TTA Standard

정보통신단체표준(국문표준)

TTAx.xx-xx.xxxx 제정일: 2019년 xx월 xx일

근사연산 동형암호 알고리즘

Homomorphic Encryption for Arithmetic of Approximate Numbers

TTA 한국정보통신기술협회
Telecommunications Technology Association

표준초안 검토 위원회 **정보보호기반 프로젝트그룹(PG501)**

표준안 심의 위원회 **정보보호 기술위원회(TC5)**

	성명	소 속	직위	위원회 및 직위	표준번호
표준(과제) 제안	이석윤	서울대학교	교수	특별위원	TTAx.xx-xx.xxxx
표준 초안 작성자	이석윤	서울대학교	교수	특별위원	TTAx.xx-xx.xxxx
	천정희	서울대학교	교수		
	이동건	서울대학교	연구원		
사무국 담당	황예지	TTA	전임연구원	-	

본 표준 발간 이전에 접수된 지식재산권 확약서 정보는 본 표준의 '부록(지식재산권 확약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 확약서는 TTA 웹사이트에서 확인할 수 있습니다.
본 표준과 관련하여 접수된 확약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장
발행처 : 한국정보통신기술협회
13591, 경기도 성남시 분당구 분당로 47
Tel : 031-724-0114, Fax : 031-724-0109
발행일 : 2019.XX

서 문

1 표준의 목적

이 표준의 목적은 기존 암호 체계와는 달리 데이터를 암호화하여 근사 연산을 수행할 수 있는 동형암호 알고리즘을 규정하는 것이다.
근사연산 동형암호로 암호화된 데이터는 평문 데이터가 유출될 가능성 없이 암호화된 상태로 서버나 클라우드에서 연산할 수 있으며 수학적으로 증명 가능한 안전성을 가진다. 이 표준은 민감한 개인 정보를 포함하고 있는 데이터를 암호화한 상태에서 복호화하지 않고 통계분석을 할 수 있어, 개인정보보호 이슈와 데이터 분석 활용이 상충되는 현재 문제를 해결할 수 있다. 이를 통해 금융, 의료, 마케팅 분야에서 개인정보를 취급하거나 데이터를 보관 처리하는 클라우드 서비스, 빅데이터 분석 서비스 등 데이터 보호 관련 4차 산업혁명의 새로운 시장을 창출할 수 있다.

2 주요 내용 요약

이 표준은 근사 연산을 지원하는 동형암호를 정의하고 있다. 본문에서 근사연산 동형암호의 암호문 생성 방법과 암호문에 대한 근사연산방법을 제시하고 재부팅을 정의한다.

3 인용 표준과의 비교

3.1 인용 표준과의 관련성

해당 사항 없음

Preface

1 Purpose of Standard

The standard is to define a homomorphic encryption algorithm which, unlike the existing cryptographic schemes, can perform approximated operations on encrypted data.Encrypted data by the homomorphic encryption for arithmetics of approximated numbers(HEAAN) can be computed on the server or in the cloud in an encrypted state without the possibility of data being leaked, while HEAAN has mathematically verifiable security. The standard allows statistical analysis of data that contains sensitive personal information without decryption, thereby solving the current conflict between privacy issues and data analysis. This will create a new market for the fourth industrial revolution related to data protection, such as cloud services that handle personal information or archive data, and big data analysis services in the financial, medical and marketing sectors.

2 Summary of Contents

The standard defines a homomorphic encryption algorithm that supports approximated operations. The standard defines a method for generating cipher-text of the algorithm and an approximation method for cipher-text, and the reboots.

3 Relationship to Reference Standards

- None.

목 차

근사연산 동형암호 알고리즘 (Homomorphic Encryption for Arithmetic of Approximate Numbers)

1 적용 범위

동형암호는 암호화된 정보를 복호화 과정 없이 연산 및 가공하는 것이 가능한 알고리즘이다. 동형암호 알고리즘을 이용한 데이터 처리는 비 동형암호를 이용한 경우보다 안전성 면에서 두 가지 장점을 지닌다. 첫째, 데이터 처리 과정에서 복호화된 데이터가 나타나지 않으므로, 평문데이터의 유출을 방지할 수 있다. 둘째, 암호문의 복호화 횟수를 최소화함으로써, 개인키를 안전하게 보호할 수 있다.
근사연산 동형암호란 계산 결과에 유의미한 영향을 미치지 않는 소수점 아래 값을 버리는 근사 연산 방식을 채택함으로써 동형암호의 연산속도를 개선한 암호 알고리즘을 말한다. 기존의 동형암호는 실수연산을 지원하기 위해서 암호문의 크기를 지수적으로 증가시켜야 하는 문제가 있어 암호화된 데이터의 연산이 실용적이지 못하다. 근사 연산 동형암호는 위에서 기술한 동형암호의 장점을 그대로 살릴 수 있으면서 암호화된 데이터의 연산속도를 획기적으로 개선하여 암호문 처리의 효율성을 제고한다.

