Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.1, Feb. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.1.141

로컬 차분 프라이버시 실제 적용 사례연구 : 프라이버시 보존형 설문조사*

정 수 용,[1†] 홍 도 원,[2‡] 서 창 호[2]
[1,2]공주대학교 (대학원생, 교수)

Case Study on Local Differential Privacy in Practice : Privacy Preserving Survey*

Sooyong Jeong,[1†] Dowon Hong,[2‡] Changho Seo[2]
[1,2]Kongju National University (Graduate student, Professor)

요 약

차분 프라이버시는 데이터 프라이버시를 보존함과 동시에 데이터를 수집 및 분석할 수 있는 기법으로써 프라이버시 보존형 데이터 활용 분야에서 널리 적용되고 있다. 이러한 차분 프라이버시의 지역적 모델인 로컬 차분 프라이버시 알고리즘은 무작위 응답을 기반으로 데이터 소유자가 직접 데이터를 가공 처리하여 공개한다. 따라서 개인은 데이터 프라이버시를 보장받을 수 있으며, 데이터 분석가는 수집된 다수의 데이터를 통해 유용한 통계적 결과값을 도출할 수 있다. 이러한 로컬 차분 프라이버시 기법은 세계적 기업인 Google, Apple, Microsoft에서 실질적으로 사용자의 데이터를 수집 및 분석할 때 활용되고 있다. 본 논문에서는 현실에 실질적으로 활용되고 있는 로컬 차분 프라이버시 기법에 대해 비교분석한다. 또한, 실제 적용 사례 연구로써 개인의 프라이버시가 결과의 신뢰성에 큰 영향을 미치는 설문 및 여론조사 시나리오를 기반으로 로컬 차분 프라이버시 기법을 적용하여 현실에서의 활용 가능성에 대해 연구한다.

ABSTRACT

Differential privacy, which used to collect and analysis data and preserve data privacy, has been applied widely in data privacy preserving data application. Local differential privacy algorithm which is the local model of differential privacy is used to user who add noise to his data himself with randomized response by self and release his own data. So, user can be preserved his data privacy and data analyst can make a statistical useful data by collected many data. Local differential privacy method has been used by global companies which are Google, Apple and Microsoft to collect and analyze data from users. In this paper, we compare and analyze the local differential privacy methods which used in practically. And then, we study applicability that applying the local differential privacy method in survey or opinion poll scenario in practically.

Keywords: Local Differential Privacy, Data Privacy, Privacy Preserving Survey, Randomized Response, Data Analysis

Received(11. 12. 2019), Modified(01. 16. 2020), Accepted(01. 16. 2020)

* 이 논문은 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2019R1A2C1003146)

* 본 논문은 2019년도 충청지부 학술대회에 발표한 우수논문을 개선 및 확장한 것임

† 주저자, jsy8630@smail.kongju.ac.kr

‡ 교신저자, dwhong@kongju.ac.kr(Corresponding author)

I. 서 론

네트워크의 발전과 클라우드 서비스의 확산으로 발생하는 데이터의 양이 크게 증가하고, 컴퓨터 연산 능력의 향상으로 데이터에 대한 분석 및 활용이 활발해지면서 데이터를 통한 새로운 가치창출이 가능해졌다. 이에 따라 데이터를 수집, 분석 및 활용하는 방안에 대한 많은 연구가 진행되고 있다. 그러나 많은 양의 데이터가 공개 및 분석되면서 프라이버시 침해 사례가 증가하고 있으며, 프라이버시를 보존하는 데이터 분석 기술이 요구되고 있다.

차분 프라이버시는 이론적으로 엄격한 개념으로써 데이터 프라이버시를 보존함과 동시에 데이터 분석을 통해 유용한 정보를 얻어낼 수 있다[1,2]. 차분 프라이버시는 데이터베이스에서 개인의 존재 유무와 상관없이 유사한 결과값을 도출하여 개인의 프라이버시를 보장하고 데이터의 활용을 가능하게 한다.

이러한 차분 프라이버시의 지역적 모델인 로컬 차분 프라이버시는 데이터 소유자가 직접 노이즈를 추가하여 제공하는 메커니즘을 수행한다[3]. 데이터 소유자는 무작위 응답(Randomized response)[4]을 기반으로 가공 처리된 데이터를 제공함으로써 프라이버시를 보호할 수 있고, 제공된 데이터를 통해 데이터 분석가는 원하는 통계적 결과값을 계산할 수 있다. 로컬 차분 프라이버시 기법에 대한 연구는 계속 진행되고 있으며[5-17] 특히, 세계적인 IT기업인 Apple, Microsoft(MS), Google은 로컬 차분 프라이버시 기법을 도입하여 실제 사용자들의 프라이버시를 보존하고 데이터를 수집 및 분석하고 있다.

Google은 2014년에 처음으로 Randomized Aggregatable Privacy-Preserving Ordinal Response(RAPPOR)[15]를 공개하면서, 실제 크롬 웹 브라우저에서 로컬 차분 프라이버시 기법을 적용하고 있음을 밝혔다. 사용자의 인터넷 접속 기록(URL)을 수집하여 악의적인 홈페이지를 탐색하고 차단하기 위해 활용하고 있다. 또한, RAPPOR는 데이터의 지속적인 활용을 위해 저장 후 재사용하는 memoization기법을 사용하고 있으며 이를 통해, 계산량적 비용을 감소시킨다. 하지만 사용자의 데이터가 조금씩 계속 바뀌는 경우 계산량적 장점을 가진 기법이 무의미해지고, 파라미터 간의 관계를 확인할 수 있는 유용성에 대한 이론적 분석에 한계점이 존재한다. 이에 Apple은 유용성에 대한 이론적 한계점을 극복하고, MS는 기존의 계산량적 장점을 살릴 수 있는 새로운 메커니즘을 제안하였다[16,17].

Apple은 원본 데이터의 기댓값을 보존하는 Count Mean Sketch(CMS) 알고리즘과 사용자의 전송 비용을 감소시켜 오직 1bit 전송으로 유용한 통계적 결과값을 계산할 수 있는 Hadamard Count Mean Sketch(HCMS) 알고리즘을 공개하였다[16]. 두 개의 알고리즘은 사용자의 스마트 디바이스 및 사파리 웹 브라우저에서 사용자의 데이터를 수집하여 사용자 단말에서의 단어 자동 완성 기능, 웹 브라우저에서의 동영상 자동 재생 등 다양한 서비스의 품질 향상을 위해 사용되고 있다.

또한, MS는 통신량적 비용을 1bit로 감소시킨 1-BitMean과 범주형 자료에 사용할 수 있는 d-BitFlip[17]을 소개하면서, 해당 메커니즘을 윈도우 운영체제의 업데이트에 포함시키고 사용자의 애플리케이션 사용 데이터를 수집 및 분석할 때 로컬 차분 프라이버시 기법을 활용하고 있음을 밝혔다.

이처럼 실제 환경에서 로컬 차분 프라이버시 기법을 적용하는 연구가 진행되고 있으며, 본 논문에서는 현재 실질적으로 적용되고 있는 기법을 비교 분석하여 새로운 환경에서의 적용 가능성에 대해 연구한다. 특히, 개인의 데이터 프라이버시를 보존하면서 유용한 통계적 결과를 도출할 수 있는 설문조사에 적용하여 현실 문제에 대한 활용 가능성을 분석한다.

최근 설문조사는 온라인상에서 다양한 디바이스를 통해 수행되고 있으며, 이때 설문조사 결과는 참여자의 민감한 정보를 포함할 수 있으므로 프라이버시 침해가 발생할 수 있다. 또한, 설문조사에서 참여자들은 본인의 응답이 노출될 가능성을 우려하여 응답을 피하거나, 꼭 참여해야 한다면 참여자 본인의 생각과 다르게 사회적으로 바람직한 방향 혹은 조사자(또는 조사기관)의 기대에 부응하는 응답을 하려는 경향이 있다. 익명성이 설문조사 결과에 미치는 영향에 대한 연구[18]에서 실시한 대학생들 대상의 설문조사는 학생들의 학번 및 주민등록번호 앞자리를 함께 제출함으로써 익명성이 보장되지 않는 환경에서 진행되었다. 이때, 학교 만족도 조사에서 약 70% 이상의 학생이 만족하는 결과가 나왔지만, 이전의 익명으로 진행된 설문조사의 경우 약 47%의 학생만이 만족하는 결과가 나타났다.

이처럼, 응답자의 익명성이 훼손될 경우 조사결과의 신뢰성이 심각하게 감소할 수 있다[18-22]. 이에 따라, 다양한 설문 및 여론조사 대행업체는 익명성을 강조하며 설문조사를 진행하고 있음을 밝혔지만, 이

는 참여자가 대행업체를 전적으로 믿는 경우에만 신뢰할 수 있는 결과가 나올 수 있다. 따라서 본 논문에서는 설문조사에서 참여자의 민감 정보 노출에 대한 우려를 없앰과 동시에 데이터 활용을 가능하게하기 위해 로컬 차분 프라이버시 기법을 적용한 프라이버시 보존형 설문 및 여론조사에 대해 연구한다. 또한, 원본 데이터와 로컬 차분프라이버시 기법을 적용한 데이터를 비교하여 현실에서의 적용 가능성을 분석한다.

