(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0076007
(43) 공개일자 2009년07월13일

(51) Int. Cl.
H04L 9/06 (2006.01) *H04L 9/32* (2006.01)
(21) 출원번호 10-2008-0001717
(22) 출원일자 2008년01월07일
심사청구일자 2008년01월07일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가1 고려대학교 내
(72) 발명자
이창훈
서울 성북구 동소문동6가 143-5 해피엔딩오피스텔 1006호
이상진
서울 노원구 중계본동 136 대림벽산아파트 103동 1102호
(뒷면에 계속)
(74) 대리인
김동진, 강경찬, 박경찬

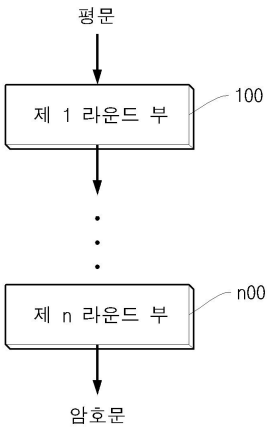
전체 청구항 수 : 총 34 항

(54) 데이터 의존 연산을 사용하는 암호화 방법

(57) 요 약

본 발명은 데이터 의존 연산을 사용하는 암호화 방법에 관한 것으로, 평문을 입력받으며, 상기 평문을 이용하여 소정의 연산을 수행한 결과인 데이터에 의존하여 암호화 연산을 수행하는 단계를 포함하며, 상기 암호화 연산을 수행하는 단계는 복수인 것을 특징으로 하여, 암호학적으로 안전하고 우수한 하드웨어 효율성을 갖춘다.

대 표 도 - 도1



(72) 발명자

몰도비얀 니콜레이 안드리비치

알렉산드롭스카야 스트리트 88-2 에피58 레닌그라드 오블. 브세볼로스크 188640 러시아

몰도비얀 알렉산드라 안드리비치

콩드레티에브스키 64-2 ap 139 세인트 피터스버그 195277 러시아

특허청구의 범위

청구항 1

평문을 입력받으며, 상기 평문을 이용하여 소정의 연산을 수행한 결과인 데이터에 의존하여 암호화 연산을 수행하는 단계를 포함하며,

상기 암호화 연산을 수행하는 단계는 복수인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 2

제 1항에 있어서, 상기 암호화 연산을 수행하는 단계는;

상기 평문을 특정 비트를 기준으로하여 복수의 서브평문으로 분할하는 단계;

상기 분할된 서브 평분과 기 설정된 라운드키를 이용하여 연산하는 단계;

상기 라운드 키를 이용한 연산의 출력을 입력을 받아서 제 1 암호화 연산을 수행하는 단계;

상기 제 1 암호화 연산의 출력 및 상기 라운드 키를 이용한 연산의 출력을 입력받아 비트를 확장하는 단계; 및

상기 라운드 키를 이용한 연산의 출력을 입력 변수로 하고, 상기 비트확장된 값을 컨트롤 변수로하여 제 2 암호화 연산을 수행하는 단계;를 포함하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 3

제 2항에 있어서,

상기 라운드키를 이용하여 연산을 수행하는 단계는 현재 평문이 전체 라운드에서 몇 번째 라운드에 해당하는지에 따라 상이한 라운드 키를 서브평문과 연산하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 4

제 3항에 있어서,

1번째 라운드는 K_1이 라운드 키 값이 되고, 2번째 라운드는 K_2가 라운드 키 값이 되며, 3번째 라운드는 K_3가 라운드 키 값이 되고, 4번째 라운드는 K_4가 라운드 키 값이 되며, 5번째 라운드는 K_1이 라운드 키 값이 되고, 6번째 라운드는 K_4가 라운드 키 값이 되며, 7번째 라운드는 K_2가 라운드 키 값이 되고, 8번째 라운드는 K_3가 라운드 키 값이 되고, 9번째 라운드는 K_2가 라운드 키 값이 되며,

상기 K_1 내지 K_4는 외부로부터 입력된 256비트의 비밀키를 4개의 64비트로 분할한 부분키인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 5

제 2항에 있어서,

상기 비트를 확장하는 단계는 (V1, Z1, V2, Z2, V3, Z3)의 조절벡터를 생성하며,

V1은 상기 입력에 대응되고, Z1은 상기 입력을 우측으로 2 비트만큼 순환이동한 값에 대응되고, V2는 상기 입력을 우측으로 6비트만큼 순환이동한 값에 대응되며, Z2는 상기 입력을 우측으로 8비트만큼 순환이동한 값에 대응되고, V3는 입력을 우측으로 10비트만큼 순환이동한 값에 대응되며, Z3는 입력을 우측으로 12비트만큼 순환이동한 값에 대응되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 6

제 2항에 있어서, 상기 제 1 암호화 연산을 수행하는 단계는;

상기 라운드 키를 이용한 연산의 출력의 비트를 분할한 각 출력값을 입력변수와 컨트롤 변수로 하여 제 1 연산

을 수행하는 단계;

상기 제 1 연산의 출력을 입력받아 각 비트들을 상호간에 치환하고, 특정 값에 매칭하는 제 2 연산을 수행하는 단계;

상기 라운드 키를 이용한 연산의 출력을 입력받아 상기 제 2 연산의 역으로 제 3 연산을 수행하는 단계; 및

상기 제 3 연산의 출력을 입력변수로 하고, 상기 제 2 연산의 출력을 컨트롤 변수로 하여 제 4 연산을 수행하는 단계;를 포함하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 7

제 6항에 있어서,

상기 제 1 연산을 수행하는 단계는 복수의 F2/2를 포함하며,

상기 각 F2/2는 x1, x2를 입력받아 v, z를 컨트롤 변수로 하여 y1, y2를 출력하며,

상기 y1은 vzx2와 vx2와 vx1과 zx1과 z와 x2 각각을 순서대로 배타적 논리합의 연산을 한 결과이며,

상기 y2는 vz1과 vz와 vx2와 zx1과 zx2와 x1 각각을 숭서대로 배타적 논리합 연산을 한 결과인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 8

제 6항에 있어서,

상기 제 2 연산을 수행하는 단계는 상기 제 1 연산의 출력의 비트들을 상호간에 치환하는 제 1 치환단계;

상기 제 1 치환단계에서 치환한 값들을 기 설정된 특정 값으로 매칭하는 제 1 매칭단계;

상기 제 2 매칭단계에서 매칭된 값의 비트들을 상호간에 치환하는 제 2 치환단계; 및

상기 제 2 치환단계에서 치환한 값들을 기 설정된 특정 값으로 매칭하는 제 2 매칭단계;를 포함하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 9

제 8항에 있어서,

상기 제 1 치환단계는 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 10번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 3번째 비트로 출력되고, 4번째 비트는 12번째 비트와 상호간에 치환되어 출력되고, 5번째 비트는 5번째 비트로 출력되고, 6번째 비트는 14번째 비트와 상호간에 치환되어 출력되고, 7번째 비트는 7번째 비트로 출력되고, 8번째 비트는 16번째 비트와 상호간에 치환되어 출력되고, 9번째 비트는 9번째 비트로 출력되고, 11번째 비트는 11번째 비트로 출력되고, 13번째 비트는 13번째 비트로 출력되고, 15번째 비트는 15번째 비트로 출력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 10

제 8항에 있어서,

상기 제 2 치환단계는 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 5번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 4번째 비트는 13번째 비트와 상호간에 치환되어 출력되고, 6번째 비트는 6번째 비트로 출력되고, 7번째 비트는 10번째 비트와 상호간에 치환되어 출력되고, 8번째 비트는 14번째 비트와 상호간에 치환되어 출력되고, 11번째 비트는 11번째 비트로 출력되고, 12번째 비트는 15번째 비트와 상호간에 치환되어 출력되고, 16번째 비트는 16번째 비트로 출력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 11

제 6항에 있어서,

상기 제 4 연산은 상기 제 1 연산과 동일한 연산과정을 수행하는 것을 특징으로 하는 데이터 의존 연산을 사용

하는 암호화 연산방법.

청구항 12

제 2항에 있어서, 상기 제 2 암호화 연산을 수행하는 단계는;

상기 라운드 키를 이용한 연산의 출력을 입력변수로 하고, 상기 비트확장된 값을 컨트롤 변수로 하여 제 5 연산을 수행하는 단계;

상기 제 5 연산의 출력의 비트들을 상호간에 치환하는 제 3 치환단계; 및

상기 제 3 치환의 출력을 입력변수로 하고, 상기 제 2 연산의 출력을 입력받아 비트를 확장한 출력을 컨트롤 변수로 하여 제 6 연산을 수행하는 단계;를 포함하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 13

제 12항에 있어서, 상기 제 5 연산을 수행하는 단계는 복수의 F8/24를 포함하며,

상기 입력의 1번째 비트부터 8번째 비트는 제 1 F8/24로 입력되고, 9번째 비트부터 16번째 비트는 제 2 F8/24로 입력되고, 17번째 비트부터 24번째 비트는 제 3 F8/24로 입력되고, 제 25번째 비트부터 32번째 비트는 제 4 F8/24로 입력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 14

제 12항에 있어서,

상기 제 3 치환단계는 상기 라운드 키를 이용한 연산의 출력에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 17번째 비트와 상호간에 치환되어 출력되고, 4번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 5번째 비트는 5번째 비트로 출력되고, 6번째 비트는 13번째 비트와 상호간에 치환되어 출력되고, 7번째 비트는 21번째와 상호간에 치환되어 출력되고, 8번째 비트는 29번째 비트와 상호간에 치환되어 출력되고, 10번째 비트는 10번째 비트로 출력되고, 11번째 비트는 18번째 비트와 상호간에 치환되어 출력되고, 12번째 비트는 26번째 비트와 상호간에 치환되어 출력되고, 14번째 비트는 14번째 비트로 출력되고, 15번째 비트는 22번째 비트로 출력되고, 16번째 비트는 30번째 비트와 상호간에 치환되어 출력되고, 19번째 비트는 19번째 비트로 출력되고, 20번째 비트는 27번째 비트와 상호간에 치환되어 출력되고, 23번째 비트는 23번째 비트로 출력되고, 24번째 비트는 31번째 비트와 상호간에 치환되어 출력되고, 28번째 비트는 28번째 비트로 출력되고, 32번째 비트는 32번째 비트로 출력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 15

제 12항에 있어서, 상기 제 6 연산을 수행하는 단계는 복수의 F^{-1}8/24를 포함하며,

상기 입력의 1번째 비트부터 8번째 비트는 제 1 F^{-1}8/24로 입력되고, 9번째 비트부터 16번째 비트는 제 2 F^{-1}8/24로 입력되고, 17번째 비트부터 24번째 비트는 제 3 F^{-1}8/24로 입력되고, 제 25번째 비트부터 32번째 비트는 제 4 F^{-1}8/24로 입력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 16

제 1항에 있어서, 상기 암호화 연산을 수행하는 단계는;

상기 평문을 특정 비트를 기준으로 하여 복수의 서브평문으로 분할하는 단계;

상기 분할된 서브 평분과 기 설정된 라운드키를 이용하여 연산하는 단계;

상기 라운드 키를 이용한 연산의 출력을 입력을 받아서 제 3 암호화 연산을 수행하는 단계;

상기 제 3 암호화 연산의 출력 및 상기 라운드 키를 이용한 연산의 출력을 입력받아 비트를 확장하는 단계; 및

상기 라운드 키를 이용한 연산의 출력을 입력 변수로 하고, 상기 비트확장된 값을 컨트롤 변수로하여 제 4 암호화 연산을 수행하는 단계;를 포함하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 17

제 16항에 있어서,

상기 라운드키를 이용하여 연산을 수행하는 단계는 현재 평문이 전체 라운드에서 몇 번째 라운드에 해당하는지에 따라 상이한 라운드 키를 서브평문과 연산하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 18

제 17항에 있어서,

1번째 라운드는 K_1이 라운드 키 값이 되고, 2번째 라운드는 K_2가 라운드 키 값이 되며, 3번째 라운드는 K_3가 라운드 키 값이 되고, 4번째 라운드는 K_4가 라운드 키 값이 되며, 5번째 라운드는 K_2이 라운드 키 값이 되고, 6번째 라운드는 K_1가 라운드 키 값이 되며, 7번째 라운드는 K_3가 라운드 키 값이 되고, 8번째 라운드는 K_4가 라운드 키 값이 되고, 9번째 라운드는 K_3가 라운드 키 값이 되며, 10번째 라운드는 K_2가 라운드 키 값이 되고, 11번째 라운드는 K_1이 라운드 키 값이 된다.

