Zero Knowledge Advertising: a new era of privacy-preserving AdTech solutions

Patricia Callejo*, Rubén Cuevas*†, Ángel Cuevas*† Mikko Kotila‡, Luke Bragg‡, Michiel Van Roey‡
*UC3M-Santander Big Data Institute, Spain
† Universidad Carlos III de Madrid, Spain
‡ Profila GmbH, Switzerland

Abstract—**The new data protection legislation along with the social pressure has encouraged the online advertising industry to propose novel privacy-preserving advertising solutions, such as Brave or Google's FLoC. While these pioneering solutions represent an undoubtedly important step towards a more ethical form of targeted advertising, they present some limitations. In this paper, we first systematically identify the limitations of the most promising industrial solutions. Armed with such knowledge, we present a new (simple and efficient) privacy preserving solution for delivering targeted ads, which combines the benefits of state-of-the-art proposals. To confirm the operational viability of our solution in the current online advertising ecosystem, we have performed extensive simulation experiments assessing that: 1) our solution requires a minimal use of device's resources (memory and CPU), making it valid for handheld devices running on limited power batteries; 2) it meets the strict timing requirements for the delivery of ads imposed in the online advertising ecosystem.**

I. Introduction

Online advertising is superior to other forms of advertising (e.g., TV or newspapers) due to its capacity of delivering personalized ads. To exploit this advantage online advertising stakeholders have developed a sophisticated tracking ecosystem able to track the activity of users online, but also offline, leveraging the sensors and GPS information retrieved from connected smartphones. The questionable tracking practices used by some players in the online advertising ecosystem has led to numerous scandals related to the potential misuse of users' personal data [1], [2], [3]. Those continuous scandals have increased the social and legal pressure [4], [5] on Ad Tech companies urging them to change their business models towards more privacy-preserving and ethics-oriented solutions.

We can find few recent efforts by industry such as Brave browser [6], Google's Federated Learning of Cohorts (FLoC) [7], [8] or Personal Data Platforms (PDPs) [9], [10], [11], [12] that represent a clear step in the market towards (at least reasonably) privacy-preserving ad delivery solutions. While these pioneering solutions represent doubtlessly important contributions to the field, they still have some limitations. For instance, some of these solutions, such as the Brave browser, operate as walled gardens limiting their operation to specific venues. Other approaches, such as FLoC or PDPs, do not offer the possibility of running full-private advertising processes (a.k.a. zero knowledge advertising) or enable auditability.

In this paper, we present a novel and comprehensive solution that tries to overcome the limitations of the existing state-of-the-art proposals. We rely in two main principles, simplicity and efficiency, and leverage basic cryptography technologies to propose a first version of our solution. The main functionalities of our solution are: 1) It is actionable in different venues (e.g., mobile apps, web browsers, video platforms) contrary to walled garden approaches; 2) It offers two operation modes: (i) the default mode is zero knowledge advertising, where personalized ads are delivered to users without sharing any users' data with third parties. (ii) The second mode is referred to as *consent-based advertising*, and it requires the proactive action of the user to activate it. In this mode, the user explicitly selects what personal data items (if any) they are willing to share with each advertiser. To illustrate this mode, let us think of a scenario where a user is willing to share: her height and weight with an advertiser A that sells clothes, her location with an advertiser B that sends last-minute restaurant discounts, and her interest on rock music with an advertiser C that promotes rock music concerts. Under this mode an advertiser with the user's consent can keep trace of the specific interactions of the user with its ads; 3) In our solution, users are rewarded for both their interaction (e.g., watching or clicking) with ads as well as for the data they voluntarily decide to share with advertisers; 4) The current Ad Tech industry builds profiles of users based on inference techniques that use for instance the browsing history of the user. The profiles resulting from these inference techniques have been proven inaccurate [13], [14]. Also, privacy-preserving solutions like Brave or FLoC stick to the use of such inference algorithms. Instead, our solution leverages self-declared information by users. This implies strong guarantees that ads delivered to a user are very likely aligned with their preferences; 5) The use of cryptography techniques generate irrefutable proofs of the actions performed by the involved stakeholders in our solution. In other words, our solution natively integrates auditing capabilities to provide safety to users, publishers, advertisers and other Ad Tech stakeholders.

To the best of the authors knowledge, there is no other privacy-preserving advertising solution offering the described set of features. Despite this, our solution does not aim at substituting existing ones, but to offer citizens and advertising stakeholders an alternative solution that aims to increase user privacy and, at the same time, increasing the efficiency of the ecosystem sending users ads more relevant to them.

While conceptual solutions may be theoretically superior to the state-of-the-art, there are practical requirements that might make such conceptual solutions non-operational in practice. In the context of online advertising solutions, these practical requirements can be roughly divided into two types. On the one hand, the online advertising market imposes a very strict time restriction since the ad delivery process must be completed in the order of few hundreds ms. On the other hand, ad delivery software operates in handheld devices (e.g., smartphones) which have limited resources (e.g., CPU or memory) and run on limited capacity batteries. So the misuse of such resources may severely affect the user experience. To confirm that our solution is operational in practice, we have run extensive simulations to measure the delay in the ad delivery process as well as the consumption of CPU and memory resources. Our results show that the ad delivery process is limited to tens of milliseconds for ads of standard size. Moreover, the CPU and memory consumption remains below 2.5% and 0.1%, respectively in all considered scenarios.

II. Background

In this section, we first describe the most commonly used advertising technology by modern advertising markets (programmatic advertising) and discuss what are the most apparent and inherent problems. Moreover, we describe the most relevant privacy preserving advertising solutions proposed by the industry in the recent past. Note that the research community has also contributed academic proposals in this field, e.g., [15], [16], [17], [18], [19], [20], which to the best of the authors knowledge have not been implemented as part of industrial/commercial solutions so far.

A. *Programmatic Advertising*

Programmatic Advertising is the common term used to refer to the online advertising solution in which the ad to be shown in the publisher's venue (webpage or mobile app) is chosen in real-time from a pool of active ad-campaigns. We make a distinction into two types of programmatic systems: *open-ecosystem* and *walled-gardens*. We explain the functionality of each of these systems next.

a) Open Ecosystem: Digital ads are shown in *publisher* venues (e.g., mobile apps or webpages). In particular, such venues offer ad spaces. Each time an ad space is available, a programmatic ad delivery process is triggered. Ad spaces are managed by the publisher's ad server, which (if available) loads a pre-configured ad in the ad space. Otherwise, it sells the ad either in a private or the open market[1]. Specifically, the publisher's ad server offers the ad space to an entity referred to as Supply Side Platform (SSP) through an ad request. This ad request includes information about: 1) the ad space (venue where it belongs, allowed type of ads, size, etc); 2) the user (e.g., age, gender, location, interests, etc.). This information is inferred through different tracking strategies (e.g., cookies, fingerprinting, etc.); 3) the device (mobile vs. fixed, Operating System, etc). SSPs typically handle ad requests from tens to hundreds publishers. The publisher along with its ad server and the SSP form the so-called *sell side* in programmatic advertising.

The SSP forwards the ad request to one (or more) Ad Exchange (AdX). The AdX gathers the information from the ad request and translates it into an OpenRTB protocol's bid request [21], which is sent to several Demand Side Platforms (DSPs). The DSPs are technology platforms where advertisers (or their agencies) configure their ad campaigns. Advertisers, their agencies and DSPs form the *buy side* in programmatic advertising AdXs are the platforms connecting the sell and buy sides. Upon the reception of a *bid request*, a DSP checks if the information it includes about the ad space, the user and the device matches any of its configured ad campaigns. If so, it replies with a *bid response* message including a bid for the offered ad space. The AdX collects all the *bid responses* from different DSPs and conducts a real-time auction to select the wining bid. The winning DSP is informed with a *win-notice* message and provides the URL from where the ad can be retrieved. Note that, ads are typically stored in either the advertiser or the DSP's ad server. Upon the reception of the ad's URL, the user's device retrieves the ad and renders it in the corresponding ad space. The advertiser that have delivered the ad pays a fee which is shared among all the involved players (publisher, SSP, AdX and DSP).

Finally, note that the programmatic ad delivery process sets strict time constrains in the order of hundreds of ms [22]. To maximize the marketing effect of ads, it is desired that the ad can be rendered as soon as possible after the webpage/mobile app loading process is triggered.

