공학석사 학위논문

FAdE: Privacy-preserving Targeted Advertising System using Functional Encryption

Functional Encryption을 이용한 프라이버시 보호 온라인 타겟 광고 시스템

2023 년 2 월

서울대학교 대학원

컴퓨터공학부

이 재 현

Functional Encryption을 이용한 프라이버시 보호 온라인 타겟 광고 시스템

FAdE: Privacy-preserving Targeted Advertising System using Functional Encryption

지도교수 권 태 경

이 논문을 공학석사 학위논문으로 제출함

2022 년 11 월

서울대학교 대학원

컴퓨터공학부

이 재 현

이 재 현의 공학석사 학위논문을 인준함

2022 년 12 월

위 원 장	이 광 근
부위원장	권 태 경
위 원	허 충 길

Abstract

FAdE: Privacy-preserving Targeted Advertising System using Functional Encryption

Jaehyun Lee
Department of Computer Science
The Graduate School
Seoul National University

As interest in protecting user privacy began to surge, the online advertising industry, a multi-billion market, is also facing the same challenge. Currently, online ads are delivered through real-time bidding (RTB) and behavioral targeting of users. This is done by tracking users across websites to infer their interests and preferences and then used when selecting ads to present to the user. The user profile sent in the ad request contains data that infringes on user privacy and is delivered to various RTB ecosystem actors, not to mention the data stored by the bidders to increase their performance and profitability. I propose a framework named **FAdE** to preserve user privacy while enabling behavioral targeting and supporting the current RTB ecosystem by introducing minimal changes in the protocols and data structure. My design leverages the functional encryption (FE) scheme to

preserve the user's privacy in behavioral targeted advertising. Specifically, I introduce a trusted third party (TTP) who is the key generator in my FE scheme. The user's profile originally used for behavioral targeting is now encrypted and cannot be decrypted by the participants of the RTB ecosystem. However, the demand-side platforms (DSPs) can submit their functions to the TTP and receive function keys. This function derives a metric, a user score, based on the user profile that can be used in their bidding algorithm. Decrypting the encrypted user profiles with the function keys results in the function's output with the user profile as its input. As a result, the user's privacy is preserved within the RTB ecosystem, while DSPs can still submit their bids through behavioral targeting. My evaluation showed that when using a user profile bit vector of length 2,000, it took less than 20ms to decrypt the encrypted user profile and derive the user score metric through the inner-product function. This is much smaller than my criteria of 50ms, which is based on the typical bidding timeframe (100–1,000ms) used in the ad industry. Moreover, my result is smaller than the state-of-the-art privacy-preserving proposals using homomorphic encryption or multi-party computations. To demonstrate the potential for real-world deployment., I build a prototype implementation of my design that consists of a publisher's website, an ad exchange (ADX), the DSP, and the TTP.

Keywords: Online Advertising, Real-time Bidding (RTB), Functional Encryption (FE), User Privacy, Encryption

Student Number: 2021-24027

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online advertising is a $189 billion industry [1], and a primary revenue stream for publishers. By allowing advertisers to promote their products and services to potential customer form the Internet, it enables users to enjoy most web contents free of charge. To most tech giants that provide web services, online advertising revenue is the main source of income. According to Statista [2], Meta and Google, rely on advertising for about 97% and 81% of their sales, respectively, and Apple recently announced the expansion of its advertising business [3].

Online advertising exists in numerous forms and methods. Traditionally, advertisers would buy certain keywords to promote their products and services when a user enters that keyword on a search engine, this is also known as *sponsored search*. Another form of advertising is *contextual advertising*, here, publishers display ads that are relevant to their online contents, which presumably is linked to the interest of visitors. Publishers sell ad spaces on their webpages to the advertising network, and advertisers use rich media environments such as video, audio, and interactive contents that utilizes the visitor's

device, geographical location, and more. Lastly, there is *behavioral advertising*, or *targeted advertising*, where the ads are based on the user's profile. A *profile* consists of the user's personal information, browsing behavior, unchecked out carts from online stores, and more. These are collected throughout the user's online browsing lifespan through cookies and user tracking.

In recent years, with growing emphasis on user privacy protection, concerns have risen regarding storing and sending user profile for advertising purposes. Regulations such as General Data Protection Regulation (GDPR) [4]] and California Consumer Privacy Act (CCPA) [5] are beginning to enforce the web service industry and in some cases tech companies were fined for not complying with privacy protection regulations [6, 7]. In response, companies such as Google and Apple are taking measures to protect user information in ways such as Privacy Sandbox [8] and ATT [9], going even further to blocking third-party cookies from their browsers. This resulted in severe cut in revenue [10, 11] since storing and utilizing user's private data is closely linked to the performance and profitability of their ad revenue stream.

My research focuses on preserving user's privacy in targeted advertising, while maintaining the current online advertising ecosystem that is based on Real-time Bidding (RTB) *targeted advertising*, while maintaining the current online advertising ecosystem that is based on Real-time Bidding (RTB) [12]. RTB, which will be described in detail in Chapter 2, is a mechanism used in both behavioral and contextual advertising. It occurs on-demand when the publisher requests an ad, and an auction is held for the targeted user/webpage. The highest bidding advertiser gets to post his or her ad as shown in Figure 1.1. The RTB ecosystem unleashed a battlefield where the advertising platforms used user data to their full potential to select ads that are highly targeted and personalized. With modern machine-driven algorithms and automations, optimizing profit through precise user targeting is crucial to their business,

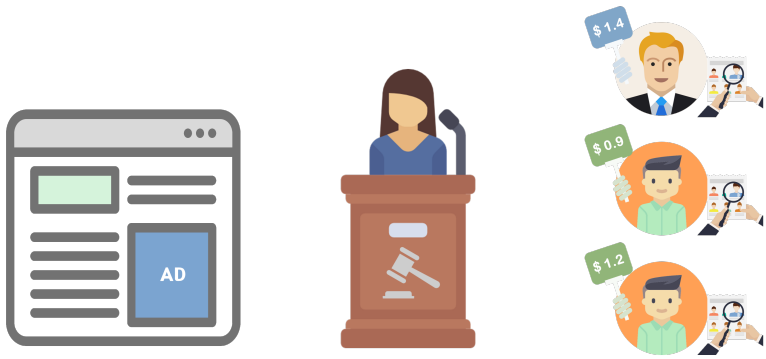


Figure 1.1: A concept of Real-Time Bidding

which lead to more user tracking and syncing [13].

In this thesis, I propose a framework named **FAdE** to preserve user privacy that is leaked and unknowingly spread through the ad network by applying functional encryption to user *profile* in *Ad/Bid Requests*. Functional encryption (FE), along with homomorphic encryption (HE), has attracted increasing attention and interests as of recently. For example, there is a study that uses FE to predict breast cancer while protecting sensitive medical information of an individual [14], and a study that uses FE to trace contact for COVID-19, where the suspected contacts of infected patients can be retrieved without privacy breaches [15]. Similar to HE, FE also evaluates a function over an encrypted data but differs in terms of key management. The public and private key pairs in HE are generated by the owner of the data. However, in FE, the keys (i.e., public key, private key, and function keys) are generated by a trusted third-party (TTP) who is trusted by the participants of the network. Another key difference between FE and HE is that the former computes the plaintext result of $f(x)$ given the encrypted data $enc(x)$, while the latter computes the encrypted result of $f(x)$ which needs to be decrypted from the data owner through the previously issued private key. In other words, the actor in the

world of HE performing the computation on the encrypted data is unable to learn the result. However, the actor embracing FE can obtain the result while preserving user privacy.

The contributions of this work are:

- It is said that behavioral targeting conflicts with privacy. In other words, if privacy is to be preserved then targeted advertising becomes unfeasible. However, I propose a framework named FAdE where targeted advertising is supported without compromising user privacy using modern cryptography, i.e., functional encryption.
- FAdE does not aim to replace the existing online advertising infrastructure based on RTB and the OpenRTB protocol, but rather enhance it by preserving privacy of users and at the same time providing the same level of behavioral targeting.
- I also define criteria for time and file size so that my proposal of using FE with RTB is applicable for real-world usage. My evaluation shows the feasibility and scalability of the proposed framework.

The rest of this thesis is organized as follows. Chapter 2 presents an in-depth overview of online advertisement including the main actors/entities and notions involved, as well as the OpenRTB protocol. Also, I introduce functional encryption, the main cornerstone of the framework, and layout the preliminaries to better understand my design. In Chapter 3, I describe FAdE and its detailed components, as well as modifications to the OpenRTB protocol for FAdE to work alongside the current ad network. Chapter 4 presents my evaluation and findings from exploring various functional encryption schemes as well as my prototype implementation. Chapter 5 reviews related works on privacy preserving online advertising compare to FAdE. Finally, I summarize my thesis in Chapter 6.

Chapter 2

Background

In this chapter, I introduce online advertising and functional encryption (FE) to help explain my proposed framework. Specifically, I give an overview of real-time bidding (RTB) and explain the key players in the ecosystem and then illustrate the bidding and ad delivery mechanism. Next, I present an overview of FE, its difference compared with fully homomorphic encryption (FHE), information leakage, and then I introduce inner product functional encryption schemes which FAdE is built upon.

2.1 Online Advertising

Online advertising has been around for more than a decade. Starting from sponsored search which allowed advertisers to buy certain keywords to promote their services or products when users searched for such terms. It has greatly contributed to search engines to provide their services for free [13]. Online advertising can be also found in publishers' websites in the form of

contextual advertising where publishers sell blocks of space on their webpages, and contextual advertising platforms provides richer media (e.g., video, audio) advertisements based on the context of the webpage or the application that is requesting the ad. To increase the user's interest on the displayed ad and impose further actions such as visiting the advertiser's website or finalizing a purchase order, advertising platforms now use techniques known as behavioral targeting. Intuitively, users' information and identifier are delivered to the advertising platforms where in conjunction with user tracking and cookie synchronizing methods, their interests are inferred from data such as visited web pages, search queries, and online purchases. Based on these data, advertisements are selected through campaign-specific machine learning models that predict users' responses to advertisements. In order to increase bid profitability and accuracy of user targeting, the advertising industry focused on collecting more and more user data which raised concerns of user privacy.

2.1.1 RTB Ecosystem

Real-Time Bidding (RTB) facilitates real-time auctions of advertising space [16] through marketplaces called AD eXchanges (ADX), allowing buyers to determine bid values for individual ad impressions. Since advertisers may not have the expertise to accurately estimate impression values using machine learning models, Demand Side Platform (DSP) supports advertisers and ad agencies by bidding for their campaigns. Similarly, Supply-Side Platform (SSP) supports publishers in optimizing their yield.

The main actors of RTB shown in Figure 2.1 is as follows:

- **Advertiser**: A company or institution that wants to pay for and promote advertising to users in an RTB environment.
- **Publisher**: A website or application that sells advertising space to ad-

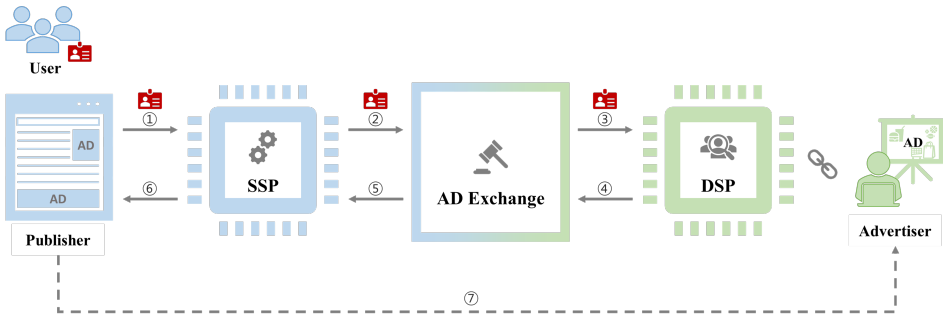


Figure 2.1: Structure of the real-time bidding protocol

vertisers and earns the profits.

