(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년11월03일
(11) 등록번호 10-2462395
(24) 등록일자 2022년10월28일

(51) 국제특허분류(Int. Cl.)
H04L 9/12 (2006.01) *G06F 7/50* (2006.01)
G06F 7/53 (2006.01) *G06F 7/72* (2006.01)
H04L 9/00 (2022.01) *H04L 9/06* (2006.01)
(52) CPC특허분류
H04L 9/12 (2013.01)
G06F 7/50 (2021.08)
(21) 출원번호 10-2022-0029611
(22) 출원일자 2022년03월08일
심사청구일자 2022년03월08일
(56) 선행기술조사문헌
KR1020210128303 A*
KR1020210130196 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
이한호
인천광역시 연수구 컨벤시아대로252번길 30, 1505동 1701호(송도동, 송도 더샵 퍼스트파크 F15BL)
김성재
경기도 고양시 일산서구 일현로 97-11, 102동 704호(탄현동, 일산 위브더제니스)
(74) 대리인
양성보

전체 청구항 수 : 총 9 항 심사관 : 강민성

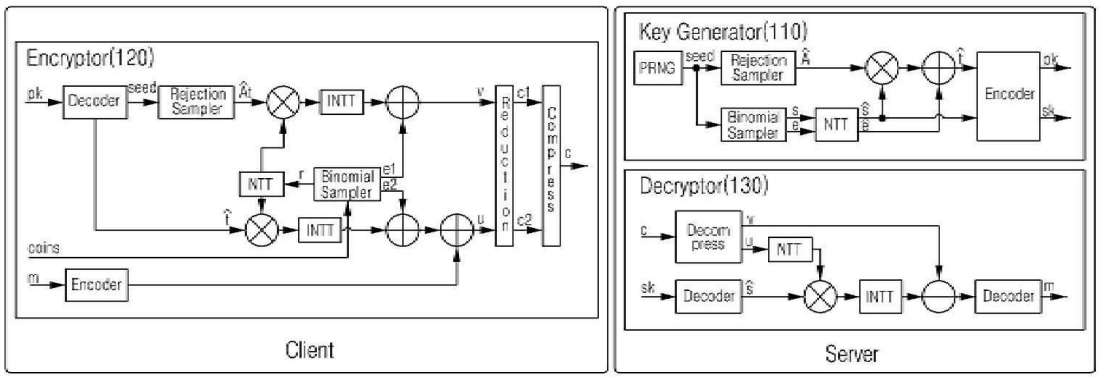
(54) 발명의 명칭 **양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템 및 방법**

(57) 요 약

모듈-LWE 기반 암호 프로세서 시스템 및 그 동작 방법이 제시된다. 본 발명에서 제안하는 암호 프로세서 시스템은 PRNG 코어에서 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하여 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 인코더를

(뒷면에 계속)

대 표 도



이용하여 공개키 및 비밀키로 인코딩하여 출력하는 키생성 프로세서, 메시지와 상기 키생성 프로세서로부터 공개키를 수신하여 상기 메시지를 인코딩하고, 상기 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력하는 암호화 프로세서 및 상기 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 메시지 디코더를 이용하여 원본 메시지를 출력하는 복호화 프로세서를 포함한다.

(52) CPC특허분류

G06F 7/53 (2013.01)
G06F 7/728 (2013.01)
H04L 9/005 (2013.01)
H04L 9/0643 (2013.01)
H04L 9/0662 (2013.01)
H04L 2209/046 (2013.01)
H04L 2209/125 (2013.01)
H04L 2209/34 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711139247
과제번호	IITP-2021-0-02052
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	대학ICT연구센터육성지원사업
연구과제명	스마트 모빌리티를 위한 인공지능 시스템반도체 핵심 기술 개발 및 인력 양성
기 여 율	1/3
과제수행기관명	인하대학교 산학협력단
연구기간	2021.07.01 ~ 2028.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711129091
과제번호	2021R1A2C1011232
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	기초연구사업
연구과제명	고신뢰성 고성능 격자-기반 양자내성암호 하드웨어 아키텍처 연구
기 여 율	1/3
과제수행기관명	인하대학교 산학협력단
연구기간	2021.03.01 ~ 2025.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호	1711134703
과제번호	20210007790012003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	방송통신산업기술개발사업
연구과제명	HW지원 프라이버시 보장 암호데이터 고속처리 기술 개발
기 여 율	1/3
과제수행기관명	인하대학교 산학협력단
연구기간	2021.04.01 ~ 2024.12.31

공지예외적용 : 있음

명 세 서

청구범위

청구항 1

PRNG 코어에서 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 인코더를 이용하여 암호화 프로세서 및 복호화 프로세서에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력하는 키생성 프로세서;

사용자가 암호화 하고자 하는 메시지와 상기 키생성 프로세서로부터 공개키를 수신하여 상기 메시지를 인코딩하고, 상기 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력하는 암호화 프로세서; 및

상기 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더를 이용하여 원본 메시지를 출력하는 복호화 프로세서

를 포함하고,

상기 암호화 프로세서는,

사용자가 입력한 메시지를 1비트 단위로 마스킹하는 메시지 인코더;

공개키를 수신하여 공개키를 역-직렬화하는 공개키 디코더;

공개키 디코더의 출력을 입력 받아 공개키 생성과 암호화에 필요한 다항식 행렬을 생성하는 거부 샘플러;

비밀키 생성과 암호화에 필요한 에러 계수를 생성하는 이항 샘플러;

이항 샘플러의 출력을 입력 받아 NTT 연산을 수행하는 NTT 코어;

거부 샘플러와 이항 샘플러의 값을 연산하는 포인트-와이즈 곱셈기 및 덧셈기;

포인트-와이즈 곱셈기를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행하는 INTT 코어;

곱셈 연산 결과에 대해 몽고메리 리덕션 연산을 수행하는 MR1;

덧셈 연산 결과에 대해 바렛 리덕션 연산을 수행하는 MR2; 및

리덕션된 데이터를 압축 및 인코딩하여 암호문으로 출력하는 컴프레스

를 포함하는 암호 프로세서 시스템.

청구항 2

제1항에 있어서,

상기 키생성 프로세서는,

의사난수를 생성하는 PRNG 코어;

PRNG 코어에서 생성한 의사난수를 이용하여 공개키 생성 및 암호화에 필요한 다항식 행렬을 생성하는 거부 샘플러;

PRNG 코어에서 생성한 의사난수를 이용하여 공개키 생성 및 비밀키 생성에 사용하기 위한 에러 계수를 생성하는 이항 샘플러;

이항 샘플러의 출력을 입력 받아 NTT 연산을 수행하는 NTT 코어;

연산 과정에 사용되는 포인트-와이즈 곱셈기 및 덧셈기; 및

연산된 결과를 직렬화하는 비밀키 인코더

를 포함하는 암호 프로세서 시스템.

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 복호화 프로세서는,

암호문을 수신하여 압축 해제 및 디코딩하는 암호문 디컴프레스;

비밀키를 수신하여 비밀키를 역-직렬화하는 비밀키 디코더;

암호문 디컴프레스의 출력을 입력 받아 NTT 연산을 수행하는 NTT 코어;

복호화 연산에 필요한 포인트-와이즈 곱셈기 및 뺄셈기;

포인트-와이즈 곱셈기를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행하는 INTT 코어;

곱셈 연산 결과에 대해 몽고메리 리덕션 연산을 수행하는 MR1;

덧셈 연산 결과에 대해 바렛 리덕션 연산을 수행하는 MR2; 및

연산된 결과로 얻은 메시지의 마스크를 제거하는 메시지 디코더

를 포함하는 암호 프로세서 시스템.

청구항 5

제1항에 있어서,

상기 PRNG 코어는 Keccak 알고리즘을 사용하는 SHA3로 샘플링의 입력 비트에 따라 출력 비트의 길이를 조정하고,

상기 거부 샘플러는 상기 PRNG 코어의 출력을 16비트 단위로 입력 받아 균일한 임의의 정수 modulo q 를 생성하는 모듈로서 19q보다 작은 값을 통과시켜 바렛 리덕션을 수행하도록 하고,

상기 이항 샘플러는 상기 PRNG 코어의 출력을 8비트 단위로 입력 받아 중심이항분포에 따라 계수 값을 생성하고,

상기 NTT 코어는 2-병렬 NTT 코어로서 총 7개의 스테이지(stage)를 포함하고, 각각의 스테이지는 PE(Processing Element)로 구성되고,

상기 포인트-와이즈 곱셈기는 베이스 곱셈기 및 누산기를 포함하는

암호 프로세서 시스템.

청구항 6

제5항에 있어서,

상기 NTT 코어의 상기 PE는 입력 값과 ROM에 저장된 값과 곱셈연산 후 순서대로 몽고메리 리덕션, BU 연산, 바렛 리덕션 연산을 수행하여 출력하는

암호 프로세서 시스템.

청구항 7

제6항에 있어서,

상기 INTT 코어의 PE는 NTT 코어의 PE와 반대되는 구조를 갖고 입력 값을 BU와 바렛 리덕션 연산한 값을 ROM에 저장된 값과 곱셈 연산 후 몽고메리 리덕션 연산을 수행하여 출력하는

암호 프로세서 시스템.

청구항 8

제5항에 있어서,

상기 베이스 곱셈기는 곱셈을 수행할 각각의 다항식으로부터 두 개씩 입력으로 받아 Bow-tie 곱셈을 수행하고,

상기 누산기는 행렬의 곱셈 방식에 따라 곱셈 연산을 수행한 후 각각의 다항식 열을 더해주는 역할을 수행하는

암호 프로세서 시스템.

청구항 9

제1항에 있어서,

상기 MR1은 몽고메리 리덕션을 통해 곱셈으로 커진 연산 결과를 줄이기 위해 NTT 코어 및 포인트-와이즈 곱셈기에 사용되고,

MR2은 바렛 리덕션을 통해 덧셈으로 커진 연산 결과를 줄이기 위해 사용되는

암호 프로세서 시스템.

청구항 10

키생성 프로세서가 PRNG 코어로 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 인코더를 이용하여 암호화 프로세서 및 복호화 프로세서에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력하는 단계;

암호화 프로세서가 사용자가 암호화 하고자 하는 메시지와 상기 키생성 프로세서로부터 공개키를 수신하여 상기 메시지를 인코딩하고, 상기 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력하는 단계; 및

복호화 프로세서 상기 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더를 이용하여 원본 메시지를 출력하는 단계

를 포함하고,

상기 암호화 프로세서가 암호문을 압축 및 인코딩하여 출력하는 단계는,

메시지 인코더를 통해 사용자가 입력한 메시지를 1비트 단위로 마스킹하고;

공개키 디코더를 통해 공개키를 수신하여 공개키를 역-직렬화하고;

거부 샘플러를 통해 공개키 디코더의 출력을 입력 받아 공개키 생성과 암호화에 필요한 다항식 행렬을 생성하고;

이항 샘플러를 통해 비밀키 생성과 암호화에 필요한 에러 계수를 생성하고;

NTT 코어를 통해 이항 샘플러의 출력을 입력 받아 NTT 연산을 수행하고;

포인트-와이즈 곱셈기 및 덧셈기를 통해 거부 샘플러와 이항 샘플러의 값을 연산하고;

INTT 코어를 통해 상기 포인트-와이즈 곱셈기를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행하고;

MR1를 통해 상기 곱셈 연산 결과에 대해 몽고메리 리덕션 연산을 수행하고;

MR2를 통해 상기 덧셈 연산 결과에 대해 바렛 리덕션 연산을 수행하고;

컴프레스를 통해 상기 리덕션된 데이터를 압축 및 인코딩하여 암호문으로 출력하는

암호화 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템 및 방법에 관한 것이다.

