산대수 문제)에 기반으로 하며 공개키와 개인키로 나뉘어 암호화와 복호화를 따로 수행한다. 즉, 대칭키 암호화 기법의 경우 하나의 키로 암호화 복호화를 수행하기 때문에 사전에 송수신자가 안전하게 대칭키를 공유해야하는 문제를 가지고 있는 반면 비대칭키 암호화 기법의 경우 공개키와 개인키로 나뉘어 있어서 따로 키를 교환할 필요 없이 송신자는 수신자의 공개키로 중요 메시지를 암호화하여 전송하면 수신자는 자신의 개인키로 암호문을 복호화하여 메시지 정보를 확인할 수 있는 장점을 가지고 있다. 이러한 비/대칭키 암호화 기법의 장단점을 SSL/TLS은 네트워크 통신환경에 잘 활용하였다.

[0004] SSL은 Secure Socket Layer의 약자로 웹 서버와 웹 브라우저 간의 안전한 암호화 통신을 위하여 응용계층과

TCP/IP 계층에서 동작하는 프로토콜로서, Netscape사에서 만들었으며 ISO 표준 정식명칭은 TLS(Transport Layer Security)이다. TLS에서는 통신하기 전에 핸드 쉐이킹 프로토콜을 통해서 서로 안전하게 대칭키를 공유하기 위한 작업을 수행하는데, 핸드 쉐이킹 과정에서 사용가능한 비·대칭키 암호화 기법 및 해쉬 종류 등 필요한 정보들을 공유한 후에 송신자는 수신자에게 사전에 정한 비대칭키 암호화 기법으로 앞으로 통신에서 사용하게 될 대칭키를 수신자의 공개키로 암호화하여 수신자에게 보낸다. 송신자로부터 받은 암호문을 수신자는 자신의 비밀키로 복호화하여 대칭키를 안전하게 획득한다. 이로써 TLS에서는 비대칭 암호화 기법과 대칭키 암호화 기법의 단점을 보완하여 안전한 통신이 가능하다.

[0005] 그러나 TLS 프로토콜 안전성은 암호화 기법의 안전성에 의존할 뿐 사용하는 비·대칭키 암호화 기법의 안전성을 보완해 주진 않는다. 또한 TLS는 실시간성과 안전성을 네트워크 상황에 맞게 적절하게 조절하지 못하는 단점을 가지고 있다.

[0006] IoT와 웨어러블 디바이스 사용으로 안전성과 실시간성이 중요시 되고 있는 오늘날 현대 사회에서는 디바이스에 기존의 TLS와 같은 암호화 모듈을 적용하여 사용하는 경우 오버헤드와 딜레이가 발생하게 된다. 즉, 실시간성을 제공하는 것이 가능하게 하려할 경우 안전성이 위협을 받게 되는 반면, 안전성을 제공하는 것이 가능하게 하려할 경우 실시간성이 떨어지게 된다.

[0007] 또한, 네트워크 통신 환경에 안전성을 제공하기 위한 암호화 기법에는 스트림 암호, 대칭키 암호화 기법, 비대칭키 암호화 기법이 있다. 비대칭키 암호화 기법은 기본적으로 키 길이가 길기 때문에 높은 안전성을 제공하지만 실시간성을 제공하지는 못한다. 반면 스트림 암호는 간단한 XOR 연산을 통해서 암호문을 생성하기 때문에 50% 확률로 평문을 예측하는 것이 가능해 실시간성은 제공하지만 안전성은 제공하지 못한다. 대칭키 암호화 기법의 경우 비대칭키 암호화 기법보다 키 길이가 짧아 빠르며 스트림 암호보다 안전하기 때문에 일반적으로 다른 암호화 기법보다 균형 잡힌 안전성과 실시간성을 보장한다. 이러한 대칭키 암호화 기법에는 ARIA, SEED, AES 등이 있는데, 각기 3가지의 키 길이를 가지고 있으며 키 길이가 길수록 높은 안전성을 제공하지만 암호화/복호화로 인한 딜레이가 발생하여 실시간성이 떨어지게 된다. 게다가, 기존의 대칭키 암호화 기법들은 양자 컴퓨터의 등장으로 더 이상 키 길이가 짧은 것은 안전성을 보장할 수 없게 되면서 키 길이가 가장 긴 것을 사용해야지만 안전성을 보장할 수 있게 되었다.

[0008] 이러한 문제를 다루기 위한 기존의 해결 방법으로는 안전성을 위해 키 길이가 긴 대칭키 암호화 기법을 사용하면서 동시에 실시간성을 제공하기 위해 사용자가 임의로 암호화 기법을 적용할 영역과 적용하지 않을 영역을 나누는 방법을 사용하였다. 그러나 기존의 이러한 방법은 암호화 기법을 적용하지 않은 영역의 경우 정보가 그대로 노출될 뿐만 아니라 안전성과 실시간성을 요하는 네트워크 환경에서 근본적인 해결책이 되지 못한다.

[0009] 이에, 리소스 제한과 계산량 한계를 가지는 IoT 디바이스, 무인 이동체, 차량 네트워크 환경에서 안전성과 실시간성이 가능한 암호화 기법에 대한 기술 개발이 요구되고 있다.

선행기술문헌

특허문헌

[0011] (특허문헌 0001) 한국 공개특허 제 10-2017-0097294호(2017.08.28. 공개)

발명의 내용

해결하려는 과제

[0012] 본 발명이 해결하고자 하는 기술적 과제는 리소스 제한과 계산량 한계를 가지는 IoT 디바이스, 무인 이동체, 차량 네트워크 환경에서 안전성과 실시간성을 제공하는 것이 가능하도록 하는 실시간 데이터 전송을 위한 블록 암호 장치 및 방법을 제공하는 것이다.

[0013] 본 발명의 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0015] 본 발명의 일 실시예에 따른 블록 암호 장치는, 서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각

평문(plaintext) 블록을 암호화하여 암호문 블록을 생성하는 블록 암호화부, 상기 블록 암호화부에서 암호화되는 현재 평문 블록의 암호화시 선택된 키 및 상기 현재 평문 블록 이전의 평문 블록에 의해 생성된 이전 메시지 인증 코드를 이용하여 메시지 인증 코드를 생성하는 메시지 인증부를 포함한다.

[0016] 바람직하게는, 평문 메시지를 서로 다른 비트 길이를 갖는 복수의 순서화된 평문 블록으로 나누는 입력 처리부를 더 포함할 수 있다.

[0017] 바람직하게는, 상기 블록 암호화부는, 상기 암호문 블록이 기 설정된 크기 이상인 경우, 상기 암호문 블록을 기 설정된 크기에 해당하는 헤더(header)와 나머지를 테일(tail)로 나누고, 상기 헤더를 암호문 블록으로 생성하며, 상기 테일을 다음 순서의 평문 블록과 합하여 블록 암호화되도록 할 수 있다.

[0018] 바람직하게는, 상기 메시지 인증부는, 상기 선택된 키와 상기 이전 메시지 인증 코드를 배타적 논리합(XOR) 연산하는 배타적 논리합 연산모듈, 상기 현재 평문 블록과 상기 배타적 논리합 연산모듈에서 연산된 값을 제1 해시 알고리즘으로 암호화하는 제1 해시모듈, 상기 제1 해시모듈에서 암호화된 값을 제2 해시 알고리즘으로 암호화하여 현재 평문 블록에 대한 메시지 인증 코드를 생성하는 제2 해시모듈을 포함할 수 있다.

[0019] 바람직하게는, 상기 현재 평문 블록에 대한 암호문 블록과 메시지 인증 코드를 수신장치로 전송하는 통신부를 더 포함하되, 상기 통신부는 암호문 블록마다 카운터값을 함께 전송할 수 있다.

[0020] 바람직하게는, 상기 블록 암호화부에서 각 평문 블록을 암호화하기 위해 사용되는 서로 다른 길이의 키들을 생성하고, 상기 생성된 키들의 순서를 정하여 패턴으로 설정하는 키 생성부를 더 포함하되, 상기 키 생성부는 네트워크 환경과 데이터 안정성을 고려하여 패턴을 설정할 수 있다.

[0021] 바람직하게는, 핸드쉐이크 과정을 통하여 서로 다른 길이를 갖는 키들의 순서가 설정된 패턴, 공개키, 사전 마스터 비밀키 중 적어도 하나를 수신장치와 공유하도록 하는 상호 인증부를 더 포함할 수 있다.

