(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0079664
(43) 공개일자 2009년07월22일

(51) Int. Cl.
G06F 9/06 (2006.01) *G06F 7/00* (2006.01)
(21) 출원번호 10-2008-0005799
(22) 출원일자 2008년01월18일
심사청구일자 2008년01월18일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가1 고려대학교 내
(72) 발명자
김희석
서울 강서구 화곡5동 스카이뷰
김태현
서울 강서구 화곡2동 846-19
(뒷면에 계속)
(74) 대리인
현종철

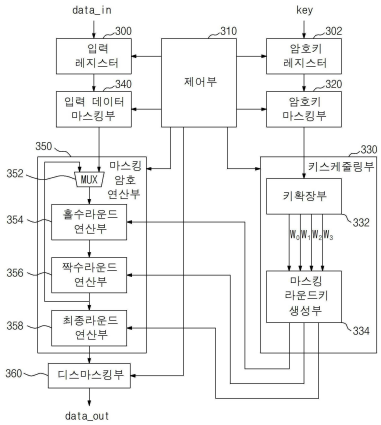
전체 청구항 수 : 총 20 항

(54) 아리아 마스킹 방법 및 이를 이용한 아리아 암호 장치 및방법

(57) 요 약

본 발명은 차분전력분석(DPA) 공격 및 키 스케줄링 공격에 안전한 아리아 마스킹 방법과 이를 이용한 아리아 암호 장치 및 방법에 관한 것이다. 본 발명에 따른 마스킹을 이용한 아라아 암호 장치는, 상기 장치를 제어하는 제어부로서, 마스킹 난수를 생성하고 상기 마스킹 난수와 두 종류의 에스박스를 이용하여 네 종류의 마스킹 에스박스를 생성하는 상기 제어부; 상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹부; 외부로부터 입력되는 암호키로부터 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성하는 키 스케줄링부; 및 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행하는 마스킹 암호연산부를 포함하며, DPA 공격 및 키 스케줄링 공격에 대한 안정성을 제공하고 하드웨어 자원 및 암호 연산시간을 감소시킨다.

대 표 도 - 도3



(72) 발명자

한동국

인천 계양구 작전2동 772번지 타임빌라 가동 201호

홍석희

서울 은평구 응암2동 451-12 202호

특허청구의 범위

청구항 1

마스킹을 이용하여 아리아(ARIA) 암호연산을 수행하는 장치에 있어서,

상기 장치를 제어하는 제어부로서, 마스킹 난수를 생성하고 상기 마스킹 난수와 두 종류의 에스박스(S-box; S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성하는 상기 제어부;

상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹부;

외부로부터 입력되는 암호키로부터 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성하는 키 스케줄링부; 및

상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행하는 마스킹 암호연산부를 포함하는 마스킹을 이용한 아리아 암호 장치.

청구항 2

제1항에 있어서,

상기 입력데이터 마스킹부는, 상기 마스킹 암호연산부가 수행하는 첫 번째 라운드 연산의 입력 값이 되는 상기 마스킹 입력데이터를 생성하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

청구항 3

제1항에 있어서,

상기 마스킹 암호연산부는, 상기 네 종류의 마스킹 에스박스에 의해 아리아 라운드 함수의 치환 계층(substitution layer)을 구성하여 상기 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 추가 마스킹을 수행하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

청구항 4

제1항에 있어서,

상기 마스킹을 이용한 아리아 암호 장치는, 상기 마스킹 암호연산부가 수행한 최종라운드 연산 결과에 대해 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력하는 디스마스킹부를 더 포함하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

청구항 5

제1항에 있어서,

상기 제어부가 생성하는 상기 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)는, 하기 수학식 1을 만족하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

[수학식 1]

$$MS_1(x \oplus m) = S_1(x) \oplus m',$$

$$MS_2(x \oplus m) = S_2(x) \oplus m',$$

$$MS_3(x \oplus m') = S_1^{-1}(x) \oplus m,$$

$$MS_4(x \oplus m') = S_2^{-1}(x) \oplus m$$

(상기 수학식 1에서, S_1 및 S_2는 상기 두 종류의 에스박스, S_1^{-1} 및 S_2^{-1}는 S_1 및 S_2에 대한 역치환 에스박스, m 및 m'는 상기 제어부가 생성한 제1 마스킹 난수 및 제2 마스킹 난수, 그리고 $\oplus$는 배타적 논리합 연산을 나타낸다)

청구항 6

제1항에 있어서,

상기 마스킹을 이용한 아리아 암호 장치는, 외부로부터 입력되는 상기 암호키를 상기 마스킹 난수에 의해 마스킹하여 마스킹 암호키(masked encryption key)를 생성하는 암호키 마스킹부를 더 포함하고, 상기 키 스케줄링부는,

상기 마스킹 암호키를 이용하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하는 키 확장부; 및

상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 상기 마스킹 라운드키를 생성하는 마스킹 라운드키 생성부를 포함하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

청구항 7

제6항에 있어서,

상기 암호키 마스킹부가 생성한 상기 마스킹 암호키는, 상기 암호키의 상위 128비트를 상기 마스킹 난수를 이용하여 마스킹한 값 및 상기 암호키의 상기 상위 128비트 이외의 나머지 비트와 0으로 패딩(padding)한 비트를 이용하여 구성되는 128비트를 상기 마스킹 난수를 이용하여 마스킹한 값을 포함하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

청구항 8

제7항에 있어서,

상기 키 확장부는, 상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이스텔(Feistel) 알고리듬을 이용하면서 추가 마스킹을 수행하여 상기 마스킹 난수에 의해 마스킹된 상기 확장키를 생성하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 장치.

청구항 9

제1항 내지 제8항 중 어느 한 항에 따른 마스킹을 이용한 아리아 암호 장치가 시스템 온 칩(system on chip) 형태로 구현된 마이크로프로세서.

청구항 10

마스킹을 이용하여 아리아(ARIA) 암호연산을 수행하는 방법에 있어서,

마스킹 난수를 생성하는 마스킹 난수 생성 단계;

상기 마스킹 난수와 두 종류의 에스박스(S-box; S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성하는 마스킹 에스박스 생성 단계;

상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹 단계;

외부로부터 입력되는 암호키로부터 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성하는 키 스케줄링 단계; 및

상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행하는 마스킹 암호연산 단계를 포함하는 마스킹을 이용한 아리아 암호 방법.

청구항 11

제10항에 있어서,

상기 입력데이터 마스킹 단계는, 상기 마스킹 암호연산 단계에서 수행되는 첫 번째 라운드 연산의 입력 값이 되는 상기 마스킹 입력데이터를 생성하는 단계인 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

청구항 12

제10항에 있어서,

상기 마스킹 암호연산 단계는, 상기 네 종류의 마스킹 에스박스에 의해 아리아 라운드 함수의 치환 계층(substitution layer)을 구성하여 상기 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 추가 마스킹을 수행하는 단계인 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

청구항 13

제10항에 있어서,

상기 마스킹을 이용한 아리아 암호 방법은, 상기 마스킹 암호연산 단계에서 수행된 최종라운드 연산 결과에 대해 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력하는 디스마스킹 단계를 더 포함하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

청구항 14

제10항에 있어서,

상기 마스킹 에스박스 생성 단계에서 생성되는 상기 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)는, 하기 수학식 2를 만족하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

[수학식 2]

$$MS_1(x \oplus m) = S_1(x) \oplus m',$$

$$MS_2(x \oplus m) = S_2(x) \oplus m',$$

$$MS_3(x \oplus m') = S_1^{-1}(x) \oplus m,$$

$$MS_4(x \oplus m') = S_2^{-1}(x) \oplus m$$

(상기 수학식 2에서, S_1 및 S_2는 상기 두 종류의 에스박스, S_1^{-1} 및 S_2^{-1}는 S_1 및 S_2에 대한 역치환 에스박스, m 및 m'는 상기 제어부가 생성하는 제1 마스킹 난수 및 제2 마스킹 난수, 그리고 $\oplus$는 배타적 논리합 연산을 나타낸다)

청구항 15

제10항에 있어서,

상기 마스킹을 이용한 아리아 암호 방법은, 외부로부터 입력되는 상기 암호키를 상기 마스킹 난수에 의해 마스킹하여 마스킹 암호키(masked encryption key)를 생성하는 암호키 마스킹 단계를 더 포함하고, 상기 키 스케줄링 단계는,

상기 마스킹 암호키를 이용하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하는 키 확장 단계; 및

상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 상기 마스킹 라운드키를 생성하는 마스킹 라운드키 생성 단계를 포함하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

청구항 16

제15항에 있어서,

상기 암호키 마스킹 단계에서 생성되는 상기 마스킹 암호키는, 상기 암호키의 상위 128비트를 상기 마스킹 난수를 이용하여 마스킹한 값 및 상기 암호키의 상기 상위 128비트 이외의 나머지 비트와 0으로 패딩(padding)한 비트를 이용하여 구성되는 128비트를 상기 마스킹 난수를 이용하여 마스킹한 값을 포함하는 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

청구항 17

제16항에 있어서,

상기 키 확장 단계는, 상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이스텔(Feistel) 알고리듬을 이용하면서 추가 마스킹을 수행하여 상기 마스킹 난수에 의해 마스킹된 상기 확장키를 생성하는 단계인 것을 특징으로 하는 마스킹을 이용한 아리아 암호 방법.

청구항 18

제10항 내지 제17항 중 어느 한 항에 따른 마스킹을 이용한 아리아 암호 방법을 수행하는 프로그램 코드가 기록된 기록매체로서 데이터 처리 및 연산을 수행하는 컴퓨터로 판독할 수 있는 상기 기록매체.

청구항 19

아리아(ARIA) 암호연산 수행시 중간 연산값을 마스킹하는 아리아 마스킹 방법에 있어서,

마스킹 난수를 생성하는 마스킹 난수 생성 단계;

상기 마스킹 난수와 두 종류의 에스박스(S-box; S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성하는 마스킹 에스박스 생성 단계;

상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹 단계;

외부로부터 입력되는 암호키를 마스킹하여 마스킹 암호키(masked encryption key)를 생성하고, 상기 마스킹 암호키를 이용하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하고, 그리고 상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키를 생성하는 키 스케줄링 마스킹 단계;

상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 상기 아리아 라운드 함수 연산 결과에 추가 마스킹을 수행하는 암호연산 마스킹 단계; 및

상기 암호연산 마스킹 단계에서 수행된 최종라운드 연산 결과에 대해 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력하는 디스마스킹 단계를 포함하는 아리아 마스킹 방법.

청구항 20

아리아(ARIA) 키 스케줄링 연산의 중간 연산값을 마스킹하는 아리아 키 스케줄링 마스킹 방법에 있어서,

마스킹 난수를 생성하는 마스킹 난수 생성 단계;

외부로부터 입력되는 암호키를 상기 마스킹 난수에 의해 마스킹하여 마스킹 암호키(masked encryption key)를 생성하는 암호키 마스킹 단계;

상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이스텔(Feistel) 알고리듬을 이용하면서 추가 마스킹을 수행하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하는 키 확장 단계; 및

상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round key)를 생성하는 마스킹 라운드키 생성 단계를 포함하는 아리아 키 스케줄링 마스킹 방법.

