

명세서

발명의 명칭: 공개키를 이용하여 암호화하는 연산 장치 및 그 암호화 방법

기술분야

- [1] 본 발명은 공개키를 이용하여 암호화하는 연산 장치 및 그 암호화 방법에 대한 것으로, 보다 상세하게는, 에러를 이용하여 공개키를 생성하고, 그 공개키로 암호화를 수행하는 연산 장치 및 그 암호화 방법에 대한 것이다.

배경기술

- [2] 암호화 방법에는 크게 대칭키 암호화 방법과 비대칭키 암호화 방법이 있다. 대칭키 암호화 방법은 암호화에 사용하는 키와 복호화에 사용하는 키가 동일하기 때문에 속도는 빠르지만 키가 외부에 유출되는 경우 보안성이 담보되지 않을 수 있다. 그러한 문제를 해결하기 위하여 비대칭키 암호화 방법이 사용된다. 비대칭키 암호화 방법에서는 제3자에게 공개하는 공개키와, 사용자 본인만 보유하는 비밀키(또는 개인 키)를 사용한다. 사용자는 공개키를 이용하여 메시지를 암호화하여 송신하고, 이를 수신한 수신자는 자신의 비밀키를 이용하여 암호문을 복호화한다.
- [3] 비대칭암호화 방법 중에서 대표적인 것이 1978년에 Ron Rivest, Adi Shamir, Leonard Adleman이 제안한 RSA 방식이다. 하지만, RSA 방식의 비대칭 암호문은 양자 컴퓨터에 의해 손쉽게 풀릴 수 있다는 단점이 있다.
- [4] 한편, 2005년에 Regev는 LWE에 기반한 공개키 암호화 방법을 제안한 바 있다(O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In STOC, LNCS, pages 84-93, 2005). 이 논문의 내용은 본 발명의 기술적 사상과 배치되지 않는 한 본 발명의 설명을 위해 본 명세서의 일부로서 합체된다. Regev가 상기 논문에서 제안한 LWE에 기반한 공개키 암호화 방법 및 그에 기초한 변형 암호화 방법을 본 명세서에서는 "LWE 기반 공개키 암호화 방법"이라고 표기한다.
- [5] 그러나 이 방식은 파라미터의 크기가 크고 속도가 느리기 때문에 사물 인터넷 환경, TLS, 클라우드 컴퓨팅, 서버 연산 등의 실제 용도에 적합하지 않은 단점이 존재한다. 이와 같은 단점을 해결하고자 본 발명자는 가우시안 샘플링(Gaussian sampling)을 이용한 암호화 방법을 2016년 11월 10일에 한국특허출원 제10-2016-0149246호로 출원한 바 있다.
- [6] 그러나 이 방법은 복호화 속도는 빠르지만, 암호화할 때에 가우시안 샘플링을 거치기 때문에 암호화 속도가 느리다는 단점이 있다.

발명의 상세한 설명

기술적 과제

- [7] 본 발명은 이상과 같은 문제점을 해결하기 위하여 고안된 것으로, 본 발명의

목적은 가우시안 샘플링을 하지 않고도 효과적으로 암호화를 할 수 있는 연산 장치 및 그 암호화 방법을 제공함에 있다.

과제 해결 수단

- [8] 이상과 같은 목적을 달성하기 위하여, 본 발명의 일 실시 예에 따른 암호화 방법은, 비밀키를 설정하고, 상기 비밀키 및 에러를 이용하여 공개키를 생성하는 단계 및 메시지에 상기 공개키를 적용한 후, 라운딩 처리를 수행하여, 상기 메시지를 암호화하는 단계를 포함한다. 여기서, 비밀키는 -1, 0, 1 중에서 랜덤하게 조합될 수 있고, 에러의 크기는, 0보다 크고 1보다 작게 설정된 에러 파라미터에 의해 결정될 수 있다.
- [9] 상기 공개키를 생성하는 단계는, 랜덤하게 결정된 값들로 구성된 제1 랜덤 행렬(A)을 산출하는 단계, 각 성분이 -1, 0, 1의 값을 갖는 열 벡터를 행렬 형태로 랜덤하게 조합하여 상기 비밀키(s)를 산출하는 단계, 이산 가우시안 분포 또는 상기 이산 가우시안 분포와 통계적 거리가 가까운 분포로부터 상기 에러(E)를 추출하는 단계, 상기 제1 랜덤 행렬 및 상기 비밀키와, 상기 에러를 모듈러 연산하여 제2 랜덤 행렬(B)을 산출하는 단계, 상기 제1 랜덤 행렬 및 상기 제2 랜덤 행렬을 포함하는 상기 공개키를 산출하는 단계를 포함한다.
- [10] 또한, 상기 메시지를 암호화하는 단계는, 각 성분이 -1, 0, 1의 값을 갖는 랜덤 벡터를 설정하는 단계, 상기 제1 랜덤 행렬 및 상기 랜덤 벡터를 연산한 제1 값, 상기 제2 랜덤 행렬 및 상기 랜덤 벡터를 연산한 결과값에 상기 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출하는 단계, 상기 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 상기 메시지에 대응되는 암호문을 산출하는 단계를 포함한다.
- [11] 또는, 상기 공개키를 생성하는 단계는, 기 설정된 계수를 가지는 다항식의 집합인 링(Ring)을 설정하는 단계, 상기 링으로부터 상기 비밀키를 산출하는 단계, 상기 링으로부터 제1 랜덤 다항식을 산출하는 단계, 이산 가우시안 분포 또는 상기 이산 가우시안 분포와 통계적 거리가 가까운 분포로부터 상기 에러를 추출하는 단계, 상기 제1 랜덤 다항식 및 상기 비밀키에 상기 에러를 모듈러 연산하여 제2 랜덤 다항식을 산출하는 단계, 상기 제1 랜덤 다항식 및 상기 제2 랜덤 다항식을 포함하는 상기 공개키를 설정하는 단계를 포함할 수도 있다.
- [12] 이 경우, 상기 메시지를 암호화하는 단계는, 각 계수가 -1, 0, 1의 값을 가지는 다항식을 랜덤하게 추출하는 단계, 상기 제1 랜덤 다항식 및 상기 다항식을 연산한 제1 값, 상기 제2 랜덤 다항식 및 상기 다항식을 연산한 결과값에 상기 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출하는 단계, 상기 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 상기 메시지에 대응되는 암호문 다항식을 산출하는 단계를 포함할 수 있다.
- [13] 또한, 상기 공개키로 암호화된 다른 메시지가 수신되면, 수신된 다른 메시지를

상기 비밀키로 복호화하는 단계를 더 포함할 수도 있다.

