정 보 보 호 학 회 지
제 32 권 제1호, 2022. 2

함수암호 기술 연구 동향

서 민 혜*

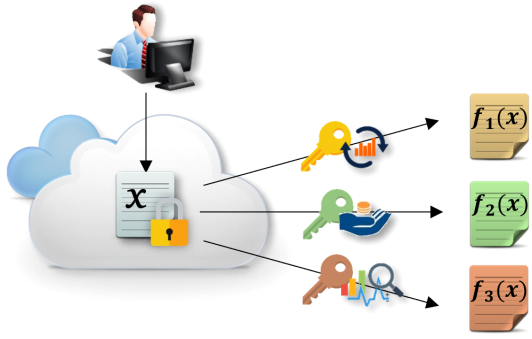
요 약

함수암호(functional encryption)는 프라이버시를 보호하면서 암호화된 데이터에 대한 연산을 수행할 수 있는 진화된 형태의 암호 기술이다. 비밀키를 가진 수신자에게 평문을 전부 제공하는 기존의 암호와 달리, 함수암호는 특정 연산에 대응하는 비밀키를 가진 수신자에게 평문에 대한 연산 결과만을 제공하기 때문에 데이터에 대한 유연한(fine-grained) 접근 제어가 가능하다. 인공지능과 같은 4차 산업혁명 시대의 대표 기술들은 데이터의 활용을 기반으로 하지만 이 과정에서 데이터 노출로 인한 사용자 프라이버시 침해 문제가 발생할 수 있다. 함수암호는 이러한 문제를 해결할 수 있는 기술로써, 프라이버시 보호와 데이터 경제 활성화를 위한 기반 기술로 활용될 수 있다. 본 논문에서는 함수암호 기술에 대한 개념을 설명하고 관련 연구 동향을 소개한다.

Ⅰ. 서 론

기존의 암호 기술은 복호화 키(decryption key)를 소유한 사용자의 경우에는 암호문으로부터 평문에 대한 모든 정보를 얻을 수 있고, 그렇지 않으면 어떠한 정보도 얻어낼 수 없는 all-or-nothing 방법으로 설계되었다. 따라서 기존의 암호 기술을 사용하면 데이터에 대한 기밀성(confidentiality)을 보장할 수는 있으나, 암호화된 데이터를 활용하기 위해서는 다시 암호문을 복호화하여 평문 상태로 만들어야 하기 때문에 데이터의 안전한 활용 관점에서는 한계가 있다.

함수암호(functional encryption)는 암호화된 데이터에 대한 연산을 지원하는 차세대 암호 기술로, 특정 함수 f에 대한 제한된 복호화 키, 즉 함수 키(functional key)를 제공한다. 함수 f에 대한 복호화 키 sk_f와 데이터 x에 대한 암호문 ct_x이 주어졌을 때, 복호화를 통해 연산의 결과인 $f(x)$가 출력되고 그 외에 x에 대한 어떠한 정보도 노출되지 않는다. 따라서 함수암호 기술을 사용하면 데이터 전체를 노출하지 않으면서도 데이터 활용에 필요한 부분 정보만을 제공할 수 있기 때문에 안전한 데이터 활용이 가능하다. 또한 사용자는 다양한 서비스를 이용할 때마다 자신의 데이터를 제공할 필요가 없으며, 데이터 x를 한 번만 암호화하여 클라우드 스토리지에 업로드하면 서비스 제공자가 필요한 연산 f_i에 대한 함수 키를 발급받아 서비스 제공에 필요한 연산 결과 $f_i(x)$를 얻을 수 있다.



〔그림 1〕 함수암호 개요

암호화된 데이터에 대한 연산을 지원하는 암호 기술로 지금까지 동형암호(homomorphic encryption)에 대한 연구가 활발히 진행되어 왔다. 함수암호와 동형암호는 암호문을 복호화하지 않고 데이터에 대한 연산을 수행할 수 있다는 점에서는 공통점이 있지만, 연산의 결과가 제공되는 형태에 있어 차이가 있다. 즉, 동형암호는 복호화 시 연산의 결과가 암호화된 형태로 출력되는 반면 함수암호는 복호화 시 연산의 결과가 평문 형태로