상기 K_1 내지 K_4는 외부로부터 입력된 256비트의 비밀키를 4개의 64비트로 분할한 부분키인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 19

제 16항에 있어서,

상기 비트를 확장하는 단계는 (V4, Z4, V5, Z5, V6, Z6)의 조절 벡터를 생성하며,

V4는 상기 제 3 암호화 연산의 입력에 대응되고, Z4은 제 3 암호화 연산의 입력을 우측으로 5비트만큼 순환이동한 값에 대응되고, V5는 제 3 암호화 연산의 입력을 우측으로 10비트만큼 순환이동한 값에 대응되고, Z5는 제 3 암호화 연산의 입력을 우측으로 15비트만큼 순환이동한 값에 대응되고, V6는 제 3 암호화 연산의 입력을 우측으로 20비트만큼 순환이동한 값에 대응되고, Z6는 제 3 암호화 연산의 입력을 우측으로 25비트만큼 순환이동한 값에 대응되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 20

제 16항에 있어서, 상기 제 3 암호화 연산을 수행하는 단계는;

상기 라운드 키를 이용한 연산의 출력을 비트 분할한 각 출력값을 입력변수와 컨트롤 변수로 하여 제 7 연산을 수행하는 단계;

상기 제 7 연산의 출력을 입력받아 각 비트들을 상호간에 치환하고, 특정 값에 매칭하는 제 8 연산을 수행하는 단계;

상기 라운드 키를 이용한 연산의 출력을 입력받아 상기 제 8 연산의 역으로 제 9 연산을 수행하는 단계; 및

상기 제 9 연산의 출력을 입력변수로 하고, 상기 제 8 연산의 출력을 컨트롤 변수로 하여 제 10 연산을 수행하는 단계;를 포함하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 21

제 20항에 있어서, 상기 제 7 연산을 수행하는 단계는 복수의 F2/2를 포함하며,

상기 각 F2/2는 x1, x2를 입력받아 v, z를 컨트롤 변수로 하여 y1, y2를 출력하며,

상기 y1는 vzx1과 vzx2와 vx1과 vx2와 zx1과 zx2와 z와 v와 x2 각각을 순서대로 배타적 논리합의 연산을 한 결과이며,

상기 y2는 vzx1과 vzx2와 vz와 vx1과 vx2와 zx1과 zx2와 x1 각각을 순서대로 배타적 논리합의 연산을 한 결과인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 22

제 20항에 있어서,

상기 제 8 연산을 수행하는 단계는 상기 제 7 연산의 출력의 비트들을 상호간에 치환하는 제 4 치환단계;

상기 제 4 치환단계에서 치환한 값들을 기 설정된 특정 값으로 매칭하는 제 3 매칭단계;

상기 제 3 매칭단계에서 매칭된 값의 비트들을 상호간에 치환하는 제 5 치환단계; 및

상기 제 5 치환단계에서 치환한 값들을 기 설정된 특정 값으로 매칭하는 제 4 매칭단계;를 포함하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 23

제 22항에 있어서,

상기 제 4 치환단계는 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 18번째 비트와 상호간에 치환되어 출력되며, 3번째 비트는 3번째 비트로 출력되고, 4번째 비트는 20번째 비트와 상호간에 치환되어 출력되며, 5번째 비트는 5번째 비트로 출력되고, 6번째 비트는 22번째 비트와 상호간에 치환되어 출력되며, 7번째 비트는 7번째 비트로 상호간에 치환되어 출력되고, 8번째 비트는 24번째 비트와 상호간에 치환되어 출력되며, 9번째 비트는 9번째 비트로 출력되고, 10번째 비트는 26번째 비트와 상호간에 치환되어 출력되며, 11번째 비트는 11번째 비트로 출력되고, 12번째 비트는 28번째 비트와 상호간에 치환되어 출력되며, 13번째 비트는 13번째 비트로 출력되고, 14번째 비트는 30번째 비트와 상호간에 치환되어 출력되며, 15번째 비트는 15번째 비트로 출력되고, 16번째 비트는 32번째 비트와 상호간에 치환되어 출력되며, 17번째 비트는 17번째 비트로 출력되고, 19번째 비트는 19번째 비트로 출력되고, 21번째 비트는 21번째 비트로 출력되고, 23번째 비트는 23번째 비트로 출력되고, 25번째 비트는 25번째 비트로 출력되고, 27번째 비트는 27번째 비트로 출력되고, 29번째 비트는 29번째 비트로 출력되고, 31번째 비트는 31번째 비트로 출력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 24

제 22항에 있어서,

상기 제 5 치환단계는 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 5번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 4번째 비트는 13번째 비트와 상호간에 치환되어 출력되고, 6번째 비트는 6번째 비트로 출력되고, 7번째 비트는 10번째 비트와 상호간에 치환되어 출력되고, 8번째 비트는 14번째 비트와 상호간에 치환되어 출력되고, 11번째 비트는 11번째 비트로 출력되고, 12번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 16번째 비트는 16번째 비트로 출력되고, 17번째 비트는 17비트로 출력되고, 18번째 비트는 21번째 비트와 상호간에 치환되어 출력되고, 19번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 20번째 비트는 29번째 비트와 상호간에 치환되어 출력되고, 22번째 비트는 22번째 비트로 출력되고, 23번째 비트는 26번째 비트와 상호간에 치환되어 출력되고, 24번째 비트는 30번째 비트와 상호간에 치환되어 출력되고, 27번째 비트는 27번째 비트로 출력되고, 28번째 비트는 31번째 비트와 상호간에 치환되어 출력되고, 32번째 비트는 32번째 비트로 출력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 25

제 20항에 있어서,

상기 제 10 연산은 상기 제 7 연산과 동일한 연산과정을 수행하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 26

제 16항에 있어서, 상기 제 4 암호화 연산을 수행하는 단계는;

상기 라운드 키를 이용한 연산의 출력을 입력받아 비트들을 상호간에 치환하는 제 6 치환단계;

상기 제 6 치환단계의 출력을 입력변수로 하고, 상기 비트를 확장한 값을 컨트롤 변수로 하여 제 11 연산을 수

행하는 단계;

상기 제 11 연산의 출력의 비트들을 상호간에 치환하는 제 7 치환단계;

상기 제 7 치환단계의 출력을 입력변수로 하고, 상기 9 연산의 출력을 입력받아 비트를 확장한 값을 컨트롤 변수로 하여 제 12 연산을 수행하는 단계; 및

상기 제 12 연산의 출력의 비트들을 상호간에 치환하는 제 8 치환단계;를 포함하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 27

제 26항에 있어서, 상기 제 6 치환단계는 입력되는 값에서 홀수번째 비트는 동일한 비트로 출력되고, 짝수번째 비트는 (2j, 2j+32)비트로 출력되며, 상기 j는 1 이상 32 이하의 값인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 28

제 26항에 있어서, 상기 제 11 연산을 수행하는 단계는 복수의 F8/24를 포함하며,

상기 입력의 1번째 비트부터 8번째 비트는 제 1 F8/24로 입력되고, 9번째 비트부터 16번째 비트는 제 2 F8/24로 입력되고, 17번째 비트부터 24번째 비트는 제 3 F8/24로 입력되고, 제 25번째 비트부터 32번째 비트는 제 4 F8/24로 입력되고, 제 33번째 비트부터 40번째 비트는 제 5 F8/24로 입력되고, 제 41번째 비트부터 48번째 비트는 제 6 F8/24로 입력되고, 제 49번째 비트부터 56번째 비트는 제 7 F8/24로 입력되고, 제 57번째 비트부터 64번째 비트는 제 8 F8/24로 입력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 29

제 13항 또는 제 28항에 있어서, 상기 F8/24는 3열의 복수의 F2/2를 포함하며,

제 1열 복수의 F2/2는 상기 라운드 키 연산부의 출력을 입력받고, V1, Z1을 컨트롤 변수로 하여 연산을 수행하며, 제 1열 1번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 3번째 F2/2로 입력되고, 제 1열 2번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되며, 제 2 출력은 제 2열 3번째 F2/2로 입력되고, 제 1열 3번째 F2/2의 제 1 출력은 제 2열 2번째 F2/2로 입력되고, 제7 2 출력은 제 2열 4번째 F2/2로 입력되며, 제 1열 4번째 F2/2의 제 1 출력은 제 2열 2번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력되며,

제 2열 복수의 F2/2는 상기 제 1열 복수의 F2/2의 출력을 입력받고, V2, Z2를 컨트롤 변수로 하여 연산을 수행하며, 제 2열 1번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력되며, 제 2열 2번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력되며, 제 2열 3번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력되며, 제 2열 4번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력되며,

제 3열 복수의 F2/2는 상기 제 2열 복수의 F2/2의 출력을 입력받고, V3, Z3를 컨트롤 변수로 하여 연산을 수행하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 30

제 26항에 있어서, 상기 제 12 연산을 수행하는 단계는 복수의 F^{-1}8/24를 포함하며,

상기 입력의 1번째 비트부터 8번째 비트는 제 1 F^{-1}8/24로 입력되고, 9번째 비트부터 16번째 비트는 제 2 F^{-1}8/24로 입력되고, 17번째 비트부터 24번째 비트는 제 3 F^{-1}8/24로 입력되고, 제 25번째 비트부터 32번째 비트는 제 4 F^{-1}8/24로 입력되고, 33번째 비트부터 40번째 비트는 제 5 F^{-1}8/24로 입력되고, 41번째 비트부터 48번째 비트는 제 6 F^{-1}8/24로 입력되고, 49번째 비트부터 56번째 비트는 제 7 F^{-1}8/24로 입력되고, 57번째 비트부터 64번째 비트는 제 8 F^{-1}8/24로 입력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 31

제 15항 또는 제 31항에 있어서, 상기 제 $F^{-1}8/24$은 3열의 복수의 F2/2를 포함하며,

제 1열 복수의 F2/2는 상기 제 3 치환단계의 출력을 입력받고, V3, Z3를 컨트롤 변수로 하여 연산을 수행하며, 제 1열 1번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 2번째 F2/2로 입력되며, 제 1열 2번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 2째 F2/2로 입력되며, 제 1열 3번째 F2/2의 제 1 출력은 제 2열 3번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력되며, 제 1열 4번째 F2/2의 제 1 출력은 제 2열 3번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력되며,

제 2열 복수의 F2/2는 상기 제 1열 복수의 F2/2의 출력을 입력받고, V2, Z2를 컨트롤 변수로 하여 연산을 수행하며, 제 2열 1번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력되며, 제 2열 2번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력되며, 제 2열 3번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력되며, 제 2열 4번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력되며,

제 3열 복수의 F2/2는 상기 제 2열 복수의 F2/2의 출력을 입력받고, V1, Z1를 컨트롤 변수로 하여 연산을 수행하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 32

제 26항에 있어서,

상기 제 7 치환단계는 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 17번째 비트로 출력되고, 4번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 5번째 비트는 33번째 비트와 상호간에 치환되어 출력되고, 6번째 비트는 41번째 비트와 상호간에 치환되어 출력되고, 7번째 비트는 49번째 비트와 상호간에 치환되어 출력되고, 8번째 비트는 57번째 비트와 상호간에 치환되어 출력되고, 10번째 비트는 10번째 비트로 출력되고, 11번째 비트는 18번째 비트와 상호간에 치환되어 출력되고, 12번째 비트는 26번째 비트와 상호간에 치환되어 출력되고, 15번째 비트는 50번째 비트와 상호간에 치환되어 출력되며, 16번째 비트는 58번째 비트와 상호간에 치환되어 출력되고, 19번째 비트는 19번째 비트로 출력되고, 20번째 비트는 27번째 비트와 상호간에 치환되어 출력되며, 21번째 비트는 35번째 비트와 상호간에 치환되어 출력되고, 22번째 비트는 43번째 비트와 상호간에 치환되어 출력되며, 23번째 비트는 51번째 비트와 상호간에 치환되어 출력되고, 24번째 비트는 59번째 비트와 상호간에 치환되어 출력되며, 28번째 비트는 28번째 비트로 출력되고, 29번째 비트는 36번째 비트와 상호간에 치환되어 출력되며, 30번째 비트는 44번째 비트와 상호간에 치환되어 출력되고, 31번째 비트는 52번째 비트와 상호간에 치환되어 출력되며, 32번째 비트는 60번째 비트와 상호간에 치환되어 출력되며, 37번째 비트는 37번째 비트로 출력되고, 38번째 비트는 45번째 비트와 상호간에 치환되어 출력되며, 39번째 비트는 53번째 비트와 상호간에 치환되어 출력되며, 40번째 비트는 61번째 비트와 상호간에 치환되어 출력되며, 46번째 비트는 46번째 비트로 출력되고, 47번째 비트는 54번째 비트와 상호간에 치환되어 출력되며, 48번째 비트는 62번째 비트와 상호간에 치환되어 출력되며, 55번째 비트는 55번째 비트로 출력되고, 56번째 비트는 63번째 비트와 상호간에 치환되어 출력되며, 64비트는 64비트로 출력되는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 33

제 26항에 있어서, 상기 제 8 치환단계는 상기 제 6 치환단계와 동일한 연산과정을 수행하는 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

청구항 34

제 3항 또는 제 17항에 있어서,

상기 라운드 키를 이용한 연산은 상기 라운드 키와 상기 서브평문과의 배타적 논리합 연산인 것을 특징으로 하는 데이터 의존 연산을 사용하는 암호화 연산방법.

명 세 서

발명의 상세한 설명

기 술 분 야

<1> 본 발명은 데이터 의존 연산을 사용하는 암호화 방법에 관한 것으로, 특히 평문이 특정 데이터에 따라서 상이하게 암호화되는 데이터 의존 연산을 사용하는 암호화 방법에 관한 발명이다.

<2> 본 발명은 컴퓨터 기술 및 전기 통신 분야에 관한 것으로서, ad-hoc, sensor, wireless 등 다양한 네트워크 환경에 적용 가능한 하드웨어 기반 블록 암호 알고리즘 개발에 관한 것이다.

