(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0069012
(43) 공개일자 2010년06월24일

(51) Int. Cl.
H04L 9/14 (2006.01) *H04L 9/28* (2006.01)
(21) 출원번호 10-2008-0127541
(22) 출원일자 2008년12월15일
심사청구일자 2008년12월15일

(71) 출원인
부경대학교 산학협력단
부산 남구 용당동 산 100번지 부경대학교내
(72) 발명자
김길호
부산 서구 초장동 110-2번지 8/2
김원희
부산광역시 남구 대연동 891-7 장미하우스 가-106
(뒷면에 계속)

전체 청구항 수 : 총 6 항

(54) 대칭단을 이용한 에이이에스 알고리즘의 암/복호 과정 동일화 설계 방법

(57) 요 약

본 발명은 대칭단을 이용한 AES 알고리즘의 암/복호 과정 동일화 설계 방법에 관한 것이다. 본 발명은 전체 r번의 진행 라운드 중에서 1라운드부터 r/2라운드까지의 r/2 라운드의 암호 과정을 진행하는 암호단, xor 연산단, rotate 연산단, and 연산단, or 연산단, 연접 연산단으로 구성되어 1번만 수행되는 대칭단, (r/2)+1라운드부터 r라운드까지의 r/2 라운드의 복호 과정을 진행하는 복호단을 포함한다.

대 표 도 - 도4

(72) 발명자

김종남

부산광역시 남구 용호동 197 GS하이츠자이 308-2801

조경연

부산시 해운대구 좌동 1271번지 신성아파트 104-1402

특허청구의 범위

청구항 1

암호단, 대칭단, 복호단으로 구성되는 것을 특징으로 하는 AES 알고리즘의 암/복호 과정 동일화 설계 방법.

청구항 2

제 1항에 있어서 암호단은 전체 r번의 진행 라운드 중에서 1라운드부터 r/2 라운드까지 전체의 r/2 횟수의 라운드만 수행하는 것을 특징으로 하는 AES 알고리즘의 암/복호 과정 동일화 설계 방법.

청구항 3

제 1항에 있어서 대칭단은 1번만 수행하는 것을 특징으로 하는 AES 알고리즘의 암/복호 과정 동일화 설계 방법.

청구항 4

제 1항에 있어서 복호단은 전체 r번의 진행 라운드 중에서 (r/2)+1 라운드부터 r 라운드까지 전체의 r/2 횟수의 라운드만 수행하는 것을 특징으로 하는 AES 알고리즘의 암/복호 과정 동일화 설계 방법.

청구항 5

제 3항에 있어서 대칭단은 xor 연산단, rotate 연산단, and 연산단, or 연산단, 연접 연산단으로 구성되는 것을 특징으로 하는 AES 알고리즘의 암/복호 과정 동일화 설계 방법.

청구항 6

제 5항에 있어서 대칭단의 결과가 아래의 [수학식 1]에 의하여 산출되는 것을 특징으로 하는 AES 알고리즘의 암/복호 과정 동일화 설계 방법.

수학식 1

도 2의 대칭단 내의 연산을 수학식으로 표기하면 다음과 같으며, 수학식에 사용된 연산자는 다음과 같다.

^ : 비트별 xor 연산자.

<<< : 왼쪽 rotate 연산자.

& : 비트별 and 연산자.

| : 비트별 or 연산자.

|| : 연접 연산자.

```
B` = ( b(A, k[20]) <<<  12) ^ B
A` = ( b(B`, k[23]) <<<  29) ^ A
D` = ( b(C, k[22]) <<<  12) ^ D
C` = ( b(D`, k[21]) <<<  29) ^ C
```

여기서 함수 b(X, Y)는 32비트 워드 2개 X, Y를 입력 받아 이 X, Y를 다시 바이트 단위로 나누어 and, or 연산을 교번 수행 하는 함수로서 다음과 같다.