본 연구는 한국연구재단을 통해 과학기술정보통신부의 기초연구사업으로부터 지원받아 수행되었습니다 (과제번호- 2021R1A4A5028900711).
* 덕성여자대학교 사이버보안전공 (조교수, mhseo@duksung.ac.kr)

〔표 1〕 Predicate Encryption 형태의 함수암호

	Functionality	Evaluation
Identity Based Encryption (IBE)	Equality of $ID \in \{0,1\}^*$	$\mathrm{Dec}(sk_{ID}, ct_{ID'}) = \begin{cases} 1 & \text{if } ID = ID' \\ 0 & otherwise \end{cases}$
Attribute Based Encryption (ABE)	Access policy Φ or Attribute set $\vec{z}$	$\mathrm{Dec}(sk_{\Phi}, ct_{\vec{z}}) = \begin{cases} 1 & \text{if } \Phi(\vec{z}) = 1 \\ 0 & otherwise \end{cases}$
Inner Product Encryption (IPE)	$\vec{v} = (v_1, \ldots, v_n) \in F_p^n$	$\mathrm{Dec}(sk_{\vec{v}}, ct_{\vec{w}}) = \begin{cases} 1 & \text{if } \sum_i v_i w_i = 0 \\ 0 & otherwise \end{cases}$
Hidden Vector Encryption (HVE)	$\vec{v} = (v_1, \ldots, v_n)$ where $v_i \in \{*\} \cup \{0,1\}^*$	$\mathrm{Dec}(sk_{\vec{v}}, ct_{\vec{w}}) = \begin{cases} 1 & \text{if } v_i = w_i \text{ for } v_i \neq * \\ 0 & otherwise \end{cases}$

출력된다. 따라서 응용 환경의 특성 및 서비스의 종류에 따라 더 적합한 암호 기술을 선택하여 사용할 수 있다.

함수암호에 대한 개념은 2011년 처음으로 정립되었다 [13]. 기본적인 공개키/대칭키 암호에서 확장된 개념인 ID 기반 암호(identity-based encryption), 속성 기반 암호(attribute-based encryption), 내적 암호(inner-product Encryption), 숨김 벡터 암호(hidden-vector Encryption)와 같은 술어 암호(Predicate Encryption, PE)도 함수암호 개념으로 해석할 수 있다. 즉, 기존의 PE 암호는 boolean 함수, 즉 1과 0만을 연산의 결과로 가지는 제한된 형태의 함수를 지원하는 함수암호로 볼 수 있다. 예를 들어 ID 기반 암호는 암호문과 복호화 키에 대응하는 ID가 동일하면 1을, 그렇지 않으면 0을 출력하는 함수암호로 해석할 수 있다. 2011년 이전에는 대부분 PE 암호 형태의 제한된 형태의 함수암호 기법들이 제안되었으며, 2012년 이후에 임의의 함수를 지원하는 기법에 대한 연구가 본격적으로 시작되었다 [24,25].

임의의 연산을 지원하는 함수암호 기법은 구분 불능 난독화(indistinguishability obfuscation)나 다중 선형 함수(multilinear map)와 같은 강력한 프리미티브를 기반으로 설계가 되었는데 [22,23], 이러한 기반 기술들은 현실적으로 구현하는 것에 한계가 있어서 대부분 이론적인 연구에 그쳤다. 따라서 함수의 표현력을 제한하면서 효율적으로 구현 가능한 함수암호 기법을 설계하는 방향으로 연구가 활발히 진행되고 있으며, 대표적으로 내적 연산과 이차 연산을 지원하는 함수암호 기법들이 다수 제안되었다 [1,9].

또한 함수 프라이버시(function privacy)를 제공하는 함수암호 기법을 개발하는 방향으로도 연구가 진행되고 있다. 암호 기술은 기본적으로 (복호화 키가 안전하게 저장되어 있을 때) 암호문으로부터 평문에 대한 정보가 노출되지 않아야 한다는 성질을 제공한다. 하지만 함수암호 기법에서는 (복호화 키에 해당하는) 함수 키가 함수 연산을 위해 제3자에게 제공될 수 있으며, 경우에 따라서는 함수 자체가 기밀성을 제공해야 하는 정보일 수 있다. 함수 프라이버시는 함수 키로부터 함수에 대한 정보가 노출되지 않아야 한다는 성질을 의미한다. 따라서 함수 프라이버시를 제공하는 함수암호는 평문에 대한 기밀성뿐만 아니라 함수에 대한 기밀성도 함께 제공한다.