본 표준에서는 근사 연산을 지원하는 동형암호화 기술에 대하여 암·복호화 방법과 덧셈, 곱셈 등의 동형연산, 노후 암호문의 재부팅 과정을 정의하고 안전성을 위한 매개변수를 기술한다. 근사 연산 동형암호 알고리즘을 실제로 구현한 코드를 명시한다.

2 인용 표준

해당 사항 없음

3 용어 정의

3.1 동형암호(Homomorphic encryption)
암호문 사이의 연산 결과가 평문에서 연산 결과의 암호문이 되는 암호 알고리즘

3.2 재부팅(bootstrapping)
연산을 여러번 거친 암호문에 포함된 잡음 데이터를 줄여 연산가능 횟수를 늘리는 작업

3.3 근사 연산 동형암호
동형 연산을 근사 연산으로 지원함으로써 동형암호의 연산속도를 제고한 암호 알고리즘

4 약어

헤안	HEAAN(Homomorphic Encryptions for Arithmetics of Approximate Number) 근사연산 동형암호 알고리즘의 이름

5 기호

본 절에서는 본문에서 사용될 기호들을 표로 정리한다. 수식 기호, 변수 기호, 매개변수 기호 순으로 나열하여 정의한다.

<표 5-1> 표 제목

$\mathbb{Z}$	모든 정수를 모은 환
$\mathbb{R}$	모든 실수를 모은 체
$\mathbb{C}$	모든 복소수를 모은 체
$R=\mathbb{Z}[x]/(x^{2n}+1)$	정수계수 다항식을 $x^{2n}+1$로 나누어 차수 2n-1 이하인 나머지를 모은 환
$\mathbb{Z}[x]/(q,x^{2n}+1)$	정수계수 다항식을 $x^{2n}+1$로 나누 차수 2n-1 이하인 다항식에 대해, 다시 각 계수를 정수 q로 나눈 나머지로 취한 다항식을 모은 환
$\mathbb{R}[x]/(x^{2n}+1)$	실수계수 다항식을 $x^{2n}+1$로 나누어 차수 2n-1 이하인 나머지를 모은 환
$R_{q_L}, R_{q_\ell}, R_{q_{\ell-1}}, R_{P\cdot q_L}$	$R=\mathbb{Z}[x]/(x^{2n}+1)$과 정수 $q_L, q_\ell, q_{\ell-1}$에 대해, 각각 순서대로 $\mathbb{Z}[x]/(q_L,x^{2n}+1)$, $\mathbb{Z}[x]/(q_\ell,x^{2n}+1)$, $\mathbb{Z}[x]/(q_{\ell-1},x^{2n}+1)$, $\mathbb{Z}[x]/(P\cdot q_L,x^{2n}+1)$

m	평문 메시지
sk	개인키
ct	암호문
evk	계산키의 집합

$dnum$	evk에 포함되어 있는 계산키의 개수
h	해밍 거리(hamming distance)
λ	안전성 매개변수
L	암호문의 레벨. 재부팅 없이 곱셈을 수행할 수 있는 최대 횟수
N	암호문을 정의할 환 및 그 다항식의 차수
p	자리수 조정성분. $\log_2$ p가 근사값의 유효자리(비트)인 정수 p
q	암호문의 모듈러스

6 동형암호를 위한 일반 모델

7절의 근사 연산 동형암호 설명을 위해 동형암호를 이루는 기본 모델을 서술한다.

6.1 구성

동형암호는 다음 세 가지 장치로 구성되어 있다.
암호자: 공개키 pk를 이용하여 동형암호화를 수행하는 장치
복호자: 개인키 sk를 이용하여 동형암호의 복호화를 수행하는 장치
연산자: 암호문 상에서 동형 연산을 수행하는 장치

6.2 장치 핵심 역할

개인키 sk는 복호자 외에는 비밀로 지켜져야 한다.
공개키 pk는 암호자 또는 연산자에게 공개되어야 한다.
매개변수 parameters는 공개한다.

6.3 알고리즘

동형암호 알고리즘은 키생성, 암호화, 복호화, 동형 덧셈, 동형 곱셈의 다섯 가지 알고리즘으로 구성되어 있다. 키생성과 암호화 알고리즘은 확률적 알고리즘이지만, 복호화, 동형 덧셈, 동형 곱셈 알고리즘은 결정적 알고리즘이다.

6.3.1 키생성 KeyGen(λ)

보안매개변수 λ가 주어지면, 공개키 pk, 개인키 sk, 매개변수 paremeters를 생성한다.

6.3.2 암호화 Encrypt(m, pk, parameters)

공개키 pk, 매개변수 parameters, 평문 m이 주어지면, 암호문 ct를 만든다.

6.3.3 복호화 Decrypt(c, sk, parameters)

개인키 sk, 매개변수 parameters, 암호문 ct가 주어지면, 평문 m을 만든다.

