(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0024965
(43) 공개일자 2016년03월07일

(51) 국제특허분류(Int. Cl.)
H04L 9/06 (2006.01)
(52) CPC특허분류
H04L 9/0618 (2013.01)
H04L 9/0631 (2013.01)
(21) 출원번호 10-2016-7002026
(22) 출원일자(국제) 2014년06월19일
심사청구일자 없음
(85) 번역문제출일자 2016년01월22일
(86) 국제출원번호 PCT/US2014/043169
(87) 국제공개번호 WO 2015/047487
국제공개일자 2015년04월02일
(30) 우선권주장
13/929,589 2013년06월27일 미국(US)

(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
아반지, 로베르토
미국 92121 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인 남앤드남

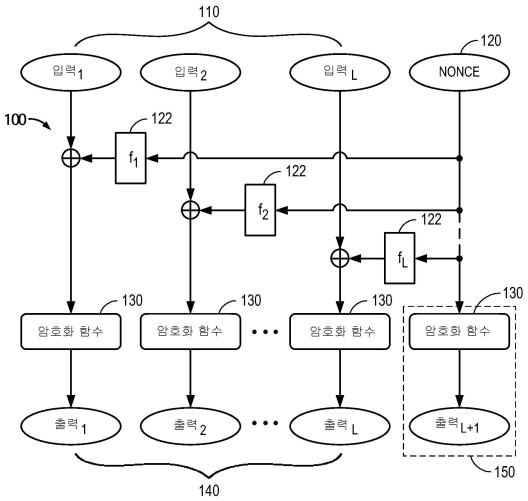
전체 청구항 수 : 총 44 항

(54) 발명의 명칭 **평문 데이터를 암호화하기 위한 방법 및 장치**

(57) 요 약

평문 데이터를 암호화하기 위한 장치 및 방법이 개시된다. 방법은 적어도 하나의 평문 데이터 입력을 수신하는 단계; Nonced 평문 데이터 출력들을 생성하도록 적어도 하나의 평문 데이터 입력에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 적어도 하나의 평문 데이터 입력에 적용된 암호화 함수의 부분의 중간 값에 함수를 통해 Nonce를 적용하는 단계; 및 암호화된 출력 데이터를 생성하도록 Nonced 평문 데이터 출력들 및/또는 중간 Nonced 데이터 출력들 중 적어도 하나에 암호화 함수를 적용하는 단계를 포함한다. 암호화된 출력 데이터는 그 후 메모리에 전송된다.

대 표 도 - 도1a



(52) CPC특허분류
H04L 2209/08 (2013.01)

명 세 서

청구범위

청구항 1

평문 데이터를 암호화하기 위한 방법으로서,

적어도 하나의 평문 데이터 입력을 수신하는 단계;

Nonced 평문 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 적용된 암호화 함수의 부분의 중간 값들에 함수를 통해 Noncee를 적용하는 단계;

암호화된 출력 데이터를 생성하도록 상기 Nonced 평문 데이터 출력들 및/또는 상기 중간 Nonced 데이터 출력들 중 적어도 하나에 상기 암호화 함수를 적용하는 단계; 및

상기 암호화된 출력 데이터를 메모리에 전송하는 단계

를 포함하는,

평문 데이터를 암호화하기 위한 방법.

청구항 2

제 1 항에 있어서,

상기 Nonce는 암호화된 방식으로 저장되는,

평문 데이터를 암호화하기 위한 방법.

청구항 3

제 1 항에 있어서,

상기 Nonce는 암호화되지 않은 방식으로 저장되는,

평문 데이터를 암호화하기 위한 방법.

청구항 4

제 1 항에 있어서,

상기 데이터 출력들에 적용되는 암호화 함수는 동일한 암호화 함수인,

평문 데이터를 암호화하기 위한 방법.

청구항 5

제 1 항에 있어서,

상기 함수는 XOR 함수를 포함하는,

평문 데이터를 암호화하기 위한 방법.

청구항 6

제 1 항에 있어서,

상기 함수는 예측할 수 없는 방식으로 상기 Nonced 평문 데이터 출력들을 교란(perturbate)하도록 상기 Nonce로부터 값들을 도출하는 수학 함수인,

평문 데이터를 암호화하기 위한 방법.

청구항 7

제 6 항에 있어서,

상기 함수는 이진 또는 산술 가산 상수들, 상기 함수에 대한 입력을 나타내는 비트들의 임의의 치환들 또는 순환 회전들을 포함하는,

평문 데이터를 암호화하기 위한 방법.

청구항 8

제 6 항에 있어서,

상기 함수는 상기 암호화 함수에 관련되는,

평문 데이터를 암호화하기 위한 방법.

청구항 9

제 1 항에 있어서,

상기 암호화 함수를 적용하는 단계는 상기 Nonce이 적용되기 이전에, 상기 암호화 함수를 모델링하는 라운드 함수들의 제 1 시퀀스에 의해 상기 평문 데이터 입력들을 암호화하고, 그 후, 상기 Nonce가 적용되어 Nonced 데이터 출력들을 생성하는 단계를 포함하는,

평문 데이터를 암호화하기 위한 방법.

청구항 10

제 9 항에 있어서,

상기 Nonced 데이터 출력들은 상기 암호화된 출력 데이터를 생성하도록 상기 암호화 함수를 모델링하는 라운드 함수들의 제 2 시퀀스에 의해 암호화되는,

평문 데이터를 암호화하기 위한 방법.

청구항 11

제 1 항에 있어서,

메모리로부터 상기 암호화된 출력 데이터를 복호화하는 단계

를 더 포함하는,

평문 데이터를 암호화하기 위한 방법.

청구항 12

코드를 포함하는 비-일시적인 컴퓨터-판독 가능 매체로서,

상기 코드는 프로세서에 의해 실행될 때, 상기 프로세서로 하여금,

적어도 하나의 평문 데이터 입력을 수신하게 하고;

Nonced 평문 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 적용된 암호화 함수의 부분의 중간 값들에 함수를 통해 Noncee를 적용하게 하고;

암호화된 출력 데이터를 생성하도록 상기 Nonced 평문 데이터 출력들 및/또는 상기 중간 Nonced 데이터 출력들 중 적어도 하나에 상기 암호화 함수를 적용하게 하고; 그리고

상기 암호화된 출력 데이터를 메모리에 전송하게 하는,

컴퓨터-판독 가능 매체.

청구항 13

제 12 항에 있어서,

상기 Nonce는 암호화된 방식으로 저장되는,

컴퓨터-판독 가능 매체.

청구항 14

제 12 항에 있어서,

상기 Nonce는 암호화되지 않은 방식으로 저장되는,

컴퓨터-판독 가능 매체.

청구항 15

제 12 항에 있어서,

상기 데이터 출력들에 적용되는 암호화 함수는 동일한 암호화 함수인,

컴퓨터-판독 가능 매체.

청구항 16

제 12 항에 있어서,

상기 함수는 XOR 함수를 포함하는,

컴퓨터-판독 가능 매체.

청구항 17

제 12 항에 있어서,

상기 함수는 예측할 수 없는 방식으로 상기 Nonced 평문 데이터 출력들을 교란하도록 상기 Nonce로부터 값들을 도출하는 수학 함수인,

컴퓨터-판독 가능 매체.

청구항 18

제 17 항에 있어서,

상기 함수는 이진 또는 산술 가산 상수들, 상기 함수에 대한 입력을 나타내는 비트들의 임의의 치환들 또는 순환 회전들을 포함하는,

컴퓨터-판독 가능 매체.

청구항 19

제 17 항에 있어서,

상기 함수는 상기 암호화 함수에 관련되는,

컴퓨터-판독 가능 매체.

청구항 20

제 12 항에 있어서,

상기 Nonce이 적용되기 이전에, 상기 암호화 함수를 모델링하는 라운드 함수들의 제 1 시퀀스에 의해 상기 평문 데이터 입력들을 암호화하고, 그 후, 상기 Nonce가 적용되어 Nonced 데이터 출력들을 생성함으로써 상기 암호화 함수를 적용하기 위한 코드

를 더 포함하는,

컴퓨터-판독 가능 매체.

청구항 21

제 20 항에 있어서,

상기 암호화된 출력 데이터를 생성하도록 상기 암호화 함수를 모델링하는 라운드 함수들의 제 2 시퀀스에 의해 상기 Nonced 데이터 출력들 암호화하기 위한 코드

를 더 포함하는,

컴퓨터-판독 가능 매체.

청구항 22

제 12 항에 있어서,

상기 메모리로부터 상기 암호화된 출력 데이터를 복호화하기 위한 코드

를 더 포함하는,

컴퓨터-판독 가능 매체.

청구항 23

평문 데이터를 암호화하기 위한 디바이스로서,

프로세서를 포함하고,

상기 프로세서는,

적어도 하나의 평문 데이터 입력을 수신하고;

Nonced 평문 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 적용된 암호화 함수의 부분의 중간 값들에 함수를 통해 Noncee를 적용하고;

암호화된 출력 데이터를 생성하도록 상기 Nonced 평문 데이터 출력들 및/또는 상기 중간 Nonced 데이터 출력들 중 적어도 하나에 상기 암호화 함수를 적용하고; 그리고

상기 암호화된 출력 데이터를 메모리에 전송하기 위한 것인,

디바이스.

청구항 24

제 23 항에 있어서,

상기 Nonce는 암호화된 방식으로 저장되는,

디바이스.

청구항 25

제 23 항에 있어서,

상기 Nonce는 암호화되지 않은 방식으로 저장되는,

디바이스.

청구항 26

제 23 항에 있어서,

상기 데이터 출력들에 적용되는 암호화 함수는 동일한 암호화 함수인,

디바이스.

청구항 27

제 23 항에 있어서,

상기 함수는 XOR 함수를 포함하는,

디바이스.

청구항 28

제 23 항에 있어서,

상기 함수는 예측할 수 없는 방식으로 상기 Nonced 평문 데이터 출력들을 교란(perturbate)하도록 상기 Nonce로부터 값들을 도출하는 수학 함수인,

디바이스.

청구항 29

제 28 항에 있어서,

상기 함수는 이진 또는 산술 가산 상수들, 상기 함수에 대한 입력을 나타내는 비트들의 임의의 치환들 또는 순환 회전들을 포함하는,

디바이스.

청구항 30

제 28 항에 있어서,

상기 함수는 상기 암호화 함수에 관련되는,

디바이스.

청구항 31

제 23 항에 있어서,

상기 암호화 함수를 적용하는 것은 상기 Nonce이 적용되기 이전에, 상기 암호화 함수를 모델링하는 라운드 함수들의 제 1 시퀀스에 의해 상기 평문 데이터 입력들을 암호화하고, 그 후, 상기 Nonce가 적용되어 Nonced 데이터 출력을 생성하는 것을 포함하는,

디바이스.