함수 프라이버시를 제공하는 함수암호 기법은 대부분 대칭키 기반으로 설계가 되는데, 공개키 기반에서는 완전한 함수 프라이버시를 제공하는 것이 본질적으로 어렵기 때문이다. 공개키 기반에서는 (공격자를 포함한) 누구나 자신이 원하는 데이터 x_i에 대한 암호문을 스스로 생성할 수 있으며, 공격 타겟인 함수 키 sk_f를 이용하여 자신이 생성한 암호문을 복호화함으로써 함수 f^*에 대한 유의미한 정보 $f^*(x_i)$를 얻어볼 수 있다. 현재까지는 페어링(pairing or bilinear map)을 이용하여 함수 프라이버시를 제공하는 함수암호 기법이 다수 설계되었으며 [18,26], 대칭키 기반의 함수암호 기법와 대칭키 암호 기법을 결합하여 제네릭(generic)하게 함수 프라이버시를 제공하는 함수암호 기법을 설계하는 방법도 제안되었다 [12]. 본 논문에서는 이와 같이 함수암호 기술에 관한 다양한 연구 동향을 살펴보고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 함수암호 기술을 이해하는데 필요한 배경지식을 살펴본다. 3장에서는 임의의 연산을 지원하는 함수암호 기술에 대한 연구 동향을, 4장에서는 제한된 연산을 지원하지만 효율적으로 구현 가능한 함수암호 기술에 대한 연구 동향을 소개한다. 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 배경지식

본 장에서는 함수암호 기술을 이해하는데 필요한 배경지식에 대해 살펴보고자 한다. 2.1절에서는 함수암호 기법의 알고리즘을 정의하고 2.2절에서는 함수암호에

서 고려하는 대표적인 안전성 모델을 설명한다.

2.1. 알고리즘 정의

공개키 기반 함수암호는 다음의 4가지 다항 시간(polynomial-time) 알고리즘으로 구성된다.

- **설정(Setup)**: 보안 상수 1^λ를 입력으로 하여 공개 파라미터(public parameter) pp와 마스터 비밀키(master secret key) msk를 출력한다.
- **키 생성(KeyGen)**: 마스터 비밀키 msk와 함수 f를 입력으로 하여 함수키(functional key) sk_f를 출력한다.
- **암호화(Encrypt)**: 공개 파라미터 pp와 메시지 x를 입력으로 하여 암호문 ct_x를 출력한다.
- **복호화(Decrypt)**: 공개 파라미터 pp, 함수키 sk_f, 그리고 암호문 ct_x를 입력으로 하여 $y = f(x)$를 출력한다.

정확성(Correctness). 임의의 보안 상수 1^λ를 입력으로 하여 설정(Setup)으로부터 생성된 (pp, msk), msk와 함수 f를 입력으로 하여 키 생성(KeyGen)으로부터 생성된 sk_f, 그리고 pp와 데이터 x를 입력으로 하여 암호화(Encrypt)로부터 생성된 ct_x에 대하여, 다음 식을 만족할 확률은 1이다.

$$\mathrm{Decrypt}(pp, sk_f, ct_x) = f(x)$$

대칭키 기반 함수암호는 공개키 기반 함수암호와 동일한 4가지 알고리즘으로 구성되지만, 암호화(Encrypt) 알고리즘에서 공개 파라미터 pp 대신 마스터 비밀키 msk를 입력으로 하여 암호문을 생성한다는 점에서 차이가 있다.

2.2. 안전성 모델

2.2.1. 선택 평문 공격에 대한 안전성

공개키 기반 함수암호 기법의 안전성 모델은 다음의 실험 $\mathrm{Exp}_{FE,A}^{IND-CPA}(1^\lambda)$으로 정의된다.

FE=(Setup, KeyGen, Encrypt, Decrypt)를 공개키 기반 함수암호 기법이라 하자. 공격자 A는 보안 상수 1^λ에 대해 챌린저 C와 다음과 같이 가상의 게임(game)을 수행한다.

