TU Delft

Delft University of Technology

AHEad

Privacy-preserving Online Behavioural Advertising using Homomorphic Encryption

Helsloot, Leon J.; Tillem, Gamze; Erkin, Zekeriya

DOI
10.1109/WIFS.2017.8267662

Publication date
2018

Document Version
Final published version

Published in
2017 IEEE Workshop on Information Forensics and Security, WIFS 2017

Citation (APA)
Helsloot, L. J., Tillem, G., & Erkin, Z. (2018). AHEad: Privacy-preserving Online Behavioural Advertising using Homomorphic Encryption. In *2017 IEEE Workshop on Information Forensics and Security, WIFS 2017* (pp. 1-6). IEEE. https://doi.org/10.1109/WIFS.2017.8267662

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

AHEad: Privacy-preserving Online Behavioural Advertising using Homomorphic Encryption

Leon J. Helsloot*, Gamze Tillem† and Zekeriya Erkin‡
Cyber Security Group, Department of Intelligent Systems, Delft University of Technology
Mekelweg 4, 2628 CD, Delft, The Netherlands
*l.j.helsloot@student.tudelft.nl, †g.tillem@tudelft.nl, ‡z.erkin@tudelft.nl

***Abstract*—Online advertising is a rapidly growing industry, forming the primary source of income for many publishers that offer free web content. The practice of serving advertisements based on individuals' interests greatly improves the expected effectiveness of advertisements, and is believed to be beneficial to publishers and users alike. However, the widespread data collection required for such behavioural advertising sparks concerns over user privacy. In this paper, we present AHEad, a privacy-preserving protocol for Online Behavioural Advertising that ensures user privacy by processing data in encrypted form. AHEad combines homomorphic encryption with a machine learning method commonly encountered in existing advertising systems. Advertisements are served based on detailed user profiles, while achieving performance linear in the size of user profiles. To the best of our knowledge, AHEad is the first protocol that preserves user privacy in behavioural advertising while allowing the use of detailed user profiles and machine learning methods.**

I. Introduction

Online advertising is a pervasive phenomenon on the Internet, backed by a multi-billion dollar industry with a worldwide spend of $178 billion in 2016 [1]. Advertisements allow publishers to offer web services free of charge, forming a primary financial pillar supporting free web content [2]. An increasing number of people object to being shown advertisements on web pages they visit, however, resulting in a rapid adoption of technological measures to block advertisements. In early 2017, it was estimated that ad blocking tools were installed on 615 million devices, amounting to 11% of the Internet population, and the adoption of such tools is predicted to increase in the future [3]. The use of ad blockers has lead to a significant loss of revenue from advertising space offered by publishers. The worldwide cost of ad blocking, in terms of missed revenue, was estimated to be $41.4 billion, or 23% of the total ad spend, in 2016 [4]. These developments threaten the business models of many free web services, necessitating measures to alleviate objections against advertising in order to attain a sustainable advertisement-supported Internet economy.

One of the objections people have to online advertisements is that widespread data collection by advertising companies infringes on user privacy [5]. 32% of respondents to a recent survey among ad blocker users indicated that privacy concerns were a reason for their use of an ad blocker [6]. In a similar survey on privacy and advertising, 94% of respondents indicated that online privacy was an important issue, and according to 70% of respondents, online advertising networks and online advertisers should be responsible for users' online privacy [7]. The widespread collection of user data that sparks privacy concerns is a key element of behavioural targeting, in which the advertisements that are shown to a user are selected based on the user's browsing behaviour. Although such tailored advertisements are recognized by users as being useful for both publishers and users, acceptance of behavioural advertising is hindered by a mistrust of advertising companies and a lack of control over the collection of information [5].

WIFS'2017, December, 4-7, 2017, Rennes, France. 978-1-5090-6769-5/17/$31.00 *©2017 IEEE.*

A. Online Behavioural Advertising

Online Behavioural Advertising (OBA) is the practice of serving advertisements based on individuals' interests. These interests are inferred from users' browsing behaviour, using data such as visited web pages, search queries, and online purchases. OBA allows for advertisements to be targeted at individual users, greatly improving the Click-Through Rate (CTR) and thus the expected effectiveness of advertisements [8]. Personalization of advertisements is typically performed using campaign-specific supervised machine learning models that predicts users' responses to advertisements. Users benefit from OBA by being served less irrelevant advertisements, whereas advertisers benefit from accurate targeting as it allows them to reach the desired audience. Finally, publishers experience an increased value of the advertising space they offer.

