

명세서

발명의 명칭: 암호문에 대한 비다항식 연산을 수행하는 장치 및 방법

기술분야

- [1] 본 개시는 근사 암호화된 암호문에 대해서 비교 연산과 같은 비다항식 연산을 수행하기 위한 장치 및 방법에 대한 것이다.

배경기술

- [2] 통신 기술이 발달하고, 전자 장치의 보급이 활발해짐에 따라, 전자 장치 간의 통신 보안을 유지하기 위한 노력이 지속적으로 이루어지고 있다. 이에 따라, 대부분의 통신 환경에서는 암호화/복호화 기술이 사용되고 있다.
- [3] 암호화 기술에 의해 암호화된 메시지가 상대방에게 전달되면, 상대방은 메시지를 이용하기 위해서는 복호화를 수행하여야 한다. 이 경우, 상대방은 암호화된 데이터를 복호화하는 과정에서 자원 및 시간 낭비가 발생하게 된다. 또한, 상대방이 연산을 위해 일시적으로 메시지를 복호화한 상태에서 제3자의 해킹이 이루어지는 경우, 메시지가 제3자에게 손쉽게 유출될 수 있다는 문제점도 있었다.
- [4] 이러한 문제를 해결하기 위하여 동형 암호화 방법이 연구되고 있다. 동형 암호화에 따르면, 암호화된 정보를 복호화하지 않고 암호문 자체에서 연산을 하더라도, 평문에 대해 연산한 후 암호화한 값과 동일한 결과를 얻을 수 있다. 따라서, 암호문을 복호화하지 않은 상태에서 각종 연산을 수행할 수 있다.
- [5] 그러나 종래의 동형 암호문은 덧셈 및 곱셈 등의 일부 연산만 지원하였다는 점에서, 동형 암호문에 대해서 비교 연산과 같은 비다항식 연산을 적용할 수 있는 방법이 요구되었다.

발명의 상세한 설명

기술적 과제

- [6] 따라서 본 개시는 상술한 바와 같은 문제점을 해결하기 위한 고안된 것으로, 근사 암호화된 암호문에 대해서 비교 연산과 같은 비다항식 연산을 수행하기 위한 장치 및 방법을 제공하는 데 있다.

과제 해결 수단

- [7] 본 개시는 이상과 같은 목적을 달성하기 위한 것으로, 본 암호문 연산 방법은 복수의 동형 암호문에 대한 비교 연산 명령을 입력받는 단계, 상기 비교 연산 명령에 대응되는 합성 함수에 상기 복수의 동형 암호문을 반영하여 연산하는 단계, 및 상기 연산된 동형 암호문을 출력하는 단계를 포함한다.
- [8] 이 경우, 상기 합성 함수는 $2^{-\alpha}$ 정확도에 대응하여 $\theta(\alpha)$ 또는 $\theta(\alpha \log \alpha)$ 복잡도(complexity)를 가질 수 있다.
- [9] 한편, 상기 비교 연산은 최대값 산출 또는 최소값 산출이고, 상기 연산하는

단계는, 제공근 연산에 대응되는 합성 함수를 이용하여 상기 최대값 산출 또는 상기 최소값 산출을 연산할 수 있다.

- [10] 이 경우, 상기 제공근 연산에 대응되는 합성 함수는, 입력값을 초기 값으로 갖는 제1 변수와 상기 입력 값에 1을 뺀 값을 초기 값으로 갖는 제2 변수를 포함하는 제1 함수 및 상기 제2 변수를 포함하는 제2 함수를 기설정된 횟수만큼 반복 연산할 수 있다.
- [11] 또한, 상기 기설정된 횟수는 상기 제공근 연산의 오차율에 대응되는 횟수일 수 있다.
- [12] 한편, 상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고, 상기 연산하는 단계는, 인버스 연산에 대응되는 함수 연산을 포함하는 합성 함수를 기설정된 횟수만큼 반복 연산할 수 있다.
- [13] 한편, 상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고, 상기 연산하는 단계는, 부호 함수에 대응되는 합성 함수를 이용하여 기설정된 횟수만큼 반복 연산할 수 있다.
- [14] 이 경우, 상기 부호 함수에 대응되는 합성 함수는, 0보다 큰 입력 값에 대해서는 출력값이 1에 가까워지고, 0보다 작은 입력 값에 대해서는 출력 값이 -1에 가까워지도록 하는 합성 함수일 수 있다.
- [15] 한편, 상기 연산하는 단계는, 상기 부호 함수에 대응되는 근사 함수이며 서로 다른 두개 이상의 합성 함수를 이용하여 상기 비교 연산을 수행할 수 있다.
- [16] 한편, 본 개시의 일 실시 예에 다른 연산 장치는 적어도 하나의 인스트럭션(instruction)을 저장하는 메모리, 및 상기 적어도 하나의 인스트럭션을 실행하는 프로세서를 포함하고, 상기 프로세서는, 상기 적어도 하나의 인스트럭션을 실행함으로써, 복수의 동형 암호문에 대한 비교 연산 명령을 입력받으면, 상기 비교 연산 명령에 대응되는 합성 함수에 상기 복수의 동형 암호문을 반영하여 연산하고, 상기 연산된 동형 암호문을 출력한다.
- [17] 이 경우, 상기 비교 연산은 최대값 산출 또는 최소값 산출이고, 상기 프로세서는, 제공근 연산에 대응되는 합성 함수를 이용하여 상기 최대값 산출 또는 상기 최소값 산출을 연산할 수 있다.
- [18] 한편, 상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고, 상기 프로세서는, 인버스 연산에 대응되는 함수 연산을 포함하는 합성 함수를 기설정된 횟수만큼 반복 연산할 수 있다.
- [19] 한편, 상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고, 상기 프로세서는, 부호 함수에 대응되는 합성 함수를 이용하여 기설정된 횟수만큼 반복 연산할 수 있다.
- [20] 한편, 상기 부호 함수에 대응되는 합성 함수는, 0보다 큰 입력 값에 대해서는 출력값이 1에 가까워지고, 0보다 작은 입력 값에 대해서는 출력 값이 -1에 가까워지도록 하는 합성 함수일 수 있다.
- [21] 한편, 상기 프로세서는, 상기 부호 함수에 대응되는 근사 함수이며 서로 다른

두개 이상의 합성 함수를 이용하여 상기 비교 연산을 수행할 수 있다.

발명의 효과

- [22] 이상과 같은 본 개시의 다양한 실시 예들에 따르면, 동형 암호문 처리 방법은 합성 함수를 이용하여 근사 암호화된 암호문에 대한 비교 연산을 수행할 수 있게 된다. 더욱이 본 개시에서 제안한 합성 함수는 낮은 복잡도를 갖는 바, 정확도가 증가하더라도 보다 빠른 연산이 가능하다.

도면의 간단한 설명

- [23] 도 1은 본 개시의 일 실시 예에 따른 네트워크 시스템의 구조를 설명하기 위한 도면,
 [24] 도 2는 본 개시의 일 실시 예에 따른 연산 장치의 구성을 나타낸 블록도,
 [25] 도 3은 본 개시의 연산 장치의 연산 동작을 설명하기 위한 도면,
 [26] 도 4는 본 개시의 일 실시 예에 따른 암호문 연산 방법을 설명하기 위한 흐름도,
 [27] 도 5는 본 개시의 일 실시 예에 따라 인버스 연산을 합성 함수를 이용하여 계산하는 방법을 설명하기 위한 도면,
 [28] 도 6은 본 개시의 일 실시 예에 따라 제곱근 연산을 합성 함수를 이용하여 계산하는 방법을 설명하기 위한 도면,
 [29] 도 7은 본 개시의 일 실시 예에 따른 최대 값 계산 및 최소 값 계산 방법을 설명하기 위한 도면,
 [30] 도 8은 본 개시의 제1 실시 예에 따른 크기 비교 방법을 설명하기 위한 도면,
 [31] 도 9는 본 개시의 제2 실시 예에 따른 크기 비교 방법을 설명하기 위한 도면,
 [32] 도 10은 부호 함수의 기울기에 대응되는 합성 함수를 찾는 방법을 설명하기 위한 도면,
 [33] 도 11은 본 개시의 제3 실시 예에 따른 크기 비교 방법을 설명하기 위한 도면,
 [34] 도 12는 부호 함수에 대응되는 합성 함수의 반복 횟수별 형태를 도시한 도면, 그리고,
 [35] 도 13은 부호 함수와 관련된 다양한 합성 함수의 반복 횟수별 형태를 도시한 도면이다.