- **User**: End user who visits the publisher's site. When a user accesses a publisher's site including an advertisement space, a series of RTB processes are performed, and the advertiser's advertisement is finally exposed to the user, and the user obtains promotional information from the advertiser by viewing or clicking the advertisement.

- **SSP**: The Supply Side Platform provides services to publishers by registering their inventories (impressions) from multiple ad networks and accepting bids and placing ads automatically. The SSP also collects user information and provides it to the RTB network.

- **DSP**: The Demand Side Platform provides services as an agency for advertisers. Advertisers entrust the DSP to decide on advertisement targets and whether to bid for advertisements. That is, DSP determines the bid amount for each ad request and bid.

- **AD Exchange**: The AD eXchange combines multiple ad networks together [17]. In essence it connects SSPs and DSPs, providing DSP with

information necessary for bidding. Next, it collects biddings from multiple DSPs and determines the winning bid which is delivered to the SSP.

Figure 2.1 also illustrates how RTB works with behavioral targeting, starting with a user visiting a website, a bid request sent, to the display of the winning ad. ① & ② Publisher and SSP code makes a bid request to an ad exchange. Bid request includes details about the ad placement as well as details about the user. ③ Ad exchange forwards the request to multiple DSPs. DSPs, using information from the publisher and of the user will place a bid to show their ad. The bid can come from any advertiser active on the DSP at the time of the bid. ④ After some time, the ad exchange will take the bids it has received and select a winner based on exchange and publisher bidding rules. ⑤ & ⑥ The winning ad is sent to the user's browser and loads from the user's browser. ⑦ The ad server which served the ad records this using data from the browser.

2.1.2 OpenRTB

OpenRTB is the communication protocol that enables real-time bidding. It was a pilot project [18] created in 2010 by three DSPs (DataXu, MediaMath, Turn) and three SSPs (Admeld, PubMatic, the Rubicon Project). It was designed to spur growth in RTB marketplaces by providing an open industry standard for communication and interoperability between buyers and sellers in the digital advertising industry [16].

Bid Request

When a user visits the publisher site/application, the publisher creates and transmits an Ad Request using the information required for advertisement bidding. In response to each Ad Request, a Bid Request is broadcast to multiple DSPs (bidders). An example bid request can be found in Appendix A.1. Here,

various information that can be used by the DSP in making an advertisement bidding decision is delivered. These include: the current website address (line 35), the size of the advertising space (lines 16-17), the information of the user (lines 49-75), and the device (lines 45-48).

Bidding Process

The bidding process starts from the time the user accesses the publisher's site including the advertising space. SSP transmits Bid Request to DSP through ADX. The DSP receives the Bid Request and determines whether the corresponding user is a suitable user for the advertisement they want to provide. The bidding function is performed using the context information of the publisher site, the user's profile, and information such as the type and size of the advertisement space, and as a result, it determines whether to bid for the bid request and the bid price to participate in bidding. After the DSP receives the bid request, a series of processes participating in bidding generally takes place within several hundred ms [19, 20]. RTB Exchange (ex. ADX) receives bids from multiple DSPs and determines win bid with high bid prices. Win bid is delivered to the user along with a link to the advertisement.

2.2 Functional Encryption

Functional encryption (FE) is a generalization of the traditional public key cryptography that enables a fine-grained access control for encrypted data [21]. In this section, I first give an overview on FE. Then, I discuss major differences between FE and fully homomorphic encryption (FHE) as well as information leakage. Finally, I present the inner-product functional encryption (IPFE), which is the main cryptographic primitive I use in FAdE. In particular, I introduce two IPFE algorithms based on two computational hardness

assumptions. Namely, the Decisional Diffie-Hellman (DDH) assumption and the Learning with Errors (LWE) assumption.

2.2.1 Overview of FE

FE is a generalization of the traditional public key cryptography, in which the public key is used to encrypt a plaintext and create a ciphertext, and the matching private key is used to decrypt the ciphertext and retrieve the plaintext. Similarly, in FE, the public key, or the master public key (MPK), is also required to encrypt plaintexts. However, instead of the private key, FE offers the possibility to partially decrypt ciphertexts with fine-grained control through the use of function keys or secret keys.

FE requires a new actor, the key authority or key generator, who is responsible for generating the keys. In this thesis, I assume that a trusted 3$^{\text{rd}}$ party (TTP) takes this role, which is depicted as Tom in Figure 2.2. Tom is trusted by Alice and Bob and generates/manages the master secret key (MSK) and the MPK. Alice is the owner of the data (e.g., health data that includes not only her height and weight but also other sensitive medical information) that requires privacy preservation, and Bob uses Alice's data to provide services such as a diet program based on her Body Mass Index (BMI). However, by doing so, Bob gains sensitive other sensitive data in the process. In FE, Alice encrypts her medical data, and unlike public key cryptography, when Bob decrypts the data using the secret key provided by Tom, the result is not Alice's medical data, but only the BMI value of Alice which Bob actually needs. The MSK is used to derive the secret key (sk_f) that is associated with a function f through the key generation algorithm. As shown in Figure 2.2, Tom sends the MPK to Alice, and she encrypts her data/message x and gets c, the encryption of the data/message x. On the other hand, Bob sends the function that he would typically use on Alice's data to Tom. Here, Tom uses his MSK and derives the

secret key sk_f. The secret key is sent back to Tom, and When Tom receives c from Alice, he computes the decryption algorithm with sk_f and c as inputs, which gives $f(x)$ as an output. Due to this property of "partial decryption" or giving an unencrypted output of an evaluation of the function f over the original message, this design makes it well suited for purposes such as cloud computing [21] or verifiable computation [22].

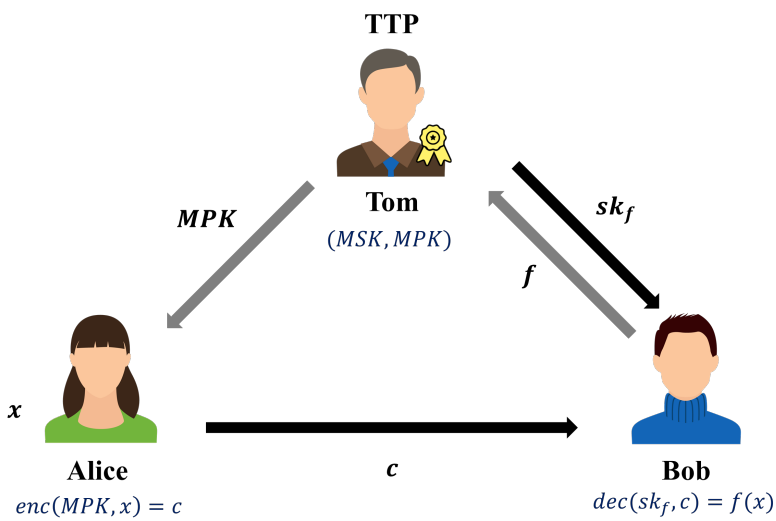


Figure 2.2: An overview of Function Encryption system with a trusted 3rd party as the key authority

2.2.2 Difference between FE and FHE

Alongside FE, fully homomorphic encryption (FHE) [23, 24, 25] is also a generalization of traditional public key cryptography. Both enable to compute algorithms over ciphertexts, however, while the output of the FE's decryption algorithm is unencrypted, the output of FHE remains encrypted. Without requiring an additional trusted authority within the system, FHE works as a traditional public key system with two additional algorithms: addition and multiplication of ciphertexts. The former takes two ciphertexts c_0 and c_1 from

corresponding plaintexts m_0 and m_1, and outputs a new ciphertext c_{0+1}, which is the encryption of $m_0 + m_1$. Similarly, the latter outputs $c_{0\times1}$, which is the encryption of $m_0 \times m_1$. In other words, in FE, the system/actor who computes the algorithm over the ciphertext have access to the results, while the actor/system of FHE only computes and never have access to the data.

Regarding the keys used in the two systems, FHE follows the traditional public key cryptography with a public key and a secret key created by the data owner. But in FE, the aforementioned authority or the TTP creates the MSK, MPK pair, and the MSK is used to generate the secret keys associated with some functions which is then distributed to the function owners.

2.2.3 Information Leakage in Functional Encryption

FE provides partially decryption of ciphertexts with fine-grained control compared to the all-or-nothing decryption in traditional public key cryptography. Given a plaintext, it is possible to build functions that partially reveals the original message, and when combined as a whole, reveals the plaintext. Also, when different ciphertexts encrypted with different MPKs, have identical outputs when computed with some set of secret keys. I can presume that the original plaintexts share some similarities or even identical.

Prevention of information leakage is out-of-scope from the perspective of this thesis. [26] and [27] have analyzed information leakage in FE, and recent research such as [28] proposes leakage-resilient IPFE scheme. I identify the TTP to be responsible for inspecting, regulating, and setting up the guidelines on the functions sent in to derive the secret keys. So that the function by itself nor the group of functions do not reveal the entire plaintext.

2.2.4 Inner Product Functional Encryption (IPFE)

FE schemes that enable the evaluation of inner products [29, 30] are called inner product functional encryption (IPFE) or inner product encryption (IPE). In IPFE, secret keys are associated with inner product functions. Given v, w vectors, sk_w is the secret key associated with the inner product function, and C_v is the encryption of v. The decryption algorithm with sk_w and C_v as inputs, outputs $\langle v \cdot w \rangle$, the inner product of two vectors. [29] proposed constructions for the inner product encryption schemes satisfying standard security definitions, under well-understood assumptions: the Decisional Diffie-Hellman (DDH) and Learning with Errors (LWE). However, they only proved their schemes to be secure against selective adversaries. [30] upgraded those schemes to provide them a full security, security against adaptive attacks. Its detailed algorithm can be found in Appendix A.2. In this thesis, I focus on the fully secure IPFE under the DDH and LWE assumptions from [30].

Chapter 3

Design

FE can be utilized in AI/ML as in [31, 32]. I present a method to transform user profile to a feature vector which can also be used as an input to a AI/ML model. In the scope of this thesis, I use the inner-product functional encryption (IPFE) scheme on targeted advertising. Figure 3.1 shows the overview of FAdE. And the notations and their corresponding descriptions are outlined in Table 3.1

A trusted 3rd party (TTP) is newly introduced, which is the key authority responsible for generating, managing, and providing the keys used in my scheme. The browser(user) receives the encryption key from the TTP and encrypts user data. The DSP sends its inner-product function to the TTP who derives the function key and sends back to the DSP. By decrypting the encrypted user data with the function key, the DSP could use this result for bidding while preserving user privacy. For the remainder of this chapter, I will first discuss my approach to preserving privacy, then, I explain the setup phase and the workflows executed by the TTP, user browser, and the DSP. Finally, I present the bidding phase of FAdE.