- [14] 한편, 본 발명의 일 실시 예에 따른 연산 장치는, 메모리, 및 비밀키를 설정하고 상기 비밀키 및 에러를 이용하여 공개키를 생성하며, 상기 비밀키 및 상기 공개키를 상기 메모리에 저장하는 프로세서를 포함한다. 여기서, 상기 프로세서는, 암호화할 메시지에 대하여 상기 공개키를 적용한 후 라운딩 처리를 수행하여 상기 메시지를 암호화하며, 상기 비밀키는 -1, 0, 1 중에서 랜덤하게 조합될 수 있고, 에러의 크기는, 0보다 크고 1보다 작게 설정된 에러 파라미터에 의해 결정될 수 있다.
- [15] 상기 프로세서는, 랜덤하게 결정된 벡터들로 구성된 제1 랜덤 행렬(A)을 산출하고, 이산 가우시안 분포 또는 상기 이산 가우시안 분포와 통계적 거리가 가까운 분포로부터 에러를 산출하며, 상기 제1 랜덤 행렬 및 상기 비밀키와, 상기 에러를 모듈러 연산하여 제2 랜덤 행렬(B)을 산출하여, 상기 제1 랜덤 행렬 및 상기 제2 랜덤 행렬을 포함하는 상기 공개키를 설정할 수 있다.
- [16] 상기 프로세서는, 각 성분이 -1, 0, 1의 값을 갖는 랜덤 벡터를 설정하고, 상기 제1 랜덤 행렬 및 상기 랜덤 벡터를 연산한 제1 값, 상기 제2 랜덤 행렬 및 상기 랜덤 벡터를 연산한 결과값에 상기 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출하며, 상기 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 상기 메시지에 대응되는 암호문을 산출할 수도 있다.
- [17] 또는, 상기 프로세서는, 기 설정된 계수를 가지는 다항식의 집합인 링(Ring)을 설정하고, 상기 링으로부터 상기 비밀키 및 제1 랜덤 다항식을 각각 산출하고, 이산 가우시안 분포 또는 상기 이산 가우시안 분포와 통계적 거리가 가까운 분포로부터 에러를 산출하며, 상기 제1 랜덤 다항식 및 상기 비밀키에 상기 에러를 모듈러 연산하여 제2 랜덤 다항식을 산출하여, 상기 제1 랜덤 다항식 및 상기 제2 랜덤 다항식을 포함하는 상기 공개키를 설정할 수도 있다.
- [18] 이 경우, 상기 프로세서는, 각 계수가 -1, 0, 1의 값을 갖는 다항식을 랜덤하게 추출하고, 상기 제1 랜덤 다항식 및 상기 다항식을 연산한 제1 값, 상기 제2 랜덤 다항식 및 상기 다항식을 연산한 결과값에 상기 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출하며, 상기 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 상기 메시지에 대응되는 암호문 다항식을 산출할 수 있다.
- [19] 또는, 상기 메모리에 저장된 상기 공개키를 브로드캐스팅하기 위한 통신부;를 더 포함할 수도 있다. 이 경우, 상기 프로세서는, 상기 브로드캐스팅된 공개키에 의해 암호화된 다른 메시지가 상기 통신부를 통해 수신되면, 수신된 상기 다른 메시지를 상기 비밀키를 이용하여 복호화할 수 있다.
- [20] 한편, 본 발명의 일 실시 예에 따른 기록 매체에서는, 랜덤하게 결정된 값들로 구성된 제1 랜덤 행렬(A)을 산출하는 단계, 각 성분이 -1, 0, 1 중 하나의 값을 갖는 열 벡터를 행렬 형태로 랜덤하게 조합하여 비밀키(s)를 산출하는 단계, 이산

가우시안 분포로부터 에러(E)를 추출하는 단계, 상기 제1 랜덤 행렬 및 상기 비밀키와, 상기 에러를 모듈러 연산하여 제2 랜덤 행렬(B)을 산출하는 단계, 상기 제1 랜덤 행렬 및 상기 제2 랜덤 행렬을 포함하는 공개키를 설정하는 단계를 순차적으로 실행하는 프로그램 코드가 저장될 수 있다.

- [21] 또는, 프로그램 코드는, 각 성분이 -1, 0, 1의 값을 갖는 랜덤 벡터를 설정하는 단계, 상기 제1 랜덤 행렬 및 상기 랜덤 벡터를 연산한 제1 값, 상기 제2 랜덤 행렬 및 상기 랜덤 벡터를 연산한 결과값에 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출하는 단계, 상기 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 상기 메시지에 대응되는 암호문을 산출하는 단계를 추가로 수행할 수도 있다.

발명의 효과

- [22] 이상과 같은 본 발명의 다양한 실시 예들에 따르면, 가우시안 샘플링을 이용하지 않으면서 좀 더 효율적이고 안정적으로 암호화를 수행할 수 있다.