Interested readers can find a more detailed description of the functionality of the open programmatic advertising in [23].

b) Walled-Garden: We refer to walled-gardens as those advertising platforms where a single player controls the full ad delivery process. Examples of relevant walled-garden players in the online advertising ecosystem are Facebook, YouTube or Brave.

The essence of the ad delivery process is the same as in the case of the open ecosystem. In particular, the walled-garden platform owns the venue where ad spaces are shown (e.g., Facebook's application), so that they play an equivalent role to publishers in the open ecosystem. In addition, walled-garden platforms allow advertisers to configure their advertising campaigns based on the type of ad and the targeted audience and device type. Advertisers also configure their budget parameters (e.g., the price they are willing to pay per ad impression or per click). This service is equivalent to the offered by DSPs in the open ecosystem.

When an ad space is available, the walled-garden platform gathers information about the ad space, the user and the device and runs a real-time auction among those ad campaigns matching the gathered information. The auction algorithms selects a wining ad campaign whose ad is delivered to the user. Walled-gardens typically force advertisers to store ads in its own platform, so that they play the equivalent role of advertiser/DSP ad server in the open ecosystem. The fee payed by the advertiser delivering the ad goes entirely to the walled-garden platform.

[1] Both markets operate similarly. The only difference is that private markets are formed by a selected group of stakeholders.

c) Privacy considerations: The online advertising ecosystem offers a clear advantage over traditional advertising channels like TV, press or radio stations: *personalization*. People's online activity can be tracked and afterwards processed to obtain a profile of each individual, which reveals their preferences and interests. Conceptually, this is a good idea, because knowing the interests of someone allows showing them ads aligned to their interests. However, the *obsession* for improving personalization has led to the development of a sophisticated and invasive tracking ecosystem mainly motivated by the digital advertising business model. This ecosystem even goes further than the online activity of people and, with the proliferation of smartphones, it is able to also track the physical mobility and places visited by an individual [24], [25], [26], [27], [28].

This sophisticated tracking ecosystem has led to different scandals [1], [2], [3] that have triggered a reaction by people and public administrations. On the one hand, despite of the described tracking ecosystem, online advertising stakeholders still largely fail to show people ads of their interests [13], [14], which translates into many users considering online ads annoying and useless. This users dissatisfaction along with the perception of privacy intrusion have led people to increasingly install ad blocker solutions [29], [30]. On the other hand, the numerous scandals related to personal data misuse have led some administrations to develop modern data protection legislation to guarantee personal data is collected, stored and processed under clear and strict conditions. Under these new regulations, some of the standard practices implemented in the Ad Tech ecosystem may be considered illegal. Examples of these regulations are the GDPR in the EU [4]; the CCPA in California [5]; the LGDP in Brazil [31]; POPI in South-Africa [32]; etc.

The described events have motivated/forced the Ad Tech ecosystem to take action and propose few privacy preserving alternatives that we introduce in the next subsection. In addition to these solutions, there are many voices asking to eliminate the third-party cookies, which is currently the most widespread technique to conduct tracking in the web.

B. Privacy Preserving Advertising Solutions

In this subsection, we present the most relevant privacy-preserving advertising solutions proposed so far: Passive Zero-Knowledge Advertising solutions (whose most prominent representative is Brave), Federated Learning of Cohorts (FLoC, proposed by Google) and Consent-Based Personal Data Platforms (PDPs).

1) Brave: Passive Zero-Knowledge Advertising: Brave is a company whose main product[2] is the web browser with the same name. Brave browser was first released in November 2019.

The differential feature offered by Brave compared to other existing browsers is that it is a privacy-preserving browser. It blocks by default all ads and third-party trackers without affecting end-users' experience while surfing the web.

The business model of Brave is advertising. It offers their users an opt-in option to activate ads in Brave. Users opting in would receive ads. There are two important differences with respect to other browsers: 1) Brave compensate each individual Brave browser user for each delivered ad with its own cryptocurrency named *Basic Attention Token* (BAT); 2) the type of ads offered by Brave are non-invasive ads in the form of notifications that appear in the right upper corner of the screen. While significantly less intrusive than display or video ads, it is not clear the marketing efficiency of this type of ads.

Brave ads are targeted ads. The browser uses the websites visited by a person in order to infer their interests and preferences. However, this information stays local in the browser and it is neither shared with third parties nor even with Brave's own back-end. Instead, Brave collects a pool of ads from the ad campaigns available and sends them to the browser instances. Hence, the matching of the person's profile to the most suitable campaign is computed locally in the browser instance. This is a change to the current programmatic advertising paradigm in which the profile of the user is sent through several third-party platforms to reach DSPs where the match between the ad campaigns and the person's profile is executed (See Section II-A).

Due to the described functionality we classify Brave as a Passive Zero Knowledge Advertising (ZKA) solution. We consider it Passive, since the profile of a user is inferred by the browser without the *active* intervention of the user.

Another relevant aspects to highlight from Brave's operation in the context of this paper are the following ones:

1) Brave operates as a walled-garden using as venue to show ads its browser.
2) Brave offers users the possibility of proactively (opt-in) deactivating the so-called shields that will allow: (i) trackers and third party cookies operate normally, (ii) users receive regular ads. The user can enable this action for a specific website or for all websites. Users choosing this option will have a similar browsing experience as in other browsers such as Google Chrome. Although this is possible, it may be complex for non-skilled users to set up this type of privacy configurations.
3) It cannot be considered a full ZKA solution since in the standard operation of its current version it still requires revealing the IP address of the device in some cases [34]. Note that the IP address has been identified by the GDPR as Personal Data. Brave claims that they do not record the IP address or share it with third parties. Brave enables the use of IPFS [35], a p2p DHT-based solution that is still in a very early phase where few content can be accessed.

2) Google's Topics: Google announced that Chrome, which accounts with roughly 2/3 of the browsers market share [36], [37], [38], would cease the use of third-party cookies[3]. This represents in practice the end of the third party cookies, what

[2] Brave has recently made the release of its second product, Brave Search, in beta mode [33].

[3] Initially, the cessation was announced for beginning of 2022, but in a latter press release Google postponed it to late 2023.

	Opennes	Zero Knowledge	Consent Based	Active vs. Passive	Reward for users	Auditability
Brave	Walled-garden	✓		Passive	✓	✓
Google Topics	✓			Passive		
PDPs	?		✓(fine-grained)	Active	✓	
Our Solution	✓	✓	✓**(fine-grained)**	**Active**	✓	✓

TABLE I: Summary of the features offered by state-of-the-art privacy-preserving advertising proposals and our solution (the symbol ✓indicates the solution offer that property; the symbol ”?” indicates that such solution may or may not offer that feature depending on the specific implementation.

has triggered an intense debate with respect to the targeting online advertising in the *post-cookie* era.

Google’s initial proposal referred to as *Federated Learning of Cohorts* (FLoC) [7], [8], has been recently discarded and substituted by a new proposal referred to as *TOPICS* [39]. In this solution, the web browser (i.e., Google Chrome) computes the top 5 interests of a user every week. These interests are extracted from the browsing history of the user, based on the categories assigned by Google to the different websites visited by the user. The interests from the last three weeks are stored. So that, when a user visit a given website, the Topics API will return 3 interests, one from each of the three previous weeks. The solution provide some features to enhance privacy. On the one hand, a third party (referred as *caller* in the context of Topics) can only receive an interest from a user, if such third party has observed the presence of that user in a website classified with such a topic. To clarify this point, let us consider the following toy example: a user U with *sports* as one of the assigned topics. A third party *T* in the website www.shoes.com can only received U’s interest *sports* if it previously watch U in a sports website (e.g., www.sports.com). On the other hand, a sixth random topic is assigned to a user every week. With a probability 5% the random topic is returned.

While Topics offers clear significant privacy improvements compare to current cookie-based targeting approach, it cannot be considered to offer strong privacy guarantees. For instance, a player able to fingerprint a user, might be able to collect the history of interests of a user along time, coming up with a large number of interests associated to a user. Moreover, if multiple players fingerprint a user, they can share the information they have learned from the user in the background. On the other hand, Google should demonstrate the marketing efficiency of Topics. Some questions that arise around this are: are 3 topics sufficient to properly target a user?; what about demographic characteristics, such as gender or age?; how accurate is the websites’ topic classification algorithm?.