발명의 실시를 위한 최선의 형태

- [36] -

발명의 실시를 위한 형태

- [37] 이하에서는 첨부 도면을 참조하여 본 개시에 대해서 자세하게 설명한다. 본 개시에서 수행되는 정보(데이터) 전송 과정은 필요에 따라서 암호화/복호화가 적용될 수 있으며, 본 개시 및 특허청구범위에서 정보(데이터) 전송 과정을 설명하는 표현은 별도로 언급되지 않더라도 모두 암호화/복호화하는 경우도 포함하는 것으로 해석되어야 한다. 본 개시에서 "A로부터 B로 전송(전달)" 또는 "A가 B로부터 수신"과 같은 형태의 표현은 중간에 다른 매개체가 포함되어 전송(전달) 또는 수신되는 것도 포함하며, 반드시 A로부터 B까지 직접

전송(전달) 또는 수신되는 것만을 표현하는 것은 아니다.

- [38] 본 개시의 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 개시의 본질에는 영향이 없으며 권리범위 역시 단계의 순서에 관계없이 정의되어야 한다. 그리고 본 명세서에서 "A 또는 B"라고 기재한 것은 A와 B 중 어느 하나를 선택적으로 가리키는 것뿐만 아니라 A와 B 모두를 포함하는 것도 의미하는 것으로 정의된다. 또한, 본 개시에서 "포함"이라는 용어는 포함하는 것으로 나열된 요소 이외에 추가로 다른 구성요소를 더 포함하는 것도 포괄하는 의미를 가진다.
- [39] 본 개시에서는 본 개시의 설명에 필요한 필수적인 구성요소만을 설명하며, 본 개시의 본질과 관계가 없는 구성요소는 언급하지 아니한다. 그리고 언급되는 구성요소만을 포함하는 배타적인 의미로 해석되어서는 안 되며 다른 구성요소도 포함할 수 있는 비배타적인 의미로 해석되어야 한다.
- [40] 그리고 본 개시에서 "값"이라 함은 스칼라값뿐만 아니라 벡터도 포함하는 개념으로 정의된다.
- [41] 후술하는 본 개시의 각 단계의 수학적 연산 및 산출은 해당 연산 또는 산출을 하기 위해 공지되어 있는 코딩 방법 및/또는 본 개시에 적합하게 고안된 코딩에 의해서 컴퓨터 연산으로 구현될 수 있다.
- [42] 이하에서 설명하는 구체적인 수학적식은 가능한 여러 대안 중에서 예시적으로 설명되는 것이며, 본 개시의 권리 범위가 본 개시에 언급된 수학적식에 제한되는 것으로 해석되어서는 아니된다.
- [43] 설명의 편의를 위해서, 본 개시에서는 다음과 같이 표기를 정하기로 한다.
- [44] $a \leftarrow D$: 분포(D)에 따라서 원소(a)를 선택함
- [45] $s_1, s_2 \in R$: S1, S2 각각은 R 집합에 속하는 원소이다.
- [46] $\text{mod}(q)$: q 원소로 모듈(modular) 연산
- [47] $\lceil \cdot \rceil$: 내부 값을 반올림함
- [48] 이하에서는 첨부된 도면을 이용하여 본 개시의 다양한 실시 예들에 대하여 구체적으로 설명한다.
- [49] 도 1은 본 개시의 일 실시 예에 따른 네트워크 시스템의 구성을 나타내는 도면이다.
- [50] 도 1을 참조하면, 네트워크 시스템은 복수의 전자 장치(100-1 ~ 100-n), 제1 서버 장치(200), 제2 서버 장치(300)를 포함할 수 있으며, 각 구성들은 네트워크(10)를 통해 서로 연결될 수 있다.
- [51] 네트워크(10)는 다양한 형태의 유무선 통신 네트워크, 방송 통신 네트워크, 광통신 네트워크, 클라우드 네트워크 등으로 구현될 수 있으며, 각 장치들은 별도의 매개체 없이 와이파이, 블루투스, NFC(Near Field Communication) 등과

같은 방식으로 연결될 수도 있다.

- [52] 도 1에서는 전자 장치가 복수 개(100-1 ~ 100-n)인 것으로 도시하였으나, 반드시 복수 개의 전자 장치가 사용되어야 하는 것은 아니며 하나의 장치가 사용될 수도 있다. 일 예로, 전자 장치(100-1 ~ 100-n)는 스마트폰, 태블릿, 게임 플레이어, PC, 랩톱 PC, 홈서버, 키오스크 등과 같은 다양한 형태의 장치로 구현될 수 있으며, 이밖에 IoT 기능이 적용된 가전 제품 형태로도 구현될 수 있다.
- [53] 사용자는 자신이 사용하는 전자 장치(100-1 ~ 100-n)를 통해서 다양한 정보를 입력할 수 있다. 입력된 정보는 전자 장치(100-1 ~ 100-n) 자체에 저장될 수도 있지만, 저장 용량 및 보안 등을 이유로 외부 장치로 전송되어 저장될 수도 있다. 도 1에서 제1 서버 장치(200)는 이러한 정보들을 저장하는 역할을 수행하고, 제2 서버 장치(300)는 제1 서버 장치(200)에 저장된 정보의 일부 또는 전부를 이용하는 역할을 수행할 수 있다.
- [54] 각 전자 장치(100-1 ~ 100-n)는 입력된 정보를 동형 암호화하여, 동형 암호문을 제1 서버 장치(200)로 전송할 수 있다.
- [55] 각 전자 장치(100-1 ~ 100-n)는 동형 암호화를 수행하는 과정에서 산출되는 암호화 노이즈, 즉, 에러를 암호문에 포함시킬 수 있다. 구체적으로는, 각 전자 장치(100-1 ~ 100-n)에서 생성하는 동형 암호문은, 추후에 비밀 키를 이용하여 복호화하였을 때 메시지 및 에러 값을 포함하는 결과 값이 복원되는 형태로 생성될 수 있다.
- [56] 일 예로, 전자 장치(100-1 ~ 100-n)에서 생성하는 동형 암호문은 비밀 키를 이용하여 복호화 하였을 때 다음과 같은 성질을 만족하는 형태로 생성될 수 있다.
- [57] [수학식 1]
- [58] $\text{Dec}(ct, sk) = \langle ct, sk \rangle = M + e \pmod{q}$
- [59] 여기서 $\langle \cdot, \cdot \rangle$ 는 내적 연산(usual inner product), ct 는 암호문, sk 는 비밀 키, M 은 평문 메시지, e 는 암호화 에러 값, $\text{mod } q$ 는 암호문의 모듈러스(Modulus)를 의미한다. q 는 스케일링 팩터(scaling factor)(Δ)가 메시지에 곱해진 결과 값 M 보다 크게 선택되어야 한다. 에러 값 e 의 절대값이 M 에 비해서 충분히 작다면, 암호문의 복호화 값 $M+e$ 는 유효숫자연산에서 원래의 메시지를 동일한 정밀도로 대체할 수 있는 값이다. 복호화된 데이터 중에서 에러는 최하위 비트(LSB) 측에 배치되고, M 은 차하위 비트 측에 배치될 수 있다.
- [60] 메시지의 크기가 너무 작거나 너무 큰 경우, 스케일링 팩터를 이용하여 그 크기를 조절할 수도 있다. 스케일링 팩터를 사용하게 되면, 정수 형태의 메시지뿐만 아니라 실수 형태의 메시지까지도 암호화할 수 있게 되므로, 활용성이 크게 증대할 수 있다. 또한, 스케일링 팩터를 이용하여 메시지의 크기를 조절함으로써, 연산이 이루어지고 난 이후의 암호문에서 메시지들이 존재하는 영역, 즉, 유효 영역의 크기도 조절될 수 있다.
- [61] 실시 예에 따라, 암호문 모듈러스 q 는 다양한 형태로 설정되어 사용될 수 있다.