배 경 기 술

[0002] 현재 많이 사용되고 있는 RSA(Rivest, Shamir, and Adleman)와 ECC(Elliptic Curve Cryptosystem) 등과 같이 인수분해 및 이산로그 문제를 이용한 기존 공개키 암호 알고리즘은 양자컴퓨터를 사용하게 되면 인수분해나 이산대수 문제를 효율적으로 해결할 수 있게 되어 다항시간 안에 해독이 가능하게 된다. 따라서, 기존 컴퓨터와 양자컴퓨터 모두로부터 안전한 공개키 암호인 양자후 암호 또는 양자내성암호(Post-Quantum Cryptography)의 필요성이 요구되고 있다. 양자내성암호는 기존의 공개키 암호가 가지고 있던 문제를 해결하고 대체할 수 있는 암호로, 현재 미국 NIST(National Institute of Standards and Technology)에서 양자내성암호 알고리즘 공모전을 통해 표준화를 진행 중에 있다.

[0003] 양자내성암호는 기존의 인수분해나 이산대수 문제가 아닌 전혀 새로운 문제에 안전성을 기반한 암호로서, 기반하는 분야에 따라 격자-기반, 코드-기반, 해시-기반, 아이소제니-기반, 다변수-기반 암호 등으로 구분된다. 그 중에 격자-기반 암호는 고차원 격자에 정의된 다양한 유형의 어려운 문제(hard problem)를 기반으로 하는 강력한 암호로서 많은 관심을 받고 있다.

[0004] 격자-기반 암호 알고리즘은 주로 에러가 포함된 선형방정식에서 균일한 랜덤 벡터를 구별하는 LWE(Learning With Error) 문제를 이용한다. LWE는 키의 크기가 커지고 연산 속도가 느린 단점이 있어 환(Ring) 안에서 수행하는 링(Ring)-LWE 문제가 제안되었다. 모듈-LWE는 격자-기반 알고리즘으로서 링-LWE를 일반화한 암호 알고리즘이며, 가장 많은 시간자원을 사용하는 다항식 행렬 곱셈의 연산 속도 향상을 위해 고성능 NTT를 필요로 한다.

[0005] 현재 양자내성암호 기술은 미래 암호 기술의 핵심으로 다양한 응용 가능성 때문에 학계 및 산업계의 관심이 높으나, 수학적 난이도가 높고 기존 암호 체계에 비해 암복호화 연산 속도가 느려, 실용화를 위해서는 고성능 양자내성암호 하드웨어 아키텍처 구현이 필요하다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 한국등록특허 제10-1952547호(2019.02.20.)

발명의 내용

해결하려는 과제

[0007] 본 발명이 이루고자 하는 기술적 과제는 모듈-LWE(Module-Learning With Error)와 NTT(Number Theoretic Transform) 코어를 이용하여 높은 데이터 처리율과 확장성을 가지는 모듈-LWE 기반 양자내성암호 프로세서 시스템 및 방법을 제공하는데 있다. 더욱 상세하게는, 2-병렬 MDF(Multi-path Delay Feedback) 기반의 NTT 코어를 사용하여 높은 연산속도를 가지고, 리소스의 재활용과 확장성이 용이한 모듈-LWE 기반 양자내성암호의 효율적인 시스템 및 방법을 제공하는데 있다.

과제의 해결 수단

[0008] 일 측면에 있어서, 본 발명에서 제안하는 양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템은 PRNG 코

어에서 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 인코더를 이용하여 암호화 프로세서 및 복호화 프로세서에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력하는 키생성 프로세서, 사용자가 암호화 하고자 하는 메시지와 상기 키생성 프로세서로부터 공개키를 수신하여 상기 메시지를 인코딩하고, 상기 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력하는 암호화 프로세서 및 상기 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더를 이용하여 원본 메시지를 출력하는 복호화 프로세서를 포함한다.

[0009] 본 발명의 실시예에 따른 키생성 프로세서는 의사난수를 생성하는 PRNG 코어, PRNG 코어에서 생성한 의사난수를 이용하여 공개키 생성 및 암호화에 필요한 다항식 행렬을 생성하는 거부 샘플러, PRNG 코어에서 생성한 의사난수를 이용하여 공개키 생성 및 비밀키 생성에 사용하기 위한 에러 계수를 생성하는 이항 샘플러, 이항 샘플러의 출력을 입력 받아 NTT 연산을 수행하는 NTT 코어, 연산 과정에 사용되는 포인트-와이즈 곱셈기 및 덧셈기 및 연산된 결과를 직렬화하는 비밀키 인코더를 포함한다.

[0010] 본 발명의 실시예에 따른 암호화 프로세서는 사용자가 입력한 메시지를 1비트 단위로 마스킹하는 메시지 인코더, 공개키를 수신하여 공개키를 역-직렬화하는 공개키 디코더, 공개키 디코더의 출력을 입력 받아 공개키 생성과 암호화에 필요한 다항식 행렬을 생성하는 거부 샘플러, 비밀키 생성과 암호화에 필요한 에러 계수를 생성하는 이항 샘플러, 이항 샘플러의 출력을 입력 받아 NTT 연산을 수행하는 NTT 코어, 거부 샘플러와 이항 샘플러의 값을 연산하는 포인트-와이즈 곱셈기 및 덧셈기, 포인트-와이즈 곱셈기를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행하는 INTT 코어, 곱셈 연산 결과에 대해 몽고메리 리덕션 연산을 수행하는 MR1, 덧셈 연산 결과에 대해 바렛 리덕션 연산을 수행하는 MR2 및 리덕션된 데이터를 압축 및 인코딩하여 암호문으로 출력하는 컴프레스를 포함한다.

[0011] 본 발명의 실시예에 따른 복호화 프로세서는 암호문을 수신하여 압축 해제 및 디코딩하는 암호문 디컴프레스, 비밀키를 수신하여 비밀키를 역-직렬화하는 비밀키 디코더, 암호문 디컴프레스의 출력을 입력 받아 NTT 연산을 수행하는 NTT 코어, 복호화 연산에 필요한 포인트-와이즈 곱셈기 및 뺄셈기, 포인트-와이즈 곱셈기를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행하는 INTT 코어, 곱셈 연산 결과에 대해 몽고메리 리덕션 연산을 수행하는 MR1, 덧셈 연산 결과에 대해 바렛 리덕션 연산을 수행하는 MR2 및 연산된 결과로 얻은 메시지의 마스크를 제거하는 메시지 디코더를 포함한다.

[0012] 본 발명의 실시예에 따른 PRNG 코어는 Keccak 알고리즘을 사용하는 SHA3로 샘플링의 입력 비트에 따라 출력 비트의 길이를 조정하고, 상기 거부 샘플러는 상기 PRNG 코어의 출력을 16비트 단위로 입력 받아 균일한 임의의 정수 modulo q를 생성하는 모듈로서 19q보다 작은 값을 통과시켜 바렛 리덕션을 수행하도록 하고, 상기 이항 샘플러는 상기 PRNG 코어의 출력을 8비트 단위로 입력 받아 중심이항분포에 따라 계수 값을 생성하고, 상기 NTT 코어는 2-병렬 NTT 코어로서 총 7개의 스테이지(stage)를 포함하고, 각각의 스테이지는 PE(Processing Element)로 구성되고, 상기 포인트-와이즈 곱셈기는 베이스 곱셈기 및 누산기를 포함한다.

[0013] 본 발명의 실시예에 따른 NTT 코어의 상기 PE는 입력 값과 ROM에 저장된 값과 곱셈연산 후 순서대로 몽고메리 리덕션, BU 연산, 바렛 리덕션 연산을 수행하여 출력한다.

[0014] 본 발명의 실시예에 따른 INTT 코어의 PE는 NTT 코어의 PE와 반대되는 구조를 갖고 입력 값을 BU와 바렛 리덕션 연산한 값을 ROM에 저장된 값과 곱셈 연산 후 몽고메리 리덕션 연산을 수행하여 출력한다.

[0015] 본 발명의 실시예에 따른 베이스 곱셈기는 곱셈을 수행할 각각의 다항식으로부터 두 개씩 입력으로 받아 Bow-tie 곱셈을 수행하고, 상기 누산기는 행렬의 곱셈 방식에 따라 곱셈 연산을 수행한 후 각각의 다항식 열을 더해주는 역할을 수행한다.

[0016] 본 발명의 실시예에 따른 MR1은 몽고메리 리덕션을 통해 곱셈으로 커진 연산 결과를 줄이기 위해 NTT 코어 및 포인트-와이즈 곱셈기에 사용되고, MR2은 바렛 리덕션을 통해 덧셈으로 커진 연산 결과를 줄이기 위해 사용된다.

[0017] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템의

암호화 방법은 키생성 프로세서가 PRNG 코어로 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 인코더를 이용하여 암호화 프로세서 및 복호화 프로세서에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력하는 단계, 암호화 프로세서가 사용자가 암호화 하고자 하는 메시지와 상기 키생성 프로세서로부터 공개키를 수신하여 상기 메시지를 인코딩하고, 상기 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력하는 단계 및 복호화 프로세서가 상기 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더를 이용하여 원본 메시지를 출력하는 단계를 포함한다.

발명의 효과

[0018] 본 발명의 실시예들에 따른 모듈-LWE 기반 암호프로세서는 모듈-LWE문제를 사용하여 확장성이 용이하도록 하고, 파이프라인이 가능한 2-병렬 구조의 NTT를 사용하여 높은 데이터 처리율과 높은 클럭 속도를 얻을 수 있어 연산 속도에 이점을 가진다. 또한 NTT 연산을 수행하는 과정에 있어서 데이터 메모리에 접근 없이 연속적인 연산이 가능해 하드웨어 복잡도를 낮출 수 있다. 추가적으로 몽고메리(Montgomery) 및 바렛(Barrett) 리덕션을 사용하여 부-채널 공격(Side-Channel attack) 중 하나인 타이밍 공격에 대한 보안성을 보장한다.

도면의 간단한 설명

[0019] 도 1은 본 발명의 일 실시예에 따른 모듈-LWE 기반 암복호화 시스템의 구성을 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 키생성 프로세서를 나타내는 도면이다.

도 3은 본 발명의 일 실시예에 따른 암호화 프로세서를 나타내는 도면이다.

도 4는 본 발명의 일 실시예에 따른 복호화 프로세서를 나타내는 도면이다.

도 5는 본 발명의 일 실시예에 따른 모듈-LWE 기반 암복호화 시스템의 암호화 방법을 설명하기 위한 흐름도이다.

도 6은 본 발명의 일 실시예에 따른 PRNG 코어를 나타내는 도면이다.

도 7은 본 발명의 일 실시예에 따른 거부 샘플러를 나타내는 도면이다.

도 8은 본 발명의 일 실시예에 따른 이항 샘플러를 나타내는 도면이다.

도 9는 본 발명의 일 실시예에 따른 2-병렬 NTT 코어를 나타내는 도면이다.

도 10은 본 발명의 일 실시예에 따른 NTT 코어의 PE(Processing Element)를 나타내는 도면이다.

도 11은 본 발명의 일 실시예에 따른 2-병렬 INTT 코어를 나타내는 도면이다.