$X = x_3 \,||\, x_2 \,||\, x_1 \,||\, x_0$

$Y = y_3 \,||\, y_2 \,||\, y_1 \,||\, y_0$

$b(X,Y) = b((x_3 \,\&\, y_3)||(x_2 \mid y_2)||(x_1 \,\&\, y_1)||(x_0 \mid y_0))$

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 블록 암호는 크게 Feistel구조와 SPN(Substitution Permutation Network)구조로 분류할 수 있으며, Feistel구조는 암/복호 알고리즘이 같다는 것이 장점이고 SPN구조는 암/복호 알고리즘이 다른 특징이 있다. 특히 하드웨어로 암호 시스템을 구현할 때 암/복호 알고리즘이 다른 SPN 구조는 Feistel 구조보다 하드웨어 면적이 약 2배 정도 증가하는 단점이 있다.

[0002] 대표적인 SPN구조의 블록 암호 알고리즘인 Rijndael은 새로운 표준 블록 암호 알고리즘을 선정하는 프로젝트인 AES 공모를 통해 채택된 표준 블록 암호 알고리즘이다. 이후 본 발명에서는 Rijndael을 AES라고 표기한다. AES는 SPN 구조로 암/복호 알고리즘이 다른 특징을 가지고 있다. 이러한 특징은 암호 알고리즘을 하드웨어로 구현할 때 암/복호 알고리즘이 같은 것보다 면적이 상당히 증가하는 단점이 된다. 이와 같은 AES의 단점을 보완하는 암/복호가 동일한 AES 암호 알고리즘에 관한 발명이다.

배 경 기 술

[0003] 데이터 무결성(Integrity)을 보장하는 수단으로 데이터에 대한 대칭 키(Symmetric key) 암호화가 일반적이다. 대칭 키 암호는 블록 암호(Block cipher)와 스트림 암호(Stream cipher)로 나눌 수 있으며, 블록 암호는 많은 알고리즘들이 개발 되었고, 특히 미국의 AES(Advanced Encryption Standard) 선정 프로젝트, 유럽의 NESSIE(New European Schemes for Signatures. Integrity. and Encryption) 프로젝트, 일본의 CRYPTREC(Cryptography Research and Evaluation Committees) 프로젝트 등 선진 각국들은 자국의 정보보호를 위한 암호 알고리즘을 공모를 통해 선정하고 있다. 우리나라는 자체 개발한 SEED와 ARIA를 표준 128비 블록 암호로 제정하였다. 초기 블록 암호는 소프트웨어로 구현되었지만 최근에는 빠른 정보처리를 위한 하드웨어 구현에 많은 연구가 진행 중이다.

[0004] AES는 SPN 구조의 블록 암호 알고리즘으로 SPN 구조는 C. E. Shannon의 혼돈(Confusion)과 확산(Diffusion) 이론을 바탕으로 구성되었으며, 혼돈은 평문과 암호문과의 상관관계를 숨기는 것을 말하고 치환을 통해 이룰 수 있고, 확산은 평문의 통계적 특성을 암호문 전반에 확산시키는 것을 말하고 이와 같은 혼돈과 확산을 반복 적용하여 안전성을 높인 블록 암호 알고리즘이 AES이다.

[0005] AES는 128비트 블록이며 키의 길이에 의해 가변적인 라운드를 적용한다. 본 발명에서는 128비트 키를 적용한 10 라운드 AES 알고리즘으로 설명한다. AES는 암/복호과정에서 생성되는 중간 결과 값 스테이트(State)를 바이트 단위로 4×4의 2차원 행렬로 간단히 표현할 수 있고, 4×4의 2차원 행렬은 기존의 행 우선이 아닌 열 우선으로 행렬의 순서를 표시한다.

[0006] AES의 라운드 함수 내에 크게 4가지의 독립적인 함수가 있으며, 라운드 함수를 C 코드로 표기하면 다음과 같다.

[0007]

```
Round(State, RoundKey)
{
        ByteSub(State);
        ShiftRow(State);
        MixColumn(State);
        AddRoundKey(State, RoundKey);
}
Final Round(State, RoundKey)
{
        ByteSub(State);
        ShiftRow(State);
        AddRoundKey(State, RoundKey);
}
```