[0022] 본 발명의 일 실시예에 따른 블록 암호 방법은, 블록 암호 장치가 평문 메시지를 블록 암호하는 방법에 있어서, 핸드쉐이크 과정을 통하여 서로 다른 길이를 갖는 키들의 순서가 설정된 패턴, 공개키, 사전 마스터 비밀키 중 적어도 하나를 수신장치와 공유하는 단계, 상기 평문 메시지를 복수의 평문 블록으로 나누고, 각 평문 블록에 대해 상기 패턴의 순서에 따라 키를 선택하여 암호문 블록과 메시지 인증 코드를 각각 생성하는 단계를 포함한다.

[0023] 바람직하게는, 각 평문 블록에 대응하는 암호문 블록과 메시지 인증 코드를 수신장치로 전송하는 단계를 더 포함할 수 있다.

발명의 효과

[0025] 본 발명에 따르면, PCB 운영모드는 독립적인 구조를 가짐으로써 패딩과 같은 추가적인 연산을 수행할 필요가 없기 때문에 실시간성이 가능하다.

[0026] 또한, PCB는 서로 다른 길이의 키를 동시에 사용함으로써 생기는 패턴 특징을 통해서 기존의 대칭키 암호화 기법의 키 길이에 대한 안전성 문제를 보완할 수 있기 때문에 높은 수준의 안전성을 제공할 수 있다.

[0027] 따라서, 본 발명의 PCB는 리소스 제한과 계산량 한계를 가지는 IoT 등과 같은 디바이스 네트워크 환경에 안전성과 실시간성을 제공하는 것이 가능하다. 뿐만 아니라 PCB는 이동성으로 인해 패킷 손실이 일어나기 쉬운 무인 이동체 및 차량 네트워크 환경에서도 오류확산 없이 복호화할 수 있다. 즉, 리소스 제한, 계산량 한계, 이동성으로 인한 패킷손실이 일어나기 쉬운 특성 등 이러한 제한들을 가진 분야에 PCB 운영모드를 적용함으로써 안전성과 실시간성이 가능하다.

[0028] 본 발명의 효과들은 이상에서 언급한 효과들로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0030] 도 1은 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호의 개념을 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호 장치를 설명하기 위한 블록도이다.

도 3은 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호 장치의 동작을 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 패턴 형식에 사용되는 서로 다른 길이의 키를 생성하는 방법을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 동일한 평문 블록에 의해 생성되는 암호문을 설명하기 위한 도면이다.

도 6은 본 발명의 일 실시예에 따른 PCB 운영 모드의 공격 모델을 설명하기 위한 알고리즘이다.

도 7은 본 발명의 일 실시예에 따른 암호문 마다 같이 제공하는 카운터 값을 설명하기 위한 도면이다.

도 8은 본 발명의 일 실시예에 따른 블록 암호 및 복호 방법을 설명하기 위한 도면이다.

도 9는 본 발명의 일 실시예에 따른 상호 인증 방법을 설명하기 위한 도면이다.

도 10은 본 발명에 따른 PCB 운영모드의 성능을 ECB 운영모드와 비교한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0031] 이하, 첨부한 도면을 참고로 하여 본 발명의 여러 실시 예들에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예들에 한정되지 않는다.

[0032] 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 동일 또는 유사한 구성요소에 대해서는 동일한 참조 부호를 붙이도록 한다. 따라서 앞서 설명한 참조 부호는 다른 도면에서도 사용할 수 있다.

[0033] 또한, 도면에서 나타난 각 구성의 크기 및 두께는 설명의 편의를 위해 임의로 나타내었으므로, 본 발명이 반드시 도시된 바에 한정되지 않는다. 도면에서 여러 층 및 영역을 명확하게 표현하기 위하여 두께를 과장되게 나타낼 수 있다.

[0034] 이하, 본 발명의 실시예에 따른 실시간 데이터 전송을 위한 블록 암호 장치 및 방법을 첨부된 도면을 참조하여 상세하게 설명하면 아래와 같다.

[0035] 도 1은 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호의 개념을 설명하기 위한 도면이다.

[0036] 도 1을 참조하면, 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호는 challenge-response을 통해 상호 인증 프로세스를 수행하는 상호인증 프로토콜(mutual authentication protocol), 서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각 평문 블록을 암호화하여 암호화문 블록을 생성하는 Patterned Cipher Block (PCB), 암호화 해시 함수의 특성을 기반으로 메시지 무결성을 검증하는 message integrity authentication을 통해 이루어진다.

[0037] 상호 인증 프로토콜은 핸드쉐이킹을 통해 송신자와 수신자가 패턴 정보와 키를 공유하도록 한다. 이러한 상호 인증 프로토콜은 챌린지-응답을 통해 상호 인증 프로세스를 안전하게 수행하면서 두 라운드 통신으로 키와 패턴 정보를 교환할 수 있다. 여기서, 상호 인증은 상대방의 신원을 식별하고 공격자(attacker)가 위장하는 것을 방지하는 것이다. 따라서, 챌린지-응답 시스템은 고유한 키를 소유한 사용자만이 수행할 수 있도록 구축해야 한다.

[0038] PCB는 독립적인 구조로 서로 다른 길이의 대칭 키들을 임의로 순서를 정하여 암호화함으로서, 패턴 형식을 가지는 기법임과 동시에 공격자가 송신자로부터 전송된 암호문을 교체하거나 위조하는 경우를 수신자가 검증가능하게 해주는 기술이다.

[0039] 일반적으로 대칭키 암호화 기법에서 키 길이가 짧을수록 실시간성은 높지만 안전성은 떨어지고, 키 길이가 길수록 안전성은 높지만 실시간성은 떨어진다. 그러나 본 발명에 따른 PCB는 서로 다른 길이의 대칭 키들을 동시에 사용하여 서로 다른 길이의 키 비율을 조절함으로써 안전성과 실시간성을 가능하게 한다. 따라서, 사용자가 기존에 하나의 키를 사용하는 경우보다 서로 다른 길이의 키들을 동시에 사용할 경우 더 안전하다. 즉, 공격자가 임의의 암호문을 해독하기 위해 공격을 시도해야하는 횟수가 증가할 뿐만 아니라 알아내야 하는 키들이 늘어남으로써 높은 수준의 안전성을 제공하는 것이 가능하다. 이전보다 높은 안전성을 제공하는 것이 가능함을 아래 수학식 1과 같이 나타낼 수 있다.

[0040] [수학식 1]

[0041] $$S = A \times \frac{L!}{\prod_{i=0}^{N-1} n_i!}$$

[0042] 여기서, S는 PCB의 암호를 해독하기 위한 총 수행횟수, L은 전체 블록의 길이, A는 패턴길이의 탐색 영역(공격자가 임의로 패턴을 추측하는 범위), N은 전체 개수, n_i는 서로 다른 키 길이로 암호화한 각각의 블록을 의미할 수 있다.

[0043] 상술한 바와 같이 PCB는 패턴(Pattern)이라는 주어진 시퀀스의 순서로 블록당 여러 개의 암호 알고리즘을 번갈아 사용하는 운영 모드(operation mode)이다. PCB 운영 모드에서 송신자와 수신자는 동일한 패턴을 가지며, 각 블록에 대해 동일한 알고리즘을 사용하여 암호화 또는 복호화를 수행한다. 이때, 공격자가 암호문들을 획득하더라도 공격자는 동일한 키로 암호화된 블록 집합을 모르기 때문에, 공유 키를 추출하기 어렵다. 이러한 공격자에 대한 방어는 보안 개선과 상대적으로 빠른 암호화/복호화 방법을 이용할 기회를 가져오며 이는 시간 비용을 줄일 수 있다.

[0044] 메시지 무결성 검증(Message Integrity Authentication)은 메시지 전송 중에 메시지 내용이 부적절하게 변조되었는지 여부를 수신자가 확인할 수 있도록 하여 공격자가 변조하는 것을 방지하는 것이다. 즉, 송신자와 수신자가 사용하는 암호 시스템은 안전하지만 공격자가 악의적으로 암호문을 변조하거나 변조한다는 것을 알기 위해서는 메시지 인증 기법이 필요하다. 이에 본 발명에 따른 메시지 무결성 검증은 서로 다른 암호학 해시 알고리즘을 두 번 사용하여, 공격자로부터 메시지의 위조를 방지한다. 즉, 메시지 인증은 암호화하기 위해 사용되는 키(Key(K_N))와 이전에 생성된 메시지 인증 코드(무결성 검증 값) H_N 을 함께 XOR 연산하고, XOR 연산한 결과 값을 암호학 해시 알고리즘의 입력 값으로 사용하여 결과 값을 얻는다. 그런 후, 획득한 결과 값을 한 번 더 암호학 해시 알고리즘의 입력 값으로 받음으로써 해당 평문에 대한 무결성 검증 결과 값을 생성할 수 있다.