본 논문의 2장에서는 차분 프라이버시 및 로컬 차분 프라이버시에 대해 설명하고, 3장에서는 로컬 차분 프라이버시 적용 모델을 비교분석한다. 4장에서는 이를 바탕으로 로컬 차분 프라이버시 기법을 적용한 프라이버시 보존형 설문조사의 결과를 분석한 후에 5장에서 결론을 내린다.

II. 연구배경

이번 장에서는 연구의 배경이 되는 차분 프라이버시와 로컬 차분 프라이버시에 대해 설명한다.

2.1 차분 프라이버시

차분 프라이버시는 2006년 Dwork[1]에 의해 처음 제안된 매우 엄격한 이론적 개념으로 데이터 활용 분야에서 널리 사용되고 있다. 특히, k-익명성, l-다양성, t-근접성 등과 같은 이전의 많은 프라이버시 보존 개념과는 다르게 한 개인이 데이터베이스에 있고 없음에 상관없이 유사한 결과값을 도출함으로써, 개인의 프라이버시를 보장하고 유용한 데이터 활용을 가능하게 한다. 차분 프라이버시는 다음과 같이 정의한다[1,2].

Definition 1. (ϵ - 차분 프라이버시).

입력 공간 $N^{|\chi|}$ 와 출력 공간 R을 갖는 임의의 메커니즘 $M: N^{|\chi|} \rightarrow R$ 은 $\| x-y \|_1 \leq 1$ 을 만족하는 인접한 두 개의 부분집합 $x, y \in N^{|\chi|}$ 와 출력 $S \subseteq R$에 대하여 다음 식을 만족할 때 ϵ-차분 프라이버시를 만족한다고 한다.

$$\Pr[M(x) \in S] \leq e^{\epsilon} \Pr[M(y) \in S]$$

이때, $\| x-y \|_1 \leq 1$ 는 두 개의 부분집합 x와 y의 원소가 한 개만 다른 것을 의미하며, $\epsilon > 0$이고 프라이버시 파라미터 ϵ이 작을수록 엄격한 프라이버시가 보장된다.

차분 프라이버시를 만족하는 메커니즘의 입력값은 한 개의 원소만 차이가 나는 인접한 두 개의 부분집합으로 정의가 되어 있다. 이때, 한 개의 원소를 한 개인으로 생각한다면 특정 데이터베이스에서 계산된 결과값을 통해서 해당 데이터베이스에 어떠한 개인의 존재 유무를 판단할 수 없음을 의미한다.

2.2 로컬 차분 프라이버시

차분 프라이버시의 지역적 모델인 로컬 차분 프라이버시[3]는 차분 프라이버시와 다르게 데이터 소유자가 누구도 믿지 않는 상황에서 데이터를 공개할 수 있다. 이때 데이터 소유자는 데이터에 무작위 응답[4]을 기반으로한 노이즈를 추가하여 공개함으로써, 개인의 프라이버시를 보존할 수 있다. 무작위 응답은 다음의 동전 던지기 예시로 쉽게 설명할 수 있다.

응답자는 특정 질의에 대해 응답을 할 때, 동전을 던져서 뒷면이 나오면 솔직한 대답을 하고 앞면이 나올 경우에는 다른 동전을 다시 한 번 던진다. 두 번째 던진 동전이 앞면인 경우에는 무조건 '예'라고 대답하고, 뒷면인 경우에는 무조건 '아니요'라고 대답을 한다. 결론적으로 응답자의 대답은 1/4의 확률로 '예', 1/4의 확률로 '아니오', 그리고 1/2의 확률로 솔직한 대답을 하는 것이다. 로컬 차분 프라이버시 메커니즘에서는 이를 확장하여, 동전 던지기에서 앞면과 뒷면이 나올 확률을 동일한 1/2이 아닌 임의의 확률을 적용하여 사용한다.

이러한 무작위 응답 기반의 노이즈를 사용하는 로컬 차분 프라이버시는 다음과 같이 정의한다[3].

Definition 2. (ϵ - 로컬 차분 프라이버시).

알고리즘 π가 임의의 입력값 v, v'과 모든 출력값 $Range(\pi)$에 대해 다음을 만족할 때, ϵ-로컬 차분 프라이버시를 만족한다고 한다.

$$\forall y \in Range(\pi): \Pr[\pi(v)=y] \leq e^{\epsilon} \Pr[\pi(v')=y] \quad (\epsilon > 0)$$

로컬 차분 프라이버시의 정의에 따르면 위의 동전 던지기 예시는 $ln3$-로컬 차분 프라이버시를 만족한다[3].

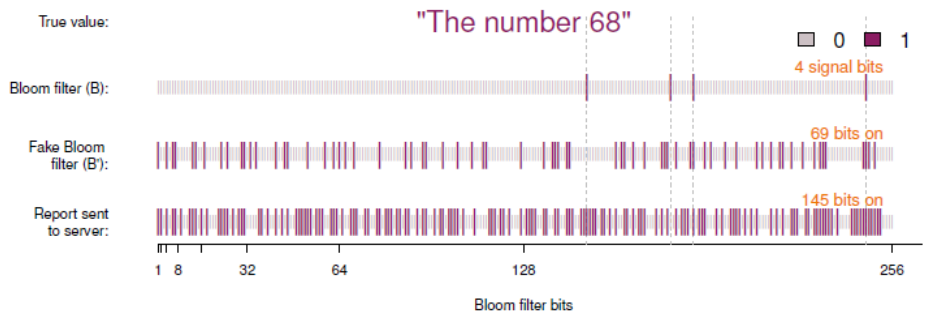

Fig. 1. Life of RAPPOR report[15]

III. 로컬 차분 프라이버시 적용 사례

로컬 차분 프라이버시에 대한 연구는 꾸준히 진행되고 있다[5-14]. 특히, 평균값 예측(Mean estimation)과 분포 예측(Histogram estimation)은 로컬 차분 프라이버시 적용 메커니즘의 주된 목표로써 많은 알고리즘이 제안되고 있다[10-14]. 세계적인 IT 기업인 Google, Apple, 그리고 MS는 로컬 차분 프라이버시를 만족하는 새로운 메커니즘을 공개함과 동시에 실제 사용자들의 데이터를 수집 및 분석하기 위해 사용하고 있음을 밝혔다. 이번 장에서는 이처럼 실제 사용자들의 데이터 수집 및 분석에 사용되고 있는 메커니즘을 분석한다.

3.1 Google[15]

Google은 실제 Chrome 웹 브라우저에서 사용자들의 인터넷 접속 기록(URL)을 수집하여 악의적인 홈페이지를 탐색하여 차단하기 위해 로컬 차분 프라이버시를 만족하는 메커니즘인 RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response)를 사용하고 있음을 공개하였다.

RAPPOR는 Bloom filter와 해시 함수(Hash function)를 사용하여 사용자의 데이터를 인코딩하고, 해당 Bloom filter에 2번의 무작위 응답을 사용하여 노이즈를 추가한다.

Fig 1은 Bloom filter에 적용되어 제공되는 RAPPOR 메커니즘의 출력값을 나타낸다. Fig 1의 첫 번째 'Bloom filter(B)'는 실제 사용자의 값이 '68'일 때, 총 4개의 해시 함수를 사용하여 Bloom filter에 나타낸 것이다. B에 첫 번째 무작위 응답을 적용하여 생성한 것이 B'이며, B'에 두 번째 무작위 응답을 적용하여 서버에 전송하는 데이터가 마지막의 'Report sent to server'이다.

Fig 2는 RAPPOR의 자세한 메커니즘을 설명한다. 먼저 사용자는 값 v를 h개의 해시 함수를 사용하여 $k$bits의 Bloom filter B에 업로드 한다. 그 후에 Bloom filter B의 각각의 비트에 대해 무작위 질의를 실행하는데, 이때 해당 확률 f는 사용자가 설정한다. 만들어진 B'은 저장되어 동일한 값 v를 사용할 때, 새롭게 계산하지 않고 저장되어있는 값을 재사용하는 memoization기법을 적용한다. 그리고 저장된 B'값에 새로운 무작위 질의를 실행하여 생성한 S를 서버에 전송한다..

서버는 사용하는 해시 함수의 충돌을 감소시키기

RAPPOR mechanisms

1. **Signal.** Hash client's value v onto the Bloom filter B of size k using h hash functions.

2. **Permanent randomized response.**
For each client's value v and bit i, $0 \le i < k$ in B, create a binary reporting value B'_i which equals to

$$B'_i = \begin{cases} 1, & with\ probabiltiy\ \frac{1}{2}f \\ 0, & with\ probability\ \frac{1}{2}f \\ B_i, & with\ probability\ 1-f \end{cases}$$

where f is a user-tunable parameter controlling the level of longitudinal privacy guarantee.
Subsequently, this B' is memoized and reused as the basis for all future reports on this distinct value v.

3. **Instantaneous randomized response.**
Allocate a bit array S of size k and initialize to 0. Set each bit i in S with probabilities

$$P(S_i = 1) = \begin{cases} q, & if\ B'_i = 1. \\ p, & if\ B'_i = 0. \end{cases}$$

4. **Report.** Send the generated report S to the server.