일 예로, 암호문의 모듈러스는 스케일링 팩터 Δ 의 지수승 $q=\Delta^L$ 형태로 설정될 수 있다. Δ 가 2라면, $q=2^{10}$ 과 같은 값으로 설정될 수 있다.

- [62] 제1 서버 장치(200)는 수신된 동형 암호문을 복호화하지 않고, 암호문 상태로 저장할 수 있다.
- [63] 제2 서버 장치(300)는 동형 암호문에 대한 특정 처리 결과를 제1 서버 장치(200)로 요청할 수 있다. 제1 서버 장치(200)는 제2 서버 장치(300)의 요청에 따라 특정 연산을 수행한 후, 그 결과를 제2 서버 장치(300)로 전송할 수 있다.
- [64] 일 예로, 두 개의 전자 장치(100-1, 100-2)가 전송한 암호문 ct_1, ct_2 가 제1 서버 장치(200)에 저장된 경우, 제2 서버 장치(300)는 두 전자 장치(100-1, 100-2)로부터 제공된 정보들을 합산한 값을 제1 서버 장치(200)로 요청할 수 있다. 제1 서버 장치(200)는 요청에 따라 두 암호문을 합산하는 연산을 수행한 후, 그 결과 값($ct_1 + ct_2$)을 제2 서버 장치(300)로 전송할 수 있다.
- [65] 동형 암호문의 성질상, 제1 서버 장치(200)는 복호화를 하지 않은 상태에서 연산을 수행할 수 있고, 그 결과 값도 암호문 형태가 된다. 본 개시에서는 연산에 의해 획득된 결과값을 연산 결과 암호문이라 지칭한다.
- [66] 제1 서버 장치(200)는 연산 결과 암호문을 제2 서버 장치(300)로 전송할 수 있다. 제2 서버 장치(300)는 수신된 연산 결과 암호문을 복호화하여, 각 동형 암호문들에 포함된 데이터들의 연산 결과값을 획득할 수 있다.
- [67] 제1 서버 장치(200)는 사용자 요청에 따라 연산을 수차례 수행할 수 있다. 이때, 제1 서버 장치(200)는 덧셈, 뺄셈 및 곱셈만으로 구성되는 연산뿐만 아니라, 인버스 연산, 제곱근 연산을 포함하는 연산도 수행할 수 있다. 또한, 제1 서버 장치(200)는 최대값 산출, 최소값 산출 및 크기 비교와 같은 비교 연산도 수행할 수 있다. 이와 같이 제1 서버 장치(200)는 연산 동작을 수행할 수 있다는 점에서, 연산 장치라 지칭될 수도 있다.
- [68] 이때, 제1 서버 장치(200)는 비교 연산을 수행함에 있어서, 비다항식 연산에 대응되는 합성 함수를 이용하여 연산을 수행할 수 있다. 본 개시에 따른 합성 함수는 $2^{-\alpha}$ 정확도에 대응하여 $\theta(\alpha)$ 또는 $\theta(\alpha \log \alpha)$ 복잡도(complexity)를 갖는바, 정확도가 높아지는 경우에 종래보다 빠른 연산이 가능하다.
- [69] 한편, 도 1에서는 제1 전자 장치 및 제2 전자 장치에서 암호화를 수행하고, 제2 서버 장치가 복호화를 수행하는 경우를 도시하였으나, 이에 한정되는 것은 아니다.
- [70]
- [71] 도 2는 본 개시의 일 실시 예에 따른 연산 장치의 구성을 나타낸 블록도이다.
- [72] 구체적으로, 도 1의 시스템에서 제1 전자 장치, 제2 전자 장치 등과 같이 동형 암호화를 수행하는 장치, 제1 서버 장치 등과 같이 동형 암호문을 연산하는 장치, 제2 서버 장치 등과 같이 동형 암호문을 복호화하는 장치 등을 연산 장치라고 지칭할 수 있다. 이러한 연산 장치는 PC(Personal computer), 노트북, 스마트폰, 태블릿, 서버 등 다양한 장치일 수 있다.

- [73] 도 2를 참조하면, 연산 장치(400)는 통신 장치(410), 메모리(420), 디스플레이(430), 조작 입력 장치(440) 및 프로세서(450)를 포함할 수 있다.
- [74] 통신 장치(410)는 연산 장치(400)를 외부 장치(미도시)와 연결하기 위해 형성되고, 근거리 통신망(LAN: Local Area Network) 및 인터넷망을 통해 외부 장치에 접속되는 형태뿐만 아니라, USB(Universal Serial Bus) 포트 또는 무선 통신(예를 들어, WiFi 802.11a/b/g/n, NFC, Bluetooth) 포트를 통하여 접속되는 형태도 가능하다. 이러한 통신 장치(410)는 송수신부(transceiver)로 지칭될 수도 있다.
- [75] 통신 장치(410)는 공개 키를 외부 장치로부터 수신할 수 있으며, 연산 장치(400) 자체적으로 생성한 공개 키를 외부 장치로 전송할 수 있다.
- [76] 그리고 통신 장치(410)는 외부 장치로부터 메시지를 수신할 수 있으며, 생성한 동형 암호문을 외부 장치로 송신할 수 있다.
- [77] 또한, 통신 장치(410)는 암호문 생성에 필요한 각종 파라미터를 외부 장치로부터 수신할 수 있다. 한편, 구현시에 각종 파라미터는 후술하는 조작 입력 장치(440)를 통하여 사용자로부터 직접 입력받을 수 있다.
- [78] 또한, 통신 장치(410)는 외부 장치로부터 동형 암호문에 대한 연산을 요청받을 수 있으며, 그에 따라 계산된 결과를 외부 장치에 전송할 수 있다. 여기서 요청받은 연산은 덧셈, 뺄셈, 곱셈과 같은 연산일 수 있으며, 비다항식 연산인 비교 연산일 수도 있다.
- [79] 메모리(420)에는 연산 장치(400)에 관한 적어도 하나의 인스트럭션(instruction)이 저장될 수 있다. 구체적으로, 메모리(420)에는 본 개시의 다양한 실시 예에 따라 연산 장치(400)가 동작하기 위한 각종 프로그램(또는 소프트웨어)이 저장될 수 있다.
- [80] 이러한 메모리(420)는 RAM이나 ROM, 플래시 메모리, HDD, 외장 메모리, 메모리 카드 등과 같은 다양한 형태로 구현될 수 있으며, 어느 하나로 한정되는 것은 아니다.
- [81] 메모리(420)는 암호화할 메시지를 저장할 수 있다. 여기서 메시지는 사용자가 각종 인용한 각종 신용 정보, 개인 정보 등일 수 있으며, 연산 장치(400)에서 사용되는 위치 정보, 인터넷 사용 시간 정보 등 사용 이력 등과 관련된 정보일 수도 있다.
- [82] 그리고 메모리(420)는 공개 키를 저장할 수 있으며, 연산 장치(400)가 직접 공개 키를 생성한 장치인 경우, 비밀 키뿐만 아니라, 공개 키 및 비밀 키 생성에 필요한 각종 파라미터를 저장할 수 있다.
- [83] 그리고 메모리(420)는 후술한 과정에서 생성된 동형 암호문을 저장할 수 있다. 그리고 메모리(420)는 외부 장치에서 전송한 동형 암호문을 저장할 수도 있다. 또한, 메모리(420)는 후술하는 연산 과정에서의 결과물인 연산 결과 암호문을 저장할 수도 있다.
- [84] 디스플레이(430)는 연산 장치(400)가 지원하는 기능을 선택받기 위한 사용자