[0008] 라운드 함수의 입력으로 State는 암호화 및 복호화 과정의 중간 결과 값을 나타내며, RoundKey는 키 스케줄링 알고리즘에 의해 생성된 라운드 키이다. ByteSub() 함수는 8비트 S-박스를 이용한 비선형 바이트 치환 함수이다. ShiftRow() 함수는 행 단위 왼쪽 rotate 연산 수행함수로 첫 번째 행은 변환하지 않으며, 두 번째 행은 1 바이트 rotate 연산, 세 번째 행은 2 바이트 rotate 연산, 네 번째 행은 3 바이트 rotate 연산을 적용하다. MixColumn() 함수는 열 단위로 혼합을 수행하는 32비트 선형 변환 함수이다. AddRoundKey() 함수는 라운드 키와 xor 연산을 수행하는 함수이다. AES의 암호화 과정에서 라운드 함수를 적용하기 전에 화이트닝 단계로서 평문과 라운드 키와 xor 연산을 먼저 적용하고 라운드 함수를 수행한다. 그리고 마지막 라운드는 MixColumn() 함수를 제외하고 라운드 함수를 수행한다. 복호화 과정은 암호화 과정에서 수행한 4개의 개별 함수들의 역 변환 함수가 존재하고 그 역변환 함수와 암호화 과정의 라운드 키의 역순으로 적용한다.

발명의 내용

해결 하고자하는 과제

[0009] 본 발명의 암/복호가 동일한 AES 알고리즘은 기존의 AES 알고리즘을 변경 없이 그대로 사용하면서 바이트단위 논리 연산과 고정된 rotate 연산으로 구성된 대칭단의 삽입만으로 암/복호가 동일한 AES 알고리즘을 만든다. 이와 같은 대칭단의 구성은 소프트웨어 및 하드웨어 구현이 쉽고 수행 속도도 빠르다. 그리고 제안하는 암/복호가 동일한 AES 알고리즘은 전체 진행 라운드의 1/2은 암호 알고리즘을 수행하고, 나머지 1/2은 복호 알고리즘을 수행하면서 중간에 대칭단을 삽입하여 암/복호 알고리즘이 동일한 개선된 AES를 만든다.

[0010] 대칭단의 설계목표는 다음과 같다.

[0011] 암/복호가 동일한 AES를 만들 것,

[0012] 동일한 라운드의 적용에서 대칭단의 삽입으로 불규칙성을 통해서 AES의 안전성을 향상시킬 것,

[0013] 소프트웨어 및 하드웨어 구현이 쉬울 것,

[0014] 기존의 AES와 소프트웨어 및 하드웨어 수행속도에서 큰 차이가 없을 것.

과제 해결수단

[0015] 본 발명은 암/복호를 동일하게 구성하는 개선된 AES 알고리즘을 개발하는 것으로 크게 3단계로 진행한다.

[0016] AES 암호 알고리즘을 전체 라운드 중 1/2만 수행하는 단계,

[0017] 대칭단을 적용하는 단계,

[0018] AES 복호 알고리즘을 나머지 1/2 라운드를 수행하는 단계로 나눌 수 있다.

[0019] 이와 같은 진행의 구성은 암/복호가 동일한 AES 알고리즘을 만들 수 있다.

효 과

[0020] PDA나 핸드폰과 같은 한정된 하드웨어 자원을 가지는 장치에서 보안이 필수적인 데이터 통신을 위한 암호 알고리즘의 하드웨어 구현은 암/복호 알고리즘이 다를 경우 암/복호 알고리즘이 같은 경우보다 하드웨어 면적이 2배 정도 증가하게 된다. 이에 본 발명으로 기존의 암/복호가 다른 AES를 암/복호가 같은 알고리즘으로 구현했으며, 기존의 AES와 유사한 안전성을 유지하면서 암/복호가 동일하기 때문에 하드웨어 구성이 간단하고 쉽게 구현할 수 있다. 또한 스마트카드 및 능동형 RFID 태그와 같은 제한된 하드웨어 및 소프트웨어 환경에서도 쉽게 구현 가능하다.

[0021] 본 발명의 핵심인 대칭단은 간단한 논리연산으로 구성되어 소프트웨어 및 하드웨어 수행속도가 빠르다. 다시 말해 기존의 AES와 소프트웨어 수행속도 테스트에서 거의 같은 수행속도를 보여주고 있으며 이는 현재 AES 알고리즘을 적용하고 있는 암호 시스템에 그대로 적용이 가능하다.

[0022] 그리고 AES 알고리즘이외의 다른 SPN구조의 블록 암호 알고리즘들과 암/복호 알고리즘이 다른 유사한 구조의 암호 알고리즘에 본 발명의 대칭단 구조를 삽입해서 암/복호 알고리즘을 동일하게 구현할 수 있고, 새로운 암/복호 알고리즘이 같은 블록 암호 설계 및 개발에도 유용한 아이디어로 사용될 것이다.

