https://doi.org/10.7840/kics.2023.48.4.445

Functional Encryption을 이용한 프라이버시 보호 온라인 광고 시스템

이 재 현•, 김 현 수*, 권 태 경°

Privacy-Preserving Real Time Bidding through Functional Encryption

Jaehyun Lee•, Hyunsoo Kim*,
Ted "Taekyoung" Kwon°

요 약

Real-Time Bidding(RTB)으로 대표되는 온라인 광고환경은 맞춤형 광고를 제공하기 위하여 사용자의 개인정보를 비롯한 다양한 정보를 제 3자인 광고주 측에 제공하고 있다. 이러한 환경에서 개인정보 유출에 대한 우려가 제기되고 있으며, 이를 해결하기 위한 다양한 대안들이 제시되고 있다. 본 논문에서는 차세대 암호화 기법 중 하나인 Functional Encryption을 RTB 환경에 도입함으로써 프라이버시를 보호하면서 맞춤형 광고를 제공하는 새로운 방안을 제안하였다. 제안 디자인에 대한 구현을 통해 실제 암호화된 정보를 바탕으로 한 맞춤형 광고가 가능함을 확인하였으며, 20ms 이내의 낮은 오버로드로 실시간성이 요구되는 RTB 환경에 실제 적용 가능함을 함께 확인하였다.

Key Words : Privacy, Real Time Bidding(RTB), Functional Encryption

ABSTRACT

Online advertisements are delivered through real-time bidding(RTB) and behavioral targeting of users. The user profile sent in the ad request contains data that infringes on user privacy and is propagated and stored throughout the RTB network by the actors seeking to increase their performance and profitability. In this paper, we propose to leverage Functional Encryption(FE) to preserve user privacy. Our implementation showed that it can be integrated to the current RTB ecosystem and provide real-time(< 20ms) privacy-preserving advertising with minimal changes.

Ⅰ. 서 론

온라인 광고시장은 연간 수백억 달러 수준의 시장 규모로 사용자가 무료로 웹 컨텐츠를 사용할 수 있게 하는 웹 생태계의 가장 큰 경제 동력이다. 온라인 광고는 광고주들이 자사의 홍보를 효과적으로 전달하는 대표적인 마케팅 방법 중 하나이며, 웹사이트에 접속한 사용자에 맞게 적합한 광고를 노출하는 맞춤형 광고가 널리 쓰이고 있다. 맞춤형 광고는 Real-Time Bidding(RTB)으로 대표되는 실시간 입찰 방식으로 제공된다[1]. 현재의 RTB 환경에서는 사용자의 관심사나 나이, 성별 등의 개인정보가 광고주 측에 전달되며 광고주는 이를 이용해 잠재적인 고객 여부를 판단하고 실시간 광고 입찰에 참여한다.

그러나 최근 개인정보 보호의 중요성이 증대되면서 광고를 위한 사용자의 개인 정보 유출에 대한 우려가 제기되고 있다. 이에 따라 Google과 같은 플랫폼 제공자는 third party cookie 차단, Privacy Sandbox와 같은 정책으로 사용자의 정보를 보호하는 조치를 취하고 있다. 이러한 제약으로 인해 기존 광고 생태계는 광고 수익에 직접적인 타격을 입고 있다. 이를 극복하기 위한 대안으로 Google의 Topics, FLEDGE와 같은 플랫폼 주도의 방식을 비롯하여 동형암호와 같은 암호학적 접근을 통한 방안들이 함께 제안되고 있다. 하지만 플랫폼 주도 방식의 경우 광고 생태계의 다른 참여자들의 반발이 존재하며, 동형암호 방식[2]의 경우 연산이 매우 오래 걸릴 뿐만 아니라 기존의 RTB 환경에 큰 구조적 변경이 불가피하여 실제 적용에는 많은 어려움이 있다.

본 논문에서는 기존의 RTB 환경을 기반으로

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2023-2021-0-02048)
◆ First Author : Seoul National University Department of Computer Science and Engineering, eradegus@snu.ac.kr, 학생회원
° Corresponding Author : Seoul National University Department of Computer Science and Engineering, tkkwon@snu.ac.kr, 정회원
* Seoul National University Department of Computer Science and Engineering, wayles@snu.ac.kr, 학생회원
논문번호 : 202302-020-B-LU, Received January 31, 2023; Revised February 20, 2023; Accepted February 20, 2023

Functional Encryption(FE)[3] 을 도입하는 디자인을 제안한다. FE는 암호화(encryption)된 데이터의 복호화(decryption) 없이 지정된 연산을 수행하고 결과를 얻는 방식이다. 이러한 특징으로 COVID-19 확진자의 카드 결제 정보와 같은 민감한 개인정보의 노출없이 이를 이용해 감염 의심자를 찾아내는 등 다양한 연구에 활용되고 있다[4].

