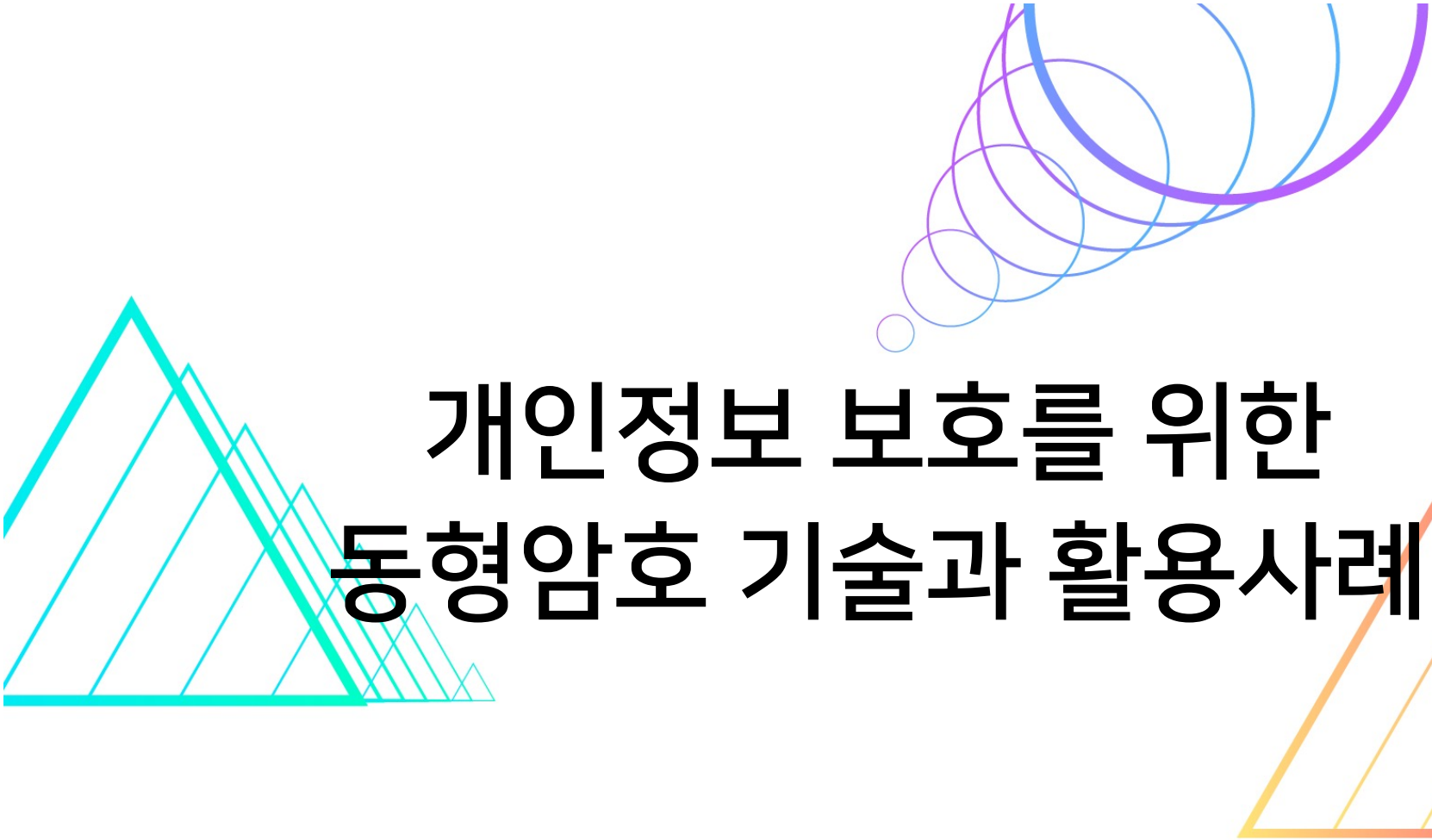


DEVIEW 2021

개인정보 보호를 위한 동형암호 기술과 활용사례

최현민 Naver Cloud

CONTENTS

DEVIEW 2021

1.빅데이터 시대, 개인정보를 보호하는 방법

1.1 데이터 활용 VS 개인의 프라이버시 보호

DEVIEW 2021

데이터 분석에서 개인의 프라이버시 보호 이슈

- 데이터를 통해 부가가치를 이끌어내는 것은 4차산업의 새로운 성장동력임
- 개인정보나 개인의 신용정보등을 통해 데이터 분석을 하는 것은 자칫 개인의 프라이버시를 침해할 수 있는 여지가 있음

1.1 데이터 활용 VS 개인의 프라이버시 보호

개인의 프라이버시 보호 방법 연구의 필요성

- 2020년 2월 데이터3법의 개정으로 데이터 활용의 핵심인 가명정보를 활용하는 것에 대한 법적 근거가 신설됨
- 개인정보처리자는 통계작성, 과학적 연구, 공익적 기록보존 등을 위하여 정보주체의 동의 없이 가명정보 처리 가능 (개인정보보호법 제28조의2)

> **제28조의2(가명정보의 처리 등)** ① 개인정보처리자는 통계작성, 과학적 연구, 공익적 기록보존 등을 위하여 정보주체의 동의 없이 가명정보를 처리할 수 있다.
>
> ② 개인정보처리자는 제1항에 따라 가명정보를 제3자에게 제공하는 경우에는 특정 개인을 알아보기 위하여 사용될 수 있는 정보를 포함해서는 아니 된다.

[출처] 개인정보보호법

법적인 제도하에 개인정보를 분석할 수 있는 길이 열렸지만 개인의 프라이버시를 기술적으로 보호하는 방법에 대한 연구는 지속적으로 필요

1.2 기존 프라이버시 보존 분석 방법과 한계

익명성 기반 프라이버시 보존 분석 방법(k-익명성 등)

- 결합 정보를 이용한 공격 등에 대해 방어를 위한 대표적인 프라이버시 보호 모델

● 〈표 1〉 공개 의료데이터 사례 ●

구분	지역 코드	연령	성별	질병
1	13053	28	남	전립선염
2	13068	21	남	전립선염
3	13068	29	여	고혈압
4	13053	23	남	고혈압
5	14853	50	여	위암
6	14853	47	남	전립선염
7	14850	55	여	고혈압
8	14850	49	남	고혈압
9	13053	31	남	위암
10	13053	37	여	위암
11	13068	36	남	위암
12	13068	35	여	위암

● 〈표 2〉 선거인명부 사례 ●

구분	이름	지역코드	연령	성별
1	김민준	13053	28	남
2	박지훈	13068	21	남
3	이지민	13068	29	여
4	최현우	13053	23	남
5	정서연	14853	50	여
6	송현준	14850	47	남
7	남예은	14853	55	여
8	성민재	14850	49	남
9	윤건우	13053	31	남
10	손윤서	13053	37	여
11	민우진	13068	36	남
12	허수빈	13068	35	여

[출처] 개인정보 비식별 조치 가이드라인

결합 정보(지역코드, 연령, 성별)를 통해 김민준님이 전립선염에 걸린 것을 유추 가능

1.2 기존 프라이버시 보존 분석 방법과 한계

익명성 기반 프라이버시 보존 분석 방법(k-익명성 등)