6.3.4 동형 덧셈

공개키 pk, 매개변수 parameters, 암호문 공간의 두 암호문 ct_1, ct_2을 서로 더한다.

6.3.5 동형 곱셈

공개키 pk, 매개변수 parameters, 암호문 공간의 두 암호문 ct_1, ct_2을 서로 곱한다.

6.4 기능상 요구사항

위의 알고리즘들은 그 관계에 있어 정확성을 만족해야 하며, 동형성질을 제공해야 한다.

6.4.1 정확성

위의 알고리즘의 관계가 정확성을 가진다는 것은, 키생성 알고리즘이 생성한 공개키, 개인키, 매개변수와 암호화 알고리즘, 복호화 알고리즘이 다음과 같은 관계를 가진다는 뜻이다 : 임의의 평문 m을 키생성에 의해 만들어진 공개키 pk와 매개변수 parameters 로 암호화하여 암호문 c를 만들었을 때, 키생성에 의해 만들어진 개인키 sk와 매개변수로 c를 복호화 하면 그 결과는 최초의 평문 m이다. 표기법은 다음과 같다.

$$\text{Decrypt}(\text{Encrypt}(m,\ pk,\ \text{parameters}),\ sk,\ \text{parameters})=m$$

6.4.2 동형 성질

두 평문 m_1, m_2과 그 각각의 암호문 ct_1, ct_2가 주어졌을 때, 두 암호문의 덧셈 및 곱셈의 결과는 그 암호문의 평문 간의 덧셈, 곱셈 각각의 암호문이 되어야 한다.

$$ct_1 = \text{Encrypt}(m_1,\ pk,\ \text{parameters})$$
$$ct_2 = \text{Encrypt}(m_2,\ pk,\ \text{parameters})$$

$$\text{Decrypt}(ct_1 + ct_2\ ,\ sk,\ \text{parameters})=m_1 + m_2$$
$$\text{Decrypt}(ct_1 \times ct_2\ ,\ sk,\ \text{parameters})=m_1 \times m_2$$

7 근사 연산 동형암호

이 절에서는 암호화된 데이터의 근사 연산을 수행하는 동형암호 알고리즘인 혜안(HEAAN)의 암·복호화 방법과 근사 연산 과정, 노후화된 암호문의 재부팅 과정을 기술하고 안전성을 위한 매개변수를 제시한다.

7.1 개요

근사 연산 동형암호는 기존의 동형암호와 그 구성이 같으나, 키생성과 복호화 알고리즘

에서 차이가 있다. 주어진 평문과 그 암호문에 대해, 근사 연산 동형암호의 복호화 알고리즘은 암호문을 평문의 근사값을 출력한다. 이 때 근사값의 오차 범위는 키생성 단계에서 사전에 결정하거나 계산해둔다.

7.2 인코딩·디코딩

혜안(HEAAN)의 암호화 알고리즘은 다항식을 평문으로 취하여 암호화하는 알고리즘이다. 혜안(HEAAN)은 임의의 데이터를 암호화하기 위해 데이터를 다항식으로 변환하는 인코딩 단계를 거치며, 반대로 암호문을 복호화한 이후에는 데이터로 되돌리는 디코딩 단계를 가진다.
인코딩 단계는 다중메시지 패킹과 자릿수 조정의 두 단계로 이루어져 있다. 다중메시지 패킹은 숫자 메시지 여러개로 구성된 메시지 벡터를 하나의 다항식으로 치환하는 작업이다. 자릿수 조정은 암호화하기 위한 데이터의 유효숫자를 지정하는 작업이다.

7.2.1 다중메세지 패킹(packing multiple messages)

<표 7-1> 다중메시지 패킹 알고리즘

입력: n개의 복소수 데이터 $\mathrm{Data}_1,\ldots,\mathrm{Data}_n$
출력: 정수 N, 실수계수의 2n-1차 다항식 $\mathrm{m'(x)}$
1. $N=2n$, $z_1=\exp(\frac{\pi i}{N}), z_2=\exp(\frac{3\pi i}{N})\cdots, z_n=\exp(\frac{(2n-1)\pi i}{N})$ 2. $m'(z_1)=\mathrm{Data}_1,\ldots,m'(z_n)=\mathrm{Data}_n$이고 차수가 2n-1인 실계수 다항식 $m'(x)$ 출력