발명의 실시를 위한 구체적인 내용

[0023] 본 발명에서 대칭단 구조를 AES 알고리즘에 적용할 때 도 1의 110 ByteSub(), 120 ShiftRow(), 130 MixColumn(), 140 AddRound Key() 함수를 변경 없이 그대로 사용한다. 그러나 전체 진행 라운드에서 1/2라운드만 암호 알고리즘을 사용하고 나머지 1/2라운드는 복호 알고리즘을 사용하고, 암호 알고리즘에서 복호 알고리즘으로 바뀌는 중간에 대칭단 연산이 삽입된다(대표도 참조).

[0024] AES의 암호 알고리즘 구성은 모든 라운드에 동일한 연산이 반복되면 수학식이 단순해져서 선형 공격에 취약하다. 그래서 1 라운드 시작 전에 AddRound Key() 함수를 수행하고 마지막 라운드에서는 MixColumn() 함수를 제외하고 있다. 본 발명에서는 AES와는 다르게 대표도와 같이 전체 라운드에서 중간에 불규칙적인 라운드인 대칭단 연산을 삽입해서 수학식이 단순해지는 점을 극복하므로 AES와 같은 1 라운드 시작 전에 AddRoundKey() 함수와 마지막 라운드에서 MixColumn() 함수를 제외하는 것이 필요하지 않다.

[0025] 대표도에서 보는 바와 같이 100 AES 암호 알고리즘을 수행하는 부분은 AES의 AddRound Key(), ByteSub(), ShiftRow(), MixColumn() 함수 순서로 1-5 라운드까지 적용한다. 이는 AES의 전체 라운드의 1/2에 해당되며, 진행과정은 AES 알고리즘과의 차이는 없고, 단지 초기 키 덧셈 부분이 본 발명에서는 라운드 함수에 포함되어 진행하면서 5 라운드 이후 사용되는 라운드 키가 도 2의 211, 221, 231, 241과 같이 대칭단에서 사용되는 32비트 4개의 라운드 키로 된다는 차이가 있다. 이와 같이 1-5 라운드까지 반복 수행 후 대표도의 200 대칭단을 수행한다. 대칭단 수행은 한번만 하며, 상세한 대칭단 수행 과정은 도 2에 나타냈다. 먼저 도 2의 210 A 32비트 암호화 과정을 수행한 첫 번째 워드는 211 b() 함수를 수행한다. b() 함수는 32비트 A와 대칭단에 적용될 첫 번째 32비트 라운드 k[20]을 입력으로 받아 32비트 워드를 바이트 단위로 나눈 다음 and, or 연산을 교번 수행한다. 그리고 도 2의 211 b() 함수 수행 후 결과값은 도 2의 212 고정된 12비트 왼쪽 rotate 연산을 수행하고 두 번째 32비트 워드 값인 도 2의 220 B와 도 2의 223 xor 연산을 수행한 후 최종적으로 도 2의 260 B` 32비트 값을 만든다. 이 도 2의 260 B`는 다시 도 2의 221 b() 함수를 수행하는데, 이때는 네 번째 32비트 라운드 키 k[23]을 사용하고 수행 방법은 도 2의 211과 같이 바이트 단위로 나눈 후 and, or 연산을 교번 수행한다. 도 2의 221 b() 함수를 수행한 후 도 2의 222 고정된 29비트 왼쪽 rotate 연산을 수행한 후 도 2의 210 A와 도 2의 213 xor 연산을 수행한 값이 최종적으로 도 2의 250 A` 32비트 결과 값이다. 세 번째 32비트 워드 값인 도 2의 230 C는 도 2의 231 b() 함수를 수행한다. 이때 세 번째 32비트 라운드 키 k[22]와 and, or 연산을 교번 수행한 후 도 2의 232 고정된 12비트 왼쪽 rotate 연산을 수행한다. 그리고 네 번째 워드 값인 도 2의 240 D와 도 2의 243 xor 연산을 수행한 후 최종적으로 도 2의 280 D`를 생성한다. 이 D`는 도 2의 241 b() 함수를 수행하는데, 이때 두 번째 32비트 라운드 키 k[21]을 사용하여 and, or 연산을 교번 적용한다. 그리고 도 2의 242 고정된 29비트 왼쪽 rotate 연산을 적용한 후 도 2의 도 2의 230 C와 도 2의 233 xor 연산을 수행한 후 최종적으로 32비

트 워드 값인 C`를 만든다.