- **설정(Setup)**: C는 Setup(1^λ)을 수행하여 공개 파라미터 pp와 마스터 비밀키 msk를 생성하고, A에게 pp를 전송한다.
- **키 질의(Key query)**: A는 함수 f를 C에게 질의하고, C는 함수키 $sk_f \leftarrow$ KeyGen(msk, f)를 생성하여 A에게 전송한다. A의 키 질의 개수는 다항식 시간 내에 계산할 수 있는 것으로 제한된다.
- **챌린지(Challenge)**: A는 동일한 길이의 두 메시지 (x_0^*, x_1^*)를 C에게 질의한다. 이때, 두 메시지는 키 질의(Key query)에서 질의된 모든 함수 f에 대하여 $f(x_0^*) = f(x_1^*)$을 만족해야 한다. C는 랜덤하게 $b \in \{0,1\}$를 선택하고, 암호문 $ct_{x_b^*} \leftarrow$ Encrypt (pp, x_b^*)를 생성하여 A에게 전송한다.
- **추측(Guess)**: A는 $b' \in \{0,1\}$을 출력한다. 만약 $b = b'$인 경우 S는 1을 출력하고, 그렇지 않으면 0을 출력한다.

공격자 위의 실험 $\mathrm{Exp}_{FE,A}^{IND-CPA}(1^\lambda)$에서 얻는 이점(advantage)은 다음과 같이 정의한다.

$$Adv_{FE,A}^{IND-CPA}(1^\lambda) = \left| \Pr\left[\mathrm{Exp}_{FE,A}^{IND-CPA}(1^\lambda) = 1\right] - \frac{1}{2} \right|$$

정의 1. 공개키 기반 함수암호 기법에 대한 임의의 다항식 시간 공격자 A에 대하여 공격자의 이점 $Adv_{FE,A}^{IND-CPA}(1^\lambda)$이 무시할 만큼 작은(negligible) 값이라면, 함수암호 기법 FE는 선택 평문 공격에 대하여 안전(indistinguishability under chosen plaintext attack, IND-CPA)하다.

2.2.2. 함수 프라이버시

대칭키 기반 함수암호 기법의 함수 프라이버시 모델은 다음의 실험 $\mathrm{Exp}_{FE,A}^{FP}(1^\lambda)$으로 정의된다.

FE=(Setup, KeyGen, Encrypt, Decrypt)를 대칭키 기반 함수암호 기법이라 하자. 공격자 A는 보안 상수 1^λ에 대해 챌린저 C와 다음과 같이 가상의 게임(game)을 수행한다.

- **설정(Setup)**: C는 보안 상수 1^λ를 입력으로 하여 공개 파라미터 pp와 마스터 비밀키 msk를 생성하고, A에게 pp를 전송한다. 또한 C는 랜덤하게 $b \in \{0,1\}$를 선택한다.
- **키 질의(Key query)**: A는 동일한 크기의 두 함수 (f_0^i, f_1^i)을 C에게 질의하고, C는 함수키 $sk_{f_b^i} \leftarrow$ KeyGen(msk, f_b^i)를 생성하여 A에게 전송한다.
- **암호문 질의(Ciphertext query)**: A는 동일한 길이의 두 메시지 (x_0^j, x_1^j)를 C에게 질의한다. 이때, 두 메시지는 키 질의(Key query)에서 질의된 모든 (f_0^i, f_1^i)에 대하여 $f_0^i(x_0^j) = f_1^i(x_1^j)$을 만족해야 한다. C는 암호문 $ct_{x_b^j} \leftarrow$ Encrypt(msk, x_b^j)를 생성하여 A에게 전송한다.
- **추측(Guess)**: A는 $b' \in \{0,1\}$을 출력한다. 만약 $b = b'$인 경우 S는 1을 출력하고, 그렇지 않으면 0을 출력한다.

공격자 위의 실험 $\mathrm{Exp}_{FE,A}^{FP}(1^\lambda)$에서 얻는 이점(advantage)은 다음과 같이 정의한다.

$$Adv_{FE,A}^{FP}(1^\lambda) = \left| \Pr\left[\mathrm{Exp}_{FE,A}^{FP}(1^\lambda) = 1\right] - \frac{1}{2} \right|$$

정의 2. 대칭키 기반 함수암호 기법에 대한 임의의 다항식 시간 공격자 A에 대하여 공격자의 이점 $Adv_{FE,A}^{FP}(1^\lambda)$이 무시할 만큼 작은(negligible) 값이라면, 함수암호 기법 FE는 완전한 함수 프라이버시(full function privacy, FP)를 제공한다.