혜안(HEAAN)이 사용하는 인코딩 방식은 다중메시지 패킹(multiple message packing)이다. 다중메시지 패킹은 다항식 하나에 가능한 한 많은 데이터를 포함함으로써 암호문이 사용하는 메모리의 낭비를 줄이고, 또한 여러 데이터를 동시에 암호화할 수 있는 형식을 제공하여 암호화의 속도를 제고한다.
그 방법으로서, 지정된 점에서 각각의 메시지(숫자)를 값으로 가지는 다항식을 주어진 메시지 벡터의 평문으로 삼는다. 이와 같이 패킹할 경우, 메시지 벡터의 성분별 덧셈 및 곱셈의 패킹과 평문 간의 덧셈 및 곱셈의 결과가 일치하게 된다.
2의 멱승인 n개의 메시지를 동시에 동형암호화하기 위한 구체적인 부호화(encoding) 방법은 다음과 같다. $z=\exp(\frac{\pi i}{2n})$와 그 거듭제곱들 $z^1, z^3, \cdots, z^{2n-1}, z^{2(n+1)-1},\ldots, z^{4n-1}$을 근으로 하는 정수 기약다항식 $x^{2n}+1$을 선택하고, 평문의 기본 환을 $R=\mathbb{Z}[x]/(x^{2n}+1)$로 정한다. 이 때 $z_1=z^{2\times 1-1},\cdots,z_n=z^{2n-1}$으로 표기한다.
기본 환을 포함하는 환 $\mathbb{R}[x]/(x^{2n}+1)$ 위에서 정의되는 기본 매장함수(canonical embedding function) J: $\mathbb{R}[x]/(x^{2n}+1)\rightarrow\mathbb{C}^n$을 $J(f(x))=(f(z_1),\ldots,f(z_n))$으로 정한다.

7.2.2 자릿수 조정(scaling process)

데이터의 정확도 수준을 지정하기 위해, 혜안(HEAAN)은 키생성 단계에서 자릿수 조정성분(scaling factor) p를 정하여 데이터의 근사 데이터를 만든다.

<표 7-2> 자릿수 조정 알고리즘

입력: $m'(x),\ q, N$
출력: 자릿수 조정 성분 p, $m(x) \bmod q$
1. $p \geq \log_2(8\sqrt{2}N + 6\sqrt{N} + 16\sqrt{hN})$인 정수 2. 입력 다항식 $m'(x)$와 자리수 성분 p를 받아 $pm'(x) = M_{N-1}x^{N-1} + \cdots + M_0$를 계산 3. For $0 \leq i \leq N-1$ 3-1. M_i의 소수점 이하를 반올림하여 정수 $[M_i]$를 계산 3-2. $m_i \equiv [M_i] \bmod q$이고 $-\frac{q}{2} < m_i \leq -\frac{q}{2}$인 정수 m_i 계산 4. $m(x) = m_{N-1}x^{N-1} + \cdots + m_0$를 출력

데이터의 데이터를 부호화한 다항식 $m'(x)$에 p를 곱한다. 이후 $p \cdot m'(x)$의 각 계수를 소숫점 아랫자리에서 반올림하여 근사다항식 $m(x) = [p \cdot m'(x)]$을 얻는다. 여기서 기호 [ ]는 괄호 사이의 다항식의 계수를 소수점 아래에서 반올림한다는 표기법이다. 데이터를 부호화한 평문은 $m(x) \bmod q$로 한다. 이 때 $\bmod q$의 의미는 다항식의 모든 계수를 q로 나눈 나머지를 취하여 그 결과로 $-\frac{q}{2} <$ 계수 $\leq \frac{q}{2}$인 다항식이라는 뜻이다.

7.2.3 디코딩

<표 7-3> 디코딩 알고리즘

입력: $m(x),\ p$
출력: $\overrightarrow{m} = (\mathrm{Data}_1, \ldots, \mathrm{Data}_n)$
1. 실수 p의 역수 p^{-1}를 입력 다항식에 곱하여 $M(x) = p^{-1}m(x)$를 계산 2. For $1 \leq i \leq n$ 2-1. 소숫점 아랫자리에서 반올림으로 $\mathrm{Data}_i = [M(z_i)]$ 계산 3. 벡터 $\overrightarrow{m} = (\mathrm{Data}_1, \cdots, \mathrm{Data}_n)$을 출력

평문 다항식 $m(x)$가 주어져 있을 때, 디코딩 단계에서는 그 다항식에 자릿수 조정성분 p의 실수에서의 역수 p^{-1}를 구하여, 그 역수를 평문 다항식에 곱한다. 그 뒤 다항식의 변수에 $z_1, \cdots, z_n$을 대입하여 메시지벡터 $\overline{m} = (p^{-1}m(z_1), \cdots, p^{-1}m(z_n))$을 얻는다. 결과로 나오는 메시지벡터는 소수점 이하를 반올림함으로써 정수 성분의 벡터를 얻는다.

7.3 암·복호화

근사 연산 동형암호는 지원하는 연산이 근사 연산이라는 점에서 일반 동형암호와 차이가 있다. 이로 인해 키생성·암호화·복호화 알고리즘을 구현함에 있어 근사 연산 동형암호는 일반 동형암호와 다소 차이가 있다. 본 표준에서는 혜안(HEAAN)의 키생성·암호화·복호화 알고리즘을 설명한다.