명 세 서

발명의 상세한 설명

기 술 분 야

<1> 본 발명은 아리아 마스킹 방법과 이를 이용한 아리아 암호 장치 및 방법에 관한 것이며, 더 상세하게는 차분전력분석 공격(DPA) 및 키 스케줄링 공격에 안전한 아리아 마스킹 방법과 이를 이용한 아리아 암호 장치 및 방법에 관한 것이다.

배 경 기 술

<2> 디지털 정보사회가 고도화되고 전자상거래가 활성화됨에 따라, 암호기술은 인터넷을 기반으로 한 사회·경제적 활동의 안정성과 신뢰성, 그리고 사용자 프라이버시 보호 등을 위한 핵심 기술로서 인식되고 있다. 국내에서는 1960년대부터 기관, 연구소, 학계 등에서 대칭키 암호 기술을 연구하여 TTA 표준 대칭키 알고리듬인 SEED를 개발하여 현재 인터넷 뱅킹, 위성방송 및 IMT-2000에서 기본적인 보안 매커니즘으로 활용하고 있다.

<3> 그러나, SEED 알고리듬은 효율성보다는 안전성에 중점을 두고 개발되어, 연산을 수행하는데 시간이 오래 걸리고, 하드웨어 모듈로 구현하여도 충분한 성능을 낼 수 없는 구조적 단점을 지니고 있다.

<4> 이러한 SEED 알고리듬의 단점을 극복하고 효율적인 키 스케줄을 사용하는 알고리듬이 바로 아리아(ARIA) 암호 알고리듬이다. ARIA는 전자 정부 구현 등으로 국가기관 대 민간(G2C, Government to Citizen)의 정보 교환의 필요성이 커지게 됨으로써 정보 보호를 목적으로 국가안보기술연구소(정부기관)에서 발표한 민관겸용 블록암호 알고리듬이다. ARIA는 2004년에 산업자원부의 KS인증(KS 규격번호: KS X 1213)을 획득한 블록암호 알고리듬이며, 학계(Academy), 연구소(Research Institute), 정부기관(Agency)으로 구성된 개발팀에 의해 개발되었기 때문에 그 앞 글자를 이용하여 명명되었다. ARIA는 ISPN(Involution Substitution Permutation Network) 구조를 갖는 암호화 과정 및 복호화 과정이 동일한 대칭키 암호 알고리듬으로서, AES(Advanced Encryption Standard)의 규격을 따르고, 다양한 각종 환경의 구현에 적합하도록 설계되어 초경량 환경 및 하드웨어 구현에 최적화하여 적용할 수 있으며, 고정 길이의 입출력(128비트), 가변 길이의 키(key) 지원(128/192/256 비트 키), 간단한 연산의 사용, 바이트 단위의 연산 등의 특징을 지닌다. 한편, ARIA의 입출력 크기와 사용 가능한 키 크기는 미국 표준 블록암호인 AES(Advanced Encryption Standard)의 경우와 동일하다.

<5> 한편, 수학적으로 안전한 것으로 알려진 알고리듬조차도 구현 단계에서 예기치 못한 부가적인 정보의 누출이 있다는 것이 알려졌고, 누출된 정보로부터 비밀 키의 값을 알아낼 수 있는 부채널 공격(Side Channel Attack)이 소개되었다. 이러한 부채널 공격이 소개되면서 많은 암호시스템 설계자들은 효율적인 대응법들을 연구하기 시작하였고, 부채널 공격 중 하나인 차분 전력 분석(Differential Power Analysis; DPA)에 대한 대응법으로는, 특히 블록암호의 경우 전력 측정에 의해 암·복호화, 또는 키 스케줄링의 연산 도중 중간값이 드러나지 않도록 하는 마스킹 기법(masking method)이 알려져 있다. 마스킹 기법은 블록암호의 구성에 따라 적용 방법이 달라질 수 있으며, 각각의 블록암호에 적합한 마스킹 기법에 대한 연구가 진행되고 있다.

<6> KS 인증을 획득한 ARIA의 기존 마스킹 방법들에는 네 종류의 S-box에서 4개의 마스킹 테이블을 생성하는 방법과, INV에 대한 테이블만을 저장하고 이에 대한 마스킹 테이블을 생성하여 S-box 연산을 수행하는 방법이 있다. 상기 방법들은 연산시간 측면과 공간적인 측면에서 각각 장점이 있다.

<7> 하지만, 기존의 마스킹 방법들은 ARIA에 사용되어지는 S-box의 특징 및 키 스케줄링과 암·복호화 연산이 병렬화되지 않는 ARIA 알고리듬 상의 특징을 고려하지 않았다. 즉, 기존의 마스킹 방법들은 마스킹 보정작업으로 인해 암호 연산시간이 상당히 길다는 문제점이 있으며, 또한 ARIA의 경우 다른 블록 암호들보다 키 스케줄링 공격에 취약함에도 기존의 마스킹 방법들은 키 스케줄링 과정에 마스킹 방법을 고려하지 않는 문제점이 있다.

발명의 내용

해결 하고자하는 과제

<8> 따라서, 본 발명이 이루고자 하는 첫 번째 기술적 과제는, 아리아 암호 연산시간을 감소시키고 DPA 공격 물론 키 스케줄링 공격에 대해서도 안전한 암호 장치를 제공하는 것이다.

<9> 본 발명이 이루고자 하는 두 번째 기술적 과제는, 아리아 암호 연산시간을 감소시키고 DPA 공격은 물론 키 스케줄링 공격에 대해서도 안전한 암호 방법을 제공하는 것이다.

<10> 본 발명이 이루고자 하는 세 번째 기술적 과제는, 아리아 암호 연산시간을 감소시키고 DPA 공격은 물론 키 스케

줄링 공격에 대해서도 안전한 아리아 마스킹 방법을 제공하는 것이다.

<11> 본 발명이 이루고자 하는 네 번째 기술적 과제는, 키 스케줄링 공격에 안전한 아리아 키 스케줄링 마스킹 방법을 제공하는 것이다.

과제 해결수단

<12> 상기와 같은 첫 번째 기술적 과제를 해결하기 위하여, 본 발명은 마스킹을 이용한 아리아(ARIA) 암호 장치에 있어서, 상기 장치를 제어하는 제어부로서, 마스킹 난수를 생성하고 상기 마스킹 난수와 두 종류의 에스박스(S-box; S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성하는 상기 제어부; 상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹부; 외부로부터 입력되는 암호키로부터 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성하는 키 스케줄링부; 및 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행하는 마스킹 암호연산부를 포함하는 마스킹을 이용한 아리아 암호 장치를 제공한다.

<13> 본 발명의 일 실시예에 따르면, 상기 입력데이터 마스킹부는, 상기 마스킹 암호연산부가 수행하는 첫 번째 라운드 연산의 입력 값이 되는 상기 마스킹 입력데이터를 생성할 수 있다.

<14> 본 발명의 일 실시예에 따르면, 상기 마스킹 암호연산부는, 상기 네 종류의 마스킹 에스박스에 의해 아리아 라운드 함수의 치환 계층(substitution layer)을 구성하여 상기 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 추가 마스킹을 수행할 수 있다.

<15> 본 발명의 일 실시예에 따르면, 상기 마스킹을 이용한 아리아 암호 장치는, 상기 마스킹 암호연산부가 수행한 최종라운드 연산 결과에 대해 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력하는 디스마스킹부를 더 포함할 수 있다.

<16> 본 발명의 일 실시예에 따르면, 상기 제어부가 생성하는 상기 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)는, 하기 수학식;

$$MS_1(x \oplus m) = S_1(x) \oplus m',$$

$$MS_2(x \oplus m) = S_2(x) \oplus m',$$

$$MS_3(x \oplus m') = S_1^{-1}(x) \oplus m,$$

<17> $$MS_4(x \oplus m') = S_2^{-1}(x) \oplus m$$

<18> 을 만족할 수 있다(상기 수학식에서, S_1 및 S_2는 상기 두 종류의 에스박스, S_1^{-1} 및 S_2^{-1}는 S_1 및 S_2에 대한 역치환 에스박스, m 및 m'는 상기 제어부가 생성한 제1 마스킹 난수 및 제2 마스킹 난수, 그리고 $\oplus$는 배타적 논리합 연산을 나타낸다).

<19> 본 발명의 일 실시예에 따르면, 상기 마스킹을 이용한 아리아 암호 장치는, 외부로부터 입력되는 상기 암호키를 상기 마스킹 난수에 의해 마스킹하여 마스킹 암호키(masked encryption key)를 생성하는 암호키 마스킹부를 더 포함하고, 상기 키 스케줄링부는, 상기 마스킹 암호키를 이용하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하는 키 확장부; 및 상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 상기 마스킹 라운드키를 생성하는 마스킹 라운드키 생성부를 포함할 수 있다.

<20> 본 발명의 일 실시예에 따르면, 상기 암호키 마스킹부가 생성한 상기 마스킹 암호키는, 상기 암호키의 상위 128비트를 상기 마스킹 난수를 이용하여 마스킹한 값 및 상기 암호키의 상기 상위 128비트 이외의 나머지 비트와 0으로 패딩(padding)한 비트를 이용하여 구성되는 128비트를 상기 마스킹 난수를 이용하여 마스킹한 값을 포함할 수 있다.

<21> 본 발명의 일 실시예에 따르면, 상기 키 확장부는, 상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이

스텔(Feistel) 알고리듬을 이용하면서 추가 마스킹을 수행하여 상기 마스킹 난수에 의해 마스킹된 상기 확장키를 생성할 수 있다.

<22> 본 발명의 일 실시예에 따르면, 상기 마스킹을 이용한 아리아 암호 장치는 시스템 온 칩(system on chip) 형태로 마이크로프로세서상에 구현될 수 있다.

<23> 상기와 같은 두 번째 기술적 과제를 해결하기 위하여, 본 발명은 마스킹을 이용한 아리아 암호 방법에 있어서, 마스킹 난수를 생성하는 마스킹 난수 생성 단계; 상기 마스킹 난수와 두 종류의 에스박스(S-box; S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성하는 마스킹 에스박스 생성 단계; 상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹 단계; 외부로부터 입력되는 암호키로부터 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성하는 키 스케줄링 단계; 및 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행하는 마스킹 암호연산 단계를 포함하는 마스킹을 이용한 아리아 암호 방법을 제공한다.

<24> 본 발명의 일 실시예에 따르면, 상기 마스킹을 이용한 아리아 암호 방법은, 데이터 처리 및 연산을 수행하는 컴퓨터로 판독할 수 있는 기록매체에 상기 컴퓨터가 판독할 수 있는 프로그램 코드로 구현할 수 있다.

<25> 상기와 같은 세 번째 기술적 과제를 해결하기 위하여, 본 발명은 아리아 마스킹 방법에 있어서, 마스킹 난수를 생성하는 마스킹 난수 생성 단계; 상기 마스킹 난수와 두 종류의 에스박스(S-box; S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성하는 마스킹 에스박스 생성 단계; 상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성하는 입력데이터 마스킹 단계; 외부로부터 입력되는 암호키를 마스킹하여 마스킹 암호키(masked encryption key)를 생성하고, 상기 마스킹 암호키를 이용하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하고, 그리고 상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키를 생성하는 키 스케줄링 마스킹 단계; 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 상기 아리아 라운드 함수 연산 결과에 추가 마스킹을 수행하는 암호연산 마스킹 단계; 및 상기 암호연산 마스킹 단계에서 수행된 최종라운드 연산 결과에 대해 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력하는 디스마스킹 단계를 포함하는 아리아 마스킹 방법을 제공한다.