The current trend in OBA is the Real-Time Bidding (RTB) mechanism of buying and selling ads [9]. In the RTB model, ad exchanges (AdX) provide marketplaces where advertising space is auctioned in real time for individual impressions. When a user visits a web page containing advertising space, the publisher offers the ad impression for bids at one or more ad exchanges, after which advertisers place their bids within a fraction of a second. Other platforms emerged along with ad exchanges to manage the complexity of RTB. Demand-Side Platforms (DSPs) bid on impressions from multiple inventories on behalf of advertisers, who may not possess the expertise required to partake in programmatic real-time auctions. Supply-Side Platforms (SSPs) assist publishers in reaching a large number of advertisers by offering advertising space to multiple inventories.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 27,2020 at 12:25:29 UTC from IEEE Xplore. Restrictions apply.

Existing literature proposes a number of methods to address privacy concerns in OBA. These methods include blocking advertisements altogether [10], obfuscating browsing behaviour [11], and anonymization [12], as well as exposing only generalized user profiles to advertising companies [13]. However, limiting the data available to advertising companies is expected to decrease the accuracy of targeted advertisements [14], and thus the value of advertisements to users, advertisers, and publishers. Finally, some work proposes cryptographic approaches to aggregate ad click statistics [13] or perform advertisement selection using secure hardware [15]. These approaches, however, assume the existence of centralized advertising networks performing simple keyword-based advertisement selection, and are thus unsuitable for use within the highly distributed RTB model.

B. Our Contributions

In this paper, we present AHEad, a novel protocol that preserves user privacy in OBA, is compatible with the RTB mechanism of buying ads, and supports highly detailed user profiles. To the best of our knowledge, this is the first protocol making use of machine learning on encrypted data for preserving user privacy in OBA tasks. AHEad is based upon machine learning techniques commonly encountered in existing OBA systems, and allows multiple data processors, specifically DSPs, to operate on the same encrypted user data. Our protocol distributes trust between parties using threshold homomorphic encryption, such that no single party can decrypt sensitive information. We achieve performance linear in the size of user profiles, and see room for further performance improvements.

II. Preliminaries

A. Logistic Regression

Although a variety of machine learning algorithms has been proposed for user response estimation tasks, logistic regression has recently been used in production systems by many advertising companies, such as Google [16], Facebook [17], and Criteo [18], and is considered a state-of-the-art model [19]. A logistic regression model is used to estimate the probability of a binary outcome y (click or no click) based on a d-dimensional predictor vector $\boldsymbol{x}$ (the user profile) [9]. Given a feature vector $\boldsymbol{x}$ and model parameters $\boldsymbol{w}$, the model predicts the click probability $\hat{y}$ using the sigmoid function:

$$\hat{y} = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{w}^\top \boldsymbol{x}}}. \tag{1}$$

After observing the actual binary click label y, we use the gradient of the logistic loss to update model parameters $\boldsymbol{w}$ using an online Stochastic Gradient Descent (SGD) method as in [16]. Given the gradient of the loss $\boldsymbol{g} = (\hat{y} - y)\boldsymbol{x}$, we perform the model update $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \boldsymbol{g}$, where η is the learning rate. Note that the learning rate can be a constant, but may also be set to decay per iteration and per coordinate of the gradient.

B. Feature Hashing

Many of the features considered in OBA are categorical. The high cardinality of some of these features results in a very large feature space, particularly if feature conjunctions are used. To encode categories from a user profile into a vector of fixed dimensionality for use in logistic regression, we use the hashing trick [20], which showed promising results in e.g. [18]. The hashing trick maps values from our high-dimensional user profile into a lower-dimensional vector $\boldsymbol{x}$ by setting x_i to a count of the values whose hash is i. The resulting d-dimensional vector $\boldsymbol{x}$ (in [18], $d = 2^{24}$ is used) is the input feature vector to the logistic regression model.