7.3.1 키생성

<표 7-4> 키생성 알고리즘

입력: 안정성 수준 변수 λ, 환의 랭크 N, hamming 거리 h, 표준편차 σ, 자릿수 조정성분 p, 암호문 모듈러스 q, 최대레벨 L, 계산키 개수 dnum
출력: $P, \mathrm{sk}, \mathrm{pk}, \mathrm{evk}$
1. $q_L = p^L q$를 계산, $P \le q_L$인 정수 2. $s(x) \in R$를 hamming 거리가 h인 랜덤다항식으로 고른다. 3. $a(x) \in R$를 랜덤하게 고른다. 4. $e(x) \in R$은 각 항의 계수가 표준편차가 σ인 이산가우스분포를 따라 고른다. 5. For $1 \le i \le dnum$ 5-1. 각항의 계수가 $-\frac{P \cdot q_L}{2} <$ 계수 $\le \frac{P \cdot q_L}{2}$인 $a_i(x) \in R$를 랜덤하게 고른다. 5-2. $e_i(x)$를 $e(x)$와 동일한 분포를 이용하여 생성 6. $\mathrm{sk} = (-s(x), 1)$, $\mathrm{pk} = (a(x), a(x)s(x) + e(x))$, $evk = \{(a_i'(x), a_i'(x)s(x) + e_i'(x) + P^{i+1} \cdot s^2(x)) \pmod{P \cdot q_L}\}_{1 \le i \le dnum}$

안전성 변수 λ가 주어졌을 때, 정수 L, 자릿수 조정성분 p, $q_L = p^L q$, 정수 P을 고르고, 암호화, 복호화, 연산에 필요한 키를 생성한다. 각각의 역할은 다음과 같다.

λ는 bit-security를 규정한다. 주로 사용하는 λ의 예로는 128, 192, 256이 있다.

q는 암호문이 실제로 가지는 정보량의 모듈러스이다.

q_L은 평문을 암호화 했을 때 생기는 암호문의 모듈러스이며, 이 때 이 암호문의 레벨을 L이라고 한다.

정수 L은 재부팅 과정을 거치지 않은 상태에서 암호문끼리의 곱셈을 수행할 수 있는 최대의 횟수이다. 정수 L보다 작거나 같은 정수 ℓ에 대해 레벨 ℓ 암호문끼리 곱셈을 수행하면 새 암호문 공간의 모듈러스는 $q_{\ell-1}$이 되고, 곱셈 가능한 횟수는 ℓ-1번이 남는다.

정수 P는 암호문끼리의 곱셈 과정에서 모듈러스 변환에 사용되는 값이고 q_L 이하로 잡는다.

0 아닌 정수 h를 정하여 비밀키에 사용될 다항식 $s(x) \in \mathbb{Z}[x]/x^{2n}+1$을 다음과 같이 고른다. 계수가 모두 {-1,0,1}의 원소이고, 그 중 0아닌 계수의 개수는 h 개가 되도록 한다. h=64를 표준으로 제언한다.

이산분포의 표준편차 σ 를 고른다. $a(x) \in R$ 을 랜덤하게 고르되 mod q_L 로 역수인 다항식이

존재하도록 고른다. e(x)∈R은 각 계수가 표준편차가 σ인 이산가우스분포를 따르도록 고른다. σ =3.2를 표준으로 제언한다.
비밀키 $\mathrm{sk}=(-s(x),1)$, 공개키 $\mathrm{pk}=(a(x),a(x)s(x)+e(x))\in(R/q_LR)^2$로 정한다.

계산키에 사용될 파라미터 $dnum \geq 1$을 자연수로 정한다. $dnum = \left\lceil \frac{\log q_L}{\log P} \right\rceil$ 을 표준으로 제언한다. 기호 ⌈ ⌉ 는 실수값의 소숫점 아래 자리를 올림을 한 정수값이다.
$1 \leq i \leq dnum$에 대해 $a_i'(X)\in R_{P\cdot q_L}$를 랜덤하게 고르고, $e_i'(X)$를 $e(X)$와 같은 분포를 이용해 고른다.
계산키 $evk=(a_i(x),a_i'(x)s(x)+e_i'(x)+P^{i+1}\cdot s^2(x)) \pmod{P\cdot q_L}_{1\leq \mathrm{i} \leq \mathrm{dnum}}$를 정한다.
$1 \leq i \leq dnum$에 대해 i번째 evk 원소는 evk_i로 표기하도록 한다.
이 계산키는 암호문의 곱셈을 구현하기 위해 사용된다.

7.3.2 암호화(encryption)

암호화에 사용되는 키는 공개키 $\mathrm{pk}=(a(x),b(x)) \bmod q_L$이며, 공개키는 공개정보이다. 암호자는 평문 $m(x)$를 암호화하기 위해 <표 7-5>와 같은 알고리즘을 따른다.