인터페이스 창을 표시한다. 구체적으로, 디스플레이(430)는 연산 장치(400)가 제공하는 각종 기능을 선택받기 위한 사용자 인터페이스 창을 표시할 수 있다. 이러한 디스플레이(430)는 LCD(liquid crystal display), OLED(Organic Light Emitting Diodes) 등과 같은 모니터일 수 있으며, 후술할 조작 입력 장치(440)의 기능을 동시에 수행할 수 있는 터치 스크린으로 구현될 수도 있다.

- [85] 디스플레이(430)는 비밀 키 및 공개 키 생성에 필요한 파라미터의 입력을 요청하는 메시지를 표시할 수 있다. 그리고 디스플레이(430)는 암호화 대상이 메시지를 선택하는 메시지를 표시할 수 있다. 한편, 구현시에 암호화 대상은 사용자가 직접 선택할 수도 있고, 자동으로 선택될 수 있다. 즉, 암호화가 필요한 개인 정보 등은 사용자가 직접 메시지를 선택하지 않더라도 자동으로 설정될 수 있다.
- [86] 조작 입력 장치(440)는 사용자로부터 연산 장치(400)의 기능 선택 및 해당 기능에 대한 제어 명령을 입력받을 수 있다. 구체적으로, 조작 입력 장치(440)는 사용자로부터 비밀 키 및 공개 키 생성에 필요한 파라미터를 입력받을 수 있다. 또한, 조작 입력 장치(440)는 사용자로부터 암호화될 메시지를 설정받을 수 있다.
- [87] 프로세서(450)는 연산 장치(400)의 전반적인 동작을 제어한다. 구체적으로, 프로세서(450)는 메모리(420)에 저장된 적어도 하나의 인스트럭션을 실행함으로써 연산 장치(400)의 동작을 전반적으로 제어할 수 있다. 이러한 프로세서(450)는 CPU(central processing unit), ASIC(application-specific integrated circuit)과 같은 단일 장치로 구성될 수 있으며, CPU, GPU(Graphics Processing Unit) 등의 복수의 장치로 구성될 수도 있다.
- [88] 프로세서(450)는 전송하고자 하는 메시지가 입력되면 메모리(420)에 저장할 수 있다. 그리고 프로세서(450)는 메모리(420)에 저장된 각종 설정 값 및 프로그램을 이용하여, 메시지를 동형 암호화할 수 있다. 이 경우, 공개 키가 사용될 수 있다.
- [89] 프로세서(450)는 암호화를 수행하는데 필요한 공개 키를 자체적으로 생성하여 사용할 수도 있고, 외부 장치로부터 수신하여 사용할 수도 있다. 일 예로, 복호화를 수행하는 제2 서버 장치(300)가 공개 키를 다른 장치들에게 배포할 수 있다.
- [90] 자체적으로 키를 생성하는 경우, 프로세서(450)는 Ring-LWE 기법을 이용하여 공개 키를 생성할 수 있다. 구체적으로 설명하면, 프로세서(450)는 먼저 각종 파라미터 및 링을 설정하여, 메모리(420)에 저장할 수 있다. 파라미터의 예로는 평문 메시지 비트의 길이, 공개 키 및 비밀 키의 크기 등이 있을 수 있다.
- [91] 링은 다음과 같은 수학식으로 표현될 수 있다.
- [92] [수학식 2]
- [93]
$$R = \mathbb{Z}_q[X]/(f(x))$$
- [94] 여기서 R은 링, $\mathbb{Z}_q$ 는 계수, $f(x)$ 는 n차 다항식이다.
- [95] 링(Ring)이란 기 설정된 계수를 가지는 다항식의 집합으로, 원소들 사이에

덧셈과 곱셈이 정의되어 있으며 덧셈과 곱셈에 대해서 닫혀 있는 집합을 의미한다. 이러한 링은 환으로 지칭될 수 있다.

- [96] 일 예로, 링은 계수가 $\mathbb{Z}_q$ 인 n 차 다항식의 집합을 의미한다. 구체적으로는, n 이 $\Phi(N)$ 일 때, N 차 사이클로토믹 다항식 (N -th cyclotomic polynomial)을 의미한다. $(f(x))$ 란 $f(x)$ 로 생성되는 $\mathbb{Z}_q[x]$ 의 이데알(ideal)을 나타낸다. Euler totient 함수 $\Phi(N)$ 이란 N 과 서로소이고 N 보다 작은 자연수의 개수를 의미한다. $\Phi_N(x)$ 를 N 차 사이클로토믹 다항식으로 정의하면, 링은 다음과 같은 수학식 3으로도 표현될 수 있다.
- [97] [수학식 3]
- [98] $R = \mathbb{Z}_q[x]/(\Phi_N(x))$
- [99] 비밀 키(sk)는 다음과 같이 표현될 수 있다.
- [100] 한편, 상술한 수학식 3의 링은 평문 공간에서 복소수를 갖는다. 한편, 동형 암호문에 대한 연산 속도를 향상하기 위하여, 상술한 링의 집합 중 평문 공간이 실수인 집합만을 이용할 수도 있다.
- [101] 이와 같은 링이 설정되면, 프로세서(450)는 링으로부터 비밀 키(sk)를 산출할 수 있다.
- [102] [수학식 4]
- [103] $sk \leftarrow (1, s(x)), s(x) \in R$
- [104] 여기서, $s(x)$ 는 작은 계수로 랜덤하게 생성한 다항식을 의미한다.
- [105] 그리고 프로세서(450)는 링으로부터 제1 랜덤 다항식($a(x)$)을 산출한다. 제1 랜덤 다항식은 다음과 같이 표현될 수 있다.
- [106] [수학식 5]
- [107] $a(x) \leftarrow R$
- [108] 또한, 프로세서(450)는 에러를 산출할 수 있다. 구체적으로, 프로세서(450)는 이산 가우시안 분포 또는 그와 통계적 거리가 가까운 분포로부터 에러를 추출할 수 있다. 이러한 에러는 다음과 같이 표현될 수 있다.
- [109] [수학식 6]
- [110] $e(x) \leftarrow D_{\alpha q}^n$
- [111] 에러까지 산출되면, 프로세서(450)는 제1 랜덤 다항식 및 비밀 키에 에러를 모듈러 연산하여 제2 랜덤 다항식을 산출할 수 있다. 제2 랜덤 다항식은 다음과 같이 표현될 수 있다.
- [112] [수학식 7]
- [113] $b(x) = -a(x)s(x) + e(x) \pmod{q}$
- [114] 최종적으로 공개 키(pk)는 제1 랜덤 다항식 및 제2 랜덤 다항식을 포함하는 형태로 다음과 같이 설정된다.
- [115] [수학식 8]
- [116] $pk = (b(x), a(x))$