청구항 32

제 31 항에 있어서,

상기 Nonced 데이터 출력들은 상기 암호화된 출력 데이터를 생성하도록 상기 암호화 함수를 모델링하는 라운드 함수들의 제 2 시퀀스에 의해 암호화되는,

디바이스.

청구항 33

제 23 항에 있어서,

상기 프로세서는 추가로, 메모리로부터 상기 암호화된 출력 데이터를 복호화하는,

디바이스.

청구항 34

평문 데이터를 암호화하기 위한 디바이스로서,

적어도 하나의 평문 데이터 입력을 수신하기 위한 수단;

Nonced 평문 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 적용된 암호화 함수의 부분의 중간 값들에 함수를 통해 Noncee를 적용하기 위한 수단;

암호화된 출력 데이터를 생성하도록 상기 Nonced 평문 데이터 출력들 및/또는 상기 중간 Nonced 데이터 출력들 중 적어도 하나에 상기 암호화 함수를 적용하기 위한 수단; 및

상기 암호화된 출력 데이터를 메모리에 전송하기 위한 수단

을 포함하는,

디바이스.

청구항 35

제 34 항에 있어서,

상기 Nonce는 암호화된 방식으로 저장되는,

디바이스.

청구항 36

제 34 항에 있어서,

상기 Nonce는 암호화되지 않은 방식으로 저장되는,

디바이스.

청구항 37

제 34 항에 있어서,

상기 데이터 출력들에 적용되는 암호화 함수는 동일한 암호화 함수인,

디바이스.

청구항 38

제 34 항에 있어서,

상기 함수는 XOR 함수를 포함하는,

디바이스.

청구항 39

제 34 항에 있어서,

상기 함수는 예측할 수 없는 방식으로 상기 Nonced 평문 데이터 출력들을 교란(perturbate)하도록 상기 Nonce로부터 값들을 도출하는 수학 함수인,

디바이스.

청구항 40

제 39 항에 있어서,

상기 함수는 이진 또는 산술 가산 상수들, 상기 함수에 대한 입력을 나타내는 비트들의 임의의 치환들 또는 순환 회전들을 포함하는,

디바이스.

청구항 41

제 39 항에 있어서,

상기 함수는 상기 암호화 함수에 관련되는,

디바이스.

청구항 42

제 34 항에 있어서,

상기 암호화 함수를 적용하는 것은 상기 Nonce이 적용되기 이전에, 상기 암호화 함수를 모델링하는 라운드 함수들의 제 1 시퀀스에 의해 상기 평문 데이터 입력들을 암호화하고, 그 후, 상기 Nonce가 적용되어 Nonced 데이터 출력을 생성하는 것을 포함하는,

디바이스.

청구항 43

제 42 항에 있어서,

상기 Nonced 데이터 출력들은 상기 암호화된 출력 데이터를 생성하도록 상기 암호화 함수를 모델링하는 라운드 함수들의 제 2 시퀀스에 의해 암호화되는,

디바이스.

청구항 44

제 34 항에 있어서,

메모리로부터 상기 암호화된 출력 데이터를 복호화하기 위한 수단

을 더 포함하는,

디바이스.

발명의 설명

기 술 분 야

[0001] [0001] 본 발명은 평문 데이터를 암호화하고 대응하는 암호문 데이터를 복호화하기 위한 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] [0002] 메모리 분석기들의 이용은 분배 콘텐츠의 무결성 및 기밀성에 대해 큰 위험을 나타낸다. 코드에 포함된 데이터를 보호하기 위해 상당한 주의가 할애하더라도, 메모리 버스의 콘텐츠들은 버스 스니핑(bus sniffing)에 의해 포착될 수 있다. 예를 들어, 미가공 콘텐츠가 렌더링에 대해 안전한 환경에서 복호화된 이후, 그것이 암호화 형태로 분배되더라도, 이는 미가공 콘텐츠를 누출시키도록 사용할 수 있다. 이는 메모리에 대한 기록들에 대응하는 전기 신호들을 "판독"함으로써 달성될 수 있다. 다른 보다 더 정교한 공격들은 심지어 이들 신호들을 리플레이(replay)하여 프로세서가 공격자에 의해 선택된 데이터를 판독하고 프로세싱하게 속일 수 있다.

[0003] [0003] 콘텐츠 제공자들은 종종 미가공 콘텐츠의 취급에 대한 특정한 요건들을 갖는다. 최소한, 콘텐츠는 보통 문으로 메모리에 저장돼선 안될 수 있다. 대부분의 경우, 물리적 공격들을 방지하기 위해 메모리 스크램블링 또는 암호화의 일부 형태가 모든 메모리 레코딩에 적용된다는 최소한의 요건이 있다. 예로서, 특정 어드레스에 기록된 데이터는 일반적으로 명확한(clear) 데이터, 어드레스 및 마스터 키의 함수이다. 이것은 동일한 데이터가 다른 어드레스에 기록될 때, 다른 인코딩을 갖는다는 것을 보장한다. 평문 데이터의 암호화를 랜덤화하기 위한 Nonce의 이용은 이들 Nonce들이 안전한 방식으로 저장되고 리트리브될 때, 리플레이 공격(replay attack)들을 방지하기 위해 사용될 수 있다.

[0004] [0004] 또한, 보안 통신에 대한 쓰루풋 요건들은 현재 스트림 및 블록 암호가 테스트 받게 하고, 전력 및 영역

요건들을 제어하는 동시에 쓰루풋을 증가하기 위한 신규한 구조들이 바람직하다.

[0005] [0005] 불행히도, 현재의 기술은 종종 비효율적이고 전력면에서 상당한 증가들 없이, 그리고 동일한 보안 레벨로 더 강한 레벨의 보호, 더 높은 쓰루풋, 하드웨어 구현들의 경우, 영역 요건들이 요구될 수 있다.

[0006]

발명의 내용

[0007] [0006] 본 발명의 양상들은 평문 데이터를 암호화하기 위한 장치 및 방법에 관련될 수 있다. 방법은 적어도 하나의 평문 데이터 입력을 수신하는 단계; 평문 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 상기 적어도 하나의 평문 데이터 입력에 적용된 암호화 함수의 부분의 중간 값에 함수를 통해 Nonce를 적용하는 단계; 및 암호화된 출력 데이터를 생성하도록 Nonced 평문 데이터 출력들 및/또는 중간 Nonced 데이터 출력들 중 적어도 하나에 암호화 함수를 적용하는 단계를 포함한다. 암호화된 출력 데이터는 그 후 메모리에 전송된다.

도면의 간단한 설명

[0008] [0007] 도 1a는 평문 데이터 입력들의 일련의 블록들이 암호화 함수 및 Nonce을 사용하여 암호화되는 프로세스를 예시하는 흐름도이다.

[0008] 도 1b는 도 1a의 역방향 복호화 프로세스를 예시하는 흐름도이다.

[0009] 도 2는 라운드들로서 지칭되는 유사한 계산 블록들의 반복들에 기초한 블록 암호의 공통 구조를 예시하는 흐름도이다.

[0010] 도 3은 Nonce로 암호화 프로세스의 중간 단계를 보강하고 라운드 함수의 제 1 및 제 2 세트를 사용하는 확장된 형태로 데이터 입력을 암호화하기 위한 프로세스를 예시하는 흐름도이다.

[0011] 도 4a는 개별 블록들의 암호화 프로세스를 상이한 방식으로 수정하기 위해 동일한 키 또는 Nonce 또는 Nonce로부터 도출된 값들을 사용하여 평문 데이터 입력들의 일련의 블록들을 암호화하기 위한 프로세스를 예시하는 흐름도이다.

[0012] 도 4b는 도 4a의 역방향 복호화 프로세스를 예시하는 흐름도이다.

[0013] 도 5는 공통 Nonce로부터 도출된 상이한 값들을 암호화 프로세스의 중간 단계에 적용함으로써 여러 상이한 출력들을 획득하는, 데이터 입력을 암호화하기 위한 프로세스를 예시하는 흐름도이다.

[0014] 도 6은, 그것을 복호화 및 재-암호화해야할 필요없이 대용량 저장소에 대한 암호화된 메모리의 저장 및 복원을 가능케 하기 위해 데이터 암호화 기술들을 구현하기 위한 예시적인 컴퓨터 하드웨어 시스템의 도면이다.

발명을 실시하기 위한 구체적인 내용

[0009] [0015] "예시적인" 또는 "예"라는 단어는 "예시, 실례 또는 예증"의 역할을 의미하는 것으로 사용된다. "예시적인" 또는 "예"로서 여기서 설명되는 임의의 양상 또는 실시예는 반드시 다른 양상들 또는 실시예들보다 선호되거나 유리한 것으로 해석되는 것은 아니다.

[0010] [0016] 본 발명의 실시예들은 메모리에 저장된 데이터의 보호를 위한 강화된 메커니즘을 제공하는 기술들에 관한 것이다. 특히, 메모리 암호화를 강화하기 위해 블록 암호(block cipher)의 기능성을 확장하는 방법들 및 프로세스들이 설명된다. 부가적으로, 이러한 기술들은 또한 아래에서 설명될 바와 같이, 성능, 쓰루풋 및 전력 소비를 개선할 수 있다. 이들 기술들은 또한 (유선 또는 무선) 네트워크를 통한 전송의 안전한 데이터 저장을 위해 성능, 쓰루풋 및 전력 소비를 개선하는데 사용될 수 있다.

[0011] [0017] 일 실시예에서, 평문 데이터 입력들의 일련의 L 블록들이 암호화 함수(예를 들어, 블록 암호)를 사용하여 암호화되는 암호화 방식이 활용된다. 블록 암호를 통한 암호화에 앞서, Nonce은 함수를 통해 평문 데이터 입력들에 적용된다. 특히, 복수의 평문 데이터 입력들을 수신하는 것; Nonced 평문 데이터 출력들을 생성하도록 복수의 평문 데이터 입력들에 함수를 통해 Nonce을 적용하는 것; 암호화된 출력 데이터를 생성하도록 Nonced 평문 데이터 출력들에 블록 암호와 같은 암호화 함수를 적용하는 것; 및 암호화된 출력 데이터를 메모리에 전송하는 것을 포함하는 평문 데이터를 암호화하기 위한 방법 또는 프로세스가 개시된다.

[0012] [0018] 일 실시예에서, 아래에서 보다 구체적으로 설명될 바와 같이, 방법은 복수의 평문 데이터 입력들을 수신하는 것; Nonced 평문 데이터 출력들을 생성하도록 복수의 평문 데이터 입력들에 그리고/또는 중간 Nonced 데이터 출력들을 생성하도록 복수의 평문 데이터 입력들에 적용된 암호화 함수의 부분의 중간 값에 함수를 통해 Nonce를 적용하는 것; 및 암호화된 출력 데이터를 생성하도록 Nonced 평문 데이터 출력들 및/또는 중간 Nonced 데이터 출력들 중 적어도 하나에 암호화 함수를 적용하는 것을 포함한다. 암호화된 출력 데이터는 그 후 메모리에 전송된다.