Fig. 2. RAPPOR[15]

위해 사용자들을 총 m개의 집단으로 나누고, 각각의 집단은 서로 다른 h개 해시 함수를 사용한다. 사용자들은 서버에 데이터를 전송할 때 서버가 그룹을 분류할 수 있도록 각자의 그룹 번호를 함께 보낸다. 서버는 각 그룹별로 $N_j(1 \le j \le m)$개, 총 N개의 데이터를 수집한다. 이를 통해, j번째 그룹에서 생성된 Bloom filter의 i번째 비트에 대한 총 개수 t_{ij}를 다음과 같이 예측할 수 있다.

$$t_{ij} = \frac{c_{ij} - (p + \frac{1}{2}fq - \frac{1}{2}fp)N_j}{(1-f)(q-p)} \quad (1)$$

여기서 f,p,q는 무작위 응답에서 사용되는 확률을 의미하고, c_{ij}는 서버가 사용자로부터 받은 데이터에서 j번째 그룹의 i번째 비트 개수를 의미한다.

서버는 계산된 t_{ij}로 이루어진 행렬 Y를 만들고, 실제 데이터에 대해 사전에 알려진 후보군을 통해 $km \times M$ 크기의 행렬 X를 생성한다. 사용자의 데이터 v는 전체 M개의 후보군 중 하나의 값을 나타내며, 행렬 X는 모든 후보군에 대해 모든 그룹에서 사용하는 해시 함수를 적용한 k비트의 Bloom filter를 모아둔 것이다. 다시 말해 Fig 1에서의 주어진 입력에 대한 Bloom filter B를 하나의 후보군에 대해 m개씩, 총 $m \times M$개의 Bloom filter를 나타낸 것이 행렬 X이다. 이렇게 만들어진 행렬 Y와 X에 선형회귀의 하나인 라쏘 회귀(Lasso regression)[23]를 사용하여 실제 사용된 데이터 후보군을 선정하고, 선정된 데이터 후보군에 정규 최소제곱 회귀(regular least-squares regression)를 사용하여 데이터의 실제 분포를 예측한다.

Fig 3은 RAPPOR를 사용하여 기존의 분포를 예측한 결과를 나타내는 그래프이다. k=32, h=2, m=64, p=0.25, q=0.75, f=0.5를 사용하여 수행한 결과로써, 프라이버시 파라미터 $\epsilon \fallingdotseq 2$를 만족한다. 왼쪽 그래프부터 각각 10000개, 10만개, 100만개의 데이터를 사용하여 진행하였고, 그래프의 파란색은 실제 값, 주황색은 예측값을 나타낸다. 데이터가 약 10만개 이상 존재해야 중간의 그래프처럼 어느 정도의 분포를 예측할 수 있으며, 100만개 이상의 데이터를 사용한다면 낮은 비율을 갖는 분포의 양쪽 꼬리를 제외하고는 원본 데이터의 분포를 추측할 수 있다.

RAPPOR는 사용자의 데이터에 대해 첫 번째 무작위 응답을 거친 Bloom filter, 즉 Fig 1에서 Bloom filter B'에 계산량적 비용을 감소하기 위해 memoization기법을 사용하고 있다. 하지만, 사용자의 데이터가 아주 조금씩 짧은 시간동안 계속해서 변한다면 변화된 값에 대해 다시 계산하고 저장해야 하기 때문에 해당 기법의 장점을 살리기 어렵고, 또한 각각의 파라미터에 따른 유용성에 대한 이론적 분석이 부족하다. MS와 Apple은 이러한 문제점을 완화할 수 있는 새로운 기법을 개발하였다.

3.2 Apple[16]

Apple은 RAPPOR의 유용성 분석에 대한 한계점을 극복하기 위해, 유용성에 대해 이론적으로 증명할 수 있는 새로운 기법을 제안하였다. 현재 스마트 디바이스와 사파리 웹 브라우저에서 사용자의 데이터를 수집 및 활용을 위해 로컬 차분 프라이버시를 만족하는 메커니즘을 적용하고 있으며, CMS(Count Mean Sketch)와 하다마드 변환(Hadamard transform)을 적용한 HCMS(Hadamard Count Mean Sketch)가 있다.

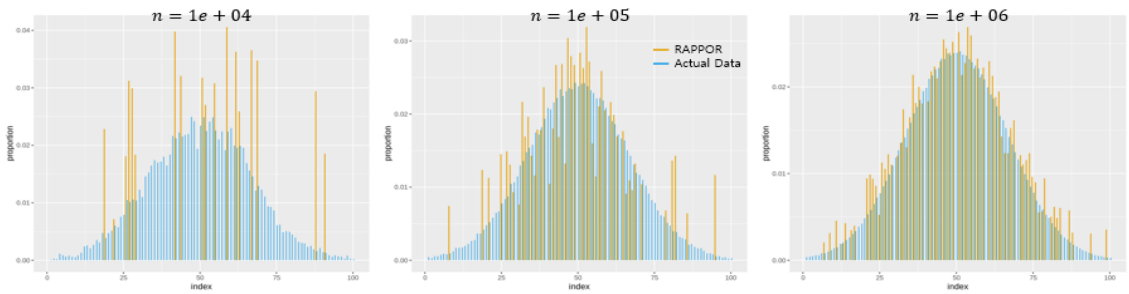

Fig. 3. Simulations of learning the normal distribution by RAPPOR.

3.2.1 CMS(Count Mean Sketch)

CMS 메커니즘은 사용자가 데이터 d를 m비트 길이의 출력값을 갖는 해시 함수에 입력하고, 출력값에 무작위 질의를 기반으로 한 노이즈를 추가하여 서버에 전송한다. 서버는 수집된 데이터를 가공처리하기 위해 Sketch-CMS 알고리즘을 사용하여 사용자들의 데이터를 행렬 M으로 변환하고, 행렬 M을 사용하여 원본 데이터의 분포를 추측한다. Fig 4는 CMS 메커니즘의 전체적인 구조를 나타낸다.

먼저, 사용자 n명의 데이터 $d^{(1)},...,d^{(n)}$와 사용자 데이터의 알려진 후보군 $\hat{D}$, 프라이버시 파라미터 ϵ, 그리고 사용자의 데이터를 입력값으로 하고 크기 m의 출력값을 갖는 3-독립 해시 함수 집합(3-wise independent hash functions family)의 해시 함수 k개가 필요하다. 3-독립 해시 함수 집합은 다음과 같이 정의되는 해시 함수 집합으로 최대 3개의 해시값에 대해 독립적인 분포를 갖는다[24].

Definition 3. (k - 독립 함수 해시 집합).
해쉬 함수 집합 $H=\{h:U\rightarrow[m]\}$에 대해 서로 다른 k개 입력값 $(x_1,...,x_k)\in U^k$과 임의의 k개 출력값 $(y_1,...,y_k)\in[m]^k$에 대해 다음을 만족할 때 k-독립이라고 한다.

$$\Pr_{h\in H}[h(x_1)=y_1\wedge...h(x_k)=y_k]=m^{-k}$$

첫 번째로, 3-독립 해시 함수 집합 H에서 임의의 해시 함수 k개를 정의한다. 다음으로, 사용자는 사용자의 데이터에 노이즈를 추가하는 $A_{dient-CMS}$ 알고리즘을 사용하여 노이즈가 추가된 데이터와 인덱스 j를 서버에 전송한다. 서버는 해당 값들과 데이터를 가공처리하는 Sketch-CMS 알고리즘을 사용하여 행렬 M을 생성하고, 마지막으로 해당 행렬과 사용자들의 데이터 분포를 예측하는 A_{server} 알고리즘을 사용하여 데이터 분포를 예측할 수 있다.

Fig 5의 $A_{dient-CMS}$ 알고리즘을 사용하는 사용자는 먼저 해시 함수의 개수 k에 대해 임의의 j를 선택하고, 크기가 m인 벡터 v의 모든 값을 -1로 초기화한다. 사용자의 값 d와 j번째 해시 함수를 사용하여 계산한 $h_j(d)$를 사용하여, 벡터 v의 $h_j(d)$번째 벡터값 $v_{h_j(d)}$를 1로 바꿔준다. 마지막으로 무작위 응답을 수행한 임의의 벡터 b와 벡터 v의 원소를 각각 곱한 결과값 $\tilde{v}$와 j를 서버에 전송한다.