3) PDPs: Fine-grained consent-based advertising: Personal Data Platforms (PDPs) offer users the possibility to handle their personal data and decide which data and with whom to share it. In the context of online advertising, these platforms allow users to decide the players they are willing to share data with, and which specific data items. The result is a fine-grained consent-based form of advertising. PDPs typically offer people a user interface (e.g., mobile app), where people can configure their data-sharing preferences (e.g., which data share and with whom). Based on the users’ configured privacy preferences, a PDP can offer audiences in bulk or individually (as it would occur in online advertising) to the advertisers providing all required data protection guarantees. Moreover, it is a common design choice among proposed PDPs to offer users explicit rewards in exchange of their data as, for instance, Brave does.

Fine-grained consent-based advertising and PDPs are quite recent concepts that are still being covered by research projects [12]. However, there have been already several start-ups proposing a PDP solution, e.g., [9], [10], [11].

III. Our solution

A. Context

The review of existing privacy-preserving advertising solutions allows us to define a set of features to frame the design of our solution and contribute a step forward in the context of privacy-preserving digital advertising. In particular, the features we consider in the design of our solution are:

1) *Openness*: We have seen that advertising solutions can operate either as walled-gardens, where ads are shown in a venue controlled by the stakeholder (e.g., Facebook or Brave), or in the open market, where ads are delivered in third party venues (e.g., FloC or open programmatic market).
2) *Zero Knwoledge*: Some privacy-preserving advertising proposals (e.g., Brave) allow to implement targeting advertising without sharing the user information with any third party.
3) *Consent-based*: Some privacy-preserving proposals offer users the possibility to explicitly consent which information can be shared with third parties such as the case of PDPs.
4) *Active vs. Passive*: We refer as *passive* solutions to those ones relying on inference algorithms to obtain the preferences of the user without their intervention (e.g., Facebook, Google or Brave). Instead, in *active* solutions users take an active role and explicitly declares their preferences (e.g., PDPs).
5) *User’s compensation*: Some privacy-preserving solutions opt to reward the users for the ads delivered to them (e.g, Brave or PDPs) whereas others (e.g., FLoC) do not compensate users.
6) *Auditability*: The use of cryptographic techniques allow to create proofs of events related to each advertising operation or event, which in turn enables the possibility of auditing the system functionality. For instance, Brave offers this functionality.

Table I summarizes the functionality offered by each of the discussed privacy-preserving solutions in Section II across the defined features.

B. Requirements

Using the 6 features introduced before as reference, our goal is to propose a solution that meets the following requirements:

- *Open* solution able to deliver ads across any potential venue that operates (now or in the future) in the programmatic ecosystem (webs, mobile apps, TV, Outdoor screens, etc).
- *Active* solution where users explicitly declare their interest, instead of relying in unreliable inference algorithms [13], [14].
- It should implement a combination of *Zero Knowledge* and fine-grained *Consent-based* advertising. In particular, it would operate a ZKA protocol by default. However, for those specific cases in which the user explicitly provides consent to share a specific set of data items with a specific third-party, such information will be shared with the indicated third party.
- It should implement a *compensation* scheme for users in order to share with them the economic benefit resulting from the advertising operation.
- It should be *auditable*.

Table I shows the features of our solution and allows to compare them to those offered by state-of-the-art solutions introduced in Section II.

We would like to highlight that the information presented in Table I should not be consider as basis to argue a given solution is better than other. The solutions considered in the table must be understood as alternative solutions offering different features, which can very well co-exist together. The purpose of Table I, in the context of this paper, is to show in a visual and simple manner that our solution is different from state-of-the-art privacy-preserving advertising proposal from a technical and a functional aspect.

C. Assumptions

We make a number of assumptions regarding functions that we include in our proposal for which a solution already exists. Next we describe them:

- **Unique Identity linked to a physical person**: We need that each user is represented in our platform through an *Avatar* linked to a real person. To this end we leverage available technology which solves this issue, e.g., [40], [41].
- **Registration Process**: Again, we assume that the registration process is provided by some existing technology. Indeed, the same solution can provide both the identity matching (discuss in the previous bullet) and the registration functions [40], [41].
- **Smart contracts between parties**: Each time a transaction between a user and a third party occurs cryptographic non-repudiable proofs are generated. These provides auditing guarantees for each transaction. For some of these transactions, specific implementations of our solution may consider the use of smart contracts. In this case, we again propose to use existing technology that solves this problem, e.g. Smart Contracts on top of Etherum [42].
- **Auditing process**: In this paper, we just describe how to create auditable transactions, so that any implementation of our solution is subject to auditing not only by the involved stakeholders but also by third parties. The details where the auditable proofs are stored (most likely a blockchain) and how the auditing process can be implemented is out of the scope of this paper. Actually, this is something specific to each implementation.
- **Compensation to users**: As indicated above, our solution is expected to reward users for their interaction with ads. In particular, the company running our solution will compensate people for sharing their data with advertisers as well as for their interaction with ads. In principle, we assume that this compensation will be done through a cryptocurrency (e.g., Cardano, BAT, etc) but other options could be implemented as well (e.g., free subscription to Netflix).
- As for the rest of literature in the context of ZKA and Content-based Advertising (CBA), we do not address the issue of advertising fraud in this paper. However, it is worth noting that our solution significantly limits the ability of an attacker to commit fraud. As described above, each avatar in our solution must be linked to a unique real-world identity. Hence, the number of accounts an attacker can create in the proposed system is limited to the number of real-world identities it has access to, which is likely to be limited to a few accounts. This significantly increases the difficulty of creating botnets or likewise large scale attack infrastructures. Studying in detail fraud aspects is left for future work.

D. Involved Players

In this subsection, we describe the main stakeholders considered in our solution. Figure 1 shows a scheme representing the described components and their interactions.

- *Avatar*: This is the user identity within our solution. This identity must be linked to a real person's identity, but no personal data is available as part of this identity. We refer to it as person's Avatar. Each individual is armed with a certificate linked to its correspondent public/private keys. Indeed, we envision using modern solutions such as hierarchical deterministic keys [43], [44] where an arbitrary number of public/private key pairs can be generated from a seed public/private key pair. This technique is used to create hierarchical deterministic crypto-currency wallets [45].
- *Ad Delivery Software*: As described above, we envision an open and democratic solution that can operate in any third-party publisher willing to show ads in the online advertising ecosystem. Therefore, our solution will be implemented in a specific software that can be integrated by third parties. In particular, in the mobile ecosystem the ad delivery software is expected to be deployed in

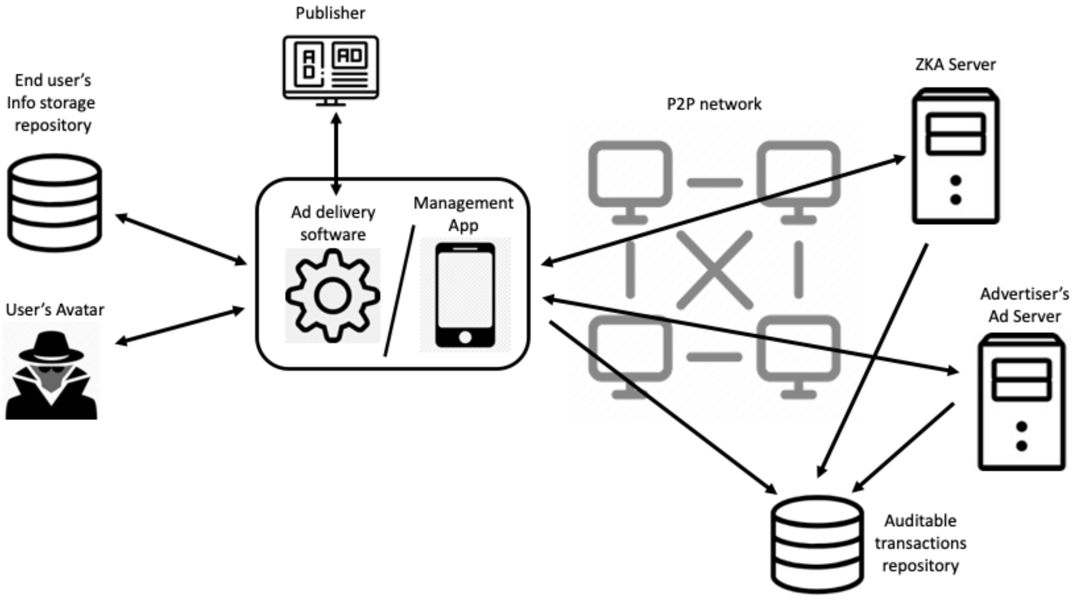


Fig. 1: Scheme of our solution including the components and the interaction among them.

an SDK to be integrated by mobile apps, whereas in desktops the ad delivery software can be either integrated within the web browser or be implemented as a web browser extension.