C. Threshold Homomorphic Encryption

Our protocol relies on an additively homomorphic cryptosystem, such as Paillier [21]. The homomorphic properties of Paillier allow the computation of $\mathcal{E}_{pk}(m_1 + m_2) = \mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2)$ by any party with access to public key pk, where $\mathcal{E}_{pk}(x)$ is the encryption of a message x under public key pk. Likewise, one can compute $\mathcal{E}_{pk}(c \cdot m_1) = \mathcal{E}_{pk}(m_1)^c$ for a public constant c.

In our protocol, we use a two-party threshold version of Paillier as described in e.g. [22], such that trust is distributed between parties. We denote an encryption using threshold Paillier as $[\cdot]$, and have omitted share decryption steps from our protocol descriptions. We use $[\cdot]_u$ and $[\cdot]_{DSP}$ to represent encryptions using any asymmetric encryption scheme under the public key of the user or DSP, respectively.

III. Protocol Design

For our privacy-preserving online advertising protocol, we consider a setting that is compatible with the existing RTB landscape. We assume that DSPs are the only parties that perform operations on user data, i.e. that personalization of advertisements is performed solely by DSPs. The majority of our protocol is therefore performed at the DSPs, and ad exchanges are responsible only for collecting bids and selecting the highest bid. SSPs are assumed to only offer advertising space to multiple ad exchanges, and are not considered in our protocol for simplification purposes.

Since some existing companies act as both ad exchange and DSP, we assume that ad exchanges and DSPs collude. To preserve privacy in the presence of colluding parties, we introduce an additional entity into our setting called Privacy Service Provider (PSP). The PSP is assumed not to collude with any party, but is not trusted with user data. Therefore, we use a threshold version of Paillier, where one share of the private key is held by the PSP, and each of the DSPs and ad exchanges holds a copy of the other share. An advertising company must thus collaborate with the PSP to decrypt and vice versa. We consider the following parties in our protocol:

Users are the actors to whom advertisements are shown when they visit web pages. A user visits web pages using a web browser, which maintains a profile describing the user's behaviour. Whenever a web page containing advertising space is visited, the web browser requests an advertisement from an ad exchange. After displaying the ad, the web browser sends click feedback to the ad exchange, indicating whether the ad was clicked or not.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 27,2020 at 12:25:29 UTC from IEEE Xplore. Restrictions apply.

TABLE I
EXPLANATION OF SYMBOLS

Symbol	Explanation
$\boldsymbol{x}$	Input feature vector obtained by feature hashing.
K_γ	Set of campaigns run by a DSP γ.
$\boldsymbol{w}_k$	Model parameters for a campaign k, where $w_{k,i}$ is the i^{th} coordinate of $\boldsymbol{w}_k$.
$\eta_{k,i}$	Learning rate parameter for campaign k and coordinate i.
$\pi(\cdot)$	Random permutation function.
$\pi^{-1}(\cdot)$	Inverse permutation of $\pi(\cdot)$, such that $\pi^{-1}(\pi(\boldsymbol{x})) = \boldsymbol{x}$.
$B_k(\cdot)$	Bidding function for campaign k.
b_k	Bid value for campaign k.
a_k	Advertisement associated with campaign k.
k	Unique campaign identifier.

Ad exchanges offer a marketplace where advertising space is auctioned in real time. Ad exchanges send bid requests to DSPs, and pick the highest bid from the responses.

DSPs bid on advertising space on behalf of one or more advertisers. For every received bid request, a DSP generates a bid for each of its campaigns by running a response prediction model on the received input, and submits these bids and the associated advertisement to the ad exchange.

PSP is a service provider that assists in performing computations in a privacy-preserving manner.

Our protocol is based on a semi-honest security model, where ad exchanges and DSPs execute the protocol together with the PSP. The ad exchanges, DSPs and PSP should not learn any information about the contents of the user profile, nor which ads were viewed or clicked, from the protocol execution. Moreover, ad exchanges and the PSP should not learn any information about the parameters of the models run by DSPs, nor the bid values submitted by DSPs.

A. Initial setup

Prior to protocol execution, a key pair for the two-party Paillier scheme is generated, either by a Trusted Third Party (TTP) or using a distributed protocol as outlined in e.g. [22]. The private key is secret shared, such that the PSP holds one part of the key, and each DSP and ad exchange holds a copy of the other part of the key. Moreover, each DSP generates a key pair for use in the profile update protocol. Finally, advertisers set up their campaigns such that DSPs can bid on their behalf.

