(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0090438
(43) 공개일자 2015년08월06일

(51) 국제특허분류(Int. Cl.)
G09C 1/06 (2006.01)
(21) 출원번호 10-2014-0011088
(22) 출원일자 2014년01월29일
심사청구일자 없음

(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
김주한
대전광역시 유성구 노은로 353 송림마을아파트 305동 1803호
이승광
대전광역시 유성구 가정로 270 한국전자통신연구원 기숙사 2관 304호
최두호
충청남도 천안시 동남구 용곡2길 한라비발디아파트 118동 701호
(74) 대리인
특허법인지명

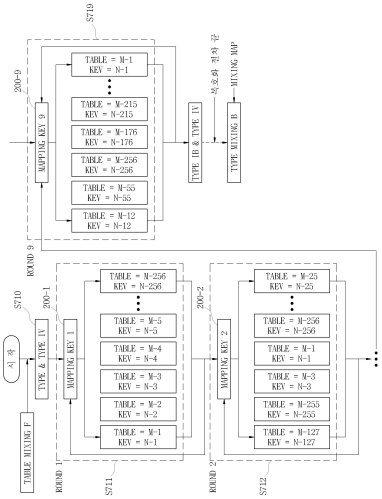
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **화이트박스 암호 장치 및 그 방법**

(57) 요 약

화이트 박스 암호화 방법이 개시된다. 이 방법은 다수의 화이트 박스 암호 테이블들을 이용한 암호화 연산을 다수의 라운드별로 수행하는 과정 및 각 라운드 별로 출력되는 결과 테이블들의 배열을 믹싱하는 과정을 포함한다.

대 표 도 - 도7



명 세 서

청구범위

청구항 1

화이트 박스 암호 테이블들을 이용한 암호화 연산을 다수의 라운드 별로 수행하는 연산부; 및

각 라운드별로 출력되는 결과 테이블들의 배열을 믹싱하는 테이블 믹싱부

를 포함하는 화이트 박스 암호화 장치.

청구항 2

제1항에 있어서, 믹싱된 상기 결과 테이블들의 배열은,

사전에 설정된 매핑 키 정보에 의해 정상 배열 순서로 복호화되는 것인 화이트박스 암호화 장치.

청구항 3

제2항에 있어서, 상기 매핑 키 정보는, 각 라운드별로 구분되는 다수의 매핑 키를 포함하는 것인 화이트 박스 암호화 장치.

청구항 4

제2항에 있어서, 상기 테이블 믹싱부는,

각 라운드 별로 출력되는 결과 테이블들의 배열을 특정 연산에 따라 무작위로 믹싱하고,

상기 매핑 키 정보는,

상기 특정 연산에 대한 반대 연산에 대한 정보를 포함하는 것인 화이트 박스 암호화 장치.

청구항 5

제2항에 있어서, 상기 테이블 믹싱부는,

각 라운드 별로 출력되는 결과 테이블들의 배열을 랜덤하게 믹싱하고,

상기 매핑 키 정보는,

랜덤하게 믹싱된 상기 결과 테이블들의 배열 순서에 대한 정보를 포함하는

것인 화이트 박스 암호화 장치.

청구항 6

제2항에 있어서, 상기 매핑 키 정보는,

외부 메모리에 저장되어 관리되는 것인 화이트박스 암호화 장치.

청구항 7

다수의 화이트 박스 암호 테이블들을 이용한 암호화 연산을 다수의 라운드 별로 수행하는 과정; 및

각 라운드 별로 출력되는 결과 테이블들의 배열을 믹싱 하는 과정;

을 포함하는 화이트 박스 암호화 방법.

청구항 8

제7항에 있어서, 믹싱된 상기 결과 테이블들의 배열은,

사전에 설정된 매핑 키 정보에 의해 정상 배열 순서로 복호화 되는 것인 화이트박스 암호화 방법.

청구항 9

제8항에 있어서, 상기 매핑 키 정보는, 각 라운드 별로 구분되는 다수의 매핑 키를 포함하는 것인 화이트 박스 암호화 방법.

청구항 10

제8항에 있어서, 상기 테이블 믹싱부는,

각 라운드 별로 출력되는 결과 테이블들의 배열을 특정 연산에 따라 무작위로 믹싱하고,

상기 매핑 키 정보는,

상기 특정 연산에 대한 반대 연산에 대한 정보를 포함하는 것인 화이트 박스 암호화 방법.

청구항 11

제8항에 있어서, 상기 테이블 믹싱부는,

각 라운드 별로 출력되는 결과 테이블들의 배열을 랜덤하게 믹싱하고,

상기 매핑 키 정보는,

랜덤하게 믹싱된 상기 결과 테이블들의 배열 순서에 대한 정보를 포함하는

것인 화이트 박스 암호화 방법.

청구항 12

제8항에 있어서, 상기 매핑 키 정보는,

외부 메모리에 저장되어 관리되는 것인 화이트 박스 암호화 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 화이트 박스 암호를 보다 안전하게 계량한 암호 기술에 관한 것이다.

배 경 기 술

[0002] 암호 기술에는 화이트 박스(White Box)와 블랙 박스(Black Box) 암호화 기술이 있다. 블랙 박스 암호화 알고리

즘은 예전 기술이고 화이트 박스 기술은 최신의 기술이고 더 안전한 기술이다.

[0003] 암호 기술은 쉽게 말해 평문(Plain text)을 암호문(Cipher text)으로 바꾸는 기술이다. 즉 암호화 기술은 평문을 암호화하여, 크래커가 알 수 없도록 하는 것이다. 이러한 암호화 기술은 소프트웨어 코드일 수도 있고, 하드웨어 장치일 수도 있다. 어떤 형태의 암호화 기술이건, 블랙 박스 또는 화이트 박스에 기반한다.