<26> 상기와 같은 네 번째 기술적 과제를 해결하기 위하여, 본 발명은 아리아 키 스케줄링 마스킹 방법에 있어서, 마스킹 난수를 생성하는 마스킹 난수 생성 단계; 외부로부터 입력되는 암호키를 상기 마스킹 난수에 의해 마스킹하여 마스킹 암호키(masked encryption key)를 생성하는 암호키 마스킹 단계; 상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이스텔(Feistel) 알고리듬을 이용하면서 추가 마스킹을 수행하여 상기 마스킹 난수에 의해 마스킹된 확장키를 생성하는 키 확장 단계; 및 상기 확장키를 이용하여 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round key)를 생성하는 마스킹 라운드키 생성 단계를 포함하는 아리아 키 스케줄링 마스킹 방법을 제공한다.

효 과

<27> 본 발명에 따른 아리아 마스킹 방법과 이를 이용한 아리아 암호 장치 및 방법은 기존의 아리아 마스킹 방법에 비해 DPA 공격에 대한 안전성은 물론 키 스케줄링 공격에 대한 안전성도 제공하며, 하드웨어 자원 및 암호 연산 시간을 감소시킨다는 이점을 제공한다.

발명의 실시를 위한 구체적인 내용

<28> 본 발명에 관한 구체적인 내용의 설명에 앞서 이해의 편의를 위해 본 발명이 해결하고자하는 기술적 과제의 개요를 우선 제시한다.

<29> 도 1에는 아리아(ARIA)의 전체 구조가 도시되어 있다. 도 1에 도시된 바와 같이 아리아는 키 에디션(Key Addition; 110), 치환 계층(Substitution Layer; 120), 및 확산 계층(Diffusion Layer; 130)으로 라운드 함수를 구성하며, 세부 사양은 하기 표 1과 같다.

표 1

기본 구조	ISPN(Involution Substitution Permutation Network) 구조		
입 출력크기	128 비트		
키 크기	128 비트	192 비트	256 비트
라운드 키 크기	128 비트		
라운드 수	12 라운드	14 라운드	16 라운드

<30>

<31> **치환 계층**

<32> 각 라운드 함수의 치환 계층은 8 비트 입출력을 가지는 두 종류의 S-Box(S_1, S_2)와 그에 대한 두 개의 역 치환 S-Box(S_1^{-1}, S_2^{-1})를 포함하여 총 네 종류의 S-box를 이용한다. S-box(S_1, S_2)는 x^{-1}, x^{247}($=x^{-8}=Dx^{-1}$, D는 x^8을 수행하는 행렬)의 아핀 변환(affine transform) 형태로 하기 수학식 1과 같이 연산을 수행한다.

수학식 1

<33> $S_1, S_2 : GF(2^8) \rightarrow GF(2^8)$ 일 때,

<34> $$S_1(x) = Bx^{-1} \oplus b,$$

<35> $$S_2(x) = CDx^{-1} \oplus c$$

<36> 상기 수학식 1에서 각각의 행렬 및 벡터 값은 하기 수학식 2와 같다.

수학식 2

<37>
$$B = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad CD = \begin{pmatrix} 0&1&0&1&0&1&1&1 \\ 0&0&1&1&1&1&1&1 \\ 1&1&1&0&1&1&0&1 \\ 1&1&0&0&0&0&1&1 \\ 0&1&0&0&0&0&1&1 \\ 1&1&0&0&1&1&1&0 \\ 0&1&1&0&0&0&1&1 \\ 1&1&1&1&0&1&1&0 \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

<38> 상기 두 개의 S-Box S_1, S_2와 그 역치환 S_1^{-1}, S_2^{-1}를 이용하여 아리아의 라운드 함수 중 치환 계층을 구성한다. 이때, 홀수 라운드 함수(F_o)의 치환 계층(120)과 짝수 라운드 함수(F_e)의 치환 계층(140)은 도 1에서 도시된 바와 같이 S_1, S_2와 S_1^{-1}, S_2^{-1}의 순서를 반대로 구성한다.

<39> **확산 계층**

<40> 아리아 라운드 함수의 확산 계층(130, 150)은 입력 16 바이트($x_0, x_1, \cdots, x_{15}$)에 대한 출력 16바이트($y_0, y_1, \cdots, y_{15}$)의 값을 하기 수학식 3과 같이 행렬 곱셈을 통하여 연산한다.

수학식 3

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix}$$

<41>

<42> 즉, 각 행마다 치환 계층을 거친 16 바이트 중 7 바이트의 xor형태로 연산되어진다.

<43> **<u>키 스케줄링</u>**

<44> 도 2에는 아리아의 키 스케줄링이 도시되어 있다. 도 2를 참조하면 KL(200)은 암호키의 상위 128비트 값이며, KR(202)은 키의 나머지 부분에 0으로 패딩(padding)한 128비트 값이다. 세 개의 키 초기화 상수 CK_1, CK_2, CK_3는 각 라운드 함수(F_o, F_e, F_o)의 키 역할을 수행한다. 즉, 3 라운드 256 비트 Feistel 알고리듬을 이용하는 키 확장 모듈(210)로부터 키 확장을 위한 네 종류의 128비트 확장키(W_0, W_1, W_3, W_4)를 생성한다. 그리고 라운드키 생성 모듈(220)에서 하기 수학식 4와 같이 라운드 키(*ek*)를 얻는다.

수학식 4

$$\begin{aligned} ek_1 &= W_0 \oplus (W_1 \gg 19), & ek_2 &= W_1 \oplus (W_2 \gg 19), \\ ek_3 &= W_2 \oplus (W_3 \gg 19), & ek_4 &= (W_0 \gg 19) \oplus W_3, \\ ek_5 &= W_0 \oplus (W_1 \gg 31), & ek_6 &= W_1 \oplus (W_2 \gg 31), \\ ek_7 &= W_2 \oplus (W_3 \gg 31), & ek_8 &= (W_0 \gg 31) \oplus W_3, \\ ek_9 &= W_0 \oplus (W_1 \ll 61), & ek_{10} &= W_1 \oplus (W_2 \ll 61), \\ ek_{11} &= W_2 \oplus (W_3 \ll 61), & ek_{12} &= (W_0 \ll 61) \oplus W_3, \\ ek_{13} &= W_0 \oplus (W_1 \ll 31), & ek_{14} &= W_1 \oplus (W_2 \ll 31), \\ ek_{15} &= W_2 \oplus (W_3 \ll 31), & ek_{16} &= (W_0 \ll 31) \oplus W_3, \\ ek_{17} &= W_0 \oplus (W_1 \ll 19) \end{aligned}$$

<45>

<46> 상기 수학식 4에서, $\oplus$는 배타적 논리합 연산을 나타내고, $\gg n$는 오른쪽으로 n비트 로테이션(rotation), $\ll n$는 왼쪽으로 n비트 로테이션을 나타낸다.

<47> 유의할 점은, 아리아의 키 스케줄링은 알고리듬의 특성상 암·복호화 연산과 병렬화되어 수행될 수 없다는 것이다.

<48> 차분 전력 분석(Differential Power Analysis; DPA) 공격이 소개되면서 여러 가지 대응법들이 소개되어왔다. 기존의 대응법들 중에서 암호 알고리듬 연산 수행 중 연산 중간값의 정보를 숨기는 일반적인 마스킹 기법이 알려져 있다.

<49> 아리아에 일반적인 마스킹 기법을 적용한 기존 제1 방법(유형소, 하재철, 김창균, 박일환, 문상재, “랜덤 마스킹 기법을 이용한 DPA 공격에 안전한 ARIA 구현”, 한국정보보호학회논문지 16(2), April 2006. 참조)은, 아리아의 네 종류의 S-box에 대하여 네 종류의 새로운 마스킹 S-box를 만든 후, 상기 네 종류의 새로운 마스킹 S-box를 이용하여 암호연산을 수행하는 것이다. 그러나, 상기 기존 제1 방법은 각 라운드의 확산 계층에 대한 연산을 마칠 때마다 원하는 마스킹 값(마스킹 S-box 생성 시 이용한 마스킹 값)을 얻기 위해 반드시 마스킹 보정 작업을 수행해야 하므로 보정을 위한 연산이 많다는 문제점이 있다. 더욱이, 총 4개의 S-box를 저장해야 하고 암호연산시마다 새로운 4개의 마스킹 S-box를 생성해야 하므로 총 8개의 S-box(2048 바이트)를 저장하기 위해 비교적 많은 저장 공간을 필요로 한다는 문제점이 있다.

<50> 상기 기존 제1 방법이 많은 저장 공간을 필요로 한다는 문제점을 보완하기 위해 저 메모리 환경 마스킹 기법을 적용한 기존 제2 방법(유형소, 하재철, 김창균, 박일환, 문상재, “저메모리 환경에 적합한 마스킹 기반의 ARIA 구현”, 한국정보보호학회논문지 16(3), June 2006. 참조)은, x^{-1}($x \in GF(2^8)$)에 대한 256 바이트의 테이블(INV)만을 저장한다. 그리고 암호연산시마다 역원에 대한 마스킹 테이블(MINV)을 구성한다. 즉, S-box의 연산 $S_2(x)=CDx^{-1}\oplus c$에서 x^{-1}의 연산은 테이블을 이용하며, 아핀 변환(affine transform)을 직접 계산하는 방법이다. 이러한 계산을 이용한 두 에스박스에 대한 출력 값의 마스킹 값은 어렵지 않게 알 수 있지만, 상기 기존 제2 방법은 마스킹 값이 일정하지 않기 때문에 확산 계층 연산을 수행하기 전 마스킹 보정 작업을 수행하도록 구성되어 있으며, S_1^{-1}, S_2^{-1}에 대한 계산에서도 MINV의 테이블 참조 전에 마스킹 보정작업을 수행하도록 구성되어 있다. 즉, 총 필요한 테이블의 개수는 두 개인 반면 치환 계층의 연산과 마스킹 보정 작업으로 인한 연산 시간이 크다는 문제점이 있다.

<51> 또한, 상기 기존 제1 방법 및 기존 제2 방법은, 키 스케줄링에 대한 마스킹 방법을 고려하지 않기 때문에 키가 노출될 수 있는 문제점이 있다(S. Mangard, "A Simple Power-Analysis(SPA) Attack on Implementations of the AES Key Expansion", ICISC 2002, LNCS 2587, pp. 343-358, Springer, 2003. 참조). 키 스케줄링 공격에 대한 대응법을 고려하지 않은 경우 키 스케줄링 과정에서 키에 대한 해밍웨이트가 노출되어 후보 키의 범위가 좁아지며, 그 결과 전수조사를 통해 AES의 전체 키를 찾을 수 있게 된다. 특히, 아리아의 경우 알고리듬의 특성상 AES와는 달리 키 스케줄링과 암호연산이 병렬적으로 수행될 수 없다. 즉, 키 스케줄링 후에 암호연산을 수행해야 한다. 그 결과, 아리아는 키 스케줄링에 대한 부분의 전력이 암호연산과 상관없이 그대로 노출되므로 AES의 경우보다 키 스케줄링 공격에 훨씬 취약하다.