B. User profiling phase

Browsing behaviour is recorded within the user's web browser to form a local user profile, such that privacy can be preserved during the user profiling phase. Local profiling can be performed using existing techniques, such as RePriv [23]. The resulting profile information is captured into a d-dimensional feature vector $\boldsymbol{x}$ using feature hashing.

Since sending the full d-dimensional feature vector during each advertisement request would incur prohibitively high communication costs, feature vectors are cached at DSPs. Caching significantly reduces the amount of communication required during the time-sensitive advertisement selection phase, and allows feature vectors to be updated in the background to minimize delays experienced by the user. To further decrease communication costs, profile updates are periodically sent by the user. Depending on the expected size of user profiles, profile updates can be performed in either an incremental fashion or by completely replacing the user profile. Incremental updates may be less costly to encrypt and transmit for users, but require computationally expensive operations at DSPs. AHEad takes the latter approach of replacing the user profile. However, it can easily be modified to process incremental updates by using the additively homomorphic properties of the encryption scheme.

Input: $\boldsymbol{x}, u$

Pick $r \in_R \mathbb{Z}_d$
$P \leftarrow \{(i + r \pmod d), [x_i]]) \mid x_i \neq 0\}$
invoke EXPAND($P, \left[(u,r)\right]_{DSP}$) at PSP

procedure EXPAND($P, \left[(u,r)\right]_{DSP}$)
  for $j \leftarrow 1, d$ **do**
    $\left[\tilde{x}_j\right] \leftarrow \begin{cases} [v] & \text{if } (j, [v]) \in P \\ [0] & \text{otherwise} \end{cases}$
  end for
  invoke UPDATE($\left[\tilde{\boldsymbol{x}}\right], \left[(u,r)\right]_{DSP}$) at DSP
end procedure

procedure UPDATE($\left[\tilde{\boldsymbol{x}}\right], \left[(u,r)\right]_{DSP}$)
  Decrypt $\left[(u,r)\right]_{DSP}$
  Select profile $\left[\boldsymbol{x}\right]$ for user u
  for $i \leftarrow 1, d$ **do**
    $[x_i] \leftarrow [\tilde{x}_{(i-r \pmod d)}]$
  end for
end procedure

Fig. 1. Profile update protocol, initiated by every user

The steps performed during a periodic profile update are outlined in Fig. 1. An explanation of the symbols used in the protocol descriptions is given in Table I. The user with unique identifier u generates a d-dimensional feature vector $\boldsymbol{x}$ from their local profile information. Due to feature hashing, $\boldsymbol{x}$ is expected to be a very sparse high-dimensional vector. To reduce the communication costs for the user, the profile update is encoded as a set of pairs $P = \{(i + r \pmod d), [x_i])\}$ for non-zero x_i, where $r \in_R \mathbb{Z}_d$. This compressed representation is sent to the PSP, along with encryption $\left[(u,r)\right]_{DSP}$. The PSP then expands P into an element-wise encryption of the original vector $\boldsymbol{x}$, with its elements rotated r times due to the randomization of indices performed by the user, setting any element not present in P to $[0]$. This expansion is sent to the relevant DSPs, along with $\left[(u,r)\right]_{DSP}$. The DSPs decrypt the user's identifier to select the relevant user profile, and the random number to rotate the received expansion back to its original indices. Finally, the DSPs replaces the previous profile information of the user with the encrypted feature vector.

C. Bidding phase

During the bidding phase, every DSP calculates a bidding price for each of their campaigns, based on a cached user

Authorized licensed use limited to: TU Delft Library. Downloaded on May 27,2020 at 12:25:29 UTC from IEEE Xplore. Restrictions apply.