[0045] 이와 같이 메시지 무결성 검증은 서로 다른 암호학 해시 알고리즘을 두 번 사용함으로써, 암호학 해시 함수의 충돌을 방지할 수 있을 뿐만 아니라 암호학 해시 비가역성 특성으로 인해 공격자가 위조하는 것을 불가능하도록 한다.

[0046] 상술한 바와 같은 블록 암호 기술은 대칭키 암호화 기법을 네트워크 환경에 맞게 암호화할 수 있는 운영 모드를 제공할 뿐만 아니라 무결성 검증을 함께 제공함으로써 인증된 운영 모드 기술이라 할 수 있다.

[0047] 도 2는 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호 장치를 설명하기 위한 블록도, 도 3은 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호 장치의 동작을 설명하기 위한 도면, 도 4는 본 발명의 일 실시예에 따른 패턴 형식에 사용되는 서로 다른 길이의 키를 생성하는 방법을 설명하기 위한 도면, 도 5는 본 발명의 일 실시예에 따른 동일한 평문 블록에 의해 생성되는 암호문을 설명하기 위한 도면, 도 6은 본 발명의 일 실시예에 따른 PCB 운영 모드의 공격 모델을 설명하기 위한 알고리즘, 도 7은 본 발명의 일 실시예에 따른 암호문 마다 같이 제공하는 카운터 값을 설명하기 위한 도면이다.

[0048] 도 2를 참조하면, 본 발명의 일 실시예에 따른 실시간 데이터 전송을 위한 블록 암호 장치(100)는 입력 처리부(110), 키 생성부(115), 블록 암호화부(120), 메시지 인증부(130)를 포함한다.

[0049] 입력 처리부(110)는 입력받은 평문 메시지를 서로 다른 비트 길이를 갖는 복수의 순서화된 평문 블록으로 나눈다. 즉, 입력 처리부(110)는 평문 메시지를 복수의 평문 블록(P_1, P_2, P_3, ... P_{N-1}, P_N)으로 변환한다. 이때, 복수의 평문 블록들은 동일한 비트 길이 또는 서로 다른 비트 길이를 갖는 블록일 수 있다. 또한, 평문 블록은 특정 순서로 처리되어야만 하고, 따라서 암호문 메시지를 복호화하는 것을 목표로 하는 역 프로세스 중에 동일한 순서가 적용될 수 있다. 따라서, 평문 블록은 순차적인 방식으로 배열된다.

[0050] 키 생성부(115)는 블록 암호화부(120)에서 평문 블록을 암호화하기 위해 사용되는 서로 다른 길이의 키들을 생성하고, 그 생성된 키들의 순서를 정하여 패턴으로 설정한다. 이때, 키 생성부(115)는 네트워크 환경과 데이터 안정성을 고려하여 패턴을 설정할 수 있다.

[0051] 본 발명에 따른 PCB는 서로 다른 길이의 대칭 키들을 동시에 사용하여 서로 다른 길이의 키 비율을 조절함으로써 안전성과 실시간성을 가능하게 한다. 대칭 키 암호에는 일반적으로 128 비트, 196 비트 및 256 비트의 세 가

지 키 길이가 사용된다. PCB 모드에서, 서로 길이가 다른 키에는 다른 암호 색인(index)이 할당된다. 따라서, 키 생성부(115)는 대칭키 암호화 기법에서 암호화하기 위해 사용되는 서로 다른 길이의 키들을, PW로 3가지 종류의 키들을 생성하는 방법, 서로 다른 3종류의 해시(hash) 함수를 사용하여 하나의 PW를 입력하였을 때 나오는 결과 값들을 비트에 맞게 잘라 사용하는 방법, 사용자가 임의로 서로 다른 3종류의 키를 각각 생성하는 방법 등을 이용하여 생성할 수 있다. 여기서, 하나의 PW로 3가지 종류의 키들을 생성하는 방법은 PW를 hash 함수의 입력 값으로 하여 나오는 결과 값을 비트에 맞게 잘라 사용하는 방법이다. hash 함수를 사용하는 이유는 공격자가 하나의 해시 결과 값을 알아내더라도 hash 함수 성질인 역상저항성으로 나머지 해쉬 결과 값들을 알 수 없기 때문이다. 예를 들면, 도 4의 (a)와 같이 키 K를 $Hash_1$ 함수에 입력하여, $F_{123}(Hash_1(K))$, $F_{196}(Hash_1(K))$, $F_{256}(Hash_1(K))$와 같이 서로 다른 길이의 키를 생성할 수 있다.

[0052] 서로 다른 3종류의 hash 함수를 사용하여 하나의 PW를 입력하였을 때 나오는 결과 값들을 비트에 맞게 잘라 사용하는 방법에서 3종류의 hash 함수를 사용하는 이유는 hash 함수는 기본적으로 충돌 저항성 성질을 만족한다고 하나 발생할지 모르는 충돌에 대해 대비하기 위해서이다. 예를 들면, 도 4의 (b)와 같이 키 K를 $Hash_1$ 함수, $Hash_2$ 함수, $Hash_3$ 함수에 각각 입력하여, $F_{123}(Hash_1(K))$, $F_{196}(Hash_2(K))$, $F_{256}(Hash_3(K))$를 각각 생성할 수 있다.

[0053] 마지막으로, 사용자가 임의로 서로 다른 3종류의 키를 각각 생성하는 방법은 가장 안전한 방법이라고 할 수 있다.

[0054] 상술한 바와 같이 키 생성부(115)는 다양한 방법으로 서로 길이가 다른 키들을 생성할 수 있다.

[0055] 블록 암호화부(120)는 서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각 평문 블록을 암호화하여 암호화문 블록을 생성한다. 이때, 블록 암호화부(120)는 DES, TripleDES, AES 또는 임의의 다른 블록 암호 알고리즘을 이용하여 각 평문 블록을 암호화할 수 있다.

[0056] PCB 모드에서 송신자와 수신자는 패턴이라고 불리는 동일한 정보를 가진다. 이 정보는 각각 암호 알고리즘으로 매핑되는 정수 시퀀스로, 암호 색인(cryptic index)이라고도 할 수 있다. 따라서, 블록 암호화부(120)는 각 평문 블록에 대해 패턴의 순서에 따라 암호 색인과 매핑된 해당 암호 알고리즘을 사용하여 암호화를 수행한다. 이때, 사용 가능한 암호 알고리즘이 n 개 있다면(각 키의 사전 마스터 키가 있는 경우), 패턴의 길이는 n보다 커야 일부 또는 전체가 PCB 모드에서 사용되고, 평문(plaintext) 블록은 서로 다른 키 길이를 사용하여 독립적으로 암호화되므로, 평문과 암호문은 일대일 관계를 유지한다.

[0057] 예를 들어, 평문 블록이 (P_1, P_2, P_3, ..., P_{N-1}, P_N)이고, 패턴이 (제1키, 제2키, 제3키, 제1키, 제2키, 제3키, 제1키, 제2키, 제3키)으로 설정되었다고 가정하여 설명하기로 한다. 이 경우, 블록 암호화부(120)는 평문 블록 P_1을 제1키에 기초하여 암호화, 평문 블록 P_2을 제2키에 기초하여 암호화, 평문 블록 P_3을 제3키에 기초하여 암호화, 평문 블록 P_4을 제1키에 기초하여 암호화 등과 같이 평문 블록의 순서에 해당하는 패턴 순서의 키를 이용하여 평문 블록을 암호화한다.