도면의 간단한 설명

- [23] 도 1은 본 발명의 일 실시 예에 따른 암호화 방법을 설명하기 위한 흐름도,
 [24] 도 2는 키 생성 과정의 일 예를 구체적으로 설명하기 위한 흐름도,
 [25] 도 3은 메시지를 암호화하는 과정의 일 예를 구체적으로 설명하기 위한 흐름도,
 [26] 도 4는 키 생성 과정의 다른 예를 구체적으로 설명하기 위한 흐름도,
 [27] 도 5는 메시지를 암호화하는 과정의 다른 예를 구체적으로 설명하기 위한 흐름도,
 [28] 도 6은 본 발명의 일 실시 예에 따른 연산 장치의 구성을 설명하기 위한 흐름도, 그리고,
 [29] 도 7은 본 발명의 일 실시 예에 따른 암호화/복호화를 수행하는 시스템을 설명하기 위한 도면이다.

발명의 실시를 위한 최선의 형태

- [30] 이하에서는 첨부 도면을 참조하여 본 발명에 대해서 자세하게 설명한다. 본 명세서에서 수행되는 정보(데이터) 전송 과정은 필요에 따라서 암호화/복호화가 적용될 수 있으며, 본 명세서 및 특허청구범위에서 정보(데이터) 전송 과정을 설명하는 표현은 별도로 언급되지 않더라도 모두 암호화/복호화하는 경우도 포함하는 것으로 해석되어야 한다. 본 명세서에서 "A로부터 B로 전송(전달)" 또는 "A가 B로부터 수신"과 같은 형태의 표현은 중간에 다른 매개체가 포함되어 전송(전달) 또는 수신되는 것도 포함하며, A로부터 B까지 직접 전송(전달) 또는 수신되는 것만을 표현하는 것은 아니다. 본 발명의 설명에 있어서 각 단계의 순서는 선행 단계가 논리적 및 시간적으로 반드시 후행 단계에 앞서서 수행되어야 하는 경우가 아니라면 각 단계의 순서는 비제한적으로 이해되어야 한다. 즉, 위와 같은 예외적인 경우를 제외하고는 후행 단계로 설명된 과정이 선행단계로 설명된 과정보다 앞서서 수행되더라도 발명의 본질에는 영향이

없으며 권리범위 역시 단계의 순서에 관계없이 정의되어야 한다. 그리고 본 명세서에서 “A 또는 B”은 A와 B 중 어느 하나를 선택적으로 가리키는 것뿐만 아니라 A와 B 모두를 포함하는 것도 의미하는 것으로 정의된다. 또한, 본 명세서에서 “포함”이라는 용어는 포함하는 것으로 나열된 요소 이외에 추가로 다른 구성요소를 더 포함하는 것도 포괄하는 의미를 가진다.

[31] 본 명세서에서는 본 발명의 설명에 필요한 필수적인 구성요소만을 설명하며, 본 발명의 본질과 관계가 없는 구성요소는 언급하지 아니한다. 그리고 언급되는 구성요소만을 포함하는 배타적인 의미로 해석되어서는 아니되며 다른 구성요소도 포함할 수 있는 비배타적인 의미로 해석되어야 한다.

[32] 그리고 본 명세서에서 “값”이라 함은 스칼라값 뿐만 아니라 벡터도 포함하는 개념으로 정의된다.

[33] 후술하는 본 발명의 각 단계의 수학적 연산 및 산출은 해당 연산 또는 산출을 하기 위해 공지되어 있는 코딩 방법 및/또는 본 발명에 적합하게 고안된 코딩에 의해서 컴퓨터 연산으로 구현될 수 있다.

[34] 이하에서 설명하는 구체적인 수학적식은 가능한 여러 대안 중에서 예시적으로 설명되는 것이며, 본 발명의 권리 범위가 본 명세서에 언급된 수학적식에 제한되는 것으로 해석되어서는 아니된다.

[35] 설명의 편의를 위해서, 본 명세서에서는 다음과 같이 표기를 정하기로 한다.

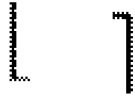
[36] 영문 대문자 볼드체: 행렬

[37] 영문 소문자 볼드체: 열 벡터(column vector)

[38] $a \leftarrow D$: 분포(D)에 따라서 원소(a)를 선택함

[39] 링 $R = \mathbb{Z}(x)/(f(x))$: 정수 위에서 다항식 연산 후에 $f(x)$ 로 나눈 나머지를 계산함

[40] : 내부값을 반올림함



[41] $\begin{bmatrix} | \\ | \end{bmatrix}$: 연결(concatenation); 같은 개수의 행을 가지는 열 벡터 또는 행렬을 나란히

연결하여 새로운 벡터 또는 행렬을 산출함

[42] $HWT_n(h)$: $\{0, \pm 1\}^n$ 에서 0이 아닌 성분의 개수가 h개인 벡터들을 모아놓은 집합을 $B_{n,h}$ 라고 할 때(즉, $B_{n,h}$ 는 $\{0, \pm 1\}^n$ 의 부분 집합), $B_{n,h}$ 에서의 균등(uniform) 분포.

[43] 이하에서는 첨부된 도면을 이용하여 본 발명의 다양한 실시 예들에 대하여 구체적으로 설명한다.

[44] 도 1은 본 발명의 일 실시 예에 따른 암호화 방법을 설명하기 위한 흐름도이다. 도 1에 따르면, 암호화에 사용될 키를 에러를 이용하여 생성한다(S110). 키에는 공개키, 비밀키가 포함된다. 구체적으로는, 비밀키를 랜덤하게 설정하고, 그 비밀키 및 랜덤 값과, 에러를 이용하여 공개키를 생성한다. 에러의 크기는 0보다 크고 1보다 작게 설정된 에러 파라미터에 의해 결정될 수 있다. 구체적으로는,

에러는 이산 가우시안 분포 또는 이산 가우시안 분포로부터 통계적 거리가 가까운 근사 분포로부터 추출할 수 있다. 또한, 비밀키로는 작은 값을 사용할 수 있다. 일 예로, 비밀키는 -1, 0, 1 과 같은 수들 중에서 랜덤하게 조합하여 설정할 수 있다. 공개키는 암호화에 사용되고, 비밀키는 암호화된 메시지를 복호하는 복호화에 사용될 수 있다. 미지의 값인 에러를 이용하여 키를 생성하게 되면, 제3자가 메시지를 획득하여 반복적으로 분석하더라도 키를 파악하기 어렵게 된다. 따라서, 보안성이 더 높아질 수 있다.