[0004] 블랙 박스에 기반한 암호 기술은 평문을 암호화하는 과정에서 암호화 키(Key)가 필요하다. 이 암호화 키는 블랙 박스로 가정한 암호화 장치 내부에 들어 있다. 블랙 박스는 그 안이 들여다 보이지 않음을 의미이다. 즉, 블랙 박스에 기반한 암호화 장치의 설계는 크래커가 이 암호화 장치 내부를 들여다 볼 수 없다는 가정에서 출발한다. 따라서 크래커는 블랙 박스 기반의 암호화 장치에 입력되는 평문과 출력되는 암호화문만을 볼 수 있다. 아마도 크래커는 두 개의 입출력 값을 계속해서 관찰하여 어떤 패턴을 알아내고자 할 것이다. 블랙 박스는 단순히 암호화 장치의 설계자가 이 암호화 장치 자체는 완벽히 안전하다고 가정하는 것이다. 즉 블랙 박스로 가정하는 것이다. 따라서 만에 하나, 이 암호화 장치 자체가 뚫려 버리면 암호화 키가 누출될 수 있다. 암호화 키가 누출되면 모든 암호화 과정은 크래커에게 완전히 공개된다.

[0005] 이보다 더 발전된 방식이 화이트 박스 암호 기술이다. 화이트 박스를 해석하면 하얀 색의 상자이지만, 다르게는 투명한 상자로 해석될 수 있다. 화이트 박스 암호화 기술은 크래커가 어떤 방법을 써서든지 결국에 암호 장치 내부를 들여다 볼 수 있다는 가정에서 출발한다. 결국 크래커가 암호화 장치 내부를 볼 수 있다면, 암호화 키(Key)를 획득할 수 있기 때문에, 설계자는 더 많은 사항을 고려해야 한다. 암호화 장치를 화이트박스로 가정하면 암호화 키(Key)를 손쉽게 장치 내에 저장할 수 없다. 따라서 일반적인 화이트 박스에서는 암호화 키가 그대로 존재하지 않고 복잡한 암호화 연산 알고리즘과 뒤섞여 존재한다. 따라서 암호화 키를 따로 얻을 수가 없다. 또한 이 알고리즘은 되돌리기(invert)가 어려운 알고리즘이다. 따라서 결과값을 갖고 원본값이나 암호화 키를 추측하기 어렵다.

[0006] 블랙 박스 기반의 암호화 기술은 Y = algorithm1 (x, key1)와 같은 수식으로 표현할 수 있고, 화이트 박스에서의 암호화 과정은 Y = algorithm2 (x)와 같은 수식으로 표현할 수 있다. 즉, 입력 정보인 암호 키를 암호 알고리즘 내부에서 쉽게 유출할 수 없는 형태로 안전하게 숨길 수 있다면 화이트 박스 기반으로 구동하는 암호화 연산 알고리즘을 해커가 모니터링 하더라도 암호 키를 유추하는 것은 어렵다.

[0007] 이와 같이, 현재의 화이트박스 암호는 암호 키를 사용하지 않기 때문에, 암호키가 누출되지 않는 장점을 가지며 또한 표준 암호 기술과 호환되는 장점이 있다. 그러나, 암호키가 숨겨진 화이트 박스 암호 알고리즘 자체가 누출되면 그를 통해 암호문을 복호화할 수 있어 보안성이 취약한 곳에서는 사용하기 힘들다.

발명의 내용

해결하려는 과제

[0008] 따라서, 본 발명의 목적은 암호 키가 숨겨진 화이트 박스 암호 알고리즘 자체가 누출되는 상황에서도 보안성 및 안전성을 유지하는 화이트박스 암호 장치 및 그 방법을 제공하는 데 있다.

과제의 해결 수단

[0009] 상기와 같은 목적을 달성하기 위한, 본 발명의 일면에 따른 화이트 박스 암호 장치는, 화이트 박스 암호 테이블들을 이용한 암호화 연산을 다수의 라운드별로 수행하는 연산부 및 각 라운드별로 출력되는 결과 테이블들의 배열을 믹싱 하는 테이블 믹싱부를 포함한다.

[0010] 본 발명의 다른 일면에 따른 화이트 박스 암호 방법은 다수의 화이트 박스 암호 테이블들을 이용한 암호화 연산을 다수의 라운드별로 수행하는 과정 및 각 라운드별로 출력되는 결과 테이블들의 배열을 믹싱 하는 과정을 포함한다.

발명의 효과

[0011] 본 발명에 의하면, 화이트박스 암호 알고리즘으로 구현된 코드(또는 테이블) 등이 누출되는 경우에도 각 라운드

사이에서 연산되는 연산 정보와 관련된 반대 연산정보를 가지고 있어야 정상적인 암호화 및 복호화가 수행 가능함으로써, 보다 안전한 화이트 박스 암호 기술을 제공할 수 있다.

도면의 간단한 설명

[0012] 도 1은 본 발명에 적용되는 화이트 박스 암호의 기본 원리를 설명하기 위한 도면이다.

도 2는 본 발명에 적용되는 화이트 박스 AES의 연산 순서를 보여주는 도면이다.

도 3은 도 2에 도시된 테이블들 중 Type2 테이블 구조를 보여주는 도면이다.

도 4는 도 2에 도시된 테이블들 중 Type1b 테이블 구조를 보여주는 도면이다.

도 5는 도 2에 도시된 테이블들 중 Type1b 테이블 구조를 보여주는 도면이다.

도 6은 본 발명의 일 실시 예에 따른 화이트 박스 암호 장치를 구성을 보여주는 블록도이다.

도 7은 본 발명의 일 실시 예에 따라 동적으로 변환되는 화이트 박스 암호를 복호화 하는 절차를 보여주는 도면이다.

도 8은 본 발명의 실시예에 따른 화이트 암호화 장치가 적용될 수 있는 컴퓨터 시스템을 나타내는 개략적인 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0013] 본 발명에서는 화이트박스 암호 알고리즘으로 구현된 코드(또는 테이블) 등이 누출되는 경우에도 각 라운드 사이에서 연산되는 연산 정보와 관련된 반대 연산정보를 가지고 있어야 정상적인 암호화 및 복호화가 가능한 방법을 제공한다. 이하, 첨부된 도면을 참조하여 본 발명의 일 실시 예에 대해 상세히 설명하기로 한다.