[0058] 상술한 바와 같이 블록 암호화부(120)는 서로 다른 길이의 키에 따라 암호화를 수행하기 때문에 키들의 순서로 인한 패턴을 가지고, 패턴 형식을 가지는 구조적인 특성을 통해 CBC와 같이 동일한 평문에 대해 암호화를 수행하였을 때 서로 다른 암호문이 생성된다. 예컨대, 도 5의 (a)와 같은 ECB는 동일한 평문블록 P1에 동일한 키를 사용하여 암호화를 수행하므로, 동일한 암호문이 생성되고, 이로 인해 공격자는 암호문만 획득해도 특정 평문의 암호문이 반복된다는 것을 알 수 있다. 반면, 도 5의 (b)와 같은 PCB는 동일한 평문블록 P1일지라도 패턴에 설정된 순서에 따라 다른 키를 사용하여 암호화를 수행하므로, 서로 다른 암호문이 생성된다. 이처럼, ECB 모드는 CPA 공격 모델에 취약하다. 그러나, PCB는 패턴화된 포맷을 적용하는 독립적인 구조를 가지고 있으므로 ECB의 기존 공격 모델은 불가능할 수 있다. 일반적으로 기존 공격은 LR Encryption Oracle을 통해 불가능하다는 것을 알 수 있다. 이는 도 6의 알고리즘 1을 통해 PCB가 IND-CPA 공격을 방어할 수 있음을 입증할 수 있다. 도 6에서 adversary는 $\mathbf{A}^{\mathbf{E}_k(\mathbf{LR}(\ldots,\mathbf{b}))}$, 평문은 MI, 암호문은 C[i]을 나타낸다.

[0059] 상술한 바와 같이 PCB는 상이한 키 길이를 사용할 수 있는 패턴 형식을 가지므로, 동일한 평문에 대해 동일한 암호문을 생성하지 않고, 이로 인해 공격자가 블록 재사용 또는 복호를 위해 기존의 패턴을 분석하는 것이 불가능하다.

[0060] 또한, 블록 암호화부(120)는 암호화된 블록이 이전 평문 블록 크기 이상인 경우, 암호화된 블록을 이전 평문 블

록 크기의 헤더와 나머지를 테일로 나누고, 헤더를 암호문 블록으로 생성하며, 테일을 다음단 평문 블록과 합하여 블록 암호화한다.

[0061] 예를 들어, 도 3을 참조하면, (N-1)번째 평문 $P_{(N-1)}$을 키 Key(K_{N-1})를 이용하여 암호화한 결과가 이전 평문블록의 사이즈와 다르므로, 블록 암호화부(120)는 암호화된 결과 a를 헤드(head)와 테일(tail)로 나눈다. 이때, 헤드는 이전 평문 블록의 사이즈에 해당하는 길이일 수 있고, 테일은 나머지 길이일 수 있다. 블록 암호화부(120)는 테일을 다음단으로 전송하여, N번째 평문 P_N과 합하여 암호화되도록 한다.

[0062] 상술한 바와 같이 블록 암호화부(120)는 암호화 스틸링(stealing) 기술을 사용하여 추가적인 패딩 작업을 요구하지 않는다. 따라서 PCB는 암호문 전송 중 일부 암호문에 비트 오류나 손실이 발생하더라도 오류 확산없이 실시간으로 전송할 수 있다.

[0063] 메시지 인증부(130)는 블록 암호화부(120)에서 암호화되는 현재 평문 블록의 암호화시 선택된 키 및 상기 현재 평문 블록 이전의 평문 블록에 의해 생성된 이전 메시지 인증 코드를 이용하여 메시지 인증 코드를 생성한다. 하는 메시지 인증부를 포함한다. 예컨대, 메시지 인증부(130)는 블록 암호화부(120)에서 선택된 키, 블록 암호화부(120)에서 암호화되는 N-1번째 평문 블록 및 N-1번째 평문 블록 이전의 N-2번째 평문 블록에 의해 생성된 N-2번째 메시지 인증 코드 H_{N-2} 을 이용하여 N-1번째 메시지 인증 코드 H_{N-1} 를 생성한다.

[0064] 메시지 인증부(130)는 서로 다른 암호학 해시 알고리즘을 두 번 사용하여, 공격자로부터 메시지의 위조를 방지하기 위한 메시지 인증 코드를 생성한다. 즉, 메시지 인증부(130)는 암호화하기 위해 사용되는 키(Key(K_N))와 이전에 생성된 메시지 인증 코드(무결성 검증 값) H_N 을 함께 XOR 연산하고, XOR 연산한 결과 값을 제1 암호학 해시 알고리즘의 입력 값으로 사용하여 결과 값을 얻는다. 그런 후, 메시지 인증부(130)는 제1 암호화 해시 알고리즘을 통해 획득한 결과 값을 제2 암호학 해시 알고리즘의 입력 값으로 사용하여 해당 평문 블록에 대한 메시지 인증 코드를 생성한다.

[0065] 이러한 메시지 인증부(130)는 배타적 논리합 연산모듈(132), 제1 해시모듈(134), 제2 해시모듈(136)을 포함한다.

[0066] 배타적 논리합 연산모듈(132)은 블록 암호화부(120)에서 선택된 키와 블록 암호화부(120)에서 암호화되는 N-1번째 평문 블록 이전의 N-2번째 평문 블록에 의해 생성된 N-2번째 메시지 인증 코드 H_{N-2} 을 배타적 논리합(XOR) 연산한다. 도 3을 참조하면, 배타적 논리합 연산모듈(132)은 H_{N-2} 와 블록 암호화부(120)에서 N-1번째 평문 블록 P_{N-1}을 암호화하기 위해 사용되는 키 Key(K_{N-1})을 배타적 논리합 연산하여 $\mathbf{K}_{(N-1)} \oplus \mathbf{H}_{(N-2)}$ 을 출력한다.

[0067] 제1 해시모듈(134)은 N-1번째 평문 블록 P_{N-1}과 배타적 논리합 연산모듈(132)에서 연산된 값을 제1 해시 알고리즘으로 암호화한다. 이때, 제1 해시모듈(134)은 예컨대 HMAC(Hash-based Message Authentication Code)을 이용하여 MAC값을 생성할 수 있다.

[0068] 제2 해시모듈(136)은 제1 해시모듈(134)에서 암호화된 값을 제2 해시 알고리즘으로 암호화한다. 이때, 제2 해시모듈(136)은 예컨대 SHA-3(Secure Hash Algorithm 3)을 이용하여 N-1번째 메시지 인증 코드 H_{N-1} 를 생성한다.

[0069] 상술한 바와 같이 메시지 인증부(130)는 각 평문 블록마다 해시(hash) 연산이 2회 실행된다. 따라서, 평문 메시지가 n 개의 평문 블록으로 나누어진 경우, n 개의 평문 블록에 대응하는 n 개의 메시지 인증 코드가 생성 및 전송되어야 하고, 2n 회의 해시 연산이 요구된다.

[0070] PCB 운영 모드를 사용하여 암호화된 암호문이 아무리 안전하더라도 메시지 무결성이 검증되지 않으면, 수신자는 통신중 공격자가 암호문의 일부를 삭제하거나 대체하는 것을 인식하지 못한다. 그러므로, 본 발명은 암호화 스틸링 기술과 함께 사용될 수 있는 메시지 무결정 검증을 이용한다. 메시지 인증부는 평문 길이 N의 (N-1) 번째 평문을 $P_{(N-1)}$이라 할 때, $P_{(N-1)}$와 $\mathbf{K}_{(N-1)} \oplus \mathbf{H}_{(N-2)}$ 을 HMAC 알고리즘의 입력값으로 입력하고, 그 알고리즘의 출력값을 해시 함수 SHA-3의 입력 값으로 입력한다. 따라서, 블록 암호 장치(100)는 메시지 인증부(130)의 최종 출력값 $\mathbf{H}_{(N-1)}\ (= \mathbf{SHA}(\mathbf{HMAC}_{\mathbf{K}_{(N-1)} \oplus \mathbf{H}_{(N-2)}}(\mathbf{P}_{(N-1)})))$ 과 블록 암호화부(120)에서

생성된 암호문 $\mathbf{C}_{(N-1)}\ (=\mathbf{E}_{\mathbf{K}_{(N-1)}}(\mathbf{P}_{(N)}||\mathbf{Tail}))$ 을 수신장치로 전송한다. 그후, 블록 암호 장치(100)는 N 번째 평문이 $P_{(N)}$이라 할 때, $K_{(N)}$과 이전의 평문 블록을 통해 얻어진 해시 값을 배타적 논리합 연산(exclusive OR operation)을 수행한다. 배타적 논리합 연산을 수행한 결과 및 $P_{(N)}$ 을 HMAC 알고리즘의 입력 값으로 입력하고, 그 알고리즘의 출력값은 해시 함수 SHA-3의 입력 값으로 입력한다. 그러면, 블록 암호 장치(100)는 최종 출력값과 N번째 평문블록을 암호화한 암호문을 수신장치로 전송한다. 결론적으로, 블록 암호 장치(100)는 메시지 무결성 검증을 위해 각 평문블록의 암호문 $\mathbf{C}_i\ (=\mathbf{E}_{\mathbf{K}_i}(\mathbf{P}_i))$ 과 함께 메시지 인증 코드 $\mathbf{H}_i\ (=\mathbf{SHA}(\mathbf{HMAC}_{\mathbf{K}_i\oplus\mathbf{H}_{(i-1)}}(\mathbf{P}_i)))$ 를 수신장치로 전송한다.