[0013] *평문 데이터 입력들의 L 블록들의 암호화의 랜덤화*

[0014] [0019] 도 1a에서 알 수 있는 바와 같이, 일 실시예에서, 복수의 평문 데이터 입력들(Input1-InputL)(110)이 수신되는 방법 또는 프로세스(100)가 수행된다. Nonce(120)은 평문 데이터 입력들(Input1-Input)L(110)에 함수(122)를 통해 적용된다. Nonce(120)은 평문 데이터의 L 블록들(Input1-InputL)(110)의 암호화를 랜덤화하는데 사용할 수 있다. 도 1a에서 알 수 있는 바와 같이, 평문 데이터의 L 블록들(Input1-InputL)(110)이 수신되고 Nonce(120)는 Nonced 평문 데이터 출력들을 생성하도록 함수들(f1, f2,..,fL)(122)에 의해 적용될 수 있다. 일 실시예에서, 아래에서 보다 상세히 설명될 바와 같이, Nonce(120)을 적용할 함수는 XOR 함수를 포함할 수 있다. 다른 실시예에서, XOR 함수 대신, 모듈식 부가 함수(modular addition function)가 사용될 수 있다. 암호화 함수(130)(예를 들어, 블록 암호)는 그 후 암호화된 출력 데이터(Output1-OutputL)(140)가 메모리에 출력되도록 Nonced 평문 데이터 출력들에 적용될 수 있다.

[0015] [0020] Nonce(120)은 (동시에 프로세싱되는 L개의 블록들(110) 중에서) 동일한 평문 블록들이 동일한 암호화를 갖는 것을 방지하기 위해 일부 변형이 될 수 있다는 것이 이해되어야 한다. 또한, L개의 병렬 암호화들에서 사용되는 암호화 함수(130)에 사용된 암호 키가 동일할 수 있기 때문에, 키 스케줄은 L번 재실행될 필요가 없다.

[0016] [0021] 또한, Nonce(120)은 아래에서 보다 상세히 설명될 바와 같이, 이용 경우에 의존하여 명확한 또는 암호화된 방식으로, 메모리의 더 작은 내부 보호된 영역에 저장되거나 또는 메인 메모리에 저장될 수 있다.

[0017] [0022] 또한, 단순화를 위해, 암호화 함수(130)에서 사용되는 특정 암호화 키는 표시되지 않는다. 그러나 암호화 함수는 암호화 프로세스의 암호화 함수에 의해 사용되는 키인 부가적인 입력을 취한다는 것이 이해되어야 한다. 또한, 암호화 함수들이 동일한 암호화 키를 사용하여 블록 암호를 반복하는 경우, 다양한 버티컬 파이프 라인들이 동일한 키 스케줄을 공유할 수 있다는 것이 또한 이해되어야 하며, 여기서 암호화 함수(130)에서 사용되기 전에 일부 고정된 비트 치환들(예컨대, 회전들)이 라운드에 적용될 수 있다. 하드웨어 구현에서, 이러한 치환들은 성능에 영향을 주어선 안 되는데, 그 이유는 이들은 단지 실리콘의 상이한 와이어링에 해당하기 때문이다.

[0018] [0023] 함수들(f1, f2,.., fL)(122)은 공격자에 대해 예측 불가능한 방식으로 Nonced 평문 데이터 출력들의 계산을 교란(perturbate)하기 위해 Nonce(120)로부터 값을 도출하는 수학 함수일 수 있다. 이들은 상수들을 통한 마스킹들, 다른 순환 회전들, 또는 암호화 함수(130)와 관련될 수 있는 다른 함수들일 수 있다. Nonce(120)이 암호 블록길이보다 큰 치수를 갖는 경우, 함수들은 단지 Nonce의 세그먼트들의 추출들일 수 있다.

[0019] [0024] 또한, 방법론(100)은 동일한 암호화 함수(130)의 L 또는 L + 1의 구현(또는 다른 암호화 함수가 활용될 수 있음)을 활용하여 병렬화(parallelizable) 가능할 수 있다. 도 1a에서 알 수 있는 바와 같이, 점선(150)들에서, 암호문의 확장은 하나의 부가적인 블록을 출력하는 L + 1번째 구현으로서 도시된다. 또한, 암호화 함수(130)에 대한 동일한 암호화 키가 각각의 블록에 대해 이용될 수 있기 때문에, 서브-키 도출이 한 번만 수행될 필요가 있고, 따라서 하드웨어 자원들을 절감한다.

[0020] [0025] 일부 실시예들에서, 충분한 보안성이 액세스 가능한 메모리 영역에서 보통문으로 Nonce(120)를 저장함으로써 제공될 수 있는데, 그 이유는 그것이 초기화 벡터의 역할과 유사한 역할을 하기 때문이다. 이 접근법의 이득은 Nonce(120)가 블록 크기보다 더 짧아질 수 있고, 이에 따라 그것이 입력 블록(110)의 선택된 비트 필드에만 함수 연산들(122)에서 적용될 수 있다는 것이다. 이 방식은 메모리 암호화에 유용할 수 있다. 예로서, L = 8로 세팅함으로써, 블록 암호가 128 비트들의 블록 크기를 갖고 캐시 라인들이 128 바이트 길이인 경우, 전체 캐시 라인들은 이들이 캐시의 마지막 레벨로부터 유출될 때, 한번에 암호화될 수 있다.

[0021] [0026] 따라서, 전술한 바와 같이, 프로세스(100)는 메모리 암호화를 강화하기 위해 블록 암호들의 기능성을 확장한다. 특히, 암호화 방식(100)은, 암호화 함수(130)를 사용하여 각각 암호화되는 평문 데이터 입력들(Input1-InputL)(110)의 일련의 L 블록들을 활용하며, 이는 암호화 함수(130)를 통한 암호화 이전에, Nonce(120)가 평문 데이터 입력(100)에 함수(122)를 통해 적용된다. 암호화 함수(130)는 암호화된 출력 데이터(Output1-

OutputL)(140)가 메모리에 출력되도록 Nonced 평문 데이터 출력들에 적용될 수 있다.

[0022] 복호화는 역방향으로 작동한다. 예를 들어, 도 1b를 참조하면, 암호화 함수(130)의 역 함수가 (입력(140)으로서 도시된) 메모리로부터 암호화된 출력 데이터에 적용될 수 있는데, 즉, 대응하는 복호화 프리미티브(decryption primitive)가, (출력들(110)로서 도시된) 원래 입력들이 복구 가능한 $Input_i$와 Nonce의 합성, 예를 들어 $Input_i \oplus Nonce$ (여기서 i= 1, 2 …, L 및 Nonce)을 계산하는데 사용될 수 있.

[0023] *블록 암호의 랜덤화*

[0024] [0027] 아래에서 설명될 바와 같이, 평문 데이터 입력들(110)은 먼저, Nonce(120)이 적용되기 이전에, (선택된 암호화 함수인) 블록 암호를 구성하는 라운드 함수들의 제 1 시퀀스에 의해 암호화되고, 그 후 Nonce가 적용되어 Nonced 데이터 출력들을 생성할 수 있다. Nonced 데이터 출력들은 그 후, 메모리에 출력되는 암호화된 출력 데이터를 생성하기 위해 (선택된 암호화 함수인) 블록 암호를 모델링하는 라운드 함수의 제 2 시퀀스에 의해 암호화될 수 있다.

[0025] [0028] 암호화 함수(130)(예를 들어, 블록 암호)를 모델링하기 위해, 다양한 구성들이 사용될 수 있다. 예를 들면, Luby-Rackoff 구성들과 같은 구성들은, 예를 들어, 파이스텔 네트워크(예컨대, DES(Data Encryption Standard)) 및 SP(Substitution-Permutation) 네트워크(예컨대, AES(Advanced Encryption Standard)이 사용될 수 있다. 둘 다의 경우들에서, 하나의 파라미터화된 비-선형 함수는 반복적으로 입력에 적용될 수 있다. 이 함수의 각각 적용은 "라운드" 또는 "라운드 함수"로 지칭될 수 있다. 라운드의 출력은 다음 라운드의 입력이다. 평문이 제 1 라운드에 대한 입력이고, 암호문은 마지막 라운드의 출력이다. 라운드 함수는 라운드 키로 불리는 추가의 파라미터를 취하고 라운드 키들은 암호화/복호화 키(예를 들어, 암호키)로부터 도출된다.

[0026] [0029] 도 2를 참조하면, 라운드 함수에 기초하여 블록 암호를 생성하기 위한 프로세스(200)의 예가 도시된다. 도 2에서 도시된 바와 같이, 평문 데이터 입력(202)은 블록 암호를 모델링하도록 라운드 함수의 복수의 N개의 라운드들(204)에 입력된다. 따라서, 블록 암호는 라운드 함수들의 복수의 N개의 라운드들(204)에 의해 모델링되며, 여기서 k1, k2, .., kN은 각각 라운드들(1, 2, .. ,N)에 대한 라운드 키이다. 출력(206)은, 평문 데이터 입력(202)에 적용된 (블록 암호를 모델링하는) 라운드 함수에 의해 암호화되는 암호화된 평문 데이터 입력(202)이다. 복호화는 반대의 정확히 동일한 프로세스일 것이란 점이 이해되어야 한다.

[0027] [0030] 예시적인 구현들이 아래에서 설명될 것이다. 예를 들어, 이 방식의 성능 효율적 구현은 가능하게는 라운드 키들을 공유하는 동일한 블록 암호의 2개의 병렬 구현을 요구할 수 있다. 하드웨어 구현 비용들을 감소시키기 위해, Nonce은 암호화의 중간에 적용될 수 있다. 이것에 의해, Nonce의 적용 이전에 암호의 부분은 단지 한 번만 구현되어야 하고 Nonce의 적용 이후 암호의 부분이 두 번 구현된다.

[0028] [0031] 예로서, 프로세스(300)를 예시하는 도 3을 참조하여, 평문 데이터 입력(302)은 N개의 라운드들의 M, 예를 들어, M 라운드 키들(k1, k2, .., kM)(1≤ M < N)을 사용하여 파라미터화된 M 라운드들(304)을 통해 암호화될 수 있다, 다음으로, Nonce(v)(306)가 적용되고, - 예를 들어, M 번째 라운드의 출력 X에 대해 Nonce를 XOR 연산하고 및 XOR 연산된 출력 및 Nonce가 추가로 독립적으로 암호화됨(별개의 블록(308) - (M + 1)번째 라운드로 프로세스를 재개한다. 도 3에서 알 수 있는 바와 같이, N-M 라운드들(310)에 대한 라운드 키들(k' 및 k")의 다음 라운드는 라운드 키들의 동일한 세트일 수 있거나, 또는 상이한 비밀 상수를 통해 마스킹되거나 상이한 회전들과 같이 서로 약간 변형될 수 있다. 부가적으로, 출력들은 연결되어(블록 314) 출력을 발생시킨다(316).