[0014] 본 발명에 적용되는 화이트 박스 암호의 기본 원리

[0015] 도 1은 본 발명에 적용되는 화이트 박스 암호의 기본 원리를 설명하기 위한 도면이다.

[0016] 화이트 박스 암호의 기본 원리는 도 1에 도시된 바와 같다. 전통적인 암호 메커니즘은 암호 키가 블랙박스 장치(믿을 수 있는 단말)에서 안전하게 유지 관리된다는 가정 하에 동작된다. 반면, 화이트박스 암호 메커니즘은 암호 키가 소프트웨어로 구현된 암호 알고리즘 속에 섞여 있기 때문에(obfuscation), 공격자가 암호 키를 쉽게 볼 수 없다는 가정 하에서 동작된다. 즉, 화이트박스 암호는 알고리듬을 큰 룩업 테이블로 만들고 그 안에 암호 키를 소프트웨어로 구현된 암호 알고리듬과 뒤섞인 상태로 숨겨둠으로써, 내부의 동작을 분석하더라도 암호 키를 쉽게 유추하지 못하도록 하는 기법이다. 암호 알고리듬을 하나의 큰 룩업 테이블로 만들면 암호 키를 숨기는 것이 용이하지만 테이블 크기가 지나치게 커져서 비현실적이므로, 테이블을 암호학적인 기법으로 적절히 분리하되 암호화 연산의 중간값이 노출되지 않도록 디코딩과 인코딩 과정을 수행하도록 하면 된다.

[0017] 도 1에 도시된 바와 같이, 기본적인 화이트 박스 암호의 기본 원리는 인코딩 과정(M_i)과 디코딩 과정(Mi^{-1})이 별도의 테이블에서 계산되므로 중간값이 노출되지 않으면서도 결국은 인코딩과 디코딩이 상쇄되면서 원래의 암호화 동작(X_i)만 수행하는 결과와 동일하게 된다.

[0018] 본 발명에 적용되는 화이트박스 AES(Advanced Encryption Standard) 동작 메커니즘

[0019] 본 발명에 적용되는 화이트박스 AES는 열을 시프트하는 ShiftRows, 라운드키를 추가하는 AddRoundKey, 키를 대체하는 SubBytes, 행을 믹싱하는 MixColumns의 반복으로 이루어진 라운드 연산을 수행한다. 즉, 본 발명에 적용되는 화이트박스 AES에서는, 최초의 key whitening을 위한 AddRoundKey 연산을 첫 번째 라운드 내에서 수행하고, 첫 번째 라운드의 AddRoundKey 연산은 다음 라운드 연산에서 수행하도록 함으로써, 매 라운드는 AddRoundKey 연산에서 시작해서 MixColumns 연산으로 끝난다. AES에서 라운드 연산을 Mix-Columns로 끝나도록

한 이유는 WB-AES 구현시 하나의 큰 룩업 테이블이 아닌, 여러 개의 작은 룩업 테이블로 만드는 과정과 관계가 있다. ShiftRows는 AddRoundKey 및 Sub-Bytes와 순서가 바뀌어도 연산 결과는 동일하므로 구현의 편의를 위해서 매 라운드 연산의 맨 앞에서 수행한다.

[0020] 도 2는 본 발명에 적용되는 화이트 박스 AES의 연산 순서를 보여주는 도면이다.

[0021] 본 발명에 적용되는 화이트 박스 AES는 Type1a, Type1b, Type2, Type3, Type4의 5가지 테이블로 구성되어 있는데, 각 테이블의 입력 데이터와 출력 데이터는 각각 2개의 nibble 입력(4-bit input)을 치환(permutation)하여 디코딩하고 인코딩하는 비선형 변환을 통해서 테이블 내부 연산이 쉽게 드러나지 않도록 구성된다.

[0022] 도 2에 도시된 바와 같이, 5가지 테이블을 이용한 AES의 연산 순서는 initial round, 9 round 및 final round를 포함하는 11개의 라운드로 이루어질 수 있다. 특히, 도 2에 도시된 연산 순서에서 Type1a, Type1b, Type2, Type3 테이블 연산 후에는 Type4 테이블 연산이 수행된다. 이는 Type1a, Type1b, Type2, Type3 내에서 수행하는 행렬 곱셈(mixing bijection) 결과들을 모아서 행렬 곱셈의 마무리를 위한 XOR 연산을 할 필요가 있는데, Type4 테이블에서 이러한 XOR 연산을 수행하기 때문에 Type4 테이블이 다른 테이블의 뒤에 따라오게 된다.

[0023] 도 3은 도 2에 도시된 테이블들 중 Type2 테이블 구조를 보여주는 도면이다.

[0024] 도 3을 참조하면, Type2 테이블에서는 AES 라운드 연산의 대부분이 수행된다. Type2 테이블에서는 입력 데이터의 디코딩, 출력 데이터의 인코딩 외에도 라운드 연산 전/후에 8×8 가역행렬을 곱해주는 8 × 8 mixing bijection 연산과 32×32 가역행렬을 곱해주는 32×32 mixing bijection 연산이 존재한다. 이들 행렬을 라운드 연산 전/후에 곱해줌으로써 라운드 연산의 중간 데이터 및 키를 공격자로부터 안전하게 숨길 수 있다.

[0025] Type3 테이블에서는 Type2 테이블에서 곱해진 8×8 행렬(8×8 mixing bijection)과 32×32 행렬(32×32 mixing bijection)에 대한 역행렬을 곱해줌으로써 Type2, Type4, Type3, Type4 테이블 연산을 모두 수행하였을 때, AES의 라운드 연산만 남을 수 있도록 한다.

[0026] Type1a 테이블과 Type1b 테이블은 AES의 안전성을 높이기 위하여 128비트 입력 및 출력 데이터에 128×8 가역행렬을 곱해주는 연산을 수행한다. 또한 Type1b 테이블은 앞서 기술한 출력 데이터가 직접 드러나지 않도록 보호해주는 기능 외에도 AES의 마지막 라운드 연산을 수행한다.