도 12는 본 발명의 일 실시예에 따른 INTT 코어의 PE(Processing Element)를 나타내는 도면이다.

도 13은 본 발명의 일 실시예에 따른 몽고메리(Montgomery) 리덕션 연산을 수행하는 MR1(Modulo Reduction 1)를 나타내는 도면이다.

도 14는 본 발명의 일 실시예에 따른 바렛(Barrett) 리덕션 연산을 수행하는 MR2(Modulo Reduction 2)를 나타내는 도면이다.

도 15는 본 발명의 일 실시예에 따른 인코더를 나타내는 도면이다.

도 16은 본 발명의 일 실시예에 따른 디코더를 나타내는 도면이다.

도 17은 본 발명의 일 실시예에 따른 컴프레스를 나타내는 도면이다.

도 18은 본 발명의 일 실시예에 따른 컴프레스의 CE(Compress Element)를 나타내는 도면이다.

도 19는 본 발명의 일 실시예에 따른 디컴프레스를 나타내는 도면이다.

도 20은 본 발명의 일 실시예에 따른 디컴프레스의 DE(Decompress Element)를 나타내는 도면이다.

도 21은 본 발명의 일 실시예에 따른 메시지 인코더를 나타내는 도면이다.

도 22는 본 발명의 일 실시예에 따른 메시지 디코더를 나타내는 도면이다.

도 23은 본 발명의 일 실시예에 따른 베이스 곱셈기(Base multiplier)를 나타내는 도면이다.

도 24는 본 발명의 일 실시예에 따른 누산기(Accumulator)를 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0020] 본 발명의 실시예에 따르면, 2-병렬 MDF(Multi-path Delay Feedback) 기반의 NTT 코어를 사용하여 높은 연산속도를 가지고, 리소스의 재활용과 확장성이 용이한 모듈-LWE 기반 양자내성암호의 효율적인 암호 프로세서 시스템을 제공하는데 있다. 본 발명의 실시예에 따른 모듈-LWE 기반 암호 프로세서 시스템은 키생성 프로세서, 암호화 프로세서, 복호화 프로세서로 구성되어 있으며, 모듈-LWE는 환(Ring) 안에서 n차 다항식을 파라미터 k에 따라 kXk의 다항식 행렬 계산을 수행한다. 본 발명의 실시예에 따른 모듈-LWE의 특성에 따라 고정된 작은 값의 파라미터 n = 256과 q =3329를 사용하며 보안 등급(Security level) 5에 해당되는 파라미터 k = 4를 기반으로 설계되었다. 또한, 암호학적 해시 알고리즘인 SHA-3(Secure Hash Algorithm-3)를 이용하여 시드(Seed)를 생성할 SHA3-512와 출력 길이를 확장할 수 있는 함수인 SHAKE128과 SHAKE256을 사용하고 다항식 행렬을 생성하는 거부 샘플러(Rejection Sampler)와 샘플링 과정에 대한 효율적 구현의 어려움과 타이밍 공격으로부터 보호하기 위해 중심 이항 분포(Centered Binomial Distribution)를 이용하여 비밀키와 에러를 생성하는 이항 샘플러(Binomial Sampler)를 사용하는 방법과 하드웨어 구조를 제안한다. 또한, NTT 다항식 곱셈기의 하드웨어 복잡도를 줄이고 클럭 속도를 높이기 위해, NTT 전처리(Preprocess) 과정을 줄여서 스테이지(stage)를 감소시키는 방법, 2-병렬 구조, 파이프라인 구조, 그리고 비트-역순(Bit-Reverse) 연산을 제거하는 방법을 제안한다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0022] 도 1은 본 발명의 일 실시예에 따른 모듈-LWE 기반 암복호화 시스템의 구성을 설명하기 위한 도면이다.

[0023] 제안하는 양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템은 키생성 프로세서(Key Generator Processor)(110), 암호화 프로세서(Encryptor Processor)(120) 및 복호화 프로세서(Decryptor Processor)(130)를 포함한다.

[0024] 본 발명의 실시예에 따른 키생성 프로세서(Key Generator Processor)(110)는 PRNG 코어에서 생성한 난수들을 거부 샘플러(Rejection Sampler) 및 이항 샘플러(Binomial Sampler)를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기((Point-Wise Multiplier))와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 비밀키 인코더(Encoder)를 이용하여 암호화 프로세서(120) 및 복호화 프로세서(130)에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력한다.

[0025] 본 발명의 실시예에 따른 키생성 프로세서(110)는 SHA3 해시를 이용한 PRNG 코어에서 생성한 의사난수를 이용하여 다항식 행렬과 에러를 생성하게 되고 NTT 연산을 수행한 뒤 곱셈 연산과 덧셈 연산을 통해 공개키와 비밀키를 생성한다.

[0026] 본 발명의 실시예에 따른 암호화 프로세서(Encryptor Processor)(120)는 사용자가 암호화 하고자 하는 메시지와 상기 키생성 프로세서(110)로부터 공개키를 수신하여 상기 메시지를 인코딩하여 1비트 단위로 마스킹하고, 상기 공개키를 디코더를 통해 공개 시드(Publicseed)와 $\hat{t}$로 분리하여 출력한 값을 키 생성 과정과 동일하게 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스(Compress)를 이용하여 암호문을 압축 및 인코딩하여 출력한다.

[0027] 본 발명의 실시예에 따른 복호화 프로세서(Decryptor Processor)(130)는 암호화 프로세서(120)로부터 암호문을 수신하여 디컴프레스(Decompress)를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서(110)로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더(Decoder)를 이용하여 원본 메시지를 출력한다.

[0028] 모듈-LWE 문제는 링-LWE 문제에 기반한다. 다항식 환(Ring)은 $R_q = Z_q/\langle x^n + 1\rangle$ 안에서 정의되며, 다항식

a(x)와 b(x)는 수학식1과 같이 나타낼 수 있다:

[0029] $$a(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}$$ 수학식1-1

[0030] $$b(x) = b_0 + b_1x + b_2x^2 + \ldots + b_{n-1}x^{n-1}$$ 수학식1-2

[0031] 각 다항식 곱셈의 빠른 연산을 위해 NTT 연산을 수행한다. $\tilde{a} = NTT(a)$는 수학식2와 같이 나타낼 수 있다:

[0032] $$\tilde{a}[i] = \sum_{j=0}^{n-1} a[j]\omega^{ij} mod\ q\ ,\ for\ i = 0.,\ 1,\ \ldots,\ n-1$$ 수학식2

[0033] 벡터(Vector) $\mathbf{a} = \mathbf{a_0}, \ldots, \mathbf{a_{n-1}}$는 다항식 a(x)의 계수이다. n은 다항식의 계수로 1의 n제곱근(root of unity)의 값을 가진다. q는 $\mathbf{q} \equiv \mathbf{1}\ \mathbf{mod}\ \mathbf{n}$을 만족하는 소수(prime number)인 q=3329를 사용한다. 수학식 1에서 n 과 w는 $\mathbf{n} * \mathbf{n}^{-1} \equiv 1\ mod\ q$, $w * w^{-1} \equiv 1\ \mathbf{mod}\ \mathbf{q}$의 조건을 만족하는 숫자로 구성된다.

[0034] 다항식에 대한 INTT 연산은 $b = INTT(\tilde{a})$라고 정의할 때, 수학식3과 같이 나타낸다:

[0035] $$b[i] = n^{-1}\sum_{j=0}^{n-1} \tilde{a}[j]\omega^{-ij} mod\ q\ ,\ for\ i = 0.,\ 1,\ \ldots,\ n-1$$ 수학식3

[0036] 모듈-LWE는 링-LWE를 일반화하여 파라미터 k에 따라 다항식 행렬을 생성한다. 다항식 행렬 $\hat{\mathbf{A}} = R_q^{k\times k}$로 쓸 수 있으며 k값에 따라 확장성이 용이하다.

[0037] 본 발명의 실시예에 따른 양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템의 키생성 프로세서(110), 암호화 프로세서(120) 및 복호화 프로세서(130)에 관하여 도 2 내지 도 3을 참조하여 더욱 상세히 설명한다.

[0039] 도 2는 본 발명의 일 실시예에 따른 키생성 프로세서를 나타내는 도면이다.

[0040] 본 발명의 실시예에 따른 키생성 프로세서는 PRNG 코어(210), 거부 샘플러(Rejection Sampler)(220), 이항 샘플러(Binomial Sampler)(230), NTT 코어(240), 포인트-와이즈 곱셈기 및 덧셈기(250) 및 비밀키 인코더(Encoder)(260)를 포함한다.

[0041] 본 발명의 실시예에 따른 PRNG 코어(210)는 Keccak 알고리즘을 사용한 SHA3 해시로 거부 샘플러(220)와 이항 샘플러(230)에 사용될 의사난수를 생성한다.

[0042] 본 발명의 실시예에 따른 거부 샘플러(220)는 PRNG 코어(210)로부터 의사난수를 입력 받아 공개키 생성에 필요한 다항식 행렬 생성에 사용한다.

[0043] 본 발명의 실시예에 따른 이항 샘플러(230)는 PRNG 코어(210)에서 생성한 의사난수를 이용하여 공개키 생성 및 비밀키 생성에 사용하기 위한 에러 계수를 생성한다.

[0044] 본 발명의 실시예에 따른 NTT 코어(240)는 이항 샘플러(230)의 출력을 입력 받아 NTT 연산을 수행한다.

[0045] 본 발명의 실시예에 따른 포인트-와이즈 곱셈기 및 덧셈기(250)는 연산 과정에 사용되고, 행렬 곱셈연산을 위해 베이스 곱셈기와 누산기를 포함한다.

[0046] 본 발명의 실시예에 따른 비밀키 인코더(260)는 연산된 결과를 직렬화(Serialize)하여 공개키와 비밀키를 생성한다.

[0047] 본 발명의 실시예에 따르면, PRNG 코어(210)는 Keccak 알고리즘을 사용한 SHA3 해시를 사용하며 공개 시드(Publicseed)와 노이즈 시드(Noiseseed)를 생성하는 SHA3-512와 출력이 가변 길이를 가지는 SHAKE128과 SHAKE256을 사용하여 의사난수를 생성한다. PRNG 코어(210)를 통해서 생성된 출력은 각각 거부 샘플러(220)에서 다항식 행렬 $\hat{\mathbf{A}}$를 생성하고 이항 샘플러에서 에러 s와 e 를 생성하여 NTT 연산을 수행한다. 생성된 행렬과 벡터를 다음과 같이 $\hat{\mathbf{t}} = \hat{\mathbf{A}}\hat{\mathbf{s}} + \hat{\mathbf{e}}$ 연산한다. 생성된 $\hat{\mathbf{t}}$를 인코딩하여 공개키를 생성하고 $\hat{\mathbf{s}}$를 인코딩하여 비밀키를 생성한다.

[0049] 도 3은 본 발명의 일 실시예에 따른 암호화 프로세서를 나타내는 도면이다.

[0050] 본 발명의 실시예에 따른 암호화 프로세서는 메시지 인코더(Encoder)(310), 공개키 디코더(Decoder)(320), 거부 샘플러(Rejection Sampler)(330), 이항 샘플러(Binomial Sampler)(340), NTT 코어(350), 포인트-와이즈 곱셈기 및 덧셈기(361, 362, 363, 364, 365), 리덕션(Reduction)(380) 및 컴프레스(Compress)(390)를 포함한다.