- [117] 상술한 키 생성 방법은 일 예에 불과하므로, 반드시 이에 한정되는 것은 아니며, 이 밖에 다른 방법으로 공개 키 및 비밀 키를 생성할 수도 있음은 물론이다.
- [118] 한편, 프로세서(450)는 공개 키가 생성되면, 다른 장치들에 전송되도록 통신 장치(410)를 제어할 수 있다.
- [119] 그리고 프로세서(450)는 메시지에 대한 동형 암호문을 생성할 수 있다. 구체적으로, 프로세서(450)는 메시지에 대해서 앞서 생성된 공개 키를 적용하여 동형 암호문을 생성할 수 있다. 이때, 프로세서(450)는 암호문의 길이를 스케일링 팩터의 크기에 대응되도록 생성할 수 있다.
- [120] 그리고 프로세서(450)는 동형 암호문이 생성되면 메모리(420)에 저장하거나, 사용자 요청 또는 기 설정된 디폴트 명령에 따라 동형 암호문을 다른 장치에 전송하도록 통신 장치(410)를 제어할 수 있다.
- [121] 한편, 본 개시의 일 실시 예에 따르면, 패킹(packing)이 이루어질 수도 있다. 동형 암호화에서 패킹을 이용하게 되면, 다수의 메시지를 하나의 암호문으로 암호화하는 것이 가능해진다. 이 경우, 연산 장치(400)에서 각 암호문들 간의 연산을 수행하게 되면, 결과적으로 다수의 메시지에 대한 연산이 병렬적으로 처리되므로 연산 부담이 크게 줄어들게 된다.
- [122] 구체적으로는, 프로세서(450)는 메시지가 복수의 메시지 벡터로 이루어지는 경우, 복수의 메시지 벡터를 병렬적으로 암호화할 수 있는 형태의 다항식으로 변환한 후, 그 다항식에 스케일링 팩터를 승산하고 공개 키를 이용하여 동형 암호화할 수도 있다. 이에 따라, 복수의 메시지 벡터를 패킹한 암호문을 생성할 수 있다.
- [123] 그리고 프로세서(450)는 동형 암호문에 대한 복호가 필요한 경우, 동형 암호문에 비밀 키를 적용하여 다항식 형태의 복호문을 생성하고, 다항식 형태의 복호문을 디코딩하여 메시지를 생성할 수 있다. 이때 생성한 메시지는 앞서 설명한 수학식 1에서 언급한 바와 같이 에러를 포함할 수 있다.
- [124] 그리고 프로세서(450)는 암호문에 대한 연산을 수행할 수 있다. 구체적으로, 프로세서(450)는 동형 암호문에 대해서 암호화된 상태를 유지한 상태에서 덧셈, 뺄셈, 또는 곱셈 등의 연산을 수행할 수 있다.
- [125] 또한, 프로세서(450)는 암호문에 대해서도 덧셈, 뺄셈 또는 곱셈 이외의 연산을 갖는 다항식에 대한 연산을 수행할 수 있다. 구체적으로, 동형 암호문은 덧셈, 뺄셈, 곱셈에 대해서는 닫혀있으나, 이외의 연산에 대해서는 닫혀있지 않다.
- [126] 따라서, 덧셈, 뺄셈, 곱셈 이외의 연산에 대해서는 상술한 3개의 연산으로 표현되는 근사 연산식을 이용하여야 한다. 이러한 점에서, 프로세서(450)는 덧셈, 뺄셈 또는 곱셈 이외의 연산에 대해서는 요청되는 연산에 대응되는 근사 함수를 이용하여 연산을 수행할 수 있다.
- [127] 이를 위하여, 종래 널리 알려진 근사함수(예를 들어, Taylor, Least square, minimax)를 이용할 수 있으나, 종래의 근사 함수는 정확도(2^{-n})에 대해서 복잡도는 $O(2^{n/2})$ 를 갖는다. 즉, 종래 방식은 정확도에 대해서 지수적으로 연산이

복잡해지는데, 정확도가 높아질 수록 많은 연산 시간이 필요해 진다.

- [128] 이러한 문제점을 해결하기 위하여, 본 개시에서는 $2^{-\alpha}$ 정확도에 대응하여 $\theta(\alpha)$ 또는 $\theta(\alpha \log \alpha)$ 복잡도(complexity)를 갖는 합성 함수, 즉 근사 함수를 이용한다. 구체적으로, 본 개시에서는 일반적인 다항식보다 훨씬 효율적인 평가를 갖는 잘 구조화된 다항식(well-structured polynomials), 즉 합성 함수를 이용한다.
- [129] 본 개시에서는 인버스 연산, 제곱근, 최대 값 또는 최소 값, 크기 비교 각각에 대한 합성 함수를 제안하고, 각 연산에 대한 합성 함수의 적용 방식은 도 5 내지 도 10을 참조하여 후술한다.
- [130] 그리고 연산 장치(400)는 연산이 완료되면, 연산 결과 데이터로부터 유효 영역의 데이터를 검출할 수 있다. 구체적으로, 연산 장치(400)는 연산 결과 데이터를 라운딩 처리를 수행하여 유효 영역의 데이터를 검출할 수 있다. 라운딩 처리란 암호화된 상태에서 메시지의 반올림(round-off)을 진행하는 것을 의미하며, 다르게는 리스케일링(rescaling)이라고 할 수도 있다.
- [131] 또한, 연산 장치(400)는 연산 결과 암호문 내의 근사 메시지 비중이 임계치를 초과하면, 암호문에 대한 재부팅 동작을 수행할 수 있다.
- [132]
- [133] 도 3은 본 개시의 연산 장치의 연산 동작을 설명하기 위한 도면이다.
- [134] 도 3을 참조하면, 각 동형 암호문(10, 20)은 근사 메시지 영역(11, 21)을 각각 포함할 수 있다. 근사 메시지 영역(11, 21)에는 메시지 및 에러(m_1+e_1, m_2+e_2)가 함께 들어가 있다.
- [135] 연산 장치(400)는 두 동형 암호문(10, 20)을 입력 값으로 하여, 특정 연산을 수행할 수 있다. 이러한 연산은 덧셈, 뺄셈, 곱셈으로 구성되는 다항식 연산일 수 있으며, 근사 함수의 이용이 필요한 다항식 연산 또는 비다항식 연산일 수도 있다.
- [136] 비다항식 연산에서 가장 대표적인 연산은 비교 연산으로, 최대값 산출, 최소값 산출, 크기 비교일 수 있다. 최대 값 산출은 두 값에서 큰 값을 출력하는 것이고, 최소 값 산출은 두 값에서 최소 값을 출력하는 것이고, 크기 비교는 어느 하나의 값이 크면 1을 출력하고, 다른 값이 크면 0을 출력하고, 두 값이 같으면 1/2을 출력하는 것이다. 한편, 크기 비교시에 출력되는 수치 값은 일 예에 불과하고, 구현시에는 다른 값을 출력하도록 구현할 수 있다.
- [137] 최대 값 산출은 아래와 같은 수학식 9로 표시할 수 있으며, 최소 값 산출은 아래와 같은 수학식 10으로 표시할 수 있다.
- [138] [수학식 9]
- [139]
$$\max(a, b) = \frac{a+b}{2} + \frac{\sqrt{(a-b)^2}}{2}$$
- [140] 여기서 a 는 제1 동형 암호문, b 는 제2 동형 암호문, $\max(a, b)$ 는 제1 동형 암호문과 제2 동형 암호문 중 큰 값을 갖는 동형 암호문의 값을 출력하는 함수이다.