<표 7-5> 암호화 알고리즘

입력: $m(x),\mathrm{pk},\sigma$
출력: ct
1. 다항식 $v(x)\in R$ 을 다음 규칙으로 생성: 모든 항은 계수가 -1,0,1 중 하나. 계수 0인 항은 n개, 계수 1인 항과 -1인 항은 각각 n/2개 2. 새로운 두 에러 $e_0(x),e_1(x)$를 각 항의 계수가 표준편차가 σ인 이산가우스분포를 따라 고른다. 3. $\mathrm{pk}=(a(x),b(x)) \bmod q_L$ 에 대해, 암호문 $\mathrm{ct}=(v(x)a(x)+e_0(x),v(x)b(x)+m(x)+e_1(x)) \bmod q_L$

위 알고리즘에서 $e_0(x),e_1(x)$를 생성할 시 사용한 이산가우스분포의 표준편차 σ는 키생성 단계에서 사용한 값과 동일하다.

7.3.3 복호화(decryption)

복호화에 사용되는 키는 비밀키 $sk=(-s(x),1)$이며, 비밀키는 정해진 사람에게만 주어진 정보이다. 복호자는 암호문 $ct=(c(x),d(x))$를 복호화하기 위해, 암호문과 비밀키의 내적 값을 구한다.

<표 7-6> 복호화 알고리즘

입력: $\mathrm{ct}, \mathrm{sk}, q$
출력: $m(x) \bmod q$
1. $\mathrm{ct} = (c(x), d(x)) \bmod q_L \quad \mathrm{sk} = (-s(x), 1)$ 2. $m(x) \equiv \langle \mathrm{ct}, \mathrm{sk} \rangle \equiv -c(x)s(x) + d(x) \bmod q$

암호문을 복호화하면 본래의 평문의 근사다항식을 복호문으로 얻는다. 이 근사다항식과 본래 평문 다항식의 계수들은 같은 유효숫자를 가진다.

7.3.4 키 교환(key-switching)

$(-s(x), 1)$가 아닌 $(1, -s(x), s(x)^2)$을 비밀키로 가지는 암호문이 있을 때, 이 암호문의 비밀키가 $(-s(x), 1)$가 되도록 암호문을 변경해준다. 이는 곱셈을 구현하는 방식 중 하나이다.

<표 7-7> 키교환 알고리즘

입력: $(f(x), g(x), h(x)) \in (R/q_\ell)^3$
출력: $\mathrm{ct}_{\mathrm{ksw}} = (c(x), d(x)) \in (R/q_\ell)^2$
1. $f_0(x)$는 $f(x)$의 계수를 모두 P로 나눈 나머지를 취한 다항식이다. 2. For $1 \le i \le dnum - 1$, 2-1. $f_i(x)$는 $\dfrac{f(x) - \sum_{k=0}^{i-1} P^k f_k(x)}{P^i}$ 의 계수를 모두 P로 나눈 나머지를 취한 다항식이다. 3. $f(x) = \sum_{i=0}^{dnum-1} P^i f_i(x)$ $\mathrm{ct}_{\mathrm{ksw}} = (c(x), d(x)) \equiv \left(g(x) + \left[P^{-1} \cdot \left(\sum_{i=0}^{dnum-1} f_i(x) \cdot evk_i \right) \right], h(x) \right) \bmod q_\ell$

$(1, -s(x), s(x)^2)$를 비밀키로 하는 암호문 $(f(x), g(x), h(x)) \in R_{q_\ell}^3$ 이 주어져있을 때, $f(x)$의 각 계수를 P진법으로 표현하여, 각 항별로 $0 \le i \le dnum - 1$ 번째 자리를 계수로 가지는 다항식 $f_i(X)$를 얻어 $f(x) = \sum_{i=0}^{dnum-1} P^i f_i(x)$로 표현한다. 그리고 다음 계산을 수행하여 $(c(X), d(X)) \in R_{q_\ell}^2$을 얻는다. $(c(x), d(x)) = \left(g(x) + \left[P^{-1} \cdot \left(\sum_{i=0}^{dnum-1} f_i(x) \cdot evk_i \right) \right], h(x) \right)$

기호 []는 반올림을 의미한다.

7.4 암호문의 연산

혜안(HEAAN)의 암호화 장치는 평문의 근사값 및 그 유효숫자를 암호화하는 장치이며, 따라서 혜안(HEAAN)이 가지는 동형성질이란 유효숫자의 덧셈과 곱셈으로 구현된다

7.4.1 덧셈

같은 레벨을 가지는 두 암호문 $(c_1(x),d_1(x)) \bmod q_\ell$, $(c_2(x),d_2(x)) \bmod q_\ell$이 주어졌을 때, 암호문의 덧셈은 처음 두 암호문과 같은 레벨을 가지는 새로운 암호문이 된다.