Input: $\left\{(\boldsymbol{w}_k, B_k, a_k) \mid k \in K_\gamma\right\}, [\boldsymbol{x}]$

for all $k \in K_\gamma$ **do**
  $[s_k] \leftarrow \prod_{i=1}^{d} [x_i]^{w_{k,i}}$
end for
Pick random permutation function $\pi_s(\cdot)$
$[\boldsymbol{s}'] \leftarrow \pi_s([\boldsymbol{s}])$
invoke $[\hat{\boldsymbol{y}}'] \leftarrow$ CALCULATE-SIGMA$([\boldsymbol{s}'])$ at PSP
$[\hat{\boldsymbol{y}}] \leftarrow \pi_s^{-1}([\hat{\boldsymbol{y}}'])$
for all $k \in K_\gamma$ **do**
  Re-randomize $[\hat{y}_k]$
  $[b_k] \leftarrow B_k([\hat{y}_k])$
  send $[a_k]_u, [b_k], [\hat{y}_k], [k]$ to PSP
end for

procedure CALCULATE-SIGMA$([\boldsymbol{s}'])$
  Decrypt $[\boldsymbol{s}']$
  for all $s'_i \in \boldsymbol{s}'$ **do**
    $\hat{y}'_i \leftarrow \sigma(s'_i)$
  end for
  return $[\hat{\boldsymbol{y}}']$
end procedure

Fig. 2. Bidding protocol, initiated by every DSP

profile. The bidding phase is initiated by the user contacting an ad exchange with a request for an advertisement. The ad exchange sends a bid request to every DSP, each of which executes the bidding protocol. For each advertising campaign k, the user response $\hat{y}$ is assumed to be estimated using a logistic regression model. The bidding price is derived from $\hat{y}$ using a linear bidding function $B(\hat{y}_k) = c_1\hat{y}_k + c_2$ for campaign-specific constants c_1 and c_2. Although a linear bidding function may not be capable of fully capturing complex bidding strategies used in practice, it is used for illustrative purposes, and could be replaced by another bidding function.

The bidding protocol, as executed by the DSPs, is outlined in Fig. 2. The sigmoid function that is used to make predictions in logistic regression models is non-trivial to compute under additively homomorphic encryption. Existing literature uses two different approaches to compute the result of the sigmoid function: approximation [24], [25], or computation in the clear [26], [27]. We argue that in our setting, revealing the input to the sigmoid function $\boldsymbol{w}^\top\boldsymbol{x}$ to the PSP is acceptable, since it does not leak information about the user's profile as $\boldsymbol{w}$ is not known to the PSP. (Note that, if $\boldsymbol{w}$ is known to the PSP, it could be possible to extract information about $\boldsymbol{x}$. Similarly, if $\boldsymbol{x}$ is known to the PSP, it could be possible to extract information about $\boldsymbol{w}$.) The result of the sigmoid function may leak information about the degree to which the user is interested in a particular topic. However, no information about the identity of the user is passed to the PSP during the bidding phase, and thus the PSP can not infer any more information than that there is a user who is interested in a particular topic. Finally, the result of the sigmoid function does not reveal information about

Input: $\boldsymbol{x}, y, [\hat{y}], [b], [k]$

$[\delta] \leftarrow [\hat{y}] \cdot [-y]$
for $i \leftarrow 1, d$ **do**
  $[g_i] \leftarrow \begin{cases} [\delta]^{x_i} & \text{if } x_i \neq 0 \\ [0] & \text{otherwise} \end{cases}$
end for
Re-randomize $[b], [k]$
invoke AGGREGATE$([\boldsymbol{g}], [k], [b])$ at PSP

procedure AGGREGATE$([\boldsymbol{g}], [k], [b])$
  Decrypt $[k]$
  for $i \leftarrow 1, d$ **do**
    $\left[\hat{g}_{k,i}\right] \leftarrow \left[\hat{g}_{k,i}\right] \cdot [g_i]$
  end for
  $\left[\hat{b}_k\right] \leftarrow \left[\hat{b}_k\right] \cdot [b]$
  if sufficient values are aggregated for campaign k **then**
    invoke UPDATE$(\left[\hat{\boldsymbol{g}}_k\right], k)$ at DSP
  end if
end procedure

procedure UPDATE$([\boldsymbol{g}], k)$
  Decrypt $[\boldsymbol{g}]$
  for $i \leftarrow 1, d$ **do**
    $w_{k,i} \leftarrow w_{k,i} - \eta_k g_i$
  end for
end procedure

Fig. 3. Model update protocol, initiated by a user after viewing an advertisement

bid values to the PSP, as the bidding functions are unknown to the PSP.