- 한계 : 공격자의 배경지식 등을 모두 예측하기 어렵기 때문에 배경지식 공격 등에 취약

〈표 3〉 k-익명성 모델에 의해 비식별된 의료데이터 사례

구분	지역 코드	연령	성별	질병	비고
1	130**	< 30	*	전립선염	다양한 질병이 혼재되어 안전
2	130**	< 30	*	전립선염	
3	130**	< 30	*	고혈압	
4	130**	< 30	*	고혈압	
5	1485*	> 40	*	위암	다양한 질병이 혼재되어 안전
6	1485*	> 40	*	전립선염	
7	1485*	> 40	*	고혈압	
8	1485*	> 40	*	고혈압	
9	130**	3*	*	위암	모두가 동일 질병(위암)으로 취약
10	130**	3*	*	위암	
11	130**	3*	*	위암	
12	130**	3*	*	위암	

[출처] 개인정보 비식별 조치 가이드라인

1.2 기존 프라이버시 보존 분석 방법과 한계

DEVIEW 2021

암호화를 통한 프라이버시 보호

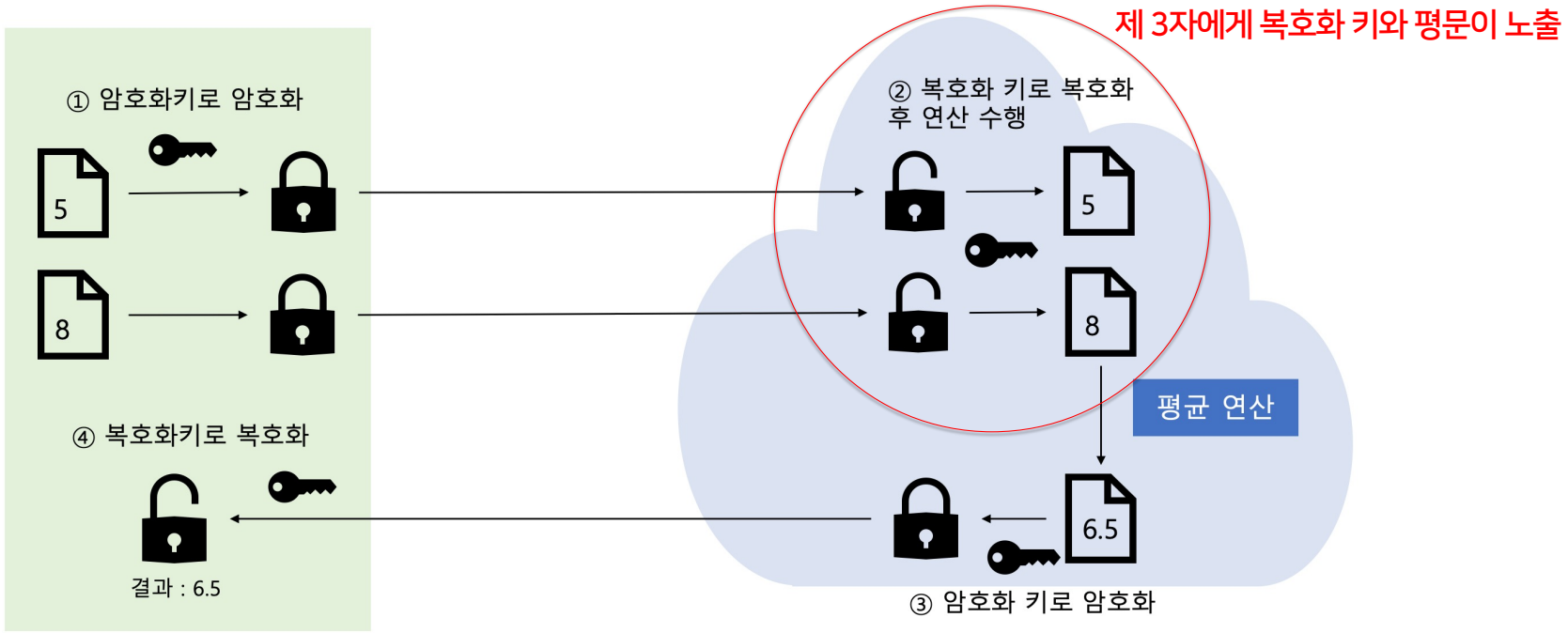
- 데이터 송/수신 및 보관 시 암호화를 통해 데이터의 기밀성을 보장
- 한계 : 데이터 분석을 위해 암호화된 데이터를 반드시 복호화 해야 하므로 데이터의 유출 및 복호화 키 정보의 유출 가능성이 있음

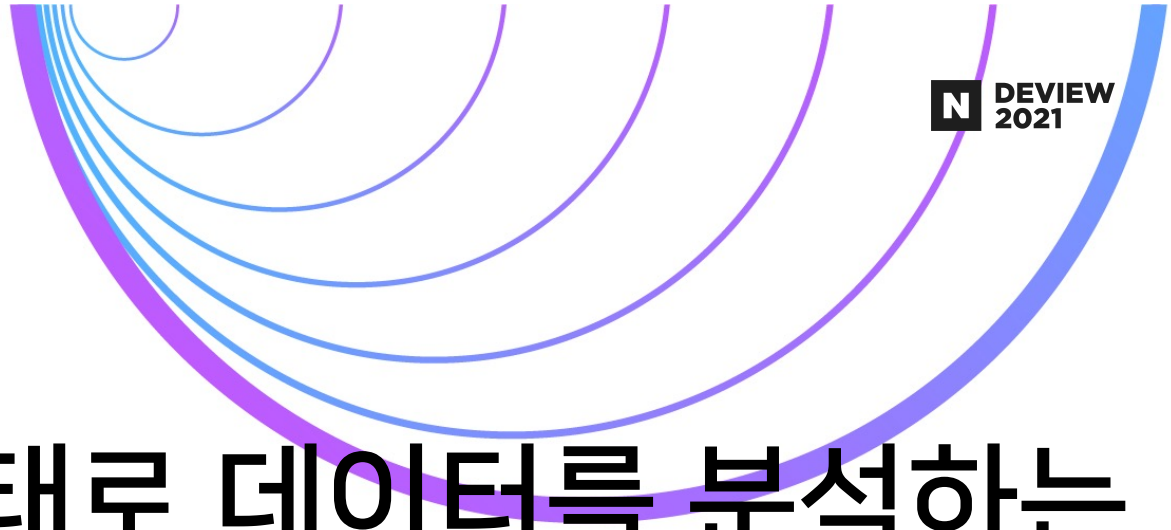
공개키 암호 : 공개키와 비밀키 두 종류의 키로 구성되어 있으며 공개키로 암호화, 비밀키로 복호화 수행가능한 암호화 기법
비밀키 암호 : 하나의 키로 구성되어 있으며 해당 키로 암호화 및 복호화를 모두 수행가능한 암호화 기법