<표 7-8> 덧셈 알고리즘

입력: 같은 레벨 ℓ의 두 암호문 $\mathrm{ct}_1, \mathrm{ct}_2$
출력: $\mathrm{ct}_{\mathrm{add}}$
1. $\mathrm{ct}_1 = (c_1(x),d_1(x)) \bmod q_\ell$, $\mathrm{ct}_2 = (c_2(x),d_2(x)) \bmod q_\ell$ 에 대해 $\mathrm{ct}_{\mathrm{add}} = (c_1(x)+c_2(x),d_1(x)+d_2(x)) \bmod q_\ell$

7.4.2 곱셈

같은 레벨을 가지는 두 암호문 $(c_1(x),d_1(x)) \bmod q_\ell$, $(c_2(x),d_2(x)) \bmod q_\ell$이 주어졌을 때, 암호문의 곱셈은 일차적으로는 처음 두 암호문과 같은 레벨을 가지는 새로운 암호문을 얻는다.

<표 7-9> 곱셈 알고리즘

입력: 같은 레벨 ℓ의 두 암호문 $\mathrm{ct}_1, \mathrm{ct}_2$
출력: 레벨 ℓ인 암호문 $\mathrm{ct}_{\mathrm{mult}}$
1. $\mathrm{ct}_1 = (c_1(x),d_1(x)) \bmod q_\ell$, $\mathrm{ct}_2 = (c_2(x),d_2(x)) \bmod q_\ell$ 에 대해 중간단계 암호문 $(c_1(x) \cdot c_2(x), c_1(x) \cdot d_2(x) + c_2(x) \cdot d_1(x), d_1(x) \cdot d_2(x)) \in R_{q_\ell}^3$ 을 얻는다. 2. 키교환 알고리즘을 이용하여 $\mathrm{ct}_{\mathrm{mult}} = (c(x),d(x)) \bmod q_\ell$ 로 변환

같은 레벨을 가지는 두 암호문 $(c_1(x),d_1(x)) \bmod q_\ell$, $(c_2(x),d_2(x)) \bmod q_\ell$ 의 곱셈을 위해, 다음과 같은 중간 단계 암호문을 생성한다.

$$(f(x),g(x),h(x)) = (c_1(x) \cdot c_2(x), c_1(x) \cdot d_2(x) + c_2(x) \cdot d_1(x), d_1(x) \cdot d_2(x)) \in R_{q_\ell}^3$$

중간 단계 암호문에 키 교환 과정(5.3.4)을 적용하여 새 암호문$\mathrm{ct}_{\mathrm{mult}} = (c(X),d(X)) \in R_{q_\ell}^2$을 얻는다.

7.4.3 레벨 조정

앞서 곱셈 알고리즘을 통해 얻는 새 암호문 $\mathrm{ct}=(c(x),d(x)) \bmod q_\ell$를 그대로 복호화할 경우, 원래의 두 암호문의 평문의 곱셈에 p를 곱한 값의 근사값이 나오게 된다. 이를 조정하기 위해 암호문 $\mathrm{ct} \in (R/q_\ell)^2$ 의 각 계수를 $-\frac{q_\ell}{2} < \text{계수} \le \frac{q_\ell}{2}$ 인 정수를 취한 뒤 실수 p로 나누어 실계수 다항식으로 이루어진 새 암호문 $\frac{1}{p} \cdot \mathrm{ct}$를 모듈러스 $q_{\ell-1}$과 함께 얻는다. 그 후 새 모듈러스 $q_{\ell-1}$에 대해, 조정된 암호문 $\mathrm{ct}_{\mathrm{rs}} = [p^{-1} \cdot \mathrm{ct}] \in (R/q_{\ell-1})^2$을 얻는다. 동형곱셈은 조정된 암호문 $\mathrm{ct}_{\mathrm{rs}}$을 주어진 두 암호문의 곱셈으로 한다.

<표 7-10> 레벨조정 알고리즘

입력: 레벨 ℓ 인 암호문 ct, 자릿수 성분 조정성분 p
출력: 레벨 ℓ-1인 암호문 $\mathrm{ct}_{\mathrm{rs}}$
1. $\mathrm{ct} = (c(x),d(x)) \bmod q_\ell$ 의 두 다항식 $c(x),d(x)$의 계수를 $-\frac{q_\ell}{2} < \text{계수} \le \frac{q_\ell}{2}$인 정수로 고른다. 2. 실수 p^{-1}를 곱한 $(p^{-1}c(x),p^{-1}d(x))$ 를 새 모듈러스 $q_{\ell-1}$ 에서 얻는다. 3. $\mathrm{ct}_{\mathrm{rs}} = [p^{-1} \cdot \mathrm{ct}] \in (R/q_{\ell-1})^2$