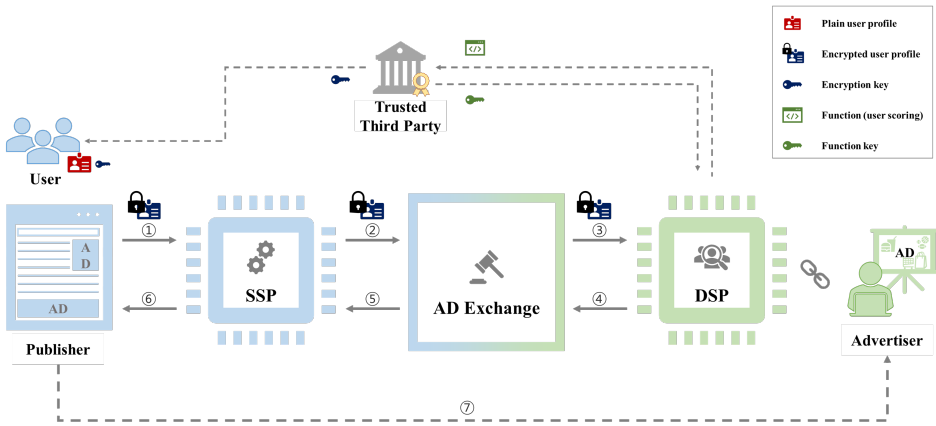


Figure 3.1: Privacy-preserving RTB using Functional Encryption

Notation	Description
MSK	Master secret key
MPK	Master public key (encryption key)
$hash_{enc}$	Hash of encryption key
$func_{\alpha}$	User scoring vector
$feKey_{\alpha}$	Function key (decryption key)
X_{user}	Plain user profile vector
C_{user}	Encrypted user profile (ciphertext)
$score_{user}$	User score result for bidding function
$find\_key()$	Find matched $feKey$ using $hash_{enc}$
$bidding\_function()$	Calculate bid price (trade secret)

Table 3.1: Notations used in this thesis

3.1 The approach to preserving privacy

3.1.1 Encrypted user profile using FE

User profile contains user information as shown in Table 3.2 and Table 3.3, and these were traditionally used for behavioral targeting. I design a user profile in the form of bit vector, that is a group of bits where the values are either true (1)

bit or false (0). The length of the bit vector is set at 2000. Currently, behavioral targeting utilizes few hundreds to thousands of categorized information, of which including user privacy related information. Google defined nearly 6,000 codes for ad targeting [33] and I found that around 600 codes are user related. IAB defined around 1,500 Audience Taxonomy which can be used for user description [34]. My survey on Google Ads API and IAB Audience Taxonomy can be found in Table 3.2 and Table 3.3. Based on my findings, I determined that bit vector of length 2,000 is sufficient for FAdE, and I expand on this in my evaluations.

Category	Description	Examples	# of features
Affinity	Valid affinity categories	Beauty / TV Lovers / Public Transit Users / ...	251
Age ranges	Age ranges	18to24 / 25to34 / 35to44 / ...	7
Genders	Genders	Male / Female / Undetermined	3
Parental status	Parental status values	Parent / Not a Parent / Undetermined	3
Income ranges	Income percentile ranges	Undetermined / 0%-50% / 50%-60% / ... / 90%+	7
Languages	Languages available for targeting	Arabic / Bengali / Bulgarian / Catalan / ...	51
Country	Country codes	Afghanistan / Albania / Algeria / ...	246
Life event	Life event values	Recently Married / Graduating Soon / ...	40
	Total		**608**

Table 3.2: Google Ads API Codes.

Category	Description	Examples	# of feature
Demographic	Quantifiable characteristics of the audience	Age Range(14) / Education(32) / Gender(5) / Household Data(103) / Personal Finance(35) / ...	197
Interest	Medium and long term interests	Academic(36) / Entertainment(26) / Hobbies(35) / Music(41) / Sports(69) / Technology(19) / ...	497
Purchase intent	Current in-market purchase intent	Consumer Electronics(30) / Consumer Packaged Goods(396) / Sporting Goods(31) / Travel (27) / ...	864
	Total		**1558**

Table 3.3: IAB Audience Taxonomy

There is an example of the bit vector. Assume a user [Male, age 33, single, likes football and comedy movies, plans to buy a new car]. Based on IAB Audience Taxonomy, this user can be represent by indexes that corresponds

to the information, 30-34(6), Male(50), Single(162), Comedy Movie(470), Soccer(663), New Vehicles(806). Here, as illustrated in Figure 3.2, I perform binary encoding and create a user profile (bit vector) where the values of the matching indexes are 1 and others 0.

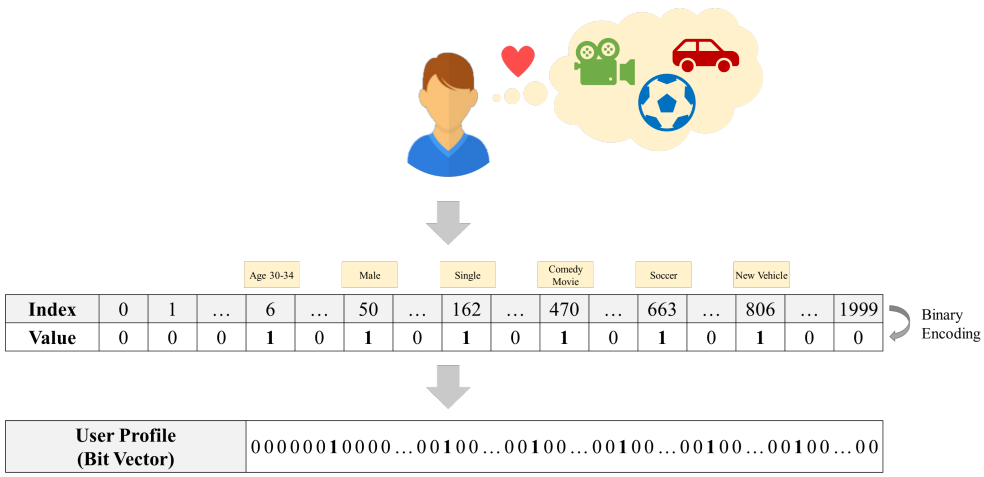


Figure 3.2: Example of Binary Encoding Process for example user

The user profile (X_{user}) is encrypted to C_{user} using the MPK and included in the user object of the bid request as shown in Figure 3.3.

Figure 3.3: Modified user object in Bid Request

3.2 Setup phase

3.2.1 TTP

The TTP, as the name implies, is trusted by the actors of the RTB. Major platforms such as Google, Meta or large Certificate Authorities (CAs) from web public key infrastructure (PKI) could take this role. In FAdE, TTP creates, manages, and delivers the keys used in the IPFE scheme. Its role consists of the following:

- Creation of the pool of master key pairs (MSK, MPK).
- Manages the validity of each key pairs, creates overlapping key pairs, and calculates hash as an identifier for each key pairs.
- Receiving a request from the browser (user), it provides the pool of currently valid MPKs and their hashes.
- When the DSP sends an inner-product function. It derives the $feKey$ using the currently valid MSKs and provides them to the DSP.

3.2.2 User Browser

In current RTB ecosystem, the browser caches the user's information and interests and use them during ad request. When the user loads the publisher's website, the browser executes the embedded code and creates the bid request. During the setup phase of FAdE, the browser (1) retrieves the pool of valid MPKs (2) performs binary encoding of user profile data to create the bit vector.

Key Storing

The retrieval of MPKs can be implemented in following methods. First, during an installation/update or a security patch of the browser, the developer of the

browser packages the current pool of MPKs along with their hashes. Or, the browser can self-monitor the pool of MPKs and request TTPs directly for an updated pool of MPKs. The upper portion of Figure 3.4 depicts the interaction between the user and the TTP.

Encryption

The browser uses the valid list of MPKs to encrypt the bit vector of the user profile, and saves them locally through the process shown in the lower portion of Figure 3.2. If there is a change in the user profile, it could re-encode the bit vector and update the list of encryptions. The encryptions are paired with each MPKs (and their hashes) and are used during the real-time bidding phase.

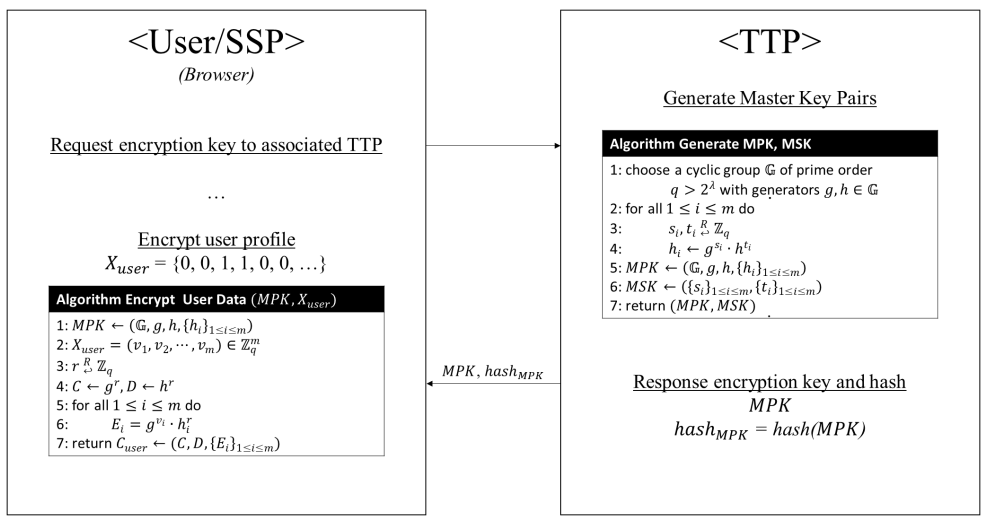


Figure 3.4: Workflow between User(browser) and TTP

3.2.3 DSP

In the setup phase, the DSP creates a *scoring function* (inner-product function) for measuring a *user score* from the bid request, based on the individual ad campaign. The role of the *scoring function* is to find how closely the user

profile matches the target demographic of the ad campaign. Assuming the bit vector length is 2,000, the length of the *scoring function* vector should also be 2,000.

The DSP sends the *scoring function* to the TTP(s), where the $feKey$s are derived. Each $feKey$ should be mapped with the corresponding MPKs hash so that the DSP could select the correct $feKey$ when decrypting the bid request. This process is depicted in Figure 3.5.

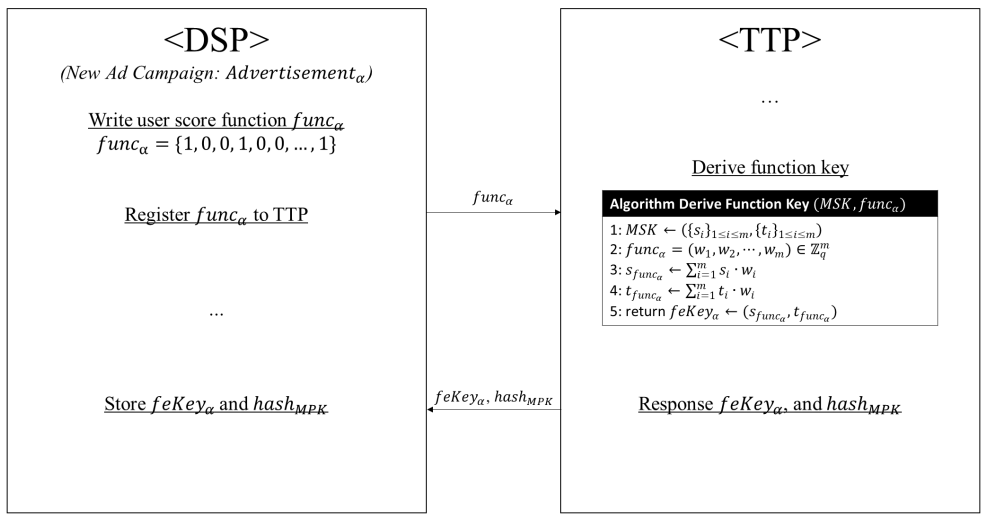


Figure 3.5: Workflow between DSP and TTP

3.3 Bidding Phase

Figure 3.6 presents the complete workflow during the bidding phase. The ADX manages the real-time auction between the browser and the DSPs, which is no different from the current RTB. I describe the actions performed in the browser and the DSPs in detail.