[0027] 도 4는 도 2에 도시된 테이블들 중 Type1b 테이블 구조를 보여주는 도면이다. 도 5는 도 2에 도시된 테이블들 중 Type1b 테이블 구조를 보여주는 도면이다.

[0028] 도 4 및 도 5를 함께 참조하면, AES의 암호화 연산은 128비트 입력 데이터에 대한 암호화 연산을 수행하는 경우, Add-RoundKey 연산을 수행한 후 10회의 라운드 연산을 수행한다. AES에서는 최초의 AddRoundKey 연산을 첫 번째 라운드 연산을 수행하는 Type2 테이블 내에서 수행하고, 첫 번째 라운드의 AddRoundKey 연산은 두 번째 라운드 연산을 수행하는 Type2 테이블 내에서 수행하므로, 마지막 라운드 연산을 수행하는 Type1b 테이블에서는 9번째 라운드용 AddRound-Key 연산과 마지막 라운드용 AddRoundKey 연산을 함께 수행한다.

[0029] 또한 Type1b 테이블의 8×8 mixing bijection 연산은 9번째 라운드 연산을 수행했던 테이블들 중 Type3 테이블에서 8×8 역행렬을 미리 곱해주고, Type1b 테이블에서 이의 역행렬인 8×8 행렬을 곱해주는 연산을 수행함으로써 서로 상쇄될 수 있도록하였다. 앞서 기술하였듯이, Type3 테이블에서는 32×32 역행렬과 8×8 역행렬을 곱해주는 기능을 수행하는데, 32×32 역행렬은 동일 라운드의 Type2 테이블에서 곱해준 32×32 행렬에 대한 역행렬을 곱해주는 것이고, 8×8 역행렬은 다음 라운드의 Type2(마지막 라운드의 경우 Type1b) 테이블에서 곱해줄 8×8 행렬에 대한 역행렬을 곱해주는 것이다. 또한 첫 번째 라운드 연산에서 Type2 테이블에서 곱해줬던 8×8 행렬에 대한 역행렬은 Type1a 테이블에서 미리 곱해주기 때문에 서로 상쇄되어 없어질 수 있다.

[0030] 이상의 설명한 각 테이블 구조를 도 2에 도시된 연산 순서에 따라 연산하면, 화이트 박스 암호가 생성된다. 도 2에 도시된 연산 순서에 따라 진행하도 보안이 취약한 디바이스에서는 위의 화이트 박스 알고리듬(화이트 박스 코드)이 쉽게 누출될 수 있다. 공격자는 암호키는 모르지만 도청한 암호문을 누출된 화이트 박스 코드를 통해 바로 복호화가 가능할 것이다. 이에 본 발명에서는 위와 같은 코드 리프팅 공격을 막기 위하여 화이트 박스 암호 구현이 동적으로 변화하고, 동적으로 변환된 정보 그 자체를 별도로 관리하여 화이트 박스 암호에 대한 보안성을 높이는 방안을 제안한다.

[0031] 도 6은 본 발명의 일 실시 예에 따른 화이트 박스 암호 장치를 구성을 보여주는 블록도이다.

[0032] 도 6을 참조하면, 본 발명의 일 실시 예에 따른 화이트 박스 암호 장치(300)는 화이트 박스 암호 생성부(100)와

저장부(200)를 포함한다. 화이트 박스 암호 생성부(100)는 도 1 내지 도 5를 참조하여 설명한 바와 같이, 화이트 박스 암호를 생성하기 위해 다수의 라운드 연산을 수행한다. 이를 위해, 화이트 박스 암호 생성부(100)는 제1 내지 제10 라운드 연산부(101 ~ 110)를 포함한다. 각 라운드 연산부는 ShiftRows, AddRoundKey, SubBytes, MixColumns의 반복으로 이루어진 라운드 연산을 수행하며, 각 라운드 연산부에서 수행되는 연산 과정 및 연산 순서는 도 1 내지 도 5에서 설명한 바와 같다. 또한 화이트 박스 암호 생성부(100)는 화이트 박스 암호 생성을 동적으로 변화시키기 위해, 라운드 연산부들(101 ~ 110) 사이에 구비된 제1 내지 제9 테이블 믹싱부(101-1 ~ 109-9)를 포함한다. 구체적으로, 제1 테이블 믹싱부(101-1)는 제1 라운드 연산부(101)로부터 도 2의 첫 번째 연산 순서(1st round)에 따라 연산된 다수의 결과 테이블들을 입력 받고, 입력된 결과 테이블들을 무작위로 믹싱하는 연산을 수행한다. 예컨대, ShiftRows가 1바이트 단위의 연산되는 경우, 제1 라운드 연산부(101)는 256개의 결과 테이블들을 출력하며, 제1 테이블 믹싱부(101-1)는 제1 라운드 연산부(101)로부터 출력되는 256개의 결과 테이블들을 무작위로 믹싱하는 연산을 수행한다. 무작위로 믹싱된 256개의 결과 테이블들은 제2 라운드 연산부(102)로 입력되고, 마찬가지로, 제2 라운드 연산부(102)는 무작위로 믹싱된 256개의 결과 테이블들을 도 2의 두 번째 연산 순서(2nd round)에 따라 연산하여 256개의 결과 테이블들을 제2 테이블 믹싱부(102-2)로 출력한다. 제2 테이블 믹싱부(102-2)는 제1 테이블 믹싱부(101-1)와 마찬가지로 256개의 결과 테이블들 무작위 믹싱하는 연산을 수행하고, 이를 도 6에는 도시되지 않은 제3 라운드 연산부로 출력한다. 이러한 진행 순서에 따라 제9 라운드 연산부(109)는 도시되지 않은 제8 테이블 믹싱부에 의해 무작위로 믹싱된 제8 라운드 연산부의 결과 테이블을 도 2의 아홉 번째 연산 순서(9th round)에 따라 연산된 256개의 결과 테이블들을 제9 테이블 믹싱부(109-9)로 출력한다. 제9 테이블 믹싱부(109-9) 또한 제9 라운드 연산부(109)에 의해 연산된 256개의 결과 테이블들을 무작위로 믹싱하여, 이를 제10라운드 연산부(110)로 출력한다. 제10 라운드 연산부(110)는 도 2의 연산 순서에 따라 연산을 수행하여 암호화가 이루어진 암호문 출력 데이터를 출력하게 된다. 이와 같이, 본 발명의 일 실시 예에 따른 화이트 박스 암호 장치는 각 라운드 별 연산 결과에 해당하는 결과 테이블들을 무작위로 믹싱함으로써, 화이트박수 암호의 생성 과정이 동적으로 변화한다.