Ⅲ. 임의의 연산을 지원하는 함수암호

본 장에서는 다중 선형 함수(Multilinear Maps, MMap)와 구분 불능 난독화(Indistinguishability Obfuscation, iO)를 기반으로 설계된 임의의 연산을 지원하는 함수암호 기법에 대한 연구 동향을 소개한다.

3.1. 다중 선형 함수(MMap) 기반

암호학적인 다중 선형 함수(MMap)는 임의의 정수들 $a_1, \dots, a_n$, 덧셈 순환군(additive cyclic group) $G_1, \dots, G_n$의 원소들 $g_1, \dots, g_n$ $(g_i \in G_i)$, 곱셈(multiplicative) 순환군 G_T가 주어졌을 때, 아래의 식을 만족하는 함수 $e: G_1 \times \cdots \times G_n \rightarrow G_T$를 의미한다.

$$e(g_1^{a_1}, \dots, g_n^{a_n}) = e(g_1, \dots, g_n)^{\prod_{i=1}^{n} a_i}$$

2016년 Garg 등은 다중 선형 함수를 이용하여 임의의 연산을 지원하는 함수암호 기법을 설계하였다 [23]. 그 이후 다중 선형 함수를 직접적으로 이용하는 방법보다는 다중 선형 함수를 이용하여 먼저 구분 불능 난독화를 설계하고, 구분 불능 난독를 이용하여 함수암호를 설계하는 방법으로 연구가 다수 진행되었다.

다중 선형 함수는 이론적으로 의미 있는 암호학적인 도구이지만, 현재까지 제안된 다중 선형 함수들은 공격이 이루어져 안전성에 논란이 있다. 대표적으로, GGH13 다중 선형 함수에 대한 공격 [17], CLT15 다중 선형 함수에 대한 공격 [16], GGH15 다중 선형 함수에 대한 공격 [15] 등이 있다. 따라서 다중 선형 함수를 이용하여 임의의 연산을 지원하는 함수암호 기법을 설계하기 위해서는 안전한 다중 선형 함수를 설계하는 연구가 선행되어야 한다.

3.2. 구분 불능 난독화(iO) 기반

구분 불능 난독화(iO)는 입출력 결과가 동일한(equivalent) 서로 다른 두 개의 회로(circuit) C_0와 C_1에 대하여, C_0을 난독화(obfuscation) 한 것과 C_1을 난독화 한 것을 계산적으로 구분할 수 없어야(computationally indistinguishable) 한다는 성질을 가진다.

임의의 연산을 지원하는 함수암호 기법은 구분 불능 난독화를 이용하여 최초로 설계되었다 [22]. 이후 여러 개의 암호문을 한 번에 처리할 수 있는 다중 입력(multi-input) 함수암호 기법을 설계하는 방법 [21], full-모델에서 안전성 증명이 가능한 기법 [31] 등 이론적인 단계에서 구분 불능 난독화를 이용하여 임의의 연

산을 지원하는 함수암호 기법을 설계하는 연구들이 다수 진행되었다.

구분 불능 난독화는 기존에는 설계가 불가능하다고 여겨졌던 많은 암호 기법들은 설계하는 도구로써 함수암호뿐만 아니라 암호학 전반에서 많은 연구가 진행되었지만, 실용적으로 구현 가능한 수준의 구분 불능 난독화는 현재까지 제안되지 않았으며 대부분의 구분 불능 난독화가 다중 선형 함수(MMap)를 기반으로 설계되어 그 안전성에 대한 논란이 있다. 이러한 이유로 구분 불능 난독화를 이용하여 임의의 연산을 지원하는 함수암호 기법을 설계하는 연구는 아직 이론적인 수준에 머무르고 있다.

구분 불능 난독화를 이용하여 함수암호 기법을 설계하는 것과 별개로, 임의의 연산을 지원하는 함수암호 기법을 이용하여 구분 불능 난독화를 설계하는 연구도 진행되었다 [8,14]. 이를 통해 임의의 연산을 지원하는 함수암호와 구분 불능 난독화를 동치 관계임을 확인할 수 있다.

Ⅳ. 제한된 연산을 지원하는 함수암호

본 장에서는 내적, 이차 등 제한된 연산을 지원하는 효율적인 함수암호 기법에 대한 연구 동향을 소개한다.