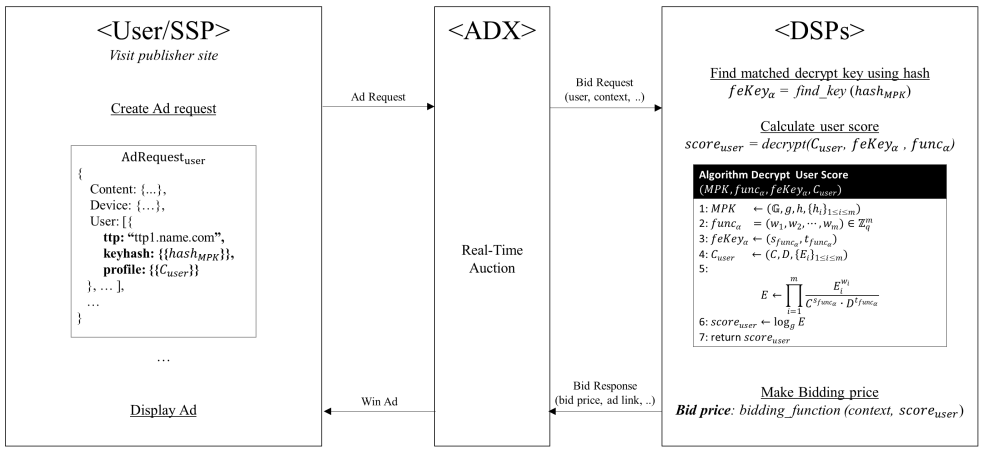


Figure 3.6: Workflow during the Real-time Bidding Process

3.3.1 Browser (User)

When the browser visits the publisher's website that includes an ad space, it creates an *ad request*. The *ad request* includes the *user* object from Figure 3.3. If the publisher's website indicates multiple TTPs, then the browser creates multiple *user* objects with C_{user} that match the provided TTPs information. In addition to C_{user} (*profile*) and *ttp*, the *user* object also includes the $hash_{enc}$ (*keyhash*), so that the DSPs could select the correct $feKey$ to decrypt the C_{user} within the object.

After the real-time auction, the winning bid is determined, and the URL of the ad is delivered to the browser and displayed to the user.

3.3.2 DSP

During the bidding phase, based on the user and contextual information, the DSPs decide on their bidding price. The user information is the aforementioned C_{user}, the encrypted user profile. And the contextual information is those related to the target webpage and details regarding the ad space.

Since, the same user profile could have multiple encrypted ciphertexts, the

DSP uses the *ttp* and *keyhash* to find the correct $feKey$ to decrypt the C_{user}. The result of the decryption is the inner-product between the user's bit vector and the DSP's inner-product function, that is, the user score value. Finally, using the user score and other contexts the DSP determines the bid price which is sent to the ADX for auction.

Scoring function

The DSPs use the encrypted user profile in the *bid request* to calculate the user score. In FAdE, the user score gives some idea of similarity between the user and the target demographic of the ad campaign. Using this as a parameter for the bidding function, the DSP calculate their actual bidding price. I believe that the specific algorithm for deciding the bidding price is a trade secret for the DSPs. However, I should preserve user privacy so that the user profile is not available to other actors of RTB. The scoring function is the main enabler. By splitting the previous bidding algorithm into two, the foremost part of assessing the user is now done through the scoring function in FAdE, and the latter part of deciding whether to bid and determining the price can still be kept confidential. Note that the scoring function $func_{\alpha}$ is sent to the TTP, where its function key $feKey_{\alpha}$ is derived.

Figure 3.7 shows an example of calculating the user score ($score_{user}$) from the user's bit vector and two different scoring functions. The values of the scoring function vector indicate the weight or the level of priority of the matching feature (e.g., gender, interests, ...).

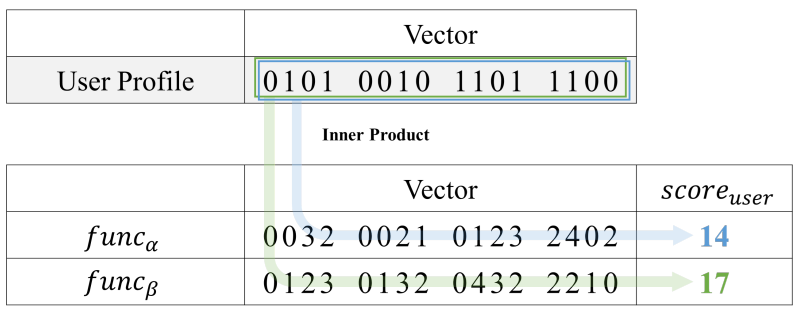


Figure 3.7: Simple example of *user scoring*. Inner product $C_{user} \cdot func$

Chapter 4

Evaluation

In this chapter, I verify the feasibility of the FE scheme to be applied to FAdE. I define criteria suitable for the characteristics of FE RTB and present measurement results of usable FE scheme.

4.1 Criteria

4.1.1 Time

In RTB, in order to support real-time auction, there is a maximum time limit for the DSP to respond to a bid for each AD Network. This is for the purpose of ensuring quick bid participation and quick ad exposure, usually within 100-1,000ms [19, 20]. DSPs perform three main operations within this time limit after receiving the Bid Request [13]. 1) Determination of suitable advertisement type 2) Determination of suitability with users 3) Set final bid and participate in bidding through internal bidding function. In FAdE, as shown in Figure 3.6, the bidding function is performed using the user score obtained

through FE decryption to obtain the final bid price. Therefore, FAdE affects the part that judges suitability with the user. I defined the time criteria as 50ms, which is shorter than the allowable time in general RTB, and confirmed the overload in actual application.

Each of the four sections of FE (Setup, FeKey Derivation, Encryption and Decryption) were measured, and among them, only Decryption, which should be processed in real time for Bid request, is subject to the time criteria of 50ms.

4.1.2 File size

The size of the files created and used in FAdE also affects storage space and transfer time. So I have to distinguish which files are used and consider their size. The purpose and transmission information of each file used in FAdE are as follows.

- MSK: Inside TTP only / No transmission
- MPK: Key for encryption / TTP to User (only once)
- $func$: DSP's scoring function / DSP to TTP (per ad campaign)
- $feKey$: Key for decryption / TTP to DSP (per function)
- X_{user}: Plain user profile / No transmission
- C_{user}: Ciphertext of user profile / User to DSP (per bid request)

Among them, the size of C_{user} transmitted when requesting an advertisement is based on the size that can be transmitted within 10ms assuming a generally fast network between ADX and DSP. To reduce latency and latency volatility, vendors use fast transport environments in ways such as network peering [35]. Therefore, it is assumed that the maximum allowable size is about 12.5 MB based on the 10 Gbps environment.

4.2 Environment

4.2.1 Testbed

The experiments were run in an Amazon c5.4xlarge instance 64-bit Ubuntu server 20.04 which has 32GB of RAM and 16 vCPU 3.00GHz Intel(R) Xeon(R) Platinum 8124M. The result are averaged over 10 runs for each test.
For each test, new input vectors X_{user} and $func$ consisting of single-digit random values were generated. X_{user} represents a user's profile, and $func$ represents a weighting function that is calculated on the user profile to obtain a user score in DSP. The measurement was carried out by increasing it up to the 2,000 vector size, which is FAdE design size. For a specific scheme I tested further to the extent that it exceeded the design size.

4.2.2 FE Library

A FE cryptographic library, named CiFEr [36] by the project "**F**unctional **EN**cryption **TEC**hnologies" (FENTEC for short) is used, which is an open-source encryption algorithm library, which contains the implementations of the functional encryption algorithm proposed by [30]. Two schemes, *cfe_damgard* and *cfe_lwe_fs*, are chosen for the inner product of FAdE. Both are based on paper by [30] and satisfy adaptive security under chosen-plaintext attacks (IND-CPA security). *cfe_damgard* and *cfe_lwe_fs* are schemes that provide full security under the DDH assumptions and LWE assumptions, respectively. From the result section, each schemes are denoted as DDH and LWE.

4.3 Result

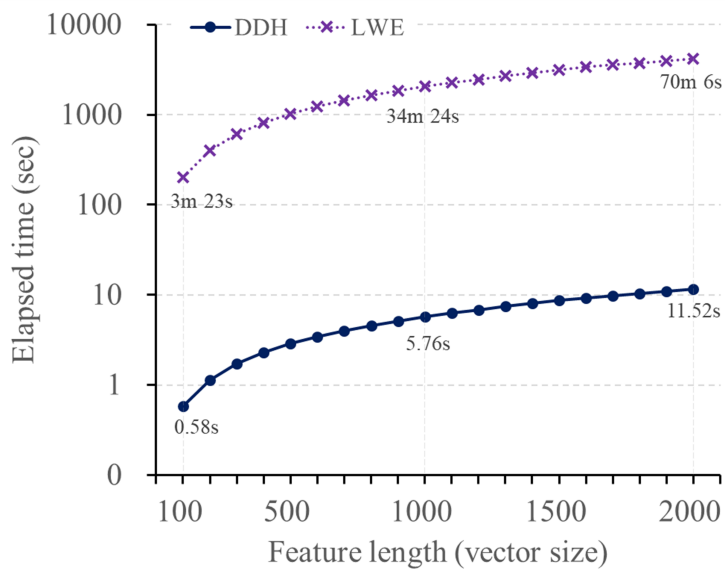
4.3.1 FAdE design

The results measured based on a user profile vector of a maximum size of 2,000 are shown in the Figure 4.1. In all measurements, it was confirmed that the

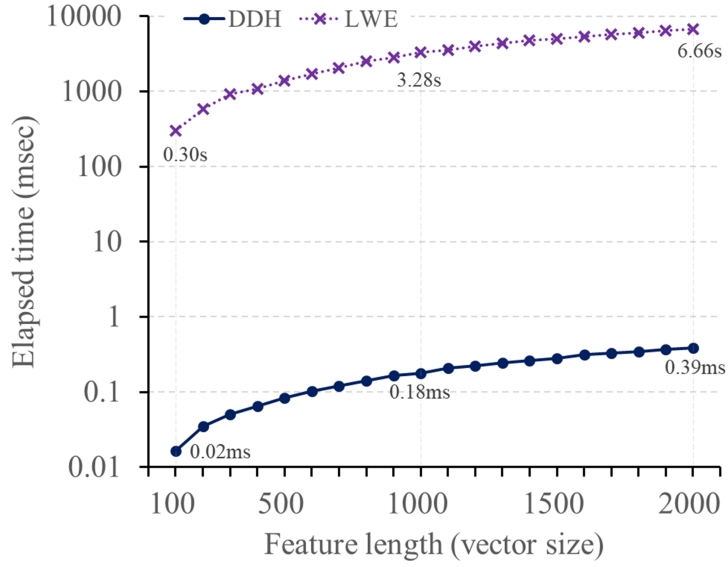
execution time increased linearly with the increase of the vector size. Setup and derivation, which showed a large time difference between DDH and LWE, were expressed in log scale y-axis, and encryption and decryption that did not show a large time difference were expressed in linear scale y-axis.