- *Publisher*: The publisher is the owner of the venues offering ad spaces, specifically mobile apps, web pages or video platforms. The publisher will integrate the ad delivery software described above, and this one will take care of choosing the ads to be shown in the publisher's venue.
- *Advertiser ad server*: In current programmatic advertising, ad campaigns are configured in DSPs. Moreover, ads are served from advertisers' ad servers or, if chosen by the advertiser, from its DSP's ad server. For simplicity, in this paper we consider a single entity, referred to as Advertiser ad server, that will take care of all the Advertiser related interactions. Note that, in a specific implementation of our solution, the functions offered by this Advertiser ad server can be divided across different entities. While this might impose some practical considerations, it does not impose any conceptual or design limitation.
- *Zero-Knowledge Ad (ZKA) Server*: A company using our solution will operate what we refer to as a Zero Knowledge Ad (ZKA) Server. This server will serve as intermediary to distribute the available ad inventory reported by advertisers to the *ad delivery software* installed in the user's device. It will receive only the data from the ad delivery software required to implement the billing and accounting process and will not receive any data which could reveal the identity of the user. This is why we refer to it as a *zero knowledge* ad server.
- *P2P network*: Our solution deploys a P2P network formed by users that voluntarily indicate their willingness to participate in such network. This P2P network guarantees that all communications started by user's avatar A in device D are routed through other members of the P2P network towards its final destination. By doing so we avoid any third party (Zero-Knowledge Ad Server Provider or the Advertisers Ad Servers) to know the actual IP address of the device involved in the communication. To the best of the authors knowledge, there is no other privacy preserving advertising solution providing such level of anonymity for the IP address of a user. We remind that the IP address is considered personal data by the GDPR.
- *End-users' info storage repository*: End user's data is stored in a storage repository in the cloud where all the info provided by the user as well as their transactions in the ad ecosystem (through the ad-delivery software) are stored. This repository is protected by a password and only the user can access it. Note that, other access schemes might be considered as well, e.g., based on cryptography certificates. We chose password-based protection due to its extensive use and familiarity even for non-skilled people.
- *Auditable transactions repository*: All executed transactions are cryptographically signed by the involved players and the signed probes are stored in this repository, which can be implemented using blockchain and smart contracts technology. Since all the players involved in a transaction provide undeniable proof of their agreement on the execution of such transaction, our solution provides no-reputability guarantees by default.
- *Management App*: This is the main app of the company implementing our solution. This app is the interface that allow users to: 1) configure their Avatar; 2) grant/revoke access to data to specific third parties to implement consent-based advertising; 3) access the list of transactions executed through the advertising platform (ads received, clicks on ads, etc) and the associated crypto-

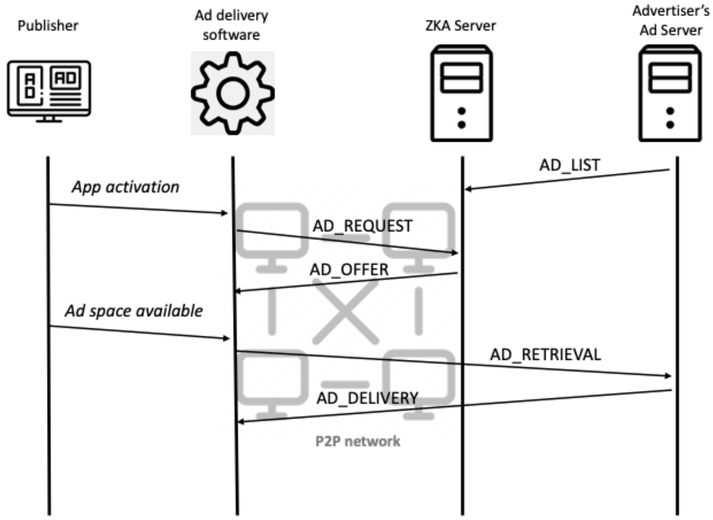


Fig. 2: ZKA protocol overview.

graphical probes; 4) have access to its associated wallet to assess the compensations received for their participation in the advertising platform. The management app is also synchronized with the end-user's info storage repository. Hence users can manage their information from multiple devices. Indeed it is expected that the management app is instantiated in the form of mobile app (so the user can use it from their mobile devices) as well as a web app (so the user can use it from any browser).

E. Protocol Description

In this section, we describe the protocol that handles the different operations to grant the desired functionality, i.e., a combination of zero knowledge and fine-grained consent-based advertising guided by the explicit instructions of the user.

We first describe the Zero Knowledge Ads delivery protocol that is the default mode of operation of our solution. Afterwards, we explain the processes involved in the fine-grained consent-based ads delivery protocol: 1) the consent granting phase, in which users can grant consent to an advertiser to access some pieces of their data; 2) the consent revoking phase, where users can revoke consent to a given advertiser to access their data; 3) the consent-based ad delivery process. Finally, we discuss the billing&accounting protocol.

1) Zero Knowledge Ads delivery: For simplicity, to describe the ZKA delivery, let us assume that the ad delivery software used is an SDK installed in a mobile app.

Figure 2 shows the exchange of messages between the different entities involved in the process. Note that, unless other way stated, each message is forwarded through a different peer that is part of a p2p network, so that the actual IP address of the user's device is never exposed to third parties (which includes the ZKA advertiser's server operator as well as the publisher and advertisers). In addition, the device use a randomly selected public/private key pair from a large pool of keys generated using a hierarchical deterministic key system. Note that the use of a single public key in all communications would make such public key an unique identifier. For the shake of readability, we avoid explicitly mentioning this in every step of the protocol. Next, we describe step by step the message exchange depicted in Figure 2.

1) The ZKA Ad Server receives from each Advertiser Ad Server an AD_LIST message with a list including for each ad: an *ad id* and the *ad's metadata*. The ad's metadata includes the following information: the type of ad (e.g., display, video), the size of the ad, the target audience (age, gender, interests, etc.), the IP address or URL of the advertiser ad server from which retrieve the ad, the public key of the advertiser ad server, an expiration time (the deadline to deliver this ad to users), the compensation associated with the different events related to an ad impression (e.g., a view or a click), etc. The ZKA Server collects the meta information of the ad inventory from potentially thousands of advertisers. It is part of the business strategy of the company owning the ZKA Server deciding how to process this ad inventory by aggregating it all together or dividing it in groups depending of different classification criteria (e.g., historical performance of ad inventory, preferential agreements with certain advertisers, etc.). This is not different from the strategies defined nowadays by AdXs or SSPs in the current digital advertising ecosystem.
2) Upon the activation of the mobile app embedding the ad delivery SDK software, the software creates an AD_REQUEST message that is sent to the ZKA Server. The only parameter included in this message is a public key.
3) Upon the reception of the AD_REQUEST message, the ZKA Server generates an AD_OFFER message formed by a list including the ad id and ad meta-information for a pool of ads selected by the ZKA Server. The AD_OFFER message is encrypted with the public key provided by the ad delivery software in the AD_REQUEST message. By doing so, only the ad delivery software will be able to access the list of received ads. Note that the ZKA Server receives neither user's data from the Ad Delivery software nor the IP address that is hidden through the p2p network.
4) Once the list of ads is available in the ad delivery software, when an ad space is available in the mobile app, the software triggers the process to retrieve an ad to fill the ad space. Note that, defining the ad selection algorithm is out of the scope of this paper. However, the ad selection algorithm should be designed such that it maximizes the probability of the user interacting with the ad. To this end, the selection algorithm should consider the user interaction with ads in the past, the user's preferences as well as the information regarding the target audience included in the ads' meta-information.
5) Once the algorithm selects an ad, the ad delivery software retrieves the IP address of the advertiser ad server from which retrieve the ad (directly from the ad meta-information or through a DNS resolution of the URL). The ad delivery software sends an AD_RETRIEVAL

message to the advertiser ad server. This message includes the following data: ad ID encrypted with the public key of the advertiser ad server (by doing so, even an attacker intercepting the message would be unable to know the requested ad), a transaction ID (this will be used as unique reference of this ad transaction) and a public key associated with the user's avatar certificate.