1.2 기존 프라이버시 보존 분석 방법과 한계

DEVIEW 2021

기존 암호화 기술 기반 데이터 분석



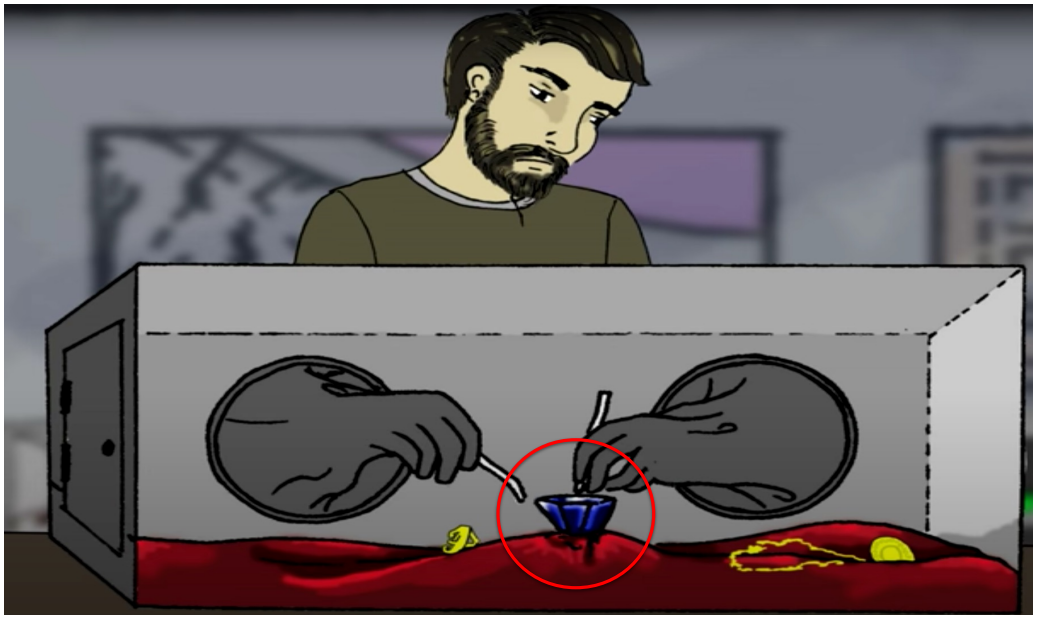


2. 암호화된 상태로 데이터를 분석하는 동형암호 기술

2.1 암호화된 상태로 연산 가능한 동형암호

DEVIEW 2021

정보를 노출하지 않은 채 연산 및 분석 할 수 있는 방법이 있을까?



[출처] 2016 암호여름학교

2.1 암호화된 상태로 연산 가능한 동형암호

DEVIEW 2021

동형암호란

- 암호화된 데이터를 복호화 없이 컴퓨터가 수행하는 모든 계산이 가능한 암호기술
- 2016년 부터 급속도로 실용화 연구가 진행되어 이론적인 연구 뿐만 아니라 실용화 관점에서 Google, MS Azure, Amazon AWS 에서도 연구 중인 기술

2.2 동형암호의 기술적 특징

DEVIEW 2021

동형암호란

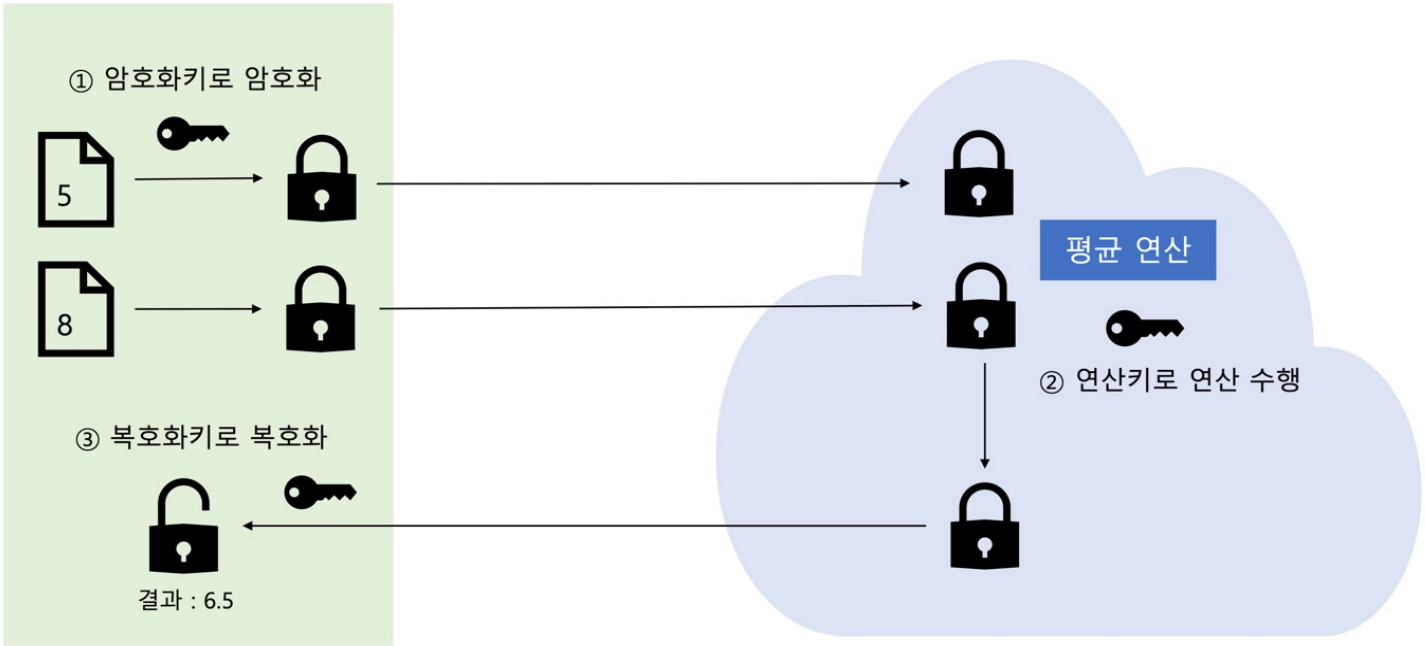
- 동형암호의 키는 암호화키, 복호화키, 연산키로 구성

	암호화키	연산키(계산키)	복호화키
종류	공개키	공개키	비밀키
기능	데이터 암호화 시 사용	암호문 간 연산 시 사용	암호문의 복호화 시 사용

2.2 암호화된 상태로 연산 가능한 동형암호

DEVIEW 2021

동형 암호 에서의 연산



2.2 동형암호의 기술적 특징

DEVIEW 2021

동형암호의 기술적 특징

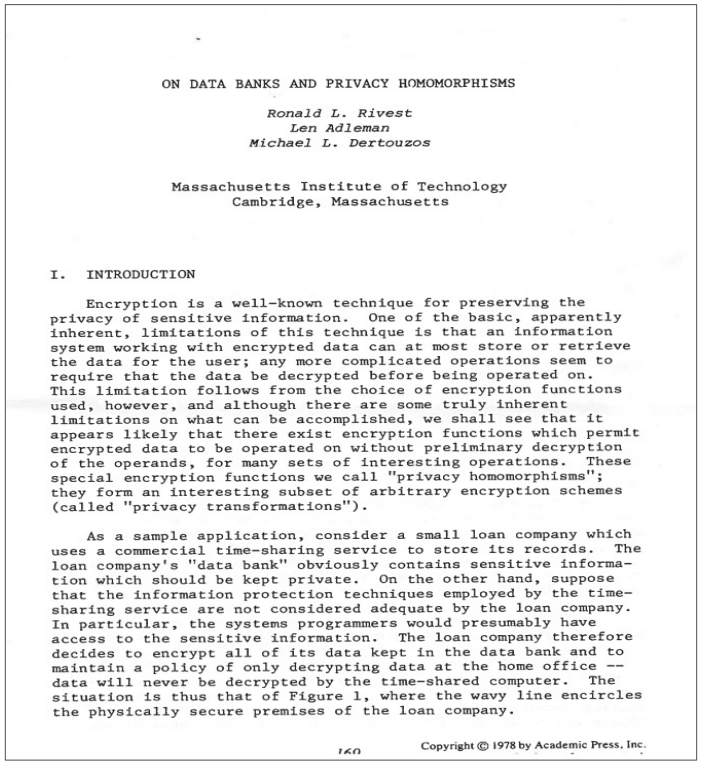
- depth : 동형암호에서 허용된 곱셈의 횟수. 동형암호에서 허용된 Depth 이상의 곱셈이 수행되면 복호화된 값을 신뢰할 수 없게 됨
- 완전 동형암호 : 일반적으로 곱셈이 반복되면 노이즈가 커지게 되어 일정 횟수 이상의 곱셈이 불가능한 유한 동형암호 알고리즘(somewhat)과 달리 곱셈의 횟수가 제한되지 않은 암호 알고리즘
- 재부팅(Bootstrapping) : 일반 동형암호 알고리즘이 완전 동형암호로 사용되기 위해 곱셈에서 증가하는 노이즈를 줄이는 과정