<52> 이러한 일차적인 문제 외에도 키가 마스킹되어 있지 않으면, 해밍 디스턴스 모델(hamming distance model)의 경우 키 에디션 부분의 연산에서 키의 해밍웨이트를 알아내는 것이 가능하다. 그 이유는 키 에디션 전후 해밍 디스턴스가 하기 수학식 5를 만족하기 때문이다.

수학식 5

<53> $$(x \oplus m) \oplus (x \oplus k \oplus m) = x \oplus (x \oplus k)$$

<54> 즉, 해밍 디스턴스 모델에서 키 에디션 부분의 전력이 마스킹 값과 상관없이 키에만 의존하기 때문에 키의 해밍웨이트가 노출될 수 있다. 따라서, 키 스케줄링에서의 효율적인 마스킹 방법도 고려할 필요가 있다.

<55> 상기 기존 방법들과 비교하면, 본 발명은 두 종류의 에스박스만을 저장한 후 키 스케줄링 및 암호연산시마다 네 종류의 마스킹 에스박스(masked S-box)를 생성하여 사용하는 것이다. 즉, 본 발명은 본래의 아리아 알고리듬보다 두 개의 에스박스만을 더 필요로 한다. 또한, 본 발명은 아리아의 키 스케줄링에 대한 마스킹을 고려한 것이다.

<56> 이하, 본 발명의 기술적 과제의 해결 방안을 명확화하기 위해 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 상세하게 설명한다. 다만, 본 발명을 설명함에 있어서 관련된 공지기술 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불명료하게 할 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 또한, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자 등의 의도 또는 관례 등에 따라 달라질 수 있을 것이다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

<57> 도 3에는 본 발명에 따른 마스킹을 이용한 아리아 암호 장치의 일례가 블록도로 도시되어 있다.

<58> 도 3을 참조하면, 본 발명에 따른 마스킹을 이용한 아리아 암호 장치는 제어부(310), 암호키 마스킹부(320), 키

스케줄링부(330), 입력데이터 마스킹부(340), 마스킹 암호연산부(350), 및 디스마스킹부(360)를 포함한다. 또한, 본 발명에 따른 아리아 암호 장치는 입력 레지스터(300) 및 암호키 레지스터(302)를 더 포함할 수 있다. 상기 입력 레지스터(300)는 평문 입력데이터를 저장하고 상기 암호키 레지스터(302)는 암호키 데이터를 저장한다.

<59> 상기 제어부(310)는 상기 아리아 암호 장치를 제어하며 마스킹 난수를 생성하고, 상기 마스킹 난수 및 두 종류의 비선형 함수인 제1 에스박스(S_1) 및 제2 에스박스(S_2)를 이용하여 네 종류의 마스킹 에스박스(masked S-box; MS_1, MS_2, MS_3, MS_4)를 생성한다. 더 상세히 설명하면, 상기 제어부(310)는 두 개의 마스킹 값(m, m')을 생성하여 하기 수학식 6을 만족하는 네 종류의 마스킹 에스박스를 생성한다.

수학식 6

$$MS_1(x \oplus m) = S_1(x) \oplus m',$$

$$MS_2(x \oplus m) = S_2(x) \oplus m',$$

$$MS_3(x \oplus m') = S_1^{-1}(x) \oplus m,$$

<60> $$MS_4(x \oplus m') = S_2^{-1}(x) \oplus m$$

<61> 상기 제어부(310)는 다음의 '마스킹 에스박스 생성 알고리듬'에 의해 상기 네 종류의 마스킹 에스박스를 생성한다.

<62> **마스킹 에스박스 생성 알고리듬**

<63> **입력** S_1, S_2, m, m'

<64> **출력** MS_1, MS_2, MS_3, MS_4

<65> 1. For x from 0 to 255

<66> a. $u = x \oplus m$, $v = S_1(x) \oplus m$, $w = S_2(x) \oplus m'$

<67> b. $MS_1(u)=v$, $MS_2(u)=w$, $MS_3(v)=u$, $MS_4(w)=u$

<68> 2. Return MS_1, MS_2, MS_3, MS_4

<69> 상기 알고리듬에 따라 마스킹 에스박스를 구성할 때, 상기 수학식 6에서 MS_1 및 MS_2에 관한 앞의 두 식이 성립됨을 쉽게 할 수 있다. MS_3 및 MS_4에 관한 뒤의 두 식이 성립 여부는 다음과 같이 증명할 수 있다.

<70> **증명**

<71> 1. S_1은 치환이므로 모든 x에 대해 $x=S_1^{-1}(t)$를 만족하는 t가 존재한다.

<72> 2. 따라서, 수학식 $MS_3(S_1(x) \oplus m') = x \oplus m$에서 x에 $S_1^{-1}(t)$를 대입하면 $MS_3(t \oplus m') = S_1^{-1}(t) \oplus m$가 성립한다.

<73> 3. 결론적으로, $MS_3(S_1(x) \oplus m') = x \oplus m$이면, $MS_3(x \oplus m') = S_1^{-1}(x) \oplus m$가 성립한다. MS_4에 대해서도 동일하게 증명할 수 있다.

<74> 상기 입력데이터 마스킹부(340)는, 평문 입력데이터를 상기 제어부가 생성한 두 개의 1바이트(byte) 마스킹 난수에 의해 마스킹하여 마스킹 입력데이터(masked input data)를 생성한다. 즉, 상기 입력데이터 마스킹부(340)는 상기 마스킹 암호연산부(350)가 수행하는 첫 번째 라운드 연산의 입력 값이 되는 상기 마스킹 입력데이터를 생성한다. 예컨대, 제1 마스킹 난수(m) 및 제2 마스킹 난수(m')를 이용하여 구성되는 16바이트 마스킹 값 m'm'mm/m'm'mm/m'm'mm/m'm'mm으로 16바이트(128비트)의 입력 평문을 마스킹한다.

<75> 상기 키 스케줄링부(330)는, 외부로부터 입력되는 암호키(key)로부터 상기 제어부(310)가 생성한 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성한다. 예컨대, 아리아의 키 스케줄링 과정에서 상기 제1 마스킹 난수(m) 및 상기 제2 마스킹 난수(m')의 배타적 논리합 연산에 의해 계산되는 제3 마스킹 난수(m")를 이용하여 16바이트의 모든 라운드키가 마스킹되도록 조정한다. 즉, m"로 이루어진 16바이트의 마스킹값 $(m")^{16}$으로 마스킹된 마스킹 라운드키를 생성한다. 상기 키 스케줄링부(330)의 구체적인 동작 및 아리아 키 스케줄링의 마스킹 방법은 나중에 다시 설명한다.

<76> 상기 마스킹 암호연산부(350)는, 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행한다. 예컨대, 상기 마스킹 암호연산부(350)는 상기 제어부(310)가 생성한 상기 네 종류의 마스킹 에스박스에 의해 아리아 라운드 함수의 치환 계층(substitution layer)을 구성하여 상기 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 추가 마스킹을 수행한다.

<77> 본 발명의 일 실시예에 따르면, 상기 마스킹 암호연산부(350)는 상기 동작을 구현하기 위해 멀티플렉서(352), 홀수라운드 연산부(354), 짝수라운드 연산부(356), 및 최종라운드 연산부(358)을 포함할 수 있다.

<78> 상기 멀티플렉서(352)는 상기 홀수라운드 연산부(354)가 첫 번째 라운드 연산을 수행하는 경우에는 상기 입력데이터 마스킹부(340)가 생성한 상기 마스킹 입력데이터를 상기 홀수라운드 연산부(354)에 전달하고, 상기 홀수라운드 연산부(354)가 다음 홀수라운드 함수 연산을 수행하는 경우에는 상기 짝수라운드 연산부(356)의 연산 결과를 상기 홀수라운드 연산부(354)에 전달한다. 상기 멀티플렉서(352)를 제어하는 제어신호는 상기 제어부(310)에서 생성될 수 있다.

<79> 도4 내지 도 6에는 상기 마스킹 암호연산부(350)가 포함하는 각각의 라운드 연산부의 일례가 블록도로 도시되어 있다.

<80> 도 4를 참조하면, 상기 홀수라운드 연산부(354)는 키 에디션부(400), 치환부(410), 확산부(420), 및 추가 마스킹부(530)를 포함할 수 있다. 상기 홀수라운드 연산부(354)의 홀수라운드 함수 연산은 키 에디션(key addition)을 수행하는 키 에디션부(400), 상기 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)에 의해 아리아 라운드 함수의 치환 계층을 구성하는 치환부(410), 및 아리아 라운드 함수의 확산 계층 연산을 수행하는 확산부(420)에 의해 수행된다.

<81> 예컨대, 첫 번째 라운드인 경우, 16바이트의 마스킹 값 $(m'm'mm)^4$로 마스킹된 상기 마스킹 입력데이터가 입력되면 상기 키 에디션부(400)는 마스킹 값 $(m")^{16}$로 마스킹된 상기 마스킹 라운드키에 의해 키 에디션을 수행하여 중간 연산값이 상기 치환부(410)의 치환 계층에 적절한 입력 값이 되도록 중간 마스킹 값을 $(mmm'm')^4$로 조정한다. 상기 치환부(410)의 치환 계층을 거친 후 중간 연산값은 중간 마스킹 값 $(m'm'mm)^4$로 마스킹된다. 상기 확산부(420)의 확산 계층 연산에서 7바이트의 값이 XOR되면, 상기 확산 계층 연산 후 중간 연산값은 중간 마스킹 값 $(mmm'm'/mmm'm')^2$으로 마스킹된다. 이때, XOR되는 상기 7바이트의 값 중에서 4바이트의 값은 m'(또는 m)으로, 나머지 3바이트의 값은 m(또는 m')으로 마스킹된 값이다. 유의할 점은, 상기 7 바이트의 값을 XOR할 때 중간 연산값이 드러나지 않도록 하기 위해, 상기 7바이트의 값의 XOR 순서를 조정하여 마스킹 값이 번갈아 가면서 XOR되도록 구현하거나, 또는 상기 7바이트 값의 XOR연산 도중 중간 연산값이 데이터 버스를 지나지 않도록 구현하여야 한다는 것이다.

<82> 상기 홀수라운드 연산부(354)는 상기 홀수라운드 함수 연산 결과가 상기 짝수라운드 연산부(356)에 의한 다음 라운드 연산의 입력 값이 되도록 추가 마스킹을 수행한다. 상기 동작을 수행하기 위해 상기 홀수라운드 연산부(354)는 상기 추가 마스킹부(430)를 포함할 수 있다. 예컨대, 상기 추가 마스킹부(430)는, 상기 확산부(420)의 확산 계층 연산 후 중간 마스킹값 $(mmm'm'/mmm'm')^2$으로 마스킹된 상기 중간 연산값이 다음 라운드에서 $(m")^{16}$으로 마스킹된 마스킹 라운드키와 XOR된 후 짝수라운드 함수의 치환 계층에 입력될 수 있는 중간 연산값이 되도록 추가적인 마스킹을 수행한다. 즉, $(mmm'm'/mmm'm')^2$으로 마스킹된 상기 중간 연산값에 $(0000/m"m"m"m")^2$를 XOR한다. 주목할 점은, 일반 아리아 알고리듬과 비교할 때 이러한 추가적인 마스킹은 본 발명에서 유일하게 요구되는 추가적인 연산이지만 그 연산량은 상당히 적다는 것이다.