[0051] 본 발명의 실시예에 따른 메시지 인코더(310)는 사용자가 입력한 메시지를 수신하여 암호화 프로세서에 적합하게 메시지를 1비트 단위로 마스크 해주는 인코더이다.

[0052] 본 발명의 실시예에 따른 공개키 디코더(320)는 공개키를 수신하여 암호화에 적합하게 공개키를 역-직렬화(De-Serialize)한다.

[0053] 본 발명의 실시예에 따른 거부 샘플러(330)는 공개키 디코더(320)의 출력을 입력 받아 공개키 생성과 암호화에 필요한 다항식 행렬을 생성한다.

[0054] 본 발명의 실시예에 따른 이항 샘플러(340)는 비밀키 생성과 암호화에 필요한 에러 계수를 생성한다.

[0055] 본 발명의 실시예에 따른 NTT 코어(350)는 이항 샘플러(340)의 출력을 입력 받아 NTT 연산을 수행한다.

[0056] 본 발명의 실시예에 따른 포인트-와이즈 곱셈기 및 덧셈기(361, 362, 363, 364, 365)는 거부 샘플러(330)와 이항 샘플러(340)의 값을 연산하고, 행렬 곱셈 연산을 위해 베이스 곱셈기와 누산기를 포함한다.

[0057] 본 발명의 실시예에 따른 INTT 코어는 포인트-와이즈 곱셈기를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행한다.

[0058] 본 발명의 실시예에 따른 리덕션(380)은 곱셈 연산 결과에 대해 몽고메리 리덕션 연산을 수행하는 MR1 및 덧셈 연산 결과에 대해 바렛 리덕션 연산을 수행하는 MR2을 포함한다.

[0059] 본 발명의 실시예에 따른 컴프레스(390)는 리덕션된 데이터를 압축 및 인코딩하여 암호문으로 출력한다.

[0060] 본 발명의 실시예에 따르면, 원본 메시지를 입력 받아 메시지 인코더(310)를 통해 메시지를 마스킹하고, 키생성 프로세서로부터 전달 받은 공개키를 공개키 디코더(320)를 통해 공개 시드(publicseed)와 $\hat{t}$으로 분리하여 키생성 프로세스 과정과 동일하게 거부 샘플러(330)와 이항 샘플러(340)로 다항식 행렬 $\hat{A}$와 에러 r, e1, e2를 생성한다. NTT 코어 및 INTT 코어와 포인트-와이즈 곱셈기를 사용하여 $\mathrm{INTT}(\hat{A}\cdot\hat{r})+e1$과 $\mathrm{INTT}(\hat{A}\cdot\hat{r})+e2$ 연산을 수행한 결과를 리덕션과 컴프레스 과정을 통해 데이터의 크기를 줄이고 압축하여 최종적으로 암호문을 출력한다.

[0062] 도 4는 본 발명의 일 실시예에 따른 복호화 프로세서를 나타내는 도면이다.

[0063] 본 발명의 실시예에 따른 복호화 프로세서는 암호문 디컴프레스(Decompress)(410), 비밀키 디코더(Decoder)(420), NTT 코어(430), 포인트-와이즈 곱셈기 및 뺄셈기(440, 460) 및 메시지 디코더(Decoder)(470)를 포함한다.

[0064] 본 발명의 실시예에 따른 암호문 디컴프레스(410)는 압축된 암호문을 수신하여 복호화 연산에 사용 가능하도록 데이터의 압축을 해제한다.

[0065] 본 발명의 실시예에 따른 비밀키 디코더(420)는 비밀키를 수신하여 암호화에 적합하게 데이터를 역-직렬화(De-Serialize)한다.

[0066] 본 발명의 실시예에 따른 NTT 코어(430)는 암호문 디컴프레스(410)의 출력을 입력 받아 NTT 연산을 수행한다.

[0067] 본 발명의 실시예에 따른 포인트-와이즈 곱셈기 및 뺄셈기(440, 460)는 복호화 연산에 필요한 행렬 곱셈연산을 위해 베이스 곱셈기와 누산기를 포함한다.

[0068] 본 발명의 실시예에 따른 INTT 코어(450)는 포인트-와이즈 곱셈기(440)를 통해 연산된 값을 입력 받아 역-NTT 연산을 수행한다.

[0069] 본 발명의 실시예에 따른 메시지 디코더(470)는 연산된 결과로 얻은 메시지의 마스크를 뺄셈기(460)를 통해 제거하여 원본 메시지를 최종 출력한다.

[0070] 본 발명의 실시예에 따르면, 복호화 프로세서는 암호문과 비밀키를 입력으로 받아 원래의 메시지로 복호화하는 과정으로 암호문 디컴프레스(410)를 통해 압축을 해제한 u, v와 디코더를 통해 역-직렬화(De-Serialize) 과정을

거친 $\hat{s}$를 이용하여 NTT 도메인에서 $u \cdot \hat{s} - v$ 연산을 수행하고 얻어진 결과를 메시지 디코더(470)를 이용하여 마스킹된 메시지를 복원하여 원본 메시지를 출력하게 된다.

[0072] 도 5는 본 발명의 일 실시예에 따른 모듈-LWE 기반 암복호화 시스템의 암호화 방법을 설명하기 위한 흐름도이다.

[0073] 본 발명의 일 실시예에 따른 양자내성암호를 위한 모듈-LWE 기반 암호 프로세서 시스템의 키생성 프로세서(Key Generator Processor), 암호화 프로세서(Encryptor Processor) 및 복호화 프로세서(Decryptor Processor)는 도 5의 단계들(510~530)을 수행하기 위해 구성될 수 있다.

[0074] 단계(510)에서, 키생성 프로세서가 PRNG 코어로 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 인코더를 이용하여 암호화 프로세서 및 복호화 프로세서에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력한다.

[0075] 단계(520)에서, 암호화 프로세서가 사용자가 암호화 하고자 하는 메시지와 상기 키생성 프로세서로부터 공개키를 수신하여 상기 메시지를 인코딩하고, 상기 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력한다.

[0076] 단계(530)에서, 복호화 프로세서 상기 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 상기 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더를 이용하여 원본 메시지를 출력한다.

[0078] 도 6은 본 발명의 일 실시예에 따른 PRNG 코어를 나타내는 도면이다.

[0079] 본 발명의 실시예에 따른 PRNG 코어는 Keccak 알고리즘을 사용하며 SHA3-512, SHAKE-128, SHAKE-256이 가능하도록 하며 샘플링의 입력 비트에 따라 출력 비트의 길이 조정이 가능하다.

[0081] 도 7은 본 발명의 일 실시예에 따른 거부 샘플러를 나타내는 도면이다.

[0082] 본 발명의 실시예에 따른 거부 샘플러는 PRNG 코어의 출력을 16비트 단위로 입력으로 받아 균일한 임의의 정수 modulo q 를 생성하는 모듈로 19q보다 작은 값을 통과시켜 바렛 리덕션을 수행한다.

[0083] 본 발명의 실시예에 따르면, 해당 동작을 파라미터 n = 256, k = 4 에 따라 256개씩 4X4=16번 거부 샘플링을 반복하여 다항식 행렬을 샘플링 한다.

[0085] 도 8은 본 발명의 일 실시예에 따른 이항 샘플러를 나타내는 도면이다.

[0086] 본 발명의 실시예에 따른 이항 샘플러는 PRNG 코어의 출력을 8비트 단위로 입력 받아서 중심이항분포(Centered Binomial Distribution)에 따라 계수(Coefficients) 값을 생성한다. 출력되는 계수 값의 범위는 -2~2사이의 값이 출력된다.

[0088] 도 9는 본 발명의 일 실시예에 따른 2-병렬 NTT 코어를 나타내는 도면이다.

[0089] 본 발명의 실시예에 따르면, 모든 데이터는 2-병렬 구조로 연산을 수행하며, NTT 코어는 총 7개의 스테이지(stage)로 구성되어 있으며, 각각의 스테이지는 PE(Processing Element)로 구성되어 있다.

[0091] 도 10은 본 발명의 일 실시예에 따른 NTT 코어의 PE(Processing Element)를 나타내는 도면이다.

[0092] 본 발명의 실시예에 따른 NTT의 PE는 곱셈기, 몽고메리 리덕션(Modulo Reduction1; MR1), 바렛 리덕션(Modulo Reduction2; MR2)과 버터플라이 유닛(Butterfly Unit; BU)으로 이루어져 있으며, PE의 입력 값과 ROM에 저장된 ω^i(zeta)값과 곱셈 연산 후에 순서대로 몽고메리 리덕션, 버터플라이 유닛 연산, 바렛 리덕션 연산을 수행하여 출력한다. 출력 값은 최종적으로 바렛 리덕션으로 인해 0~q사이의 값을 출력한다.

[0094] 도 11은 본 발명의 일 실시예에 따른 2-병렬 INTT 코어를 나타내는 도면이다.

[0095] 본 발명의 실시예에 따르면, 모든 데이터는 2-병렬 구조로 연산을 수행하며, INTT 코어는 총 7개의 스테이지로

구성되어 있고, 각각의 스테이지는 PE(Processing Element)로 구성되어 있다.

[0097] 도 12는 본 발명의 일 실시예에 따른 INTT 코어의 PE(Processing Element)를 나타내는 도면이다.

[0098] 본 발명의 실시예에 따른 INTT의 PE는 곱셈기, 몽고메리 리덕션, 바렛 리덕션과 버터플라이 유닛으로 이루어져 있다. INTT의 PE는 NTT의 PE와는 반대되는 구조로 입력 값을 버터플라이 유닛과 바렛 리덕션 연산한 값을 ROM에 저장된 ω^i (zeta)값과 곱셈연산 후에 몽고메리 리덕션 연산을 수행하여 출력한다.

[0100] 도 13은 본 발명의 일 실시예에 따른 몽고메리(Montgomery) 리덕션 연산을 수행하는 MR1(Modulo Reduction 1)를 나타내는 도면이다.

[0101] 본 발명의 실시예에 따르면 MR1은 곱셈으로 인해 커진 값을 줄여주기 위해 사용한다. 입력에 대해 $a \cdot R^{-1} \text{ modulo } q$ 값인 -q+1 ~ q-1 사이의 값을 출력한다. 여기서 파라미터 q와 같이 사용되는 Inverse q 값은 $q^{-1} \bmod 2^{16}$ 값인 62209를 사용한다.

[0103] 도 14는 본 발명의 일 실시예에 따른 바렛(Barrett) 리덕션 연산을 수행하는 MR2(Modulo Reduction 2)를 나타내는 도면이다.

[0104] 본 발명의 실시예에 따른 MR2는 덧셈 연산으로 커진 결과를 환(Ring) 내의 값으로 줄여 주기 위해 바렛 리덕션을 수행한다. 입력에 대해 a modulo q 값인 0 ~ q 사이의 값을 출력한다. 여기서 사용되는 v는 $2^{26}/q + 1$ 인 20159를 사용한다.

[0106] 도 15는 본 발명의 일 실시예에 따른 인코더를 나타내는 도면이다.

[0107] 본 발명의 실시예에 따른 인코더는 연산된 결과를 8비트 단위로 직렬화하여 공개키 및 비밀키로 인코딩하는데 사용된다.

[0109] 도 16은 본 발명의 일 실시예에 따른 디코더를 나타내는 도면이다.