- [45] 전송할 메시지가 존재하는 경우, 생성된 공개 키를 이용하여 암호화하여, 암호문을 생성할 수 있다(S120). 구체적으로는, 전송하고자 하는 메시지에 공개키를 적용하고, 라운딩 처리를 수행한다. 라운딩 처리란 암호문을 더 작은 모듈러스 영역으로 보내기 위한 처리를 의미한다. 구체적으로는, 벡터의 라운딩 처리란, 실수 연산으로 p/q 를 곱한 후, 그 결과인 벡터의 각 성분(실수)에 대해 가장 가까운 정수 값을 대응시키는 처리를 의미한다. p 와 q 가 2의 멱승으로 설정된 경우에는, 라운딩 처리에 의해서 벡터 각 성분의 최하위 비트가 일정 개수 제거된다. 이러한 방식으로 라운딩 처리가 이루어지게 되면, 제3자가 암호문을 획득하더라도 복호화하는데 어려워지므로, 보안성이 향상될 수 있다. 연산 장치는 생성된 암호문을 외부 장치로 전송할 수 있다.
- [46] 도 1에서는 키 생성 및 암호문 생성 단계가 하나의 연산 장치에서 수행되는 것을 도시 및 설명하였으나, 반드시 이와 같이 구현되는 것은 아니다. 예를 들어, 하나의 연산 장치가 공개키를 생성한 후, 외부 장치들로 브로드캐스팅할 수 있다. 외부 장치는 공개키를 수신한 후, 그 공개키를 이용하여 메시지를 암호문으로 만들 수 있다. 외부 장치는 암호문을 다시 연산 장치로 전송하고, 연산 장치는 비밀키를 이용하여 암호문을 복호화할 수 있다. 이 경우, 도 1에서 설명한 실시 예는 키 생성 단계 및 공개키 전송 단계를 포함하는 형태로 수행될 수 있다.
- [47] 도 1의 키 생성 단계(S110)에서는 LWE(Learning With Error) 기법이 적용되고, 암호화 단계(S120)에서는 LWR(Learning with Rounding)기법이 적용될 수 있다.
- [48] 도 2는 본 발명의 일 실시 예에 따라 LWE 기법이 적용된 키 생성 방법을 설명하기 위한 흐름도이다.
- [49] 도 2의 각 단계들을 수행하기 위해서는 파라미터를 먼저 설정할 수 있다. 구체적인 파라미터는 다음과 같다.
- [50] l: 평문 메시지 비트의 길이
- [51] m: 공개키의 행(row)의 개수
- [52] n: 공개키의 열(column)의 개수
- [53] p: 암호문의 각 원소의 최대값
- [54] q: 공개키의 각 원소의 최대값
- [55] hr: 열 벡터 r의 0이 아닌 원소의 개수
- [56] t: 평문 각 원소의 최대값

- [57] $hs =$ 비밀키의 0이 아닌 원소의 개수
- [58] $hs < n$
- [59] $hr < m$
- [60] α 는 0보다 크고 1보다 작은 에러 파라미터
- [61] α 가 증가하면 안전도가 증가하고 p 와 q 역시 커지지만 연산 속도가 느려질 수 있다.
- [62] 상술한 파라미터 중에서 p 는 암호문의 각 성분이 가질 수 있는 값의 개수, q 는 공개키의 각 성분이 가질 수 있는 값의 개수, t 는 평문 메시지의 각 성분이 가질 수 있는 값의 개수로 정의할 수도 있다.
- [63] 상술한 파라미터들 간의 조건은 반드시 이에 한정되는 것은 아니며, 실시 예에 따라서 다양하게 변경될 수 있다. 일 예로, p 가 t 로 나누어지고, q 가 p 로 나누어지는 값으로 설정될 수도 있으며, 이러한 조건 없이 임의의 값으로 설정될 수도 있다.
- [64] 도 2에 따르면, 연산 장치는 랜덤하게 결정된 값들로 구성된 제1 랜덤 행렬(A)을 산출한다(S210). 제1 랜덤 행렬(A)은 다음 수학적식과 같이 산출할 수 있다.
- [65] [수학적식 1]
- [66]
$$A \leftarrow \mathbb{Z}_q^{m \times n}$$
- [67] 수학적식 1에서 $\mathbb{Z}_q$ 는 $-q/2$ 부터 $q/2$ 까지 범위 내에 속하는 모든 정수들의 집합을 의미한다. A는 $\mathbb{Z}_q$ 중에서 랜덤하게 선택된 값들로 구성된 $m \times n$ 행렬을 의미한다.
- [68] 연산 장치는, 비밀키(s)를 랜덤하게 생성한다(S220). 연산 장치는 성긴 작은 벡터(sparse small vector)로 비밀키를 생성할 수 있다. 비밀키는 벡터로 이루어질 수도 있고, 벡터들의 조합인 행렬로 이루어질 수도 있다. 일 예로, -1, 0, 1 중 적어도 하나를 행렬 형태로 랜덤하게 조합한 비밀 행렬(S : signed binary secret matrix)를 산출할 수 있다.
- [69] 비밀행렬(S)이 비밀키(sk)로 사용되는 경우, 비밀키는 다음과 같이 표현될 수 있다.
- [70] [수학적식 2]
- [71]
$$sk \leftarrow S \in \{0, \pm 1\}^{n \times l}$$
- [72] 연산 장치는 에러(E)를 추출한다(S230). 에러(E)는 행렬 형태로 이산 가우시안 분포 또는 그와 통계적 거리가 가까운 분포로부터 산출될 수 있다.