서버는 n명의 사용자들로부터 수집한 노이즈가 추가된 결과값과 j, 그리고 Fig 6의 Sketch-CMS 알고리즘을 사용하여 행렬 M을 생성한다. 모든 사

CMS mechanisms

Input : $d^{(1)},\dots,d^{(n)}\in D^n$; $\epsilon,k,m,\hat{D}\subseteq D$
Output : Histogram$\left(\tilde{f}(d):d\in\hat{D}\right)$

1. From a set of three-wise independent hashes mapping D to $[m]$, select a set $H=\{h_1,\dots,h_k\}$ of k uniformly at random.
2. **for** $i\in[n]$ **do**
   $\left(\tilde{v}^{(i)},j^{(i)}\right)\leftarrow A_{client-CMS}\left(d^{(i)};\epsilon,H\right)$
3. **Construct** $M\leftarrow$ Sketch $-$ CMS$\left(\left(\tilde{v}^{(1)},j^{(1)}\right),\dots,\left(\tilde{v}^{(n)},j^{(n)}\right);\epsilon,k,m\right)$
4. **for** $d\in\hat{D}$ **do**
   $\tilde{f}(d)\leftarrow A_{server}(d;M,\epsilon,H)$

Fig. 4. Count Mean Sketch

$A_{client-CMS}$ algorithm

Input : d $\in D$; ϵ,H
Output : $\tilde{v}$, index j

1. **Sample** j uniformly at random from $[k]$
2. **Initialize** a vector $v\leftarrow-\mathbf{1}\in\mathbb{R}^m$
3. **Set** $v_{h_j(d)}\leftarrow 1$
4. **Sample** $b\in\{-1,+1\}^m$, where each b_l is i.i.d. where $\Pr[b_l=+1]=\frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$
5. $\tilde{v}\leftarrow(v_1b_1,\dots,v_mb_m)$

Fig. 5. Client-side $A_{dient-CMS}$ algorithm

Sketch $-$ CMS algorithm

Input : d $=\left\{\left(\tilde{v}^{(1)},j^{(1)}\right),\dots,\left(\tilde{v}^{(n)},j^{(n)}\right)\right\}$; ϵ,k,m
Output : Sketch martix **M**

1. **Set** $c_\epsilon\leftarrow\frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}$
2. **for** $i\in[n]$ **do**
   $\tilde{x}^{(i)}\leftarrow k\cdot\left(\frac{c_\epsilon}{2}\cdot\tilde{v}^{(i)}+\frac{1}{2}\cdot\mathbf{1}\right)$
3. Initialize $M\in\{0\}^{k\times m}$
4. **for** $i\in[n]$ **do**
   for $l\in[m]$ **do**
   $M_{j^{(i)},l}\leftarrow M_{j^{(i)},l}+\tilde{x}_l^{(i)}$

Fig. 6. $Sketch-CMS$ algorithm

용자들의 값을 일반화하기 위해 각각의 데이터에 대해 Fig 6의 line 2 연산을 수행한 후에, 해당 값들을 j에 맞추어 모든 값이 0으로 초기화된 행렬 M에 값들을 다 더해준다.

최종적으로, 서버는 사용자들의 데이터가 가공 처리된 행렬 M과 데이터의 후보군 값을 사용하여 사용자들의 데이터 분포를 계산한다(Fig 7).

이러한 CMS 메커니즘을 사용하여 계산된 데이터의 분포에 대한 유용성은 각각의 파라미터에 따라 다음과 같이 보장될 수 있다.

Theorem 1. (CMS에 대한 유용성 보장).

프라이버시 파라미터 $\epsilon>0$, $m,k>1$일 때, 사용자의 값 $d\in\hat{D}$를 사용하여 계산한 CMS 메커니즘의 출력값인 $\tilde{f}(d)$는 d에 대한 빈도 추정값이고, $f(d)$를 d에 대한 실제 값이라고 한다면 다음을 보장할 수 있다.

$$E[\tilde{f}(d)]=f(d)$$

$$Var[\tilde{f}(d)]\le(\frac{m}{m-1})^2\times(\frac{e^{\epsilon/2}}{(e^{\epsilon/2}-1)^2}+\frac{1}{m}+\frac{\sum_{d'\in D}f(d')^2}{n\bullet k\bullet m})\times n$$

A_{server} algorithm

Input : $d\in D; M\in\mathbb{R}^{k\times m}$
Output : $\tilde{f}(d)$

1. Construct $\tilde{f}\ : D\rightarrow\mathbb{R}$ where

$$\tilde{f}(d)=\left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{l=1}^{k}M_{l,h_l(d)}-\frac{n}{m}\right)$$

Fig. 7. A_{server} algorithm

3.2.2 HCMS(Hadamard Count Mean Sketch)

CMS의 유용성에 대한 정리를 분석해 보면, 예측값의 분산을 감소시키기 위해서는 $\frac{\sum_{d\in D}f(d')^2}{n\bullet k\bullet m}$값을 줄여야 한다. 즉, $k\bullet m$을 증가시키면 전체적인 분산을 감소시킬 수 있는데, k와 m을 증가시키게 되면 통신량 및 계산량적 비용이 많이 발생한다. 따라서 이러한 제한적인 부분을 극복하고자 하다마드 변환(Hadamrd transform)을 적용한 메커니즘인 HCMS를 제안하였다.

HCMS는 사용자가 서버에 데이터를 전송할 때, 전체 m비트의 값을 보내는 것이 아니라 하다마드 변환을 사용하여 1비트의 값을 전송하기 때문에 1비트 전송만으로 사용자들의 데이터 값을 예측할 수 있다. 하다마드 행렬은 다음과 같이 재귀적으로 정의할 수 있다.

$$H_1=[1]\qquad H_l=\begin{bmatrix}H_{l/2} & H_{l/2}\\ H_{l/2} & -H_{l/2}\end{bmatrix}\tag{2}$$

H_l의 열벡터들은 직교이고, $H_lH_l=l\bullet I_l$을 만족한다. 여기서 $l\bullet I_l$은 크기가 $l\times l$인 단위행렬의 모든 원소 값이 l인 것을 의미한다.

HCSM는 하다마드 변환을 사용하여 Fig 8과 같이 진행된다.

먼저, Fig 9는 HCSM의 사용자가 서버에게 데이터를 전송하기 위해 사용하는 알고리즘으로, 크기가 m인 벡터 v를 0으로 초기화하고 $[k]$에서 임의로 선택한 j번째 해시 함수를 사용하여 $h_j(d)$를 계산한다. 그 후에 v의 $h_j(d)$번째 비트인 $v_{h_j(d)}$를 1로 바

HCMS mechanisms

Input : $d^{(1)},\dots,d^{(n)}\in D^n;\ \epsilon,k,m,\hat{D}\subseteq D$
Output : Histogram$(\tilde{f}(d):d\in\hat{D})$

1. From a set of three-wise independent hashes mapping D to $[m]$, select a set $H=\{h_1,\dots,h_k\}$ of k uniformly at random.
2. **for** $i\in[n]$ **do**
   $(\tilde{w}^{(i)},j^{(i)},l^{(i)})\leftarrow A_{client-HCMS}(d^{(i)};\epsilon,H)$
3. **Construct** $M\leftarrow$ Sketch − HCMS$\left((\tilde{w}^{(1)},j^{(1)},l^{(1)}),\dots,(\tilde{w}^{(i)},j^{(i)},l^{(i)});\epsilon,k,m\right)$
4. **for** $d\in\hat{D}$ **do**
   $\tilde{f}(d)\leftarrow A_{server}(d;M,\epsilon,H)$

Fig. 8. Hadamard Count Mean Sketch

$A_{client-HCMS}$ algorithm

Input : $d\in D;\ \epsilon,H$
Output : $\tilde{v}$, index j

1. **Sample** j uniformly at random from $[k]$
2. **Initialize** a vector $v\leftarrow-\mathbf{1}\in\mathbb{R}^m$
3. **Set** $v_{h_j(d)}\leftarrow1$
4. **Transform** $w\leftarrow H_mv$.
5. **Sample** l uniformly at random from $[m]$
6. **Sample** $b\in\{-1,+1\}$, which is $+1$ with probability $\frac{e^\epsilon}{e^\epsilon+1}$
7. **Set** $\tilde{w}\leftarrow bw_l$

Fig. 9. $A_{dient-HCMS}$ algorithm

Sketch – HCMS algorithm

Input : $\mathrm{d} = \{(\tilde{w}^{(1)}, j^{(1)}, l^{(1)}), \ldots, (\tilde{w}^{(i)}, j^{(i)}, l^{(i)})\}$; ϵ, k, m
Output : $\tilde{v}$, index j

1. **Set** $c_\epsilon \leftarrow \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}$
2. **for** $i \in [n]$ **do**
   $\tilde{x}^{(i)} \leftarrow k \cdot \left(\frac{c_\epsilon}{2} \cdot \tilde{v}^{(i)} + \frac{1}{2} \cdot \mathbf{1}\right)$
3. **Initialize** $M \in \{0\}^{k \times m}$
4. **for** $i \in [n]$ **do**
   $M^H_{j^{(i)}, l^{(i)}} \leftarrow M^H_{j^{(i)}, l^{(i)}} + \tilde{x}_l^{(i)}$
5. **Transform** the rows of sketch back $M^H \leftarrow M^H H_m^T$

Fig. 10. Sketch-HCMS algorithm

꿔준다. 그리고 크기가 m인 하다마드 행렬 H_m을 사용하여 하다마드 변환을 수행한 후에, 임의의 l을 $[m]$에서 선택한다. 마지막으로 무작위 응답을 수행한 $b \in \{-1, +1\}$와 변환된 사용자의 값인 w벡터의 l번째 원소를 곱하여 j, l과 함께 서버로 전송한다.

Fig 10은 사용자들의 데이터를 수집하여 서버가 행렬 M을 생성하는 Sketch-HCMS 알고리즘을 나타낸다. 이전의 Fig 6의 Sketh-CMS와 동일하게 수행되고, 마지막에 하다마드 변환을 적용한 사용자들의 데이터에 역변환을 수행하여 최종적인 행렬 M^H을 생성한다.

최종적으로, 서버가 사용자들의 데이터 분포를 계산하는 알고리즘은 Fig 7과 동일하게 진행된다.

HCMS 메커니즘을 사용하여 계산된 분포에 대한 유용성 보장은 다음과 같이 정리할 수 있다.