D. Auction phase

During the auction phase, the PSP and the ad exchange engage in a secure comparison protocol, such as described in e.g. [28], to select the highest bid and associated advertisement without either party learning which bid won the auction. At the start of the auction phase the PSP holds encryptions of all bids submitted by the DSPs through the bidding protocol. These bids consist of encryptions of the bid value, the advertisement, the predicted response, and the campaign identifier. The bids are randomly permuted by the PSP and sent to the ad exchange. After execution of the secure comparison protocol, the ad exchange holds an encryption of the index of the highest bid, which is decrypted by the ad exchange. This allows the ad exchange to forward the encrypted bid information to the user, who decrypts and displays the advertisement.

E. Model update phase

After an advertisement is shown to a user, the response prediction model associated with the shown advertisement can be updated to learn from the observed user action y, which is either click (1) or no click (0). In the current non-privacy-preserving setting, only clicks are reported, and non-clicks are inferred from the absence of clicks. Since we want to

Authorized licensed use limited to: TU Delft Library. Downloaded on May 27,2020 at 12:25:29 UTC from IEEE Xplore. Restrictions apply.

TABLE II
SYMBOLS USED IN COMPUTATIONAL ANALYSIS

Symbol	Description
ν	Number of non-zero elements in the user's profile.
d	Dimensionality of user profiles.
κ	Number of campaigns of a DSP.
K	Total number of campaigns.
ζ	Number of model updates aggregated per campaign.

TABLE III
NUMBER OF OPERATIONS PERFORMED PER SUBPROTOCOL

Protocol	Operation	User	AdX	DSP	PSP
Profile update	Encryption	ν			$d-\nu$
	Decryption			1	
Bidding	Encryption			2κ	K
	Share decryption			κ	K
	Multiplication			$\kappa(d-1)$	
	Exponentiation			κd	
	Randomization			κ	
	Bidding function			κ	
Auction	Comparison		$K-1$		$K-1$
	Decryption		1		
Model update	Exponentiation	ν			
	Encryption	$d-\nu$			
	Multiplication	1			$1+d$
	Share decryption			$1+\frac{d}{\zeta}$	$1+\frac{d}{\zeta}$

hide whether a user clicked on an advertisement, however, we always report something. Note that we cannot reveal $\hat{y}-y$, since its value leaks information about y, and we cannot reveal $\hat{y}$ to the PSP as the PSP could link that to values observed during the bidding phase. Therefore, we must rely on users to calculate the gradients used for model updates.

The model update protocol is outlined in Fig. 3. The user calculates the gradient of the loss function $\boldsymbol{g} = (\hat{y}-y)\boldsymbol{x}$ in the encrypted domain. DSPs have their model parameters $\boldsymbol{w}$ in the clear, so an update of $\boldsymbol{w}$ based on a single user's response reveals information about the user's profile $\boldsymbol{x}$ to the DSP. Therefore, the user sends $\boldsymbol{g}$ to the PSP, where gradients of multiple users are aggregated before revealing the resulting vector to the DSP. The DSP can then update the response prediction model based on the aggregated gradients of a small batch of users. Moreover, the PSP aggregates bid values received by users on a per-campaign basis, such that advertisers can be billed without revealing individual bid values.

IV. COMPUTATIONAL ANALYSIS

To evaluate the computational complexity of the proposed protocol, we provide both a theoretical analysis in terms of the number of cryptographic operations performed by parties participating in the protocol, and a set of measurements obtained from a proof-of-concept implementation.

A. Computational Complexity

The variables used in the computational analysis are described in Table II. Note that ν, κ, K and ζ are expected to be several orders of magnitude smaller than d. Table III lists the amortized number of operations performed by each party for each subprotocol. While the proposed protocol requires a large number of cryptographic operations, all subprotocols have a complexity at most linear in the number of campaigns and size of user profiles. The bidding protocol requires the largest number of operations due to the computation of $[\boldsymbol{w}^\top \boldsymbol{x}]$. However, this computation is trivially parallelizable and can thus be greatly sped up. Moreover, the required number of operations can be reduced by employing a sparsity-inducing model such as the Follow The (Proximally) Regularized Leader (FTRL) model described in [16]. Further speed improvements can be achieved by packing multiple values into a single ciphertext in the model update phase, reducing the number of encryptions performed by the user, the number of multiplications performed by the DSP and the number of share decryptions performed by the DSP and PSP. Moreover, packing reduces the amount of communication required between the parties.