[0033] 한편, 동적으로 변화된 화이트박수 암호를 복호화하기 위해, 각 라운드 연산 후에 무작위로 믹싱된 결과 테이블들의 배열을 정상적으로 복원하기 위한 매핑키(mapping key) 정보가 제공된다. 이 매핑키 정보는 도 6에 도시된 저장부(200)에 저장되어 별도로 관리된다. 이러한 매핑키 정보는 무작위로 믹싱된 결과 테이블들을 복호화하기 위해 각 라운드 별로 구분될 수 있으며, 각 라운드 별 구분될 매핑키를 이용하여 각 라운드별 무작위로 믹싱된 결과 테이블들의 배열이 정상적으로 복원될 수 있다.

[0034] 이와 같이, 각 라운드 연산 후에 무작위로 믹싱된 결과 테이블들의 배열을 정상적으로 복원할 수 있는 매핑키 정보가 없다면, 화이트박스 암호 코드가 자체가 누출되어도, 공격자는 중간에 임의의 연산과 관련된 정보 즉, 매핑키 정보를 보유하고 있지 않기 때문에 누출된 화이트박스 암호 코드를 이용하여 암호문을 복호화할 수 없게 된다.

[0035] 한편, 도 6에서는 다수의 라운드 연산부 및 다수의 테이블 믹싱부가 각각 분리된 형태로 도시되어 있으나, 설명의 이해를 돕기 위해, 기능적으로 분리된 예를 도시한 것다. 따라서, 다수의 라운드 연산부와 다수의 테이블 믹싱부는 각각 하나의 라운드 연산부 및 하나의 테이블 믹싱부로 구현될 수 있다.

[0036] 도 7은 본 발명의 일 실시 예에 따라 동적으로 변환되는 화이트 박스 암호를 복호화 하는 절차를 보여주는 도면이다. 특별히 한정하지 않는 이상 아래의 각 단계의 수행 주체는 도 6에 도시된 화이트 박스 암호 생성부로 가정한다.

[0037] 도 7을 참조하면, S710에서, 제1 라운드 연산부(101)에서 Type 1A 테이블 및 Type IV 테이블을 입력받는 과정이 수행된다.

[0038] S711에서는, 제1 테이블 믹싱부(101-1)에 의해 제1 라운드(Round 1)에서 무작위로 믹싱된 256개의 테이블들(Table = m-1 ~ m-256)의 배열을 복원하는 과정이 수행된다. 구체적으로, 도 6에 도시된 매핑키 정보에 포함된 제1 매핑키(200-1)를 이용하여 제1 라운드(Round 1)에서 무작위로 믹싱된 256개의 테이블들의 배열을 복원한다. 예컨대, 제1 매핑키(200-1)는 제1 라운드(Round 1)에서 256개의 테이블들을 무작위로 믹싱하는 연산에 대한 반대 연산 정보를 포함할 수 있다. 각 테이블이 n-1부터 n-256까지 넘버링된 키(key = n-1 ~ n-256)로 표현할 때, 넘버링된 각 테이블이 임의의 연산에 따라 무작위로 믹싱되면, 넘버링된 키들 또한 상기 임의의 연산에 따라 믹싱된다. 따라서, 상기 임의의 연산에 대한 반대 연산을 통해 상기 임의의 연산에 따라 믹싱된 키 배열을 원래의 키 배열로 복원한다.

[0039] S712에서, 제2 라운드(Round 2)에서 무작위로 믹싱된 256개의 테이블들(Table = m-1 ~ m-256)의 배열을 복원하

는 과정이 수행된다. 이 복원 과정은 제1 매핑키(200-1)를 이용하여 복원되며, 그 복원 과정은 S711에서 수행된 방식과 동일하다.

[0040] 테이블의 배열을 복원하는 과정은 각 라운드별로 진행된다. S719에서, 제2 라운드(Round 9)에서 무작위로 믹싱된 256개의 테이블들(Table = m-1 ~ m-256)의 배열을 복원하는 과정이 수행된다. 이 복원 과정은 제9 매핑키(200-1)를 이용하여 복원되며, 각 라운드별 테이블의 배열이 복원되는 과정이 종료된다.

[0041] 이후, 복원된 테이블이 제10 라운드 연산에 따른 연산 순서(Type IB 테이블 -> Type IV 테이블)에 따라 연산되고, 일련의 복호화 절차가 종료된다.

[0042] 이와 같이, 라운드 단위의 결과 테이블을 믹싱하고, 믹싱과 관련된 정보를 찾아갈 수 있도록 해당 매핑 키(210)가 별도로 관리된다. 이렇게 함으로써, 이 mapping key를 가지고 있어야 테이블들의 배열이 정상적으로 이루어져 암호화/복호화가 가능함으로써, 하이트박스 암호 코드가 누출되어도 중간에 임의의 연산과 관련된 정보를 가지고 있지 않는 한 정상적인 암호화/복호화를 진행할 수 없으므로, 보다 안전한 화이트 박스 암호 기술을 제공할 수 있게 된다.

[0043] 한편, 다른 실시 예에서, 테이블의 배열을 복원하는 절차는 라운드 단위가 아니라 Type별 (Type 1A, Type IV, Type II, Type IV 등)로 진행할 수 있다. 그리고 테이블들의 배열을 믹싱하는 과정에서, 특정 연산 방법이 사용되는 경우, 매핑키(mapping key)는 연산 정보를 내포하는 수준으로 간단히 구현될 수 있다. 만일 그렇지 않고 랜덤하게 믹싱하는 경우에는 매핑키는 배열 정보를 포함한다.