Theorem 2. (HCMS에 대한 유용성 보장).
프라이버시 파라미터 $\epsilon > 0$, 임의의 사용자 값 $d \in \widehat{D}$에 대해서, $\tilde{f}(d)$를 HCMS 메커니즘을 사용하여 계산된 d에 대한 빈도 추정값이고, $f(d)$를 d에 대한 실제 값이라고 한다면 다음을 보장할 수 있다.

$$E[\tilde{f}(d)] = f(d)$$

$$Var[\tilde{f}(d)] \le \left(\frac{m}{m-1}\right)^2 \times \left(\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)^2 + \frac{\sum_{d' \in D} f(d')^2}{n \cdot k \cdot m}\right) \times n$$

Table 1은 실제 메커니즘이 적용되고 있는 환경과 사용되고 있는 변수를 나타낸다. 일반적으로 프라이버시 파라미터 $\epsilon \ge 2$를 사용하며, 해시 함수 개수 k와 벡터의 크기 m은 각각 적용되는 메커니즘과 상황에 따라 다른값을 사용한다.

3.3 Microsoft(MS)[17]

RAPPOR 메커니즘은 계산량적 비용을 줄일 수 있는 memoization기법을 적용하였지만, 사용자의 데이터 값이 조금씩 자주 변하게 된다면 해당 기법의 장점을 살릴 수 없다. MS는 이러한 계산량적 비용 감소의 장점을 살리기 위해 해당 기법을 새롭게 변형하여 적용한 새로운 메커니즘을 제안하였다.

MS는 평균값을 추측하기 위해 사용자의 데이터를 1비트만 전송하는 1BitMean 메커니즘과 분포를 예측하기 위한 dBitFlip 메커니즘을 제안하고, 해당 메커니즘은 윈도우 운영체제의 업데이트를 통해 배포되어 애플리케이션 사용 데이터 수집에 사용된다.

3.3.1 1BitMean

1BitMean 메커니즘은 사용자의 데이터를 1비트 크기로 수집하여 데이터에 대한 평균값을 추정한다. 가장 기본적으로 사용자들은 Fig 11의 수집 알고리즘에 나타나는 것처럼 t시간에 사용자 i의 데이터인 $x_i(t) \in [0, m]$에 무작위 응답을 적용하여 1비트 값을 전송하게 된다. 전체 n명의 사용자에 대한 데이터를 수집한 서버는 평균 추정 알고리즘을 사용하여 t 시간의 평균값 $\sigma(t)$의 추정값 $\hat{\sigma}(t)$을 계산할 수 있다.

Table 1. Used mechanisms and parameters for each cases in Apple.

Using case	Mechanism	ϵ	k	m	p
Discovering Popular Emojis	CMS	4	65,536	1024	2600
Safari Auto-play Intent	CMS	8	65,536	1024	250,000
Identifying High Energy and Memory Usage in Safari	HCMS	4	1024	32,768	250,000
Understanding HealthKit Usage	CMS	2	65,536	256	x

Collection algorithm for 1BitMean

When the collection of counter $x_i(t)$ at time t is requested by the data collector, each user i sends one bit $b_i(t)$.

$$b_i(t) = \begin{cases} 1, & with\ probability\ \frac{1}{e^\epsilon+1} + \frac{x_i(t)}{m} \cdot \frac{e^\epsilon-1}{e^\epsilon+1} \\ 0, & otherwise. \end{cases}$$

Mean estimation algorithm for 1BitMean

Data collector obtains the bits $\{b_i(t)\}_{i\in[n]}$ for n users and estimates $\sigma(t)$.

$$\tilde{\sigma}(t) = \frac{m}{n}\sum_{i=1}^{n} \frac{b_i(t)\cdot(e^\epsilon+1)-1}{e^\epsilon-1}$$

Fig. 11. Algorithm of collection and mean estimation for 1BitMean

MS는 위의 알고리즘을 기초로 하여, Fig 12에 나타나는 것처럼 다양한 기법을 함께 적용하여 사용한다. 먼저, 데이터 값이 조금씩 자주 변하는 상황에서도 memoization 기법을 사용할 수 있도록 α-point rounding기법을 함께 사용한다. 추가적으로 사용자의 데이터 값이 변하는 시점의 유출을 막기 위해, 서버로 전송하기 전에 한 번 더 무작위 응답을 수행하여 안전하게 데이터를 전송한다.

α-point rounding은 사용자의 데이터 $x_i(t)\in[0,m]$를 범위 s로 나누어 계산하는 방법이다. 먼저 정수 m의 약수 중, 임의의 s를 설정하고 각각의 사용자들은 임의의 값 $\alpha_i\in\{0,...,s-1\}$를 선택한다. 그리고 다음의 식을 만족하는 $\frac{m}{s}+1$ 개의 값에 1BitMean 알고리즘을 적용하여 계산된 출력값을 저장하고 재사용한다.

$$A = \{ls\}_{0 \le l \le \frac{m}{s}} \tag{3}$$

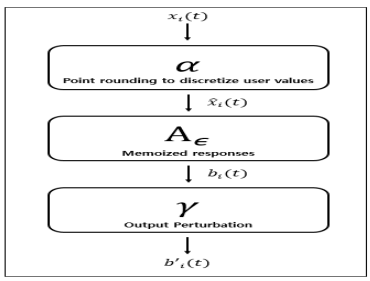

Fig. 12. Framework of 1BitMean

사용자는 데이터를 서버로 전송할 때, 데이터 x_i에 대해 α_i를 사용하여 다음과 같이 집합 A의 원소 L 또는 R로 대칭시킨다. 이때, L과 R은 x_i에 가장 인접한 A의 원소를 의미한다.

$$\hat{x}_i = \begin{cases} L & \text{if } x_i+\alpha_i < R \\ R & otherwise. \end{cases} \tag{4}$$

이처럼 원본 데이터를 집합 A의 원소로 대칭시키기 때문에 사용자의 값이 주기적으로 조금씩 변하더라도 다시 계산하지 않고 저장된 값을 재사용 할 수 있다.

그리고 사용자들의 값이 변하는 시점의 유출을 방지하기 위해 확률 γ를 사용하여 다음과 같이 출력값에 무작위 응답을 수행한다.

$$\hat{b}_i(t) = \begin{cases} b_i(t) & with\ probability\ 1-\gamma \\ 1-b_i(t) & otherwise \end{cases} \tag{5}$$

이때, $b_i(t)$는 α-point rounding 후에 저장된 값으로, $\hat{x}_i$에 1BitMean 알고리즘을 적용한 계산된 출력값을 의미한다.

3.3.2 dBitFlip

dBitFlip 메커니즘은 사용자들의 데이터가 범주형 자료인 경우, 데이터의 분포를 예측하기 위해 사용한다. 먼저 각각의 사용자는 전체 k개의 항목에 대해 d개의 항목 $j_1,...,j_d$를 임의로 선택한다. 그리고 각각의 항목에 해당하는 사용자의 값인 $v_i(t)\in[k]$에 무작위 응답을 적용한 값인 $b_{i,j_p}(t)\in\{0,1\}$를 j_d와 함께 서버에 전송한다(Fig 13). 그리고 계산된 결과값인 $b_{i,j_p}(t)$는 저장하여 재사용한다.

서버는 Fig 13의 분포 추정 알고리즘을 사용하여 전체 사용자들의 항목별 분포 $h_t(v)$에 대한 예측값 $\hat{h}_t(v)$를 계산할 수 있다.

1BitMean 메커니즘은 memoization 기법에서 발생할 수 있는 취약점을 보안하기 위해 출력값에 무작위 응답을 한 번 더 적용하였지만, dBitFlip 메커니즘은 k보다 작은 d값을 사용하여 계산된 값들 사이의 충돌을 발생시켜 발생할 수 있는 취약점을 보안하였다.

Collection algorithm for dBitFlip

Each user i randomly draws d bucket numbers without replacement from $[k]$, denoted by $j_1, j_2, \ldots, j_d$.
When the collection of discretized bucket number $v_i(t) \in [k]$ at time i is requested by the data collector, each user i sends a vector:

$$b_i(t) = [(j_1, b_{ij_1}(t)), (j_2, b_{ij_2}(t)), \ldots, (j_d, b_{ij_d}(t))]$$

where $b_{ij_p}(t)$ is a random 0-1 bit,

$$\text{with } Pr[b_{ij_p}(t) = 1] = \begin{cases} e^{\epsilon/2}/(e^{\epsilon/2}+1) & \text{if } v_i(t) = j_p, \text{ for } p = 1, 2, \ldots, d. \\ 1/(e^{\epsilon/2}+1) & \text{if } v_i(t) \neq j_p \end{cases}$$

Histogram estimation algorithm for dBitFlip

Data collector obtains the bits $\{b_i(t)\}_{i \in [n]}$ for n users and estimates $\sigma(t)$.

$$\tilde{\sigma}(t) = \frac{m}{n} \sum_{i=1}^{n} \frac{b_i(t) \cdot (e^{\epsilon}+1) - 1}{e^{\epsilon} - 1}$$

Fig. 13. Algorithm for collection and histogram estimation in dBitFlip

IV. 프라이버시 보존형 설문조사

최근의 설문조사는 온라인 상에서 다양한 디바이스를 통해서 수행되고 있으며, 이때 설문조사의 결과는 참여자들의 민감한 정보를 포함할 수 있다. 이를 통해 개인의 프라이버시 침해가 발생할 수 있다. 특히, 불특정 다수를 대상으로 진행하는 것이 아닌 특정 지역 또는 특정 그룹 안에서 진행하는 설문조사의 경우 참여자는 본인의 생각대로 설문조사에 참여하지 못할 수 있다. 예를 들어, 기업의 직원들을 대상으로 진행하는 설문조사에서 참여자들은 본인의 응답이 노출되어 발생할 수 있는 불이익을 걱정하여 기업적으로 바람직한 방향 또는 기업의 기대에 부응할 수 있는 응답을 하려는 경향이 있다[18-22].