구체적으로는, 에러는 다음 수학적식과 같이 표현될 수 있다.

[73] [수학적식 3]

$$[74] \quad \mathbf{E} = (e_1 \parallel \dots \parallel e_l) \leftarrow D_{aq}^{m \times l}$$

[75] 수학적식 3은 표준편차가 $\sqrt{aq}$ 인 이산 가우시안 분포를 독립적으로 m번

샘플링하여 열 벡터를 산출하고 이를 다시 l개 연결하여 에러 행렬(E)을 산출한다는 의미이다.

[76] 에러 행렬(E)은 반드시 수학적식 3과 같은 방식으로 산출하여야 하는 것은 아니며, LWE 기반 공개키 암호화 방법에 적용 가능하도록 공지된 다른 방식으로 산출할 수도 있음은 물론이다.

[77] 연산 장치는, 생성한 제1 랜덤 행렬 및 비밀키와, 에러를 이용하여 제2 랜덤 행렬(B)을 산출한다(S240). 제2 랜덤 행렬(B)은 다음과 같은 수학적식에 따라 산출될 수 있다.

[78] [수학적식 4]

$$[79] \quad \mathbf{B} = -\mathbf{AS} + \mathbf{E} \in \mathbb{Z}_q^{m \times l}$$

[80] 연산 장치는, 제1 랜덤 행렬 및 제2 랜덤 행렬을 이용하여 공개키를 산출한다(S250). 구체적으로는, 제1 랜덤 행렬 및 제2 랜덤 행렬을 연결하여 다음과 같이 공개키를 산출할 수 있다.

[81] [수학적식 5]

$$[82] \quad \mathbf{pk} \leftarrow (\mathbf{A} \parallel \mathbf{B}) \in \mathbb{Z}_q^{m \times (n+l)}$$

[83] 이상과 같이 연산 장치는 에러를 이용하여 공개키를 생성할 수 있다. 도 2의 각 단계의 순서는 일 예에 불과하며, 실시 예에 따라 각 단계들의 순서는 서로 바뀔 수 있다.

[84] 비밀키 및 공개키가 생성되면, 연산 장치는 이를 저장할 수 있다. 실시 환경에 따라서는, 연산 장치는 외부 장치들과 비밀키 및 공개키 중 적어도 하나를 공유할 수도 있다.

[85] 한편, 상술한 바와 같이 암호화는 라운딩 함수를 이용하여 이루어질 수 있다. 도 3은 본 발명의 일 실시 예에 따른 LWR 기법의 암호문 생성 방법을 설명하기 위한 흐름도이다. 도 3에 따르면, 전송할 메시지가 있는 연산 장치는 랜덤 벡터를 설정한다(S310). 랜덤 벡터는 각 성분이 -1, 0, 1의 값을 갖도록 설정될 수 있다. 구체적으로는 다음 수학적식과 같이 설정될 수 있다.

[86] [수학적식 6]

$$[87] \quad \mathbf{r} \leftarrow \mathcal{HWT}_m(h_r)$$

[88] 연산 장치는 공개키 및 랜덤 벡터를 이용하여 메시지에 대한 1차 연산을

수행한다(S320). 1차 연산이란 라운딩 처리가 이루어지기 전의 암호문 상태로 만드는 과정을 의미한다.

- [89] 구체적으로는, 공개키에 포함된 제1 랜덤 행렬과 랜덤 벡터를 연산한 제1 값을 산출한다. 또한, 공개키에 포함된 제2 랜덤 행렬 및 랜덤 벡터를 연산한 결과값에, 메시지를 인코딩한 값을 추가한 제2 값을 산출한다. 1값 및 2값은 상술한 1차 연산의 결과값을 의미하며, 그 산출 순서는 특정한 순서로 한정되지 않는다.

- [90] 구체적으로는 제1값(c1) 및 제2값(c2)은 다음과 같은 방식으로 산출될 수 있다.

- [91] [수학식 7]

- [92] $c_1' = rA$

- [93] $c_2' = rB + M \lfloor \frac{q}{2} \rfloor$

- [94] 여기서, r은 랜덤 벡터, A는 제1 랜덤 행렬, B는 제2 랜덤 행렬, M은 메시지를 의미한다. 수학식 7에서는 메시지 M에 대하여 q/2를 라운딩한 값을 곱하였지만, 다르게는 q/t를 라운딩한 값을 곱할 수도 있다. 이와 같이 메시지를 라운딩 처리 등과 같은 연산을 수행하여 값을 변경시키는 것을 본 명세서에서는 인코딩이라 한다.

- [95] 1차 연산 결과값이 산출되면, 라운딩 처리를 수행하여 메시지에 대응되는 암호문을 산출한다(S330).

- [96] 암호문은 반드시 수학식 8과 같은 방법으로 생성되어야 하는 것은 아니다. 다른 예에 따르면, 암호문은 다음과 같은 방법으로 생성될 수 있다.

- [97] [수학식 8]

- [98] $c \leftarrow (c_1, c_2) \in \mathbb{Z}_p^{n+1}$

- [99] $c_1 \leftarrow \lfloor (p/q) \cdot c_1' \rfloor \in \mathbb{Z}_p^n; c_1' \leftarrow A^T r$

- [100] $c_2 \leftarrow (p/t) \cdot m + \lfloor (p/q) \cdot c_2' \rfloor \in \mathbb{Z}_p^1; c_2' \leftarrow B^T r$

- [101] 수학식 7은 메시지의 인코딩 값을 라운딩 처리 전에 더하는 방식이고, 수학식 8은 라운딩 후에 더하는 방식이다. 상술한 두 방법은 q가 p의 배수이고, p가 t의 배수인 경우 동일한 값을 갖는 암호문을 산출할 수 있다.