[141] [수학식 10]

$$[142] \quad \min(a, b) = \frac{a+b}{2} - \frac{\sqrt{(a-b)^2}}{2}$$

[143] 여기서 a는 제1 동형 암호문, b는 제2 동형 암호문, min(a, b)는 제1 동형 암호문과 제2 동형 암호문 중 작은 값을 갖는 동형 암호문의 값을 출력하는 함수이다.

[144] 상술한 수학식 9, 10을 참조하면, 덧셈, 뺄셈, 상수 2 나누기 등은 동형 암호문 상태에서 연산할 수 있으나, 제곱근 연산은 동형 암호문 상태에서 불가능하다. 따라서, 제곱근 연산에 대해서는 근사 다항식이 필요하며, 입력 값을 초기 값으로 갖는 제1 변수와 입력 값에 1을 뺀 값을 초기 값으로 갖는 제2 변수에 대한 제1 함수와 제2 변수를 갖는 제2 함수를 기설정된 횟수만큼 반복 연산하는 합성 함수를 이용한다. 제곱근 연산의 구체적인 동작은 도 6을 참조하여 후술한다.

[145] 이러한 제곱근 연산에 대응되는 합성 함수를 이용하면, 최대 값 산출 및 최소 값 산출은 다음과 같이 표시될 수 있다.

[146] [수학식 11]

$$[147] \quad \max(a, b; d) = \frac{a+b}{2} + \frac{\text{Sqrt}((a-b)^2; d)}{2}$$

[148] [수학식 12]

$$[149] \quad \min(a, b; d) = \frac{a+b}{2} - \frac{\text{Sqrt}((a-b)^2; d)}{2}$$

[150] 여기서, a는 제1 동형 암호문, b는 제2 동형 암호문, d는 반복 횟수, Sqrt((a-b)²; d)는 제곱근 연산에 대응되는 합성 함수이다.

[151] 크기 비교는 $\frac{a^k}{a^k + b^k}$ 와 같은 수학식을 반복 연산하는 것으로 근사할 수 있으며,

상술한 크기 비교에 대응되는 수학식은 동형 암호문에 대한 인버스 연산을 포함한다. 인버스 연산은 동형 암호문 상태에서 바로 불가능한바, 본 개시에서는 인버스 연산에 대응되는 합성 함수를 이용한다. 인버스 연산의 구체적인 동작은 도 5를 참조하여 후술한다.

[152] 한편, 크기 비교는 부호 함수를 이용하여 근사할 수 있으며, 이러한 부호 함수는 0이 아닌 입력 값에서의 함수 출력 값이 수평 하도록 하는 합성 함수(즉, 0보다 큰 입력 값에 대해서는 출력 값이 1에 가까워지고, 0보다 작은 입력 값에 대해서는 출력 값이 -1에 가까워지도록 하는 합성 함수) 및/또는 0이 아닌 입력 값에서 함수 출력 값이 수평 하도록 하는 합성 함수로 근사화할 수도 있다. 이와 같은 합성 함수를 이용하는 동작에 대해서는 도 9 및 도 10을 참조하여 후술한다.

[153] 한편, 상술한 연산에 의한 암호문(30)은 각 근사 메시지 간의 연산 결과(m3+e3)가 담긴 근사 메시지 영역(31)을 포함할 수 있다. 만약, 평문

공간(32)이 없어지거나 한계치보다 작아지게 된 경우, 연산 장치(400)는 재부팅 동작을 수행할 수 있다.

[154]

[155] 도 4는 본 개시의 일 실시 예에 따른 암호문 연산 방법을 설명하기 위한 흐름도이다.

[156] 도 4를 참조하면, 복수의 동형 암호문에 대한 비교 연산 명령을 수신할 수 있다(S410). 이러한 명령은 외부 장치로부터 입력될 수 있으며, 연산 장치에서 직접 입력될 수도 있다. 그리고 이러한 비교 연산은 최소값 산출, 최대값 산출, 크기 비교 중 하나일 수 있다.

[157] 그리고 비교 연산 명령에 대응되는 합성 함수에 복수의 동형 암호문을 반영하여 연산할 수 있다(S420). 구체적으로, $\theta(\alpha)$ 또는 $\theta(\alpha \log \alpha)$ 복잡도(complexity)를 갖는 합성 함수를 이용하여 비교 연산을 수행할 수 있다.

[158] 그리고 연산된 동형 암호문을 출력할 수 있다(S430). 구체적으로, 연산을 요청한 장치에 연산 결과를 출력할 수 있다.

[159] 이상과 같이 본 개시에 따른 암호문 연산 방법은 $2^{-\alpha}$ 정확도에 대응하여 $\theta(\alpha)$ 또는 $\theta(\alpha \log \alpha)$ 복잡도(complexity)로 연산을 진행할 수 있는바, 높은 정확도가 필요한 경우에도 보다 빨리 연산을 수행할 수 있다.

[160]

[161] 도 5는 본 개시의 일 실시 예에 따라 인버스 연산을 합성 함수를 이용하여 계산하는 방법을 설명하기 위한 도면이다.

[162] 도 5를 참조하면, 변수 x 및 양수 d 에 대한 인버스 함수($\text{Inv}(x;d)$)는 2에 입력 값(x)을 뺀 값을 초기 값(a_0)으로 갖는 제1 변수(a_n)와 1에 입력 값(x)을 뺀 값을 초기 값(b_0)으로 갖는 제2 변수(b_n)가 세팅되고, 이후에 제2 변수(b_n)를 갖는 제1 함수(b_n^2)와 제1 변수(a_n)와 제2 변수(b_n)를 갖는 제2 함수($a_n \cdot (1 + B_{n+1})$)가 기설정된 횟수만큼 반복되는 합성 함수이다.

[163] 이러한 인버스 함수는 아래의 수학적식 13과 같은 Goldschmidts' 분할 알고리즘은 변경한 것이다.

[164] [수학적식 13]

$$[165] \quad \frac{1}{x} = \frac{1}{1 - (1 - x)} = \prod_{i=0}^{\infty} (1 + (1 - x)^{2^i}) \approx \prod_{i=0}^d (1 - (1 - x)^{2^i})$$

[166] 이와 같은 인버스에 대한 합성 함수는 $\theta(\alpha)$ 복잡도만을 갖는다.

[167]

[168] 도 6은 본 개시의 일 실시 예에 따라 제곱근 연산을 합성 함수를 이용하여 계산하는 방법을 설명하기 위한 도면이다.

[169] 도 6을 참조하면, 변수 x 및 양수 d 에 대한 제곱근 연산($\text{Sqrt}(x;d)$)은 초기에 입력 값(x)을 초기값(a_0)으로 갖는 제1 변수(a_n)와 입력 값(x)에 1을 뺀 값을 초기 값(b_0)으로 갖는 제2 변수(b_n)가 세팅되고, 이후에 제1 변수(a_n)와 제2 변수(b_n)를 갖는

제1 함수($a_n(1 - \frac{b_n}{2})$)와 제2 변수(b_n)를 포함하는 제2 함수($b_n^2(\frac{b_n-3}{4})$)가

기설정된 횟수만큼 반복되는 합성 함수이다.

[170] 같은 제공근에 대한 합성 함수는 에러 바운드(error bound) $2^{-\alpha}$ 에 대해서 $\theta(\alpha)$ 복잡도를 갖는다.

[171]

[172] 도 7은 본 개시의 일 실시 예에 따른 최대 값 계산 및 최소 값 계산 방법을 설명하기 위한 도면이다.

[173] 도 7을 참조하면, 최대 값 계산 및 최소 값 계산은 상술한 수학식 9, 10과 같이 표현할 수 있으며, 수학식 9 또는 10 내의 제공근 연산은 도 6에 도시한 바와 같은 제공근 연산에 대한 합성 함수를 이용할 수 있다.