배 경 기 술

<3> 최근 유비쿼터스 컴퓨팅, 센서 네트워크, 홈 네트워크, 휴대 인터넷 등과 같은 새로운 정보통신 환경이 현실적으로 구현되어 상용화되고 있다. 그러나 현재 사용되고 있는 AES, SEED, ARIA 등과 같은 표준 블록 암호 알고리즘들은 개발 당시 컴퓨팅 요구 사항에 맞게 설계되어 있기 때문에, 경량의 고속 연산을 요구하는 신규 응용 네트워크 환경에서 사용되기에는 무리가 있다. 특히, 기존의 알고리즘들은 이론적인 안정성에 중점을 두고 설계되어 있고 하드웨어 구현 효율성이 좋지 못하다.

<4> 기존의 데이터 의존 연산을 사용하는 블록 암호 알고리즘들은 대부분 데이터 의존 비트 순환 연산(Data-Dependent Rotation, DDR)이나 데이터 의존 치환 함수(Data-Dependent Permutation, DDP)를 사용한다. 그러나 DDR은 상대적으로 작은 데이터의 변화를 유도하고 DDP는 선형 함수이기 때문에 암호학적으로 좋지 못한 특성을 갖고 있다.

발명의 내용

해결 하고자하는 과제

<5> 본 발명은 암호학적으로 취약한 DDP 대신에 데이터에 의존하는 SPN박스(Controlled Substitution-Permutation Networks, CSPN)를 설계한다. 그리고 CSPN과 대치함수인 S 박스들로 구성된 인볼루션(Involution)함수와 결합하는 신규 암호 설계 논리를 제시하여, 암호학적으로 안전할 뿐만 아니라 우수한 하드웨어 효율성을 갖춘 데이터 의존 연산을 사용하는 암호화 방법을 제공하는 데 목적이 있다.

과제 해결수단

<6> 상기 목적을 이루기 위한 본 발명의 데이터 의존 연산을 사용하는 암호화 방법은 평문을 입력받으며, 상기 평문을 이용하여 소정의 연산을 수행한 결과인 데이터에 의존하여 암호화 연산을 수행하는 단계를 포함하며, 상기 암호화 연산을 수행하는 단계는 복수인 것을 특징으로 한다.

<7> 상기 암호화 연산을 수행하는 단계는 상기 평문을 특정 비트를 기준으로하여 복수의 서브평문으로 분할하는 단계와, 상기 분할된 서브 평분과 기 설정된 라운드키를 이용하여 연산하는 단계와, 상기 라운드 키를 이용한 연산의 출력을 입력을 받아서 제 1 암호화 연산을 수행하는 단계와, 상기 제 1 암호화 연산의 출력 및 상기 라운드 키를 이용한 연산의 출력을 입력받아 비트를 확장하는 단계와, 상기 라운드 키를 이용한 연산의 출력을 입력변수로 하고, 상기 비트확장된 값을 컨트롤 변수로하여 제 2 암호화 연산을 수행하는 단계를 포함하는 것을 특징으로 한다.

<8> 상기 암호화 연산을 수행하는 단계는 상기 평문을 특정 비트를 기준으로 하여 복수의 서브평문으로 분할하는 단계와, 상기 분할된 서브 평분과 기 설정된 라운드키를 이용하여 연산하는 단계와, 상기 라운드 키를 이용한 연산의 출력을 입력을 받아서 제 3 암호화 연산을 수행하는 단계와, 상기 제 3 암호화 연산의 출력 및 상기 라운드 키를 이용한 연산의 출력을 입력받아 비트를 확장하는 단계와, 상기 라운드 키를 이용한 연산의 출력을 입력변수로 하고, 상기 비트확장된 값을 컨트롤 변수로하여 제 4 암호화 연산을 수행하는 단계를 포함하는 것을 특징으로 한다.

효 과

<9> 본 발명의 데이터 의존 연산을 사용하는 암호화 방법은 암호학적으로 안전하고 우수한 하드웨어 효율성을 갖춘다.

발명의 실시를 위한 구체적인 내용

<10> 이하, 첨부된 도면을 참조하여 본 발명의 데이터 의존 연산을 사용하는 암호화 방법을 상세히 설명한다.

<11> 도 1은 본 발명의 데이터 의존 연산을 사용하는 암호화 모듈을 나타낸 도면이다. 도시된 바와 같이, 본 발명의 데이터 의존 연산을 사용하는 암호화 모듈은 평문을 입력받는다. 암호화 모듈은 상기 입력된 평문과, 평문을 이용하여 소정의 연산을 수행한 결과인 데이터에 의존하여 암호화 연산을 수행하는 복수의 라운드부(100, …)를 포함한다. 상기 라운드부(100, …)의 개수는 입력되는 평문의 비트수에 관련된다. 즉, 암호화 모듈은 입력되는 평문이 64비트인 경우에 9개의 라운드부를 포함하고, 입력되는 평문이 128비트인 경우에 11개의 라운드부를 포함한다. 한편, 상기 암호화 모듈은 상위 라운드의 복수의 출력이 하위 라운드로 입력될때 상호간에 도치된다. 다만, 암호화 모듈의 마지막 라운드인 9번째 라운드 및 11번째 라운드에서는 상기 도치없이 출력된다.

<12> 도 2는 도 1의 라운드부의 일실시 예를 나타낸 도면이다. 일실시 예는 평문의 입력비트가 64비트인 경우이다. 도시된 바와 같이 라운드부는 비트 분할부(200), 라운드 키 연산부(210), 복수의 CSPN(220, 230)부 및 비트 확장부(240)를 포함한다.

<13> 상기 비트 분할부(200)는 외부로부터 입력된 평문을 특정 비트를 기준으로 하여 복수의 서브평문으로 분할한다. 암호화 모듈은 암호화될 평문을 입력받는다. 비트 분할부(200)는 상기 입력받은 평문을 특정 비트를 기준으로 복수의 서브평문으로 분할하여, 상기 분할된 각 서브평문이 상이한 암호화 연산을 통해 암호화될 수 있도록 한다. 예를 들어 평문이 64비트인 경우에, 비트 분할부(200)는 상기 64비트의 평문을 최상위 비트로부터 32번째 비트를 기준으로 분할한다. 즉, 비트 분할부(200)는 각 32비트의 복수의 서브평문을 출력한다. 다만, 입력되는 평문 및 서브평문으로 분할하는데 기준이 되는 특정 비트는 상기 예에 한정되는 것은 아니다.

<14> 상기 라운드 키 연산부(210)는 상기 비트 분할부(200)에서 출력된 각 서브평문과 기 설정된 라운드키를 이용하여 연산한다. 라운드 키 연산부(210)는 현재 평문이 전체 라운드에서 몇 번째 라운드에 해당하는지에 따라 상이한 라운드키를 서브평문과 연산한다. 상기 라운드 키 연산부(210)는 상기 라운드 키와 서브평문과의 배타적 논리합(Exclusive OR) 연산을 수행한다. 도 3은 암호화 모듈에 기 설정된 라운드 키의 값이다. 도시된 바와 같이, 상기 K_1 내지 K_4는 128비트의 임의의 비밀키 K를 4개의 32비트로 분할한 부분키들이다. 상기 라운드 키 값은 현재 라운드가 몇 번째 라운드인지에 따라서 상이한 값을 갖는다. 1번째 라운드는 K_1이 라운드 키 값이 되고, 2번째 라운드는 K_2가 라운드 키 값이 되며, 3번째 라운드는 K_3가 라운드 키 값이 되고, 4번째 라운드는 K_4가 라운드 키 값이 되며, 5번째 라운드는 K_1이 라운드 키 값이 되고, 6번째 라운드는 K_4가 라운드 키 값이 되며, 7번째 라운드는 K_2가 라운드 키 값이 되고, 8번째 라운드는 K_3가 라운드 키 값이 되고, 9번째 라운드는 K_2가 라운드 키 값이 된다.

<15> 상기 복수의 CSPN부(220, 230)는 상기 라운드 키 연산부(210)의 출력을 입력받아서 암호화한다. 제 1 CSPN부(220)는 상기 라운드 키 연산부(210)의 출력을 입력변수 및 컨트롤 변수로 하여 소정의 암호화 연산을 수행한다. 제 2 CSPN부(230)는 상기 라운드 키 연산부(210)의 출력을 입력변수로 하고, 상기 제 1 CSPN부(220)의 입력변수와 제 1 CSPN부(220)의 출력이 비트확장된 값을 컨트롤 변수로 하여 암호화 연산을 수행한다.

<16> 상기 비트 확장부(240)는 상기 제 1 CSPN부(220)의 입력변수 및 출력의 비트를 확장시킨다. 비트 확장부(240)는 상기 제 1 CSPN부(220) 입력변수 및 출력을 입력받아서 비트가 확장된 조절 벡터를 생성한다. 즉, 비트 확장부는 2번의 비트 확장 연산을 수행한다. 비트 확장부(240)는 16비트의 제 1 CSPN부(220) 입력변수 및 출력을 입력받아서 96비트 조절 벡터를 생성한다. 도 4는 도 2의 비트 확장부에서 출력되는 조절 벡터의 실시예를 보인 도면이다. 도시된 바와 같이, 비트 확장부(240)는 제 1 CSPN부(220) 입력변수를 입력받아서 (V1, Z1, V2, Z2, V3, Z3)의 조절 벡터를 생성한다. V1은 제 1 CSPN부(220) 입력변수에 대응되고, Z1은 제 1 CSPN부(220) 입력변수를 우측으로 2비트만큼 순환이동한 값에 대응되고, V2는 제 1 CSPN부(220) 입력변수를 우측으로 6비트만큼 순환이동한 값에 대응되고, Z2는 제 1 CSPN부(220) 입력변수를 우측으로 8비트만큼 순환이동한 값에 대응되고, V3는 제 1 CSPN부(220) 입력변수를 우측으로 10비트만큼 순환이동한 값에 대응되고, Z3는 제 1 CSPN부(220) 입력변수를 우측으로 12비트만큼 순환이동한 값에 대응된다.

<17> 도 5는 도 2의 제 1 CSPN부를 상세히 나타낸 도면이다. 도시된 바와 같이, 제 1 CSPN부(220)는 제 1 데이터 의존 연산부(500), SPN부(510), 역 SPN부(520) 및 제 2 데이터 의존 연산부(530)를 포함한다.

<18> 제 1 데이터 의존 연산부(500)는 상기 라운드 키 연산부의 출력을 비트 분할하여 상기 각 비트 분할된 라운드

키 연산부의 출력을 입력변수와 컨트롤 변수로 하여 연산을 수행한다. 제 1 데이터 의존 연산부(500)는 16비트의 변수를 입력받고, 16비트의 컨트롤 변수를 이용하여 소정의 연산을 수행한다. 도 6은 도 5의 제 1 데이터 의존 연산부를 상세히 나타낸 도면이다. 도시된 바와 같이, 제 1 데이터 의존 연산부(500)는 복수의 F2/2를 포함한다. 즉, 제 1 F2/2는 각 1비트의 x1, x2를 입력받아서 v1, z1을 컨트롤 변수로 하여 y1, y2를 출력하고, 제 2 F2/2는 1비트의 x3, x4를 입력받아서 v2, z2를 컨트롤 변수로 하여 y3, y4를 출력한다. 제 1 데이터 의존 연산부(500)는 8개의 F2/2를 포함한다.

<19> 상기 각 F2/2는 x1, x2의 변수를 입력받아서 v, z의 컨트롤 변수를 이용하여 제 1 출력 y1, 제 2 출력 y2를 출력하며, 연산은 하기 수학식 1과 같다.

수학식 1

<20> 상기 수학식 1에서 제 1 출력인 y1은 vzx2와 vx2와 vx1과 zx1과 z와 x2 각각을 순서대로 배타적 논리합의 연산을 한 결과이다. 제 2 출력인 y2는 vzx1과 vz와 vx2와 zx1과 zx2와 x1 각각을 순서대로 배타적 논리합의 연산을 한 결과이다.

<21> 상기 SPN부(510)는 상기 제 1 데이터 의존 연산부(500)의 출력을 입력받아서 소정의 연산을 수행한다. 도 7은 도 5의 SPN부를 상세하게 나타낸 도면이다. 도시된 바와 같이, 상기 SPN부(510)는 복수의 치환부와 S 박스부를 포함한다. 상기 제 1 데이터 의존 연산부(500)의 출력은 제 1 치환부(700)로 입력된다. 제 1 치환부(700)는 상기 제 1 데이터 의존 연산부(500)에서 출력된 값의 비트들을 치환한다.

<22> 제 1 치환부(700)로 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 10번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 3번째 비트로 출력되고, 4번째 비트는 12번째 비트와 상호간에 치환되어 출력되고, 5번째 비트는 5번째 비트로 출력되고, 6번째 비트는 14번째 비트와 상호간에 치환되어 출력되고, 7번째 비트는 7번째 비트로 출력되고, 8번째 비트는 16번째 비트와 상호간에 치환되어 출력되고, 9번째 비트는 9번째 비트로 출력되고, 11번째 비트는 11번째 비트로 출력되고, 13번째 비트는 13번째 비트로 출력되고, 15번째 비트는 15번째 비트로 출력된다.