2.3 동형암호의 연구현황

DEVIEW 2021

동형암호의 시작

- 동형암호는 1978년 Rivest, Adleman, 그리고 Dertouzos 의 논문 [1] 에서 그 개념이 처음 제안된 후, 동형암호에 대한 다양한 연구들이 이어져 옴
- 하지만 곱셈 연산을 반복할수록 내부의 노이즈가 커져 어느 한계치 이상부터는 복호화 불가능



ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts

I. INTRODUCTION

Encryption is a well-known technique for preserving the privacy of sensitive information. One of the basic, apparently inherent, limitations of this technique is that an information system working with encrypted data can at most store or retrieve the data for the user; any more complicated operations seem to require that the data be decrypted before being operated on. This limitation follows from the choice of encryption functions used, however, and although there are some truly inherent limitations on what can be accomplished, we shall see that it appears likely that there exist encryption functions which permit encrypted data to be operated on without preliminary decryption of the operands, for many sets of interesting operations. These special encryption functions we call "privacy homomorphisms"; they form an interesting subset of arbitrary encryption schemes (called "privacy transformations").

As a sample application, consider a small loan company which uses a commercial time-sharing service to store its records. The loan company's "data bank" obviously contains sensitive information which should be kept private. On the other hand, suppose that the information protection techniques employed by the time-sharing service are not considered adequate by the loan company. In particular, the systems programmers would presumably have access to the sensitive information. The loan company therefore decides to encrypt all of its data kept in the data bank and to maintain a policy of only decrypting data at the home office -- data will never be decrypted by the time-shared computer. The situation is thus that of Figure 1, where the wavy line encircles the physically secure premises of the loan company.

160

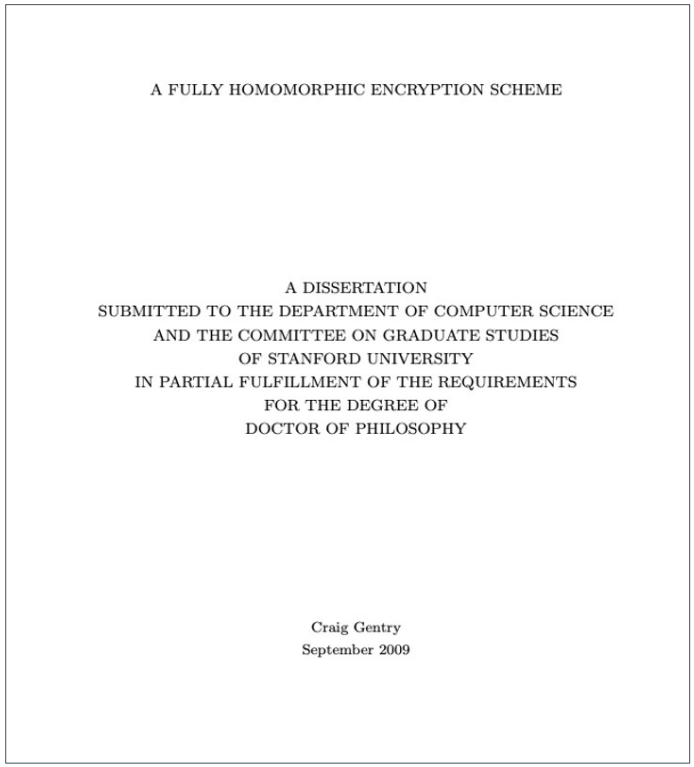
[1]http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.500.3989&rep=rep1&type=pdf

2.3 동형암호의 연구현황

DEVIEW 2021

완전동형암호의 탄생

- 2009년 Gentry의 논문[2]에서 재부팅(Bootstrapping)을 통해 무한한 횟수의 곱셈 연산이 가능한 완전동형암호(Fully homomorphic encryption)의 기술적 가능성 제시



A FULLY HOMOMORPHIC ENCRYPTION SCHEME

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Craig Gentry
September 2009

[2] https://crypto.stanford.edu/craig/craig-thesis.pdf

2.3 동형암호의 연구현황

동형암호의 최신 동향

- 2016년 Cheon 외 3인은 반올림 연산이 효율적으로 진행 가능한 동형암호 알고리즘 설계 (이하 CKKS)
- 현재 CKKS 기반 동형암호 알고리즘의 연구가 활발히 지속되고 있음

	시작년도	특징	연산	ISO 표준
1세대	2009년	최초 완전동형암호	Bit	
2세대	2011년	최초의 사용 가능한 동형 암호	정수	BGV, BFV
3세대	2013년	작은 데이터 처리에 효과적인 동형암호	Bit	TFHE
4세대	2016년	최초의 실수 연산 지원하는 동형암호	실수, 정수	CKKS

[출처] 크립토랩

2.4 동형암호 기반 프라이버시 보존 분석의 장점

DEVIEW 2021

기존 프라이버시 보존 분석과 비교한 장점

- 기존의 익명성 모델들과는 달리 동형암호는 평문 그대로 암호화 하여 연산하므로 데이터의 정확성이 보존됨
- 기존 암호화 방식과 달리 동형암호는 복호화가 필요 없이 암호화된 상태로 연산 가능
- 비식별화 처리를 따로 진행할 필요가 없으므로 데이터의 손실 없이 분석 가능

따라서 데이터를 비식별화없이 암호화 상태에서 연산하므로 데이터의 정보노출 방지

2.5 동형암호 기반 프라이버시 보존 분석의 제약점

DEVIEW 2021

동형암호 기반 분석 방법의 제약

- 동형암호의 암호문 사이즈는 평문 사이즈에 비해 수십 배 이상 커지고, 키 사이즈가 RSA등 기존 암호화 알고리즘과 비교하여 상당히 큼
- 평문의 연산 속도에 비해 동형암호 내의 연산 속도가 수십 배 이상 느리고, 특정 연산은 GPU 환경에서 효율적인 수행 가능

Depth 별 키 사이즈 및 암호 블럭 용량

Depth	키 사이즈(MB)	암호 블럭 용량(KB)
0	5.3	65.7
1	23.0	262.3
5	296.5	1573.1
11	1282.0	6291.6
full	14020.0	62914.8

[출처] 크립토랩

2.6 효율적인 동형암호 사용을 위한 환경

DEVIEW 2021

따라서 동형암호 기반 분석을 효율적으로 하기 위해서는..