[0071] 한편, 본 발명에 따른 블록 암호 장치(100)는 chain 형태로 이전 암호문이 다음 암호문 생성에 영향을 주는 구조이기 때문에, 공격자가 중간에 암호문을 위조하거나 교환하더라도 수신자는 $\mathbf{H}_i$ 를 통해서 실제 송신자로부터 전송된 암호문인지를 검증할 수 있다. 이러한 메시지 무결성 메커니즘의 보안은 암호화 해시 함수의 세 가지 속성으로 나타낼 수 있다. 첫 번째 역상저항성(preimage resistance)은 주어진 해시 값에 대해 해시 값을 생성하는 입력 값을 찾기가 어렵다는 것을 정의한다. 두번째 역상저항성은 입력 값에 대한 해시 값을 변경하지 않으면서 입력 값으로부터 다른 입력 값을 발견하는 것이 계산 상으로 불가능하다는 것을 정의한다. 세번째 충돌 저항성은 동일한 해시 값을 생성하는 2 개의 입력 값을 찾는 것이 계산상으로 어렵다는 것을 정의한다. 따라서 암호화 해시 함수는 해시 값을 통해 원본 텍스트를 재현할 수 없는 단방향 함수이다. 또한 이전 SHA 및 HMAC 보안 문제를 고려하여 안전성이 입증된 SHA-3와 HMAC를 함께 사용하여 메시지 무결성의 보안을 향상시킬 수 있다.

[0072] 한편, 본 발명에 따른 블록 암호 장치(100)는 상호 인증부(미도시)를 더 포함할 수 있다. 상호 인증부는 핸드쉐이크 과정을 통하여 송신자와 수신자가 서로의 공개키, 패턴 등을 공유하도록 한다.

[0073] 즉, 송신자와 수신자가 정보를 암호화하고 안전하게 패턴 정보를 전송하기 위해서는 상호 인증을 통해 안전하게 세션을 설정해야 한다. 따라서, 상호 인증부는 challenge-response을 통해 안전한 상호 인증 프로세스를 만족시킬 수 있는 프로토콜을 이용하고, 이 프로토콜을 통해 공격자가 자신을 송신자로 위장하는 것을 막을 수 있다.

[0074] 또한, 본 발명에 따른 블록 암호 장치(100)는 블록 암호화부(120)에서 생성된 암호문 블록과 메시지 인증부(130)에서 생성된 메시지 인증 코드를 수신장치로 전송하는 통신부(미도시)를 더 포함할 수 있다. 이때, 통신부는 암호화문 블록마다 카운터값을 함께 전송할 수 있다.

[0075] 실제 PCB를 네트워크 환경에 적용하기 위해서는 네트워크 계층의 TCP 또는 UDP 특성을 고려해야 한다. 네트워크 계층에서는 TCP와 달리 UDP를 사용하면 패킷 손실이 자주 발생할 수 있다. 따라서 CBC나 스트림 암호를 사용하면 오류 확산이 발생하거나 적절한 암호문을 얻을 수 없다. 그러나, PCB는 패턴 길이 L을 알면, 도 6과 같이 패킷 손실 및 스킵을 검출하기 위해 암호문 블록에 대한 카운터 크기 M을 제공할 수 있으므로, 손실된 블록만 재전송 또는 무시되므로 UDP 환경에서 사용할 수 있다. 즉, PCB는 독립된 구조적인 특성을 가지기 때문에 대칭키 암호화 기법을 통해서 생성되는 암호문마다 도 7과 같이 Counter 값을 제공하는 것이 가능하다. 따라서, 네트워크 환경에서 TCP뿐만 아니라 패킷 손실이 빈번하게 일어날 수 있는 UDP에서도 오류확산 없이 제대로 된 평문들을 얻을 수 있으며 중간에 패킷 손실이 일어나더라도 Counter 값을 통해 손실된 패킷을 재전송하는 것이 가능하다.

[0076] 한편, 본 발명의 블록 암호 장치(100)는 전용의 하드웨어 또는 CPU, 마이크로프로세서, 마이크로 컨트롤러 또는 SoC (System on Chip) 등 다양한 형태로 구현될 수 있으며, 하드웨어 및 소프트웨어가 결합된 형태로 구현될 수 있다. 또한 설명을 위해 도시된 블럭들과 같이 각각 구분되는 형태로 구현될 수도 있으나, 동일 하드웨어에 소프트웨어의 형태로 구현되거나, 동일 하드웨어에서 동일한 기본 소프트웨어 블록을 재사용하여 서로 다른 기능 블록을 구현할 수도 있다.

[0077] 도 8은 본 발명의 일 실시예에 따른 블록 암호 및 복호 방법을 설명하기 위한 도면이다.

[0078] 도 8을 참조하면, 송신장치와 수신장치는 핸드 쉐이크 과정을 통해 서로의 공개키, 사전 마스터 비밀키, 패턴 정보 등을 공유한다(S810). 송신장치와 수신장치가 상호 인증을 수행하는 방법에 대한 상세한 설명은 도 9를 참조하기로 한다.

[0079] 단계 S810이 수행되면, 송신장치는 평문 메시지를 N개의 평문 블록으로 나누고(S820), 각 평문블록에 대해 암호

문 블록과 메시지 인증 코드를 각각 생성한다(S830). 즉, 송신장치는 서로 다른 길이를 갖는 키들의 순서에 따라 키를 선택하여 각 평문블록을 암호화하여 암호화문 블록을 생성한다. 또한, 송신장치는 해당 평문블록을 암호화하기 위해 사용되는 키와 이전에 생성된 메시지 인증 코드를 XOR 연산하고, XOR 연산한 결과 값과 해당 평문블록을 제1 암호학 해시 알고리즘의 입력값으로 사용하여 결과값을 획득하고, 그 결과값을 제2 암호학 해시 알고리즘의 입력 값으로 사용하여 해당 평문 블록에 대한 메시지 인증 코드를 생성한다.

[0080] 단계 S830이 수행되면, 송신장치는 각 평문 블록에 대응하는 암호화문 블록과 메시지 인증 코드를 수신장치로 전송하고(S840), 수신장치는 암호화문 블록과 메시지 인증 코드를 이용하여 평문 블록을 복호하고 메시지 인증을 수행한다(S850).

[0081] 도 9는 본 발명의 일 실시예에 따른 상호 인증 방법을 설명하기 위한 도면이다.

[0082] 도 9를 참조하면, 송신장치는 수신장치의 공개 키(PK_B)를 사용하여 prime값 G^x를 암호화하여 랜덤 값 r^A 및 메시지 인증 코드 H_A 와 함께 수신장치로 전송한다(S910). 즉, 송신장치는 사전 마스터 비밀 키(pre-master secret key)를 생성하기 위해 수신장치의 공개 키(PK_B)를 사용하여 G^x를 암호화하고, 수신장치가 올바른 수신자인지 확인하기 위해 r^A를 전송한다. 또한, 송신장치는 메시지 인증 코드 H_A를 수신장치에게 전송하여 송신장치가 보낸 것인지 검증할 수 있도록 한다.

[0083] 단계 S910이 수행되면, 수신장치는 G^y와 r^B를 송신장치의 공개키(PK)로 암호화하고, 송신장치로부터 수신한 r^A를 사전 마스터 비밀 키 K로 암호화한 다음 이를 메시지 인증 코드 H_B와 함께 송신장치로 전송한다(S920). 즉, 수신장치는 송신장치로부터 수신한 암호문을 자신의 비밀 키(SK)로 복호화하여 G^x 및 r^A를 획득하고, G^{xy} (mod N) 연산에 의해 사전 마스터 비밀 키(K)를 생성한다. 그런 후, 수신장치는 G^y와 r^B를 송신장치의 공개키(PK)로 암호화하고, 송신장치로부터 수신한 r^A를 사전 마스터 비밀 키 K로 암호화한 다음 이를 메시지 인증 코드 H_B와 함께 송신장치로 전송한다.

