What's the Gist?
Privacy-Preserving Aggregation of User Profiles

Igor Bilogrevic[1], Julien Freudiger[2], Emiliano De Cristofaro[3], Ersin Uzun[2]
[1] Google, Switzerland[†] [2] PARC, USA [3] UCL, UK[†]

Abstract

Over the past few years, online service providers have started gathering increasing amounts of personal information to build user profiles and monetize them with advertisers and data brokers. Users have little control of what information is processed and are often left with an *all-or-nothing* decision between receiving free services or refusing to be profiled. This paper explores an alternative approach where users only disclose an *aggregate model* – the "gist" – of their data. We aim to preserve data utility and simultaneously provide user privacy. We show that this approach can be efficiently supported by letting users contribute encrypted and differentially-private data to an aggregator. The aggregator combines encrypted contributions and can only extract an aggregate model of the underlying data. We evaluate our framework on a dataset of 100,000 U.S. users obtained from the U.S. Census Bureau and show that (i) it provides accurate aggregates with as little as 100 users, (ii) it generates revenue for both users and data brokers, and (iii) its overhead is appreciably low.

1 Introduction

The digital footprint of Internet users is growing at an unprecedented pace, driven by the pervasiveness of online interactions and large number of posts, likes, check-ins, and content shared everyday. This creates invaluable sources of information that online service providers use to profile users and serve targeted advertisement. This economic model, however, raises major privacy concerns [11, 16, 40] as advertisers might excessively track users, data brokers might illegally market consumer profiles [45], and governments might abuse their surveillance power [17, 18] by obtaining datasets collected for other purposes (i.e., monetization). Consequently, consumer advocacy groups are promoting policies and legislations providing greater control to users and more transparent collection practices [30, 40].

Along these lines, several efforts – such as OpenPDS, personal.com, Sellbox, and Handshake – advocate a novel, user-centric paradigm: users store their personal information in "data vaults", and directly manage with whom to share their data. This approach has several advantages, namely, users maintain data ownership (and may monetize their data), while data brokers and advertisers benefit from more accurate and detailed personal information [26, 42]. Nevertheless, privacy still remains a challenge as users need to trust data vaults operators and relinquish their profiles to advertisers [7, 38].

To address such concerns, the research community has proposed to maintain data vaults on user devices and share data in a privacy-preserving way. Prior work can be grouped into three main categories: (1) serving ads locally, without revealing any information to advertisers/data brokers [19, 28, 41]; (2) relying on a

[†] Work done while authors were at PARC.

trusted third party to anonymize user data [4, 35]; and (3) relying on a trusted third party for private user data aggregation [2, 9, 10]. Unfortunately, these approaches suffer from several limitations. First, localized methods prevent data brokers and advertisers from obtaining user statistics. Second, anonymization techniques provide advertisers with significantly reduced data utility and are prone to re-identification attacks [29]. Finally, existing private aggregation schemes rely on a trusted third party for differential privacy (e.g., a proxy [10], a website [2], or mixes [9]; also, aggregation occurs after decryption, thus making it possible to link contributions and users.

Motivated by the above challenges, this paper proposes a novel approach to privacy-preserving aggregation of user data. Rather than contributing data *as-is*, users combine their data into *an aggregate model* – the "gist." Intuitively, users contribute encrypted and differentially-private data to an aggregator that extracts a statistical model of the underlying data (e.g., probability density function of the age of contributing users). Our approach addresses issues with existing work in that it does not depend on a third-party for differential privacy, incurs low computational overhead, and addresses linkability issues between contributions and users. Moreover, we propose a metric to dynamically value user statistics according to their inherent amount of "valuable" information (i.e., sensitivity): for instance, aggregators can assess whether age statistics in a group of participants are more sensitive than income statistics. To the best of our knowledge, our solution provides the first privacy-preserving aggregation scheme for personal data monetization.

Our contributions can be summarized as follows:

1. We design a privacy-preserving framework for monetizing user data, where users trade an aggregate of their data instead of actual values.

2. We define a measure of the sensitivity of different data aggregates. In particular, we adopt the information-theoretic Jensen-Shannon divergence [24] to quantify the distance between the actual distribution of a data attribute, and a distribution that does not reveal actionable information [15], such as the uniform distribution.

3. We show how to rank aggregates based on their sensitivity, i.e., we design a dynamic valuation scheme based on how much information an aggregate leaks.

We evaluate our privacy-preserving framework on a real, anonymized dataset of 100,000 US users (obtained by the Census Bureau) with different types of attributes. Our results show that our framework (i) provides accurate aggregates with as little as 100 participants, (ii) generates revenue for users and data aggregators depending on the number of contributing users and sensitivity of attributes, and (iii) has low computational overhead on user devices (0.3 ms for each user, independently of the number of participants). Interestingly, we find that data brokers have an incentive to direct their investments on small groups of users representative of a certain population. In summary, our approach provides a novel perspective to the privacy-preserving monetization of personal data, and finds a successful balance between data accuracy for advertisers, privacy protection for users, and incentives for data aggregators.

Paper Organization. The rest of the paper is organized as follows. Next section introduces the system architecture and the problem statement. Then, Section 3 presents our framework and Section 4 reports on our experimental evaluation. After reviewing related work in Section 5, we conclude the paper in Section 6.