4.1. 내적 연산을 지원하는 함수암호

내적(inner product) 연산은 길이가 같은 두 벡터 $x=(x_1,\dots,x_n)$, $y=(y_1,\dots,y_n)$에 대하여 아래와 같이 정의된다.

$$\langle x,y\rangle = x_1y_1+x_2y_2+\cdots+x_ny_n$$

내적 연산은 가중 평균(weighted average)과 같은 통계치 계산, 거리 계산을 통한 유사도(similarity) 측정, 선형 회귀(linear regression) 계산을 통한 예측 모델 생성, 인공지능의 신경망 구조 등 다양한 응용 환경에서 사용되고 있다.

내적 연산을 지원하는 함수암호(Functional Encryption for Inner Product, FE-IP)는 2015년 Abdalla 등에 의해 처음으로 제안되었으며 [1], 이후 안전성 향상, 함수 프라이버시 제공, 다중 입력으로의 확장 등 다양한 방향으로 연구가 진행되었다.

안전성 측면에서, 최초로 제안된 FE-IP 기법 [1]은 공격자가 초기에 공격 목표로 사용할 벡터를 선택해야 하는 selective-모델에서 안전성을 증명하는 한계가 있었으며, Agrawal 등은 이러한 제약이 없는 full-모델에서 안전성 증명이 가능한 기법을 제안하였다 [7]. 이후 기존의 구분 불가능성(indistinguishability) 기반의 증명보다 더 강한 안전성을 제공하는 시뮬레이션(simulation) 기반으로 증명 가능한 FE-IP 기법이 제안되었다 [6].

함수 프라이버시를 제공하는 FE-IP 기법은 2015년 Bishop 등에 의해 처음 제안되었다 [11]. 하지만 안전성 증명 과정에서 키 질의 (x_0,x_1)와 암호문 질의 (y_0,y_1)의 기본적인 조건인 $\langle x_0,y_0\rangle=\langle x_1,y_1\rangle$ 외에도 추가로 $\langle x_0,y_0\rangle=\langle x_0,y_1\rangle=\langle x_1,y_0\rangle=\langle x_1,y_1\rangle$을 만족해야 한다는 제약이 있는 약한(weak) 모델을 사용했다는 점에서 한계가 있었다. 이후 이러한 제약이 없는 완전한 함수 프라이버시(full function privacy)를 제공하는 기법이 제안되었으며 [18], 성능을 향상시키기 위한 연구들도 다수 제안되었다 [10,26].

기능성 확장 측면에서, 다중 입력(multi-input) 암호문을 처리하는 FE-IP 기법이 제안되었으며 [4], 그 외에도 서로 다른 사용자들로부터 생성된 암호문들을 한 번에 복호화 할 수 있는 다중 사용자(multi-client) 기법 [3], 신뢰 기관이 필요 없는 탈중앙화(decentralizing) 기법 [2], 함수키에 사용자 정보를 결합하여 추적이 가능한(traceable) 기법 [19], 신뢰 기관과 암호화 주체를 검증 가능한(verifiable) 기법 [30] 등 다양한 기능이 추가된 FE-IP 기법들이 제안되었다.

4.2. 이차 연산을 지원하는 함수암호

이차 함수(quadratic function) 연산은 $n\times m$ 행렬 $F=\{f_{i,j}\}_{i\in[n],j\in[m]}$와 두 벡터 $x=(x_1,\dots,x_n)$, $y=(y_1,\dots,y_m)$에 대하여 아래와 같이 정의된다.

$$x^TFy=\sum_{i,j}f_{i,j}x_iy_j$$

이차 함수는 분산(variance) 또는 평균 제곱근 편차(root-mean-square)와 같은 통계치, 유클리드 거리(euclidean distance), 이차 회귀(quadratic regression)

모델 등을 표현하는 데 사용할 수 있다.

이차 함수를 지원하는 함수암호는 2017년 Baltico 등에 의해 처음 제안되었으며 [9], 성능을 개선하거나 [32] 다중 입력을 지원하는 기법도 설계되었다 [5]. 또한 이차 연산을 지원하는 함수암호를 이용하여 프라이버시를 보장하는 기계 학습을 구현하는 연구 결과도 발표되었다 [29].