<23> 제 0 내지 제 3 S 박스부(702, …708)는 상기 제 1 치환부(700)에서 상호간에 비트가 치환되어 출력된 값을 특정 값으로 매칭시킨다. 제 0 내지 제 3 S 박스부(702, …708) 각각은 상기 제 1 치환부(700)로부터 각 4개의 4비트 출력값들을 입력받는다. 제 0 S박스부(702)는 제 1 치환부(700)의 출력 중에서 1번째 비트부터 4번째 비트의 출력을 입력받고, 제 1 S박스부(704)는 제 1 치환부(700)의 출력 중에서 5번째 비트부터 8번째 비트까지의 출력을 입력받고, 제 2 S박스부(706)는 제 1 치환부(700)의 출력 중에서 9번째 비트부터 12번째 비트까지의 출력을 입력받고, 제 3 S박스부(708)는 제 1 치환부(700)의 출력 중에서 13번째 비트부터 16번째 비트까지의 출력을 입력받는다. 각 S 박스부(702, …708)는 도 8에 도시되어 있는 값으로 입력받은 값을 매칭시킨다. 그리고 0 S박스부(702)가 출력하는 값이 상위 비트의 값이 되며, 3 S 박스부(708)가 출력하는 값이 하위 비트의 값이 된다.

<24> 제 2 치환부(710)는 상기 제 0 내지 제 3 S박스부(702, …708)의 출력을 입력받아서 상호 비트간에 치환한다. 제 2 치환부(710)로 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 5번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 4번째 비트는 13번째 비트와 상호간에 치환되어 출력되고, 6번째 비트는 6번째 비트로 출력되고, 7번째 비트는 10번째 비트와 상호간에 치환되어 출력되고, 8번째 비트는 14번째 비트와 상호간에 치환되어 출력되고, 11번째 비트는 11번째 비트로 출력되고, 12번째 비트는 15번째 비트와 상호간에 치환되어 출력되고, 16번째 비트는 16번째 비트로 출력된다.

<25> 상기 제 4 내지 제 7 S 박스부(712, …718)는 상기 제 2 치환부(710)에서 비트가 치환되어 출력된 값을 특정 값으로 매칭시킨다. 제 4 내지 제 7 S 박스부(712, …718) 각각은 상기 제 2 치환부(710)로부터 각 4개의 4비트 출력값들을 입력받는다. 제 4 S박스부(712)는 제 2 치환부(710)의 출력 중에서 1번째 비트부터 4번째 비트의 출력을 입력받고, 제 5 S박스부(714)는 제 2 치환부(710)의 출력 중에서 5번째 비트부터 8번째 비트까지의 출력을 입력받고, 제 6 S박스부(716)는 제 2 치환부(710)의 출력 중에서 9번째 비트부터 12번째 비트까지의 출력을 입력받고, 제 7 S박스부(718)는 제 2 치환부(710)의 출력 중에서 13번째 비트부터 16번째 비트까지의 출력을 입력받는다. 각 S 박스부(712, …718)는 도 8에 도시되어 있는 값으로 매칭된다.

<26> 역 SPN부(520)는 상기 라운드 키 연산부(210)를 비트 분할한 출력을 입력받는다. 상기 역 SPN부(520)는 상기 SPN부(510)를 역으로 연산을 수행하는 것과 동일하다. 즉, 제 4 내지 제 7 S박스부가 상기 라운드 키 연산부의 출력을 입력받아 매칭된 값을 출력하고, 상기 출력값을 제 2 치환부가 입력받아서 상호 비트간의 치환을 수행하

고, 제 0 내지 제 3 S박스부는 상기 제 2 치환부가 출력한 값을 입력받아 기 설정된 매칭된 값을 출력하고, 상기 출력값을 제 1 치환부가 입력받아서 상호 비트간의 치환을 수행한다.

<27> 제 2 데이터 의존 연산부(530)는 상기 역 SPN부(520)의 출력을 입력받고, 상기 SPN부(510)의 출력을 컨트롤 변수로 하여 소정의 연산을 수행한다. 상기 제 2 데이터 의존 연산부(530)에서의 연산은 상기 제 1 데이터 의존 연산부(500)에서의 연산과 동일하므로 상세한 설명은 생략하기로 한다.

<28> 제 1 CSPN부(220)의 출력은 상기 SPN부(510)의 출력과 제 2 데이터 의존 연산부(530)의 출력의 비트덧셈이다. 즉, 제 1 CSPN부(220)의 출력은 상기 SPN부(510)의 출력을 상위 비트로 하고, 제 2 데이터 의존 연산부(530)의 출력을 하위 비트로 한다.

<29> 도 9는 도 2의 제 2 CSPN부를 상세하게 나타낸 도면이다. 도시된 바와 같이, 상기 제 2 CSPN부는 제 3 데이터 의존 연산부(900), 제 3 치환부(910) 및 제 4 데이터 의존 연산부(920)를 포함한다.

<30> 상기 제 3 데이터 의존 연산부(900)는 상기 라운드 키 연산부(210)의 출력을 입력변수로 하고, 상기 비트 확장부(240)의 출력을 컨트롤 변수로 하여 소정의 연산을 수행한다. 제 3 데이터 의존 연산부(900)는 32비트를 입력변수로 하고, 96비트를 출력변수로 한다. 제 3 데이터 의존 연산부(900)는 F2/2를 반복적으로 중복 사용한다.

<31> 제 3 데이터 의존 연산부는 도 10에 도시한 바와 같이 복수의 F8/24를 포함한다. 도시된 바와 같이, 1번째 비트부터 8번째 비트는 제 1 F8/24로 입력되고, 9번째 비트부터 16번째 비트는 제 2 F8/24로 입력되고, 17번째 비트부터 24번째 비트는 제 3 F8/24로 입력되고, 제 25번째 비트부터 32번째 비트는 제 4 F8/24로 입력된다.

<32> 도 11은 도 10의 F8/24를 상세히 나타낸 도면이다. 도시된 바와 같이, F8/24는 3열의 F2/2를 포함하며, 제 1 열의 F2/2는 상기 라운드 키 연산부의 출력을 입력받고, 제 2 열의 F2/2는 제 1열의 F2/2의 출력을 입력받고, 제 3 열의 F2/2는 제 2열의 F2/2의 출력을 입력받는다. 각 열에는 복수의 F2/2가 포함된다.

<33> 제 1열 복수의 F2/2는 상기 라운드 키 연산부의 출력을 입력받고, V1, Z1을 컨트롤 변수로 하여 상기 수학식 1의 연산을 수행한다. 제 1열 1번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 3번째 F2/2로 입력된다. 제 1열 2번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 3번째 F2/2로 입력된다. 제 1열 3번째 F2/2의 제 1 출력은 제 2열 2번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력된다. 제 1열 4번째 F2/2의 제 1 출력은 제 2열 2번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력된다.

<34> 제 2열 복수의 F2/2는 상기 제 1열 복수의 F2/2의 출력을 입력받고, V2, Z2를 컨트롤 변수로 하여 연산을 수행한다. 제 2열 1번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력된다. 제 2열 2번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력된다. 제 2열 3번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력된다. 제 2열 4번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력된다.

<35> 제 3열 복수의 F2/2는 상기 제 2열 복수의 F2/2의 출력을 입력받고, V3, Z3를 컨트롤 변수로 하여 연산을 수행한다. 상기 제 3열 복수의 F2/2의 출력은 F8/24의 출력이 된다.

<36> 제 3 치환부(910)는 상기 제 3 데이터 의존 연산부(900)에서 출력된 값의 비트들을 상호간에 치환한다. 제 3 치환부(910)로 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 17번째 비트와 상호간에 치환되어 출력되고, 4번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 5번째 비트는 5번째 비트로 출력되고, 6번째 비트는 13번째 비트와 상호간에 치환되어 출력되고, 7번째 비트는 21번째와 상호간에 치환되어 출력되고, 8번째 비트는 29번째 비트와 상호간에 치환되어 출력되고, 10번째 비트는 10번째 비트로 출력되고, 11번째 비트는 18번째 비트와 상호간에 치환되어 출력되고, 12번째 비트는 26번째 비트와 상호간에 치환되어 출력되고, 14번째 비트는 14번째 비트로 출력되고, 15번째 비트는 22번째 비트로 출력되고, 16번째 비트는 30번째 비트와 상호간에 치환되어 출력되고, 19번째 비트는 19번째 비트로 출력되고, 20번째 비트는 27번째 비트와 상호간에 치환되어 출력되고, 23번째 비트는 23번째 비트로 출력되고, 24번째 비트는 31번째 비트와 상호간에 치환되어 출력되고, 28번째 비트는 28번째 비트로 출력되고, 32번째 비트는 32번째 비트로 출력된다.

<37> 상기 제 4 데이터 의존 연산부(920)는 상기 제 3 치환부(910)의 32비트 출력을 입력변수로 하고, 상기 SPN부(510)의 출력을 입력변수로 하는 비트 확장부(240)의 96비트 출력을 컨트롤 변수로 하여 소정의 연산을 수행한

다.

<38> 상기 제 4 데이터 의존 연산부(920)는 도 12와 같이 복수의 $F^{-1}8/24$을 포함한다. 도시된 바와 같이, 1번째 비트부터 8번째 비트는 제 1 $F^{-1}8/24$로 입력되고, 9번째 비트부터 16번째 비트는 제 2 $F^{-1}8/24$로 입력되고, 17번째 비트부터 24번째 비트는 제 3 $F^{-1}8/24$로 입력되고, 25번째 비트부터 32번째 비트는 제 4 $F^{-1}8/24$로 입력된다.

<39> 도 13은 $F^{-1}8/24$를 상세히 나타낸 도면이다. 도시된 바와 같이, $F^{-1}8/24$는 3열의 F2/2를 포함하며, 제 1열의 F2/2는 상기 제 3 치환부(910)의 출력을 입력받고, 제 2 열의 F2/2는 제 1열의 F2/2의 출력을 입력받고, 제 3열의 F2/2는 제 2열의 F2/2의 출력을 입력받는다. 각 열에는 복수의 F2/2가 포함된다.

<40> 제 1열 복수의 F2/2는 상기 제 3 치환부(910)의 출력을 입력받고, V3, Z3를 컨트롤 변수로 하여 상기 수학식 1의 연산을 수행한다. 제 1열 1번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 2번째 F2/2로 입력된다. 제 1열 2번째 F2/2의 제 1 출력은 제 2열 1번째 F2/2로 입력되고, 제 2 출력은 제 2열 2째 F2/2로 입력된다. 제 1열 3번째 F2/2의 제 1 출력은 제 2열 3번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력된다. 제 1열 4번째 F2/2의 제 1 출력은 제 2열 3번째 F2/2로 입력되고, 제 2 출력은 제 2열 4번째 F2/2로 입력된다.

<41> 제 2열 복수의 F2/2는 상기 제 1열 복수의 F2/2의 출력을 입력받고, V2, Z2를 컨트롤 변수로 하여 연산을 수행한다. 제 2열 1번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력된다. 제 2열 2번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력된다. 제 2열 3번째 F2/2의 제 1 출력은 제 3열 1번째 F2/2로 입력되고, 제 2 출력은 제 3열 2번째 F2/2로 입력된다. 제 2열 4번째 F2/2의 제 1 출력은 제 3열 3번째 F2/2로 입력되고, 제 2 출력은 제 3열 4번째 F2/2로 입력된다.

<42> 제 3열 복수의 F2/2는 상기 제 2열 복수의 F2/2의 출력을 입력받고, V1, Z1를 컨트롤 변수로 하여 연산을 수행한다. 상기 제 3열 복수의 F2/2의 출력은 $F^{-1}8/24$의 출력이 된다.

<43> 제 2 CSPN부(230)의 출력은 상기 제 4 데이터 의존 연산부(920)의 출력이다.

<44> 도 14는 도 1의 라운드부의 다른 실시예를 나타낸 도면이다. 도시된 바와 같이, 라운드부는 비트 분할부(1400), 라운드 키 연산부(1410), 복수의 CSPN부(1420, 1430) 및 비트 확장부(1440)를 포함한다.

<45> 상기 비트 분할부(1400)는 외부로부터 입력된 평문을 특정 비트를 기준으로 복수의 서브평문으로 분할한다. 비트 분할부(1400)에 대한 상세한 설명은 상기 일실시예에서 설명하였으므로 이하에서는 생략하기로 한다.

<46> 상기 라운드 키 연산부(1410)는 상기 비트 분할부(1400)에서 출력된 각 서브평문과 기 설정된 라운드키를 이용하여 연산한다. 라운드 키 연산부(1410)는 현재 평문이 전체 라운드에서 몇 번째 라운드에 해당하는지 및 서브평문이 평문에서 어느 위치에 있는지에 따라 상이한 라운드키를 서브평문과 연산한다. 상기 라운드 키 연산부(1410)는 상기 라운드 키와 서브평문과의 배타적 논리합(Exclusive OR) 연산을 수행한다. 도 15는 암호화 모듈에 기 설정된 라운드 키의 값이다. 도시된 바와 같이, 상기 K_1 내지 K_4는 256비트의 비밀키 K를 4개의 64비트로 분할한 부분키들이다. 상기 라운드 키 값은 현재 라운드가 몇 번째 라운드인지에 따라서 상이한 값을 갖는다. 1번째 라운드는 K_1이 라운드 키 값이 되고, 2번째 라운드는 K_2가 라운드 키 값이 되며, 3번째 라운드는 K_3가 라운드 키 값이 되고, 4번째 라운드는 K_4가 라운드 키 값이 되며, 5번째 라운드는 K_2이 라운드 키 값이 되고, 6번째 라운드는 K_1가 라운드 키 값이 되며, 7번째 라운드는 K_3가 라운드 키 값이 되고, 8번째 라운드는 K_4가 라운드 키 값이 되고, 9번째 라운드는 K_3가 라운드 키 값이 되며, 10번째 라운드는 K_2가 라운드 키 값이 되고, 11번째 라운드는 K_1이 라운드 키 값이 된다.