7.5 암호문의 재부팅

7.4에서 정의한 암호문 간의 연산, 특히 곱셈이 반복될 경우 새 암호문들의 모듈러스는 작아지게 되는데, 결과적으로 새 암호문들은 덧셈과 곱셈이 가능한 횟수가 줄어든다. 이를 노후화된 암호문이라 한다. 평문을 갓 암호화한 암호문은 연산 가능한 횟수가 최대인 상태인데, 이와 같이 연산 횟수가 최대 상태인 암호문을 새 암호문이라고 한다.
재부팅이란 노후화된 암호문을 새 암호문으로 교환해주는 알고리즘이다. 혜안(HEAAN)에서 사용하는 재부팅의 원리는 노후화된 암호문을 복호화한 뒤 다시 암호화하는 것이다. 단, 이 과정은 모두 암호화된 상태로 수행해야 하며 이를 재암호화(recryption)이라고 한다. 재암호화에는 두 가지 조건이 요구된다.
1. 암호문의 복호화와 복호문의 암호화 과정은 모두 덧셈과 곱셈으로 표현되어야 한다.
2. 재암호화가 발생시키는 노후화가 본래의 암호문에 영향을 미치지 않도록, 별도의 키를 이용하여 재암호화해야 한다.

혜안(HEAAN)에서는 곱셈구현 단계에서 재부팅 알고리즘 고려하고 있으며, 곱셈 후 줄어든 암호문 공간을 qL로 확장하는 것으로 구현되어 있다.
혜안(HEAAN)재부팅 알고리즘의 기반이론 및 실험에 관해서는 부록 I-4의 참조문헌 Bootstrapping for approximate homomorphic encryption을 참조한다.

7.6 추천 매개변수

안전성을 고려한 매개변수의 예시로 <표 7-11>과 같이 기술할 수 있다.

<표 7-11> $\lambda = 128, 192, 256$를 만족하는 추천 매개변수

λ	128	192	256
N	4096	4096	8192
$\log_2(q_L)$	71	41	53
h	64	64	64
σ	3.2	3.2	3.2

부 속 서 A

(본 부속서는 표준 내용의 일부임)

동형암호의 산업적 활용 범례

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. 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, 그 메시지가 제3자에게 손쉽게 유출될 수 있다는 문제점도 있었다. 이러한 문제를 해결하기 위하여 동형 암호화 방법이 연구되고 있다. 동형 암호화는 암호화된 정보를 복호화하지 않고 암호문 자체에서 연산을 하더라도, 평문에 대해 연산한 후 암호화한 값과 동일한 결과를 얻을 수 있는 암호화 알고리즘이다. 따라서, 암호문을 복호화하지 않은 상태에서 각종 연산을 수행할 수 있다. 더욱이 근사 연산 동형암호는 실생활에서 축적되는 근삿값 데이터에 대하여 암호화된 연산을 지원하므로 기계학습을 통한 데이터 분석에 유용하게 활용된다.

부록 I

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

참조구현 값

본문의 알고리즘은 기 구현된 코드로서 https://github.com/snucrypto/HEAAN에 그 코드가 공개되어 있다. 해당 코드와의 비교를 위한 참조구현값은 별도 파일로 첨부한다.

랜덤 시드 생성은 C++코드 기본 제공 함수인 srand함수를 사용.
시드값은 7을 사용함.

SecretKey, PublicKey, Ciphertext는 R/q_L의 원소 두 개를 차수 순서대로 16진법으로 나열한 값임.

부 록 II-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

지식재산권 확약서 정보

II-1.1 지식재산권 확약서

- 발명의 명칭: 동형 암호화를 수행하는 단말 장치와 그 암호문을 처리하는 서버 장치 및 그 방법들
- 권리자의 성명: 서울대학교 산학협력단
- 등록번호: 1020170173608
- 등록(출원) 연월일: 2017년 12월 15일
- 실시조건: 지식재산권을 합리적 조건하에 비차별적으로 실시
- 확약서 접수일:

II-1.2 지식재산권 확약서

- 발명의 명칭: 근사 암호화된 암호문에 대한 연산을 수행하는 장치 및 방법
- 권리자의 성명: 서울대학교 산학협력단
- 출원 번호: 1020180087928
- 출원 연월일:
- 실시조건: 지식재산권을 합리적 조건하에 비차별적으로 실시
- 확약서 접수일:

부 록 II-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

시험인증 관련 사항

해당 사항 없음

부 록 II-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

본 표준의 연계(family) 표준

해당 사항 없음

부 록 II-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

참고 문헌

Homomorphic encryption for arithmetic of approximate numbers
천정희, 김미란, 김안드레이, 송용수 저 (2017년 12월)
International Conference on the Theory and Application of Cryptology and Information Security (pp. 409-437). Springer, Cham.

Cheon, J. H., Han, K., Kim, A., Kim, M., & Song, Y. (2018, April).
Bootstrapping for approximate homomorphic encryption
천정희, 김미란, 김안드레이, 송용수, 한규형 저 (2018년 4월)
Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 360-384). Springer, Cham.

부 록 II-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

영문표준 해설서

해당 사항 없음

부 록 II-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

표준의 이력

판수	채택일	표준번호	내용	담당 위원회
제1판	2019.xx.xx	-	근사연산 동형암호 알고리즘	정보보호기반 프로젝트그룹(PG501)