[0026] 대칭단 수행 후 대표도의 300 AES 복호 알고리즘을 수행하고 암호 알고리즘에서 사용된 4개의 복호용 역 알고리즘을 사용한다. AES 복호 알고리즘의 적용 순서는 도 3의 310 InvMixColumn(), 320 InvByteSub(), 330 InvShiftRow(), 340 InvAddRoundKey() 함수 순으로 적용하고 6-10 라운드를 반복 수행하여 최종적으로 암호문을 만든다.

[0027] 복호과정은 대표도의 순서를 그대로 적용하고, 단지 라운드 키의 입력이 암호화 과정의 역순으로 입력하면 된다. 본 발명에서 제안한 알고리즘의 키 스케줄링은 AES의 키 스케줄링을 그대로 사용한다.

도면의 간단한 설명

[0028] 대표도는 본 발명의 전체 알고리즘 진행도.

[0029] 도 1은 도 4의 100 AES 암호 알고리즘 진행과정으로 AES 전체 라운드 중에서 1/2 라운드만 진행.

[0030] 도 2는 도 4의 200 대칭단 진행과정으로 한번만 수행.

[0031] 도 3은 도 4의 300 AES 복호 알고리즘 진행과정으로 AES 전체 라운드 중에서 나머지 1/2 라운드만 진행.

[0032] 도 4는 본 발명의 전체 알고리즘 진행도

[0033] 도면의 주요 부분에 대한 부호 설명.

[0034] 100 : AES 암호 알고리즘 수행부.

[0035] 110 : AES 암호 라운드 함수 중 라운드 키와 xor 연산 수행부.

[0036] 120 : AES 암호 라운드 함수 중 바이트 단위 비선형 치환 수행부.

[0037] 130 : AES 암호 라운드 함수 중 32비트 워드 단위로 첫째 행은 변화 없고, 둘째 행은 1바이트 왼쪽 rotate 연산 수행, 셋째 행은 2바이트 왼쪽 rotate 연산 수행, 마지막으로 넷째 행은 3바이트 왼쪽 rotate 연산 수행부.

[0038] 140 : AES 암호 라운드 함수 중 32비트 워드 단위로 열 단위로 선형 변환 수행부.

[0039] 200 : 대칭단 적용부.

[0040] 210 : 128비트 스테이트 중에서 첫 번째 32비트 값.

[0041] 211 : 32비트 210 A와 라운드 키 k[20]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.

[0042] 212 : 211 과정을 수행 후 12비트 왼쪽 rotate 연산 수행부.

[0043] 223 : 212 과정을 수행 후 32비트 210 B와 xor 연산 수행부.

[0044] 260 : 223 과정을 수행 후 32비트 값을 저장.

[0045] 220 : 128비트 스테이트 중에서 두 번째 32비트 값.

[0046] 221 : 260에 저장된 32비트 값과 라운드 키 k[23]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.

[0047] 222 : 221 과정을 수행 후 왼쪽 29비트 rotate 연산 수행부.

[0048] 213 : 222 과정을 수행 후 32비트 210 A와 xor 연산 수행부.

[0049] 250 : 213 과정을 수행 후 32비트 값 저장.

[0050] 230 : 128비트 스테이트 중에서 세 번째 32비트 값.

[0051] 231 : 230에 저장된 32비트 값과 라운드 키 k[22]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.

[0052] 232 : 231 과정을 수행 후 왼쪽 29비트 rotate 연산 수행부.

[0053] 243 : 232 과정을 수행 후 32비트 240 D와 xor 연산 수행부.

[0054] 280 : 243 과정을 수행 후 32비트 값 저장.

[0055] 240 : 128비트 스테이트 중에서 네 번째 32비트 값.

[0056] 241 : 280에 저장된 32비트 값과 라운드 키 k[21]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.

[0057] 242 : 242 과정을 수행 후 왼쪽 12비트 rotate 연산 수행부.