<47> 상기 복수의 CSPN부(1420, 1430)는 상기 라운드 키 연산부의 출력을 입력받아서 암호화한다. 제 3 CSPN부(1420)는 상기 라운드 키 연산부(1410)의 출력을 입력변수 및 컨트롤 변수로 하여 소정의 암호화 연산을 수행한다. 제 4 CSPN부(1430)는 상기 라운드 키 연산부(1410)의 출력을 입력변수로 하고, 상기 제 3 CSPN부(1420)의 입력변수와 제 3 CSPN부(1420)의 출력이 비트확장된 값을 컨트롤 변수로 하여 암호화 연산을 수행한다.

<48> 상기 비트 확장부(1440)는 상기 제 3 CSPN부(1420)의 입력변수 및 출력의 비트를 확장시킨다. 비트 확장부(1440)는 상기 제 3 CSPN부(1420) 입력변수 및 출력을 입력받아서 비트가 확장된 조절 벡터를 생성한다. 즉, 비

트 확장부는 2번의 비트 확장 연산을 수행한다. 비트 확장부(1440)는 32비트의 제 3 CSPN부(1420) 입력변수 및 출력을 입력받아서 192비트 조절벡터를 생성한다. 도 16은 도 14의 비트 확장부에서 출력되는 조절 벡터의 실시예를 보인 도면이다. 도시된 바와 같이, 비트 확장부(1440)는 제 3 CSPN부(1420) 입력변수를 입력받아서 (V4, Z4, V5, Z5, V6, Z6)의 조절벡터를 생성한다. V4는 제 3 CSPN부(1420) 입력변수에 대응되고, Z4은 제 3 CSPN부(1420) 입력변수를 우측으로 5비트만큼 순환이동한 값에 대응되고, V5는 제 3 CSPN부(1420) 입력변수를 우측으로 10비트만큼 순환이동한 값에 대응되고, Z5는 제 3 CSPN부(1420) 입력변수를 우측으로 15비트만큼 순환이동한 값에 대응되고, V6는 제 3 CSPN부(1420) 입력변수를 우측으로 20비트만큼 순환이동한 값에 대응되고, Z6는 제 3 CSPN부(1420) 입력변수를 우측으로 25비트만큼 순환이동한 값에 대응된다.

<49> 도 17은 도 14의 제 3 CSPN부를 상세히 나타낸 도면이다. 도시된 바와 같이, 제 3 CSPN부(1420)는 제 5 데이터 의존 연산부(1700), SPN부(1710), 역 SPN부(1720) 및 제 6 데이터 의존 연산부(1730)를 포함한다.

<50> 제 5 데이터 의존 연산부(1700)는 상기 라운드 키 연산부(1410)의 출력을 비트 분할한 각 값을 입력변수와 컨트롤 변수로 하여 소정의 연산을 수행한다. 제 5 데이터 의존 연산부(1700)는 32비트의 변수를 입력받고, 32비트의 컨트롤 변수를 이용하여 소정의 연산을 수행한다. 제 5 데이터 의존 연산부(1700)는 복수의 F2/2를 포함한다. 상기 F2/2는 하기 수학식 2와 같다.

수학식 2

<51> 상기 각 F2/2는 x1, x2의 변수를 입력받아서 v, z의 컨트롤 변수를 이용하여 제 1 출력 y1, 제 2 출력 y2를 출력하며, 상기 수학식 2에서 제 1 출력인 y1는 vzx1과 vzx2와 vx1과 vx2와 zx1과 zx2와 z와 v와 x2 각각을 순서대로 배타적 논리합의 연산을 한 결과이다. 제 2 출력인 y2는 vzx1과 vzx2와 vz와 vx1과 vx2와 zx1과 zx2와 x1 각각을 순서대로 배타적 논리합의 연산을 한 결과이다.

<52> 상기 SPN부(1710)는 상기 제 5 데이터 의존 연산부(1700)의 출력을 입력받아서 소정의 연산을 수행한다. 상기 SPN부(1710)는 도 18에 도시된 바와 같이 복수의 치환부와 S 박스부를 포함한다. 상기 제 5 데이터 의존 연산부(1700)의 출력은 제 4 치환부(1800)로 입력된다. 제 4 치환부(1800)는 상기 출력된 값의 비트들을 치환한다.

<53> 제 4 치환부(1800)로 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 18번째 비트와 상호간에 치환되어 출력되며, 3번째 비트는 3번째 비트로 출력되고, 4번째 비트는 20번째 비트와 상호간에 치환되어 출력되며, 5번째 비트는 5번째 비트로 출력되고, 6번째 비트는 22번째 비트와 상호간에 치환되어 출력되며, 7번째 비트는 7번째 비트로 상호간에 치환되어 출력되고, 8번째 비트는 24번째 비트와 상호간에 치환되어 출력되며, 9번째 비트는 9번째 비트로 출력되고, 10번째 비트는 26번째 비트와 상호간에 치환되어 출력되며, 11번째 비트는 11번째 비트로 출력되고, 12번째 비트는 28번째 비트와 상호간에 치환되어 출력되며, 13번째 비트는 13번째 비트로 출력되고, 14번째 비트는 30번째 비트와 상호간에 치환되어 출력되며, 15번째 비트는 15번째 비트로 출력되고, 16번째 비트는 32번째 비트와 상호간에 치환되어 출력되며, 17번째 비트는 17번째 비트로 출력되고, 19번째 비트는 19번째 비트로 출력되고, 21번째 비트는 21번째 비트로 출력되고, 23번째 비트는 23번째 비트로 출력되고, 25번째 비트는 25번째 비트로 출력되고, 27번째 비트는 27번째 비트로 출력되고, 29번째 비트는 29번째 비트로 출력되고, 31번째 비트는 31번째 비트로 출력된다.

<54> 제 8 내지 제 15 S 박스부(1810, …, 1840)는 상기 제 4 치환부(1800)에서 비트가 치환되어 출력된 값을 특정 값으로 매칭시킨다. 제 8 내지 제 15 S 박스부 (1810, …, 1840)각각은 상기 제 4 치환부(1800)로부터 각 4개의 4비트 출력값들을 입력받는다. 제 8 S박스부(1810)는 제 4 치환부(1800)의 출력 중에서 1번째 비트부터 4번째 비트의 출력을 입력받고, 제 9 S박스부(1820)는 제 4 치환부(1800)의 출력 중에서 5번째 비트부터 8번째 비트까지의 출력을 입력받고, 제 10 S박스부는 제 4 치환부(1800)의 출력 중에서 9번째 비트부터 12번째 비트까지의 출력을 입력받고, 제 11 S박스부는 제 4 치환부(1800)의 출력 중에서 13번째 비트부터 16번째 비트까지의 출력을 입력받고, 제 12 S 박스부는 제 4 치환부(1800)의 출력 중에서 17번째 비트부터 20번째 비트까지의 출력을 입력받고, 제 13 S 박스부는 제 4 치환부(1800)의 출력 중에서 21번째 비트부터 24번째 비트까지의 출력을 입력받고, 제 14 S 박스부(1830)는 제 4 치환부(1800)의 출력 중에서 25번째 비트부터 28번째 비트까지의 출력을 입력받고, 제 15 S 박스부(1840)는 제 4 치환부(1800)의 출력 중에서 29번째 비트부터 32번째 비트까지의 출력을 입력받는다. 각 S 박스부(1810, …, 1840)는 상기 도 8에 도시되어 있는 값으로 매칭된다.

<55> 제 5 치환부(1850)는 상기 제 8 내지 제 15 S박스(1810, …, 1840)의 출력을 입력받아서 상호 비트간에 치환한다. 제 5 치환부(1850)로 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 5번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 4번째 비트는 13번째 비트

와 상호간에 치환되어 출력되고, 6번째 비트는 6번째 비트로 출력되고, 7번째 비트는 10번째 비트와 상호간에 치환되어 출력되고, 8번째 비트는 14번째 비트와 상호간에 치환되어 출력되고, 11번째 비트는 11번째 비트로 출력되고, 12번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 16번째 비트는 16번째 비트로 출력되고, 17번째 비트는 17비트로 출력되고, 18번째 비트는 21번째 비트와 상호간에 치환되어 출력되고, 19번째 비트는 25번째 비트와 상호간에 치환되어 출력되고, 20번째 비트는 29번째 비트와 상호간에 치환되어 출력되고, 22번째 비트는 22번째 비트로 출력되고, 23번째 비트는 26번째 비트와 상호간에 치환되어 출력되고, 24번째 비트는 30번째 비트와 상호간에 치환되어 출력되고, 27번째 비트는 27번째 비트로 출력되고, 28번째 비트는 31번째 비트와 상호간에 치환되어 출력되고, 32번째 비트는 32번째 비트로 출력된다.

<56> 상기 역 제 8 내지 제 15 S 박스부(1860, …, 1890)는 상기 제 5 치환부(1850)에서 비트가 치환되어 출력된 값을 특정 값으로 매칭시킨다. 상기 제 5 치환부(1850)의 출력값을 복수개로 분할되어 출력된다. 상기 역 제 8 내지 제 15 S 박스부(1860, …, 1890) 각각은 상기 제 5 치환부(1850)로부터 각 4개의 4비트 출력값들을 입력받는다. 역 제 8 S박스부(1860)는 제 5 치환부(1850)의 출력중에서 1번째 비트부터 4번째 비트의 출력을 입력받고, 역 제 9 S박스부(1870)는 제 5 치환부(1850)의 출력 중에서 5번째 비트부터 8번째 비트까지의 출력을 입력받고, 역 제 10 S박스부는 제 5 치환부(1850)의 출력 중에서 9번째 비트부터 12번째 비트까지의 출력을 입력받고, 역 제 11 S박스부는 제 5 치환부(1850)의 출력 중에서 13번째 비트부터 16번째 비트까지의 출력을 입력받고, 역 제 12 S 박스부는 제 5 치환부(1850)의 출력 중에서 17번째 비트부터 20번째 비트까지의 출력을 입력받고, 역 제 13 S 박스부는 제 5 치환부(1850)의 출력 중에서 21번째 비트부터 24번째 비트까지의 출력을 입력받고, 역 제 14 S 박스부(1880)는 제 5 치환부(1850)의 출력 중에서 25번째 비트부터 28번째 비트까지의 출력을 입력받고, 역 제 15 S 박스부(1890)는 제 5 치환부(1850)의 출력 중에서 29번째 비트부터 32번째 비트까지의 출력을 입력받는다. 각 역 S 박스부(1860, …, 1890)들은 상기 S 박스부들이 값을 매칭하는 과정과 반대의 과정으로 값을 매칭한다.

<57> 역 SPN부(1720)는 상기 라운드 키 연산부의 출력를 비트 분할한 출력을 입력받는다. 상기 역 SPN부(1720)는 상기 SPN부(1710)를 역으로 연산을 수행하는 것과 동일하다.

<58> 제 6 데이터 의존 연산부(1730)는 상기 역 SPN부(1720)의 출력을 입력받고, 상기 SPN부(1710)의 출력을 컨트롤 변수로 하여 소정의 연산을 수행한다. 상기 제 6 데이터 의존 연산부(1730)에서의 연산은 상기 제 5 데이터 의존 연산부(1700)에서의 연산과 동일하므로 상세한 설명은 생략하기로 한다.

<59> 제 3 CSPN부(1420)의 출력은 상기 SPN부의 출력과 제 6 데이터 의존 연산부(1730)의 출력의 비트 덧셈이다. 즉, 제 3 CSPN부(1420)의 출력은 상기 SPN부(1710)의 출력을 상위 비트로 하고, 제 6 데이터 의존 연산부(1730)의 출력을 하위 비트로 한다.

<60> 도 19는 도 14의 제 4 CSPN부를 상세하게 나타낸 도면이다. 도시된 바와 같이, 상기 제 2 CSPN부(230)는 제 6 치환부(1900), 제 7 데이터 의존 연산부(1910), 제 7 치환부(1920), 제 8 데이터 의존 연산부(1930) 및 제 8 치환부(1940)를 포함한다.

<61> 상기 제 6 치환부(1900)는 상기 라운드 키 연산부의 출력을 입력받아, 상호간에 비트를 치환한다. 제 6 치환부(1900)로 입력되는 값에서 홀수번째 비트는 동일한 비트로 출력되고, 짝수번째 비트는 (2j, 2j+32)비트로 출력된다. 단, 상기 j는 1 이상 32 이하의 값이다.