[0110] 본 발명의 실시예에 따른 디코더는 인코딩된 공개키 및 비밀키를 13비트 단위로 역-직렬화(De-Serialization)하여 암호화 및 비밀키의 연산에 사용 가능하도록 디코딩하는데 사용된다.

[0112] 도 17은 본 발명의 일 실시예에 따른 컴프레스를 나타내는 도면이다.

[0113] 본 발명의 실시예에 따른 컴프레스는 암호화 프로세서 내의 연산 결과를 CE(Compress Element) 연산을 수행하고 8비트 단위로 직렬화를 통해 암호문의 크기를 압축하고 인코딩하는 역할을 수행한다.

[0115] 도 18은 본 발명의 일 실시예에 따른 컴프레스의 CE(Compress Element)를 나타내는 도면이다.

[0116] 본 발명의 실시예에 따른 CE는 입력의 <<3에 대해 q/2 값인 1664를 더해주고 q로 나눈 결과를 하위 5비트만을 사용하여 출력하는 구조를 가지고 있다. 여기서 나눗셈은 많은 연산 시간과 자원을 소모하기 때문에 바렛 리덕션 알고리즘을 이용하여 계산한다.

[0118] 도 19는 본 발명의 일 실시예에 따른 디컴프레스를 나타내는 도면이다.

[0119] 본 발명의 실시예에 따른 디컴프레스는 수신 받은 암호문을 역-직렬화하고 각각 DE(Decompress Element) 연산을 수행하여 복호화 연산이 가능하도록 압축을 해제하고 디코딩하는 역할을 수행한다.

[0121] 도 20은 본 발명의 일 실시예에 따른 디컴프레스의 DE(Decompress Element)를 나타내는 도면이다.

[0122] 본 발명의 실시예에 따른 DE는 입력에 대해 q를 곱해주고 16을 더해준 값을 하위 5비트만을 출력하는 구조로 되어있다.

[0124] 도 21은 본 발명의 일 실시예에 따른 메시지 인코더를 나타내는 도면이다.

[0125] 본 발명의 실시예에 따른 메시지 인코더는 사용자가 입력한 256비트의 메시지를 1비트 단위로 메시지 인코더에서 (q+1)/2 로 마스킹하여 13비트로 인코딩하여 암호화에 사용된다.

[0127] 도 22는 본 발명의 일 실시예에 따른 메시지 디코더를 나타내는 도면이다.

[0128] 본 발명의 실시예에 따른 메시지 디코더는 사용자가 복호화 연산을 수행하여 마스킹된 메시지를 원래의 사용자

가 입력한 메시지로 디코딩을 수행한다. 입력에 대해 <<1 값에 q/2인 1664를 더해주고 q로 나눈 결과를 하위 1 비트만을 출력한다. 총 256번 수행하여 사용자가 입력한 원래 256비트를 출력한다.

[0130] 도 23은 본 발명의 일 실시예에 따른 베이스 곱셈기(Base multiplier)를 나타내는 도면이다.

[0131] 본 발명의 실시예에 따른 베이스 곱셈기는 포인트-와이즈 곱셈기의 하위 모듈로 사용된다. 곱셈을 수행할 각 다항식으로부터 두 개씩 입력으로 받아 Bow-tie 곱셈을 수행한다. 베이스 곱셈기의 연산 과정은 수학식 4와 같이 쓸 수 있다:

[0132] $$r0 = a2 \cdot b2 \cdot Zeta(or\ -Zeta) + a1 \cdot b1$$ 수학식4-1

[0133] $$r1 = a2 \cdot b1 + a1 \cdot b2$$ 수학식4-2

[0134] 곱셈 연산으로 인해 커지는 값은 몽고메리 리덕션 모듈인 MR1을 사용하여 줄여 주고 16비트의 크기로 출력하게 된다.

[0136] 도 24는 본 발명의 일 실시예에 따른 누산기(Accumulator)를 나타내는 도면이다.

[0137] 본 발명의 실시예에 따른 누산기는 포인트-와이즈 곱셈기의 하위 모듈로 사용된다. 모듈-LWE 문제에 기반한 암호는 다항식 행렬의 곱셈에 기반하기 때문에 행렬의 곱셈 방식에 따라 곱셈 연산을 수행한 후에 각 다항식 열을 더해주는 역할을 수행한다.

[0138] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0139] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0140] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

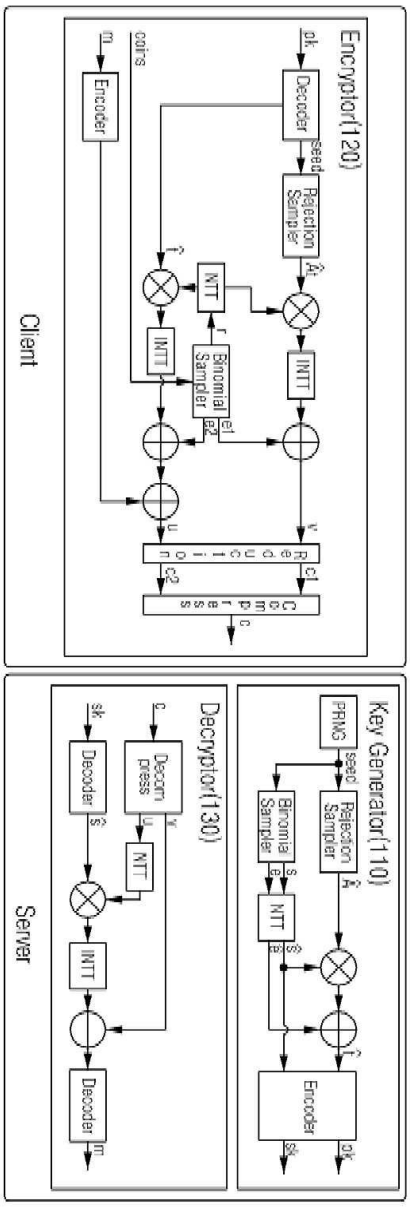
[0141] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과

다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

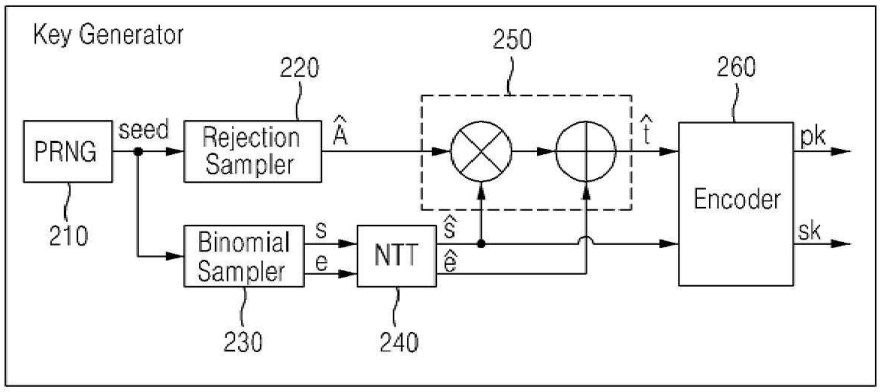
[0142] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

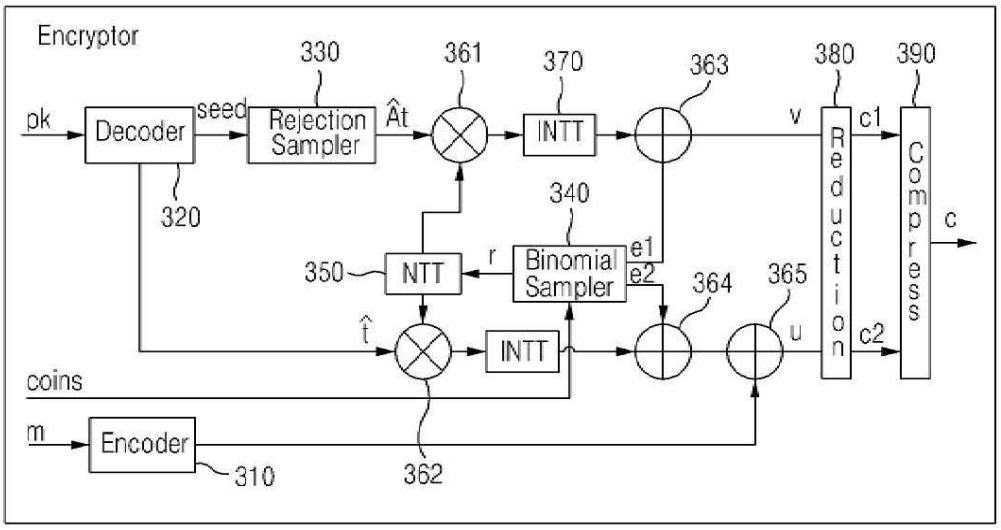
도면1



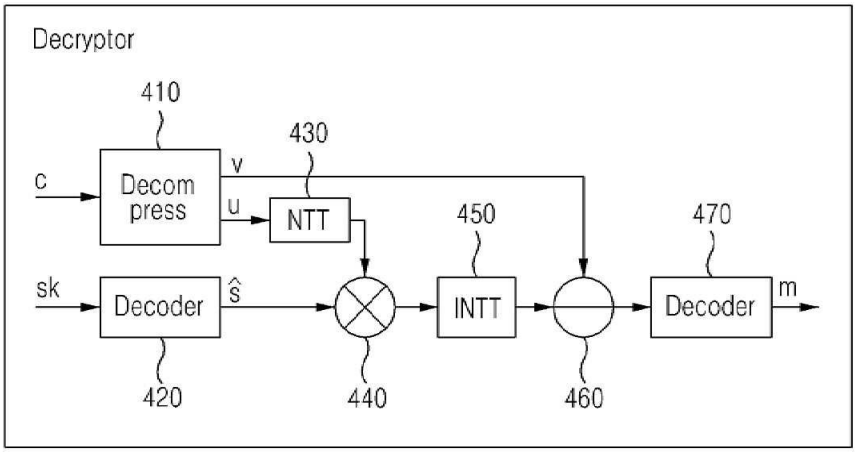
도면2


Key Generator
250
220
260
PRNG
seed
Rejection Sampler
Â
t̂
pk
Encoder
sk
210
Binomial Sampler
s
e
NTT
ŝ
ê
230
240