4.3. 기타 연구

최근에는 3차 다항식(cubic polynomial) 연산을 지원하는 함수암호 기법이 처음으로 제안되었으며 [33]. 다항식 연산 외에도 안전한 교집합 연산(Private Set Intersection, PSI)을 지원하는 기법 [27], 순서 노출 암호(Order Revealing Encryption, ORE) [28] 등 다양한 종류의 연산을 지원하는 함수암호 기법들이 제안되고 있다. 또한 하드웨어를 이용한 구현을 통해 함수암호의 실용성을 높이는 연구 결과도 제안되었다 [20].

Ⅴ. 결 론

본 논문에서는 함수암호의 개념에 대한 설명과 연구 동향에 대해 살펴보았다. 함수암호는 프라이버시 문제를 해소하면서 데이터를 안전하게 활용할 수 있게 해주는 차세대 암호 기술로써 인공지능, 자율주행차, IoT, 빅데이터 분석 등 4차 산업혁명 시대의 핵심 기술 및 다양한 서비스에 활용될 것으로 기대된다. 현재까지 효율적으로 동작하는 함수암호는 내적 및 이차 연산을 지원하는 기법에 대한 연구가 대부분이며, 함수암호 기술의 활용도를 높이기 위해서는 다양한 종류의 연산에 대해 효율적으로 동작하는 함수암호 기술 개발이 필요할 것으로 보인다.

참 고 문 헌

[1] M. Abdalla, F. Bourse, A.D. Caro, D. Pointcheval, “Simple Functional Encryption Schemes for Inner Products,” *PKC 2015*, LNCS 9020, pp. 733-751, 2015.

[2] M. Abdalla, F. Benhamouda, M. Kohlweiss, H. Waldner, “Decentralizing Inner-Product Functional Encryption,” *PKC 2019*, LNCS 11443, pp. 128-157, 2019.

[3] M. Abdalla, F. Bourse, H. Marival, D. Pointcheval, A. Soleimanian, H. Waldner, “Multi-Client Inner-Product Functional Encryption in the Random-Oracle Model,” *SCN 2020*, LNCS 12238, pp. 525-545, 2020.

[4] M. Abdalla, R. Gay, M. Raykova, H. Wee, “Multi-input Inner-Product Functional Encryption from Pairings,” *EUROCRYPT 2017*, LNCS 10210, pp. 601-626, 2017.

[5] S. Agrawal, R. Goyal, J. Tomida, “Multi-input Quadratic Functional Encryption from Pairings,” *CRYPTO 2021*, LNCS 12828, pp. 208-238, 2021.

[6] S. Agrawal, B. Libert, M. Maitra, R. Titiu, “Adaptive Simulation Security for Inner Product Functional Encryption,” *PKC 2020*, LNCS 12110, pp. 34-64, 2020.

[7] S. Agrawal, B. Libert, D. Stehle, “Fully Secure Functional Encryption for Inner Products from Standard Assumption,” *CRYPTO 2016*, LNCS 9816, pp. 333-362, 2016.

[8] P. Ananth, A. Jain, “Indistinguishability Obfuscation from Compact Functional Encryption,” *CRYPTO 2015*, LNCS 9215, pp. 308-326, 2015.

[9] C.E.Z. Baltico, D. Catalano, D. Fiore, R. Gay, “Practical Functional Encryption for Quadratic Functions with Applications to Predicate Encryption,” *CRYPTO 2017*, LNCS 10401, pp. 67- 98, 2017.

[10] M. Barbosa, D. Catalano, A. Soleimanian, B. Warinschi, “Efficient Function-Hiding Functional Encryption: From Inner-Products to Orthogonality,” *CT-RSA 2019*, LNCS 11405, pp. 127-148, 2019.

[11] A. Bishop, A. Jain, L, Kowalczyk, “Function-Hiding Inner Product Encryption,” *ASIACRYPT 2015*, LNCS 9452, pp. 470-491, 2015.

[12] Z. Brakerski, G. Segev, “Function-Private Functional Encryption in the Private-Key

Setting," *TCC 2015*, LNCS 9015, pp. 306-324, 2015.

[13] D. Boneh, A. Sahai, B. Waters, "Functional Encryption: Definitions and Challenges," *TCC 2011*, LNCS 6597, pp. 253-273, 2011.

[14] N. Bitansky, V. Vaikuntanathan, "Indistinguishability Obfuscation from Functional Encryption," *FOCS 2015*, pp. 171-190, 2015.