Ⅱ. 연구배경

2.1 Real-Time Bidding(RTB)

온라인 광고 생태계인 RTB의 주요 참여자는 다음과 같다.

- Advertiser: 비용을 지불하고 사용자에게 광고를 프로모션하기를 원하는 기업 및 기관.
- Publisher: 광고공간을 Advertiser에게 판매하고 그 수익을 얻는 웹사이트.
- User: Publisher 사이트에 방문하는 사용자.
- SSP(Supply Side Platform): User 정보를 RTB 네트워크에 제공하며, 광고 수익을 Publisher에 제공.
- DSP(Demand Side Platform): Advertiser로부터 광고 입찰 결정을 위임받아 입찰에 참여.
- ADX: SSP와 DSP를 연결한다. 실시간 경매를 진행하고 win bid 결과를 SSP에게 전송.

사용자가 광고 영역이 있는 웹사이트에 접속하면 RTB를 통해 실시간 광고 입찰이 이루어지고, 최종 낙찰 광고가 사용자에게 노출된다. 이때 그림 1에서와 같이 RTB 네트워크에 사용자 정보(User Profile)가 전달되며, 이는 입찰 판단과 금액 결정에 활용된다. 전달되는 정보에는 웹사이트의 주제, 접속 단말의 종류, 광고 공간의 크기, 사용 언어, 위치, 더 나아가 사용자의 성별, 나이, 관심사 등 사용자 정보에 해당하는 민감정보가 다수 포함되어 사생활 침해를 발생시킨다.

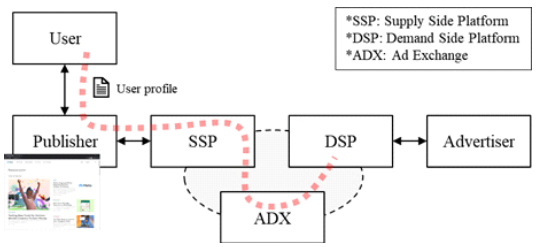

그림 1. 기존 RTB에서의 사용자 정보
Fig. 1. User profile in the current RTB

2.2 Functional Encryption(FE)

FE는 암호화된 데이터를 원본 데이터로 복호화 하지 않은 채 사전 정의한 연산을 수행하고 그 연산의 수행 결과를 얻어내는 방식이다. 암호화된 데이터에 연산을 수행한다는 점에서 다른 차세대 암호인 동형암호와 유사하며, function을 미리 정의하고 얻어낸 function secret key의 역할과 그 최종 연산 수행 결과는 평문이라는 점에서 차이가 있다. 본 논문에 사용된 FE 구조를 그림 2에서 나타낸다. 데이터 소유자 *Alice*, 데이터 사용주체인 *Bob* 그리고 키의 생성 및 관리를 담당하는 Trusted Third Party(TTP) 역할의 *Tom* 으로 정의하며, 본 연구에서 FE의 동작은 다음과 같다.

1. Create master public (mpk, msk) and secret keys
2. Derive function secret key (sk_f) for the function (f) using the msk
3. $c \leftarrow enc(mpk, x)$
4. $y = f(x) \leftarrow \mathrm{dec}(sk_f, c)$

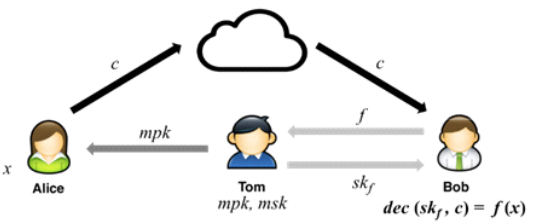

그림 2. TTP를 포함한 Functional Encryption
Fig. 2. Functional Encryption with TTP

Ⅲ. 제안 디자인

전달되는 데이터 중 사용자 프로파일은 그림 3과 같이 암호화된 데이터로 광고 생태계에 전달된다. 이 데이터의 사용 주체인 DSP는 광고와 user의 적합여부를 판단하는 function을 정의하고 대응하는 sk_f를 이용해 암호화된 사용자 프로파일에 연산을 수행한다. 본 디자인에서는 FE의 여러 스킴 중 내적 연산(inner product)을 사용하였다. 내적 연산은 길이가 같은 두 벡터에 대하여 아래와 같이 정의된다.