<62> 상기 제 7 데이터 의존 연산부(1910)는 상기 제 6 치환부(1900)의 출력을 입력변수로 하고, 상기 비트 확장부(1440)의 출력을 컨트롤 변수로 하여 소정의 연산을 수행한다. 제 7 데이터 의존 연산부(1910)는 64비트를 입력변수로 하고, 192비트를 출력변수로 한다. 제 7 데이터 의존 연산부(1910)는 F2/2를 반복적으로 중복 사용한다.

<63> 제 7 데이터 의존 연산부(1910)는 도 20과 같이 복수의 F8/24를 포함한다. 도시된 바와 같이, 1번째 비트부터 8번째 비트는 제 1 F8/24로 입력되고, 9번째 비트부터 16번째 비트는 제 2 F8/24로 입력되고, 17번째 비트부터 24번째 비트는 제 3 F8/24로 입력되고, 제 25번째 비트부터 32번째 비트는 제 4 F8/24로 입력되고, 제 33번째 비트부터 40번째 비트는 제 5 F8/24로 입력되고, 제 41번째 비트부터 48번째 비트는 제 6 F8/24로 입력되고, 제 49번째 비트부터 56번째 비트는 제 7 F8/24로 입력되고, 제 57번째 비트부터 64번째 비트는 제 8 F8/24로 입력된다.

<64> 상기 제 7 치환부(1920)는 상기 제 7 데이터 의존 연산부(1910)에서 출력된 값의 비트들을 상호간에 치환한다. 제 7 치환부(1920)로 입력되는 값에서 1번째 비트는 1번째 비트로 출력되고, 2번째 비트는 9번째 비트와 상호간에 치환되어 출력되고, 3번째 비트는 17번째 비트로 출력되고, 4번째 비트는 25번째 비트와 상호간에 치환되어

출력되고, 5번째 비트는 33번째 비트와 상호간에 치환되어 출력되고, 6번째 비트는 41번째 비트와 상호간에 치환되어 출력되고, 7번째 비트는 49번째 비트와 상호간에 치환되어 출력되고, 8번째 비트는 57번째 비트와 상호간에 치환되어 출력되고, 10번째 비트는 10번째 비트로 출력되고, 11번째 비트는 18번째 비트와 상호간에 치환되어 출력되고, 12번째 비트는 26번째 비트와 상호간에 치환되어 출력되고, 15번째 비트는 50번째 비트와 상호간에 치환되어 출력되며, 16번째 비트는 58번째 비트와 상호간에 치환되어 출력되고, 19번째 비트는 19번째 비트로 출력되고, 20번째 비트는 27번째 비트와 상호간에 치환되어 출력되며, 21번째 비트는 35번째 비트와 상호간에 치환되어 출력되고, 22번째 비트는 43번째 비트와 상호간에 치환되어 출력되며, 23번째 비트는 51번째 비트와 상호간에 치환되어 출력되고, 24번째 비트는 59번째 비트와 상호간에 치환되어 출력되며, 28번째 비트는 28번째 비트로 출력되고, 29번째 비트는 36번째 비트와 상호간에 치환되어 출력되며, 30번째 비트는 44번째 비트와 상호간에 치환되어 출력되고, 31번째 비트는 52번째 비트와 상호간에 치환되어 출력되며, 32번째 비트는 60번째 비트와 상호간에 치환되어 출력되며, 37번째 비트는 37번째 비트로 출력되고, 38번째 비트는 45번째 비트와 상호간에 치환되어 출력되며, 39번째 비트는 53번째 비트와 상호간에 치환되어 출력되며, 40번째 비트는 61번째 비트와 상호간에 치환되어 출력되며, 46번째 비트는 46번째 비트로 출력되고, 47번째 비트는 54번째 비트와 상호간에 치환되어 출력되며, 48번째 비트는 62번째 비트와 상호간에 치환되어 출력되며, 55번째 비트는 55번째 비트로 출력되고, 56번째 비트는 63번째 비트와 상호간에 치환되어 출력되며, 64번째비트는 64번째비트로 출력된다.

<65> 상기 제 8 데이터 의존 연산부(1930)는 상기 제 7 치환부(1920)의 64비트출력을 입력변수로 하고, 상기 SPN부(1710)의 출력을 입력변수로 하는 비트 확장부(1440)의 출력을 컨트롤 변수로 하여 소정의 연산을 수행한다. 제 8 데이터 의존 연산부(1930)는 64비트를 입력변수로 하고, 192비트를 출력변수로 한다.

<66> 상기 제 8 데이터 의존 연산부(1930)는 도 21과 같이 복수의 $F^{-1}8/24$을 포함한다. 도시된 바와 같이, 1번째 비트부터 8번째 비트는 제 1 $F^{-1}8/24$로 입력되고, 9번째 비트부터 16번째 비트는 제 2 $F^{-1}8/24$로 입력되고, 17번째 비트부터 24번째 비트는 제 3 $F^{-1}8/24$로 입력되고, 25번째 비트부터 32번째 비트는 제 4 $F^{-1}8/24$로 입력되고, 33번째 비트부터 40번째 비트는 제 5 $F^{-1}8/24$로 입력되고, 41번째 비트부터 48번째 비트는 제 6 $F^{-1}8/24$로 입력되고, 49번째 비트부터 56번째 비트는 제 7 $F^{-1}8/24$로 입력되고, 57번째 비트부터 64번째 비트는 제 8 $F^{-1}8/24$로 입력된다.

<67> 상기 제 8 데이터 의존 연산부(1930)의 출력은 제 8 치환부(1940)가 입력받아서, 입력의 비트들 상호간에 치환을 하여 출력한다. 상기 제 8 치환부(1940)에서 비트가 치환되는 연산은 상기 제 6 치환부(1900)에서의 연산과 동일하다. 제 4 CSPN부(1430)의 출력은 상기 제 8 치환부(1940)의 출력이 된다.

도면의 간단한 설명

<68> 도 1은 본 발명의 데이터 의존 연산을 사용하는 암호화 모듈을 나타낸 도면.

<69> 도 2는 도 1의 라운드부의 일실시 예를 나타낸 도면.

<70> 도 3은 암호화 모듈에 기 설정된 라운드 키의 값을 나타낸 도면.

<71> 도 4는 도 2의 비트 확장부에서 출력되는 조절 벡터의 실시예를 보인 도면.

<72> 도 5는 도 2의 제 1 CSPN부를 상세히 나타낸 도면.

<73> 도 6은 도 5의 제 1 데이터 의존 연산부를 상세히 나타낸 도면.

<74> 도 7은 도 5의 SPN부를 상세히 나타낸 도면.

<75> 도 8은 S 박스부에서 입력받은 값이 매칭되는 관계를 나타낸 도면.

<76> 도 9는 도 2의 제 2 CSPN부를 상세히 나타낸 도면.

<77> 도 10은 도 9의 제 3 데이터 의존 연산부를 상세히 나타낸 도면.

<78> 도 11은 도 10의 F8/24를 상세히 나타낸 도면.

<79> 도 12는 도 9의 제 4 데이터 의존 연산부를 상세히 나타낸 도면.

<80> 도 13은 도 12의 $F^{-1}8/24$를 상세히 나타낸 도면.

<81> 도 14는 도 1의 라운드부의 다른 실시예를 나타낸 도면.

<82> 도 15는 암호화 모듈에 기 설정된 라운드 키의 값을 나타낸 도면.

<83> 도 16은 도 14의 비트 확장부에서 출력되는 조절 벡터의 실시예를 보인 도면.

<84> 도 17은 도 14의 제 3 CSPN부를 상세히 나타낸 도면.

<85> 도 18은 도 17의 SPN부를 상세히 나타낸 도면.

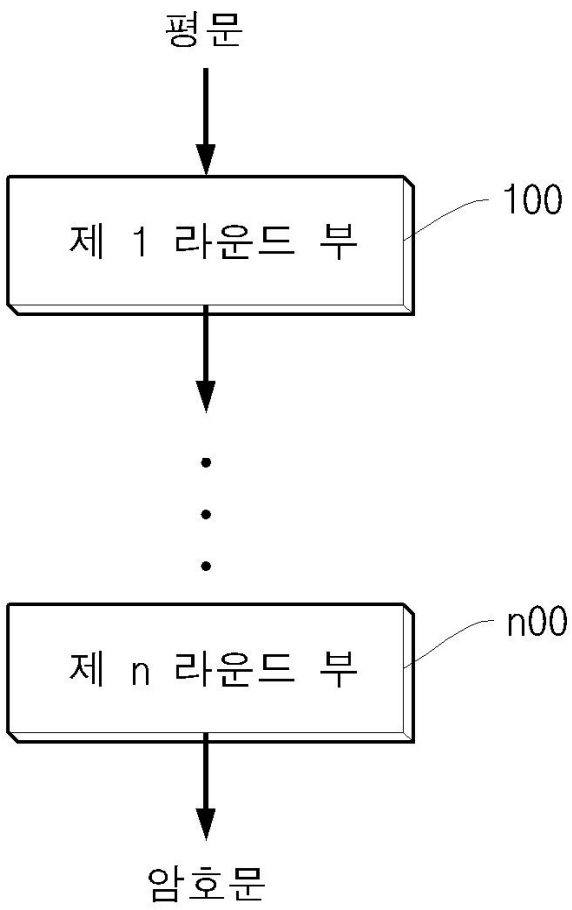
<86> 도 19는 도 14의 제 4 CSPN부를 상세히 나타낸 도면.

<87> 도 20은 도 19의 제 7 데이터 의존부를 상세히 나타낸 도면.

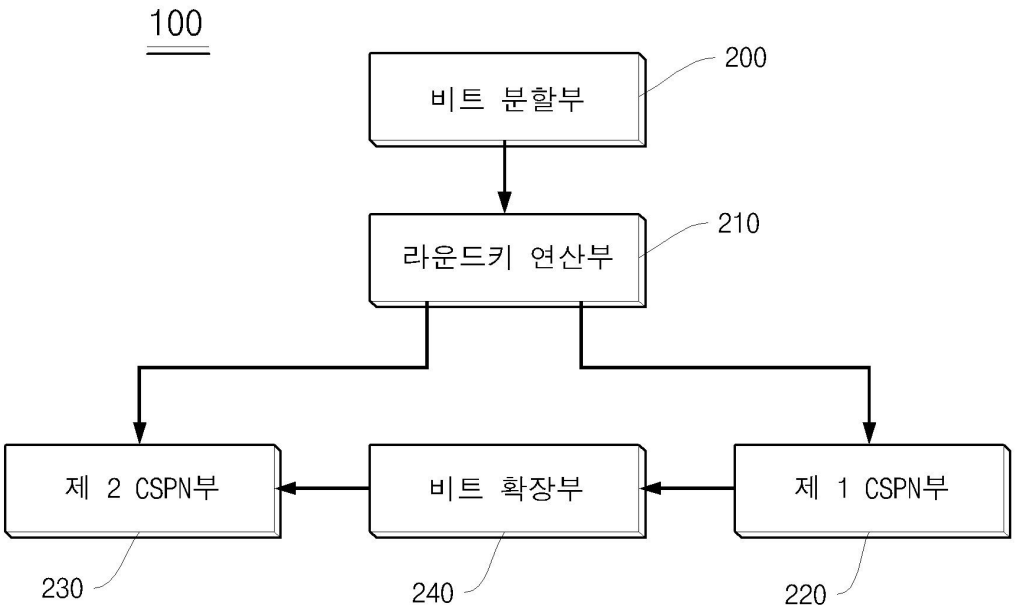
<88> 도 21은 도 19의 제 8 데이터 의존부를 상세히 나타낸 도면.