[0029] [0032] 약간 상이한 구현의 다른 예가 Nonce의 그리고 X의 비트들의 치환들로 구성될 수 있다. 예를 들어, X $= X_{hi} \| X_{lo}$ (동일한 길이의 2 비트 스트링들의 연결로서의 분해)로 세팅되고 v(Nonce)가 $v = v_{hi} \| v_{lo}$로 세팅되는 경우, A는 $A = X_{hi} \| vlo$가 될 것이고 B는 $B = X_{lo} \| v_{hi}$가 될 것이다. 따라서, 마지막 N - M 라운드들이 충분한 확산(sufficient diffusion)을 갖는 경우, 출력의 둘 다의 절반들(C 및 D) 상에 X 및 Nonce 둘 다의 충분한 영향이 있다. 이것은 단지 일 예이며, 다른 비트 치환들이 가능하는 것이 이해되어야 한다. 그러나 블록 크기가 충분히 큰 경우, 이 방식은 동일한 평문에 대해 동일한 암호

문들의 빈번한(부분적) 반복을 야기하지 않을 수 있다. 따라서 $A = (X_{hi} \oplus v_{hi}) \| v_{lo}$ 및 $B = (X_{lo} \oplus v_{lo}) \| v_{hi}$ 과 같은 수식과 같이 Nonce가 다음 입력의 전체에 영향을 미치는 것이 바람직할 수 있다. 여기서 유의점은 프로세스가 쉽게 반전 가능하다는 것이고 - 이에 따라 복호화 프로세스가 N - M 라운드들을 수행하면, Nonce 복원 가능하게 될 수 있다는 것이 이해되어야 한다. 또한, 연결 함수(314)는 마지막 병렬 라운드들의 두 개의 출력들의 연결일 수 있고 - 그러나 두 입력들의 임의의 다른 비트 치환이 여기서 사용될 수 있다. 프로세스는, 마지막 N - M은 라운드만 제외하고 처음 M 라운드들의 하드웨어 구현들이 복제되어야 할 필요가 없다는 점에서 유리하다. 복호화는, 이 경우에 또한 역방향으로 작동한다. Nonce(v)가 복원될 때까지, 출력의 두 개의 "측들"(C 및 D)이 마지막 N - M 라운드들에 대해 병렬로 복호화되고, 별개의 동작이 반전되고, 그 후 입력의 복호화가 M 라운드들에서 완료된다.

[0030] [0033] 도 4a을 참조하면, 다양한 라운드들의 이용에 따라 적합하게 변형된 이후 Nonce(420)가 각각의 블록에 부가되는, 평문 데이터 입력들의 L개의 블록들(Input1-InputL)(410)을 동시에 암호화하기 위한 이전 기술들을 일반화하는 프로세스(400)의 예가 예시된다. 특히, 도 4a의 프로세스(400)는 평문 데이터 입력들(410)이 Nonce(120)가 적용되기 이전에, 라운드 함수들(M 라운드들(404))의 제 1 시퀀스에 의해 암호화되고, 그 후, Nonce(420)가 적용되어 Nonced 데이터 출력들을 생성할 수 있다는 것을 예시한다. 도 4a에서 알 수 있는 바와 같이, 평문 데이터의 L 블록들(Input1-InputL)(410)은 수신되고, 상이한 Nonced 데이터 출력들을 생성하기 위해, 함수들(f1, f2,., fL)(422)이 Nonce(420)에 적용된다. 일 실시예에서, Nonce(420)를 적용할 함수는 XOR 함수를 포함할 수 있다. 대안적으로, 모듈식 부가들 또는 차감들과 같은 다른 쉽게 반전 가능한 함수들이 Nonce(이로부터 도출된 값들)을 적용하는데 사용할 수 있다. Nonced 데이터 출력들은 그 후 메모리에 출력되는 암호화된 출력 데이터(440)를 생성하기 위해 라운드 함수의 제 2 시퀀스(N-M 라운드들(406))에 의해 암호화될 수 있다. M 및 N-M 라운드들(404 및 406)을 이용함으로써 전체 암호화 함수가 그에 의해 모델링 및 적용된다는 것이 이해되어야 한다. 또한 메모리에 출력되는 암호화된 출력 데이터를 생성하기 위해 라운드 함수(404 및 406)의 L(nonce가 암호화되지 않은 경우) 또는 L+1(nonce가 암호화된 경우) 구현을 이용하여 방법론(400)은 병렬화될 수 있다.

[0031] [0034] 도 1a를 참조하여 설명한 바와 같이, 함수(f1, f2,.., fL)(422)는 실질적으로 동일한 역할을 수행한다는 것이 이해되어야 한다. 그러나 기본 블록 암호의 (M + 1)번째 라운드(406)까지 함수들이 구현되지 않는다는 사실은 Nonce로부터 보다 복잡한 도출을 허용한다. AES 구현의 경우, AES 키 스케줄링 절차의 일부 변경이 함수들을 생성하기 위해 이용될 수 있다. 일 실시예에서, 함수들은 블록 암호의 처음 M 라운드들(404)과 병렬로 계산될 수 있다. 다양한 라운드들에 모두 동일한 라운드 키들을 공급하는 것뿐만 아니라, 그들에게 각각의 버티컬 파이프라인에 대해 고유한, 일부 고정된 치환들 및/또는 마스킹들을 적용하는 것이 유리할 수 있다. 또한, Nonce(420)가 초기화 벡터의 역할과 유사한 역할을 수행할 수 있고, 여전히 충분히 안전할 수 있을 때, 액세스 가능하거나 보호된 메모리 영역에 단지 보통문으로 Nonce(420)을 저장하는 것이 이용 경우 요건들에 의존하여, 유리할 수 있다.

[0032] [0035] 복호화는 역방향으로 작동한다. 예를 들어, 도 4b를 참조하여, 입력들(440)은 도 4a의 암호화의 출력이고, 출력들(410)은 원래의 입력들(즉, 원래 평문 입력들)에 대응해야 한다.

[0033] *자원 절감*

[0034] [0036] 이전의 방식들 모두는, 평문이 암호화 함수에 의해 직접 암호화된다는 아이디어에 기초한다는 것이 이해되어야 한다. 그러나 블록 암호들에 대한 동작들의 여러 모드들은 암호문을 유도하기 위해(예를 들어, 카운터(CTR) 모드) 평문에 대해 XOR 연산되는 키 스트림을 생성하도록 암호 프리미티브를 이용한다. 이러한 타입의 암호화의 예들이 아래에서 설명될 것이다. 키 스트림 생성을 위한 자원들을 절감하도록 시도할 때, 너무 많은 절감이 보안성을 비용으로 발생하지 않는다는 것, 즉 다양한 키 스트림 블록들이 서로 상관되지 않게 나타나야 한다는 것이 보장될 필요가 있다. 예를 들면, 여러 입력 블록들을 암호화하기 위해 "키 스트림"으로부터의 블록을 재사용하는 것이 매력적일 수 있는데 - 메모리 암호화의 시나리오에서, 이것은 메모리 암호화 회로들의 영역의 문제를 쉽게 해결할 수 있다. 그러나 평문의 두 블록들(P1 및 P2)이 동일한 패드 π로, 둘 다 XOR 연산되는 경우, 암호문 블록은 $P1 \oplus P2 = C1 \oplus C2$ 를 만족하는 $C1 = P1 \oplus \pi$ 및 $C2 = P2 \oplus \pi$ 가 될 것이다. 이는 평문에 대한 중요한 정보를 드러낼 수 있고 따라서 중요한 정보를 저장하기에 적합하지 않

다. 그러나 키 스트림의 두 개 이상의 블록들 중 제 1 라운드만을 계산하고 그 후 마지막 라운드를 개별적으로 수행하는데 공통 하드웨어를 이용하는 것이 유리할 수 있다. 이러한 방법의 보안성은 사용된 암호의 감소된 라운드 버전의 암호해독(cryptanalysis) 및 일부 라운드 후 중간 값의 예측성에 의존한다.

[0035] [0037] 이것의 예는 도 5를 참조하여 표시된다. 이 예시적인 실시예에서 프로세스(500)에서, 입력(502) 및 Nonce(v)(520)는 L개의 키 스트림 블록들을 생성하는데 사용된 값들이다. 입력(502)은 평문이 아니다. 유사하게, Output1, Output2,…OutputL(540)은 암호문이 아니라, 암호문의 L 블록들은 CTR 동작 모드에서와 같이 이들 값들에 대해 XOR 연산된다(또는 블록 암호의 암호화 프리미티브만을 이용하는 다른 암호화 모드들의 일부 변형들에서 더 복잡한 방식으로 이용됨). 다른 양상들에서, 도 5는, Nonce(520) 이전에 라운드 키들의 제 1 라운드(M 라운드들(504))가 적용되고 그 후 Nonce(520)가 적용되어 Nonced 데이터 출력들을 생성하는 것을 포함해서, 도 4a와 유사하다. Nonce(520)는 Nonced 출력을 생성하기 위해 함수들(f1, f2,.., fL)(522)에 의해 적용될 수 있다. Nonce(520)를 적용할 함수는 XOR 함수를 포함할 수 있다. Nonced 출력들은 그 후, 암호화된 출력(540)을 생성하기 위해 라운드 키들의 제 2 라운드(N-M 라운드들(506))에 의해 암호화 수 있다.

[0036] AES(예를 들어, AES-128)가 블록 암호로 선택되는 경우, 현재 암호 해독 결과의 관점에서 M=3 또는 4가 사용될 수 있다. 6 또는 7 라운드로 감소되는 AES-128가 공격하는데 여전히 상당히 어렵게 되는 근거는, 공격자는 입력을 제어 수 있는 경우에만 가능하다는 것인데, 이는 이 상황에서 가능하지 않다. 예를 들어, 이용 경우가 전체 캐시 라인들이 암호화되는 메모리 암호라고 가정하면, 이들은 128 바이트이고, 따라서 우리는 8 블록(L = 8)이 필요하다. 이는 M = 3인 경우, 80 대신에, AES의 총 3 + 8 * 7 = 59 라운드들이 HW로 구현될 필요가 있는 것을 의미하며, 이는 약 26%의 영역 및 전력을 절감하게 한다. M = 4인 경우, 약 35%의 절감을 위해, 구현되는 AES의 라운드의 수는 4 + 8* 6 = 52이다. 절감들은 마지막 N - M 라운드들에 대한 키 스케줄이 모든 파이프라인들에 대해 공통적인 경우 더 큰 비트가 될 수 있는데 - 아마도 병렬 파이프 라인에서 라운드 키들의 일부 고정된 비트 치환들을 갖지만 어쩌면 그보다 크지 않음 - , 이는 (M + 1)번째 라운드로의 입력들에 대해 XOR 연산되는 값들과 상이한 nonce로부터 도출하기 위한 로직에 의한 오프셋을 그것이 초과해야 하기 때문이다.