큰 사이즈의 키와 암호문의 저장 및 권한 관리

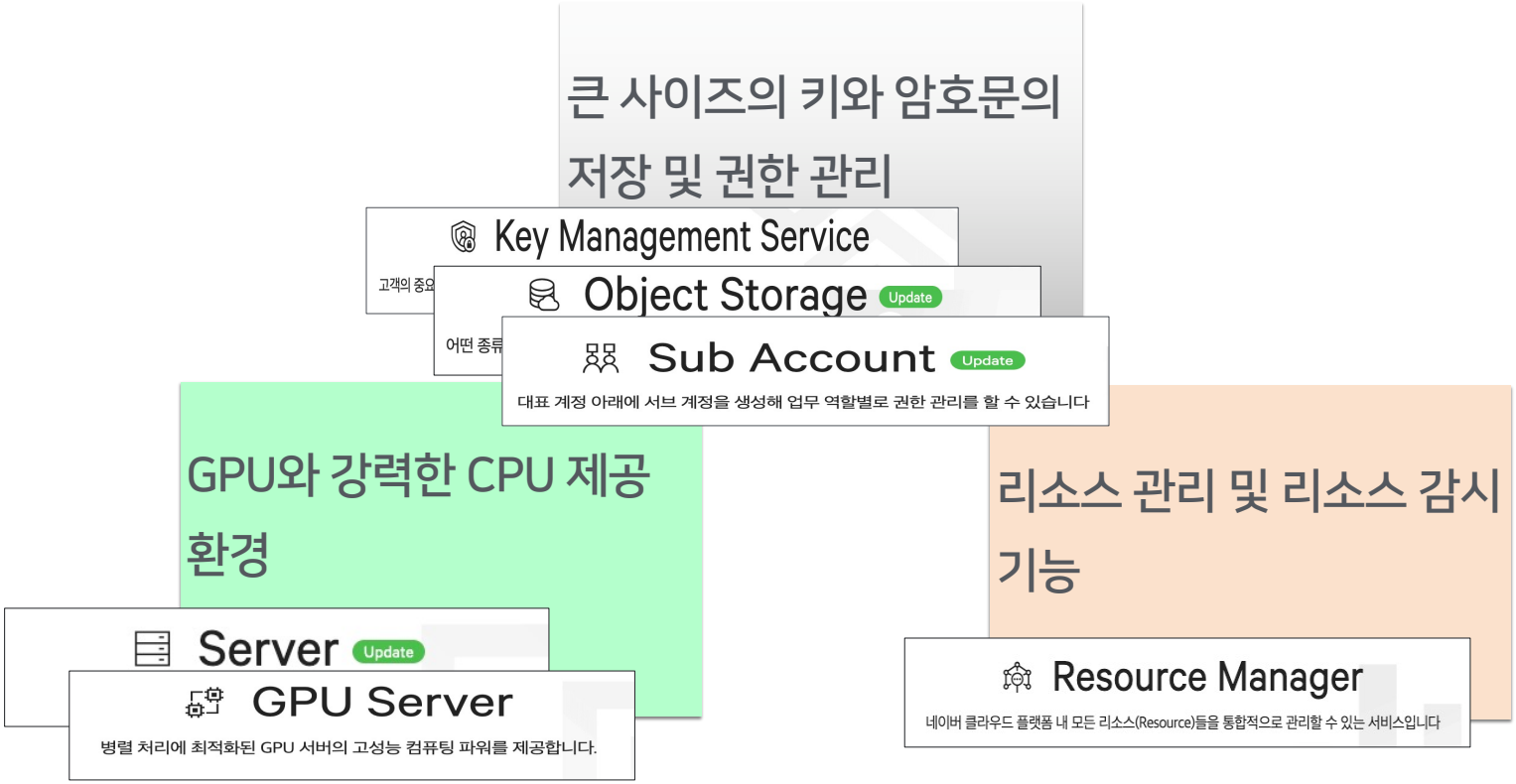
GPU와 강력한 CPU 제공 환경

리소스 관리 및 리소스 감시 기능

2.6 효율적인 동형암호 사용을 위한 환경

DEVIEW 2021

클라우드 환경이 적합



3. 동형암호 기반 클라우드 서비스

3.1 클라우드에서의 동형암호의 활용

클라우드에서의 동형암호의 활용

- 암호화키 및 연산키, 암호문 등을 저장하고 권한 부여에 따라 공유를 쉽게 할 수 있는 클라우드의 Storage 사용 시 보관 및 권한 관리가 유용함
- 주로 빅데이터를 다루는 동형암호의 특징상 CPU, GPU 등 다양한 컴퓨팅 환경이 제공되는 클라우드에서 동형암호 사용함이 유리함

3.1 클라우드에서의 동형암호의 활용

서버 리소스에 따른 연산 별 성능 차이

- 연산 시간을 살펴보면 GPU 서버에서 통계 연산은 CPU 서버에서의 통계 연산보다 빠르고, 키 생성 및 암/복호화 연산은 CPU 서버에서 더 빠르게 진행됨

	CPU/GPU Info	vCPU	Mem	KeyGen	Enc	Sum	Avg	Var	Corr
GPU	Intel Xeon Gold 5220 CPU @ 2.20GHz / GV 100GL [Tesla V100 PCIe 32GB]	16	32GB(GPU)	837초	1.9	0.63	0.1	0.099	0.053
CPU	Intel Xeon Gold 5220 CPU @ 2.20 GHz	64	64GB	229초	1	1	1	1	1

[표] CPU 서버와 GPU 서버와의 성능비교표, 키 생성 시간의 단위는 초, 나머지 연산은 CPU서버 기준 연산 시간 비율을 나타냄

- 따라서, 동형암호 기술을 효율적으로 사용하기 위해서는 CPU 및 GPU 리소스를 연산의 종류에 따라 사용하는 것이 효율적 => 클라우드 환경에서 효율성이 증대됨

3.2 HEaaN Homomorphic Analytics

DEVIEW 2021

HEaaN Homomorphic Analytics 개요

세계 최고수준의 동형암호 기술(HEaaN) 적용

클라우드 환경에서도 신용정보, 금융거래정보 등 민감한 데이터의 노출없이 통계 분석과 기계 학습 가능

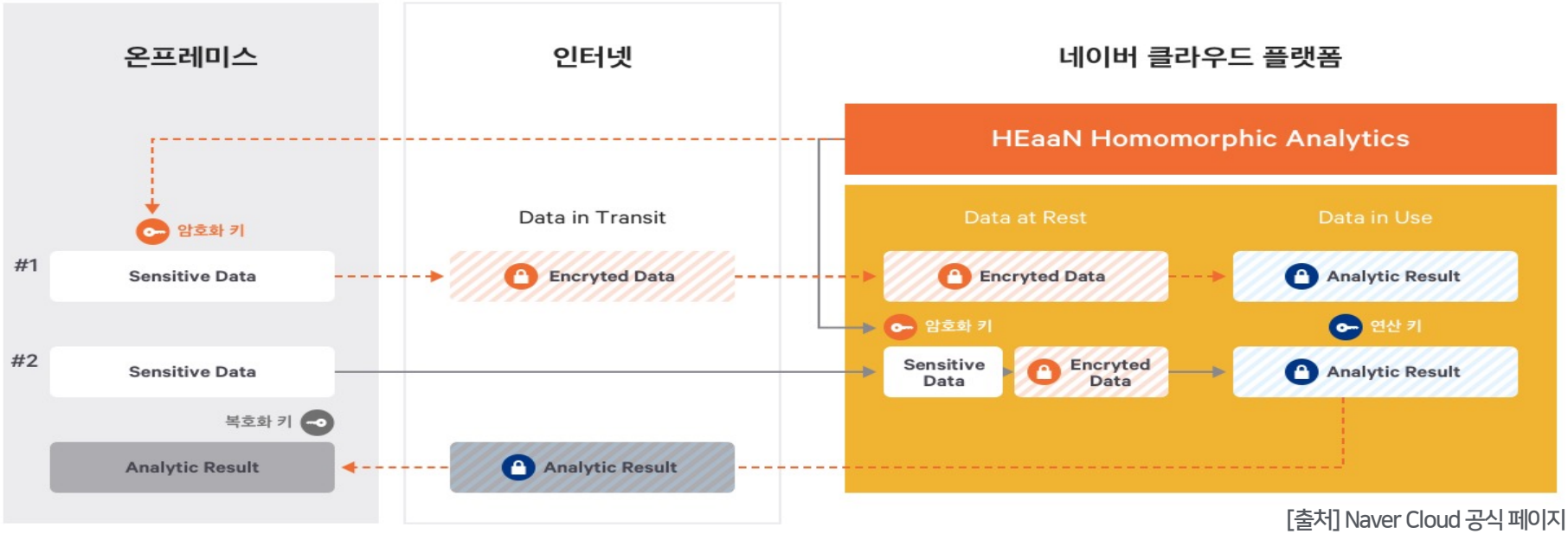
금융, 공공 기관, 기업간 데이터 교환 및 결합 시 비식별화없이 데이터 분석을 통한 분석의 품질을 유지