6) Upon the reception of the `AD_RETRIEVAL` message, the advertiser ad server responds with an `AD_DELIVERY` message, which includes: the ad encrypted with the public key provided by the ad delivery software in the correspondent `AD_RETRIEVAL` message and the transaction ID of the correspondent `AD_RETRIEVAL` message.
7) Finally, the ad delivery software places the ad in the corresponding ad space.

The described process just depicts the basic functionality scheme, which is subject to different improvements/modifications. For instance, the ad selection algorithm could be executed in background to create a predefined sorted list of ads, so that the ad delivery software can prefetch a number of ads (e.g., 4 ads). Upon an ad slot is available, the ad delivery software would deliver immediately one of the prefetched ads to the mobile app for showing it to the user. Using ads prefetching the ad delivery delay would be negiglible.

2) Fine-grained consent-based Ad delivery: In this subsection we provide the details of our protocol for: 1) allowing the user to grant consent to an advertiser to access and process certain data of the user. We envision a process driven by the advertisers. An advertiser makes an offer in which it indicates the data it requires from the users as well as the compensation it is willing to deliver for the data; 2) allowing the user to revoke the consent to an advertiser; 3) deliver consent-based ads.

Note that from the user's perspective, the consent granting and revoking processes involve the management app whereas the consted-based ad delivery process requires the participation of the ad delivery software instead.

As in the case of ZKA delivery, unless otherwise stated, every message sent from the management app or the ad delivery software is forwarded through the p2p network so the IP address of the user's device is hidden.

a) Consent Granting: Figure 3 shows the exchange of messages we propose to govern the consent granting process that we describe next.

1) The advertiser ad server sends to the ZKA server a `CONSENT_GRANTING_OFFER` including: the data requested from the user (e.g., age, gender and location or age, gender and top interests); the offered compensation in the selected crypto-currency; the advertiser-ID and its URL (in case the user want to learn more from the advertiser); the IP address of the advertiser ad server; the offer ID; a public key associated with the advertiser. Obviously, an advertiser may send more than one offer. The ZKA server collects the `CONSENT_GRANTING_OFFERS` from multiple advertisers.

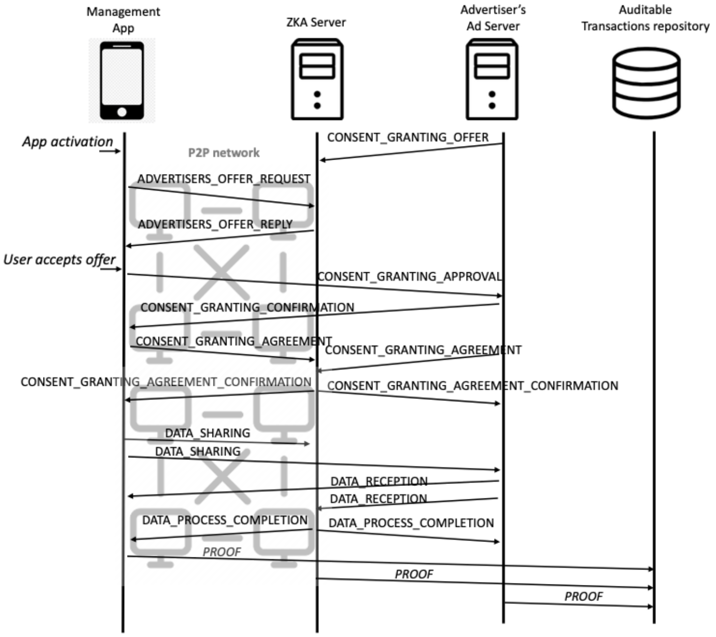


Fig. 3: Consent-granting protocol overview.

2) Upon the activation of the Management App, it sends an `ADVERTISERS_OFFER_REQUEST` to the ZKA server. This message only includes a public key associated with the user's avatar.
3) Upon the reception of the `ADVERTISERS_OFFER_REQUEST`, the ZKA ad server sends to the Management app an `ADVERTISERS_OFFER_REPLY` including the information of a pool of advertisers' offers. Note that the algorithm to select which offers to send to each request is out of the scope of the paper. For simplicity, we consider that all offers available at the ZKA server are included in every `ADVERTISERS_OFFER_REPLY` message. The message is encrypted with the management app public key, so that even if an attacker intercepts the message will not be able to read or modify it.
4) The management app processes the `ADVERTISERS_OFFER_REPLY` message and presents the different offers in a visual form to the user. If the user accepts an offer, the management app generates a `CONSENT_GRANTING_APPROVAL` message, which includes the following information: transaction ID (this is an unique ID of the transaction); a tuple formed by the offer-ID, advertiser-ID, user-ID and compensation encrypted with the public key of the advertiser. Note that the user-ID is a random number generated for the sole purpose of the relation between this advertiser and the user. Note, that the user-ID used for different advertisers is different and it is not linked to any PII data Moreover, the management app generates an undeniable proof of the consent granted by the user. To this end, it generates a token, which is a version of the previous tuple (transaction-ID, offer-ID, advertiser-ID, user-ID and compensation) signed with a private-key from the user's avatar. The `CONSENT_GRANTING_APPROVAL`

is sent to the advertiser ad server.

5) The advertiser ad server, upon the reception of the CONSENT_GRANTING_APPROVAL message responds with a CONSENT_GRANTING_CONFIRMATION message. This message includes: the transaction-ID as well as a token which is the signed version of the tuple introduced above (transaction-ID, offer-ID, advertiser-ID and compensation). It is signed with a private key from the advertiser. The confirmation message is sent to the Management App.
6) At this stage, and prior to proceed with the data sharing, both entities, the Management App and the advertiser's Ad Server, send a CONSENT_GRANTING_AGREEMENT message to the ZKA Ad Server. This message includes the transaction ID and the compensation information. The Management App signs it with a private key from the user's avatar and the advertiser ad server signs it with a private key from the advertiser.
7) Upon the reception of both CONSENT_GRANTING_AGREEMENT messages, the ZKA ad server sends a CONSENT_GRANTING_AGREEMENT_CONFIRMATION message to both, the Management App and the advertiser ad server. This message includes the transaction ID and compensation information signed with the private key of the company running the ZKA ad server. At this point the three involved parties have generated undeniable probes that they are aware of this transaction and approve it.
8) On the reception of the CONSENT_GRANTING_AGREEMENT_CONFIRMATION, the Management App sends a DATA_SHARING message to the advertiser ad server, that includes the data agreed to be shared with the advertiser. This message includes the transaction ID and the shared data encrypted with the public key of the advertiser ad server. Therefore, only the advertiser would be able to access the shared data. Finally, this message is also signed with the private key of the user's avatar, so that the user is providing undeniable proof that they shared such data. Hence, if the user is faking the shared data, there will be an evidence that can make the user accountable for faking the data. Moreover, the Management App sends a replica of the DATA_SHARING to the ZKA ad server without including the encrypted data part to provide even more guarantees the ZKA cannot access the user's shared data.
9) Upon the reception of the DATA_SHARING message, the advertiser answers with a DATA_RECEPTION message informing that the data has been received. This message includes the transaction ID signed with the private key of the advertiser's ad server. This message is sent to both the Ad Manager app and the ZKA ad server.
10) Upon the reception of both DATA_SHARING and the DATA_RECEPTION messages, the ZKA ad server sends a DATA_PROCESS_COMPLETION message to both the Management App and the advertiser ad server. This message includes the transaction ID and is signed with the private key of the ZKA ad server.

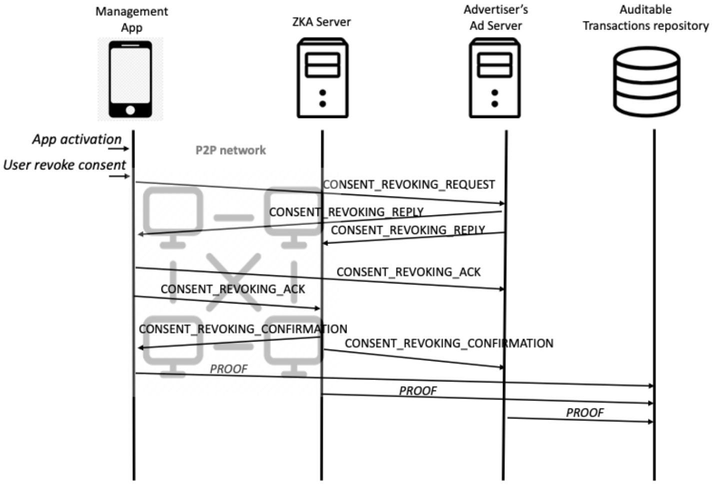


Fig. 4: Consent-revoking protocol overview.