[0037] *Nonce의 계산*

[0038] [0038] 일 실시예에서, 새로운 블록(또는 L 블록들의 세트)이 메모리에 기록될 필요가 있을 때마다, Nonce이 갱신될 수 있다. 블록 암호가 충분한 확산을 갖는 경우(또는 그것이 마지막 N - M 라운드들에서 충분한 확산을 갖는 경우), Nonce을, 예를 들면, s 비트들 만큼만 시프트하고, 그 후 Nonce에 s의 새로운 신선한 랜덤 비트들을 추가하는 것이 충분할 수 있다. 예를 들어, Nonce에 대해 $v \leftarrow (v << s) \oplus r$ 로서 계산될 수 있으며, 여기서 r는 s 비트들의 스트링이다. 또한, 신선한 비트들은 최상위 포지션으로부터 시프트될 수 있거나 또는 v는 독립적으로 시프트되고 갱신되는 다양한 서브-레지스터들에서 분할될 수 있다. 그러나 이 전략이 사용되는 경우, Nonce은 보통문으로 저장돼서는 안 되고 암호화되어야 하는데, 그 이유는 이들을 보통문으로 저장하는 것은 미래의 Nonce들을 부분적으로 예측 가능하게 할 가능성이 있어서, 가능하게는 암호해석을 돕는다. Nonce는 (a) 데이터가 저장될 위치인 물리적 메모리 어드레스에 독립적인 값이 되거나 또는 (b) 그 어드레스에 종속적이 될 수 있다는 것이 추가로 주목되어야 한다. 후자의 경우에 대해, 그것은 (i)물리적 메모리 어드레스 및 (ii) 랜덤 값, (암호화된) 카운터, 또는 위에서 설명된 방법들에 의해 또는 다른 방법에 의해 계산된 값의 연결일 수 있다.

[0039] *예시적인 하드웨어*

[0040] [0039] 위에서 설명된 방법들 및 프로세스들을 구현할 수 있는 예시적인 컴퓨터 하드웨어(600)가 도 6에서 예시된다. 컴퓨터 시스템(600)은 버스를 통해 전기적으로 커플링(또는 그렇지 않으면, 적절한 통신할 수 있음)될 수 있는 하드웨어 엘리먼트들을 포함하는 것으로 도시된다. 하드웨어 엘리먼트들은 적어도 하나 메인 프로세서(602)(예를 들어, 중앙 처리 장치(CPU))뿐만 아니라 다른 프로세서들(604)을 포함할 수 있다. 이들 프로세서들은 범용 프로세서들 및/또는 하나 이상의 특수-목적 프로세서들(예컨대, 디지털 신호 프로세싱 칩, 그래픽 가속 프로세서들 등)일 수 있다는 것이 인지되어야 한다. 프로세서들은 각각의 메모리 관리 유닛들(MMU를)(610)에 커플링될 수 있으며, 이 MMU들(610)은 결국, (점선에 둘러싸인) 캐시(612)(예를 들어, 캐시들은 또는 존재하거나 존재하지 않을 수 있고 그리고/또는 별개이거나 또는 다른 엘리먼트들에 통합될 수 있음)를 통해, 암호화기 프로세싱 유닛(620)에 및/또는 메모리(630) 및/또는 저장 디바이스들(640)에 커플링될 수 있다. 아래에서 설명될 바와 같이, 암호화기(620)는 메모리에 저장될 데이터에 대한 메모리 암호화를 강화하기 위해 암호 블록들의 기능성을 확장하도록 이전에 기술된 방법들 및 프로세스들을 이용할 수 있다.

[0041] [0040] 컴퓨터(600)는 다른 디바이스들(도시되지 않음), 예컨대, 입력 디바이스들(예를 들어, 키보드, 마우스, 키패드, 마이크로폰, 카메라 등); 출력 디바이스들(예를 들어, 디스플레이 디바이스, 모니터, 스피커, 프린터 등)을 포함할 수 있다는 것이 이해되어야 한다. 컴퓨터(600)는 추가로, 하나 이상의 메모리 엘리먼트들, 저장 디바이스들(630, 640)을 포함(및/또는 이들과 통신함)할 수 있으며, 이는 로컬 및/또는 네트워크 액세스 가능 저장소를 포함할 수 있고 및/또는, 제한 없이, 디스크 드라이브, 드라이브 어레이, 광학 저장 디바이스, 고상 저장 디바이스, 예컨대, 랜덤 액세스 메모리("RAM") 및/또는 판독-전용 메모리("ROM")을 포함할 수 있으며, 이들은, 프로그래밍 가능하고, 플래시-업데이트 가능하고 기타 등등이 가능할 수 있다. 컴퓨터(600)는 또한 모뎀, 네트워크 카드(무선 또는 유선), 적외선 통신 디바이스, 무선 통신 디바이스 및/또는 칩셋(예컨대, 블루투스 디바이스, 802.11 디바이스, Wi-Fi 디바이스, WiMax 디바이스, 셀룰러 통신 디바이스 등) 등을 포함할 수 있는 통신 서브시스템을 포함할 수 있다. 통신 서브시스템은 데이터가 네트워크, 다른 컴퓨터 시스템들 및/또는 여기서 설명된 임의의 다른 디바이스들과 교환되도록 허용할 수 있다. 컴퓨터(600)는 모바일 디바이스, 비-모바일 디바이스, 무선 디바이스, 유선 디바이스 등일 수 있고, 무선 및/또는 유선 연결들을 가질 수 있으며, 임의의 타입의 전자 또는 컴퓨팅 디바이스일 수 있다는 것이 이해되어야 한다.

[0042] [0041] 일 실시예에서, 데이터가 암호화되는 위치에 저장되는 경우(결정 블록(650)), 암호화기(620)(예를 들어, 데이터를 암호화하기 위한 디바이스)는, 복수의 평문 데이터 입력들(Input1-InputL)(110)을 수신하는 것; 랜덤화되는 Nonced 평문 데이터 출력을 생성하도록 함수들(f1, f2,., fL)(122)을 통해 Nonce(122)를 적용하는 것; 및 암호화된 출력 데이터(Output1-OutputL)(140)가 메모리(630)에 출력되도록 Nonced 평문 데이터 출력에 암호화 함수(130)를 적용하는 것을 포함하는, (도 1a을 또한 참조하여) 이전에 설명된 프로세스를 구현할 수 있다. 이 데이터는 추가로 저장소(640)에 저장될 수 있다. 다른 실시예들에서, 이전에 설명된 바와 같이, 암호화 함수를 적용하기 위해 암호화기(620)는 Nonce가 적용되기 이전에 암호화 함수를 모델링하는 라운드 함수들의 제 1 시퀀스를 이용하여 평문 데이터 입력들을 암호화할 수 있다. 그 후, Nonced 데이터 출력을 생성하도록 Nonce가 적용된다. Nonced 데이터 출력들은 그 후 메모리(630)에 출력된 암호화된 출력 데이터를 생성하기 위해 암호화 함수를 모델링하는 라운드 함수의 제 2 시퀀스에 의해 암호화될 수 있다. 이러한 구현 예들은 앞서 상세히 설명된 바와 같이 도 2 내지도 5에서 예시된다.

[0043] [0042] 그러나 결정 블록(650)에서, 암호화되는 위치에 데이터가 저장되지 않은 것으로 결정되는 경우, 데이터는 일반적으로 메모리(630)에 저장될 수 있고 및/또는 보통의 메모리 맵핑 입력들/출력들 및 제어(655)는 저장소(640)에 대한 직접적인 메모리 액세스(DMA) 제어를 구현하기 위해 사용될 수 있다.

[0044] [0043] 일반적으로, 메모리 암호화가 이용 가능할 때, 그의 콘텐츠는 가상 메모리 시스템의 저장 디바이스에 이들이 기록하기 이전에 복호화될 필요가 있다. 그러나 이를 수용하기 위해, 본 발명의 실시예들에 따라, DMA 데이터 전달 채널은 메모리(630)(예컨대, RAM, DDR RAM 등)의 실제 암호화된 콘텐츠를 판독하는데 사용될 수 있고 저장 디바이스(640)(예를 들어, 하드 드라이브 또는 플래시 메모리)의 섹터에 이들을 기록하는 것은 물론, 섹터로부터 판독하고, 직접적으로 메모리(630)에 콘텐츠를 넣는데 사용될 수 있다. 따라서, 이러한 메모리 암호화 방법들은 물리적 어드레스들과 독립적일 수 있고, 페이지들은 부가적인 암호화/복호화 오버헤드 없이 밖으로 그리고 다시 안으로 스왑(swap)될 수 있다.

[0045] [0044] 위에서 설명된 시스템의 이점은, 메모리 콘텐츠들이 스왑 파일로 그리고 메모리로 다시 이동될 때마다 그들이 복호화 및 재-암호화될 필요가 없다는 것이며, 이는 상당한 전력 절감 및 시간 절감들을 발생시킨다. 또한, 여기서 설명되는 기술들은 물리적 또는 전기적 메모리 공격들 - 즉, 메모리의 직접적인 판독에 대해 - 에 대한 양호한 직접적인 보호를 제공할 뿐만 아니라, 동일한 위치에 대해 동일하거나 상관된 데이터의 반복된 기록들이 효과적으로 랜덤화되기 때문에 사이드 채널로서 버스 트래픽을 이용하는 공격에 대해 저항성을 제공한다. 또한, 여기서 설명되는 기술들은 비교적 작은 부가적인 하드웨어 구현을 요구한다. 또한, 여기서 설명되는 기술들은 충분히 일반적이어서, 이들은 본질적으로 임의의 흔히-사용되는 블록 암호에 적용될 수 있게 된다. 부가적으로, 각각의 라운드의 입력 및 출력 크기들은 모두 동일해야 할 필요는 없고 마스킹 동작들은 이러한 경우들에서 최소한으로만 적응되어야 한다. 또한, 암호화된 메모리를 절감하기 위한 직접적인 DMA 채널은 또한 전력 소비 및 시간 면에서 상당한 절감을 가져올 수 있다.