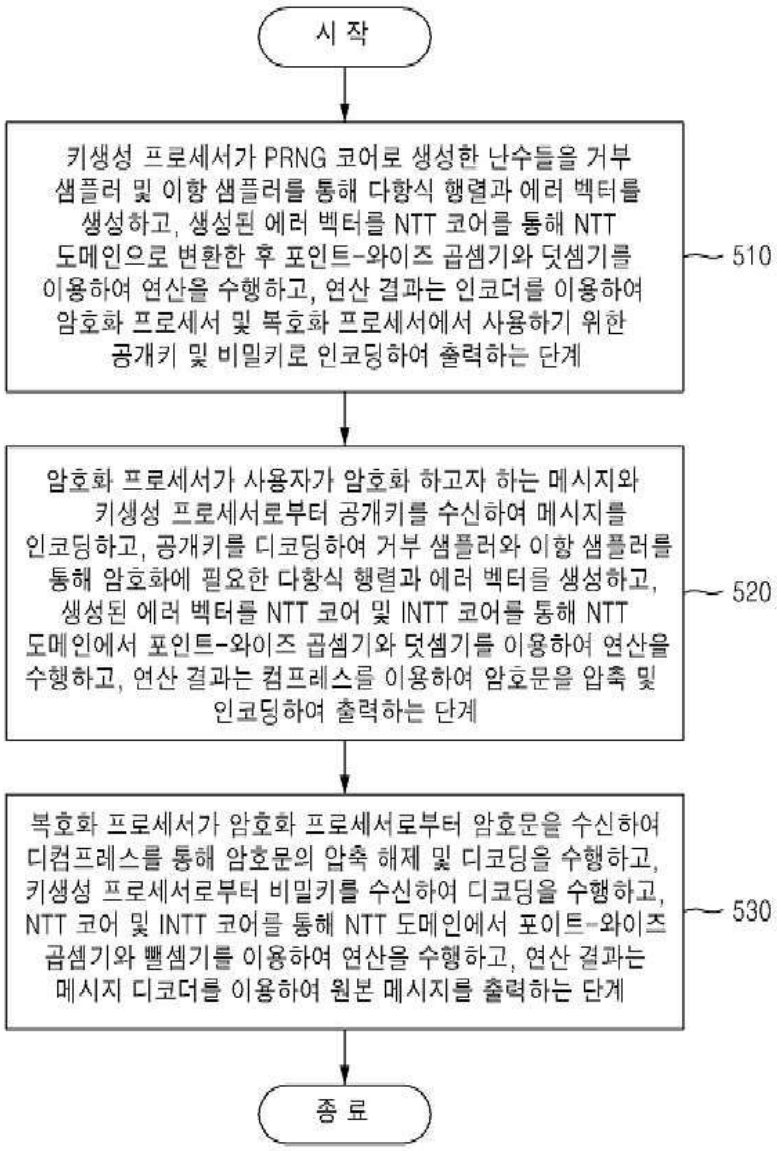
도면3


Encryptor
330
361
370
363
380
390
pk
Decoder
seed
Rejection Sampler
Ât
INTT
v
Reduction
c1
Compress
c
320
340
350
NTT
r
Binomial Sampler
e1
e2
364
365
t̂
INTT
u
c2
coins
m
Encoder
310
362

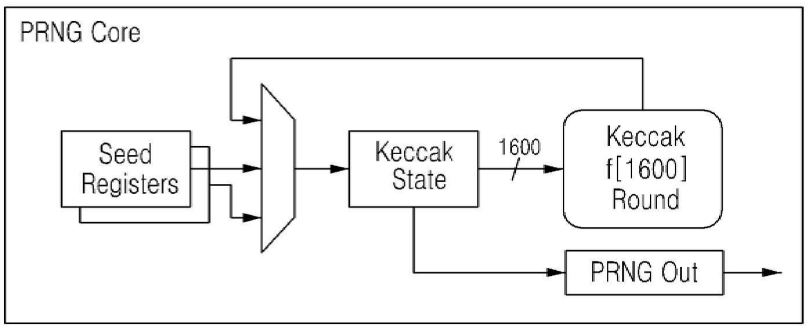
도면4


Decryptor
410
430
c
Decom press
v
u
NTT
450
470
sk
Decoder
ŝ
INTT
Decoder
m
420
440
460

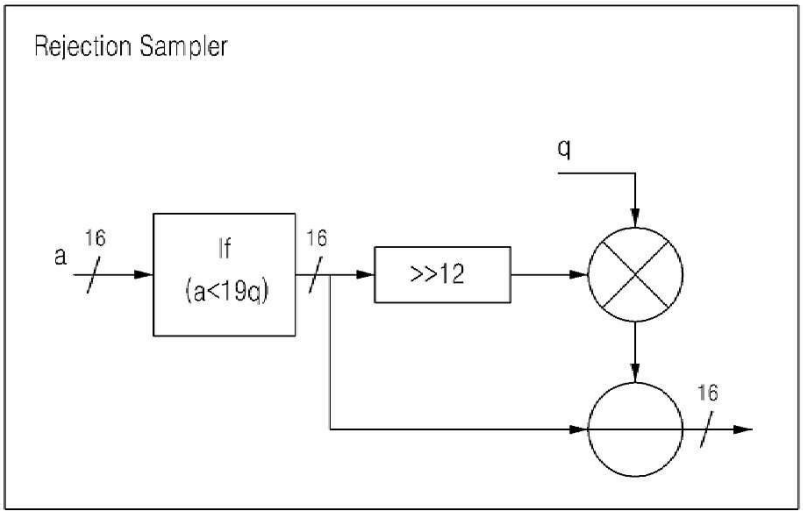
도면5


시 작
키생성 프로세서가 PRNG 코어로 생성한 난수들을 거부 샘플러 및 이항 샘플러를 통해 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어를 통해 NTT 도메인으로 변환한 후 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 인코더를 이용하여 암호화 프로세서 및 복호화 프로세서에서 사용하기 위한 공개키 및 비밀키로 인코딩하여 출력하는 단계
510
암호화 프로세서가 사용자가 암호화 하고자 하는 메시지와 키생성 프로세서로부터 공개키를 수신하여 메시지를 인코딩하고, 공개키를 디코딩하여 거부 샘플러와 이항 샘플러를 통해 암호화에 필요한 다항식 행렬과 에러 벡터를 생성하고, 생성된 에러 벡터를 NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포인트-와이즈 곱셈기와 덧셈기를 이용하여 연산을 수행하고, 연산 결과는 컴프레스를 이용하여 암호문을 압축 및 인코딩하여 출력하는 단계
520
복호화 프로세서가 암호화 프로세서로부터 암호문을 수신하여 디컴프레스를 통해 암호문의 압축 해제 및 디코딩을 수행하고, 키생성 프로세서로부터 비밀키를 수신하여 디코딩을 수행하고, NTT 코어 및 INTT 코어를 통해 NTT 도메인에서 포이트-와이즈 곱셈기와 뺄셈기를 이용하여 연산을 수행하고, 연산 결과는 메시지 디코더를 이용하여 원본 메시지를 출력하는 단계
530
종 료