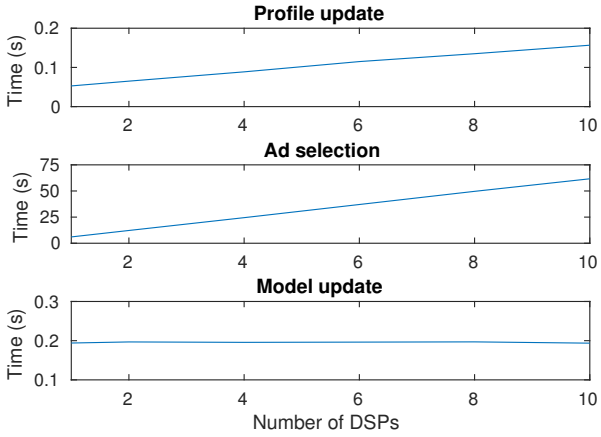

Fig. 4. Total computation time required by a single run of each subprotocol for an increasing number of DSPs, with a fixed profile size $d = 2^{12}$ and number of campaigns per DSP $\kappa = 10$. Model update time is averaged using $\zeta = 10$.

B. Implementation

To measure the runtime of the protocol, we made a proof-of-concept implementation in C++. The implementation simulates all parties within a single process thread, thus performing all operations sequentially. All cryptographic operations use a key length of 2048 bits, and real values such as model weights are represented as 16-bit fixed-point numbers. Furthermore, data packing is used to speed up model updates.

The runtime tests were executed on a mobile workstation running Arch Linux on an Intel® Core™ i7-3610QM 2.3 GHz quad-core processor with 8 GB RAM. Fig. 4 shows the impact of the number of DSPs, and thus the total number of campaigns, on the total computation time. The time spent in the profile update protocol increases linearly with the number of DSPs since a profile update must be processed by each DSP. In a realistic setting, DSPs would operate in parallel, resulting in constant-time performance of the profile update protocol. The bidding and auction protocols cannot be fully parallelized due to the operations performed by the PSP and ad exchange, and thus scale linearly in the number of DSP. Since the model update

Authorized licensed use limited to: TU Delft Library. Downloaded on May 27,2020 at 12:25:29 UTC from IEEE Xplore. Restrictions apply.

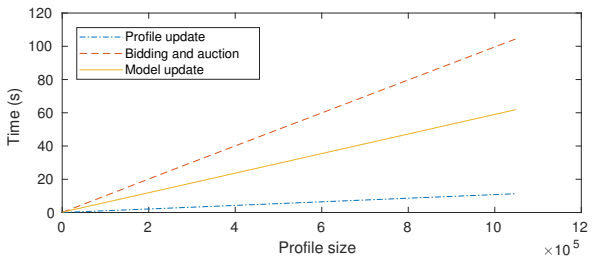

Fig. 5. Total computation time required by a single run of each subprotocol for an increasing profile size d, with a single DSP running a single campaign. Model update time is averaged using $\zeta = 10$.

protocol is only performed once for every advertisement shown, its runtime is independent of the number of DSPs.

Fig. 5 shows the impact of profile dimensionality on the total computation time. It is apparent that, although all subprotocols scale linearly with profile size, high-dimensional profiles as used in practice result in dozens of seconds of computation time for every shown advertisement if not parallelized.

V. Conclusion and Future Work

In this paper we present, to the best of our knowledge, the first protocol using machine learning over encrypted data to preserve privacy in OBA. DSPs must work together with a semi-honest, non-colluding PSP to estimate a user's response and the corresponding bid price within the encrypted domain, after which the PSP and ad exchange run a privacy-preserving auction to select the winning bid. Encrypted reports of ad clicks or views are aggregated by the PSP for billing and model update purposes. At no point are the contents of the user profile or the shown advertisement revealed to any party other than the user themselves, nor are model parameters revealed to any party other than the DSP. Finally, individual bid prices are revealed to no party at all.