First, in the case of (a) Setup, based on the 2,000 size, the LWE scheme took 70 minutes and 6 seconds, and the DDH took 11.52 seconds. The Setup section is an operation to create an MSK and MPK pair. It is performed in TTP and is not time critical because it has a low execution frequency (ex. key update). (b) $feKey$ derivation was observed at 6.66s and 0.39ms in LWE and DDH, respectively. This section is an operation performed in the TTP when the DSP requests the TTP to register a new advertising campaign. Similarly, it is not Time Critical, but it is confirmed that DDH is more advantageous than LWE. (c) Encryption is an operation performed by the browser when a change in user profile or update of a key is detected in the user's browser. LWE was measured at 0.53s and DDH at 5.78s. This section is also an operation performed in advance before the advertisement request and is not time-critical. Interestingly, it was confirmed that LWE was processed in a short execution time without much effect on the vector size. (d) Decryption is an operation performed by DSP when a user accesses a publisher's site and requests an actual advertisement. As a section that must satisfy the time criteria of 4.1.1, 20ms for DDH and 5ms for LWE, both schemes confirmed a sufficiently smaller value than my criteria 50ms. In particular, LWE shows results that are not significantly affected by the increase in vector size.

Based on the vector size of 2,000, the file size created in each of the two schemes is the same as Table 4.1. In the result, the size of the file was recorded as a text file after saving the output result using *cfe_vec_print()* and

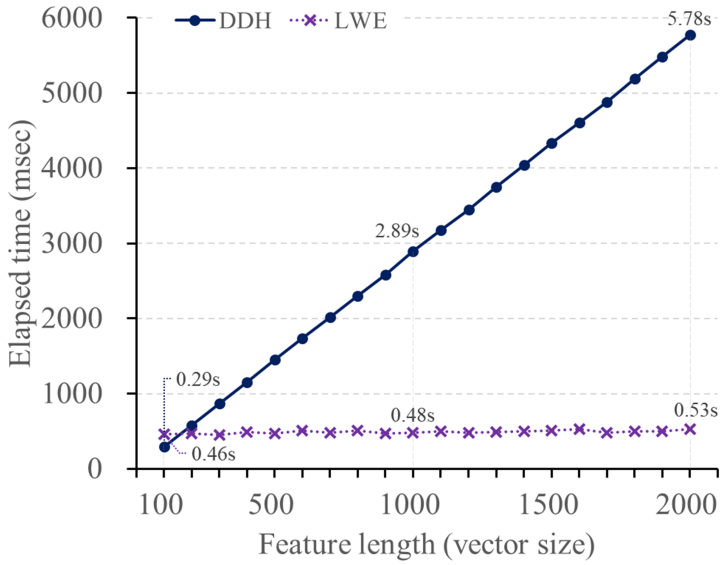


(a) Setup

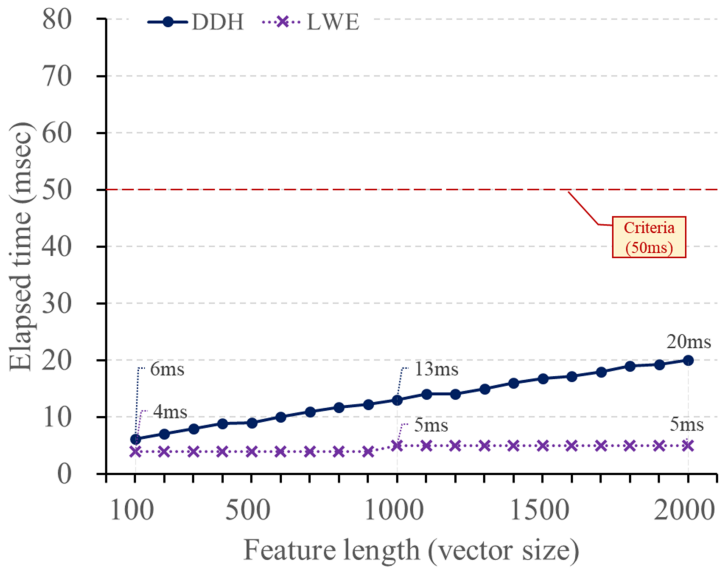


(b) $feKey$ Derivation

Figure 4.1: Vector length from 100 to 2,000



(c) Encryption



(d) Decryption

Figure 4.1: Vector length from 100 to 2,000 (cont.)

	MSK	MPK	$func$	$feKey$	X_{user}	C_{user}
DDH	2.47MB	1.24MB	6.00KB	1.24KB	6.00KB	1.24MB
LWE	1.34GB	29.72MB	6.00KB	2.27MB	6.00KB	3.91MB

Table 4.1: Result of file size when vector length is 2,000

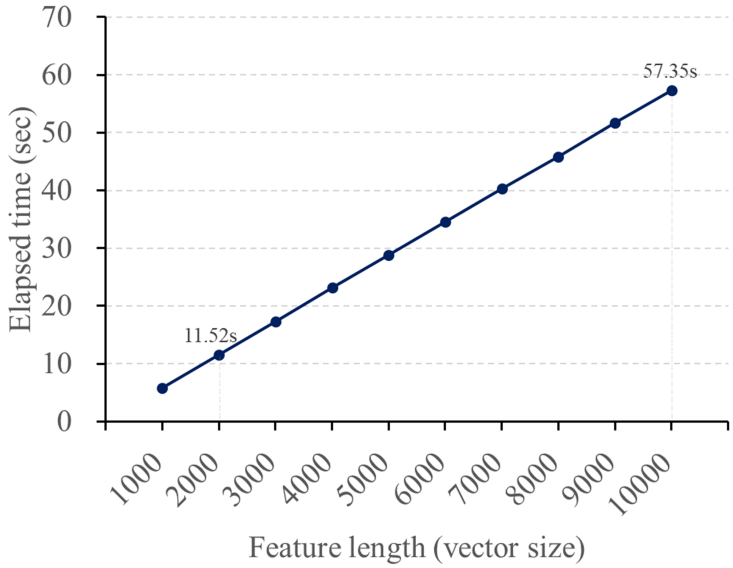
cfe_mat_print() of CiFEr library for the same standard for each scheme.
As shown in the 4.1.2 file size criteria above, data transmitted at the time of real-time advertisement request is only C_{user}, and other files are not transmitted at that time. As in the measurement result, in both schemes, C_{user} increased hundreds of times compared to the existing X_{user}. However, assuming a high transmission speed between ADX and DSP (generally at the level of 10Gbps), both schemes satisfy the criteria with a small size that can be transmitted in less than 5ms.
When measured with FAdE design, a 2000-length vector size, the LWE scheme has a very fast Encryption / Decryption time, but a relatively slow setup time and a large file size. On the contrary, the DDH scheme is relatively slow in Encryption / Decryption, but it is confirmed that it is still within the range of my tight criteria. In particular, in the case of DDH, it was judged that it would be more advantageous in the TTP structure that supports multiple key pairs because the setup time and the file size to be generated and transmitted are small.

4.3.2 Extra test

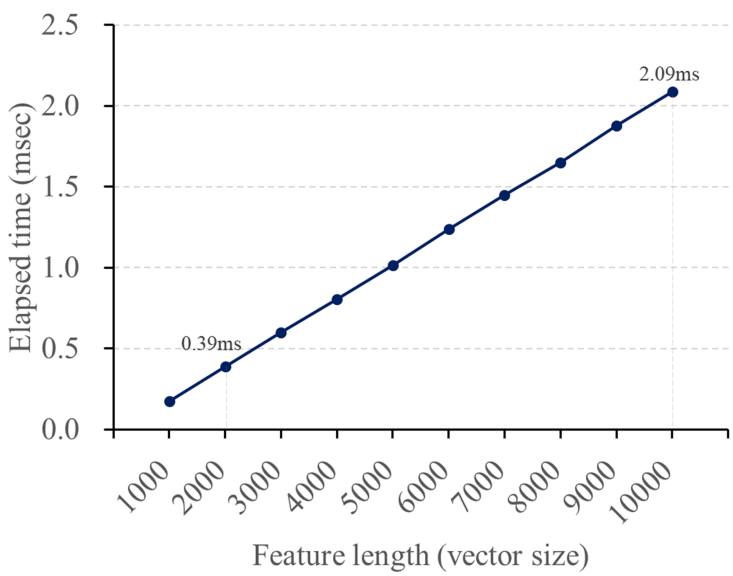
For additional scalability of the user profile, additional feasibility check was performed with a larger vector size than FAdE design. It proceeded up to 10,000 in units of 1,000 and measured in the same way as before. However, the LWE scheme did not proceed with the measurement due to the very large

time required and memory usage starting from the size of over 2,000.

Each section showed the same result as Figure 4.2 based on the size of 1,000

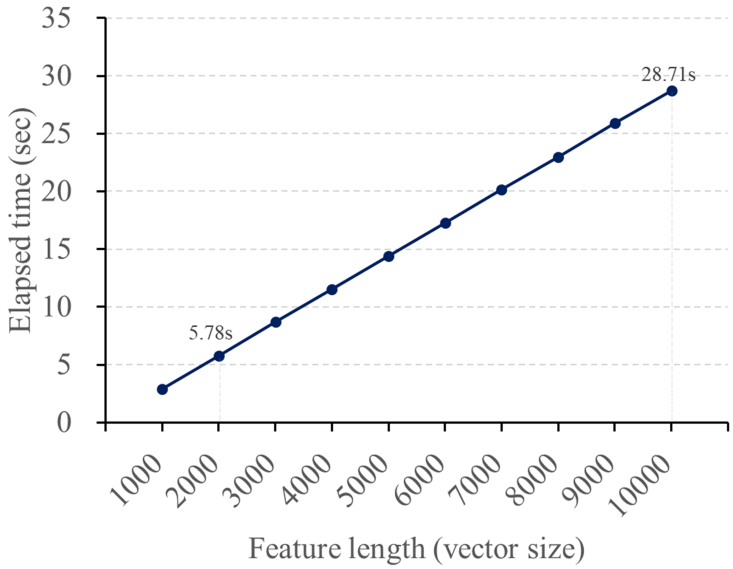


(a) Setup

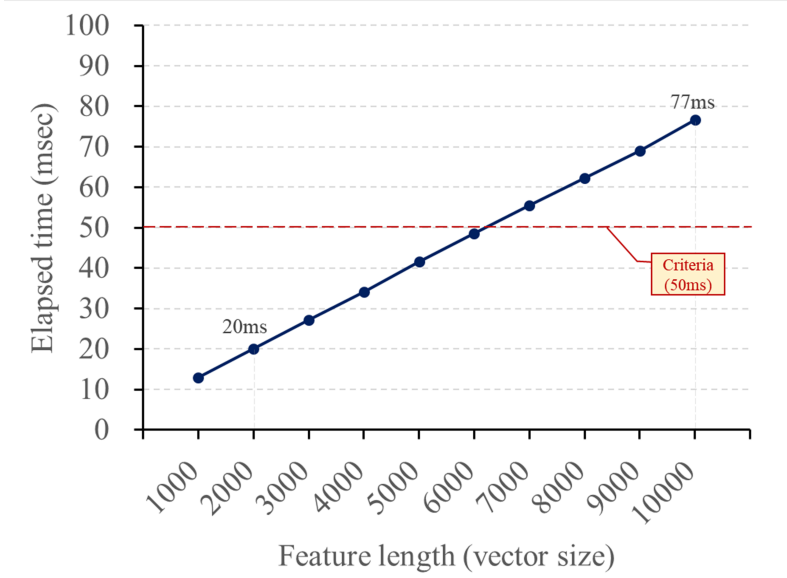


(b) $feKey$ Derivation

Figure 4.2: Vector length from 1,000 to 10,000



(c) Encryption



(d) Decryption

Figure 4.2: Vector length from 1,000 to 10,000 (cont.)

to 10,000 vector. In the case of decryption, it was not satisfied with my tight

	MSK	MPK	$func$	$feKey$	X_{user}	C_{user}
DDH	12.37MB	6.19MB	30.00KB	1.24KB	30.00KB	6.19MB

Table 4.2: Result of file size when vector length is 10,000

criteria of 50ms, but if I use a faster computing environment than mine, I expect that the size of 10,000 is also performed within the criteria and is still applicable. In case of file size, C_{user} increased in proportion to the increase of X_{user}, and it still shows the size of 6.19MB that can be transferred within 10ms.