[0046] [0045] 또한, 이전에 설명된 바와 같이, Nonce은 구현에 의존하여, 명확한 암호화되지 않은 방식으로 또는 암호화된 방식으로, 메인 메모리(630)에 저장될 수 있다. 대안적으로 이전에 설명된 바와 같이, Nonce은 특수한 메모리의 작은 보호된 영역에 저장될 수 있다.

[0047] [0046] 또한, 일 예에서, 고정 키가 디바이스 부팅 시에 랜덤으로 선택되는 경우, 대응하는 키 스케줄은 그 시

간에 미리 계산될 수 있다는 것이 이해되어야 한다. 특정 예로서, 마스터 키 또는 필요한 경우, 키에 배치될 수 있는 메모리 어드레스의 종속성이 있을 수 있다. 추가로 예로서, Nonce는 페이지 값 당 고정된 값(이 경우, 모든 도출된 상수들, 예컨대, 함수들(f1, f2,..,fL)의 출력들이 미리 계산될 수 있음)일 수 있거나, 또는 물리적 메모리 어드레스에 의존할 수 있다. 이러한 예시적인 방식들은 단순화 목적들을 위해 사용될 수 있다.

[0048] [0047] 이전에 설명된 바와 같이, 블록 암호의 기능성을 연장함으로써 메모리에 저장된 데이터의 보호를 위한 강화된 메커니즘을 제공하기 위한 기술들은 소프트웨어, 펌웨어, 하드웨어, 이들의 결합들 등으로서 구현될 수 있다는 것이 이해되어야 한다. 일 실시예에서, 이전에 설명된 기능들은 이전에 요구되는 기능(예를 들어, 도 1-5의 방법 동작들)을 달성하기 위해 컴퓨터(600)의 하나 이상의 프로세서들(예를 들어, 암호화기(620) 또는 다른 프로세서들)에 의해 구현될 수 있다. 또한, 도 1-5를 참조하여 이전에 설명된 바와 같이, 복호화는 단순히 역방향으로 작동한다.

[0049] [0048] 이전에 설명된 바와 같이, 이전에 설명된 본 발명의 양상들은 디바이스의 프로세서들에 의해 명령들의 실행과 함께 구현될 수 있다는 것이 이해되어야 한다. 특히, 프로세서를 포함하지만 이에 한정되지 않는 디바이스들의 회로는, 본 발명의 실시예들에 따른 방법이나 프로세스를 실행하기 위해, 명령의 실행, 루틴 또는 프로그램의 제어 하에 동작할 수 있다. 예를 들어, 이러한 프로그램은, (예를 들어 메모리 및/또는 다른 위치에 저장되는) 소프트웨어 또는 펌웨어로 구현될 수 있고, 프로세서들, 및/또는 디바이스들의 다른 회로에 의해 구현될 수 있다. 또한, 용어 프로세서, 마이크로프로세서, 회로, 제어기 등은 로직, 명령, 명령어, 소프트웨어, 펌웨어, 기능성 등을 실행할 수 있는 임의의 타입의 로직 또는 회로를 지칭한다는 것이 이해되어야 한다.

[0050] [0049] 디바이스가 모바일 또는 무선 디바이스인 경우, 이들이 하나 이상의 무선 통신 링크들을 경유하여, 임의의 적절한 무선 통신 기술에 기초하거나 또는 그렇지 않고 이를 지원하는 무선 네트워크 통해 통신할 수 있다는 것이 이해되어야 한다. 예를 들어, 일부 양상들에서, 무선 디바이스 및 다른 디바이스들은 무선 네트워크를 포함하는 네트워크와 연관될 수 있다. 일부 양상들에서, 네트워크는 인체 영역 네트워크 또는 개인 영역 네트워크(예를 들어, 울트라 광대역 네트워크)를 포함할 수 있다. 일부 양상들에서, 네트워크는 로컬 영역 네트워크 또는 광역 네트워크를 포함할 수 있다. 무선 디바이스는 예를 들어, 3G, LTE, 어드밴스드 LTE, 4G, CDMA, TDMA, OFDM, OFDMA, WiMAX, 및 WiFi와 같은 다양한 무선 통신 기술들, 프로토콜들 또는 표준들 중 하나 이상을 지원하거나, 또는 그렇지 않고 이용할 수 있다. 유사하게, 무선 디바이스는 다양한 대응하는 변조 또는 멀티플렉싱 방식들 중 하나 이상을 지원하거나, 또는 그렇지 않고 이용할 수 있다. 무선 디바이스는 이에 따라 위의 또는 다른 무선 통신 기술들을 사용하여 하나 이상의 무선 통신 링크들을 통해 설정하고 통신하기 위해 적절한 컴포넌트들(예를 들면, 공중 인터페이스)을 포함할 수 있다. 예를 들어, 디바이스는 무선 매체를 통한 통신을 용이하게 하는 다양한 컴포넌트들(예를 들어, 신호 생성기들 및 신호 프로세서들)을 포함할 수 있는 연관된 전송기 및 수신기 컴포넌트들(예를 들어, 전송기 및 수신기)을 갖는 무선 트랜시버를 포함할 수 있다. 잘 알려진 바와 같이, 모바일 무선 디바이스는, 그에 따라 다른 모바일 디바이스, 셀 폰들, 다른 유선 및 무선 컴퓨터들, 인터넷 웹 사이트 등과 무선으로 통신할 수 있다.

[0051] [0050] 본 발명의 교시는 다양한 장치들(예를 들어, 디바이스들)에 통합(예를 들어, 다양한 장치들 내에 구현 또는 이에 의해 수행)될 수 있다. 예를 들어, 여기에서 교시된 하나 이상의 양상들이 컴퓨터, 유선 컴퓨터, 무선 컴퓨터, 전화(예를 들어, 셀룰러 전화), 개인용 디지털 보조기기("PDA"), 태블릿, 모바일 컴퓨터, 모바일 디바이스, 비-모바일 디바이스, 유선 디바이스, 무선 디바이스, 랩톱 컴퓨터, 엔터테인먼트 디바이스(예를 들어, 음악 또는 비디오 디바이스), 헤드셋(예를 들어, 헤드폰들, 이어폰, 등), 의료용 디바이스(예를 들어, 생체 센서, 심박수 모니터, 보수계, EKG 디바이스 등), 사용자 I/O 디바이스, 고정된 컴퓨터, 데스크톱 컴퓨터, 서버, POS(point-of-sale) 디바이스, 엔터테인먼트 디바이스 셋톱 박스, ATM, 또는 임의의 다른 적합한 전자/컴퓨팅 디바이스에 통합될 수 있다. 이들 디바이스들은 상이한 전력 및 데이터 요구 사항을 가질 수 있다.

[0052] [0051] 일부 양상들에서, 무선 디바이스는 통신 시스템을 위한 액세스 디바이스(예를 들어, Wi-Fi 액세스 포인트)를 포함할 수 있다. 이러한 액세스 디바이스는, 유선 또는 무선 통신 링크를 통해, 예를 들어, 다른 네트워크(예를 들어, 광역 네트워크, 이를테면, 인터넷 또는 셀룰러 네트워크)에 대한 접속을 제공할 수 있다. 따라서, 액세스 디바이스는 다른 디바이스(예를 들어, WiFi 스테이션)가 다른 네트워크 또는 일부 다른 기능에 액세스하게 할 수 있다.

[0053] [0052] 당업자는, 정보 및 신호들이 다양한 상이한 기술들 및 기술들을 이용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 전술한 설명을 통해 참조될 수 있는 데이터, 명령들, 커멘드들, 정보, 신호들, 비트들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장 또는 입자들, 광학장 또는 입자들, 또는 이들의 임의의 조

합으로 표현될 수 있다.

[0054] [0053] 본원에 개시된 실시예와 관련하여 설명된 다양한 예시적인 논리 블록, 모듈, 회로, 및 알고리즘 단계는, 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이 둘의 조합으로 구현될 수 있다는 것을 당업자는 추가로 이해할 것이다. 하드웨어와 소프트웨어의 상호 교환 가능성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들이 이들의 기능성의 관점에서 일반적으로 상술되었다. 이러한 기능이 하드웨어 또는 소프트웨어로 구현되는지 여부는 전체 시스템에 부과된 특정 애플리케이션 및 설계 제약들에 의존한다. 당업자는 각각의 특정 애플리케이션마다 다양한 방식들로 설명된 기능을 구현할 수 있지만, 이러한 구현 결정은 본 발명의 범위를 벗어나게 하는 것으로 해석되어서는 안 된다.

[0055] [0054] 본 명세서에 개시된 실시예와 관련하여 설명된 다양한 예시적인 논리 블록, 모듈, 및 회로들은, 범용 프로세서, 디지털 신호 프로세서(DSP), 주문형 집적 회로(ASIC), 필드 프로그래밍 가능 게이트 어레이(FPGA), 또는 기타 프로그래밍 가능 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트, 또는 본원에 설명된 기능을 수행하도록 설계된 이들의 임의의 결합으로 구현 또는 수행될 수 있다. 범용 프로세서는 마이크로프로세서일 수 있지만, 대안적으로, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로 제어기, 또는 상태 머신일 수 있다. 프로세서는 또한 컴퓨팅 장치들의 조합, 예를 들어, DSP와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합된 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성으로 구현될 수 있다.

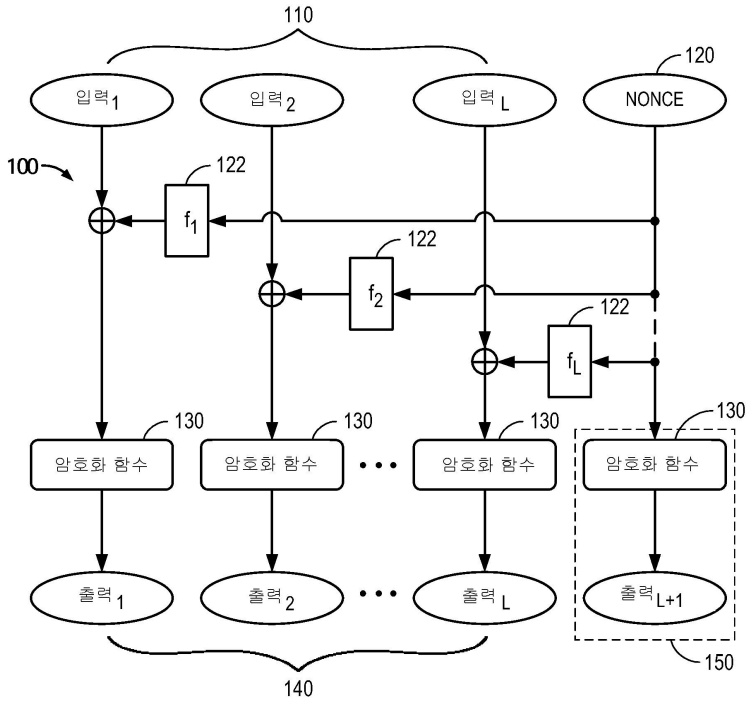
[0056] [0055] 본원에 개시된 실시예들과 관련하여 설명된 방법 또는 알고리즘의 단계들은 하드웨어, 프로세서에 의해 실행되는 소프트웨어 모듈, 또는 이 둘의 조합으로 직접 구현될 수 있다. 소프트웨어 모듈은, RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터들, 하드 디스크, 착탈식 디스크, CD-ROM, 또는 당업계에 공지된 저장 매체의 임의의 다른 형태에 상주할 수 있다. 예시적인 저장 매체는 프로세서에 커플링되어, 프로세서는 저장 매체로부터 정보를 판독하고, 저장 매체에 정보를 기록할 수 있다. 대안적으로, 저장 매체는 프로세서에 통합될 수 있다. 프로세서 및 저장 매체는 ASIC에 상주할 수 있다. ASIC는 사용자 단말에 상주할 수 있다. 대안적으로, 프로세서 및 저장 매체는 사용자 단말에 개별 컴포넌트로서 상주할 수 있다.