[0044] 도 8은 본 발명의 실시예에 따른 화이트 암호화 장치가 적용될 수 있는 컴퓨터 시스템을 나타내는 개략적인 블록도이다.

[0045] 도 8에 도시된 바와 같이, 이 컴퓨터 시스템은(500)은 디스플레이(512), 키보드(514), 컴퓨터(516)와 외부 장치(518)를 포함한다. 상기 컴퓨터(516)는 중앙처리장치(central processing unit; CPU)와 같은 하나 이상의 프로세서나 마이크로 프로세서들을 포함한다. CPU(520)는 수학적 계산과 내장메모리(522), 오히려 램(random access memory; RAM)과/또는 롬(read only memory; ROM), 부가적인 메모리(524)에 저장된 소프트웨어를 실행하는 기능을 제어하는 역할을 수행한다. 부가적인 메모리(524)는 예를 들어, 대용량 메모리 저장장치(mass memory strage), 하드 디스크 드라이브(hard disk drivers), 플로피 디스크 드라이브(floppy disk drives), 마그네틱 테이브 드라이브(magnetic tape drives), 컴팩트 디스크 드라이브(compact disk drives), 프로그램 카트리지(program cartridge)와 카트리지 인터페이스(cartridge interfaces), 비디오 게임 장치에서 발견되는, EPROM 또는 PROM, 또는 이와 비슷한 기술로 알려진 저장 매체와 같이 제거할 수 있는 메모리 칩을 포함한다. 도 8에서와 같이, 이런 부가적인 메모리(524)는 물리적으로 컴퓨터(516)의 내부 또는 외부에 있게 된다.

[0046] 컴퓨터 시스템(500)은 또한 컴퓨터 프로그램들 또는 다른 명령들이 로드될 수 있도록(load)하는 다른 비슷한 방법들을 포함한다. 그러한 방법들은, 예를 들어, 커뮤니케이션 인터페이스(526)가 소프트웨어와 데이터가 컴퓨터 시스템(500)과 외부 시스템 사이에서 이동할 수 있게 한다. 커뮤니케이션 인터페이스(526)의 예는 모뎀, 이더넷 카드(ethernet card), 시리얼 또는 병행의 커뮤니케이션 포트(a serial or parallel communication port)와 같은 네트워크 인터페이스를 포함한다. 커뮤니케이션 인터페이스(526)를 경유하여 전송되는 소프트웨어와 데이터는 전자, 전자기와 광학 또는 커뮤니케이션 인터페이스(526)들에 의해 받아들여질 수 있는 다른 신호들의 형태이다. 다수의 인터페이스들은 역시, 단수의 컴퓨터 시스템(500)에서 제공될 수 있다.

[0047] 상기 컴퓨터(516)으로부터의 인풋과 아웃풋(input and output)은 인풋/아웃풋(I/O) 인터페이스(528)에 의하여 운영된다. 이러한 I/O 인터페이스(528)는 상기 디스플레이(512), 키보드(514), 외부 장치(518)와 다른 그와 같은 컴퓨터 시스템(500)의 요소들을 컨트롤한다.

[0048] 본 발명은 오직 이러한 조건 하에서 편의를 위한 목적으로 이용된다. 본 발명은 다른 컴퓨터 장치들과 제어 시스템(500)들에 적용될 수 있음이 보다 명백해질 수 있을 것이다. 따라서 컴퓨터 장치는 전화, 휴대폰, 텔레비전, 텔레비전 셋업 유닛, 컴퓨터 판매점(point of sale computers), 현금 자동 인출기, 랩탑 컴퓨터, 서버들, 개인적인 전자 어시스턴트와 자동차들을 가지는 온갖 종류의 어플라이언스(appliance)들을 포함한 각종 시스템을 포함하는 것이다. 도 8에서 참조하는 바와 같이, 그런 컴퓨터 장치는 부가적인 구성요소들을 포함하거나 어떤 구성요소들은 삭제할 수 있다.

[0049] 이상에서, 설명의 목적으로, 본 발명의 실시예들에 대한 철저한 이해를 제공하기 위하여 다양한 세부적인 내용이 개시되었다. 하지만, 당업자들은 본 발명을 실시하는데 이러한 구체적인 내용들이 요구되지 않는다는 것을 이해할 수 있을 것이다. 다른 경우에서 본 발명이 불명확하게 되지 않도록 잘 알려진 전기적인 구조들 및 회로

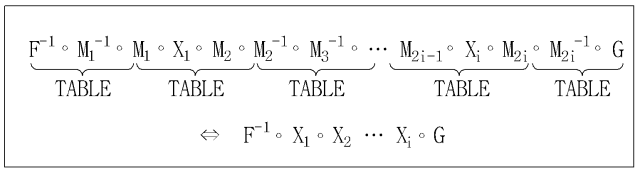
들이 블록도의 형태로 도시되었다. 예를 들어 여기에 기재된 본 발명의 실시예들은 소프트웨어 루틴(software routine), 하드웨어 회로(hardware circuit), 펌웨어(firmware), 또는 이들의 조합으로 구현되었는지에 대한 세부 사항들이 제공되지 않았다.

[0050] 본 발명의 실시예들은 기계 판독 가능 저장 매체(machine readable storage medium)에 저장되는 소프트웨어 제품(software product)으로 구현될 수 있다(컴퓨터 판독가능 매체(computer-readable medium), 프로세서 판독 가능 매체(processor-readable medium), 컴퓨터 판독 가능 프로그램을 가진 컴퓨터 사용가능 매체(computer usable medium)들도 포함된다. 상기 기계 판독 가능 매체(machine-readable medium)는 디스켓, 씨디 롬(compact disk read only memory: CD-ROM), 휘발성(volatile) 또는 비휘발성(non-volatile)의 메모리 장치, 또는 기타 저장 메커니즘을 포함한다. 상기 기계 판독 가능 매체는 자성(magnetic), 광학의(optical), 또는 전자 저장 매체(electrical storage medium)를 포함하는 모든 적합한 유형의 매체일 수 있다. 기계 판독 가능 매체는 다양한 명령들(instructions), 코드 시퀀스(code sequence), 구성 정보(configuration information) 또는 다른 데이터들을 포함할 수 있고, 기계판독 가능 매체가 실행될 때 프로세서로 하여금 발명의 실시예에 따른 방법의 단계들을 수행하도록 한다. 본 발명의 기술 분야에 통상의 지식을 가진 자들은 본 발명을 실시하는 데 필수적인 명령(instruction)들과 연산(operation)들이 기계 판독 가능 매체(machine-readable medium)에 저장될 수 있다는 것을 이해할 수 있을 것이다. 상기 기계 판독 가능 매체는 상술한 태스크들(tasks)을 수행하는 회로와 인터페이스(interface)할 수 있다.