[174] 예를 들어, 동형 암호문 A, B에 대한 최대 값 계산이 요청되면, 연산 장치(700)는 도 7에 도시된 바와 같이 ①(A+B)/2 값과 (A-B)/2 값을 계산하고, ②(A-B)/2를 제공 계산하고 제공 계산된 값을 제공근 연산에 대응되는 합성 함수를 이용하여 기설정된 횟수(d)만큼 반복하고, ③ 합성 함수의 결과와 앞서 계산한 (A+B)/2 값을 더하기 연산하여 최대값을 산출할 수 있다.

[175] 한편, 동형 암호문 A, B에 대한 최소값 값 계산이 요청되면, 연산 장치(700)는 도 7에 도시된 바와 같이 ①(A+B)/2 값과 (A-B)/2 값을 계산하고, ②(A-B)/2를 제공 계산하고 제공 계산된 값을 제공근 연산에 대응되는 합성 함수를 이용하여 기설정된 횟수(d)만큼 반복하고, ③ 앞서 계산한 (A+B)/2 값에 합성 함수의 결과를 빼기 연산하여 최소값을 산출할 수 있다.

[176] 한편, 이상에서는 두 개의 값을 비교하여 최대값 또는 최소값을 출력하는 것으로 도시하고 설명하였지만, 구현시에는 세 개 이상의 동형 암호문에 대해서도 최대값 산출 및 최소값 산출을 수행할 수도 있다. 예를 들어, 세 개의 동형 암호문이 있다면, 제1과 제2 동형 암호문을 비교한 이후에 그 결과와 나머지 동형 암호문을 비교하여 셋 중에 가장 큰 값 또는 셋 중에 가장 작은 값을 산출할 수도 있다.

[177]

[178] 도 8은 본 개시의 제1 실시 예에 따른 크기 비교 방법을 설명하기 위한 도면이다.

[179] 예를 들어, 두 값이 다르면 0 또는 1의 값을 출력하고, 두 값이 동일하면 0.5를 출력하는 비교 함수는 다음과 같이 근사화할 수 있다.

[180] [수학식 14]

[181]
$$\text{comp}(a, b) \approx \sigma_k(\log a - \log b) = \frac{e^{k \log a}}{e^{k \log a} + e^{k \log b}} = \frac{a^k}{a^k + b^k}$$

[182] 여기서, a는 제1 동형 암호문, b는 제2 동형 암호문, comp(a,b)는 입력된 두 개의 동형 암호문을 비교하여 a가 크면 1을 출력하고, b가 크면 0을 출력하고, a와 b가 동일하면 0.5를 출력하는 함수이다.

- [183] 수학식 14를 참조하면, 비교 함수에 대한 근사 함수는 인버스 연산이 포함되어 있다. 따라서, 수학식 14와 같은 근사 함수를 이용하여 동형 암호문 A, B에 대한 크기 비교 연산을 수행하는 경우, 연산 장치(400)는 도 8에 도시된 바와 같이 초기 값(a_0, b_0)을 산출하고, 초기값 산출 이후에 ① 도 5와 관련하여 설명한 바와 같은 인버스에 대응되는 합성함수를 이용하여 $\text{Inv}(a^{m_n} + b^{m_n}; d)$ 연산을 수행할 수 있다. 그리고, ② 인버스 연산 값에 a^{m_n} 값을 곱하는 연산, ③ 1에 앞서 계산된 값(a_{n+1})을 빼는 연산을 기설정된 횟수(t)만큼 반복하여 비교 연산 결과 값을 산출할 수 있다.
- [184]
- [185] 도 9는 본 개시의 제2 실시 예에 따른 크기 비교 방법을 설명하기 위한 도면이다.
- [186] 한편, 크기 비교 함수는 아래와 같이 부호 함수로 나타낼 수 있다.
- [187] [수학식 15]
- [188] $\text{comp}(a, b) = (\text{sgn}(a - b) + 1) / 2$
- [189] 여기서, a 는 제1 동형 암호문, b 는 제2 동형 암호문, $\text{comp}(a, b)$ 는 입력된 두 개의 동형 암호문을 비교하여 a 가 크면 1을 출력하고, b 가 크면 0을 출력하고, a 와 b 가 동일하면 0.5을 출력하는 함수이다. 그리고, $\text{sgn}()$ 는 부호 함수이다.
- [190] 따라서, 크기 비교 함수에 대한 근사화는 부호 함수에 대한 근사화로 나타낼 수 있다.
- [191] 부호 함수가 같은 수학식 16을 만족하는 경우, 부호 함수 계산을 위한 합성함수는 아래와 같은 수학식 17과 같이 표현될 수 있다.
- [192] [수학식 16]
- [193] Prop I. $f(-x) = -f(x)$
 Prop II. $f(1) = 1, f(-1) = -1$
 Prop III. $f'(x) = c(1 - x^2)^n$ for some constant $c > 0$
- [194] 여기서 $f(x)$ 는 다항 함수이다.
- [195] [수학식 17]
- [196]
$$f_n(x) = \sum_{i=0}^n \frac{1}{4^i} \cdot \binom{2i}{i} \cdot x (1 - x^2)^i$$
- [197] 이와 같은 부호 함수 계산을 위한 합성 함수는 0보다 큰 입력 값에 대해서는 출력값이 1에 가까워지고, 0보다 작은 입력 값에 대해서는 출력 값이 -1에 가까워지도록 하는 합성 함수이다.
- [198] 도 9를 참조하면, 동형 암호문에 대한 크기 비교 연산 명령이 요청되면, 연산 장치(400)는 ① 두 값의 차이를 계산하는 연산($a - b$), ② 해당 연산 값을 수학식 17을 이용하여 기설정된 횟수(d)만큼 반복 연산하는 단계, ③ 연산 값에 상수 연산을 수행하는 단계를 수행하여 비교 연산을 수행할 수 있다.
- [199] 한편, 도 9의 방식은 부호 함수가 양 끝단이 1 또는 -1로 수렴하도록, 즉 수렴하도록 하는 방식을 이용한 것이다. 한편, 이상적인 부호 함수의 또 다른 특징은

0의 값에서 기울기가 무한대의 값 즉 0에서의 기울기가 수직 형태를 가져야 한다는 점에서, 이하에서는 이러한 점을 고려한 합성 함수를 설명한다.

[200]

[201] 도 10은 부호 함수의 기울기에 대응되는 합성 함수를 찾는 방법을 설명하기 위한 도면이다.

[202] 크기 비교 함수에 대한 근사화는 앞서 설명한 바와 같이 부호 함수에 대한 근사화로 나타낼 수 있다.

[203] [수학식 18]

[204] Prop I. $g(-x) = -g(x)$ (Origin Symmetry)
 Prop IV. $\exists 0 < \delta < 1$ s.t. $x < g(x) \leq 1$ for all $x \in (0, \delta]$. (Toward $[1 - c, 1]$)
 and $g([\delta, 1]) \subseteq [1 - c, 1]$ (Keep in $[1 - c, 1]$)

[205] 여기서 $g(x)$ 는 부호 함수 계산을 위한 또 다른 합성 함수이고, c 는 출력 값의 수렴 범위로, $1/4$, $1/3$ 등 1보다 작은 값으로 정해질 수 있다.

[206] 부호 함수가 수학식 18을 만족하는 경우, 연산 장치(400)는 도 10에 도시된 바와 같은 방식을 이용하여 $g(x)$ 에 대한 함수를 산출할 수 있다.

[207] 그리고 도 10과 같은 방식으로 합성 함수를 산출한 경우, 도 9와 관련하여 설명한 합성 함수와의 조합을 이용하여 크기 비교 방법을 수행할 수도 있다. 이에 대해서는 도 11을 참조하여 이하에서 설명한다.