<83> 도 5를 참조하면, 상기 짝수라운드 연산부(356)는 짝수라운드 함수 연산을 수행하는 키 에디션부(500), 치환부

(510), 및 확산부(520)와 추가적인 마스킹을 수행하는 추가 마스킹부(530)를 포함할 수 있다. 도 5에 도시된 바와 같이, 상기 짝수라운드 연산부(356)의 구체적인 동작 및 작용은 상기 홀수라운드 연산부(354)와 쌍대적으로 설명될 수 있다. 다만, 다음 라운드가 최종라운드인 경우에는 상기 추가 마스킹부(530)는 상기 짝수라운드 함수 연산 결과가 상기 최종라운드 연산부(358)에 입력될 수 있는 중간 연산값이 되도록 $(m"m"m"m"/0000)^2$로 추가 마스킹을 수행한다. 그리고 그 결과값을 상기 최종라운드 연산부(358)에 전달한다.

<84> 도 6을 참조하면, 상기 최종라운드 연산부(358)는 키 에디션부(600, 620) 및 치환부(610)를 포함할 수 있다. 상기 최종라운드 연산부(358)의 최종라운드 함수 연산은 키 에디션을 수행하는 키 에디션부(600, 620) 및 상기 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)에 의해 아리아 라운드 함수의 치환 계층을 구성하는 치환부(410)에 의해 수행된다. 아리아의 최종라운드 함수는 확산 계층 연산을 포함하지 않기 때문에 상기 최종라운드 연산부(358)는 다른 라운드 연산부(354, 356)와는 달리 확산부를 포함하지 않는다.

<85> 본 발명의 일 실시예에 따르면, 상기 마스킹 암호연산부(350)는 상기 최종라운드 연산부(358)를 별도로 구성하지 않고 상기 짝수라운드 연산부(356)의 키 에디션부(500) 및 치환부(510)를 이용하여 구현할 수 있다. 예컨대, 짝수라운드의 다음 라운드가 최종라운드인 경우 상기 짝수라운드 연산부(356)의 출력이 다시 상기 짝수라운드 연산부(356)에 입력되고, 상기 키 에디션부(500)에서 마스킹 라운드키의 에디션을 수행하고, 그리고 상기 치환부(510)의 치환 계층을 거친 후 최종적인 키 에디션부(620)만을 추가하여 마지막 마스킹 라운드키의 키 에디션을 수행하도록 구현할 수 있다. 아리아의 최종라운드 함수는 짝수라운드 함수와 동일한 치환 계층을 사용하기 때문이다.

<86> 도 3에서는 상기 마스킹 암호연산부(350)를 루프 회로로 구현하였지만, 당업자라면 다양한 변형 실시가 가능할 것이다.

<87> 상기 디스마스킹부(360)부는 상기 마스킹 암호연산부(350)가 수행한 최종라운드 연산 결과에 대해 마스킹 값을 벗겨내는 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력한다. 예컨대, 상기 최종라운드 연산부(358)가 연산 결과를 출력하면 상기 디스마스킹부(360)는 상기 연산 결과에 16바이트 마스킹 값 $(m'm'mm)^4$를 XOR하여 디스마스킹을 수행한다.

<88> 이하, 상기 키 스케줄링부(330)의 구체적인 동작 및 아리아 키 스케줄링의 마스킹 방법을 상세히 설명한다.

<89> 아리아의 키 스케줄링에 대한 공격은 DPA(Differential Power Analysis)를 고려하지 않는다. 그 이유는 아리아의 키 스케줄링은 고정된 키에 대해 항상 같은 연산을 수행하기 때문에 랜덤한 평문을 다수 입력해 파형을 얻는 DPA방법은 키 스케줄링 과정에서 의미가 없기 때문이다. 따라서 키 스케줄링 공격에 대한 대응법은 키의 해밍웨이트를 숨기는 마스킹 기법으로 충분한 대응법이 된다.

<90> 상기 암호키 마스킹부(320)부는, 외부로부터 입력받은 암호키(key)를 상기 제어부(310)가 생성한 마스킹 난수에 의해 마스킹하여 마스킹 암호키(masked encryption key)를 생성한다.

<91> 상기 키 스케줄링부(330)는, 상기 마스킹 암호키를 이용하여 상기 제어부(310)가 생성한 마스킹 난수에 의해 마스킹된 상기 마스킹 라운드키(masked round-key)를 생성한다. 상기 키 스케줄링부(330)는 상기 동작을 구현하기 위해 키 확장부(332) 및 마스킹 라운드키 생성부(334)를 포함할 수 있다.

<92> 도 7에는 상기 암호키 마스킹부(320) 및 상기 키 스케줄링부(330)의 상기 키 확장부(332)의 일례가 구체적으로 도시되어 있다.

<93> 도 7을 참조하면, 상기 암호키 마스킹부(320)는 상기 암호키의 상위 128비트(KL)를 $(m")^{16}$로 마스킹한 값 및 상기 암호키의 상기 상위 128비트 이외의 나머지 비트와 0으로 패딩(padding)한 비트를 이용하여 구성되는 128비트(KR)를 $(m'm'mmmmm'm')^2$로 마스킹한 값을 포함하는 마스킹 암호키를 생성한다.

<94> 상기 키 스케줄링부(330)의 상기 키 확장부(332)는 상기 마스킹 암호키를 이용하여 상기 제어부(310) 생성한 마스킹 난수에 의해 마스킹된 확장키(MS_1, MS_2, MS_3, MS_4)를 생성한다. 예컨대, 도 7에 도시된 바와 같이 상기 키 확장부(332)는 상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이스텔(Feistel) 알고리듬을 이용하면서 추가 마스킹을 수행하여 마스킹 값 $(m")^{16}$으로 마스킹된 네 개의 16바이트 확장키(MS_1, MS_2, MS_3, MS_4)를 생성한다. 도 7에서 F_o 및 F_e는 각각 홀수라운드 함수 및 짝수라운드 함수 블록을 나타내며, CK1, CK2, CK3는 각각 라

운드 함수 F_o, F_e, F_o의 키 역할을 수행하는 고정된 상수를 나타낸다. 기존의 3라운드 256비트 파이스텔 알고리듬에 추가된 배타적 논리합 연산부분($\oplus$)들은 상기 16바이트의 확장키들(MS_1, MS_2, MS_3, MS_4)을 모두 $(m'')^{16}$으로 마스킹된 값으로 얻기 위해 추가적으로 필요한 연산들이다.

<95> 상기 마스킹 라운드키 생성부(334)는, 상기 확장키를 이용하여 상기 제어부(310)가 생성한 마스킹 난수에 의해 마스킹된 상기 마스킹 라운드키를 생성한다. 예컨대, 상기 마스킹 라운드키 생성부(334)는 상기 네 개의 확장키(MS_1, MS_2, MS_3, MS_4)를 이용하여 상기 수학식 4의 일반 아리아 알고리듬의 첫 번째 라운드키(ek_1)에 대응하는 첫 번째 마스킹 라운드키(ek_1')는 하기 수학식 7과 같이 생성할 수 있다.

수학식 7

<96> $$ek_1' = W_0 \oplus (m'' \gg 3)^{16} \oplus (W_1 \gg 19)$$

<97> 상기 수학식 7에서, $m'' \gg 19 = m'' \gg 3$이기 때문에 상기 첫 번째 마스킹 라운드키(ek_1')를 $(m'')^{16}$로 마스킹 된 값으로 생성하게 된다. 상기 마스킹 라운드키 생성부(334)는 유사한 방식에 의해 다른 모든 마스킹 라운드키를 상기 수학식 4에 대응하여 $(m'')^{16}$로 마스킹 된 값으로 생성할 수 있다.

<98> 일 실시예에 따르면, 본 발명에 따른 마스킹을 이용한 아리아 암호 장치는 고차 DPA 공격에 충분히 대응할 수 있도록 첫 번째 라운드 및 최종라운드의 마스킹 에스박스의 입·출력 바이트마다 마스킹 값을 다르게 구현할 수 있다.

<99> 일 실시예에 따르면, 본 발명에 따른 마스킹을 이용한 아리아 암호 장치는 고차 DPA 공격에 충분히 대응할 수 있도록 S. Mangard가 제안한 방법(C. Herbst, E. Oswald, S. Mangard, "An AES Smart Card Implementation Resistant to Power Analysis Attacks," ACNS 2006, LNCS 3989, pp. 239-252, Springer, 2006. 참조)을 적용하여 동일한 마스킹 값을 사용하면서 첫 번째 라운드와 최종라운드의 키 에디션, 치환 계층, 확산 계층의 순서(각각의 16 byte 연산에 대한 순서)를 랜덤하게 바꾸도록 구현할 수 있다.

<100> 일 실시예에 따르면, 본 발명에 따른 마스킹을 이용한 아리아 암호 장치는 마이크로프로세서(microprocessor)상에 시스템 온 칩(system on chip) 형태로 구현될 수 있다.

<101> 도 8에는 본 발명에 따른 마스킹을 이용한 아리아 암호 방법의 일례가 흐름도로 도시되어 있다.

<102> 도 8을 참조하면, 우선 상기 제어부(310)에서는 마스킹 난수를 생성하고, 상기 마스킹 난수와 두 종류의 에스박스(S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)를 생성한다(S800). 예컨대, 제1 마스킹 난수(m) 및 제2 마스킹 난수(m')를 생성하고, m, m' 및 S_1, S_2를 이용하여 상기 마스킹 에스박스 생성 알고리듬에 따라 네 종류의 마스킹 에스박스를 생성한다.

<103> 상기 입력데이터 마스킹부(340)에서는 상기 마스킹 난수에 의해 입력데이터를 마스킹하여 마스킹 입력데이터(masked input data)를 생성한다(S810). 예컨대, 상기 m 및 m'로 이루어진 마스킹 값으로 입력데이터를 마스킹하여 마스킹 입력데이터를 생성한다.

<104> 상기 키 스케줄링부(330)에서는 외부로부터 입력되는 암호키로부터 상기 마스킹 난수에 의해 마스킹된 마스킹 라운드키(masked round-key)를 생성한다(S820). 예컨대, 상기 암호키 마스킹부(320)에서 상기 암호키를 마스킹하여 마스킹 암호키를 생성하고, 상기 키 확장부(332)에서 상기 마스킹 암호키로부터 마스킹 값 $(m'')^{16}$로 마스킹된 확장키(MS_1, MS_2, MS_3, MS_4)를 생성하고, 그리고 상기 마스킹 라운드키 생성부(334)에서 상기 확장키(MS_1, MS_2, MS_3, MS_4)를 이용하여 마스킹 값 $(m'')^{16}$로 마스킹된 상기 마스킹 라운드키를 생성한다.

<105> 상기 마스킹 암호연산부(350)에서는 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 상기 마스킹 입력데이터에 대해 각각의 라운드 연산을 수행한다(S830). 예컨대, 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과에 추가 마스킹을 수행하여 상기 각각의 라운드 연산 결과가 다음 라운드 연산의 입력 값이 되도록 한다.