[0084] 단계 S920이 수행되면, 송신장치는 자신이 생성한 사전 마스터 비밀 키 K를 사용하여 수신장치로부터 받은 암호문(r)을 해독하여 r이 자신이 보낸 r^A임을 확인하고, K를 가지고 자신의 SK와 r^B 및 P를 암호화하여 수신장치로 전송한다(S930). 즉, 송신장치는 수신장치로부터 수신한 암호문을 자신의 비밀키 SK로 복호하여 G^y와 r^B를 획득하고, G^{xy} (mod N) 연산을 통해 사전 마스터 비밀키 K를 생성한다. 상술한 과정을 통해 송신장치와 수신장치가 Diffie-Hellman Key exchangece를 이용하여 사전 마스터 비밀키 K를 안전하게 공유할 수 있다. 그런 후, 송신장치는 생성된 사전 마스터 비밀 키 K를 사용하여 수신장치로부터 받은 암호문(r)을 해독하여 r이 자신이 보낸 r^A임을 확인하고, K를 가지고 자신의 SK와 r^B 및 P를 암호화하여 수신장치로 전송한다.

[0085] 단계 S930이 수행되면, 수신장치는 송신장치로부터 받은 암호문을 해독하여 r을 검증하고, 송신장치의 SK를 이용하여 HMAC 연산으로부터 획득한 메시지 인증 코드 H를 검증하며, 자신이 패턴 정보를 올바르게 수신했음을 나타내는 암호문을 송신장치로 전송한다(S940). 이때, 수신장치는 이전에 수신한 H_A를 HMAC 연산에서 획득한 H와 비교하여, 메시지 인증 코드를 검증하고, K를 사용하여 송신장치의 SK를 암호화한 암호문과 P를 송신장치의 SK로 서명한 P^{SK}를 송신장치로 전송하여, 자신이 패턴 정보 P를 올바르게 수신했음을 알린다.

[0086] 단계 S940이 수행되면, 송신장치는 수신장치로부터 받은 암호문을 통해 획득한 수신장치의 SK를 사용하여 HMAC 연산으로 획득한 H가 이전에 수신한 H_B와 동일한지 여부를 확인한다.

[0087] 상술한 과정을 통해 송신장치와 수신장치는 상호 인증을 수행할뿐만 아니라, 사전 마스터 비밀 키와 패턴 정보를 안전하게 공유한다(S950).

[0088] 도 10은 본 발명에 따른 PCB 운영모드의 성능을 ECB 운영모드와 비교한 그래프이다.

[0089] 도 10을 참조하면, PCB 운영모드가 블록단위 암호화 중에서 가장 빠른 ECB보다 암호화 복호화를 수행하는 속도가 더 빠르다는 것을 확인할 수 있다. 기본적으로 PCB와 ECB는 독립적인 구조를 가진다. 그러나 PCB는 ECB와 달리 서로 다른 길이의 키를 동시에 사용함으로써 생기는 패턴 특성 때문에 높은 안전성을 제공하는 것이 가능하

다. 즉, PCB는 기존의 대칭키 암호화 기법이 가지는 키 길이에 대한 안전성 문제를 보완하여 높은 수준의 안전성을 제공할 뿐만 아니라 동시에 암호화 복호화를 수행하는 속도가 블록단위 암호화 중에서 가장 빠르므로 실시간성을 제공하는 것이 가능하다.

[0090] 지금까지 참조한 도면과 기재된 발명의 상세한 설명은 단지 본 발명의 예시적인 것으로서, 이는 단지 본 발명을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0092] 100 : 블록 암호 장치