[208]

[209] 도 11은 본 개시의 제3 실시 예에 따른 크기 비교 방법을 설명하기 위한 도면이다.

[210] 도 11을 참조하면, 동형 암호문에 대한 크기 비교 연산 명령이 요청되면, 연산 장치(400)는 ① 두 값의 차이를 계산하는 연산(a-b), ② 해당 연산 값을 도 10과 같은 방식으로 산출된 합성 함수($g_n(x)$)에 기설정된 횟수(d_g)만큼 반복 연산하는 단계, ③ 해당 연산 값을 수학식 17($f_n(x)$)을 이용하여 기설정된 횟수(d_f)만큼 반복 연산하는 단계, ④ 연산 값에 상수 연산을 수행하는 단계를 수행하여 비교 연산을 수행할 수 있다.

[211]

[212] 도 12는 부호 함수에 대응되는 합성 함수의 반복 횟수별 형태를 도시한 도면이고, 도 13은 부호 함수와 관련된 다양한 합성 함수의 반복 횟수별 형태를 도시한 도면이다.

[213] 도 12를 참조하면, 도 9와 관련하여 설명한 합성 함수($f_n(x)$)는 반복 횟수가 증가할수록 더욱 이상적인 형태를 갖게됨을 확인할 수 있다.

[214] 도 13의 좌측 도면을 참조하면, 도 10과 관련하여 설명한 합성 함수는 도 9의 합성 함수보다 큰 기울기를 갖지만 양 끝단에서 오차가 발생함을 확인할 수 있다. 따라서, 초기에는 도 10의 합성 함수를 이용하고, 후기에는 도 9의 합성 함수를 이용하는 경우, 즉 도 11과 같은 방식으로 이용한 조합된 합성 함수를

이용하면, 도 9의 합성 함수만을 이용하는 것보다 동일한 반복횟수 대비보다
이상적인 형태를 가질 수 있다.

[215] 구체적으로, 도 13의 우측 도면을 참조하면, 도 9의 합성 함수를 5회 적용한
것보다 도 10의 합성 함수를 2회 적용하고, 도 9의 합성 함수를 3회 적용한 것이
보다 이상적인 형태를 가짐을 확인할 수 있다.

[216] 한편, 상술한 다양한 실시 예에 따른 암호문 처리 방법은 각 단계들을 수행하기
위한 프로그램 코드 형태로 구현되어, 기록 매체에 저장되고 배포될 수도 있다.
이 경우, 기록 매체가 탑재된 장치는 상술한 암호화 또는 암호문 처리 등의
동작들을 수행할 수 있다.

[217] 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.

[218] 이상 첨부 도면을 참고하여 본 개시에 대해서 설명하였지만 본 개시의
권리범위는 후술하는 특허청구범위에 의해 결정되며 전술한 실시 예 및/또는
도면에 제한되는 것으로 해석되어서는 안 된다. 그리고 특허청구범위에 기재된
개시의, 당업자에게 자명한 개량, 변경 및 수정도 본 개시의 권리범위에
포함된다는 점이 명백하게 이해되어야 한다.

산업상 이용가능성

[219] -

[220]

청구범위

- [청구항 1] 복수의 동형 암호문에 대한 비교 연산 명령을 입력받는 단계;
상기 비교 연산 명령에 대응되는 합성 함수에 상기 복수의 동형 암호문을 반영하여 연산하는 단계; 및
상기 연산된 동형 암호문을 출력하는 단계;를 포함하는 암호문 연산 방법.
- [청구항 2] 제1항에 있어서,
상기 합성 함수는 $2^{-\alpha}$ 정확도에 대응하여 $\theta(\alpha)$ 또는 $\theta(\alpha \log \alpha)$ 복잡도(complexity)를 갖는 암호문 연산 방법.
- [청구항 3] 제1항에 있어서,
상기 비교 연산은 최대값 산출 또는 최소값 산출이고,
상기 연산하는 단계는,
제공된 연산에 대응되는 합성 함수를 이용하여 상기 최대값 산출 또는 상기 최소값 산출을 연산하는 암호문 연산 방법.
- [청구항 4] 제3항에 있어서,
상기 제공된 연산에 대응되는 합성 함수는,
입력값을 초기 값으로 갖는 제1 변수와 상기 입력 값에 1을 뺀 값을 초기 값으로 갖는 제2 변수를 포함하는 제1 함수 및 상기 제2 변수를 포함하는 제2 함수를 기설정된 횟수만큼 반복 연산하는 함수인 암호문 연산 방법.
- [청구항 5] 제4항에 있어서,
상기 기설정된 횟수는
상기 제공된 연산의 오차율에 대응되는 횟수인 암호문 연산 방법.
- [청구항 6] 제1항에 있어서,
상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고,
상기 연산하는 단계는,
인버스 연산에 대응되는 함수 연산을 포함하는 합성 함수를 기설정된 횟수만큼 반복 연산하는 암호문 연산 방법.
- [청구항 7] 제1항에 있어서,
상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고,
상기 연산하는 단계는,
부호 함수에 대응되는 합성 함수를 이용하여 기설정된 횟수만큼 반복 연산하는 암호문 연산 방법.
- [청구항 8] 제7항에 있어서,
상기 부호 함수에 대응되는 합성 함수는,
0보다 큰 입력 값에 대해서는 출력값이 1에 가까워지고, 0보다 작은 입력 값에 대해서는 출력 값이 -1에 가까워지도록 하는 합성 함수인 암호문 연산 방법.
- [청구항 9] 제7항에 있어서,

상기 연산하는 단계는,

상기 부호 함수에 대응되는 근사 함수이며 서로 다른 두 개 이상의 합성 함수를 이용하여 상기 비교 연산을 수행하는 암호문 연산 방법.

[청구항 10]

연산 장치에 있어서,

적어도 하나의 인스트럭션(instruction)을 저장하는 메모리; 및

상기 적어도 하나의 인스트럭션을 실행하는 프로세서;를 포함하고,

상기 프로세서는, 상기 적어도 하나의 인스트럭션을 실행함으로써,

복수의 동형 암호문에 대한 비교 연산 명령을 입력받으면, 상기 비교 연산

명령에 대응되는 합성 함수에 상기 복수의 동형 암호문을 반영하여

연산하고, 상기 연산된 동형 암호문을 출력하는 연산 장치.

[청구항 11]

제10항에 있어서,

상기 비교 연산은 최대값 산출 또는 최소값 산출이고,

상기 프로세서는,

제공된 연산에 대응되는 합성 함수를 이용하여 상기 최대값 산출 또는

상기 최소값 산출을 연산하는 연산 장치.

[청구항 12]

제10항에 있어서,

상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고,

상기 프로세서는,

인버스 연산에 대응되는 함수 연산을 포함하는 합성 함수를 기설정된

횟수만큼 반복 연산하는 연산 장치.

[청구항 13]

제10항에 있어서,

상기 비교 연산은 크기 비교에 따라 기설정된 값을 출력하는 연산이고,

상기 프로세서는,

부호 함수에 대응되는 합성 함수를 이용하여 기설정된 횟수만큼 반복

연산하는 연산 장치.

[청구항 14]

제13항에 있어서,

상기 부호 함수에 대응되는 합성 함수는,

0보다 큰 입력 값에 대해서는 출력값이 1에 가까워지고, 0보다 작은 입력

값에 대해서는 출력 값이 -1에 가까워지도록 하는 합성 함수인 암호문

연산 장치.

[청구항 15]

제13항에 있어서,

상기 프로세서는,

상기 부호 함수에 대응되는 근사 함수이며 서로 다른 두 개 이상의 합성

함수를 이용하여 상기 비교 연산을 수행하는 암호문 연산 장치.