실제 창원대학교에서는 익명성이 설문조사의 결과에 미치는 영향에 대해 실험하였다[18]. 학생들을 대상으로 학번과 주민등록번호 앞자리의 입력을 통해 모든 참여자의 신분을 알 수 있는 설문조사와 익명의 설문조사를 시행하였다. 그 결과, 학생들의 신분이 드러날 때는 익명의 설문조사보다 학교에 대한 만족도가 약 23% 포인트 높게 집계되었다.

이처럼 개인 프라이버시 침해 가능성과 본인의 응답 노출 가능성은 설문 및 여론조사의 결과에 큰 영향을 미치기 때문에, 조사결과의 신뢰성이 심각하게 감소할 수 있다. 이에 따라, 다양한 설문 및 여론조사 대행업체는 익명성을 강조하며, 개인의 프라이버시가 보존되는 설문조사를 진행하고 있음을 밝혔다. 하지만, 신뢰성 있는 결과를 도출하기 위해서는 참여자가 대행업체를 전적으로 믿어야 한다는 한계점이 있다.

따라서 이번 장에서는 개인이 누구도 믿지 않는 상황에서 프라이버시 노출 가능성을 낮추고, 신뢰할 만한 결과물을 계산할 수 있는 로컬 차분 프라이버시를 적용한 설문조사에 대해 분석한다.

4.1 로컬 차분 프라이버시 적용 설문조사

3장에서 살펴본 많은 기업들이 제안한 로컬 차분 프라이버시를 만족하는 메커니즘은 크게 평균값 계산과 분포 추정을 목적으로 이루어졌다. 또한, 설문 및 여론조사도 개개인의 응답보다는 전체적인 분포 또는 평균값을 알기 위해 진행된다. 따라서 차분 프라이버시 메커니즘을 설문조사 시나리오에 적용한다면 유용하게 사용할 수 있다.

각각의 메커니즘을 다시 살펴보면, Google의 RAPPOR, Apple의 CMS 및 HCMS 그리고 MS의 dBitFlip은 데이터의 분포를 추정을 목적으로 사용되고, 특히 HCMS는 추가적인 연산을 통해 1비트 크기의 데이터만을 전송한다. 그리고 MS의

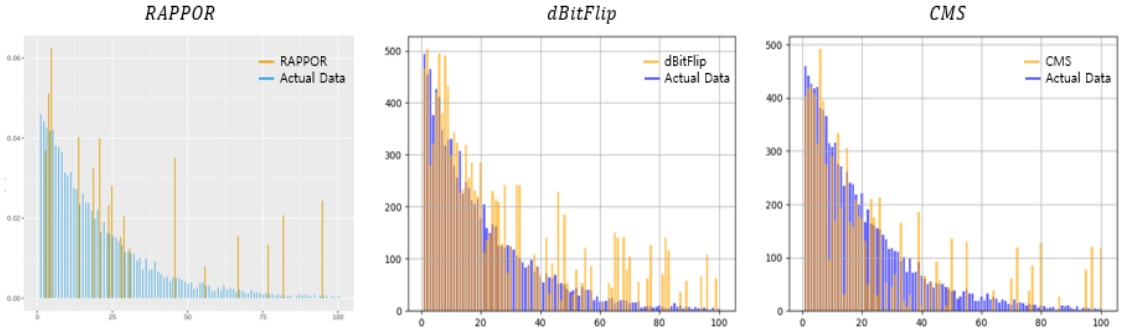

Fig. 14. Simulations of learning the exponential distribution by RAPPOR, dBitFlip, and CMS.

1BitMean은 1비트 크기의 데이터를 전송하여 데이터의 평균값을 계산하기 위해 사용된다.

설문 및 여론조사에서의 질문에 대한 답변은 5개, 7개 등 평균적으로 10개 이하로 이루어져 있다. 3장에서 소개한 메커니즘들은 사용자들의 데이터 후보군이 매우 큰 상황에서 이루어졌지만, 설문조사는 상대적으로 매우 작은 후보군을 가지고 있다. 따라서 추가적인 연산을 사용하여 1비트의 메시지만 전송하는 메커니즘은 적절하지 않다. 즉, 1비트의 데이터만을 전송하는 메커니즘을 제외한 RAPPOR, CMS, dBitFlip 메커니즘을 사용한다.

설문 및 여론조사는 목적과 환경에 따라 참여자의 수는 다양하지만, 만 명 이하의 표본을 대상으로 수행하는 경우가 많다. 따라서 먼저 만개의 데이터를 대상으로 수행한 결과값을 비교하여 각각의 메커니즘에 대한 사용 여부를 결정하고, 사용 결정된 메커니즘에 대해 설문조사에서의 적용 가능성을 분석한다.

Fig 14는 지수 분포를 따르는 임의의 데이터 10000개를 생성하여 RAPPOR, CMS, dBitFlip을 수행한 결과를 나타낸다. RAPPOR의 경우, 이전의 Fig 3에서와 동일하게 기존의 분포와는 완전히 다른 몇 개의 항목에 대한 데이터만 계산되었다. 하지만, dBitFlip과 CMS는 RAPPOR와 비교할 때 더 많은 항목에 대해 결과값을 생성하고, 비율이 높은 항목에 대해서는 상대적으로 좋은 유용성을 갖는다고 할 수 있다. 결론적으로, 10000개 이하의 데이터를 대상으로 RAPPOR를 수행하는 것은 부적절하다. 따라서 본 논문에서는 설문조사 시나리오에서의 CMS와 dBitFlip에 대한 적용 가능성을 분석한다.

4.2 로컬 차분 프라이버시 메커니즘 적용 가능성 분석

설문조사(또는 여론조사) 환경에 적용하기 위해서는 알맞은 변수를 선택하여 각각의 메커니즘을 사용해야 한다.

CMS는 사용자가 전달하는 메시지의 크기 m과 해시 함수의 개수 k, 그리고 프라이버시 파라미터 ϵ을 각각의 환경에 맞게 설정해야 한다. Table 1의

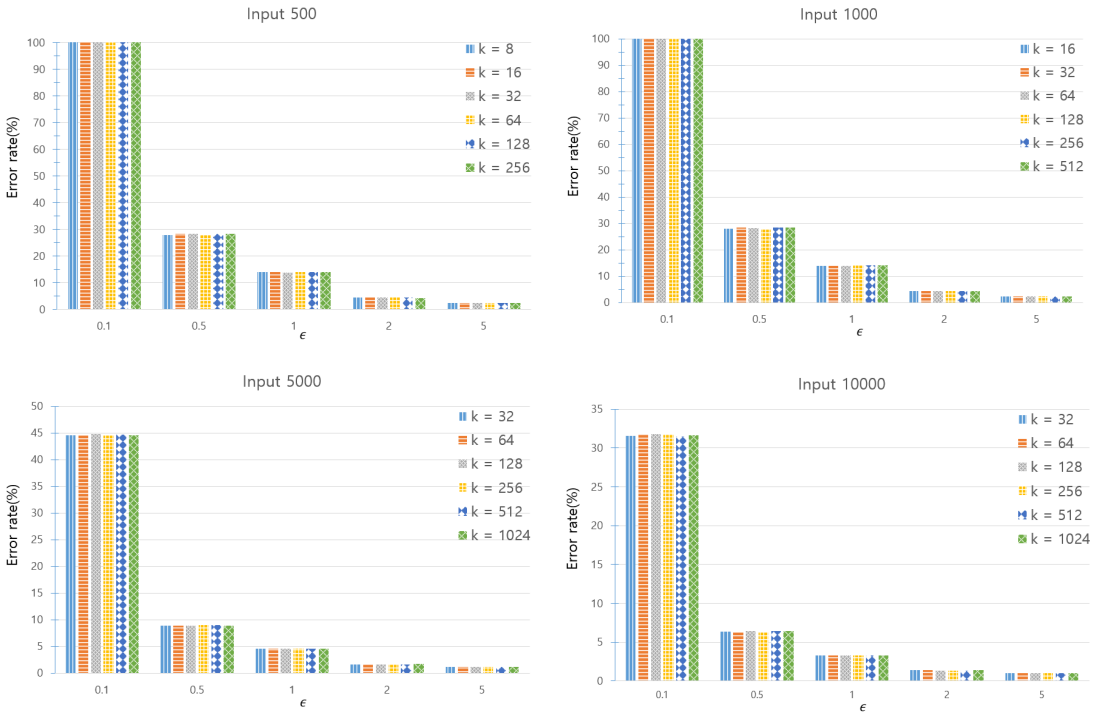

Fig. 15. Simulations of learning the uniform distribution with CMS to measure the mean of maximum error rate for each parameter. Each graph is simulated with different the size of input. We fix the size of vector($m = 128$) and use the various parameters such as $\epsilon \in \{0.1, 0.5, 1, 2, 5\}$, $k = 2^l (l \in [3,10])$

실제 Apple에서 적용한 값을 사용할 수 있지만, 해당 값들은 사용자들의 데이터 후보군 개수 p가 매우 큰 상황에서 수행되었기 때문에 적절하지 않을 수 있다. 따라서 후보군 개수 $p=5$인 일반적인 설문조사 상황에 알맞은 변수를 실험을 통해 확인한다.