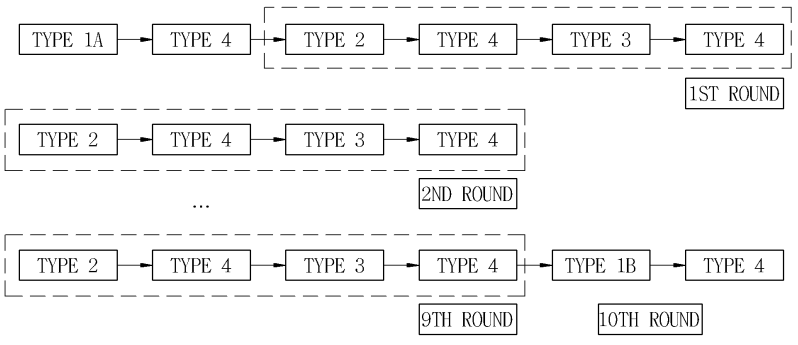
[0051] 본 발명의 상술한 실시예들은 예시를 목적으로만 기재되었고, 다수의 서로 다른 종류의 소프트웨어 또는 소프트웨어 조각들이 본 발명에 따른 강화된 보안의 혜택을 받을 수 있는 것이 자명하다. 또한, 본 발명의 기술분야에서 통상의 지식을 가진 자에 의하여 본 발명의 범위를 벗어나지 않는 한 특정한 실시예에 대한 변경, 수정 및 변형이 가능하고, 본 발명의 범위는 여기에 첨부된 특허청구범위에 의해서만 정해진다.