Once the described protocol has been executed, the three entities involved in the process have agreed on the existence of the transaction and the completion of the data sharing and have generated a signed proof of their agreement. We have described an option in which the undeniable proofs consists of cryptographic proofs through signed messages that, as we describe later in the paper, are stored in the *auditable transactions repository*, e.g., a blockchain. However, other alternative approaches such as smart contracts can be also used in the described protocol with very minor modifications.

b) Revoking consent: Figure 4 shows graphically the exchange of messages that allows a user to revoke the consent granted to an advertiser to use their data.

1) When the user indicates their willingness to revoke the consent granted to an advertiser through the Management app, this one generates a CONSENT_REVOKING_REQUEST message. This message includes the transaction ID associated with the corresponding consent granting process, identifying the specific consent instance to be revoked. The message is signed with a private key from the user's avatar certificate and it is sent to both the the advertiser ad server and the ZKA server.
2) Upon the reception of the CONSENT_REVOKING_REQUEST message, the advertiser ad server sends a CONSENT_REVOKING_REPLY message to the Management app and the ZKA server. This message is signed with the private key of the advertiser. In the CONSENT_REVOKING_REPLY, the advertiser ad server also includes the data it is storing from the user and is planning to remove. This information is encrypted using the public key of the user's avatar, so only the Management app can read it.
3) The Management app verifies whether the the data to be removed is correct (it corresponds to the shared data in the consent granting process) and, if that is the case, sends a CONSENT_REVOKING_ACKNOWLEDGMENT to both the advertiser ad server and ZKA ad server. This message includes the transaction ID and is signed with

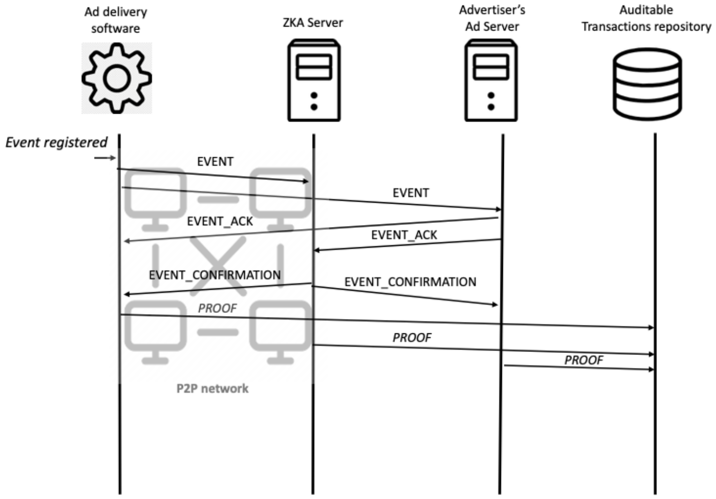


Fig. 5: Billing & Accounting protocol overview.

a private key from the user's avatar.

4) Finally once the CONSENT_REVOKING_REPLY and CONSENT_REVOKING_ACKNOWLEDGMENT is received by the the ZKA server, it generates a CONSENT_REVOKING_CONFIRMATION including the transaction ID, which is signed with the private key of the ZKA server. This message is sent to both the Management app and the advertiser ad server.

When the described process is concluded, the three parties have generated cryptographically undeniable proofs that they have implemented the consent revoking task. In addition, the advertiser provides proof of the data it had about the user and that it must remove due to the consent revoking process. Note that, if the advertiser does not delete the data and keeps using it after the consent revoking process has been completed, there is undeniable proof that the advertiser is exploiting personal data without the user's consent. This will represent a violation of most modern data protection regulations. Therefore, our solution would allow, for instance, data protection authorities to audit misbehaving advertisers. In our view, this is an important mechanism to discourage advertisers considering to ignore users' consent revocation.

Finally, as in the consent granting process, the cryptographical proofs can be extended to use smart contracts.

c) Consent-based Ads delivery: The consent-based ads delivery uses exactly the same message exchange described for the Zero Knowledge Advertising case. The only difference is that the ad delivery software includes the user ID (encrypted with the public key of the advertiser ad server) in the messages shared with the advertiser ad server. By doing so, the advertiser can match this ad with the specific user who granted the consent.

3) Billing & Accounting: The final function to implement is the billing & accounting process, which we address in this subsection. Figure 5 shows the messages exchange of our protocol to implements this function. Note that this process is triggered by an event associated with an ad (e.g., a user view or click in an ad). The process is the same independently of the type of event.

1) Upon the completion of an event, the ad delivery software generates an EVENT message including the following information: type of event (e.g., click or view), an event ID (a single identifier for the current event), the transaction ID (corresponding to the ad associated with this event), a wallet ID (the wallet where the cryptocurrency payment should be transferred) and the compensation associated from the event (extracted from the meta information of the ad). The ad delivery software signs the EVENT message with a private key from the user's avatar and sends it to both the advertiser ad server and the ZKA server. Note that we propose the use of Hierarchical Deterministic wallets [45], so that different transactions are associated to different public/private key pars from the wallet. The use of a single wallet ID would make this one a potential unique identifier of the users' avatar.
2) Upon the reception of the EVENT message, the advertiser ad server creates an EVENT_ACK message including the: type of event, event ID, transaction ID and the compensation associated with the event. The message is signed with its private key and sent to the ad delivery software and the ZKA server.
3) Finally, once the EVENT and EVENT_ACK are received by the ZKA server, this one generates an EVENT_CONFIRMATION message including the same information as the EVENT_ACK message. This message is signed with the private key of the the ZKA server.

It is important to note that, the consent granting also generates a monetary transaction and thus it should undergo the billing and accounting process. The process is the same as the one described above with slight modifications: the event type is consent granting and the transaction ID is the transaction associated with the consent granting process.

F. Auditability

In the previous section we have described the protocol that governs the functionality of our solution. In each of the described processes there are three involved players: 1) the advertiser ad server; 2) the ZKA ad server and 3) the ad delivery software or the management app (depending on the specific process). In every process each of these entities generate cryptographically signed proofs that confirm their agreement with respect to the process. In some cases, these confirmations include data exposing the specific activity of the entity within the process. For instance, in the consent revoking process, the advertiser ad server declares the data that it is going to delete.

The generated proof by an entity is sent to the other two entities, such that the three entities possess for each process its own proof but also the proof generated by the other two entities.

These proofs are uploaded to the *auditable transactions repository*, so that they can be validated and available in case any dispute between the parties requires so. In particular, we suggest this entity to implement a blockchain for auditing purposes.

Note that upon a dispute, e.g., a user rejecting it granted consent to an advertiser or an advertising still using data from a user after the consent revoking process has been executed, the proofs available in the *auditable transactions repository* can be accessed to settle the dispute since they provide undeniable guarantees of the actions taken by each party.

IV. Evaluation

The discussion presented in Sections II and III have made clear the main conceptual differences between our proposal and the state-of-the-art alternatives. Note that, we claim the design of our solution represents the main contribution of this paper, since it provides a new approach to deliver digital ads with full data protection guarantees, that extends the portfolio of the already existing solutions (See Table I).

Hence, since the main motivation of our proposal comes from a conceptual standpoint, and we acknowledge that our proposal is meant to co-exist with other existing approaches, a head-to-head performance comparison of our solution with the existing alternatives does not add much value to the paper. Instead, we must carefully evaluate that our solution qualifies from a technical and performance point of view to operate under the requirements imposed by the current online advertising ecosystem. As described in Section II, programmatic advertising set strict time constrains to guarantee that ads are rendered in hundreds of ms. From a quantitative performance perspective, our solution should be able to perform the complete ad delivery process within the established time constrains in the programmatic ecosystem. The equivalent to our solution within the current ad delivery process in programmatic advertising corresponds to the steps 4 to 7 of the ZKA protocol described in Sec. III-E1 and Figure 2. This is the part subject to timing constrains. For completeness, we also present timing results related to the steps 1 to 3 of the ZKA protocol, which is meant to send to the *ad delivery software* a list of available ads.