차세대 암호기술을 누구나 쉽게 사용할 수 있도록 툴킷들과 고성능의 클라우드 인프라를 제공

3.2 HEaaN Homomorphic Analytics

DEVIEW 2021

HEaaN Homomorpihc Analytics의 구조



[출처] Naver Cloud 공식 페이지

3.3 HEaaN Homomorphic Analytics의 기능

DEVIEW 2021

암호화 상태 유지를 통한 데이터 유출의 방지

- 데이터의 전송 및 저장, 분석 과정에서 암호화 상태 유지를 통해 데이터 유출 방지
- 키 생성, 암복호화 및 다양한 연산 기능을 제공

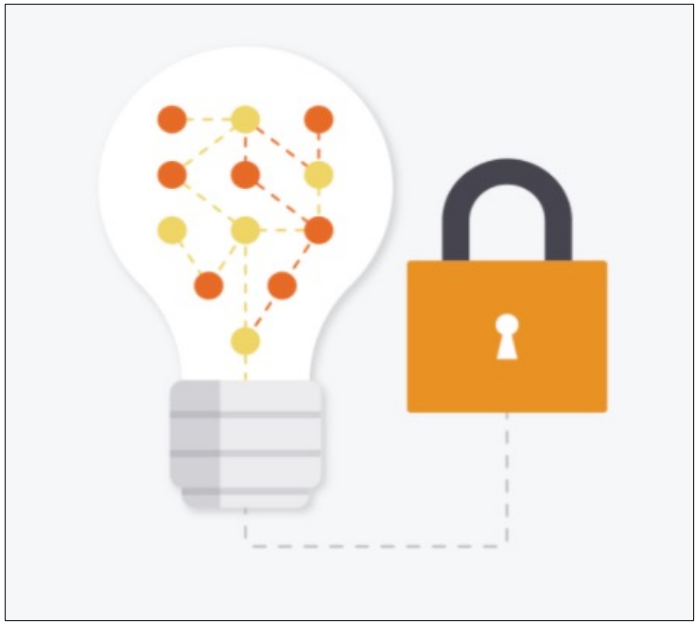
[출처] Naver Cloud 공식 페이지

3.3 HEaaN Homomorphic Analytics의 기능

DEVIEW 2021

통계 분석 및 머신러닝을 통한 인사이트 제공

- 평균, 분산, 표준오차 등 통계 분석에서 필수적인 연산들을 암호화된 상태에서 분석 가능
- 로지스틱 회귀(Logistic Regression) 기반의 머신러닝을 위해 데이터 정규화 및 학습과 추론 가능
- 학습된 모델 평가를 통해 더욱 정교한 로지스틱 회귀 모델 개발 가능



[출처] Naver Cloud 공식 페이지

3.3 HEaaN Homomorphic Analytics의 기능

DEVIEW 2021

파이혜안(pi-HEaaN)

- Python 스크립트를 통해 혜안이 제공하는 동형암호 연산 기능을 구현하고, 암호화되지 않은 평문 데이터로 실행 결과 확인가능
- 작성한 스크립트를 업로드하면 암호화된 데이터로 클라우드 환경에서 연산 수행 가능

```python
def poly(evaluator, mult_key, ciphertext):
    poly_coeffs = [1, 2, 3, 4, 5]
    ciphertext_tmp0, ciphertext_tmp1, ciphertext_tmp2 = heaan.Ciphertext(), heaan.Ciphertext(), heaan.Ciphertext()
    ciphertext_result = heaan.Ciphertext()

    for index, value in enumerate(coeff):
        if index == 0:
            coeff_zero = value
        elif index == 1:
            evaluator.mult(ciphertext, value, ciphertext_tmp1)
        else:
            evaluator.mult(ciphertext, ciphertext_tmp0, mult_key, ciphertext_tmp0)
            evaluator.mult(ciphertext_tmp0, value, ciphertext_tmp2)
            evaluator.add(ciphertext_tmp1, ciphertext_tmp2, ciphertext_tmp1)
    evaluator.add(ciphertext_tmp1, coeff_zero, ciphertext_result)
    return ciphertext_result

poly_result = poly(evalutator, mult_key, ciphertext)
```

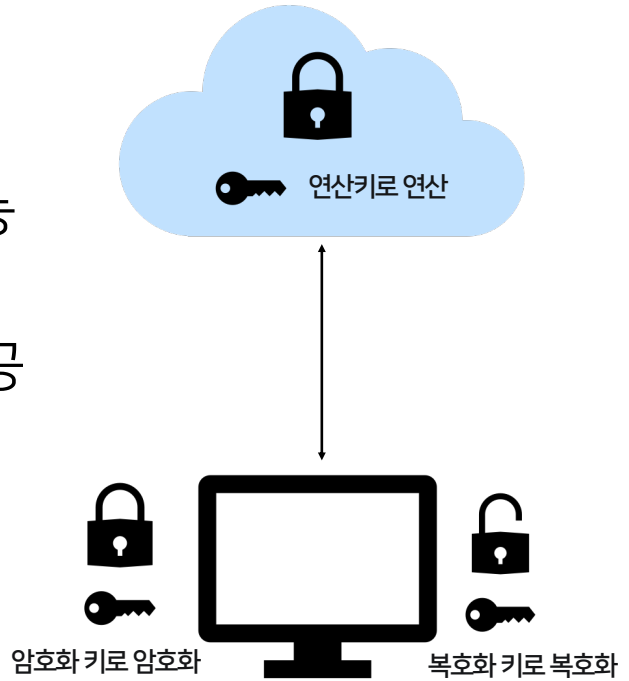
pi-HEaaN 스크립트 예시

3.3 HEaaN Homomorphic Analytics의 기능

DEVIEW 2021

데이터 소유권을 지킬 수 있는 에이전트 제공

- 사용자 환경에서 실행할 수 있는 에이전트를 통해 동형암호 키 생성 및 데이터를 암호화 후 업로드 기능 제공
- 클라우드에서 암호화된 데이터 연산의 수행기능 제공
- 암호화된 결과는 로컬로 다운로드하여 복호화 가능