4.4 Prototyping

The prototype shows that targeted advertising is possible using an encrypted user profile. It consisted of 1 TTP, 3 DSP, 1 ADX and 2 simulated users. TTP generates a key and provides the keys to each player, and ADX delivers a bid request and proceeds with an auction to deliver a win ad to the user. The three DSPs have different advertisements (car promotion, child product and furniture), define target users for each advertisement, and create user score functions for each. Two simulated users, Jeff(Figure 4.3) and Zoe(Figure 4.4), each have a user profile composed of different bit vectors according to personal information and interests.

When access the publisher site, the user's encrypted profile(C_{user}) is delivered to the DSPs, and the DSPs participate in bidding by obtaining a score using C_{user} and the score function. In this prototype, the more suitable users for the advertisement, the higher the probability of submitting a high bid price with a high score. Figure 4.5 shows that ADX looks at the bids from the DSPs, determines the win bid, and delivers the advertisement to the user.

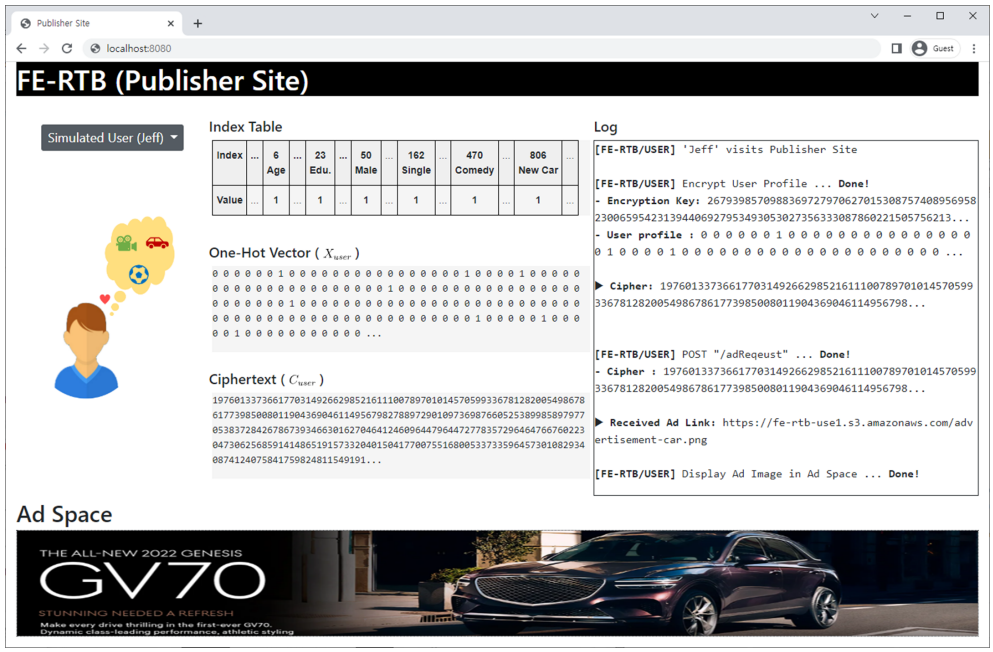


Figure 4.3: FAdE Prototyping - Simulated user 'Jeff'

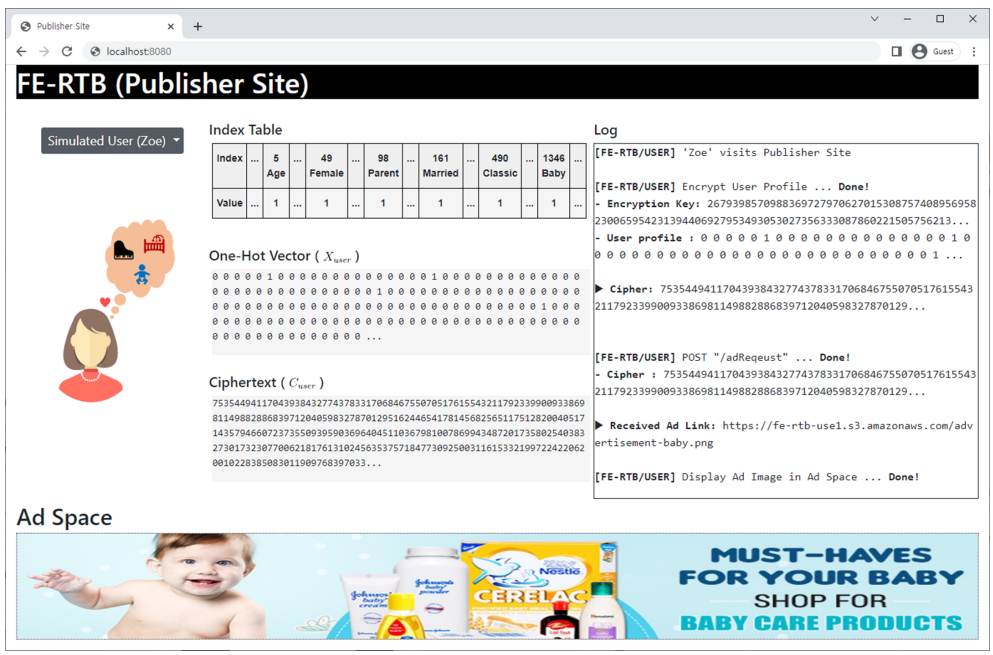
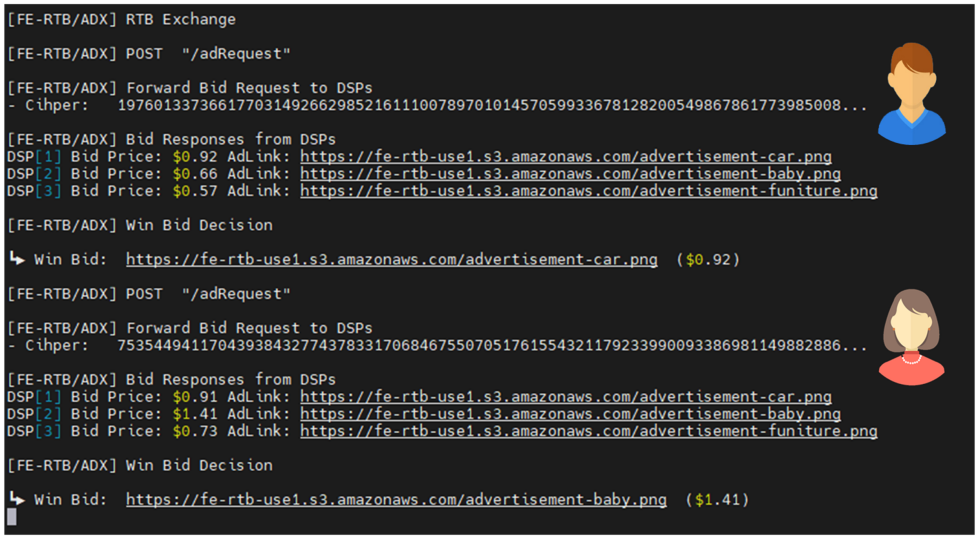


Figure 4.4: FAdE Prototyping - Simulated user 'Zoe'

ADX



DSPs

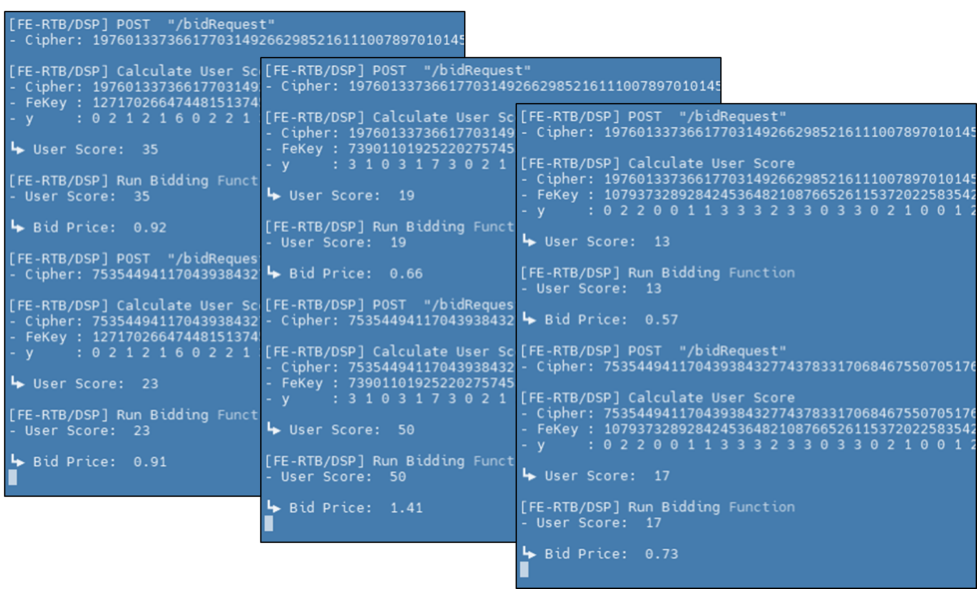


Figure 4.5: Bidding process on ADX and DSPs

Chapter 5

Related work

There have been several works on privacy-preserving advertising from different sectors. Particularly, from the tech industry, I notice the Improving Web Advertising Business Group (WebAdvBG) [37] and the Private Advertising Technology Community Group (PATCG) [38] from the World Wide Web Consortium (W3C). They are putting a lot of effort on developing new web platform features to support web advertising and provide users with privacy guarantees with a strong technical basis. Notable ideas and proposals from this groups are *FLoC: Federated Learning of Cohorts* [39], which has now been replaced by Google's Topics API [40], and Google's Fledge [41] implementation from the TURTLEDOVE proposal, the successor of PIGIN. For other proposals and detailed information, refer to the GitHub repository of the WebAdv BG [42].

For the remainder of this thesis, I focus on literatures of academia that cover complete advertising pipeline like FAdE. Table 5.1 compares the targeting accuracy, privacy leakage, trusted third party, RTB support, and practicality/scalability of FAdE to other proposals.

	Targeting Accuracy	Privacy Leakage	Trusted 3rd Party	Supports RTB	Scalable / Practical
Adnostic [43]	Contextual	Contextual	No	No	No
Privad [44]	Limited targeting	Broad interests	Yes (Dealer)	No	Yes
ObliviAd [45]	Fully targeted	None (TEE)	Yes (TEE)	No	No
AHEad [46]	Fully targeted	None	No	Yes	No (> 100s per bid)
BAdASS [47]	Fully targeted	None	No*	Yes	Yes (< 30ms per bid)
Pri-RTB [48]	Fully targeted	None	No	Yes	No
Themis [49]	Fully targeted	None	Yes (PoA Blockchain)	No	N/A
Adveil [50]	Fully targeted	None	Yes (Tor network)	No	No (use of Tor)
FAdE	**Fully targeted**	**None**	**Yes (IPFE Key Master)**	**Yes**	**Yes (< 20ms for user score)**

* BAdASS splits trust among DSPs. A single malicious DSP can disrupt the correctness of the protocol.

Table 5.1: Comparison of FAdE to other proposals

Adnostic [43] is a browser extension that preserves privacy by performing targeting locally on the client, this is already a departure from the RTB ecosystem. In addition, Adnostic only uses contextual features during targeting which is insufficient to provide well-chosen ads to the user. To ensure accurate accounting between advertisers, publishers, and ad-networks without compromising user privacy, Adnostic utilizes homomorphic encryption (HE) and zero-knowledge proofs (ZKP). However, this is a bottleneck to the entire system and not appropriate for real-world usage.