[0057] [0056] 하나 이상의 예시적인 실시예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수 있다. 컴퓨터 프로그램 제품으로서 소프트웨어로 구현되는 경우, 기능들은, 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 전송될 수 있다. 컴퓨터 판독 가능 매체는 한 장소에서 다른 장소로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체 및 컴퓨터 저장 매체 둘 다를 포함한다. 저장 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 이용 가능한 매체일 수 있다. 제한적이지 않은 예로서, 이러한 컴퓨터 판독 가능 매체는, RAM, ROM, EEPROM, FLASH 메모리, CD-ROM 또는 다른 광학 디스크 저장소, 자기 디스크 저장소 또는 다른 자기 저장 디바이스, 또는 명령 또는 데이터 구조의 형태로 원하는 프로그램 코드를 전달하거나 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스 될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독 가능 매체라고 적절하게 지칭된다. 예를 들어, 소프트웨어가 동축 케이블, 광섬유 케이블, 꼬임 쌍선, 디지털 가입자 회선(DSL), 또는 무선 기술들(이를테면, 적외선, 라디오, 마이크로파)을 이용하여 웹사이트, 서버 또는 다른 원격 소스로부터 전송된다면, 동축 케이블, 광섬유 케이블, 꼬임 쌍선, DSL 또는 무선 기술들(이를테면, 적외선, 라디오, 마이크로파)이 매체의 정의에 포함된다. 본원에 사용되는 디스크(disk 및 disc)는 컴팩트 디스크(disc)(CD), 레이저 디스크(disc), 광학 디스크(disc), 디지털 다용도 디스크(disc)(DVD), 플로피 디스크(disk) 및 블루레이 디스크(disc)를 포함하며, 여기서 디스크(disk)들은 일반적으로 데이터를 자기적으로 재생하는 한편, 디스크(disc)들은 데이터를 레이저를 이용하여 광학적으로 재생한다. 이들의 결합은 또한 컴퓨터 판독 가능 매체들의 범위 내에 포함되어야 한다.

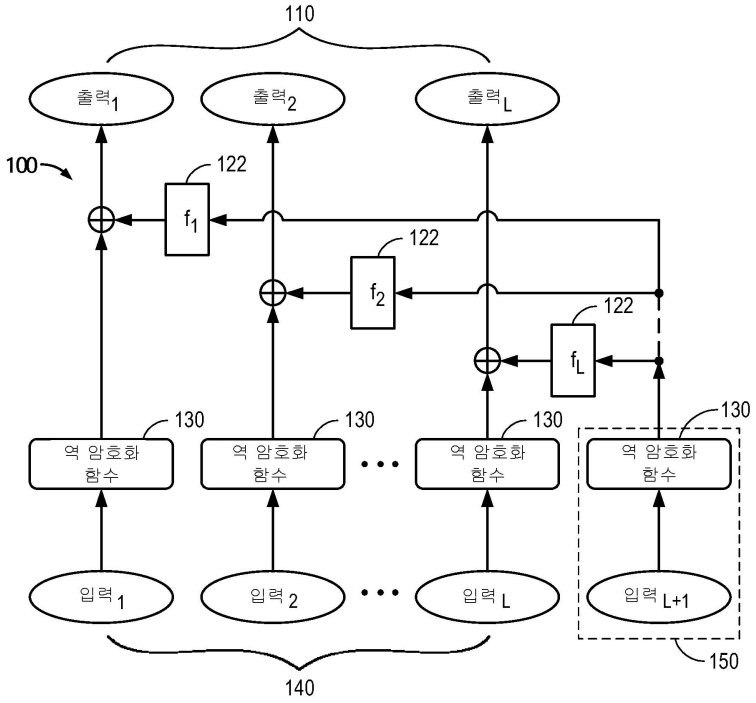
[0058] [0057] 개시된 실시예의 이전 설명은 당업자가 본 발명을 실시하거나 이용할 수 있도록 제공된다. 이들 실시예들에 대한 다양한 변형은 당업자에게 쉽게 명백할 이며, 여기에 정의된 일반적인 원리들은 본 발명의 사상 또는 범위를 벗어나지 않고 다른 실시예에 적용될 수도 있다. 따라서, 본 발명은 여기에 도시된 실시예들로 한정되도록 의도되는 것이 아니라, 본원에 개시된 원리들 및 신규한 특징들과 일치하는 최광의 범위와 일치하여야 한다.

도면

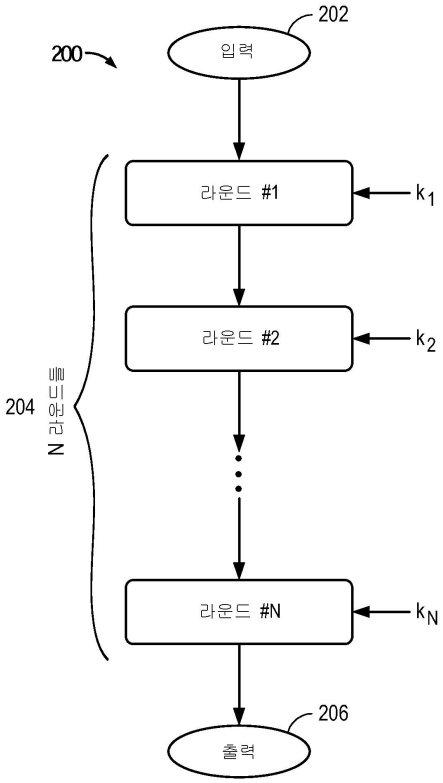
도면1a


110
입력 1
입력 2
입력 L
NONCE
120
100
122
f1
122
f2
122
fL
130
암호화 함수
130
암호화 함수
130
암호화 함수
130
암호화 함수
출력 1
출력 2
출력 L
출력 L+1
140
150

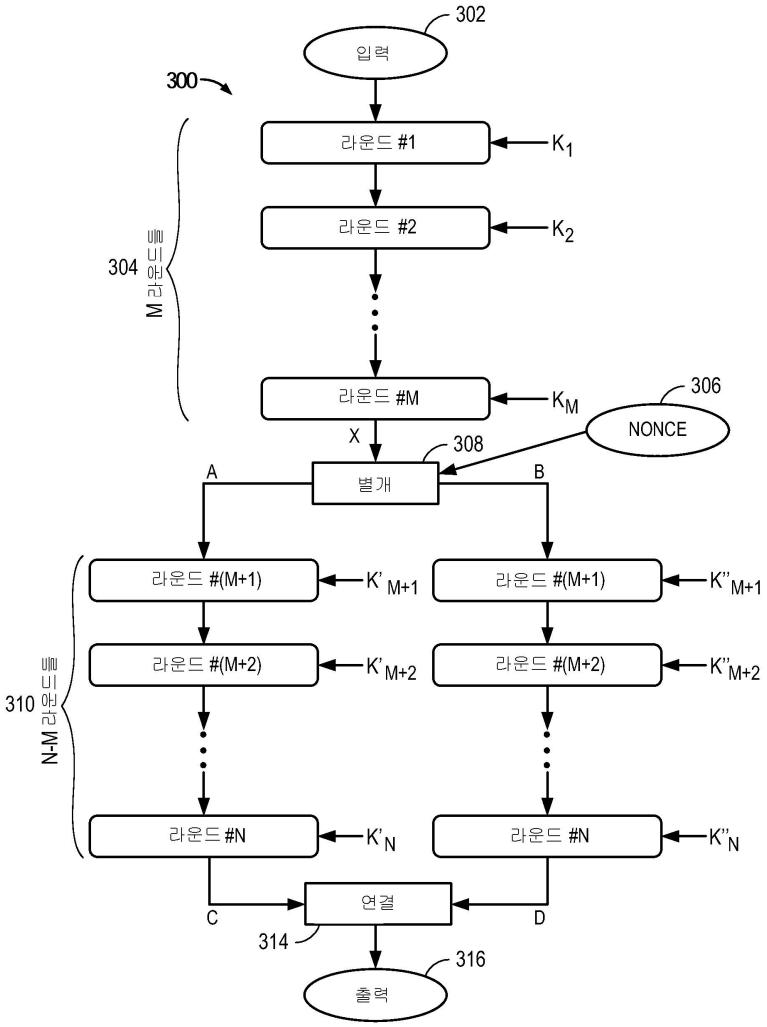
도면1b


110
출력 1
출력 2
출력 L
100
122
f1
122
f2
122
fL
130
역 암호화 함수
130
역 암호화 함수
130
역 암호화 함수
130
역 암호화 함수
입력 1
입력 2
입력 L
입력 L+1
140
150