Fig 15는 CMS를 다양한 환경에서 실험하여 최대 에러값을 측정한 결과이다. 해당 실험은 메시지의 크기($m=128$)를 고정하고, ϵ과 k에 따라 사용자의 데이터 후보군이 5개인 상황에서 발생하는 최대 에러 즉, 실제 입력값과 추측값의 차이 중 가장 큰 값에 대해 3000번 반복한 평균값을 계산하였다. 해당 실험은 균일분포를 사용하여 생성한 임의의 데이터를 각각 500개, 1000개, 5000개, 10000개를 사용하였다. 데이터를 500개, 1000개를 사용한 위쪽의 그래프에서 $\epsilon=0.1$인 경우를 살펴보면 평균 최대 에러율은 100% 즉, 원본 데이터의 값이 완전히 뒤집혀 나온 결과라고 파악할 수 있다. 따라서 데이터가 1000개 이하라면 $\epsilon>0.1$을 사용해야 한다. 해시 함수의 개수 k가 에러율에 가장 많은 영향을 끼친 환경은 입력값 1000개, $\epsilon=0.5$에서 $k=256, 512$이고, 이때 에러율의 차이가 약 0.48% 포인트 발생한다. 일반적으로 입력값이 5000개 이상, $\epsilon\geq 1$인 경우에는 k에 따른 에러율의 차이가 약 0.05% 이하로 발생한다. 결론적으로 CMS에서 해시 함수의 개수 k는 평균 최대 에러율에 큰 영향을 주지 않고, 따라서 본 논문에서는 $k=512$를 사용하여 에러율을 측정하고 dBitFlip과 비교한다.

dBitFlip은 데이터 후보군 개수 k에 대해 임의의 d를 선택하여 사용한다. 이때, k보다 작은 d를 사용하여 출력값들 사이의 충돌을 발생시켜 공격자가 출력값으로 통해 사용자의 데이터를 추론할 수 있는 memoization 기법의 취약점을 보안하였다. 하지만, memoization 기법을 적용하지 않고 데이터를 전송할 때마다 새롭게 계산된 값을 전송한다면, 해당 취약점은 발생하지 않기 때문에, $d=k$를 적용한 dBitFlip 메커니즘을 사용할 수 있다. 또한, 본 논문에서 고려하는 설문조사 환경에서는 데이터의 후보군 개수가 5개이기 때문에 memoization 기법을

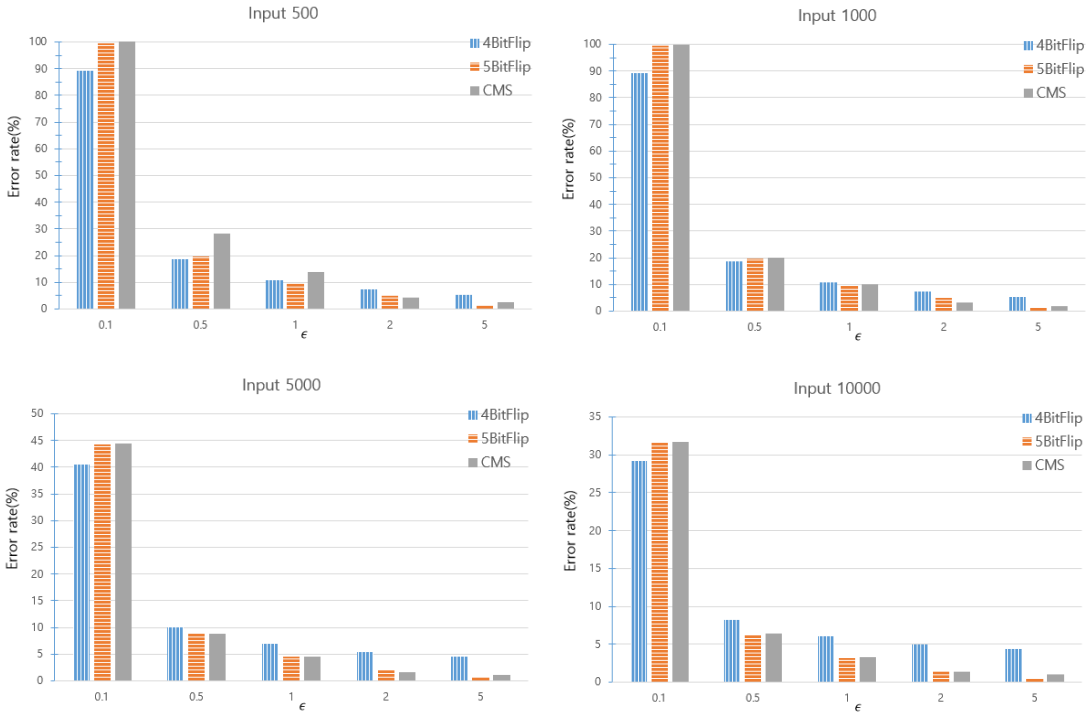

Fig. 16. Simulations of learning the uniform distribution with CMS, 4BitFlip, and 5BitFlip to measure the mean of maximum error rate for each parameter. Each graph is simulated with different the size of input.

적용하여 발생하는 장점인 계산량적 비용 감소가 상대적으로 큰 영향을 주지 않는다. 따라서 memoization을 적용하지 않은$d=k$를 만족하는 5BitFlip과 memoization을 적용하고 $d<k(d=4)$를 만족하는 4BitFlip을 모두 고려하여 비교한다.

4.3 실험 결과

설문 및 여론조사에 적합한 메커니즘을 찾기 위해 Apple의 CMS와 MS의 dBitFlip를 비교분석한다. 먼저 CMS는 $k=512, m=128$을 적용하고, dBitFlip은 $d=4,5$을 적용한 4BitFlip과 5BitFlip을 구현하여 비교한다. Fig 16은 총 3가지 메커니즘에 대해 입력값의 개수와 프라이버시 파라미터 ϵ에 따른 최대 에러의 평균값을 나타낸 그래프이다. 전체 3000번 반복실험의 평균값을 나타내며, 입력값은 균일분포를 사용한 임의의 500, 1000, 5000, 10000개의 데이터를 생성하여 동일한 데이터에 대해 실험하였다.

입력값이 500개, 1000개인 위의 2개 그래프에 대해, ϵ이 0.5 이하인 경우에는 4BitFlip, ϵ이 1, 5인 경우에는 5BitFlip의 에러율이 가장 낮고, ϵ이 2인 경우 CMS가 가장 낮게 나타난다. $\epsilon=0.1$인 경우, 에러율이 약 90% 이상이므로 현실적으로 사용이 어렵다. 상대적으로 엄격한 프라이버시를 보장하면서 에러율을 20%로 설정한다면, $\epsilon=0.5$에서 4BitFlip을 적용하는 것이 좋다. Apple에서 실제 적용하고 있는 프라이버시 파라미터 $\epsilon \geq 2$에 해당하는 $\epsilon=2,5$를 확인해보면, 4BitFlip을 제외한 나머지 2개의 메커니즘은 평균적으로 약 4%의 에러율을 나타낸다.

입력값 5000개인 경우, $\epsilon=1$인 경우에 모든 메커니즘의 평균 에러율이 약 10% 이하로 감소하고, 마지막으로 입력값 10000개에 대한 실험 결과는 이전의 5000개에 대한 결과와 비교했을 때, 전체적인 에러율이 감소하고 $\epsilon=5$인 경우 5BitFlip과 CMS는 에러율이 1%이하로 감소한다.

CMS와 5BitFlip을 비교하면 평균적으로 에러율은 2% 포인트 차이가 발생한다. 따라서, 10000개 이상의 데이터에 대해서는 두 메커니즘의 결과값에 큰 차이가 발생하지 않을 것으로 예상된다.

실제 로컬 차분 프라이버시를 적용한 프라이버시 보존형 설문조사(또는 여론조사)를 사용할 때, 조사자(또는 조사기관)는 참여자의 수와 환경에 따라 적절한 메커니즘을 선정해야 한다. 10000명 이하의 참여자를 가정한 실험결과 전반적으로 5BitFlip 메커니즘의 에러율이 가장 낮게 나오지만, 해당 메커니즘은 참여자가 데이터를 전송할 때 매번 새롭게 계산해서 전송해야하는 단점이 있다. 따라서 참여자가 계산된 값을 저장하여 사용하는 CMS 메커니즘을 적용한다면, 5BitFlip보다 에러율이 소폭 증가하지만 참여자의 계산량적 비용을 감소시킬 수 있다. 또한, 참여자의 수가 1000명 이하인 상황에서 상대적으로 엄격한 프라이버시($\epsilon \leq 0.5$)를 보장하기 위해서는, 사용자의 계산량적 비용이 감소되는 4BitFlip을 사용할 때 에러가 가장 적은 결과값을 계산할 수 있다. 미자막으로 10000명 이상인 경우에는 5BitFlip과 CMS 모두 유용한 결과값을 계산할 수 있으며, 상황에 따라 최저 에러율을 갖는 메커니즘을 Table 2에 표로 정리하였다.