도면

도면1



도면2



도면3

라운드	1	2	3	4	5	6	7	8	9
Qj	k1	k2	k3	k4	k1	k4	k2	k3	k2

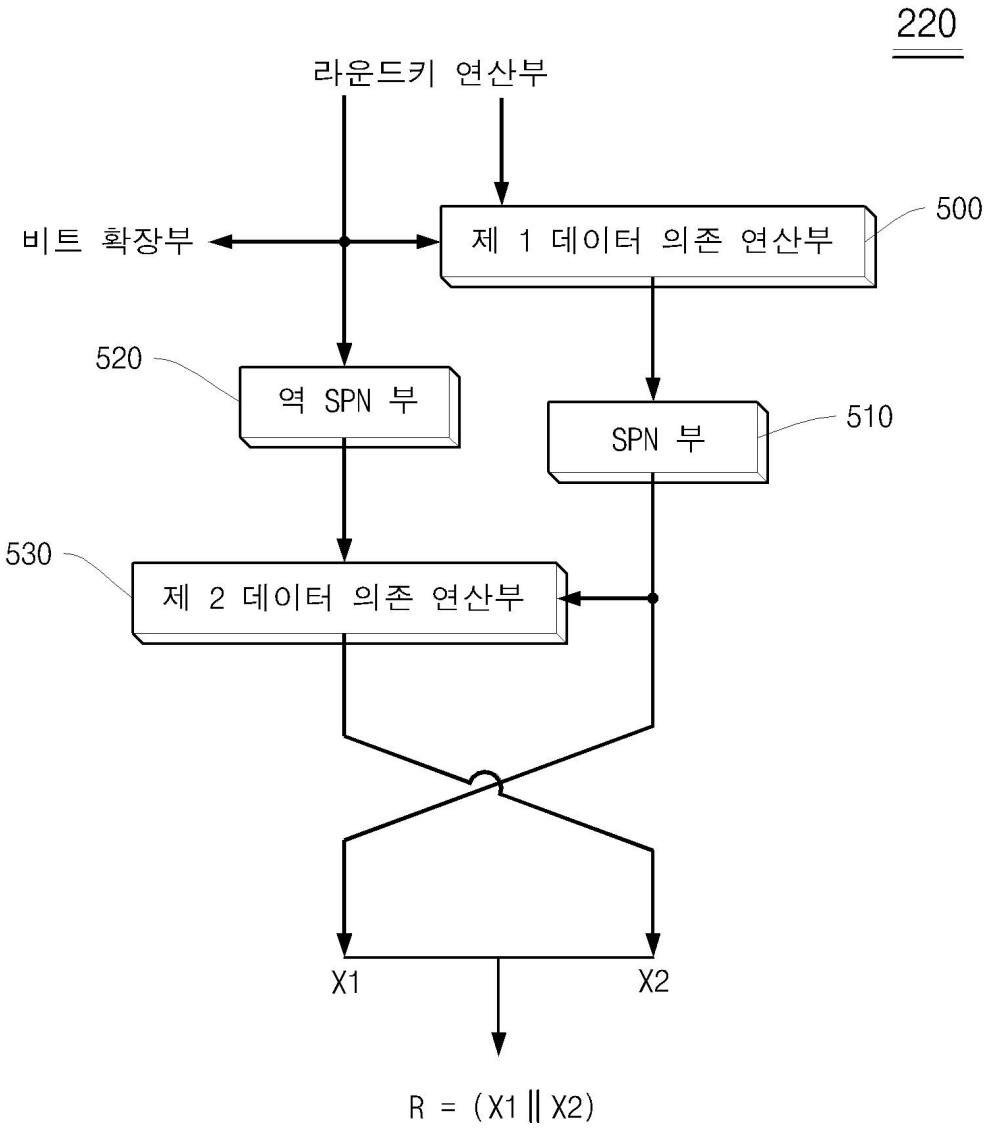
도면4

$$E(x) = V = (V1, Z1, V2, Z2, V3, Z3)$$

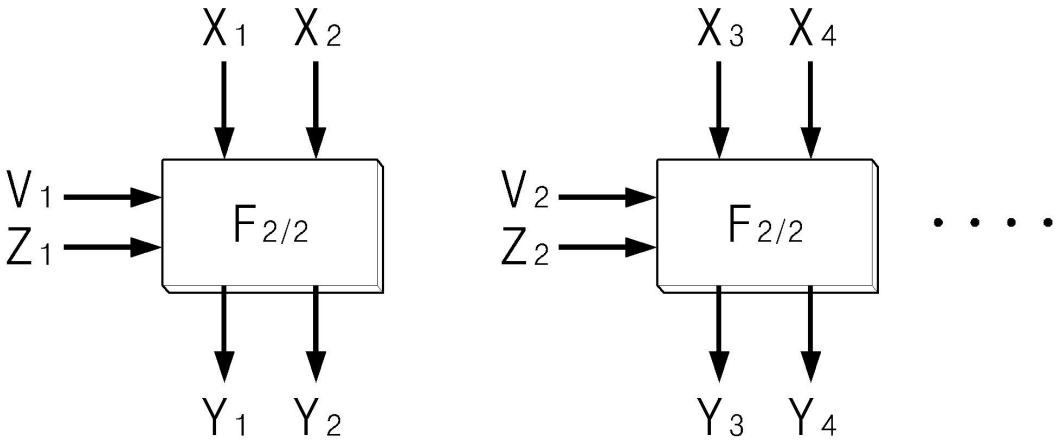
$$V1 = X, \quad Z1 = X^{\ggg 2}; \quad V2 = X^{\ggg 6}; \quad Z2 = X^{\ggg 8}$$

$$V3 = X^{\ggg 10}, \quad Z3 = X^{\ggg 12}$$

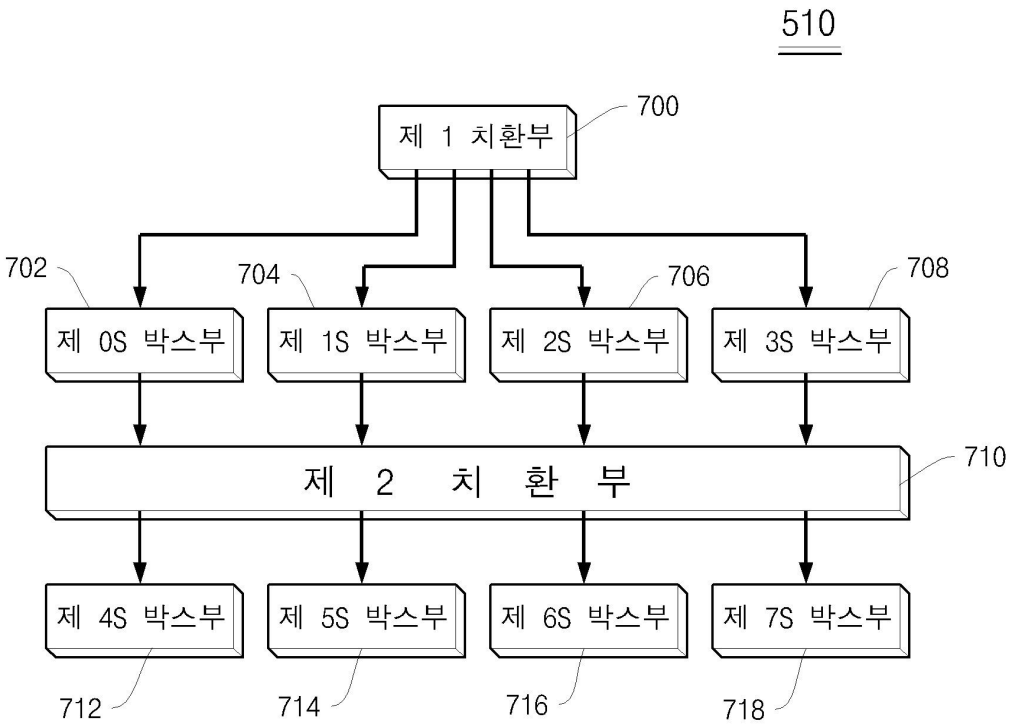
도면5


220
라운드키 연산부
비트 확장부
제 1 데이터 의존 연산부
500
520
역 SPN 부
SPN 부
510
530
제 2 데이터 의존 연산부
X1
X2
R = (X1 || X2)

도면6


X1
X2
V1
Z1
F2/2
Y1
Y2
X3
X4
V2
Z2
F2/2
Y3
Y4

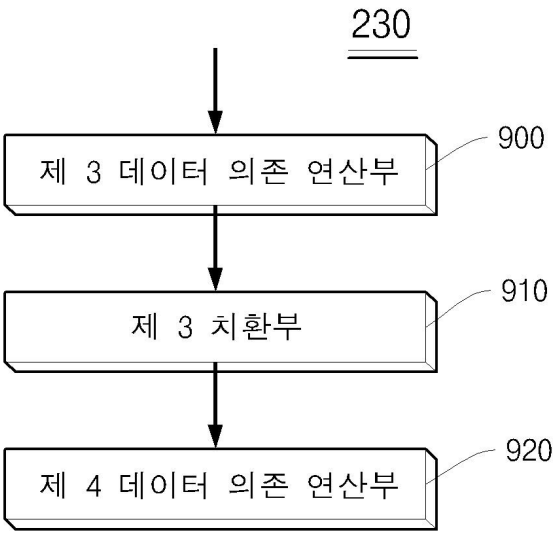
도면7



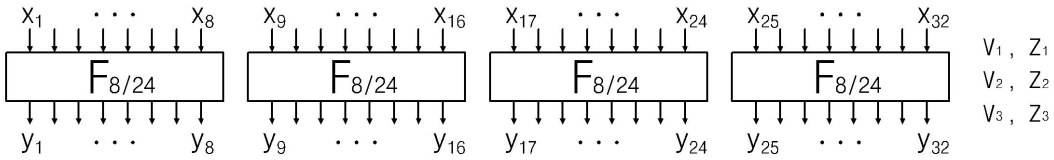
도면8

제 0S 박스부	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
제 1S 박스부	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
제 2S 박스부	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
제 3S 박스부	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
제 4S 박스부	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
제 5S 박스부	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
제 6S 박스부	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
제 7S 박스부	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2