도면

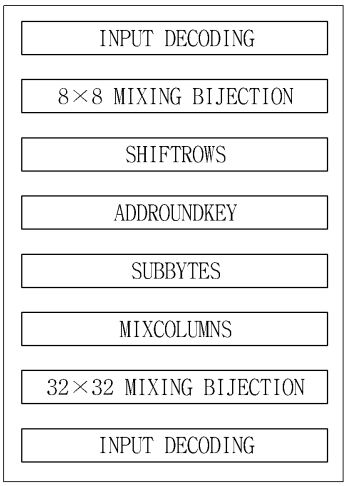
도면1



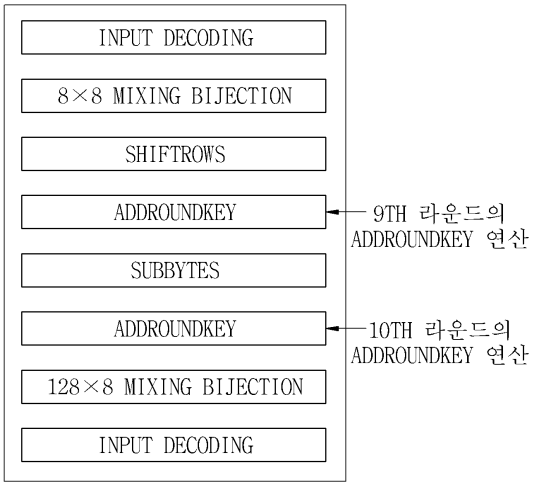
도면2



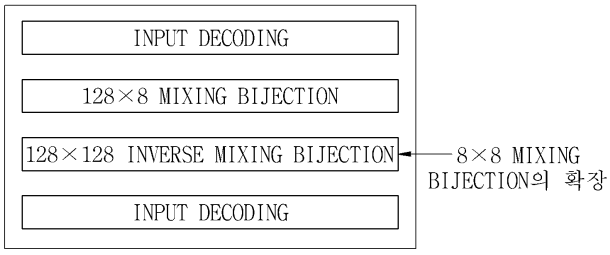
도면3


INPUT DECODING
8×8 MIXING BIJECTION
SHIFTROWS
ADDROUNDKEY
SUBBYTES
MIXCOLUMNS
32×32 MIXING BIJECTION
INPUT DECODING

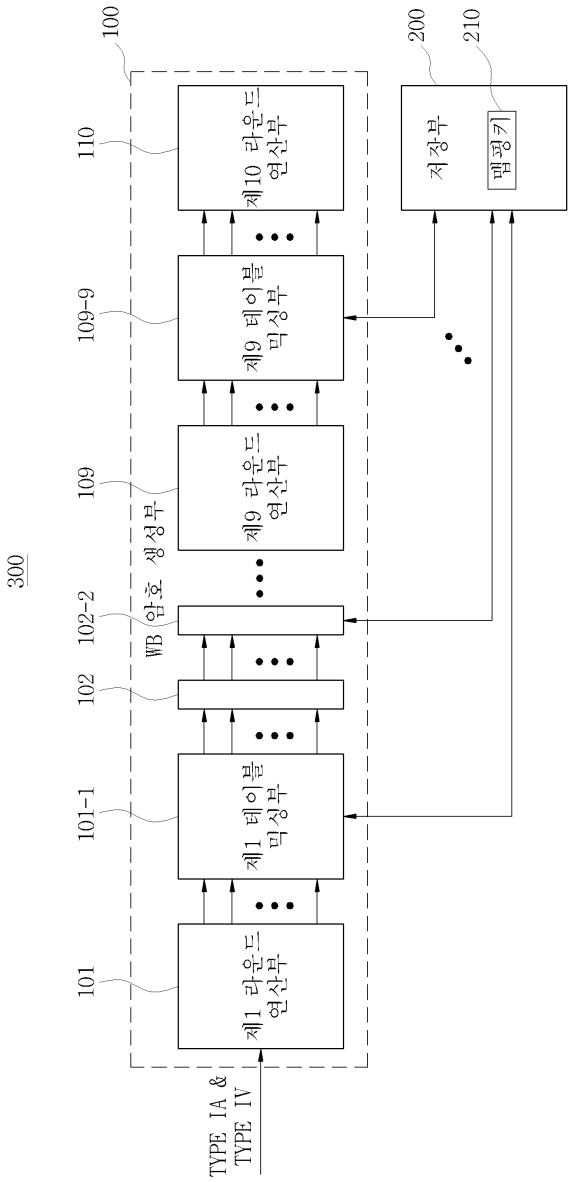
도면4


INPUT DECODING
8×8 MIXING BIJECTION
SHIFTROWS
ADDROUNDKEY
9TH 라운드의 ADDROUNDKEY 연산
SUBBYTES
ADDROUNDKEY
10TH 라운드의 ADDROUNDKEY 연산
128×8 MIXING BIJECTION
INPUT DECODING

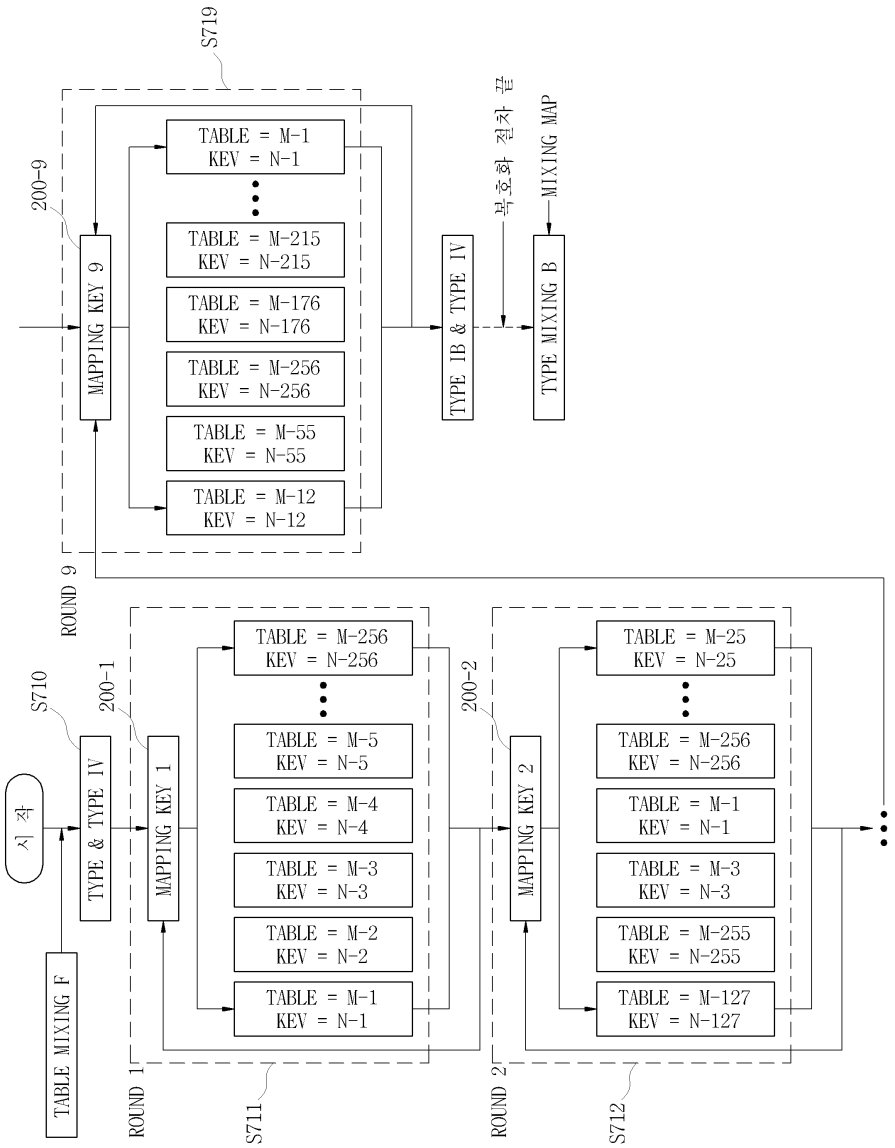
도면5


INPUT DECODING
128×8 MIXING BIJECTION
128×128 INVERSE MIXING BIJECTION
8×8 MIXING BIJECTION의 확장
INPUT DECODING

도면6


300
100
110
제10 라운드 연산부
109-9
제9 테이블 믹싱부
109
제9 라운드 연산부
WB 암호 생성부
102-2
102
101-1
제1 테이블 믹싱부
101
제1 라운드 연산부
TYPE 1A & TYPE IV
200
210
저장부
팸평키

도면7


S719
200-9
MAPPING KEY 9
TABLE = M-1
KEY = N-1
TABLE = M-215
KEY = N-215
TABLE = M-176
KEY = N-176
TABLE = M-256
KEY = N-256
TABLE = M-55
KEY = N-55
TABLE = M-12
KEY = N-12
TYPE IB & TYPE IV
복호화 절차 끝
TYPE MIXING B
MIXING MAP
ROUND 9
S710
200-1
시 작
TYPE & TYPE IV
MAPPING KEY 1
TABLE = M-256
KEY = N-256
TABLE = M-5
KEY = N-5
TABLE = M-4
KEY = N-4
TABLE = M-3
KEY = N-3
TABLE = M-2
KEY = N-2
TABLE = M-1
KEY = N-1
200-2
MAPPING KEY 2
TABLE = M-25
KEY = N-25
TABLE = M-256
KEY = N-256
TABLE = M-1
KEY = N-1
TABLE = M-3
KEY = N-3
TABLE = M-255
KEY = N-255
TABLE = M-127
KEY = N-127
TABLE MIXING F
ROUND 1
S711
ROUND 2
S712

도면8