110 : 입력 처리부

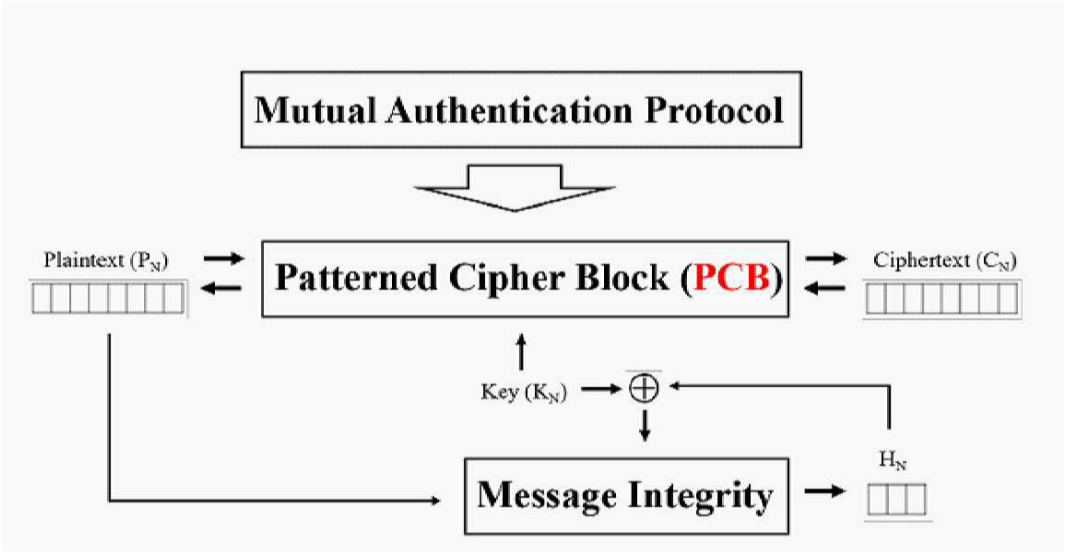
115 : 키 생성부

120 : 블록 암호화부

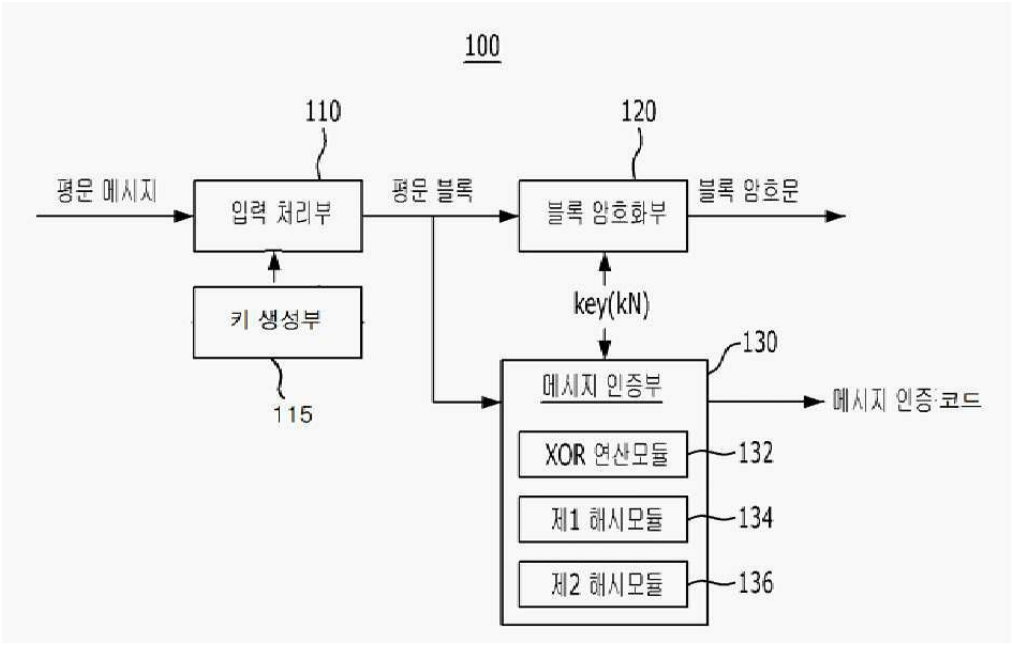
130 : 메시지 인증부

도면

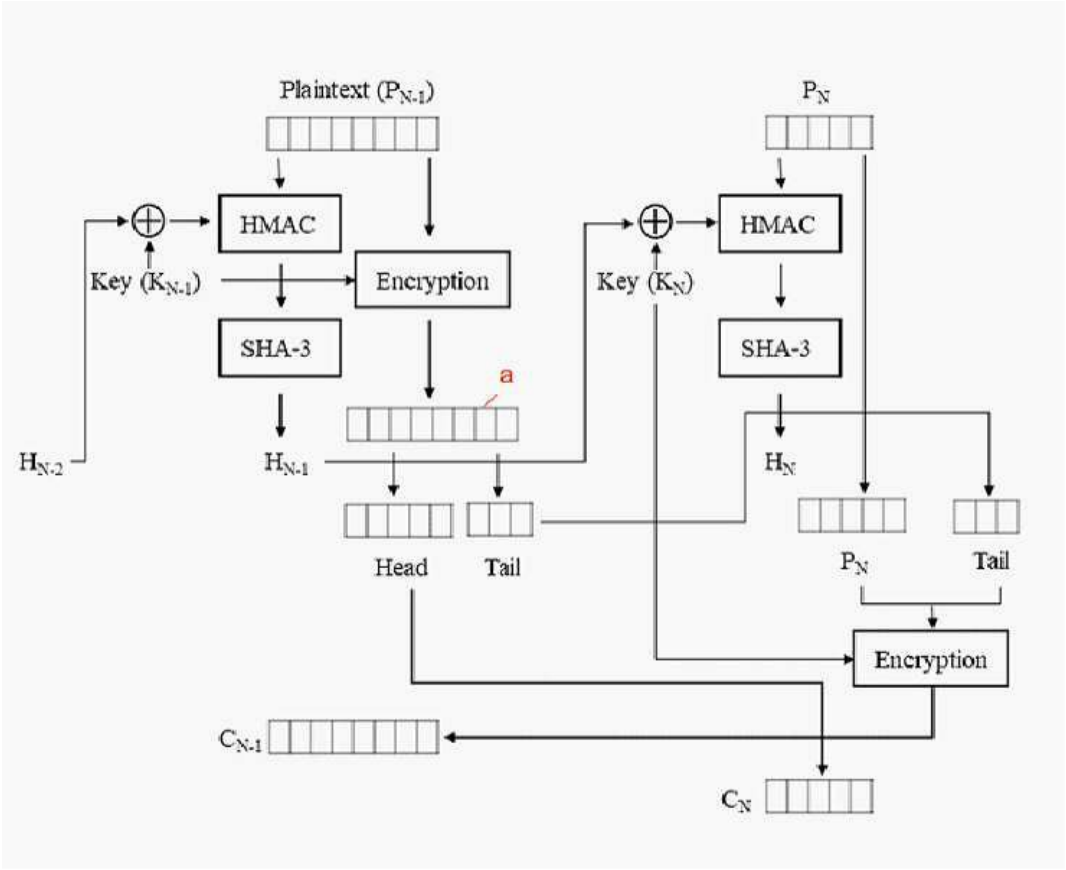
도면1



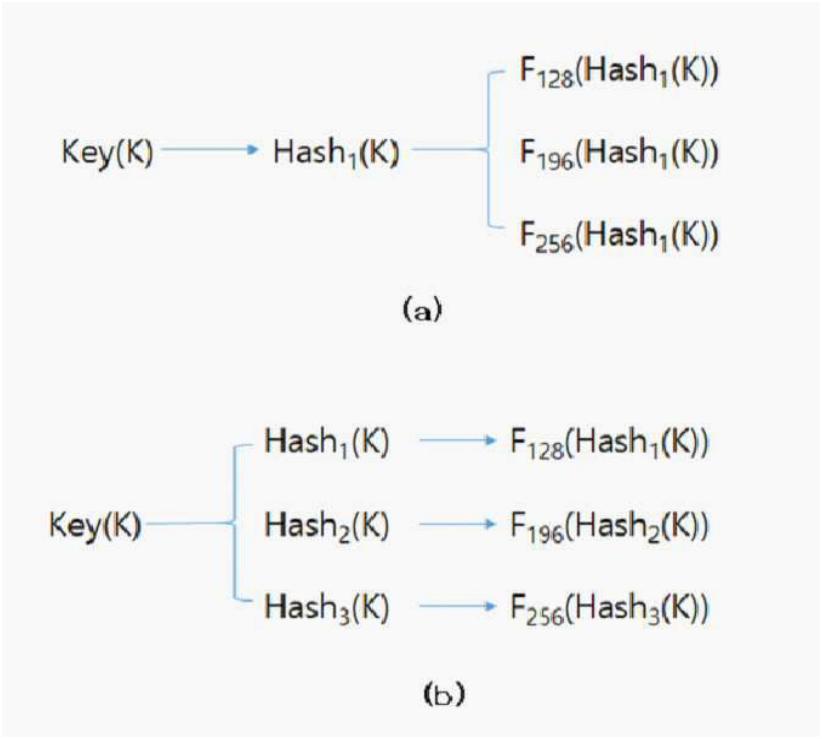
도면2


100
110
120
평문 메시지
입력 처리부
평문 블록
블록 암호화부
블록 암호문
키 생성부
115
key(kN)
130
메시지 인증부
메시지 인증코드
XOR 연산모듈
132
제1 해시모듈
134
제2 해시모듈
136

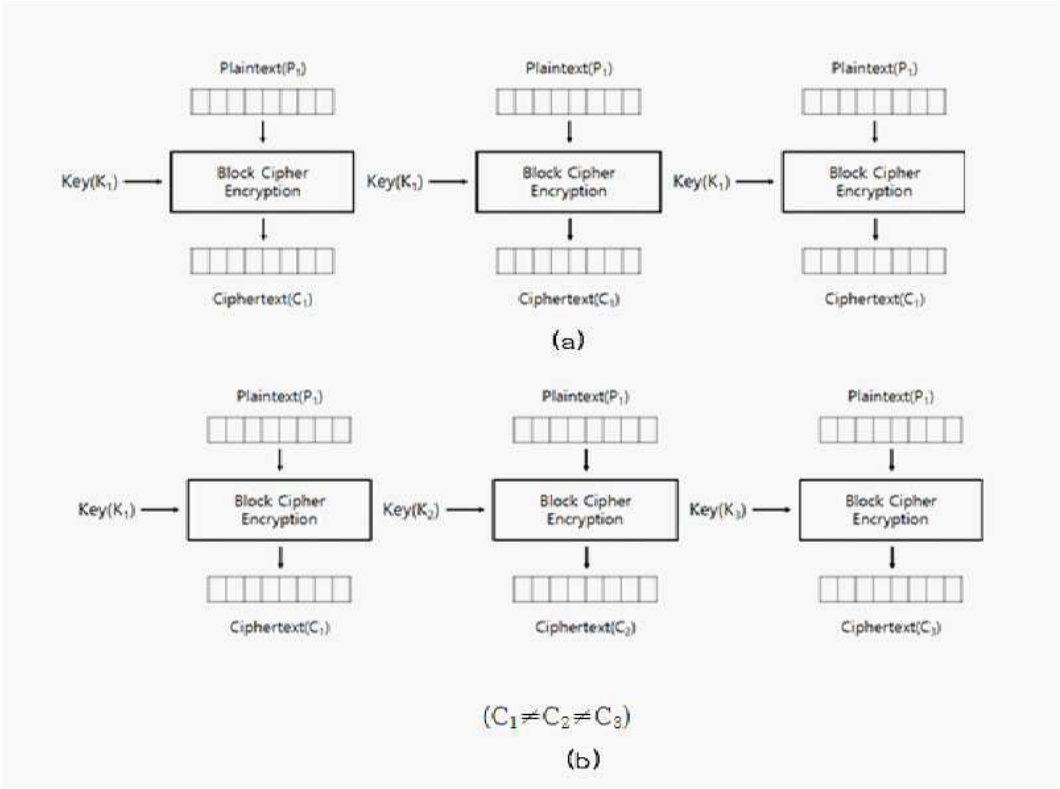
도면3


Plaintext (P_{N-1})
P_N
HMAC
HMAC
Key (K_{N-1})
Encryption
Key (K_N)
SHA-3
SHA-3
a
H_{N-2}
H_{N-1}
H_N
Head
Tail
P_N
Tail
Encryption
C_{N-1}
C_N