- [102] 이상과 같이 암호문을 생성하면, 기 설정된 개수의 하위 비트가 제거되고, 가우시안 샘플링을 이용하지 않아도 된다. 또한, 비밀키와 랜덤 벡터를 0, -1, 1과 같은 성긴 작은 벡터(sparse small vector)로 구성하기 때문에 안전성이 보장되는

동시에 암호화 속도가 매우 빨라지게 된다. 암호문의 크기 역시 현저하게 줄어들 수 있다.

- [103] 한편, 도 1의 암호화 방법에서는 도시하지 않았으나, 상술한 방식으로 생성한 암호문, 즉, 공개키를 이용하여 암호화된 메시지가 수신되면, 그 메시지를 복호화하는 단계가 더 추가될 수도 있다. 복호화는 상술한 비밀키를 이용하여 수행할 수 있다. 복호화는 다음과 같은 방식으로 이루어질 수 있다.

[104] [수학식 9]

$$[105] \quad \mathbf{m} \leftarrow \lfloor t / p \cdot (\mathbf{S}^T \mathbf{a} + \mathbf{b}) \rfloor \pmod{t}$$

- [106] 한편, 이상에서는 키 생성은 LWE 기법, 암호화는 LWR 기법에 기초하여 수행되었지만, 본 발명의 다른 실시 예에 따르면 키 생성 및 암호화는 링(Ring)을 기반으로 이루어질 수도 있다. 설명의 편의를 위하여 이러한 기법은 R-LWE 기법, R-LWR 기법으로 명명하여 설명한다.

- [107] 도 4는 링을 기반으로 키를 생성하는 방법을 설명하기 위한 흐름도이다. 구체적으로는, 키 생성 과정에서는 R-LWE 기법이 사용될 수 있다. 도 4에 따르면, 연산 장치는 파라미터 및 링을 설정한다(S410). 파라미터에 대해서는 상술한 부분에서 구체적으로 기재하였으므로 중복 설명은 생략한다. 링은 다음과 같은 수학식으로 표현될 수 있다.

[108] [수학식 10]

$$[109] \quad R = \frac{\mathbb{Z}_q[x]}{(f(x))}$$

- [110] 링(Ring)이란 기 설정된 계수를 가지는 다항식의 집합을 의미한다. 일 예로, 링은 계수가 $\mathbb{Z}_q$ 인 n 차 다항식의 집합을 의미한다. 수학식 10에서 $f(x)$ 는 n 차 다항식을 의미한다. 구체적으로는, n 이 $\Phi(N)$ 일 때, N 차 사이클로토믹 다항식(cyclotomic polynomial)을 의미한다. $(f(x))$ 란 $f(x)$ 로 생성되는 $\mathbb{Z}_q[x]$ 의 이데알(ideal)을 나타낸다. $\Phi(N)$ 이란 N 과 서로 소이고 N 보다 작은 자연수의 개수를 의미한다. $\Phi_N(x)$ 를 N 차 사이클로토믹 다항식으로 정의하면, 링은 다음과 같은 수학식으로도 표현될 수 있다.

[111] [수학식 11]

$$[112] \quad R = \frac{\mathbb{Z}_q[x]}{\Phi_N(x)}$$

- [113] 다음으로, 연산 장치는, 링으로부터 비밀키(sk)를 산출한다(S420).

[114] [수학식 12]

$$[115] \quad \mathbf{sk} \leftarrow \mathbf{S}(\mathbf{x}) \leftarrow \mathbf{R}$$

- [116] $\mathbf{S}(\mathbf{x})$ 는 각 계수들이 0, -1, 1로 이루어진 다항식이다. 예를 들어, $\mathbf{S}(\mathbf{x})$ 는 $\text{HWT}_n(h_s)$ 의 분포를 따르도록 산출될 수 있다. 산출된 다항식들을 비밀키 sk로 설정한다.

- [117] 다음으로, 링으로부터 제1 랜덤 다항식($\mathbf{a}(x)$)을 산출한다(S430). 제1 랜덤

다항식은 다음과 같이 표현될 수 있다.

[118] [수학식 13]

[119] $\mathbf{a}(x) \leftarrow \mathbf{R}$

[120] 또한, 연산 장치는 이산 가우시안 분포 또는 그와 통계적 거리가 가까운 분포로부터 에러를 추출한다(S440). 에러는 다음과 같은 수학식으로 표현될 수 있다.

[121] [수학식 14]

[122]

$$e(x) \leftarrow D_{\text{aq}}^n$$

[123] 에러까지 산출되면, 연산 장치는 제1 랜덤 다항식 및 비밀키에 에러를 모듈러 연산하여 제2 랜덤 다항식을 산출할 수 있다(S450). 제2 랜덤 다항식은 다음과 같이 표현될 수 있다.

[124] [수학식 15]

[125] $b(x) = a(x)s(x) + e(x)$

[126] 최종적으로 공개키(pk)는 제1 랜덤 다항식 및 제2 랜덤 다항식을 포함하는 형태로 다음과 같이 설정된다(S460).

[127] [수학식 16]

[128] $\mathbf{pk} = (\mathbf{a}(x), \mathbf{b}(x))$

[129] 도 5는 R-LWR 기법을 기반으로 암호문을 생성하는 방법을 설명하기 위한 흐름도이다. 도 5에 따르면, 다음 수학식과 같이 랜덤하게 다항식을 추출한다(S510).

[130] [수학식 17]

[131] $r(x) \in \{-1, 0, 1\}^n$

[132] 다음으로, 공개키 및 다항식을 이용하여 메시지에 대한 1차 연산을 수행한다(S520). 1차 연산이란 상술한 바와 같이 제1 랜덤 다항식 및 제2 랜덤 다항식과, 상술한 다항식, 메시지 등을 이용하여 제1 값 및 제2 값을 산출하는 과정을 의미한다. 일 예로, 제1 랜덤 다항식 및 다항식을 연산한 제1 값, 공개키에 포함된 제2 랜덤 다항식 및 다항식을 연산한 결과값에 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출할 수 있다. 제1 값 및 제2 값(c_1' , c_2')은 다음 수학식과 같이 표현될 수 있다.