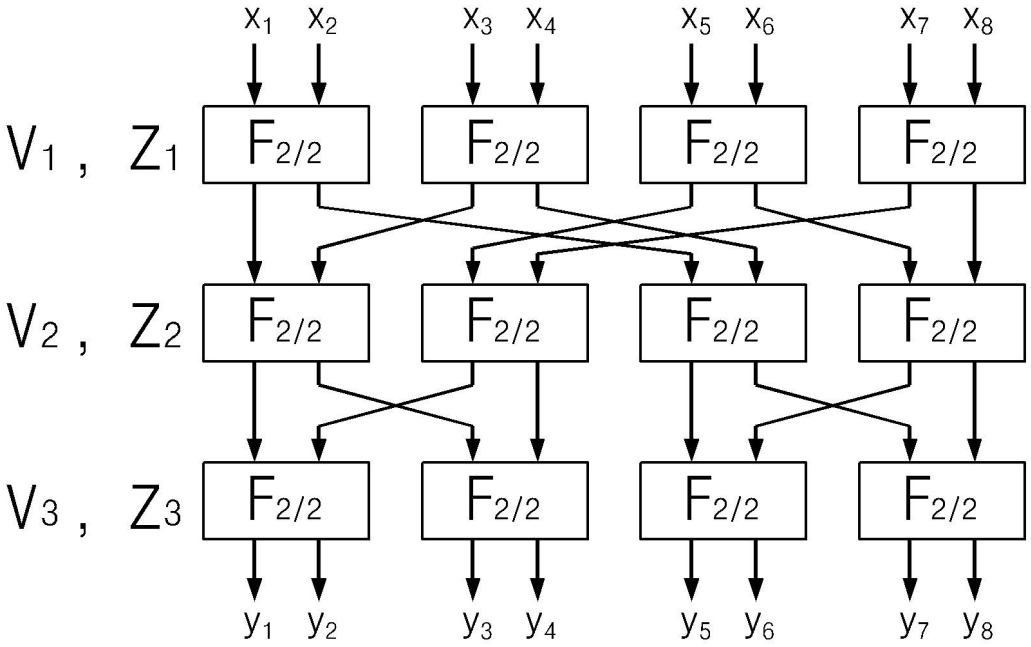
도면9



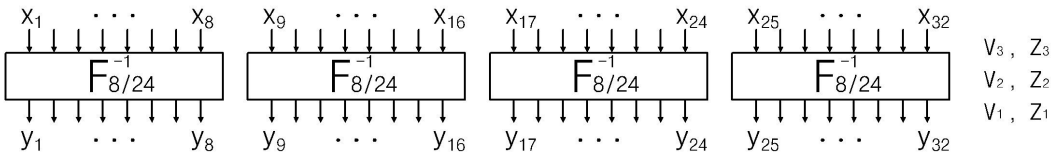
도면10



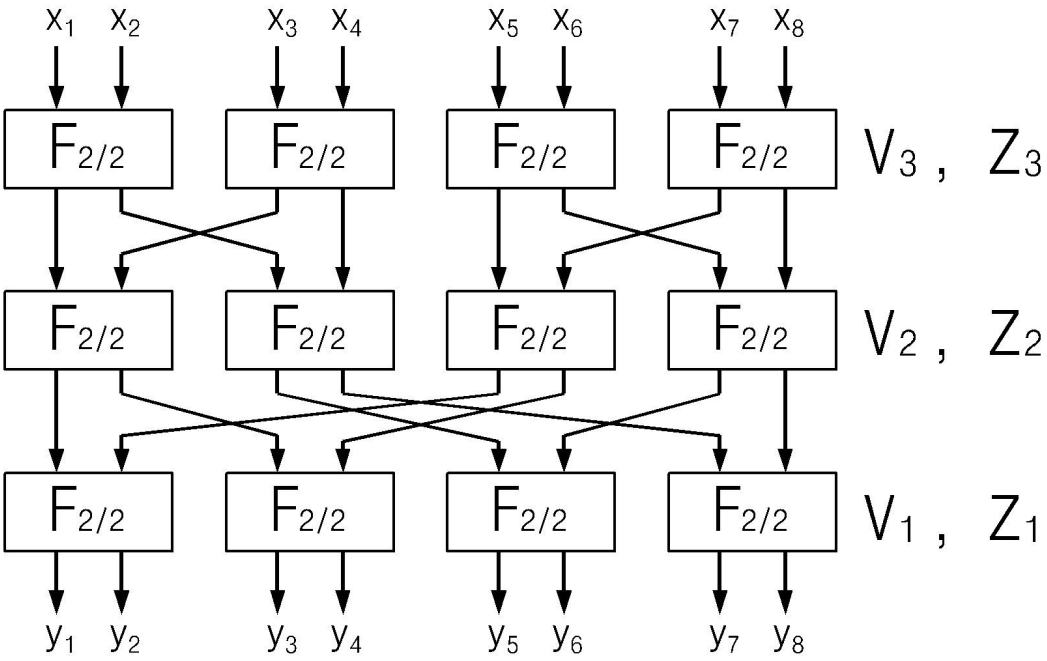
도면11



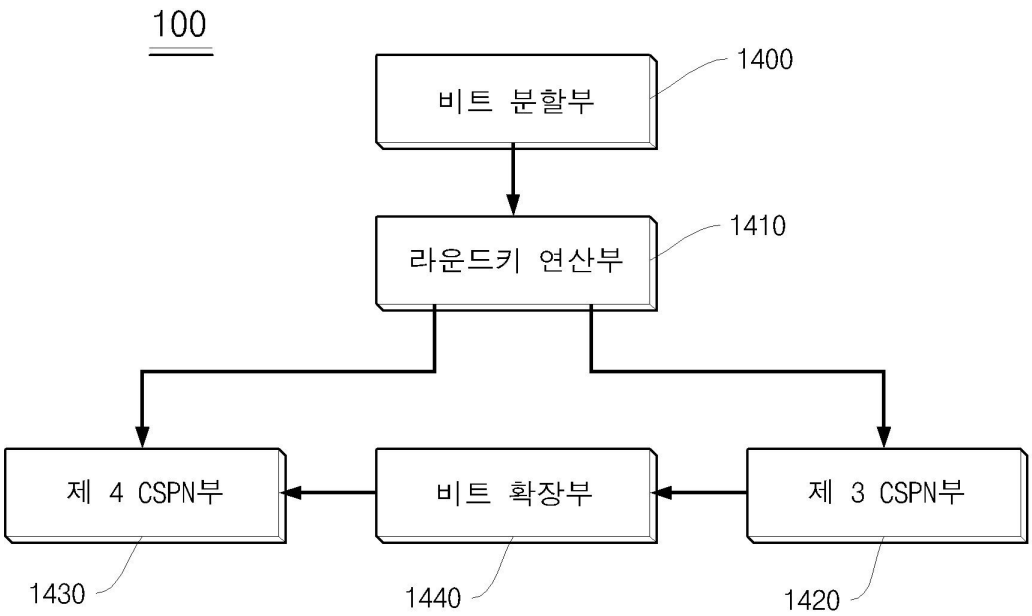
도면12



도면13



도면14



도면15

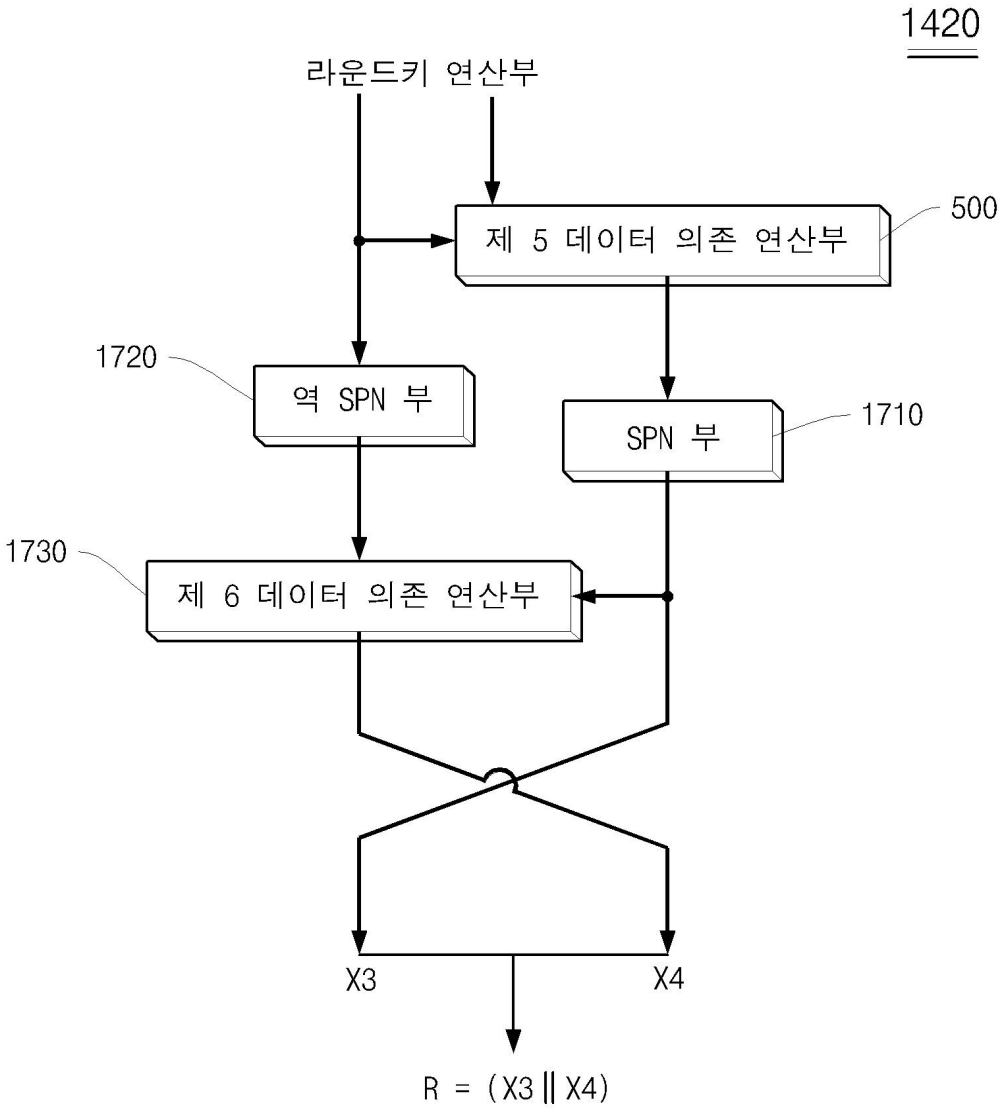
라운드	1	2	3	4	5	6	7	8	9	10	11
Qj	k1	k2	k3	k4	k2	k1	k3	k4	k3	k2	k1

도면16

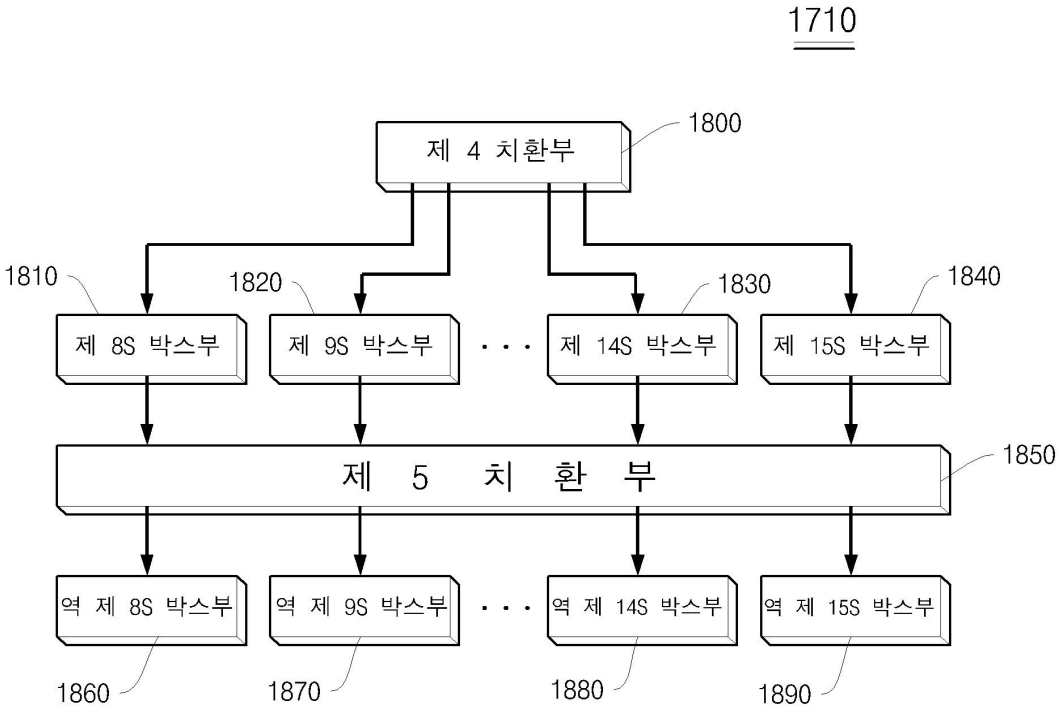
$$E(x) = V = (V4, Z4, V5, Z5, V6, Z6)$$

$$Vi = X^{\ggg 10(i-4)}, \quad Zi = X^{\ggg 10(i-3)-5} \quad (i = 4, 5, 6 \ldots)$$

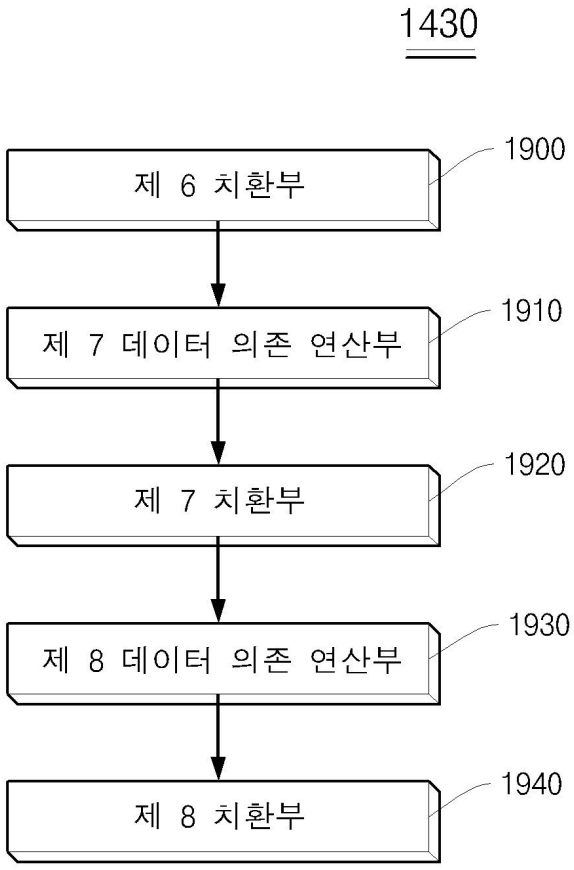
도면17


1420
라운드키 연산부
제 5 데이터 의존 연산부
500
1720
역 SPN 부
SPN 부
1710
1730
제 6 데이터 의존 연산부
X3
X4
R = (X3 || X4)

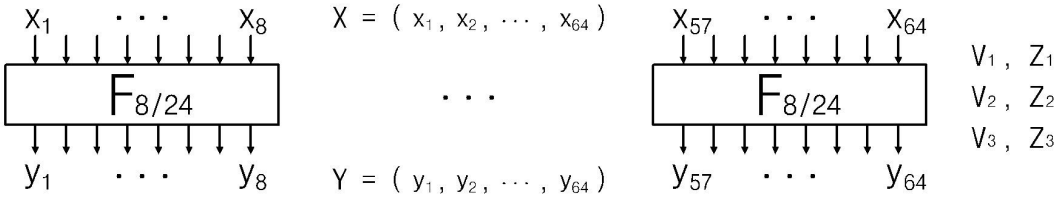
도면18


1710
제 4 치환부
1800
1810
1820
1830
1840
제 8S 박스부
제 9S 박스부
제 14S 박스부
제 15S 박스부
제 5 치 환 부
1850
역 제 8S 박스부
역 제 9S 박스부
역 제 14S 박스부
역 제 15S 박스부
1860
1870
1880
1890

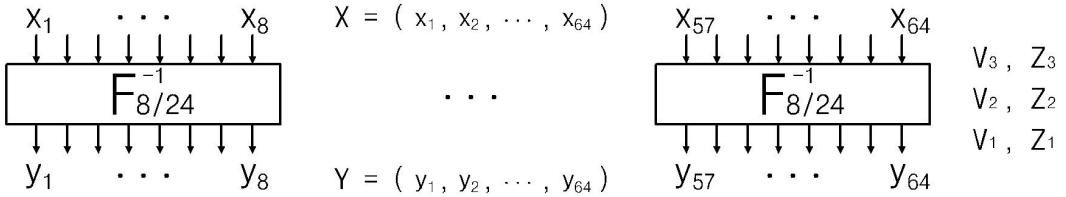
도면19


1430
제 6 치환부
1900
제 7 데이터 의존 연산부
1910
제 7 치환부
1920
제 8 데이터 의존 연산부
1930
제 8 치환부
1940

도면20


x1 · · · x8
F8/24
y1 · · · y8
X = (x1 , x2 , · · · , x64)
· · ·
Y = (y1 , y2 , · · · , y64)
x57 · · · x64
F8/24
y57 · · · y64
V1 , Z1
V2 , Z2
V3 , Z3

도면21


x1 · · · x8
F-1 8/24
y1 · · · y8
X = (x1 , x2 , · · · , x64)
· · ·
Y = (y1 , y2 , · · · , y64)
x57 · · · x64
F-1 8/24
y57 · · · y64
V3 , Z3
V2 , Z2
V1 , Z1