Moreover, the ad delivery software is expected to run in different types of devices from desktops to mobile phones. The use of resources (CPU, memory) is important in all type of devices but specially in mobile phones. Mobile phones operate a more limited architecture in terms of CPU and memory and use limited capacity batteries. Having this mind, our solution should make a limited use of these resources.

Finally, note that the obtained results can be generalized to the consent-based advertising alternative since the process is essentially the same, excepting few small variations in the information added in some of the messages (i.e., the user id), whose impact in the overall delay and resources consumption is negligible.

To assess the performance of our solution in the referred dimensions (time constrains and resources consumption), we have run extensive simulation experiments. In particular, in the rest of the section, we first describe the simulation setup and then we present the results obtained from the conducted simulation experiments. In particular we evaluate 3 metrics: delay of ad delivery process and CPU and memory consumption.

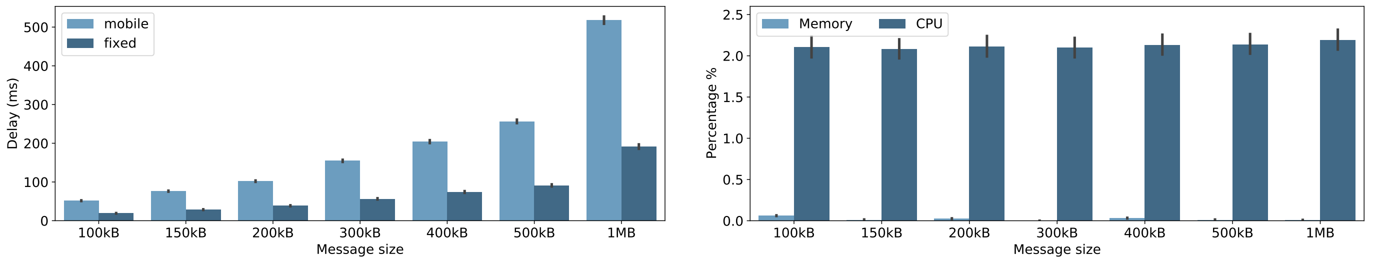
A. Simulation setup

Our simulation considers all the players involved in the ZKA delivery process: the mobile app, the ad delivery software, the ZKA server and the advertisers ad server.

The simulator has been implemented in Python. Next we detail the implementation of the different functions involved in the ad delivery process:

- We have implemented the exchange of messages using standard HTTPS/TCP sockets between the different players.
- The fields of the messages have been implemented using a serialization mechanism. This is a very common approach to encode data in Internet protocol payloads.
- To implement the cryptographic operations, i.e., encryption and signature functions we have used the Fernet library [4]. We assume a scheme URL-safe base64-encoded 32-byte key utilizing an implementation of symmetric authenticated cryptography, which is commonly used in the web.
- We have emulated the communication between the different players using an open stack infrastructure. Each player (ad delivery software, advertiser ad server and ZKA ad server) is set up in an independent open stack instance with 4GB RAM and 4vCPUs. Note that mid-range smartphones are equipped with similar resources to our open stack instances. Instead, back-end servers offer significantly more powerful resources.
- The ad selection process should happen in the order of few ms (in the worst case). Hence, we can safely assume that its contribution to the overall delay is negligible.
- We consider the advertiser ad server is hosted in a data-center owned by the advertiser or by a Content Delivery Network (CDN). In any of these cases, it is expected to account with network connections in the order of Gbps for both up and downstream.
- We assume that devices hosting the ad delivery software are end-user devices (smartphones, tablets, laptops or desktops) which could then be connected to either the fix or the mobile network access infrastructure. To simulate the download and upload rates associated to these devices we use as reference the data from the Ookla Speedtest [46] that provides the average upload and download rates for broadband fixed and mobile infrastructure across hundreds of countries. In each simulation run we select a random value for the upload (download) rate of the device from a range defined by the max and min upload (download) rates reported by Ookla for the 100 countries with fastest networks.
- Devices hosting proxy nodes from the p2p network are end-users' devices, which are selected in a smart way to guarantee good performance. To this end, in our simulation we assume that p2p proxies are end-user devices connected to broadband networks selected from the same country as the device hosting the ad delivery software. Hence, the upload and download rates of the p2p proxy

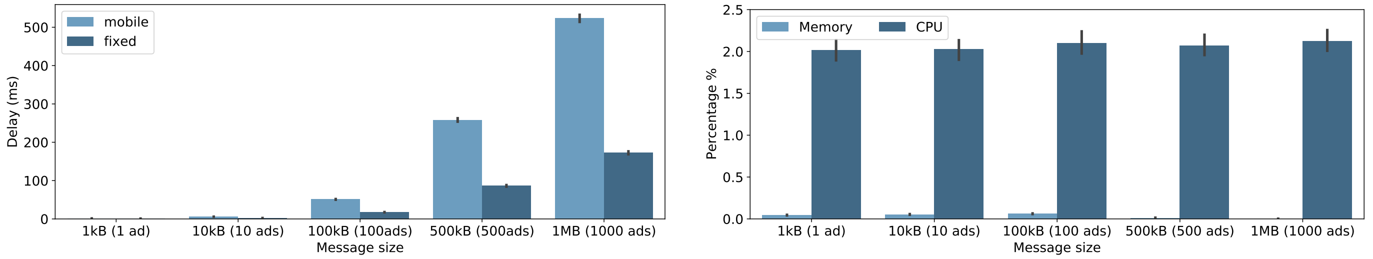
[4] https://cryptography.io/en/latest/fernet/



(a) Delay of communications

(b) Percentage of Memory and CPU consumption

Fig. 6: Performance results for steps 1 to 3 of the ad delivery protocol corresponding to the delivery of the metainformation of ads from the *ZKA server* to the *ad delivery software*



(a) Delay of communications

(b) Percentage of Memory and CPU consumption

Fig. 7: Performance results for steps 4 to 7 of the ad delivery protocol corresponding to the delivery of the ad from the *advertiser ad server* to the *ad delivery software*

would be accordingly obtained from the fixed broadband data from Ookla Speedtest.

- To simulate the effect of the p2p network, we consider a worst case scenario, in which the proxy receives the entire message from the source before forwarding it to the destination. This technique is referred to as *Stop-and-Wait* in communication protocols and it is known to be inefficient. However, in the context of our paper it serves the purpose to find an upper-bound for the delay introduced by our protocol.
- We assume that each ad metainformation register included in the `AD_OFFER` has a size of 1KB. Note that this is an upper bound of the actual expected size of the metainformation from an ad. In our simulation, we consider the `AD_OFFER` message include metainformation for N= [1,10,100,500,1000] ads.
- The common ad size in programmatic advertising is 150KB [47]. In our simulations, we consider the following ad sizes [100, 150, 200, 300, 400, 500, 1000] KB. Note that we consider sizes up to 6,66 times larger than the reference size of 150KB.

B. Results

In this subsection, we present the results from the simulations regarding the communication delay, the % of CPU utilization and the % of memory consumption for the two parts of the ZKA delivery protocol.

On the one hand, Figure 6 shows the results for the first part of the ZKA delivery protocol, involving steps 1 to 3 (See Section III-E1), which are meant to deliver a list of available ads to the *ad delivery software*. On the other hand, Figure 7 presents the results for the second part of the ZKA delivery protocol (steps 4 to 7), which is dedicated to the ad delivery process and is subject to delay restrictions set up in programmatic advertising.

Note that for each of the reported scenarios we have run 500 simulation samples. We report the average value of the considered metric (delay, % of memory consumption and % of CPU utilization) as a bar and the 95 confident interval in the form of an error bar.

On the one hand, we observe that our proposed solution meets the expected performance requirements to operate in any device and under the timing restrictions imposed by the online programmatic advertising, even when we consider the p2p proxies implement the inefficient *Stop and Wait* forwarding technique. In particular, for recommended ad sizes (150kB) the average delay of the ad delivery process in our simulations is 28.51 ms and 76.34 ms, for devices connected to fixed and mobile networks, respectively. Even in the considered extreme case with ads of 1MB (6,66 larger than the typical ad size) the average delay is bounded to 191.25 ms and 518.04 ms for devices fixed and mobile network connections, still in the range of few hundreds ms. Finally, it is worth mentioning that the use of ad prefetching techniques as described in Section III would lead to negligible ad delivery delay, since ads would have been prefetched and thus available locally in the device for each new ad space.