[133] [수학식 18]

[134] $c_1'(x) = r(x) a(x)$

[135] $c_2'(x) = \frac{q}{t} m(x) + r(x) b(x)$

[136] 1차 연산이 수행되면, 그 결과값을 라운딩 처리하여 암호문 다항식을 산출할 수 있다(S530). 일 예로, 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 메시지에 대응되는 암호문 다항식을 산출할 수 있다. 암호문 다항식은 다음과 같이 표현될 수 있다.

[137] [수학식 19]

$$[138] \quad c(x) = (c_1(x), c_2(x))$$

$$[139] \quad c_1(x) = \lfloor p/q \cdot r(x)a(x) \rfloor$$

$$[140] \quad c_2(x) = \lfloor p/q(q/t \cdot m(x) + r(x)b(x)) \rfloor$$

[141] 상술한 암호문 다항식은 링에서 연산될 수 있다. 여기서 $c_2(x)$ 는 다음 수학식과 같이 연산될 수도 있다.

[142] [수학식 20]

$$[143] \quad c_2(x) = \left\lfloor \frac{p}{t} m(x) + \left\lfloor \frac{p}{q} \cdot r(x) \cdot b(x) \right\rfloor \right\rfloor \text{ in } R_p$$

[144] 도 4 및 도 5에서 설명한 바와 같이 링을 기반으로도 키 생성 및 암호문이 생성될 수 있다. 연산 장치는 생성된 공개 키를 외부 장치들로 제공하여 줄 수 있다. 공개키를 수신한 외부 장치는 연산 장치로 전송할 메시지가 있으면 공개 키를 이용하여 암호화한 후 전송할 수 있다. 연산 장치가 암호화된 메시지를 수신하면, 비밀키를 이용하여 복호화를 수행한다. 도 4 및 도 5와 같은 방식으로 생성된 암호문을 복호화하는 방법은 다음 수학식과 같이 표현될 수 있다.

[145] [수학식 21]

$$[146] \quad c_2(x) - S(x)c_1(x) = \frac{p}{t}m(x) + \Delta e \xrightarrow{\lfloor t/p \cdot x \rfloor} m(x)$$

[147] 여기서, 노이즈 Δe 는 $\lfloor t/p \cdot x \rfloor$ 에 의해서 최하위 비트가 삭제됨에

따라서 복호화 과정에서 사라지게 된다. 도 4 및 도 5와 같은 방식으로 암호화를 하더라도, 암호화 과정에서 최하위 비트가 삭제되므로 상술한 다른 실시 예와 같은 효과를 얻을 수 있다.

[148] 이상과 같은 키 생성, 암호화, 복호화 동작은 연산 장치에 의해 수행될 수 있다. 연산 장치는 컴퓨터가 될 수 있지만 반드시 이에 한정되는 것은 아니다. 예를 들어, 통신 기능을 구비한 서버 장치, 휴대폰, 랩탑, 태블릿 등과 같은 다양한 단말 장치들도 상술한 동작을 수행할 수 있다. 본 명세서에서는 상술한 다양한 실시 예들에 따른 동작을 수행할 수 있는 장치를 통칭하여 연산 장치라 한다.

[149] 도 6은 본 발명의 일 실시 예에 따른 연산 장치의 구성을 나타내는 블록도이다. 도 6에 따르면, 연산 장치(100)는 메모리(110) 및 프로세서(120)를 포함한다. 메모리(110)는 연산 장치(100)를 구동하기 위한 O/S나 각종 소프트웨어, 데이터

등을 저장하기 위한 구성요소이다. 메모리(110)는 RAM이나 ROM, 플래시 메모리, HDD, 외장 메모리, 메모리 카드 등과 같은 다양한 형태로 구현될 수 있으며, 어느 하나로 한정되는 것은 아니다.

- [150] 프로세서(120)는 메모리(110)에 저장된 프로그램을 실행시켜 각종 연산 작업을 수행하기 위한 구성요소이다.
- [151] 일 예로, 프로세서(120)는 비밀키를 설정하고, 에러 및 비밀키를 이용하여 공개키를 생성할 수 있다. 프로세서(120)는 생성된 비밀키 및 공개키를 메모리에 저장할 수 있다.
- [152] 저장된 키는 다양한 동작에 사용될 수 있다. 일 예로, 외부로 전송해야 할 메시지 또는 암호화가 요구되는 메시지가 입력되면, 프로세서(120)는 메모리(110)에 저장된 공개키를 메시지에 적용한 후 라운딩 처리를 수행하여 메시지를 암호화한다. 여기서 비밀키는 작은 값으로 설정될 수 있다. 구체적으로는, 상술한 실시 예들에서 설명한 바와 같이 -1, 0, 1 중에서 랜덤하게 조합될 수 있다. 프로세서의 암호화 동작의 다양한 실시 예들에 대해서는 상술한 도 1 내지 도 5에 대한 부분에서 구체적으로 기재하였으므로 중복 설명은 생략한다.
- [153] 한편, 도 6에서는 도시를 생략하였으나, 연산 장치(100)는 통신부(미도시)를 더 포함할 수도 있다. 통신부는 외부 장치들과 통신을 수행하기 위한 구성요소이다.
- [154] 프로세서(120)는 메모리(110)에 저장된 공개키를 통신부를 통해 브로드캐스팅할 수 있다. 또는, 공개 키를 이용할 외부 장치가 특정된다면, 그 외부 장치의 IP(Internet Protocol) 주소로 공개 키를 전송하여 줄 수도 있다.
- [155] 프로세서(120)는, 공개 키를 수신한 외부 장치들에서 암호화가 이루어지면, 통신부를 통해서 암호문을 수신할 수도 있다. 프로세서(120)는 메모리(110)에 저장된 비밀키를 이용하여, 암호문을 복호화할 수 있다.
- [156] 이 밖에, 연산 장치(100)는 그 종류에 따라서 입력 수단, 디스플레이, 버스 등과 같은 다양한 구성요소들을 더 포함할 수도 있지만, 본 발명의 다양한 실시 예들과는 직접적인 관련성이 없으므로 그 도시 및 설명은 생략한다.
- [157] 도 7은 본 발명의 일 실시 예에 따른 암호화/복호화를 수행하는 시스템을 설명하기 위한 도면이다. 도 7에 따르면, 시스템은 서버 장치(100)와 다양한 외부 장치(710-1 ~ 710-n)를 포함한다. 각 장치들은 네트워크(700)를 통해 서로 연결된다.
- [158] 서버 장치(100)는 상술한 다양한 실시 예들에서 설명한 방식으로 공개 키를 생성한 후, 네트워크를 통해 브로드캐스팅한다. 네트워크(700)에 연결된 다양한 종류의 장치들(710-1 ~ 710-n)들은 공개키를 수신할 수 있다. 이 중 장치 2(710-2)가 서버 장치(100)로 액세스하고자 하고자 하는 경우, 장치 2(710-2)는 전송할 메시지를 공개키를 이용하여 암호화한 후, 그 암호문을 전송하여 줄 수 있다. 서버 장치(100)는 암호문을 수신한 후, 비밀 키를 이용하여 복호화할 수 있다.