<106> 상기 디스마스킹부(360)에서는 상기 마스킹 암호연산부(350)에서 수행된 최종라운드 연산 결과에 대해 디스마스

킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력한다(S840).

<107> 본 발명에 따른 마스킹을 이용한 아리아 암호 방법의 세부과정은 상기 본 발명에 따른 마스킹을 이용한 아리아 암호 장치에 관한 상세한 설명을 참조한다.

<108> 도 9에는 본 발명에 따른 아리아 마스킹 방법의 일례가 흐름도로 도시되어 있다.

<109> 도 9를 참조하면, 우선 두 개의 마스킹 난수(m, m')를 생성하고 m, m'과 두 종류의 에스박스(S_1, S_2)를 이용하여 네 종류의 마스킹 에스박스(MS_1, MS_2, MS_3, MS_4)를 생성한다(S900). 다음으로, m 및 m'로 이루어지는 마스킹 값으로 입력데이터를 마스킹한다(S910).

<110> 또한, 외부로부터 입력되는 암호키를 마스킹하여 마스킹 암호키를 생성하고, 상기 마스킹 암호키를 이용하여 m 및 m'의 배타적 논리합 결과인 m"에 의해 마스킹된 확장키를 생성하고, 그리고 상기 확장키를 이용하여 m"에 의해 마스킹된 마스킹 라운드키를 생성함으로써 키 스케줄링 과정을 마스킹한다(S920).

<111> 상기 키 스케줄링 마스킹 방법을 구체적으로 설명하면, 상기 암호키의 상위 128비트(KL)를 $(m")^{16}$로 마스킹하고, 상기 암호키의 상기 상위 128비트 이외의 나머지 비트와 0으로 패딩(padding)한 비트를 이용하여 구성되는 128비트(KR)를 $(m'm'mmmmm'm')^{2}$로 마스킹하여 마스킹 암호키를 생성한다. 그 다음, 상기 마스킹 암호키를 입력으로 하여 3라운드 256비트 파이스텔 알고리듬을 이용하면서 추가 마스킹을 수행하여 마스킹 값 $(m")^{16}$로 마스킹된 확장키를 생성한다. 그리고나서 상기 확장키를 이용하여 마스킹 값 $(m")^{16}$로 마스킹된 마스킹 라운드키를 생성함으로써 키 스케줄링 과정을 마스킹한다.

<112> 다음으로, 상기 마스킹 라운드키 및 상기 네 종류의 마스킹 에스박스를 이용하여 아리아 라운드 함수 연산을 수행하고, 상기 아리아 라운드 함수 연산 결과가 다음 라운드 연산의 입력 값이 되도록 상기 아리아 라운드 함수 연산 결과에 추가 마스킹을 수행하여 암호연산 과정을 마스킹한다(S930).

<113> 마지막으로, 상기 암호연산 과정에서 수행된 최종라운드 연산 결과에 대해 디스마스킹(dismasking)을 수행하여 상기 입력데이터에 대응하는 암호 데이터를 출력한다(S940).

<114> 본 발명에 따른 아리아 마스킹 방법의 세부과정은 상기 본 발명에 따른 마스킹을 이용한 아리아 암호 장치에 관한 상세한 설명을 참조한다.

<115> 본 발명에 따른 마스킹을 이용한 아리아 암호 방법 및 아리아 마스킹 방법은, 데이터 처리 및 연산을 수행하는 컴퓨터로 판독할 수 있는 기록매체에 컴퓨터가 읽어드릴 수 있는 프로그램 코드로서 구현하는 것이 가능하다. 컴퓨터가 판독할 수 있는 기록매체에는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다.

<116> 컴퓨터가 판독할 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한, 컴퓨터가 판독할 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽어드릴 수 있는 코드가 저장되고 실행될 수 있다.

<117> 또한, 본 발명에 따른 마스킹을 이용한 아리아 암호 방법 및 아리아 마스킹 방법은 데이터 처리 및 연산을 수행하는 마이크로프로세서상에 상기 마이크로프로세서를 동작시키는 프로그램 코드로 구현하는 것이 가능하다.

<118> 이상에서, 설명의 편의를 위해 본 발명을 암호화 장치 및 방법에 한정하여 상세히 설명하였지만, 아리아는 암호화 과정 및 복호화 과정이 동일한 ISPN(Involution Substitution Permutation Network) 구조를 가지는 대칭키 암호 알고리듬이기 때문에, 복호화 장치 및 방법으로도 본 발명을 설명할 수 있음은 자명하다.

<119> 하기 표 2에는 일반 아리아 알고리듬(ARIA), 상기 기존 제1 방법, 및 상기 기존 제2 방법과 본 발명에 따른 마스킹을 이용한 아리아 암호 방법을 500000번 수행하여 그 연산 시간을 비교한 결과와 안정성을 비교한 결과 등이 나타나 있다.

표 2

	ARIA	기존 제1방법	기존 제2방법	본 발명
테이블크기	1024byte	2048byte	512byte	1536byte
키 스케줄	3.484s	해당 없음	해당 없음	4.047s
마스킹테이블	해당 없음	2.121s	0.637s	2.127s
암호연산	0.814s	1.125s	7.325s	0.851s
키 스케줄링 공격	무방비	무방비	무방비	안전
HW분석공격	무방비	무방비	무방비	안전

<120>

<121> 상기 표 2에서, HW 분석 공격이란 키가 마스킹되어 있지 않은 경우 키 에디션 연산 중에 키의 해밍웨이트를 알아내는 공격을 의미한다. 상기 표 2에 나타난 바와 같이, 본 발명에 따른 마스킹 방법은 두 종류의 에스박스만을 사용하며 암호 연산 중 추가적인 시간이 거의 들지 않는다. 따라서, 본 발명은 상기 기존 제1 방법에 비해 저장공간뿐만 아니라 연산시간 및 안전성 측면에서도 더 우수하다는 장점이 있다. 또한, 상기 기존 제2 방법의 경우 테이블 저장공간은 크지 않지만 치환 계층에 대한 연산과 마스킹 보정을 위한 추가적인 연산을 수행해야 하며, S_1, S_2, S_1^{-1}, S_2^{-1}가 존재하지 않기 때문에 키 스케줄링 과정에서 마스킹 방법 사용하려면 상당한 연산시간이 필요할 것이 자명하다. 따라서, 본 발명은 상기 기존 제2 방법에 비해 연산시간을 상당히 단축할 수 있다는 장점이 있다.

<122> 상술한 바와 같이, 본 발명에 따른 아리아 마스킹 방법과 이를 이용한 암호 장치 및 방법은, 기존의 아리아 마스킹 방법에 비해 DPA 공격에 대한 안전성은 물론, 키 스케줄링 공격에 대한 안전성도 제공한다. 또한, 기존의 아리아 마스킹 방법에 비해 하드웨어 자원 및 암호 연산시간을 감소시킨다는 이점을 제공한다.

<123> 지금까지 본 발명에 대하여 그 바람직한 실시예를 중심으로 살펴보았다. 그러나 당업자라면 본 발명의 본질적인 기술적 사상에서 벗어나지 않는 범위에서 본 발명이 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 즉, 본 발명의 범위는 상술한 설명이 아니라 특허청구범위에 나타나 있으며, 본 발명과 균등범위 내에 있는 모든 차이점은 본 발명에 포함되는 것으로 해석되어야 할 것이다.

도면의 간단한 설명

<124> 도 1은 ARIA의 전체 구조를 나타낸 블록도.

<125> 도 2는 ARIA의 키 스케줄링을 나타낸 블록도.

<126> 도 3은 본 발명에 따른 마스킹을 이용한 아리아 암호 장치의 일례를 나타낸 블록도.

<127> 도 4는 본 발명에 따른 마스킹을 이용한 아리아 암호 장치의 홀수라운드 연산부를 나타낸 블록도.

<128> 도 5는 본 발명에 따른 마스킹을 이용한 아리아 암호 장치의 짝수라운드 연산부를 나타낸 블록도.

<129> 도 6은 본 발명에 따른 마스킹을 이용한 아리아 암호 장치의 최종라운드 연산부를 나타낸 블록도.

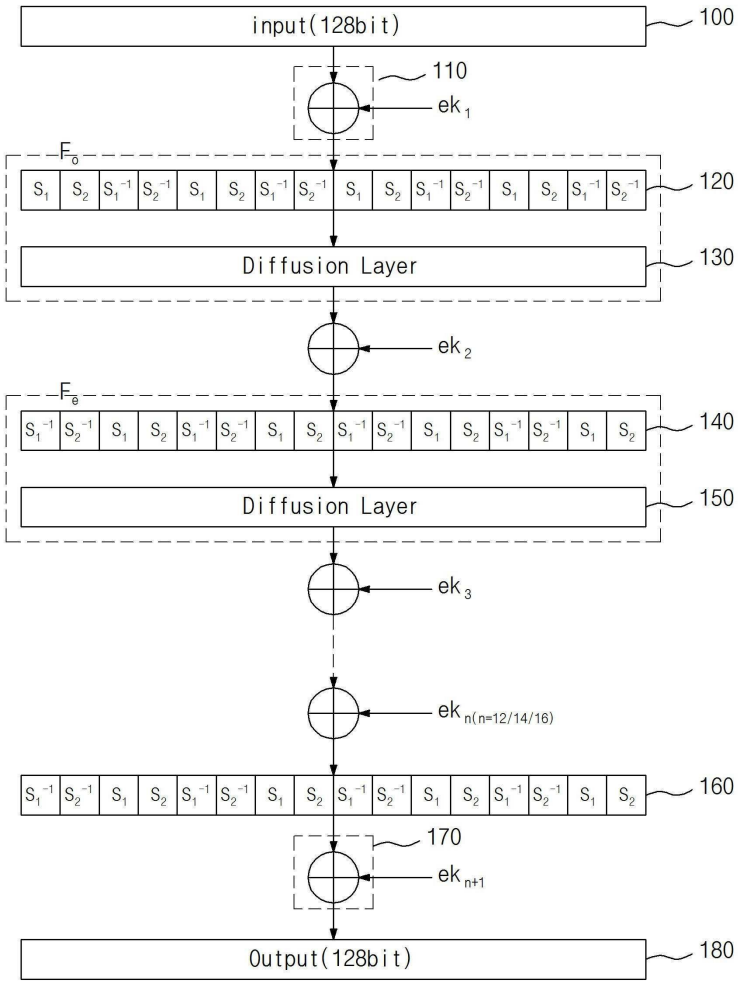
<130> 도 7은 본 발명에 따른 마스킹을 이용한 아리아 암호 장치의 암호키 마스킹부 및 키 확장부를 나타낸 블록도.

<131> 도 8은 본 발명에 따른 마스킹을 이용한 아리아 암호 방법의 일례를 나타낸 흐름도.