$$x = (x_1, \ldots, x_n), \qquad y = (y_1, \ldots, y_n)$$
$$\langle x, y \rangle = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \tag{1}$$

복호화 연산은 DSP가 진행하며, 그 결과는 평문 형태

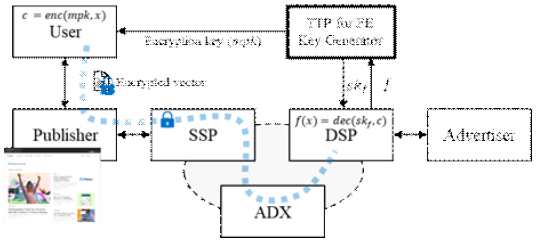

그림 3. FE를 적용한 RTB에서의 정보 전달
Fig. 3. Proposed RTB design using FE

의 사용자 프로파일이 아닌 사용자 프로파일 벡터와 function 벡터의 내적 연산 값이 된다.

3.1 사용자 프로파일(User profile)

기존 RTB 광고 요청에서 전달되는 사용자 프로파일은 json 형태의 평문으로 되어 있다. 본 디자인에서는 RTB의 Audience Taxonomy[5] 인덱스 정의를 기반으로 사용자 프로파일을 0과 1로 구성된 1,679 길이의 bit 벡터로 quantization한다. 사용자의 브라우저는 현재 사용자가 벡터의 각 인덱스에 해당하는 속성과 일치하는 경우 해당 값을 1로 채워 벡터를 완성한다. 예를 들어 위 인덱스 정의에 따르면 "30대 초반 미혼 남성"의 사용자 프로파일에 해당하는 인덱스 정보는 "30-34세[6], 남성[50], 미혼[162]" 과 같다. 해당 인덱스를 1로 채우고 나머지는 0으로 채워 완성한다.

이렇게 생성된 사용자 프로파일 벡터는 mpk를 이용해 암호화하여 보관하고 광고요청시 기존 평문 형태의 사용자 프로파일 대신 전달한다.

3.2 Function

광고 입찰에 참여하는 DSP는 노출하고자 하는 광고의 타겟 고객을 미리 정의하고 사용자 프로파일 벡터와 동일한 인덱스 정의를 기반으로 function을 구성한다. 여기에서 function은 타겟으로 하는 인덱스의 값에 정수 범위의 원하는 가중치 값을 부여한 weight 벡터 형태이며 따라서 DSP는 광고의 타겟층에 따라 다르게 작성한다. DSP는 이를 TTP에 제출하여 sk_f를 얻는다. 이 sk_f와 광고 요청 발생 시 전달받는 사용자 프로파일 벡터로 복호화 하면 두 벡터의 내적 연산 결과를 얻게 된다. DSP들은 이 값을 업체들마다 내부적으로 가지고 있는 입찰 알고리즘에 유저 적합도 점수로서 활용하고 그 결과를 통해 실시간 광고 입찰에 참여한다.

Ⅳ. Prototype 및 성능 평가

4.1 Prototyping

프로토타입에서는 publisher 역할의 웹페이지를 구현하고 간결한 형태의 user를 임의로 생성하도록 하였다. 또한 세 종류의 광고(자동차, 육아, 여행)를 보유한 3개의 DSP 서버를 구성하였고 각 광고에 적합한 대상고객을 가정하여 그에 맞는 가중치 벡터 function을 작성하였다. FE 관련 코드는 오픈소스 라이브러리인 CiFEr[6]를 사용하였으며, DSP는 연산을 통해 얻은 user 적합도에 비례하여 높은 금액을 입찰하도록 구현하였다. 그림 4는 축약된 형태로 구현된 프로토타입이며, 임의의 사용자와 낙찰된 광고를 나타낸다. json 형태의 사용자 정보는 사용자 프로파일 벡터로 구성되고 암호화되어 전송된다. 그림의 상황에서 DSP들은 사용자가 남성인지 여부를 모른 채 입찰에 참여하였고 남성고객을 타겟으로 하였던 자동차 광고가 다른 DSP 보다 상대적으로 높게 입찰하여 최종적으로 해당 광고가 제공되었다.