- [159] 이 경우, 장치 2(710-2)도 상술한 바와 같은 키 생성 동작을 수행할 수 있다면, 장치 2(710-2)도 자체적으로 공개 키 및 비밀 키를 생성할 수 있다. 장치 2(710-2)는 공개 키를 서버 장치(100)로 전송하여 줄 수 있다. 이 경우, 장치 2(710-2)는 공개 키 자체도 암호화하여 전송하여 줄 수도 있다.
- [160] 서버 장치(100)는 장치 2(710-2)로부터 공개 키가 제공되면, 장치 2(710-2)로 전송할 메시지가 있을 때, 수신된 공개 키를 이용하여 암호화한 후 전송할 수 있다. 이에 따라, 서버 장치(100) 및 장치 2(710-2) 사이의 통신 보안성이 더욱 강화될 수도 있다.
- [161] 또는, 서버 장치(100)는 비밀키 및 공개키를 생성한 후, 그 키들을 모두 특정 장치에게 제공하여 줄 수도 있다. 비밀키 및 공개키를 수신한 장치는 외부 장치들과의 통신에서 이를 이용할 수 있다.
- [162] 한편, 이상과 같은 다양한 실시 예들에 따른 키 생성, 암호화, 복호화 동작 중 적어도 하나를 수행하기 위한 프로그램 코드가 기록 매체에 저장된 상태로 배포될 수도 있다. 이 경우, 기록 매체가 탑재된 장치는 상술한 동작들을 수행할 수 있다. 일 예로, 랜덤하게 결정된 값들로 구성된 제1 랜덤 행렬(A)을 산출하는 단계, 각 성분이 -1, 0, 1 중 하나의 값을 가지는 열 벡터를 행렬 형태로 랜덤하게 조합하여 비밀키(s)를 산출하는 단계, 이산 가우시안 분포 또는 그와 통계적 거리가 가까운 분포로부터 에러(E)를 추출하는 단계, 제1 랜덤 행렬 및 비밀키와, 에러를 모듈러 연산하여 제2 랜덤 행렬(B)을 산출하는 단계, 제1 랜덤 행렬 및 제2 랜덤 행렬을 포함하는 공개키를 설정하는 단계, -1, 0, 1 중에서 랜덤 벡터를 설정하는 단계, 제1 랜덤 행렬 및 랜덤 벡터를 연산한 제1 값, 제2 랜덤 행렬 및 랜덤 벡터를 연산한 결과값에 메시지를 인코딩한 값을 추가한 제2 값을 각각 산출하는 단계, 제1 및 제2 값 각각에 대해서 기 설정된 개수의 하위 비트를 제거하기 위한 라운딩 처리를 수행하여, 상기 메시지에 대응되는 암호문을 산출하는 단계를 순차적으로 수행하기 위한 프로그램 코드가 기록 매체에 저장될 수 있다.
- [163] 이러한 기록 매체는, ROM, RAM, 메모리 칩, 메모리 카드, 외장형 하드, 하드, CD, DVD, 자기 디스크 또는 자기 테이프 등과 같은 다양한 유형의 컴퓨터 판독 가능 매체가 될 수 있다.
- [164] 이상 설명한 바와 같이, 본 발명의 다양한 실시 예에 따라 키 생성 및 암호화를 수행하게 되면, 양자 컴퓨터 시대에서도 안전하며, 암호화/복호화 속도를 크게 향상시킬 수 있고 암호문 크기도 줄일 수 있다. 특히, 암호화 과정에서 생성하는 랜덤 값과 비밀키를 작은 벡터를 이용하여 설정하고, 암호화 과정에서 최하위 비트들을 제거하는 처리 등을 수행하여 효율성을 극대화시킬 수 있다.
- [165] 이상 첨부 도면을 참고하여 본 발명에 대해서 설명하였지만 본 발명의 권리범위는 후술하는 특허청구범위에 의해 결정되며 전술한 실시 예 및/또는 도면에 제한되는 것으로 해석되어서는 아니된다. 그리고 특허청구범위에 기재된 발명의, 당업자에게 자명한 개량, 변경 및 수정도 본 발명의 권리범위에

포함된다는 점이 명백하게 이해되어야 한다.