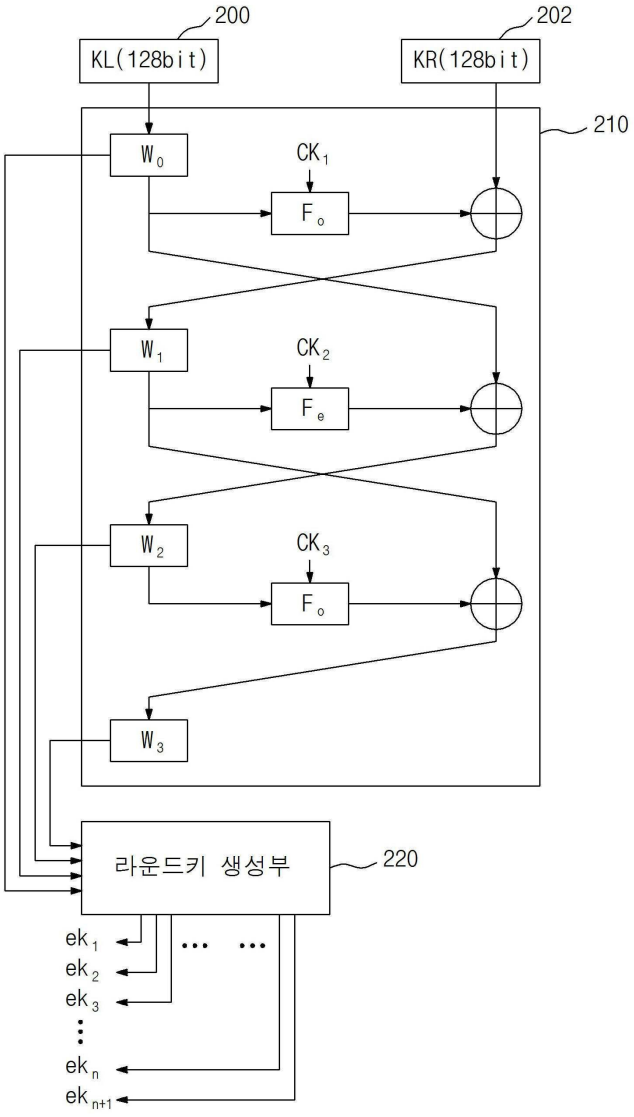
<132> 도 9는 본 발명에 따른 아리아 마스킹 방법의 일례를 나타낸 흐름도.

도면

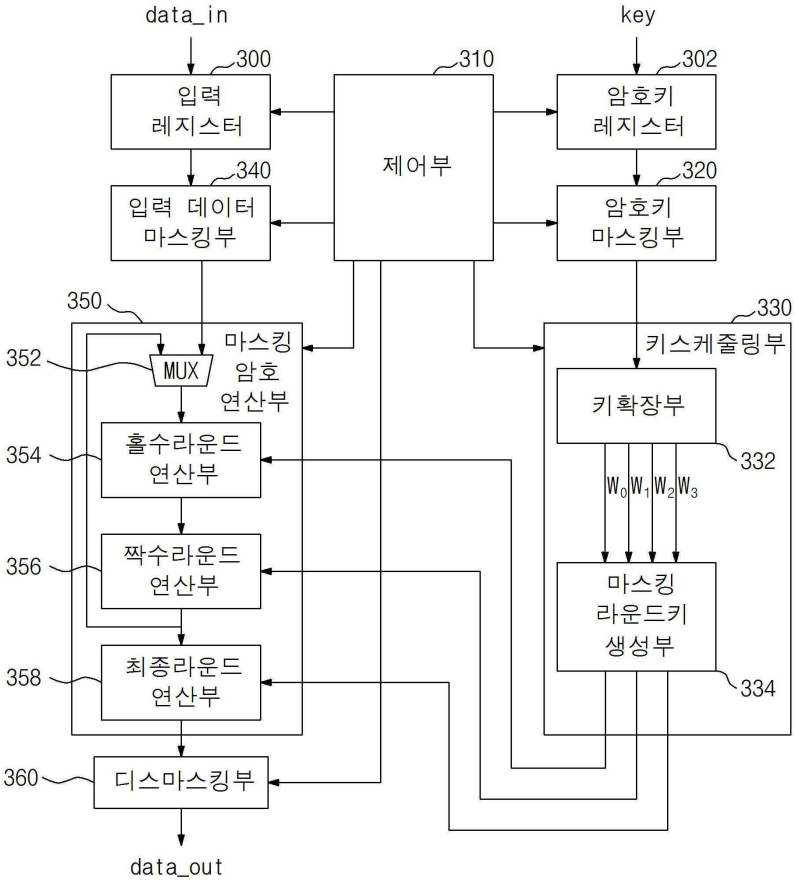
도면1


input(128bit)
100
110
ek1
Fo
S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1
120
Diffusion Layer
130
ek2
Fe
S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2
140
Diffusion Layer
150
ek3
ekn(n=12/14/16)
S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2 S1-1 S2-1 S1 S2
160
170
ekn+1
Output(128bit)
180

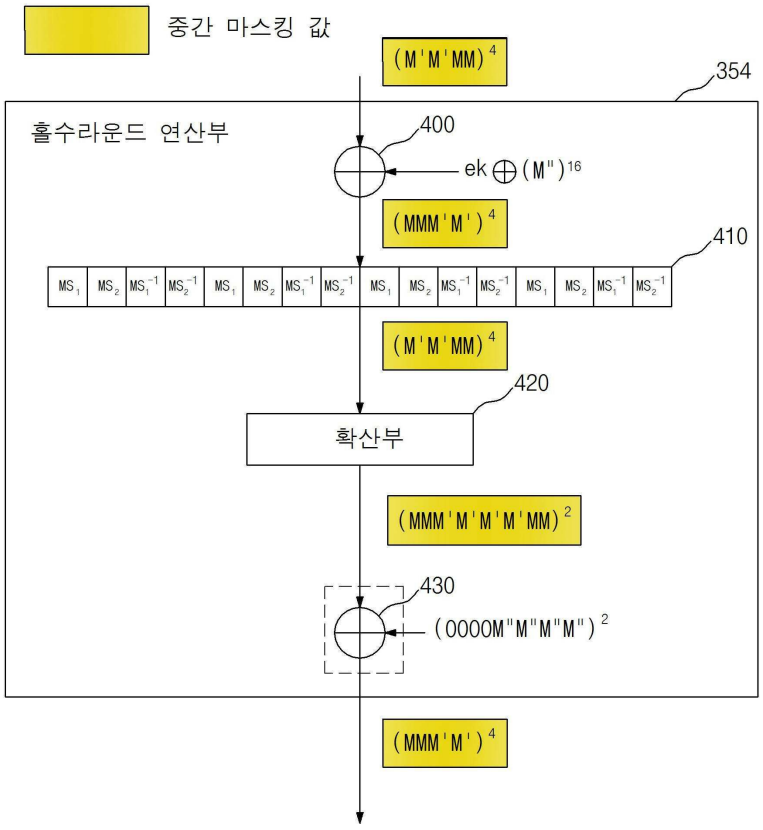
도면2


200
202
KL(128bit)
KR(128bit)
210
W0
CK1
Fo
W1
CK2
Fe
W2
CK3
Fo
W3
라운드키 생성부
220
ek1
ek2
ek3
ekn
ekn+1

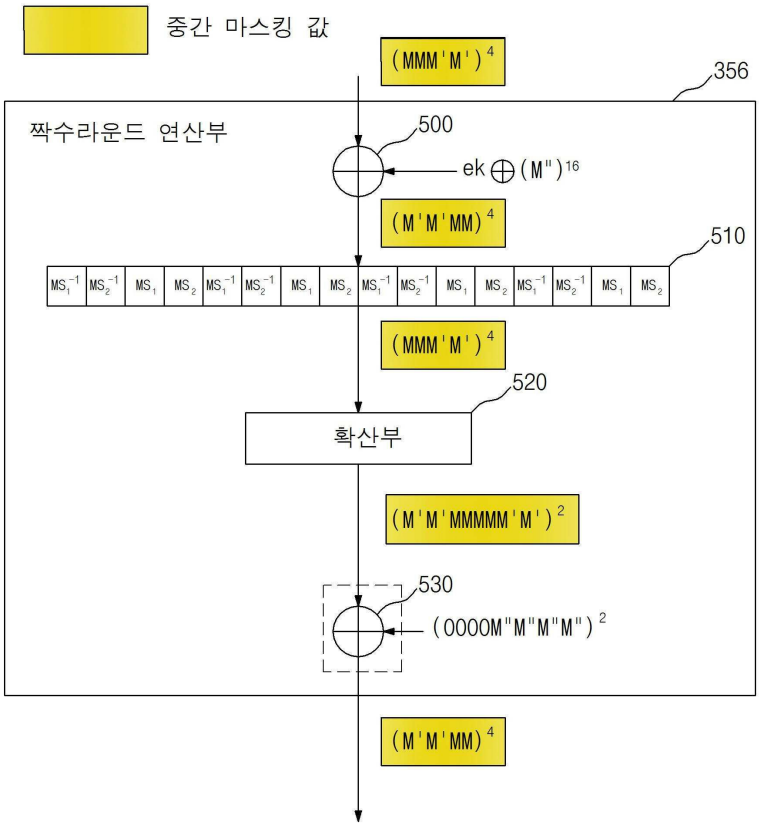
도면3


data_in
key
300
310
302
입력
레지스터
제어부
암호키
레지스터
340
320
입력 데이터
마스킹부
암호키
마스킹부
350
330
352
MUX
마스킹
암호
연산부
키스케줄링부
키확장부
354
홀수라운드
연산부
332
W0 W1 W2 W3
356
짝수라운드
연산부
마스킹
라운드키
생성부
358
최종라운드
연산부
334
360
디스마스킹부
data_out

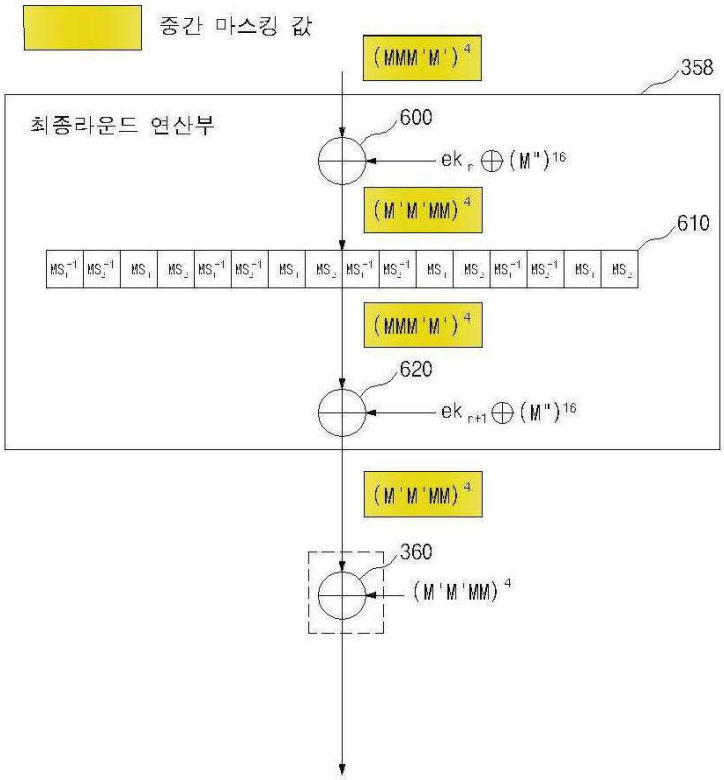
도면4


중간 마스킹 값
(M'M'MM)4
354
홀수라운드 연산부
400
ek ⊕ (M")16
(MMM'M')4
410
MS1 MS2 MS1-1 MS2-1 MS1 MS2 MS1-1 MS2-1 MS1 MS2 MS1-1 MS2-1 MS1 MS2 MS1-1 MS2-1
(M'M'MM)4
420
확산부
(MMM'M'M'MM)2
430
(0000M"M"M"M")2
(MMM'M')4