도면4


Key(K)
Hash1(K)
F128(Hash1(K))
F196(Hash1(K))
F256(Hash1(K))
(a)
Key(K)
Hash1(K)
Hash2(K)
Hash3(K)
F128(Hash1(K))
F196(Hash2(K))
F256(Hash3(K))
(b)

도면5


Plaintext(P1)
Key(K1)
Block Cipher Encryption
Ciphertext(C1)
Plaintext(P1)
Key(K1)
Block Cipher Encryption
Ciphertext(C1)
Plaintext(P1)
Key(K1)
Block Cipher Encryption
Ciphertext(C1)
(a)
Plaintext(P1)
Key(K1)
Block Cipher Encryption
Ciphertext(C1)
Plaintext(P1)
Key(K2)
Block Cipher Encryption
Ciphertext(C2)
Plaintext(P1)
Key(K3)
Block Cipher Encryption
Ciphertext(C3)
(C1≠C2≠C3)
(b)

도면6

Adversary $A^{E_k(LR(\cdot,\cdot,b))}$

$M_0 \leftarrow 0^n \| 1^n$;
$M_1 \leftarrow 0^n \| 0^n$;

$C[1]C[2] \leftarrow E_k(LR(M_0,M_1,b))$

If C[1] = C[2] then return 1 else return 0

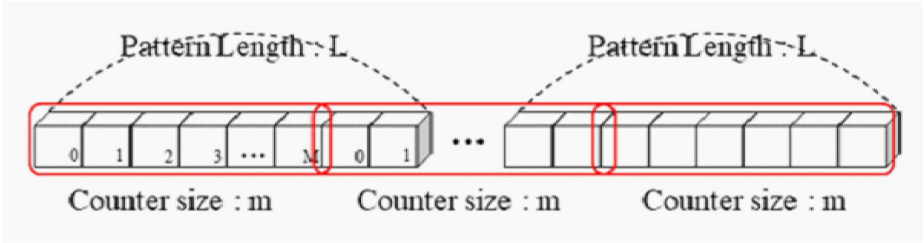
$E_k(LR(M_l,M_r,b)) \begin{cases} E_k(Ml)\ if\ b = 1 \\ E_k(Mr)\ if\ b = 0 \end{cases}$

C[i] denotes the i-th block of a string C.

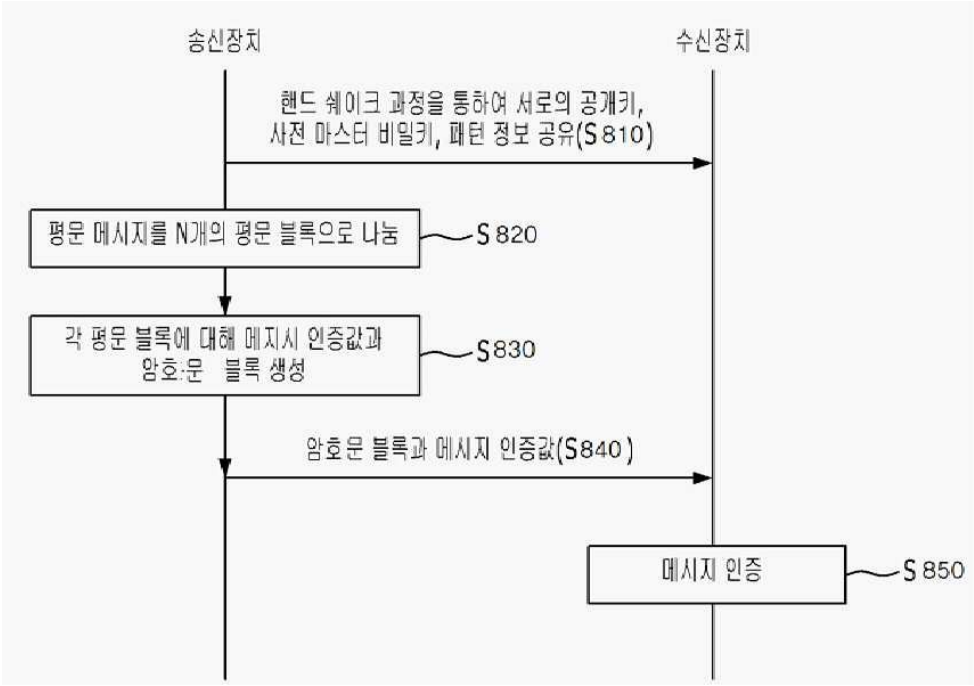
In the right word, $C[1]C[2] = E_k(0^n)\|E_k(0^n)$ and $E_k(0^n)$ is not equal to $E_k(0^n)$
In the left word, $C[1]C[2] = E_k(0^n)\|E_k(1^n)$ and $E_k(0^n)$ is not equal to $E_k(1^n)$

$Adv_{SE}^{IND-CPA}(A) = 0 - 0 = 0$, Because of different key

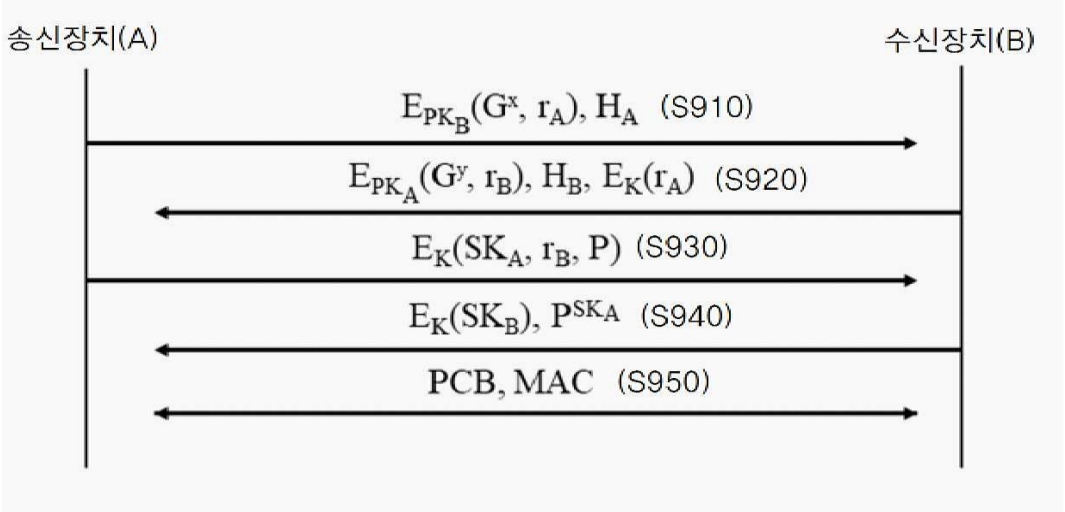
도면7



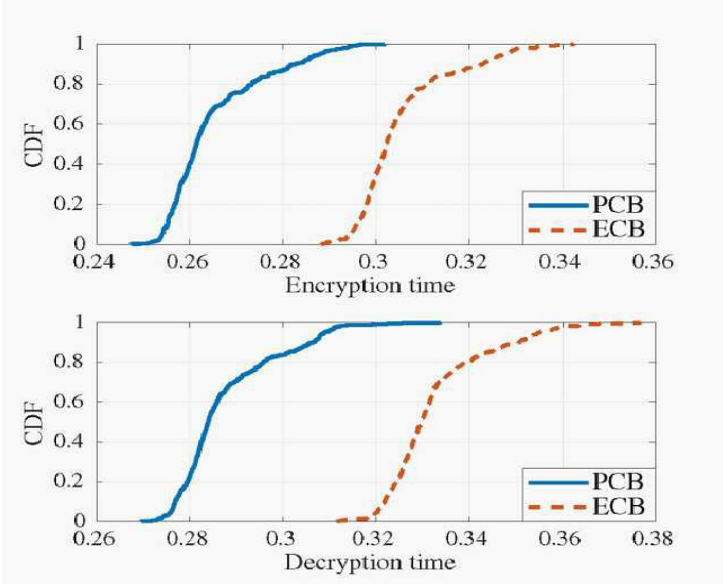
도면8



도면9


송신장치(A)
수신장치(B)
$E_{PK_B}(G^x, r_A), H_A$ (S910)
$E_{PK_A}(G^y, r_B), H_B, E_K(r_A)$ (S920)
$E_K(SK_A, r_B, P)$ (S930)
$E_K(SK_B), P^{SK_A}$ (S940)
PCB, MAC (S950)

도면10


CDF
Encryption time
PCB
ECB
CDF
Decryption time
PCB
ECB