[0058] 233 : 242 과정을 수행 후 32비트 230 C와 xor 연산 수행부.

[0059] 270 : 233 과정을 수행 후 32비트 값 저장.

[0060] 300 : AES 복호 알고리즘 수행부.

[0061] 310 : AES 복호 라운드 함수 중 32비트 워드 단위로 열 단위로 선형 변환 수행부.

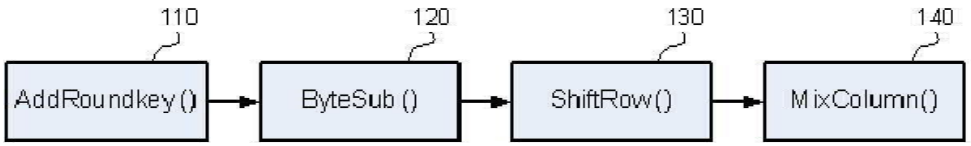
[0062] 320 : AES 복호 라운드 함수 중 바이트 단위 비선형 치환 수행부.

[0063] 330 : AES 복호 라운드 함수 중 32비트 워드 단위로 첫째 행은 변화 없고, 둘째 행은 1바이트 오른쪽 rotate 연산 수행, 셋째 행은 2바이트 오른쪽 rotate 연산 수행, 마지막으로 넷째 행은 3바이트 오른쪽 rotate 연산 수행부.

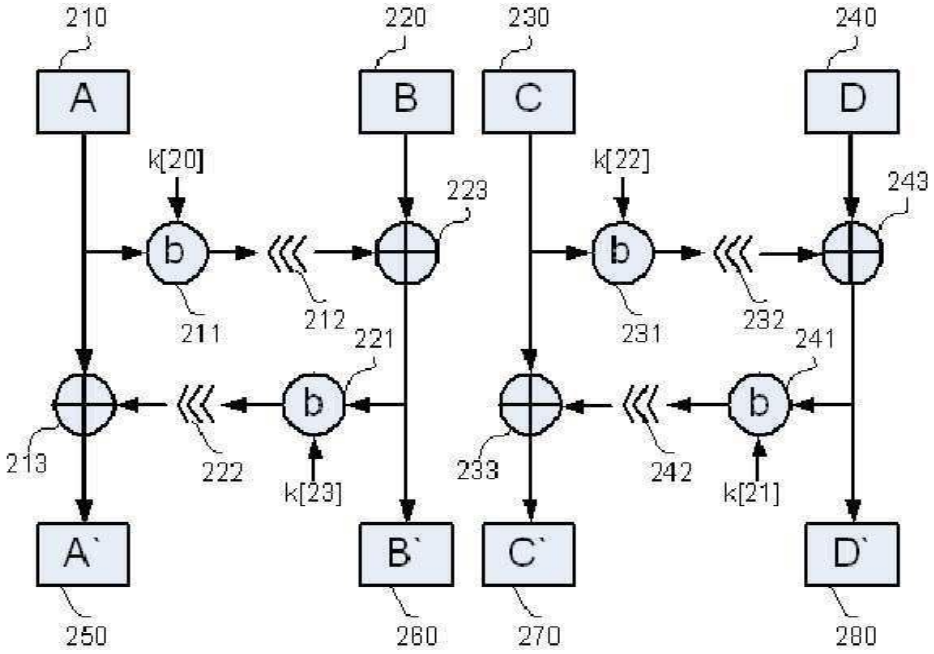
[0064] 340 : AES 복호 라운드 함수 중 라운드 키와 xor 연산을 수행부.

도면

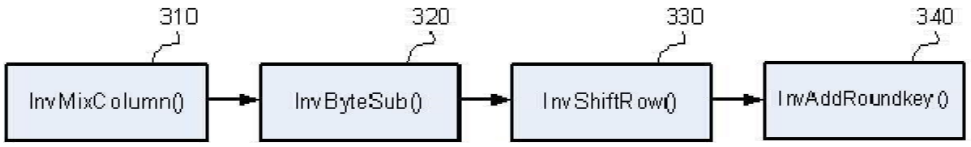
도면1



도면2



도면3


310
320
330
340
InvMixColumn()
InvByteSub()
InvShiftRow()
InvAddRoundkey()

도면4


100
200
300
AES
암호단
대칭단
AES
복호단