On the other hand, we observe that the resource consumption imposed by our solution into devices running the *ad delivery software* is affordable even for handheld devices for both the ads delivery and the metainformation delivery processes. In particular, the CPU utilization and memory consumption are smaller than 2,5% and 0,1%, respectively, in all considered cases.

Finally, we would like to highlight that the bandwidth overhead generated by our protocol is negligible. Most of the bandwidth consumption is associated with the payload of the messages. First, the size of the `AD_DELIVERY` message is determined by the size of the ad (typically few hundreds KB), which is common to any ad delivery platform and not specific to ours. Second, the size associated to the `AD_OFFER` message including the metainformation of ads. As we have discussed before, the matainformation of an ad is expected to be encoded in registers with a size <1KB. Hence, the total bandwidth consumed by these type of messages is expected to be negligible in comparison to the overall bandwidth consumed by regular web pages and mobile apps.

V. Conclusion

Different social, political and regulatory actions have urged the online advertising industry to revisit their privacy-related practices. This has led to the development of several privacy-preserving advertising approaches.

In this paper, we propose a novel privacy-preserving advertising solution that presents a combination of functionalities that, to the best of the authors knowledge, is not offered by state-of-the-art solutions in the field. In particular, our solution is designed to operate in multiple venues (webpages, mobile apps, etc). In addition, it offers a combination of zero knowledge advertising (which does not share data from users with third parties) with a fine-grained consent based advertising (which shares with those third parties explicitly indicated by a user the specific data selected by such user). Moreover, our solution operates based on explicitly declared information and preferences from users, instead of using inference mechanisms, and compensates users for sharing their data and interact with the ads. Finally, based on cryptography technology, our solution enables full auditability.

We have described in detail the protocol that will serve as the basis for the implementation of our solution as well as evaluated its performance through extensive simulations. Our evaluation confirms that the proposed solution can be implemented in practice. On the one hand, our solution meets the delay requirements imposed by the programmatic advertising delivery of ads. On the other hand, it present a limited use of resources (memory and cpu) from those devices running it.

As future work, we plan to implement a first prototype of our solution in the next months that will incorporate the feedback received from academia and industry with suggestions to improve the design presented in this paper.

References

[1] Wikipedia., "Cambridge analytica scandal," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal

[2] Wall Street Journal., "Google is fined $57 million under europe's data privacy law," 2019. [Online]. Available: https://www.nytimes.com/2019/01/21/technology/google-europe-gdpr-fine.html

[3] BBC, "Digital ad industry accused of huge data breach," 2021. [Online]. Available: https://www.bbc.com/news/technology-57232253

[4] THE EUROPEAN PARLIAMENT AND OF THE COUNCIL, "General Data Protection Regulation," https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN.

[5] California State Legislature, "California Consumer Privacy Act," https://www.caprivacy.org/.

[6] "Brave browser," https://brave.com/.

[7] Google, "FLoC whitepaper," https://github.com/google/ads-privacy/blob/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf.

[8] GitHub, "FLoC repository," https://github.com/WICG/floc.

[9] "WIBSON," https://wibson.io/en.

[10] "MyDataMood," https://mydatamood.com/.

[11] "Datawallet," https://datawallet.com/.

[12] H. E. U. Program, "PIMCITY: Building the next generation of personal data platforms." [Online]. Available: https://www.pimcity.eu/

[13] M. A. Bashir, U. Farooq, M. Shahid, M. F. Zaffar, and C. Wilson, "Quantity vs. quality: Evaluating user interest profiles using ad preference managers." in *NDSS*, 2019.

[14] P. Reserch., "Facebook algorithms and personal data." [Online]. Available: https://www.pewresearch.org/internet/2019/01/16/facebook-algorithms-and-personal-data/

[15] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising," in *Proceedings Network and Distributed System Symposium*, 2010.

[16] S. Guha, B. Cheng, and P. Francis, "Privad: Practical privacy in online advertising," in *USENIX conference on Networked systems design and implementation*, 2011, pp. 169–182.

[17] M. Backes, A. Kate, M. Maffei, and K. Pecina, "Obliviad: Provably secure and practical online behavioral advertising," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 257–271.

[18] L. J. Helsloot, G. Tillem, and Z. Erkin, "Ahead: privacy-preserving online behavioural advertising using homomorphic encryption," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 2017, pp. 1–6.

[19] Y. Pang, B. Wang, F. Wu, G. Chen, and B. Sheng, "Prota: A privacy-preserving protocol for real-time targeted advertising," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2015, pp. 1–8.

[20] G. Pestana, I. Querejeta-Azurmendi, P. Papadopoulos, and B. Livshits, "Themis: A decentralized privacy-preserving ad platform with reporting integrity," *arXiv preprint arXiv:2106.01940*, 2021.

[21] IAB Tech Lab, "Openrtb specification v3.0," Jun. 2020. [Online]. Available: https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/OpenRTB%20v3.0%20FINAL.md

[22] Google. [Online]. Available: https://developers.google.com/authorized-buyers/rtb/start

[23] A. Pastor, R. Cuevas, Cuevas, and A. Azcorra, "Establishing trust in online advertising with signed transactions," *IEEE Access*, vol. 9, pp. 2401–2414, 2021.

[24] L. Barkhuus and A. K. Dey, "Location-based services for mobile telephony: a study of users' privacy concerns." in *Interact*, vol. 3. Citeseer, 2003, pp. 702–712.

[25] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann, "Tracking human mobility using wifi signals," *PloS one*, vol. 10, no. 7, p. e0130824, 2015.

[26] A. Ilhan and K. J. Fietkiewicz, "Data privacy-related behavior and concerns of activity tracking technology users from germany and the usa," *Aslib Journal of Information Management*, 2020.

[27] J. L. Kröger, P. Raschke, and T. R. Bhuiyan, "Privacy implications of accelerometer data: a review of possible inferences," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 2019, pp. 81–87.

[28] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, 2017.

[29] M. Malloy, M. McNamara, A. Cahn, and P. Barford, "Ad blockers: Global prevalence and impact," in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 119–125. [Online]. Available: https://doi.org/10.1145/2987443.2987460

[30] "Ad blocker usage and demographic statistics in 2021." [Online]. Available: https://backlinko.com/ad-blockers-users

[31] Brazilian Government, "Lei Geral de Protecao de Dados Pessoais," http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709compilado.htm.

[32] South African Government, "Protection of Personal Information Act," https://popia.co.za/.

[33] "Brave search," https://brave.com/search/.

[34] "Brave Browser Privacy Policy," https://brave.com/privacy/browser/.

[35] "IPFS," https://ipfs.io/#how.

[36] P. Callejo, R. Cuevas, and Á. Cuevas, "An ad-driven measurement technique for monitoring the browser marketplace," *IEEE Access*, vol. 7, pp. 181 339–181 347, 2019.

[37] StatCounter, "Browser Market Share Worldwide," https://gs.statcounter.com/browser-market-share.

[38] ——, "Browser Market Share Worldwide," https://gs.statcounter.com/browser-market-share/mobile/worldwide.

[39] Google, "The Topics API." [Online]. Available: https://github.com/jkarlin/topics

[40] IOHK, "Atala PRISM." [Online]. Available: https://atalaprism.io

[41] "NYM." [Online]. Available: https://nymtech.net/

[42] "Introduction to Smart Contracts." [Online]. Available: https://ethereum.org/en/developers/docs/smart-contracts/

[43] D. Khovratovich and J. Law, "Bip32-ed25519: Hierarchical deterministic keys over a non-linear keyspace," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2017, pp. 27–31.

[44] C.-I. Fan, Y.-F. Tseng, H.-P. Su, R.-H. Hsu, and H. Kikuchi, "Secure hierarchical bitcoin wallet scheme against privilege escalation attacks," in *2018 IEEE Conference on Dependable and Secure Computing (DSC)*, 2018, pp. 1–8.

[45] "Hierarchical determinisitc wallet."

[46] "Ookla Speed Test." [Online]. Available: https://www.speedtest.net/global-index

[47] "IAB New Ad Portfolio: Advertising Creative Guidelines." [Online]. Available: https://iabtechlab.com/standards/new-ad-portfolio/