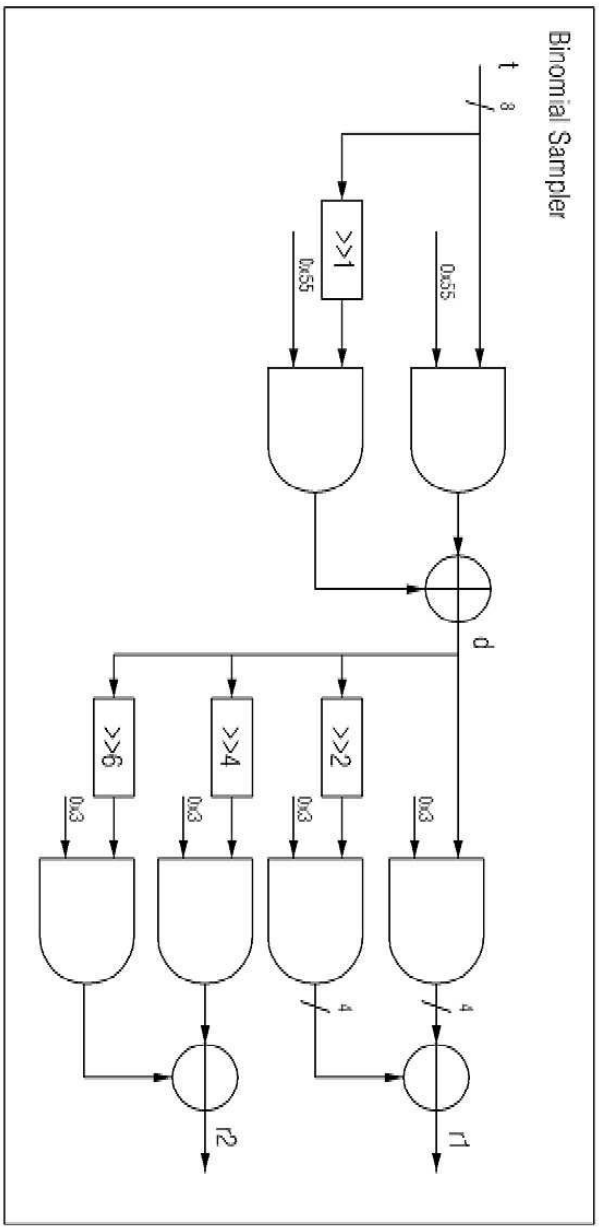
도면6


PRNG Core
Seed Registers
Keccak State
1600
Keccak f[1600] Round
PRNG Out

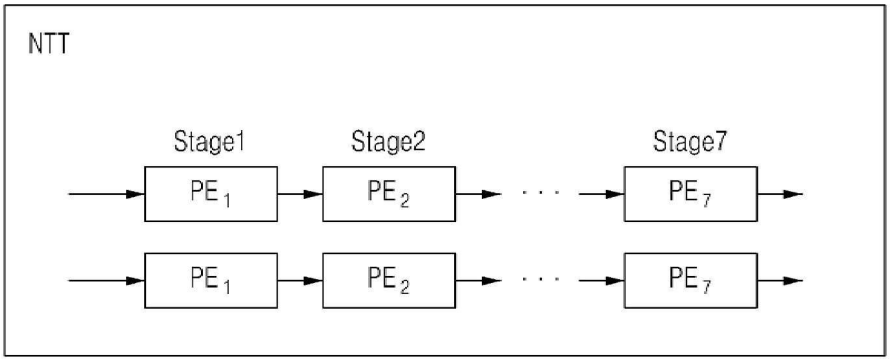
도면7



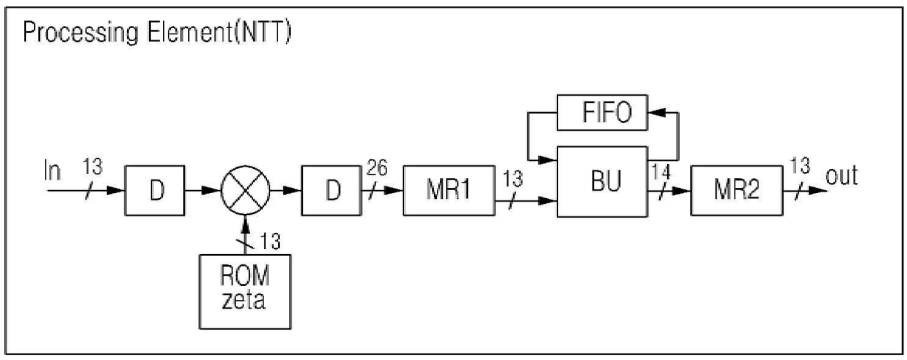
도면8


Binomial Sampler
t
8
>>1
0x55
0x55
d
>>6
>>4
>>2
0x3
0x3
0x3
0x3
4
4
r2
r1

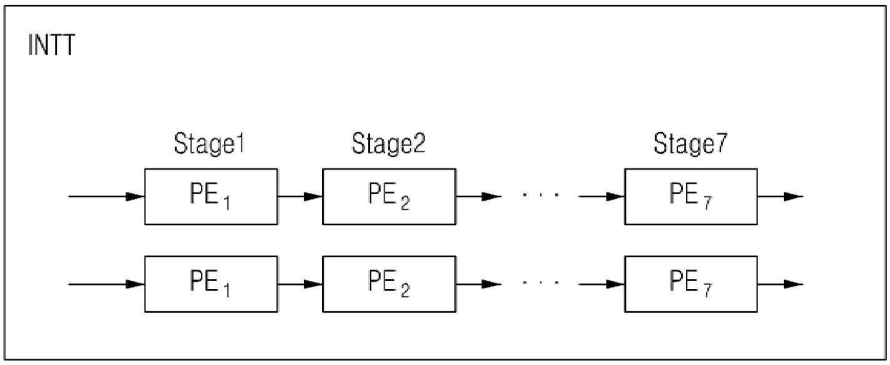
도면9


NTT
Stage1
Stage2
Stage7
PE1
PE2
PE7
PE1
PE2
PE7

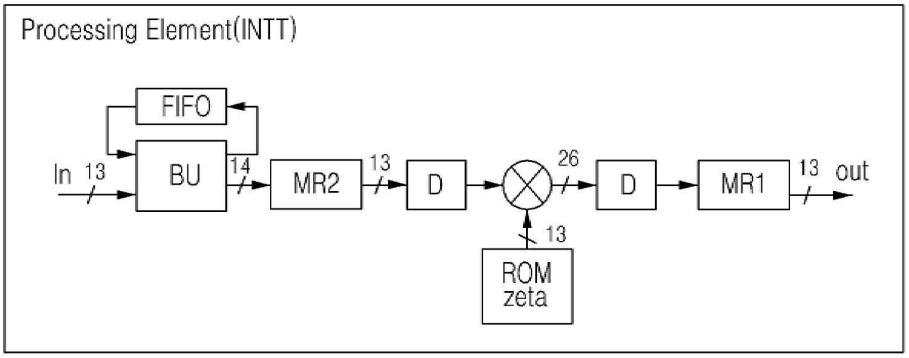
도면10



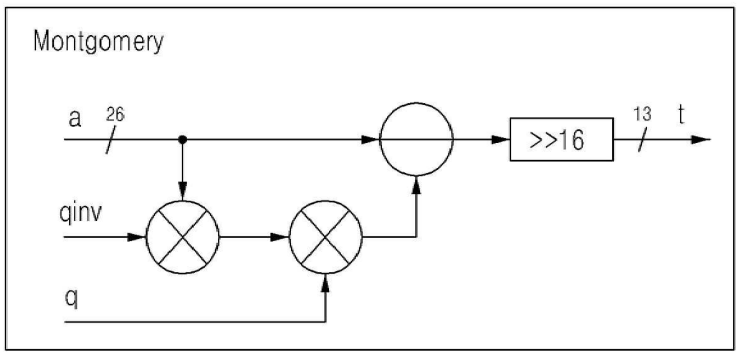
도면11



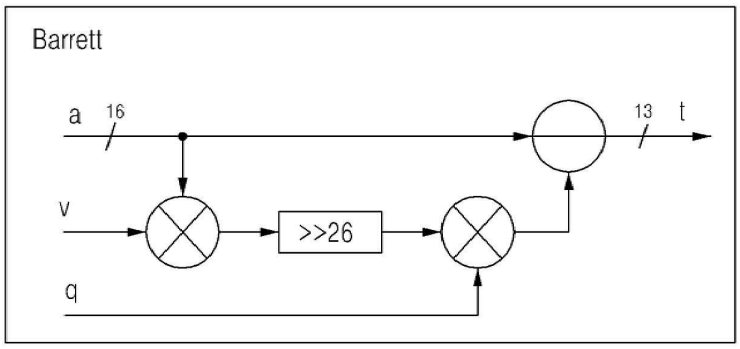
도면12



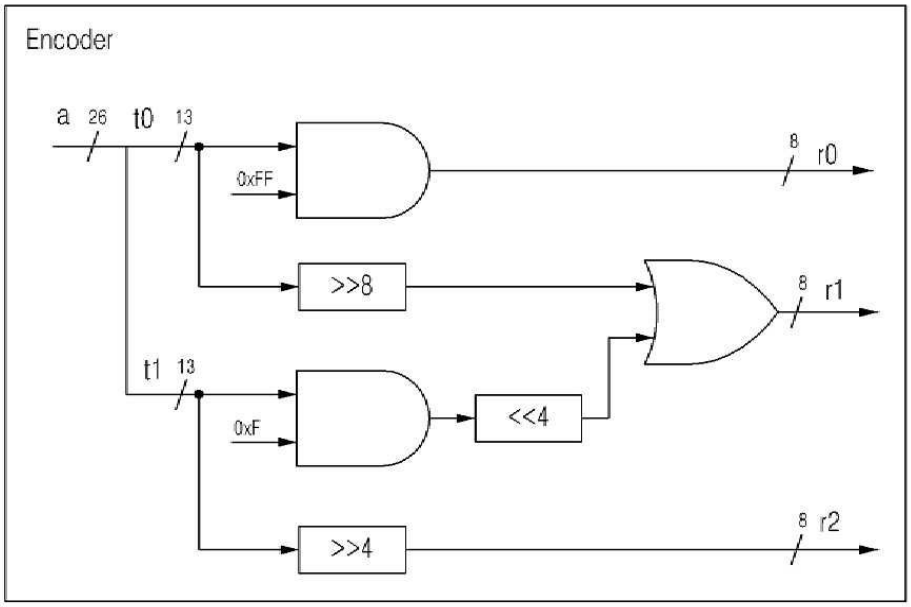
도면13



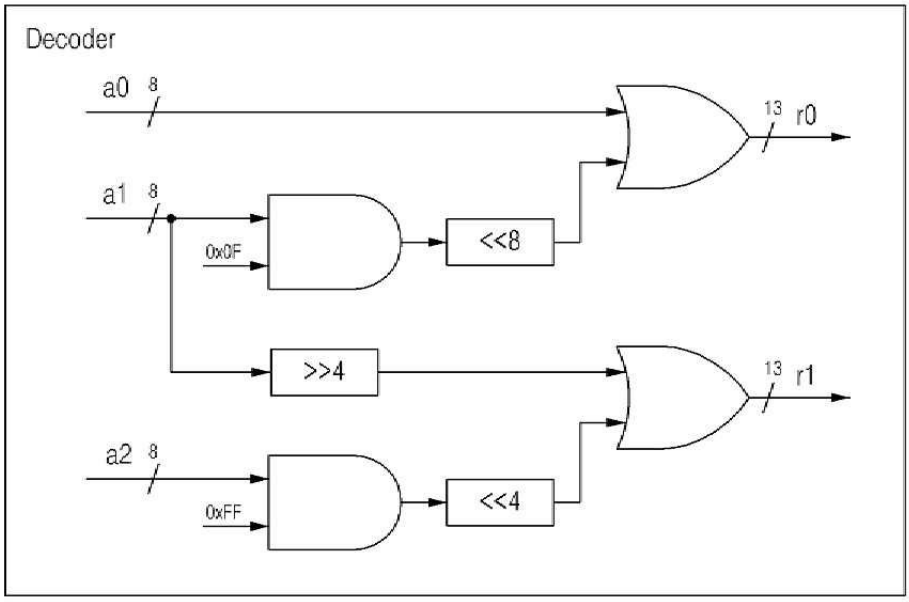
도면14


Barrett
a
16
13
t
v
>>26
q

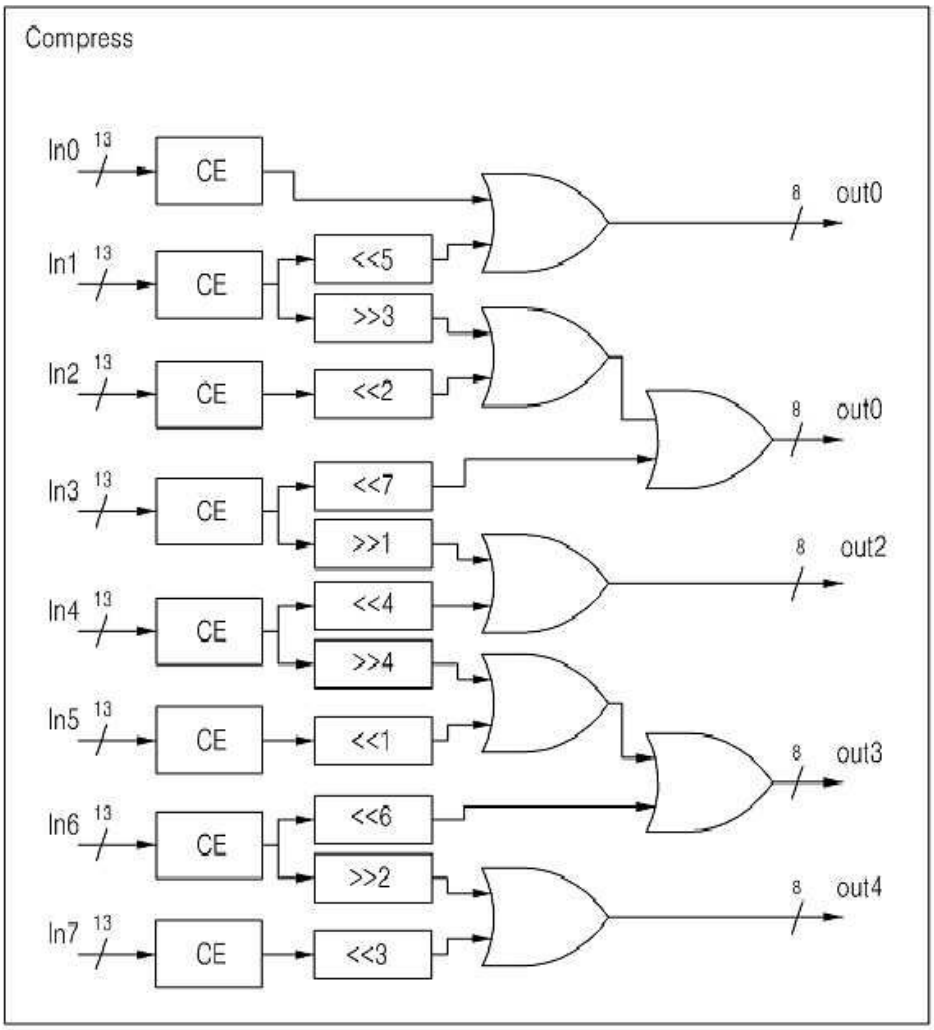
도면15


Encoder
a
26
t0
13
0xFF
8
r0
>>8
8
r1
t1
13
0xF
<<4
>>4
8
r2

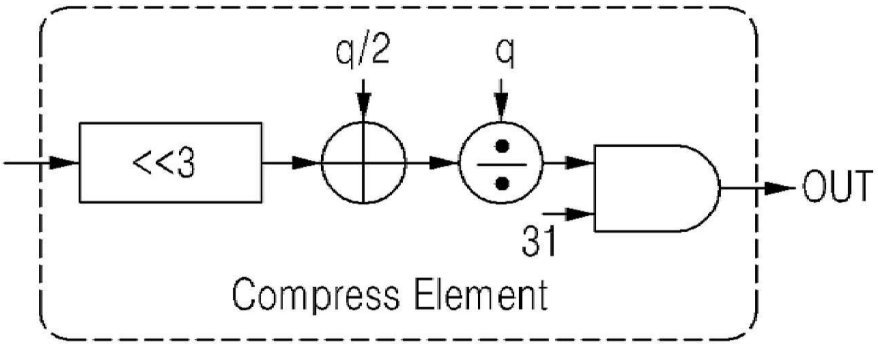
도면16


Decoder