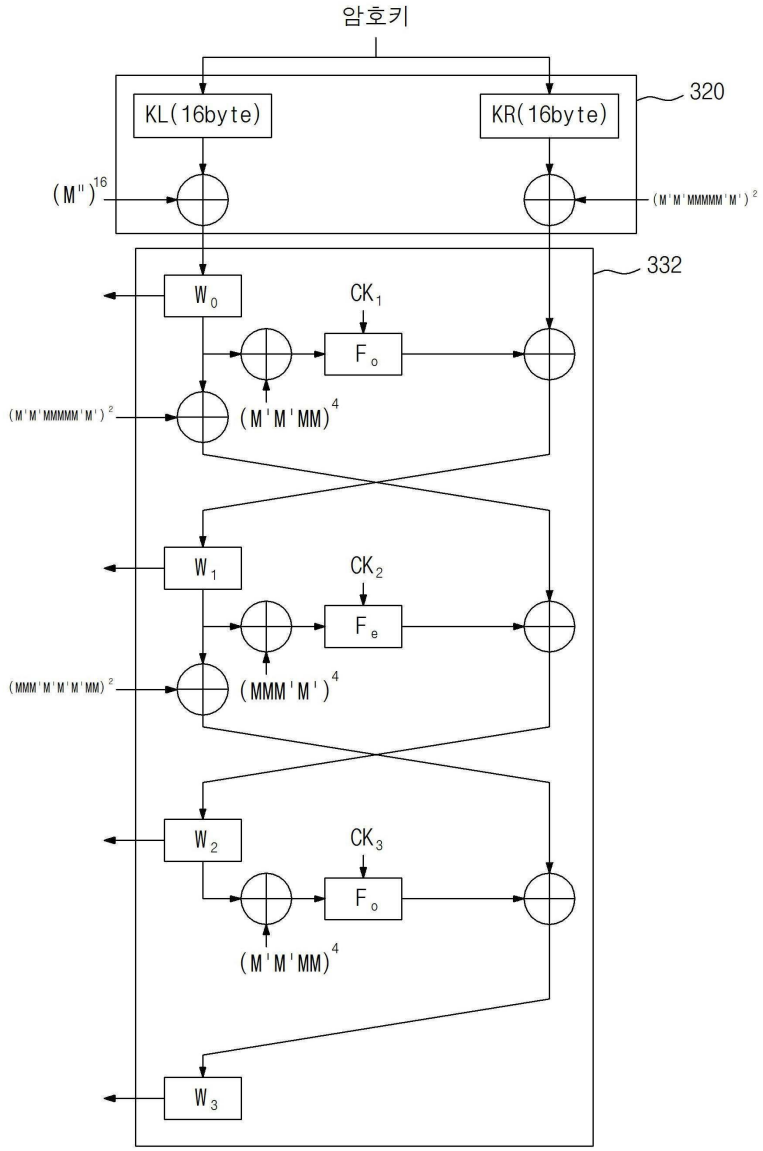
도면5


중간 마스킹 값
(MMM'M')4
356
짝수라운드 연산부
500
ek⊕(M")16
(M'M'MM)4
510
(MMM'M')4
520
확산부
(M'M'MMMMM'M')2
530
(0000M"M"M"M")2
(M'M'MM)4

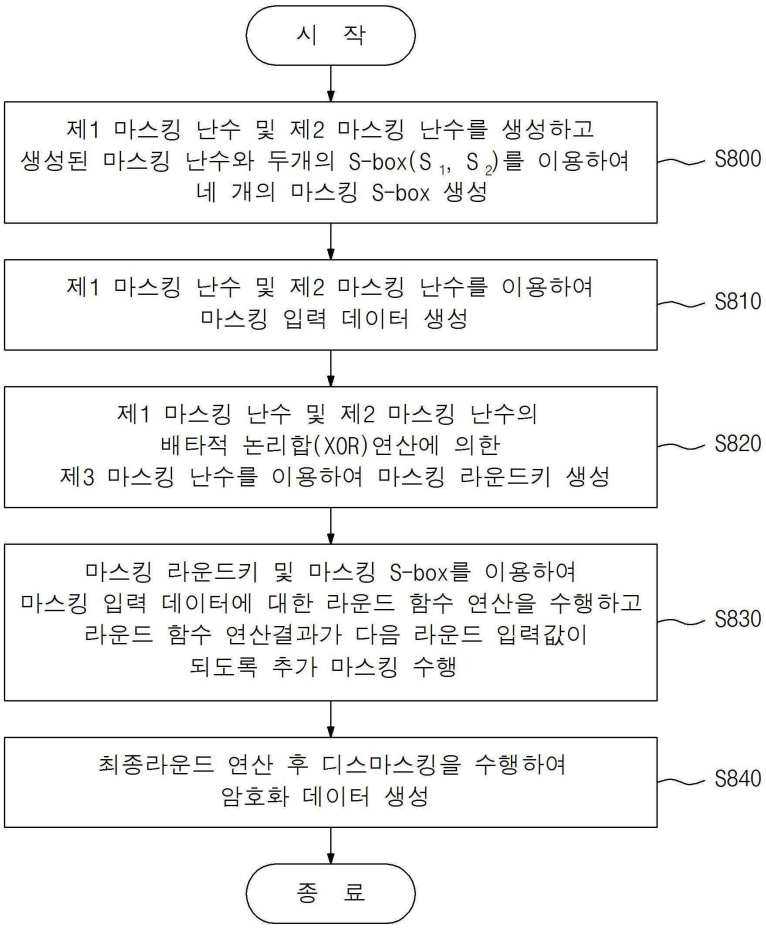
도면6


중간 마스킹 값
(MMM'M')4
358
최종라운드 연산부
600
ek r⊕(M")16
(M'M'MM)4
610
(MMM'M')4
620
ek r+1⊕(M")16
(M'M'MM)4
360
(M'M'MM)4

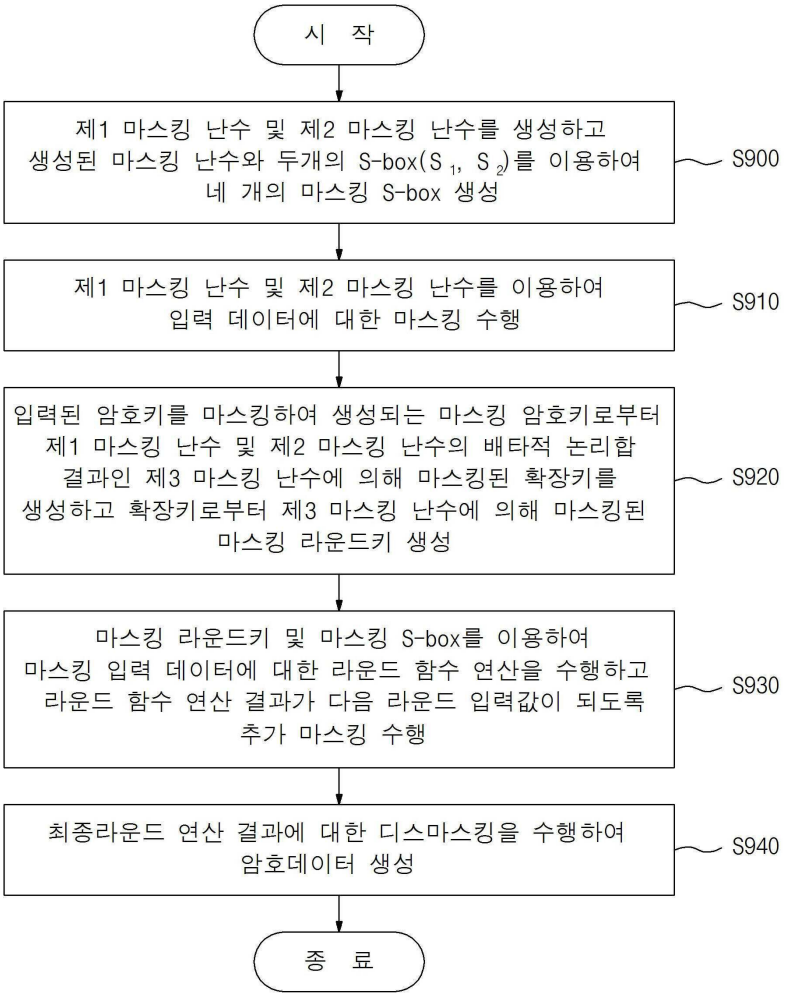
도면7


암호키
KL(16byte)
KR(16byte)
320
(M")16
(M'M'MMMM'M')2
332
W0
CK1
Fo
(M'M'MMMM'M')2
(M'M'MM)4
W1
CK2
Fe
(MMM'M'M'M'MM)2
(MMM'M')4
W2
CK3
Fo
(M'M'MM)4
W3

도면8


시 작
제1 마스킹 난수 및 제2 마스킹 난수를 생성하고
생성된 마스킹 난수와 두개의 S-box(S 1, S 2)를 이용하여
네 개의 마스킹 S-box 생성
S800
제1 마스킹 난수 및 제2 마스킹 난수를 이용하여
마스킹 입력 데이터 생성
S810
제1 마스킹 난수 및 제2 마스킹 난수의
배타적 논리합(XOR)연산에 의한
제3 마스킹 난수를 이용하여 마스킹 라운드키 생성
S820
마스킹 라운드키 및 마스킹 S-box를 이용하여
마스킹 입력 데이터에 대한 라운드 함수 연산을 수행하고
라운드 함수 연산결과가 다음 라운드 입력값이
되도록 추가 마스킹 수행
S830
최종라운드 연산 후 디스마스킹을 수행하여
암호화 데이터 생성
S840
종 료

도면9


시 작
제1 마스킹 난수 및 제2 마스킹 난수를 생성하고 생성된 마스킹 난수와 두개의 S-box(S_1, S_2)를 이용하여 네 개의 마스킹 S-box 생성
S900
제1 마스킹 난수 및 제2 마스킹 난수를 이용하여 입력 데이터에 대한 마스킹 수행
S910
입력된 암호키를 마스킹하여 생성되는 마스킹 암호키로부터 제1 마스킹 난수 및 제2 마스킹 난수의 배타적 논리합 결과인 제3 마스킹 난수에 의해 마스킹된 확장키를 생성하고 확장키로부터 제3 마스킹 난수에 의해 마스킹된 마스킹 라운드키 생성
S920
마스킹 라운드키 및 마스킹 S-box를 이용하여 마스킹 입력 데이터에 대한 라운드 함수 연산을 수행하고 라운드 함수 연산 결과가 다음 라운드 입력값이 되도록 추가 마스킹 수행
S930
최종라운드 연산 결과에 대한 디스마스킹을 수행하여 암호데이터 생성
S940
종 료