The computation time required by AHEad quickly increases with profile dimensionality, resulting in over 100 seconds of computation per bid for $d = 2^{20}$ on our modest hardware. Moreover, an encrypted user profile of size 2^{20} requires 4 Gib of storage space and communication bandwidth between DSPs and the PSP. Nevertheless, our work shows promising initial results in terms of user privacy and achieves performance linear in the profile size, and calls for future research to provide further performance improvements. We are confident that, combined with sufficiently powerful hardware, the resulting protocols could spark a revolution in user privacy in OBA.

References

[1] V. Letang and L. Stillman, "Global Advertising Forecast," MAGNA, 2016.

[2] J. Deighton and H. M. Brierley, "Economic Value of the Advertising-Supported Internet Ecosystem," Interactive Advertising Bureau, 2012.

[3] PageFair, "The state of the blocked web," 2017.

[4] PageFair and Adobe, "The cost of ad blocking," 2015.

[5] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang, "Smart, Useful, Scary, Creepy: Perceptions of Online Behavioral Advertising," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 2012.

[6] M. An, "Why People Block Ads (And What It Means for Marketers and Advertisers)," HubSpot Research, 2016.

[7] TRUSTe, "2011 Consumer Research Results: Privacy and Online Behavioural Advertising," 2011.

[8] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen, "How Much Can Behavioral Targeting Help Online Advertising?" In *Proceedings of the 18th International Conference on World Wide Web*, 2009.

[9] J. Wang, W. Zhang, and S. Yuan, "Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting," *ArXiv e-prints*, 2016. arXiv: 1610.03013 [cs.GT].

[10] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. Weippl, "Block me if you can: A large-scale study of tracker-blocking tools," in *2nd IEEE European Symposium on Security and Privacy*, 2017.

[11] M. Degeling and T. Herrmann, "Your Interests According to Google - A Profile-Centered Analysis for Obfuscation of Online Tracking Profiles," *ArXiv e-prints*, 2016. arXiv: 1601.06371 [cs.CY].

[12] F. Papaodyssefs, C. Iordanou, J. Blackburn, N. Laoutaris, and K. Papagiannaki, "Web Identity Translator: Behavioral Advertising and Identity Privacy with WIT," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, 2015.

[13] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy Preserving Targeted Advertising.," in *Network and Distributed System Symposium*, 2010.

[14] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "Online advertising: Analysis of privacy threats and protection approaches," *Computer Communications*, 2017.

[15] 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*, 2012.

[16] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica, "Ad Click Prediction: A View from the Trenches," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[17] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. Candela, "Practical Lessons from Predicting Clicks on Ads at Facebook," in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 2014.

[18] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and Scalable Response Prediction for Display Advertising," *ACM Trans Intell Syst Technol*, 2015.

[19] A. Szwabe, P. Misiorek, and M. Ciesielczyk, "Logistic regression setup for RTB CTR estimation," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, 2017.

[20] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature Hashing for Large Scale Multitask Learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[21] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology — EUROCRYPT '99*, 1999.

[22] C. Hazay, G. L. Mikkelsen, T. Rabin, and T. Toft, "Efficient RSA Key Generation and Threshold Paillier in the Two-Party Setting," in *Topics in Cryptology – CT-RSA 2012*, 2012.

[23] M. Fredrikson and B. Livshits, "RePriv: Re-imagining Content Personalization and In-browser Privacy," in *2011 IEEE Symposium on Security and Privacy*, 2011.

[24] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy Preserving Deep Computation Model on Cloud for Big Data Feature Learning," *IEEE Trans. Comput.*, 2016.

[25] T. Chen and S. Zhong, "Privacy-Preserving Backpropagation Neural Network Learning," *IEEE Trans. Neural Netw.*, 2009.

[26] C. Orlandi, A. Piva, and M. Barni, "Oblivious Neural Network Computing via Homomorphic Encryption," *EURASIP J Inf Secur*, 2007.

[27] Y. Aono, T. Hayashi, L. Trieu Phong, and L. Wang, "Scalable and Secure Logistic Regression via Homomorphic Encryption," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, 2016.

[28] M. Nateghizad, Z. Erkin, and R. L. Lagendijk, "An efficient privacy-preserving comparison protocol in smart metering systems," *EURASIP J. on Info. Security*, 2016.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 27,2020 at 12:25:29 UTC from IEEE Xplore. Restrictions apply.