a0
8
13
r0
a1
8
0x0F
<<8
>>4
13
r1
a2
8
0xFF
<<4

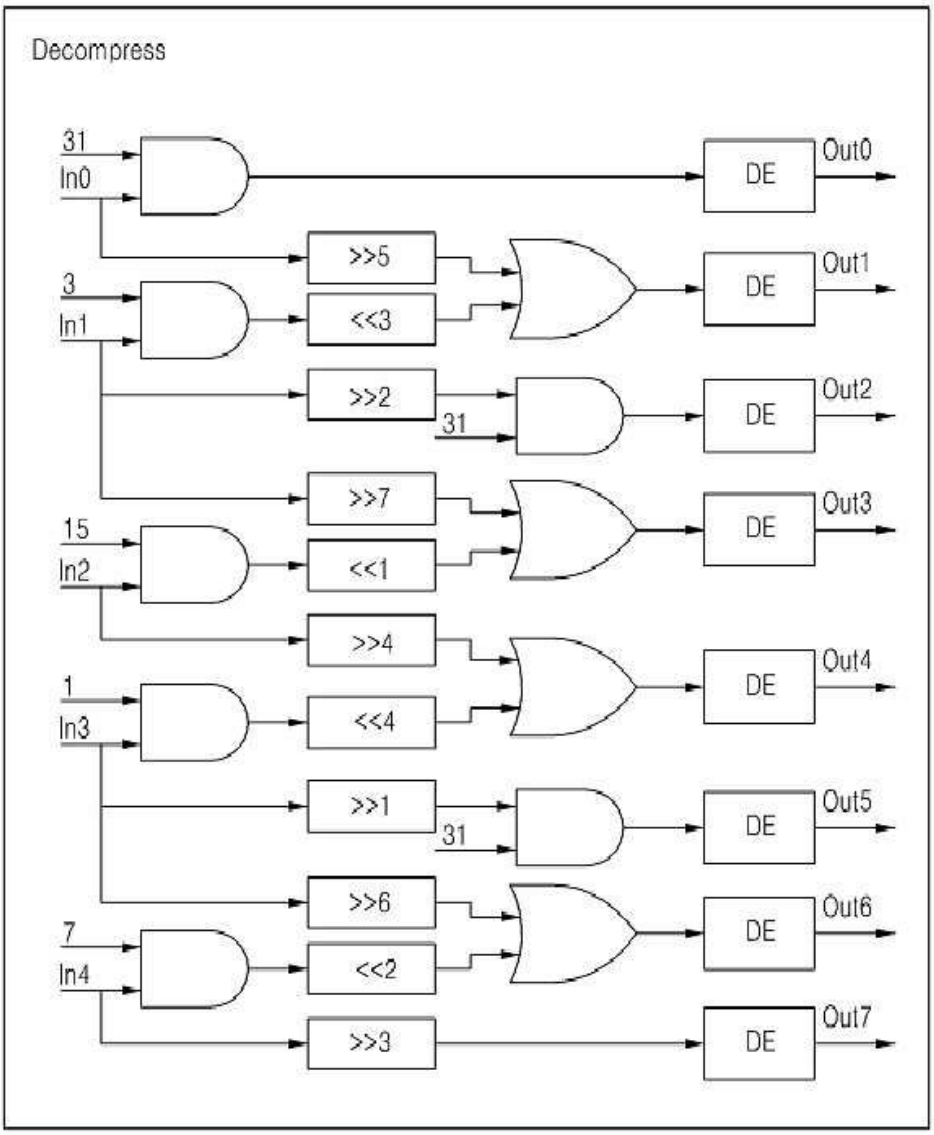
도면17


Compress
In0
13
CE
In1
13
CE
<<5
>>3
In2
13
CE
<<2
In3
13
CE
<<7
>>1
In4
13
CE
<<4
>>4
In5
13
CE
<<1
In6
13
CE
<<6
>>2
In7
13
CE
<<3
8
out0
8
out0
8
out2
8
out3
8
out4

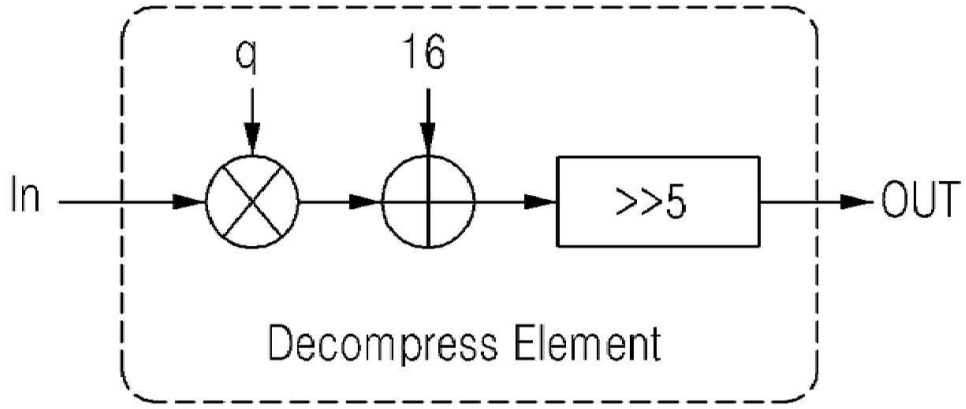
도면18


q/2
q
<<3
OUT
31
Compress Element

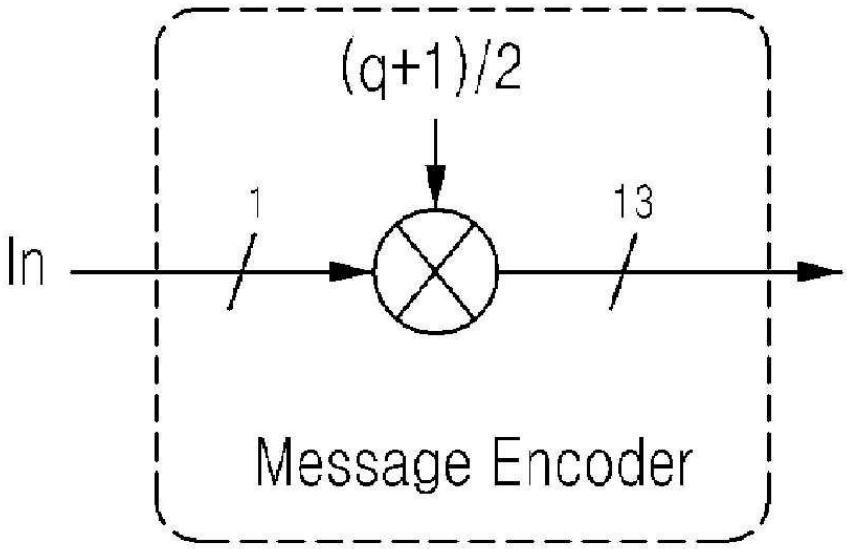
도면19


Decompress
31
In0
DE
Out0
>>5
3
In1
<<3
Out1
>>2
31
Out2
>>7
15
In2
<<1
Out3
>>4
1
In3
<<4
Out4
>>1
31
Out5
>>6
7
In4
<<2
Out6
>>3
Out7

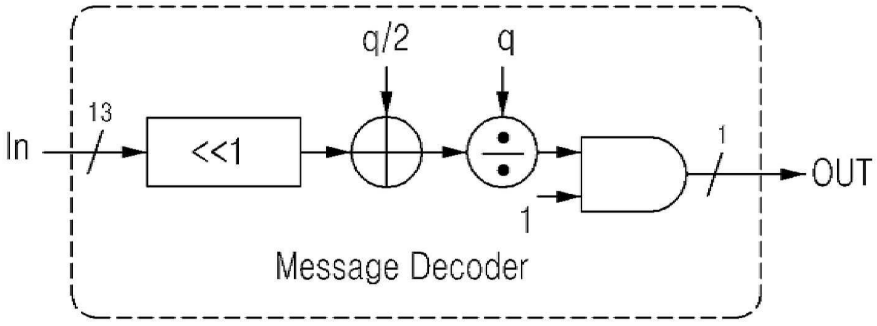
도면20



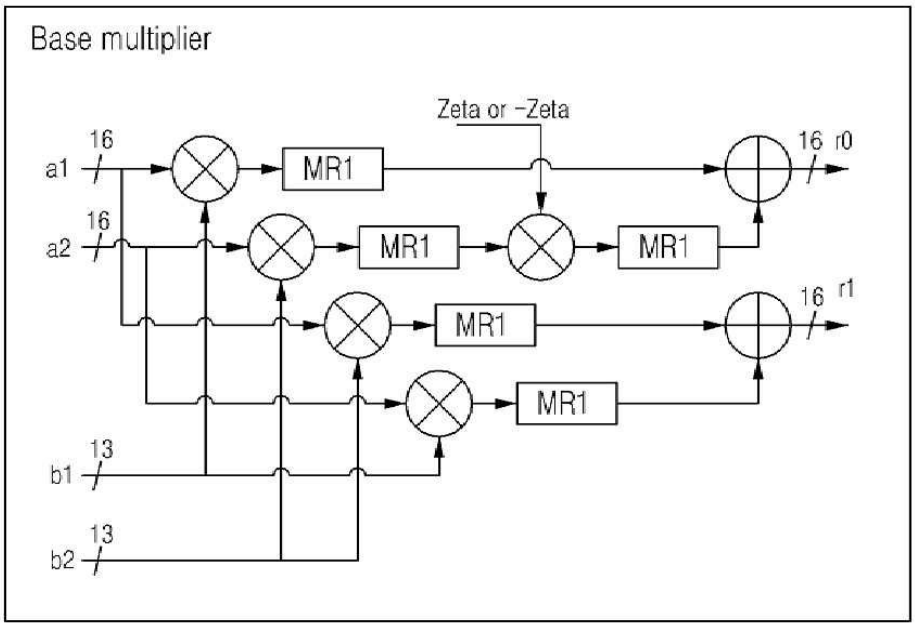
도면21



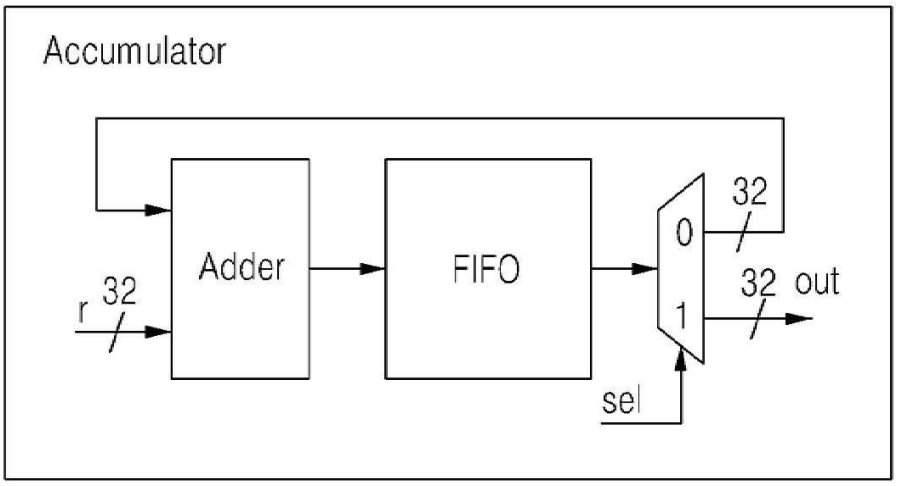
도면22



도면23


Base multiplier
Zeta or -Zeta
a1
16
a2
16
b1
13
b2
13
MR1
MR1
MR1
MR1
MR1
16 r0
16 r1

도면24


Accumulator
Adder
FIFO
r
32
0
1
32
32 out
sel