Privad [44] provides targeting based on broad interest categories that are locally determined by the client. Privad introduces an anonymizing proxy called the Dealer, to provide user privacy and enforce fraud prevention. The Dealer is assumed not to collude with the broker, who brings together advertisers, publishers, and users in the current model (e.g., Google). The Dealer provides user privacy by managing communication between the user and the broker.

ObliviAd [45] is a provably secure architecture for privacy preserving online behavioral advertising (OBA) by heavily relying on a Trusted Execution

Environment (TEE) which is a secure remote co-processor (SC) and Oblivious RAM (ORAM). The TEE is used in all stages of the advertising pipeline, from ad targeting to unlinkability of reports and fraud prevention. The user first sends his or hers encrypted behavioral profile to the SC which then selects the ads that match best based on the algorithm. To prevent the ad network from learning which ads are selected, they leverage an ORAM scheme. This architecture is not practical since the ORAM can only serve a single client request at a time and concurrent accesses would require a replica of the ORAM instance per-request. In addition, TEEs have seen a series of powerful attacks since ObliviAd was published.

BAdASS [47] and **AHEad** [46] are both designed to be compatible with the RTB ecosystem, which is similar to my approach. This enables behavioral targeting based on highly detailed user profiles. BAdASS leverages the highly fragmented nature of the RTB landscape to distribute trust among DSPs and uses a multi-party computation (MPC) protocol to preserve user privacy. The authors claim that BAdASS is the first protocol to allow sub-second behavioral targeting of advertisements while preserving user privacy can obtain sensitive information. On the other hand, AHEad uses threshold homomorphic encryption to preserve privacy within the RTB model. The authors also agreed in BAdASS that the use of expensive cryptographic schemes results in large computational costs and the amount of time it spends for calculating a single bid (more than 100 seconds) is nowhere close to practical.

Pri-RTB [48] claims to support the RTB ecosystem, and similar to AHEad, uses additively homomorphic encrypted user profiles. However, the suggested protocol requires additional communication between the browser (user) and the ADX which adds round trip times (RTTs) and increase the overall latency.

Moreover, by using computationally expensive HE, Pri-RTB is practically unusable in current RTB ecosystem.

Themis [49] is a Brave browser's contribution to privacy-preserving targeted advertising. It replaces the role of the Broker with a permissioned blockchain run by Publishers or foundations such as the Electronic Frontier Foundation (EFF). Themis additionally supports payment to users for their interaction with ads. Privacy for payments, metrics, and auditing of the blockchain is based on a Proof of Authority (PoA) protocol. In addition to the removal of the Broker, both targeting, and delivery is performed locally by the clients. As a result, all users must download both the targeting model and the entire database of ads and ad features to their local device.

AdVeil [50] proposes a modular privacy-preserving advertising ecosystem with formal guarantees for end users. To preserve privacy in targeting, it uses a single-server PIR protocol and a locality-sensitive hashing mechanism to allow the users to learn which ads to fetch from the broker. Both the ad retrieval and the reporting scheme rely on an anonymizing proxy to ensure the unlinkability between the user's preferences and the queries issued to the ad broker. Using anonymous proxies correctly such as Tor is not trivial for average web users. In addition, many ISPs and private networks block access to such networks, which effectively prevents AdVeil users from successfully fetching and displaying.

Chapter 6

Conculsion

In this thesis, I introduced the privacy issues of the existing Real-time Bidding ecosystem for targeted advertising and suggested a novel approach, FAdE, which is the introduction of Functional Encryption in RTB to solve these privacy issues. In doing so, I defined a reasonable size and encoding method of the user profile and designed the role and behavior of each player in the RTB, such as Browser, DSP, and TTP. I evaluated the performance and criteria of the FE schemes, DDH and LWE, and the file size. Also, I presented a prototype that delivers targeted advertisements using an encrypted user profile. My results show the practical feasibility of FAdE to provide user privacy in targeted advertising. I hope to see further work on this topic to improve user privacy in online targeted advertising.

Bibliography

[1] IAB. Internet Advertising Revenue Report: Full Year 2021. https://www.iab.com/insights/internet-advertising-revenue-report-full-year-2021/.

[2] Statista. Biggest revenue source of leading online and tech companies in most recently reported quarter ending March 2022. https://www.statista.com/statistics/218701/largest-source-of-revenue-of-leading-tech-companies/.

[3] Bloomberg. Apple Finds Its Next Big Business: Showing Ads on Your iPhone. https://www.bloomberg.com/news/newsletters/2022-08-14/apple-aapl-set-to-expand-advertising-bringing-ads-to-maps-tv-and-books-apps-l6tdqqmg.

[4] GDPR. General Data Protection Regulation. https://gdpr-info.eu/.

[5] CCPA. California Consumer Privacy Act 2019. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.

[6] The Wall Street Journal. Google, Amazon Fined $163 Million as France Takes Hard Line on Privacy. https://www.wsj.com/articles/google-

amazon-fined-163-million-as-france-takes-hard-line-on-privacy-11607601278.

[7] Reuters. S.Korea fines Google, Meta billions of won for privacy violations. https://www.reuters.com/technology/skorea-fines-google-meta-over-accusations-privacy-law-violations-yonhap-2022-09-14/.

[8] Google. The Privacy Sandbox. https://privacysandbox.com/.

[9] Apple. ATT (App Tracking Transparency). https://developer.apple.com/documentation/apptrackingtransparency.

[10] Miguel Alcobendas, Shunto Kobayashi, and Matthew Shum. The impact of privacy measures on online advertising markets. *Available at SSRN 3782889*, 2021.

[11] The Wall Street Journal. Facebook Feels $10 Billion Sting From Apple's Privacy Push. https://www.wsj.com/articles/facebook-feels-10-billion-sting-from-apples-privacy-push-11643898139.

[12] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the seventh international workshop on data mining for online advertising*, pages 1–8, 2013.

[13] Miguel Alcobendas, Shunto Kobayashi, and Matthew Shum. The impact of privacy measures on online advertising markets. *Available at SSRN 3782889*, 2021.

[14] Chang-Ji Wang, Pan-Pan Li, Xin-Yu Zhou, and Ning Liu. Privacy-preserving breast cancer prediction via inner-product functional encryption. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 539–543. IEEE, 2021.

[15] Wooil Kim, Hyubjin Lee, and Yon Dohn Chung. Safe contact tracing for covid-19: A method without privacy breach using functional encryption techniques based-on spatio-temporal trajectory data. *PloS one*, 15(12):e0242758, 2020.

[16] IAB. OpenRTB(Real-Time Bidding). `https://iabtechlab.com/standards/openrtb/`.

[17] Shanmugavelayutham Muthukrishnan. Ad exchanges: Research issues. In *International workshop on internet and network economics*, pages 1–12. Springer, 2009.

[18] Ad Age. Ad Tech Companies Band Together To Form New Coalition. `https://adage.com/article/digital/ad-tech-companies-form-online-buying-coalition/147624`.

[19] Shelly Palmer. 200 Milliseconds: The Life of a Programmatic RTB Ad Impression. `https://www.shellypalmer.com/2014/06/200-milliseconds/`.

[20] Google. Latency Restrictions and Peering. `https://developers.google.com/authorized-buyers/rtb/peer-guide`.

[21] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 457–473. Springer, 2005.

[22] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to delegate and verify in public: Verifiable computation from attribute-based encryption. In *Theory of Cryptography Conference*, pages 422–439. Springer, 2012.

[23] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[24] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*, 2012.

[25] Paulo Martins, Leonel Sousa, and Artur Mariano. A survey on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017.

[26] Damien Ligier, Sergiu Carpov, Caroline Fontaine, and Renaud Sirdey. Information leakage analysis of inner-product functional encryption based data classification. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 303–3035. IEEE, 2017.

[27] Sergiu Carpov, Caroline Fontaine, Damien Ligier, and Renaud Sirdey. Illuminating the dark or how to recover what should not be seen in fe-based classifiers. In *Proceedings of Privacy Enhancing Technologies*, volume 2020, pages 5–23, 2020.

[28] Linru Zhang, Xiangning Wang, Yuechen Chen, and Siu-Ming Yiu. Leakage-resilient inner-product functional encryption in the bounded-retrieval model. In *International Conference on Information and Communications Security*, pages 565–587. Springer, 2020.

[29] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In *IACR International Workshop on Public Key Cryptography*, pages 733–751. Springer, 2015.

[30] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure func-

tional encryption for inner products, from standard assumptions. In *Annual International Cryptology Conference*, pages 333–362. Springer, 2016.

[31] Tilen Marc, Miha Stopar, Jan Hartman, Manca Bizjak, and Jolanda Modic. Privacy-enhanced machine learning with functional encryption. In *European Symposium on Research in Computer Security*, pages 3–21. Springer, 2019.

[32] Théo Ryffel, Edouard Dufour-Sans, Romain Gay, Francis Bach, and David Pointcheval. Partially encrypted machine learning using functional encryption. *arXiv preprint arXiv:1905.10214*, 2019.

[33] Google. Ads API. `https://developers.google.com/google-ads/api/reference/data/codes-formats`.

[34] IAB. Audience Taxonomy. `https://iabtechlab.com/standards/audience-taxonomy/`.

[35] Google. Network peering. `https://peering.google.com/#/options/peering`.

[36] FENTEC. CiFEr - Functional Encryption library. `https://github.com/fentec-project/CiFEr`.

[37] W3C. Improving Web Advertising Business Group. `https://www.w3.org/community/web-adv/`.

[38] W3C. Private Advertising Technology Community Group. `https://patcg.github.io/`.

[39] Google. FLoC (Federated Learning of Cohorts). `https://developer.chrome.com/en/docs/privacy-sandbox/floc/`.

[40] Google. Topics API - Interest-based advertising. `https://developer.chrome.com/en/docs/privacy-sandbox/topics/`.

[41] Google. FLEDGE - On-device auctions for custom audiences. `https://developer.chrome.com/en/docs/privacy-sandbox/fledge/`.

[42] W3C. WebAdv BG. `https://github.com/w3c/web-advertising`.

[43] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. In *Proceedings Network and Distributed System Symposium*, 2010.

[44] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*, 2011.

[45] Michael Backes, Aniket Kate, Matteo Maffei, and Kim Pecina. Obliviad: Provably secure and practical online behavioral advertising. In *2012 IEEE Symposium on Security and Privacy*, pages 257–271. IEEE, 2012.

[46] Leon J Helsloot, Gamze Tillem, and Zekeriya Erkin. Ahead: privacy-preserving online behavioural advertising using homomorphic encryption. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2017.

[47] Leon J Helsloot, Gamze Tillem, and Zekeriya Erkin. Badass: Preserving privacy in behavioural advertising with applied secret sharing. In *International Conference on Provable Security*, pages 397–405. Springer, 2018.

[48] Erdong Deng, Huajun Zhang, Peilin Wu, Fei Guo, Zhen Liu, Haojin Zhu, and Zhenfu Cao. Pri-rtb: Privacy-preserving real-time bidding for secur-

ing mobile advertisement in ubiquitous computing. *Information Sciences*, 504:354–371, 2019.

[49] Gonçalo Pestana, Iñigo Querejeta-Azurmendi, Panagiotis Papadopoulos, and Benjamin Livshits. Themis: Decentralized and trustless ad platform with reporting integrity. *arXiv preprint arXiv:2007.05556*, 2020.

[50] Sacha Servan-Schreiber, Kyle Hogan, and Srinivas Devadas. Adveil: A private targeted advertising ecosystem. *Cryptology ePrint Archive*, 2021.