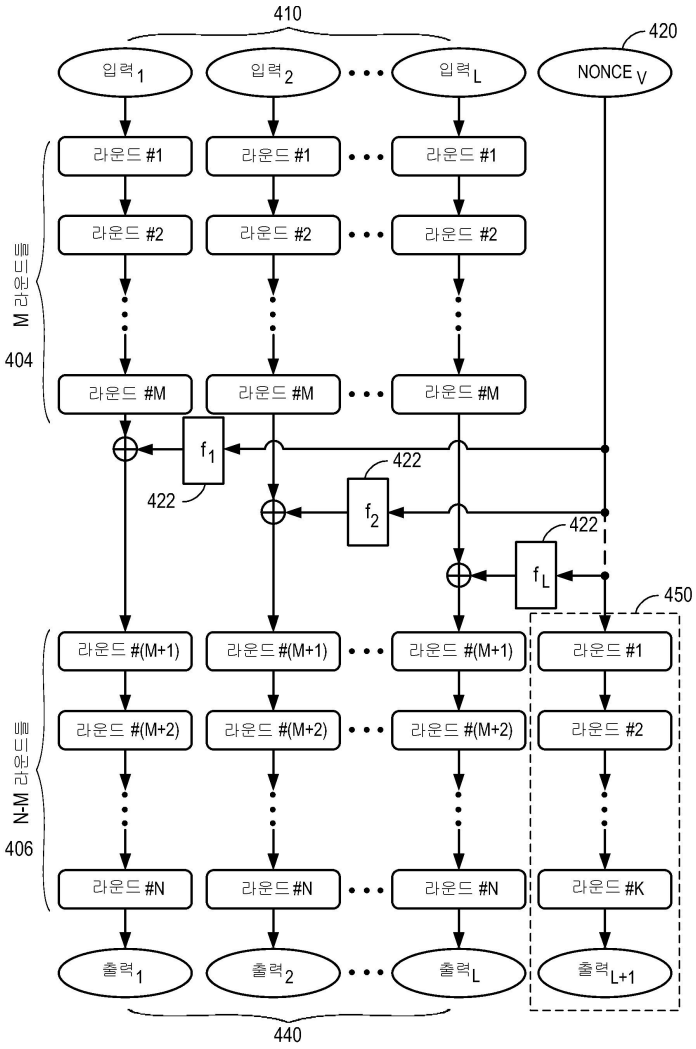
도면2



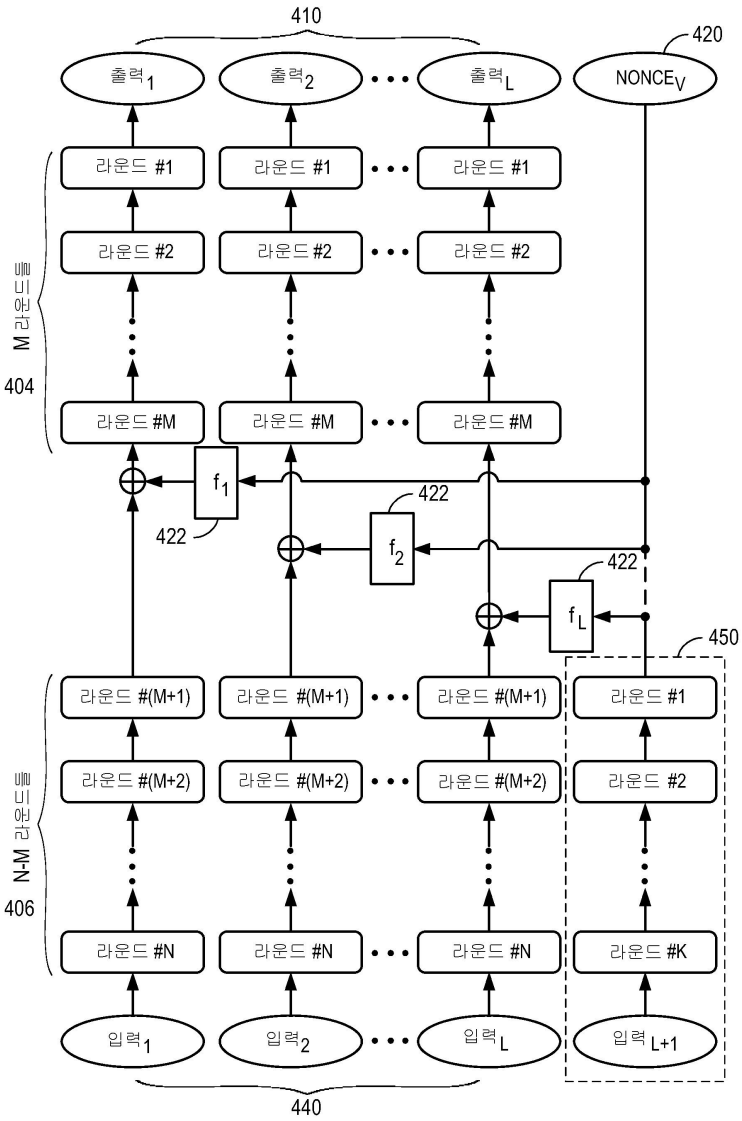
도면3


302
입력
300
라운드 #1
K_1
라운드 #2
K_2
304 M 라운드들
라운드 #M
K_M
306
NONCE
X
308
별개
A
B
310 N-M 라운드들
라운드 #(M+1)
K'_{M+1}
라운드 #(M+1)
K''_{M+1}
라운드 #(M+2)
K'_{M+2}
라운드 #(M+2)
K''_{M+2}
라운드 #N
K'_N
라운드 #N
K''_N
C
연결
D
314
316
출력

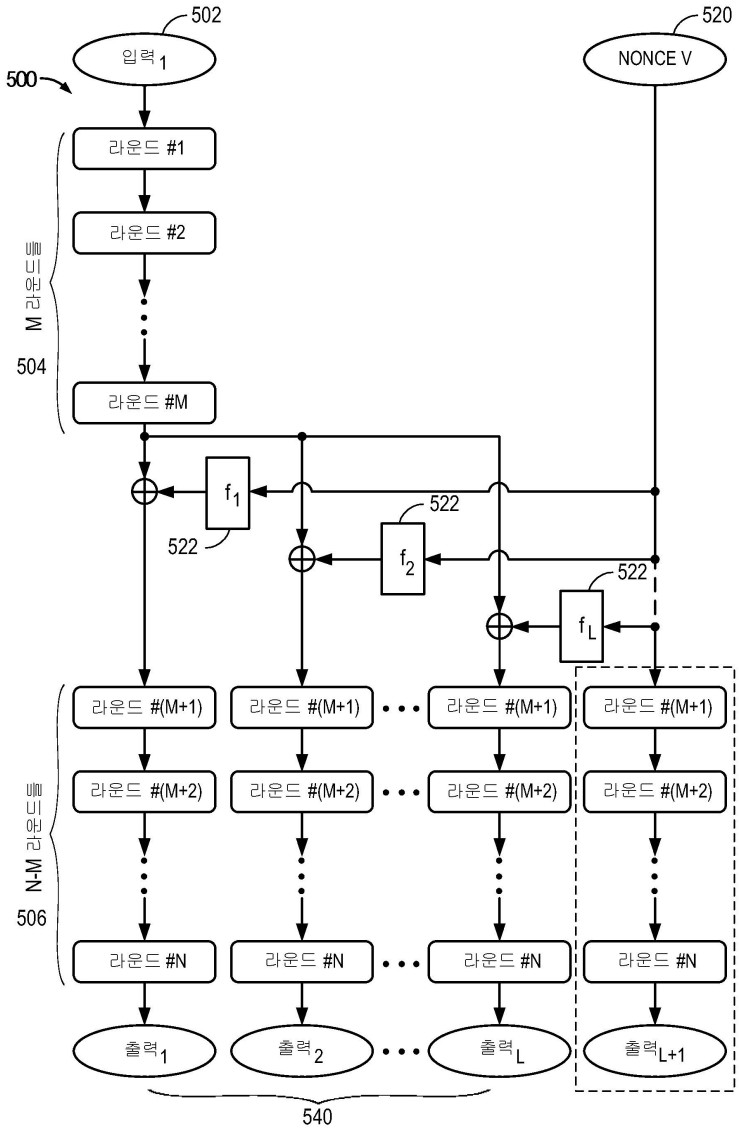
도면4a


410
420
입력 1
입력 2
입력 L
NONCE V
라운드 #1
라운드 #2
라운드 #M
M 라운드들
404
f1
f2
fL
422
450
라운드 #(M+1)
라운드 #(M+2)
라운드 #N
라운드 #K
N-M 라운드들
406
출력 1
출력 2
출력 L
출력 L+1
440

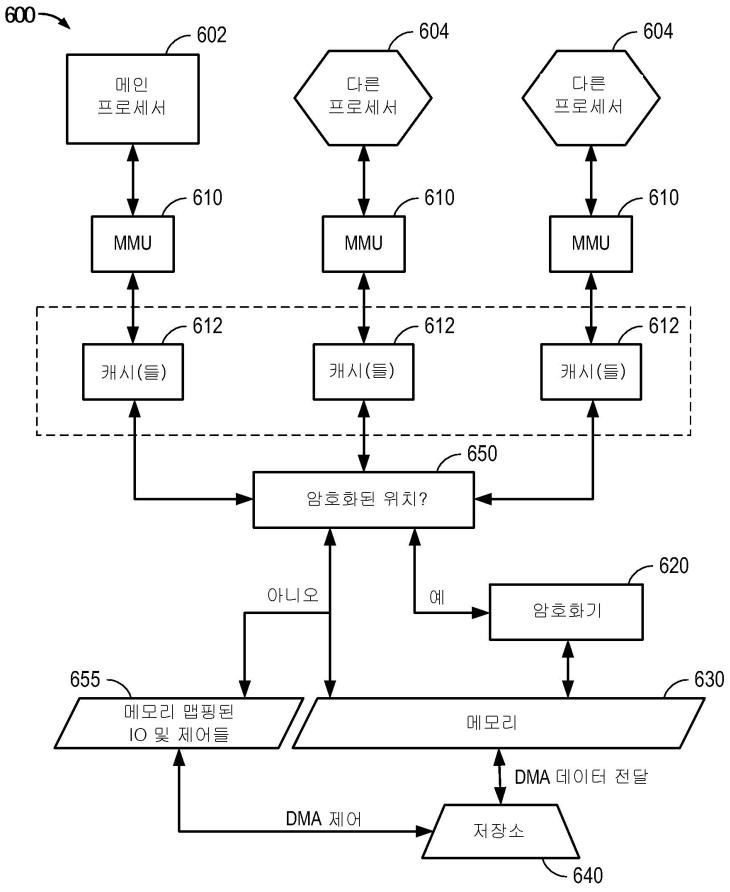
도면4b


410
420
출력1
출력2
출력L
NONCEV
라운드 #1
라운드 #2
라운드 #M
M 라운드들
404
f1
f2
fL
422
450
라운드 #(M+1)
라운드 #(M+2)
라운드 #N
라운드 #K
N-M 라운드들
406
입력1
입력2
입력L
입력 L+1
440

도면5


502
입력 1
500
520
NONCE V
라운드 #1
라운드 #2
라운드 #M
M 라운드들
504
f1
522
f2
fL
라운드 #(M+1)
라운드 #(M+2)
라운드 #N
N-M 라운드들
506
출력1
출력2
출력L
출력L+1
540

도면6


600
602
메인
프로세서
604
다른
프로세서
604
다른
프로세서
610
MMU
610
MMU
610
MMU
612
캐시(들)
612
캐시(들)
612
캐시(들)
650
암호화된 위치?
아니오
예
620
암호화기
655
메모리 맵핑된
IO 및 제어들
630
메모리
DMA 데이터 전달
DMA 제어
저장소
640