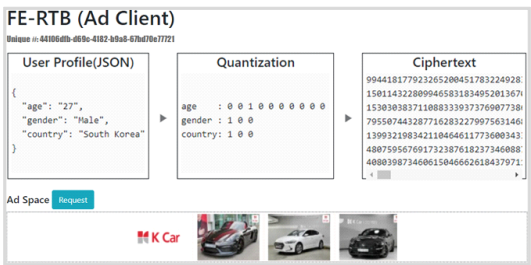

그림 4. 가상의 유저 기반 프로토타입 구현
Fig. 4. Prototyping using simulated user

4.2 성능 분석

RTB 환경에서는 일반적으로 광고 요청 시점으로부터 100-1,000ms 이내에 광고 입찰이 이루어져야 한다. 제안한 디자인에서 추가된 FE 연산 중 key의 생성과 암호화 연산은 광고 요청 이전 시점에 수행된다. 반면 DSP의 복호화 연산은 광고 요청 시점 이후로 수행되므로 실시간성을 필요로 한다. 따라서 성능상의 오버로드를 확인하기 위해 벡터의 크기 별 복호화 연산에 소요되는 시간을 측정하였다. 라이브러리에서 제공하는 IPFE의 두 가지 대표 스킴인 DDH와 LWE를 AWS EC2 c5.4xlarge 환경에서 측정하였으며, 그림 5와 같이 최대 벡터 크기인 2,000을 기준으로 각각 20ms, 5ms의 매우 낮은 오버로드로 디자인 적용이 가능함을 확인하였다.

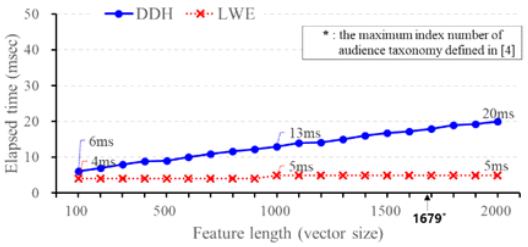

그림 5. 복호화 시간 측정 결과
Fig. 5. Elapsed time of FE decryption

V. 결 론

본 논문에서는 Functional Encryption을 온라인 광고 환경에 도입함으로써 프라이버시를 보호하면서 맞춤형 광고를 제공하는 새로운 방안을 제안하였다. 제안된 디자인에서 암호화된 사용자 정보를 이용해 맞춤형 광고가 가능함을 확인하였다. 성능상의 오버로드 또한 실시간 광고 환경에 충분히 적용가능한 범위에 있음을 확인하였다. 향후 후속연구로 본 연구에서 다루지 않은 여러 FE 방식 중 RTB 환경에 가장 적합한 스킴을 찾는 연구를 진행해 볼 수 있을 것이다.

References

[1] J. Wang, Z. Weinan, and S. Yuan, "Display advertising with real-time bidding(RTB) and behavioural targeting," in *Foundations and Trends® in Information Retrieval*, vol. 11, no. 4-5, pp. 297-435, Jul. 2017. (http://dx.doi.org/10.1561/1500000049)

[2] E. Deng, et al., "Pri-RTB: Privacy-preserving real-time bidding for securing mobile advertisement in ubiquitous computing," in *Inf. Sci.*, vol. 504, pp. 354-371, Dec. 2019. (https://doi.org/10.1016/j.ins.2019.07.034)

[3] D. Boneh, S. Amit, and W. Brent, "Functional encryption: Definitions and challenges," in *TCC 2011*, pp. 253-273, Providence, USA, Mar. 2011. (https://doi.org/10.1007/978-3-642-19571-6_16)

[4] H. Lee, S. Park, I. Baek, W. Kim, and Y. D. Chung, "Finding COVID-19 possible cases based on card payment data: A method without privacy breaches using functional encryption techniques," in *Proc. KIISE*, vol. 2020, no. 7, pp. 1679-1681, Jul. 2020. (http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09874886) (https://doi.org/10.1371/journal.pone.0242758)

[5] IAB, *Audience Taxonomy*, Retrieved Jan. 28, 2023, from https://iabtechlab.com/standards/audience-taxonomy

[6] FENTEC, *CiFEr - Functional Encryption library*, Retrieved Jan. 28, 2023, from https://github.com/fentec-project/CiFEr