Appendix A

A.1 Bid Request Sample (OpenRTB 2.5)

```
{
  "id":"123456789316e6ede735f123ef6e32361bfc7b22",
  "at":2,
  "cur":[
    "USD"
  ],
  "imp":[
    {
      "id":"1",
      "bidfloor":0.03,
      "iframebuster":[
        "vendor1.com",
        "vendor2.com"
      ],
      "banner":{
        "h":250,
        "w":300,
        "pos":0,
        "battr":[
          13
        ],
        "expdir":[
```

```
          2,
          4
        ]
      }
    }
  ],
  "site":{
    "id":"102855",
    "cat":[
      "IAB3-1"
    ],
    "domain":"www.foobar.com",
    "page":"http://www.foobar.com/1234.html",
    "publisher":{
      "id":"8953",
      "name":"foobar.com",
      "cat":[
        "IAB3-1"
      ],
      "domain":"foobar.com"
    }
  },
  "device":{
    "ua":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/537
        .13 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2",
    "ip":"123.145.167.10"
  },
  "user":{
    "id":"55816b39711f9b5acf3b90e313ed29e51665623f",
    "buyeruid":"5456787654678765678987656789876543",
    "yob":1984,
    "gender":"M",
    "data":[
      {
        "id":"6",
        "name":"Data Provider 1",
        "segment":[
          {
            "id":"12341318394918",
            "name":"auto intenders"
```

```
            },
            {
              "id":"1234131839491234",
              "name":"auto enthusiasts"
            },
            {
              "id":"23423424",
              "name":"data-provider1-age",
              "value":"30-40"
            }
          ]
        }
    ]
  }
}
```

A.2 Functional Encryption Algorithm

Algorithms based on "*Fully secure functional encryption for inner products, from standard assumptions.* Agrawal et al. CRYPTO 2016"

With DDH Assumption

Algorithm Setup$(1^\lambda, 1^l)$

choose a cyclic group $\mathbb{G}$ of prime order $q > 2^\lambda$ with generators $g, h \in \mathbb{G}$
for all $1 \leq i \leq l$ **do**
    sample $s_i, t_i \hookleftarrow \mathbb{Z}_q$
    $h_i \leftarrow g^{s_i} \cdot h^{t_i}$
end for
$MPK \leftarrow (\mathbb{G}, g, h, \{h_i\}_{1 \leq i \leq l})$
$MSK \leftarrow (\{s_i\}_{1 \leq i \leq l}, \{t_i\}_{1 \leq i \leq l})$
return (MPK, MSK)

Algorithm Keygen$(MSK, \mathbf{w})$

$MSK \leftarrow (\{s_i\}_{1\leq i\leq l}, \{t_i\}_{1\leq i\leq l})$

$\mathbf{w} = (w_1, w_2, \cdots, w_l) \in \mathbb{Z}_q^l$

$s_{\mathbf{w}} \leftarrow \Sigma_{i=1}^{l} s_i \cdot w_i$

$t_{\mathbf{w}} \leftarrow \Sigma_{i=1}^{l} t_i \cdot w_i$

return $sk_{\mathbf{w}} \leftarrow (s_{\mathbf{w}}, t_{\mathbf{w}})$

Algorithm Encrypt$(MPK, \mathbf{v})$

$MPK \leftarrow (\mathbb{G}, g, h, \{h_i\}_{1\leq i\leq l})$

$\mathbf{v} = (v_1, v_2, \cdots, v_l) \in \mathbb{Z}_q^l$

sample $r \hookleftarrow \mathbb{Z}_q$

$C \leftarrow g^r, D \leftarrow h^r$

for all $1 \leq i \leq l$ **do**

    $E_i = g^{v_i} \cdot h_i^r$

end for

return $C_{\mathbf{v}} \leftarrow (C, D, \{E_i\}_{1\leq i\leq l})$

Algorithm Decrypt$(MPK, \mathbf{w}, sk_{\mathbf{w}}, C_{\mathbf{v}})$

$MPK \leftarrow (\mathbb{G}, g, h, \{h_i\}_{1\leq i\leq l})$

$\mathbf{w} = (w_1, w_2, \cdots, w_l) \in \mathbb{Z}_q^l$

$sk_{\mathbf{w}} \leftarrow (s_{\mathbf{w}}, t_{\mathbf{w}})$

$C_{\mathbf{v}} \leftarrow (C, D, \{E_i\}_{1\leq i\leq l})$

compute

$$E \leftarrow \prod_{i=1}^{l} \frac{E_i^{w_i}}{C^{s_{\mathbf{w}}} \cdot D^{t_{\mathbf{w}}}}$$

return $log_g E$

With LWE Assumption

Algorithm Setup$(1^n, 1^l, P, V)$

set integers $m, q \geq 2$, a real $\alpha \in (0, 1)$ and a distribution τ over $\mathbb{Z}^{l\times m}$

set $K = lPV$

sample $\mathbf{A} \hookleftarrow \mathbb{Z}^{m\times n}$ and $\mathbf{Z} \hookleftarrow \tau$

compute $\mathbf{U} = \mathbf{Z} \cdot \mathbf{A} \in \mathbb{Z}_q^{l\times n}$

$MPK \leftarrow (\mathbf{A}, \mathbf{U}, K, P, V)$

$MSK \leftarrow (\mathbf{Z})$

return (MPK, MSK)

Algorithm Keygen$(MSK, \mathbf{w})$

1: $MSK \leftarrow (\{s_i\}_{1\leq i\leq l}, \{t_i\}_{1\leq i\leq l})$

2: $\mathbf{w} \in \mathbf{V} = \{0, \cdots, V-1\}^l$

3: **return** $Z_{\mathbf{w}} \leftarrow \mathbf{w}^T \cdot \mathbf{Z} \in \mathbb{Z}^m$

Algorithm Encrypt$(MPK, \mathbf{v})$

1: $MPK \leftarrow (\mathbf{A}, \mathbf{U}, K, P, V)$

2: $\mathbf{v} \in \mathcal{P} = \{0, \cdots, P-1\}^l$

3: sample $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n$ and $\mathbf{e}_0 \hookleftarrow D_{\mathbb{Z},\alpha q}^m$ and $\mathbf{e}_1 \hookleftarrow D_{\mathbb{Z},\alpha q}^l$

4: compute

$$\mathbf{c}_0 = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}_0 \in \mathbb{Z}_q^m$$

5: compute

$$\mathbf{c}_1 = \mathbf{U} \cdot \mathbf{s} + \mathbf{e}_1 + \left\lfloor \frac{q}{K} \right\rfloor \cdot \mathbf{y} \in \mathbb{Z}_q^l$$

6: **return** $C_{\mathbf{v}} \leftarrow (\mathbf{c}_0, \mathbf{c}_1)$

Algorithm Decrypt$(MPK, \mathbf{w}, z_{\mathbf{w}}, C_{\mathbf{v}})$

1: $MPK \leftarrow (\mathbf{A}, \mathbf{U}, K, P, V)$

2: $\mathbf{w} \in \mathbf{V} = \{0, \cdots, V-1\}^l$

3: $z_{\mathbf{w}} \leftarrow \mathbf{w}^T \cdot \mathbf{Z} \in \mathbb{Z}^m$

4: $C_v \leftarrow (\mathbf{c}_0, \mathbf{c}_1)$

5: compute

$$\mu' = \langle \mathbf{w}, \mathbf{c}_1 \rangle - \langle z_{\mathbf{w}} \cdot \mathbf{c}_0 \rangle \ mod\ q$$

6: compute

$$\mu \in \{-K+1, \cdots, K-1\} \text{ that minimizes } \left| \left\lfloor \frac{q}{K} \right\rfloor \cdot \mu - \mu' \right|$$

7: **return** μ

국문초록

최근 사용자 개인 정보 보호에 대한 관심이 급증하면서 수십억 규모의 시장인 온라인 광고 산업도 같은 문제에 직면해 있다. 현재의 온라인 광고는 Real-time Bidding (RTB)과 사용자 타깃 광고 (targeted advertising)로 대표된다. 이는 웹 사이트에서 사용자의 정보를 바탕으로 관심과 선호도를 추정하고 이를 이용해 사용자에게 표시할 적절한 광고를 입찰, 선택하는 방식이다. 광고 요청을 위해 전송되는 user profile에는 사용자의 개인 정보를 침해하는 데이터가 포함되어 있으며, RTB 생태계의 여러 참여자에게 있는 그대로 전달되는 문제점이 있다.

본 연구는 사용자의 개인 정보를 보호하는 동시에 기존의 프로토콜 및 데이터 구조에는 최소한의 변경을 도입함으로써 현재의 RTB 생태계에서 계속해서 타깃 광고가 가능하도록 지원하는 FAdE를 제안한다. 제안하는 디자인은 Functional Encryption (FE)과 그 key 생성자인 Trusted 3rd Party (TTP)의 도입을 통해 개인정보 보호가 가능한 타깃 광고를 제공한다.

본 디자인에서는, 기존 타깃 광고를 위해 사용되던 user profile을 암호화(encrypt)하여 전달하므로 다른 RTB 환경의 참여자가 해독(decrypt)할 수 없다. Demand Side Platform (DSP)은 광고 요청에 대한 입찰 여부와 입찰가격을 결정하기 위해 암호화된 유저 데이터(encrypted user data, ciphertext)를 사용한다. DSP는 사전에 사용자의 점수를 연산하기 위한 function을 작성하고 이를 TTP에 제출하여 function key를 획득한다. 이 function key를 이용해 암호화된 유저 데이터를 해독(decrypt) 하면 DSP의 내부 입찰 알고리즘에 메트릭(metric)으로 활용할 수 있는 user score를 얻게 되고 이를 입찰 결정에 활용하게 된다. 결과적으로 RTB 환경 내에서 사용자의 개인정보는 보호하면서 DSP는 사용자의 숨겨진 정보를 기반으로 타깃 광고 입찰에 참여할 수 있다.

마지막으로, FAdE 디자인의 실제 활용 가능성에 대한 분석을 진행한다. user profile은 충분한 길이로 확인된 2,000 길이의 ‘0’과 ‘1’로 이루어진 벡터 (bit

vector) 형태로 생성한다. 이 user profile vector를 FE로 암호화(encrypt)한 후, weight vector에 해당하는 임의의 function과 벡터 내적(Inner product) 연산에 소요되는 시간을 측정하였을 때, user score를 도출하는 데 20ms 미만이 소요되는 것을 확인한다. 이는 광고 업계에서 일반적으로 사용되는 입찰 제한 시간(100-1,000ms)을 바탕으로 정의한 본 연구의 자체 기준 50ms 보다 충분히 작은 값에 해당한다. 이 결과는 동형 암호화(Homomorphic Encryption) 또는 Multi-Party Computation(MPC) 등을 사용하는 온라인 광고에서의 다른 개인정보 보호 제안보다 성능 상의 이점을 갖는다. 또한 제안 디자인을 활용해 타깃광고가 실제로 가능함을 확인하기 위해 Publisher 웹사이트, Ad Exchange(ADX), 3개의 DSP 그리고 TTP로 구성된 제안 디자인의 프로토타입 구현을 제시한다.

본 연구에서 제안된 FAdE를 통해 사용자의 개인 정보는 보호하면서 기존과 같은 수준의 타깃 광고가 가능하고, 이를 수용 가능한 수준의 적은 오버헤드로 적용이 가능하였음을 확인하였다. 연구의 결과가 향후 실제 온라인 광고 생태계에서 사용자의 프라이버시 보호에 기여할 수 있을 것으로 기대한다.

주요어: 프라이버시, 암호화, Real-time Bidding (RTB), Functional Encryption
학번: 2021-24027