V. 결 론

로컬 차분 프라이버시는 데이터 소유자가 직접 데이터에 무작위 응답 기반의 노이즈를 추가여 데이터를 제공한다. 따라서 데이터 소유자는 누구도 믿지 않는 상황에서 데이터를 제공할 수 있고, 데이터 분석가는 해당 데이터를 통해 원하는 통계적 결과값을 계산할 수 있다. 이러한 로컬 차분 프라이버시를 만족하는 메커니즘은 현재 세계적인 IT기업인 Apple,

Tabel 2. The lowest error rate mechanisms for each number of participants and privacy parameter ϵ.

The number of participants	$\epsilon=0.1$	$\epsilon=0.5$	$\epsilon=1$	$\epsilon=2$	$\epsilon=5$
10000	x	5BitFlip	5BitFlip	CMS	5BitFlip
5000	x	5BitFlip	5BitFlip	CMS	5BitFlip
1000	x	4BitFlip	5BitFlip	CMS	5BitFlip
500	x	4BitFlip	5BitFlip	CMS	5BitFlip

Google, 그리고 MS에서 실제 사용자들의 데이터를 수집 및 활용하기 위해 사용되고 있다.

본 논문에서는 로컬 차분 프라이버시를 만족하는 메커니즘을 현실 문제에 적용할 수 있는 활용 가능성을 분석하기 위해 설문조사(또는 여론조사) 시나리오에 적용하였다. 설문조사의 결과에는 참여자의 민감한 정보가 포함될 수 있으므로 프라이버시 침해가 발생할 수 있다. 또한, 참여자들은 응답 노출 가능성을 우려하여 정확한 응답을 제출하지 않는다. 이처럼 익명성은 설문조사 결과의 신뢰성에 큰 영향을 끼친다. 다시 말해, 응답자가 누구도 믿지 않는 상황에서 프라이버시가 보존된다면 보다 정확한 응답을 할 것이며 결과의 신뢰성은 높아질 것이다. 따라서 이러한 설문조사에 로컬 차분 프라이버시를 만족하는 메커니즘을 적용한다면 응답자와 조사자 모두에게 유용한 조사가 이루어 질 수 있다.

일반적으로 10개 이하의 응답으로 구성되어진 설문조사는 많은 인원에 대해 수행하지 않고 소수의 표본을 추출해 수행한다. 또한, 응답결과의 분포가 중요한 결과값으로 사용되고, 이러한 상황에서는 MS의 dBitFlip과 Apple의 CMS가 적합하다고 판단하여 수행하였다. 이때, 조사자(또는 조사기관)는 설문조사가 이루어지는 환경과 응답자의 수에 따라 적절한 방법을 선택하여 적용해야 한다. 1000명 이하의 응답자를 대상으로 엄격한 프라이버시($\epsilon < 1$)를 보장하기 위해서는 4BitFlip을 적용하는 것이 상대적으로 에러를 줄일 수 있다. 또한, 10000명 이하의 응답자에 대해 수행되는 설문조사에서는 5BitFlip을 적용하는 것이 가장 에러가 적고, CMS를 적용하면 상대적으로 에러가 소폭 증가하지만 응답자의 계산량적 비용을 줄일 수 있다. 그리고, 10000명 이상의 응답자에 대해 수행될 때, 5BitFlip과 CMS 모두 유용한 결과값을 도출해 낼 것이다.

로컬 차분 프라이버시에 많은 연구들이 진행되고 있으며, 본 논문은 설문조사 시나리오에 적용하여 현실 적용 가능성을 분석하였다. 이처럼 이론적으로 엄격한 로컬 차분 프라이버시를 현실 문제에 적용하기 위한 지속적인 연구가 필요하다.

References

〔1〕 C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," Proceedings of the 3rd Theory of Cryptography Conference, pp. 265-284. Mar, 2006

〔2〕 C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential Privacy under Continual Observation," Proceedings of the forty-second ACM symposium on Theory of computing, STOC' 10, pp.715-724, Jun, 2010

〔3〕 C. Dwork, and A. Roth, "The Algorithmic Foundations of Differential Privacy," Foundations and Trends in Theoretical Computer Science, 9(3-4), pp. 211-407, Aug, 2014

〔4〕 S. L. Warner, "Randomized Response :A Survey Technique for Eliminating Evasive Answer Bias," Journal of the American Statistical Association, 60(309), pp. 63-69, Apr, 1965

〔5〕 G. Cormode, T. Kulkami, and D. Srivastava, "Marginal Release Under Local Differential Privacy," Proceedings of the 2018 International Conference on Management od Data, pp. 131-146, Jun, 2018

〔6〕 X. Ren, et al, "LoPub : High-Dimensional Crowdsourced Data Publication With Local Differential Privacy," IEEE Transactions on Information Forensics and Security, 13(9), Sep, 2018

〔7〕 Peng Liu et al, "Local Differential Privacy for Social Network Publishing," Neurocomputing, https://doi.org/10.1016/j.neucom.2018.11.104, 2018

〔8〕 Z. Zhao et al, "LDPard: Effective Location-Record Data Publication via Local Differential Privacy," IEEE Access, 7, pp. 31435-31445, 2019

〔9〕 J. C. Duchi, M. J. Wainwright, and M. I. Jordan, "Minimax Optimal Procedures for Locally Private

Estimation," Journal of the Americans Statistical Association, 113(521), May, 2018
〔10〕 Z. Qin et al, "Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 192-203, Aug, 2016
〔11〕 M. Bum, J. Nelson, and U. Stemmer, "Heavy Hitters and the Structure of Local Privacy," Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 435-447, Jun, 2018
〔12〕 S. Wang et al, "Local Differential Private Data Aggregation for Discrete Distribution Estimation," IEEE Transactions on Parallel and Distributed Systems, 30(9), Sep, 2019
〔13〕 J. Jia, and N. Z. Gong, "Calibrate: Frequency Estimation and Heavy Hitter Identification with Local Differential Privacy via Incorporating Prior Knowledge," IEEE INFOCOM 2019, pp. 2008-2016, Apr, 2019
〔14〕 T. Murakami, and Y. Kawamoto, "Utility-Optimized Local Differential Privacy Mechanisms for Distribution Estimation," USENIX Security Symposium-2019, pp. 1877-1894, Aug, 2019
〔15〕 U. Erlingsson, V. Pihur, and A, Korolova, "RAPPOR : Randomized Aggregatable Privacy-Preserving Ordinal Response," Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054-1067, Nov, 2014
〔16〕 Differential Privacy Team, "Learning with Privacy at Scale," Apple Machine Learning Journal, 1(8), Dec, 2017
〔17〕 B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting Telemetry Data Privately," proceedings of NIPS 2017, pp. 3574-3583, Dec, 2017
〔18〕 G. H. Lee, and J. D. Young, "A Study on the Measurement Methods and Cases of Personal Information Leakage Risks of Private Companies," Review of KIISC, 18(3), pp. 92-100, Jun, 2008
〔19〕 P. K. Tyagi, "The Effects of Appeals, Anonymity, and Feedback on Main Survey Response Patterns from Salespeople," Journal of the Academy of Marketing Science, 17(3), pp. 235-241, Jun, 1989
〔20〕 V. Toepoel, Doing Surveys Online, SAGE, Nov, 2015
〔21〕 A. E. Arafa et al, "Perspectives of Online Survey in Dermatology," Journal of The European Academy of Dermatology and Vernerology, 33(3), pp. 511-520, Mar, 2019
〔22〕 R. Tourangeaus et al, "The Presentation of a Web Survey, Nonresponse and Measurement Error among Members of Web Panel," Journal of Official Statistics, 25(3), pp. 299-321, Sep, 2009
〔23〕 R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," Journal of the Royal Statistical Society, Series B, 58(1), pp. 267-288, Jan, 1996
〔24〕 T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithm, 3rd Ed., The MIT Press, 1990

〈저자소개〉

정 수 용 (Sooyong Jeong) 학생회원
2018년 2월: 공주대학교 응용수학과 학사
2018년 3월~현재: 공주대학교 융합과학과 석사과정
〈관심분야〉 암호모듈 구현, 데이터 보안

홍 도 원 (Dowon Hong) 종신회원
1994년 2월: 고려대학교 수학과 학사
2000년 2월: 고려대학교 수학과 박사
2000년 4월~2012년 2월: 한국전자통신연구원 팀장, 책임연구원
2012년 3월~현재: 공주대학교 응용수학과 교수
〈관심분야〉 암호기술, 프라이버시 보호기술

서 창 호 (Changho Seo) 종신회원
1990년: 고려대학교 수학과 학사
1992년: 고려대학교 수학과 이학석사
1996년: 고려대학교 수학과 이학박사
1996년~1996년: 국방과학연구소 선임연구원
1996년~2000년: 한국전자통신연구원 선임연구원, 팀장
2000년~현재: 공주대학교 응용수학과 교수
〈관심분야〉 암호알고리즘, PKI, 무선인터넷 보안 등