[15] J.H. Cheon, W. Cho, M. Hhan, J. Kim, C. Lee, "Statistical Zeroizing Attack: Cryptanalysis of Candidates of BP Obfuscation over GGH15 Multilinear Map," *CRYPTO 2019*, LNCS 11694, pp. 253-283, 2019.

[16] J.H. Cheon, P.-A. Fouque, C. Lee, B. Minaud, H. Ryu, "Cryptanalysis of the New CLT Multilinear Map over the Integers," *EUROCRYPT 2016*, LNCS 9665, pp. 509-536, 2016.

[17] J.H. Cheon, K. Han, C. Lee, H. Ryu, D. Stehle, "Cryptanalysis of the Multilinear Map over the Integers," *EUROCRYPT 2015*, LNCS 9056, pp. 3-12, 2015.

[18] P. Datta, R. Dutta, S. Mukhopadhyay, "Functional Encryption for Inner Product with Full Function Privacy," *PKC 2016*, LNCS 9614, pp. 164-195, 2016.

[19] X.T. Do, D.H. Phan, D. Pointcheval, "Traceable Inner Product Functional Encryption," *CT-RSA 2020*, LNCS 12006, pp. 564-585, 2020.

[20] B. Fisch, D. Winayagamurthy, D. Boneh, S. Gorbunov, "IRON: Functional Encryption using Intel SGX," *CCS 2017*, pp. 765-782, 2017.

[21] S. Goldwasser, S.D. Gordon, V. Goyal, A. Jain, J. Katz, F.-H. Liu, A. Sahai, E. Shi, H.-S. Zhou, "Multi-input Functional Encryption," *EUROCRYPT 2014*, LNCS 8441, pp. 578-602, 2014.

[22] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, B. Waters, "Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits," *FOCS 2013*, pp. 40-49, 2013.

[23] S. Garg, C. Gentry, S. Halevi, M. Zhandry, "Functional Encryption Without Obfuscation," *TCC 2016*, LNCS 9563, pp. 480-511, 2016.

[24] S. Goldwasser, Y. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich, "Reusable garbled circuits and succinct functional encryption," *STOC 2013*, pp. 555-564, 2013.

[25] S. Gorbunov, V. Vaikuntanathan, H. Wee, "Attribute-based encryption for circuits," *STOC 2013*, pp. 545-554, 2013.

[26] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, D.J. Wu, "Function-Hiding Inner Product Encryption is Practical," *SCN 2018*, LNCS 11035, pp. 544-562, 2018.

[27] K. Lee, M. Seo, "Functional encryption for set intersection in the multi-client setting," *Design, Codes and Cryptography,* vol. 90, pp. 17-47, 2022.

[28] K. Lewi, D.J. Wu, "Order-Revealing Encryption: New Constructions, Applications, and Lower Bounds," *CCS 2016*, pp. 1167-1178, 2016.

[29] T. Ryffel, D. Pointcheval, F. Bach, E. D.-Sans, R. Gay, "Partially Encrypted Deep Machine Learning using Functional Encryption," *NIPS 2019*, vol. 32, pp. 4517-4528, 2019.

[30] N. Soroush, V. Iovion, A. Rial, P.B. Roenne, P.Y.A. Ryan, "Verifiable Inner Product Encryption Scheme," *PKC 2020*, LNCS 12110, pp. 65-94, 2020.

[31] B. Waters, "A Punctured Programming Approach to Adaptively Secure Functional Encryption," *CRYPTO 2015*, LNCS 9216, pp. 678-697, 2015.

[32] H. Wee, "Functional Encryption for Quadratic Functions from k-Lin, Revisited," *TCC 2020*, LNCS 12550, pp. 210-228, 2020.

[33] Z. Zhang, F. Zhang, "Functional encryption for cubic polynomials and implementation," *Theoretical Computer Science,* vol. 885, pp. 41-54, 2021.

〈저 자 소 개〉

서 민 혜 (Minhye Seo)

종신회원

2012년 2월 : 고려대학교 수학과 졸업
2020년 2월 : 고려대학교 정보보호대학원 정보보호학과 박사
2020년 3월~2020년 8월 : 고려대학교 정보보호연구원 연구교수
2020년 9월~현재 : 덕성여자대학교 사이버보안전공 조교수

<관심분야> 암호, 인증, 프라이버시