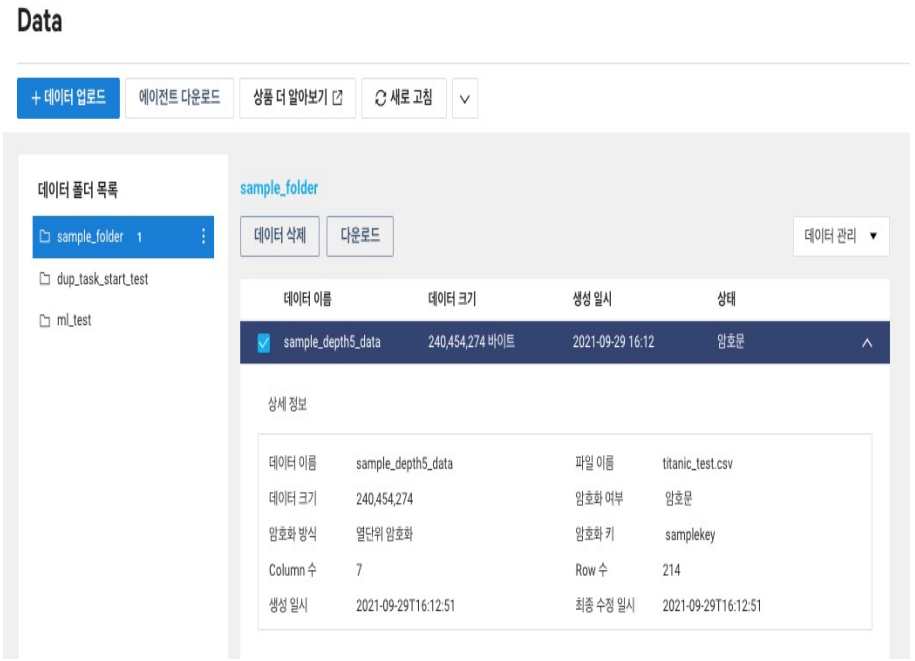


Agent의 작동 구조

3.3 HEaaN Homomorphic Analytics의 기능

DEVIEW 2021

콘솔에서 키 생성 및 데이터 암호화 예시



3.4 HEaaN 라이브러리 소개

DEVIEW 2021

CKKS 알고리즘의 연구 동향

- 2016년 Cheon 외 3인은 반올림 연산이 효율적으로 진행 가능한 동형암호 알고리즘을 설계 (이하 CKKS)
- 2017년 재부팅을 통해 CKKS를 완전동형화 가능성 증명 후
- 2018년 동형암호화된 상태에서 Logistic Regession medel의 학습 및 추론이 가능함을 증명
- 2020년 기계학습 방법 중 하나인 SVM(Support Vector Machine)에서 동형암호를 통해 개인정보를 보호하는 방법에 대한 연구
- 2021년 GPU 환경에서 부트스트래핑 성능의 최적화에 관련된 연구

3.4 HEaaN 라이브러리 소개

HEaaN (Homomorphic Encryption for Arithmetic of Approximate Numbers)

- CKKS 기반 기술인 RLWE(Ring Learing with errors)는 소인수분해 기반의 RSA와 달리 양자 컴퓨팅 능력을 가진 공격에도 안전하다고 알려져 있음
- HEaaN은 CKKS 알고리즘을 소개한 논문의 제목이자 이를 구현한 라이브러리
- HEaaN.STAT은 HEaaN 제품군 중 기초 통계 기능 및 암호문 간의 정렬, 로지스틱 회귀 기능을 제공

3.5 시연

DEVIEW 2021

시연

4. 활용 사례

4.1 해외 사례

DEVIEW 2021

Google Transpiler

- 개발자가 간단한 문자열 처리나 산술 연산 등 기본 연산을 위한 코드를 작성 후 동형 암호화된 데이터에서 연산 가능한 코드로 변환
- 완전 동형암호 오픈소스 라이브러리인 TFHE 를 활용

```
int sum(int a, int b) {
  return a + b;
}
```

→

```
#include <tfhe.h>

// Full adder
void sum (LweSample* result,
    const LweSample* a,
    const LweSample* b,
    const int nb_bits,
    const TfheKeySet* bk) {
  LweSample* carry = new_ciphertext(bk->params);
  LweSample* temp = new_ciphertext(bk->params);

  // Initialize the carry to 0
  bootsCONSTANT(&carry, 0, bk);

  // Compute bit wise addition
  for (int i = 0; i < nb_bits; i++) {
    // Compute sum
    bootsXOR(&temp, &a[i], &b[i], bk);
    bootsXOR(&result[i], &temp, &carry, bk);

    // Compute carry
    bootsAND(&carry, &carry, &temp, bk);
    bootsAND(&temp, &a[i], &b[i], bk);
    bootsOR(&carry, &temp, &carry, bk);
  }

  delete_ciphertext(carry);
  delete_ciphertext(temp);
}
```

[출처] https://storage.googleapis.com/pub-tools-public-publication-data/pdf/bec910c4455e5f0464ce112678424dac729a4448.pdf

4.1 해외 사례

DEVIEW 2021

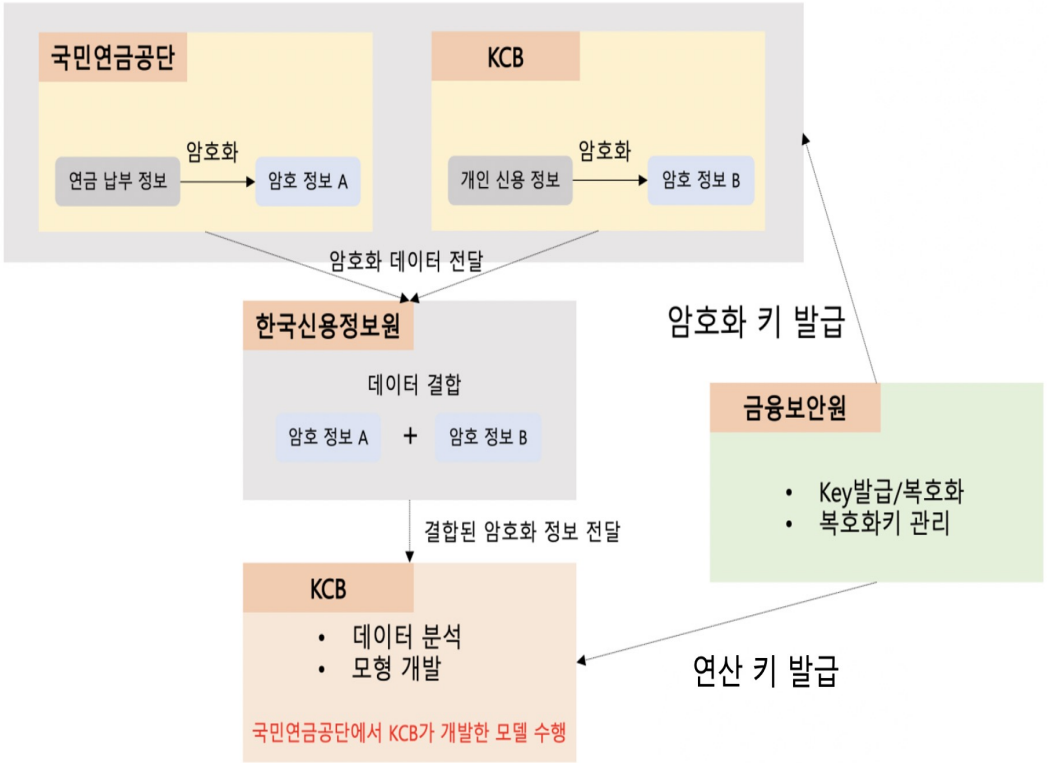
Microsoft Edge 사용자에게 Password Monitor 기능 제공

- Edge브라우저에 저장 된 암호의 노출 여부를 제공
- 준동형 암호화 기법을 사용하여 사용자의 패스워드를 노출하지 않은 채 해당 패스워드가 Breached Credential DB에 저장된 값인지 확인 후 해당 결과를 사용자에 리턴
- 사용자는 해당 결과를 통해 자신의 계정 정보의 유출 여부를 알 수 있음

4.2 국내 사례

KCB 신용데이터 분석 사례

- 동형암호 전문 기업인 크립토랩의 HEaaN의 기술 적용
- 동형암호 기술을 적용하여 약 234만명의 국민연금 데이터와 KCB의 신용데이터를 결합 후 분석



4.2 국내 사례

DEVIEW 2021

드론 제어시스템에 동형암호 기술 적용

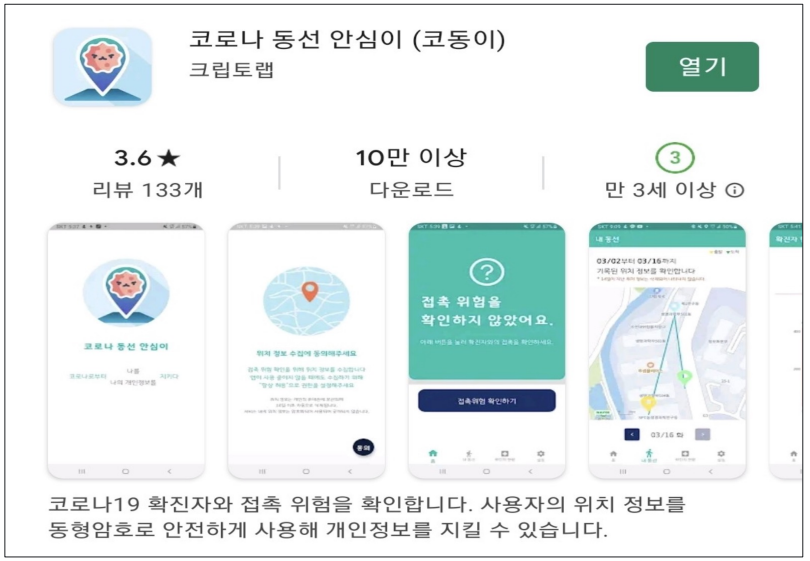
- 동형암호 전문 기업인 크립토랩의 HEaaN의 기술 적용
- 심형보 서울대 전기·정보공학부 교수팀은 천정희 서울대 수리과학부 교수팀과 공동으로 2016년 9월 국제자동제어연합(IFAC)이 개최한 학술대회에서 동형암호로 보호한 상태에서도 드론을 제어할 수 있다는 시뮬레이션 결과 발표

[출처] https://www.dongascience.com/news.php?idx=25207

4.2 국내 사례

코동이(코로나 동선 안심이)

- 동형암호 전문 기업인 크립토랩의 HEaaN의 기술 적용
- 동형암호를 활용하여 사용자의 위치 정보를 암호화한 후 코로나 확진자 동선과 겹쳤는지 확인 가능한 서비스
- 개인정보를 암호화 했기 때문에, 서버 등에 개인정보를 직접적으로 노출하지 않고도 확진자와의 접촉 여부 확인 가능

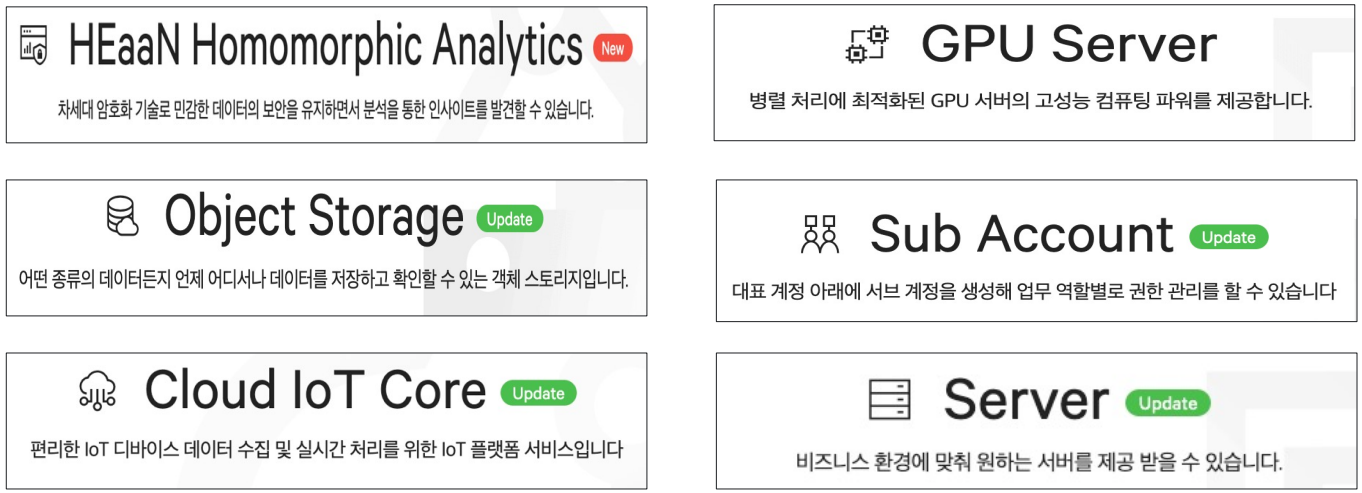


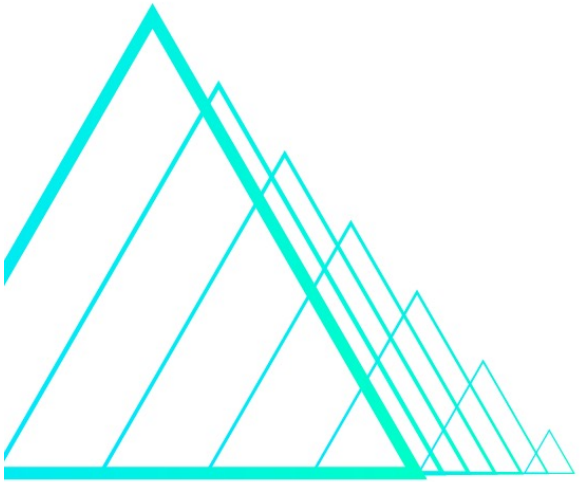
[출처] 구글플레이스토어

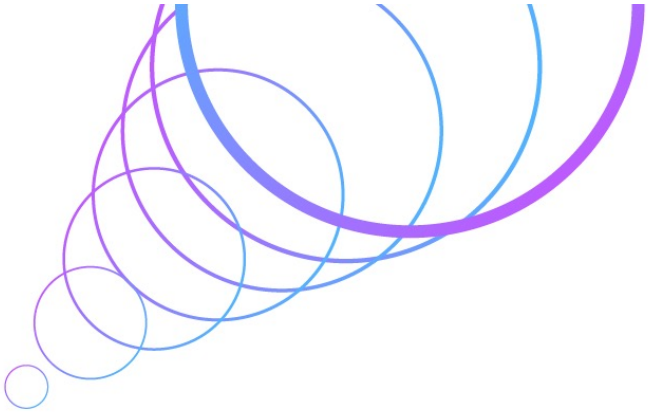
4.3 NCP 서비스를 통한 동형암호 서비스 연계

DEVIEW 2021

Naver Cloud에서 동형암호 서비스의 활용







Thank You