2 System Architecture

This section introduces the problem definition and presents participating entities.

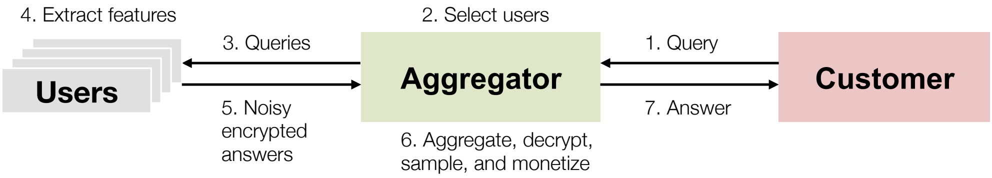


Figure 1: System architecture and basic protocol. Users contribute encrypted profiles to the aggregator. The aggregator combines encrypted profiles and obtains plaintext data models, which it monetizes with customers.

2.1 Problem Statement

We consider a system comprised of three entities: A set of users $\mathbb{U} = \{1, \ldots, N\}$, a data aggregator $\mathbb{A}$, and a customer $\mathbb{C}$. The system architecture is illustrated in Fig. 1. Customers query the data aggregator for user information, while users contribute their personal information to the data aggregator. The aggregator acts as a proxy between users and customers by aggregating (and monetizing) user data. The main goal of this paper is to propose practical techniques to aggregate and monetize user personal data in a privacy-preserving way, i.e., without revealing personal information to other users or third parties.

2.2 System Model

Users. We assume that users store a set of personal attributes such as age, gender, and preferences locally. Each user $i \in \mathbb{U}$ maintains a profile vector $\mathbf{p_i} = [x_{i,1}, \ldots, x_{i,K}]$, where $x_{i,j} \in \mathcal{D}$ is the value of attribute j and $\mathcal{D}$ is a suitable domain for j. For example, if j represents the age of user i, then $x_{i,j} \in \{1, \ldots, M_j\}$, $M_j = 120$, and $\mathcal{D} \subset \mathbb{N}$.

In practice, users can generate their personal profiles manually, or leverage profiles maintained by third parties. Several social networks allow subscribers to download their online profile. A Facebook profile, for example, contains numerous Personally Identifiable Information (PII) items (such as age, gender, relationships, location), preferences (movies, music, books, tv shows, brands), media (photos and videos) and social interaction data (list of friends, wall posts, liked items).

Following the results of recent studies on user privacy attitudes [3, 7, 26], we assume that each user i can specify a privacy-sensitivity value $0 \leq \lambda_{i,j} \leq 1$ for each attribute j. A large $\lambda_{i,j}$ indicates high privacy sensitivity (i.e., lower willingness to disclose). In practice, $\lambda_{i,j}$ can assume a limited number of discrete values, which could represent the different levels of sensitivity according to Westin's Privacy Indexes [22].

We assume that users want to monetize their profiles while preserving their privacy. For instance, users may be willing to trade an *aggregate* of their online behavior, such as the *frequency* at which they visit different *categories* of websites, rather than the exact time and URLs.

Finally, we assume that user devices can perform cryptographic operations consisting of multiplications, exponentiations, and discrete logarithms.

Data Aggregator. A data aggregator $\mathbb{A}$ is an untrusted third-party that performs the following actions: (1) it collects encrypted attributes from users, (2) it aggregates contributed attributes in a privacy-preserving way, and (3) it monetizes users' aggregates according to the amount of "valuable" information that each attribute conveys.

We assume that users and $\mathbb{A}$ sign an agreement upon user registration that authorizes $\mathbb{A}$ to access the aggregated results (but not users' actual attributes), to monetize them with customers, and to take a share of the revenue from the sale. It also binds $\mathbb{A}$ to redistribute the rest of the revenue among contributing users.

Customer. We consider a customer $\mathbb{C}$ willing to obtain aggregate information about users and to pay for it. $\mathbb{C}$ can have commercial contracts with multiple data brokers. Similarly, a data aggregator can have contracts with multiple customers. $\mathbb{C}$ interacts with a data aggregator $\mathbb{A}$ and does not communicate directly with users. $\mathbb{C}$ obtains available attributes, and initiates an aggregation by querying the data aggregator for specific attributes.

2.3 Applications

The proposed system model is well-suited to many real-world scenarios, including market research and online tracking use cases. For instance, consider a car dealer $\mathbb{C}$ that wants to assess user preferences for car brands, their demographics, and income distributions. A data aggregator $\mathbb{A}$ might collect aggregate information about a representative set of users $\mathbb{U}$ and monetize it with the car dealer $\mathbb{C}$. Companies such as Acxiom currently provide this service, but raise privacy concerns [39]. Our solution enables such companies to collect aggregates of personal data instead of actual values and reward users for their participation.

Another example is that of an online publisher (e.g., a news website) $\mathbb{C}$ that wishes to know more about its online readership [2]. In this case, the aggregator $\mathbb{A}$ is an online advertiser that collects information about online users $\mathbb{U}$ and monetizes it with online publishers.

Finally, our proposed model can also be appealing to data aggregators in healthcare [12]. Healthcare data is often fragmented in silos across different organizations and/or individuals. An healthcare aggregator $\mathbb{A}$ can compile data from various sources and allow third parties $\mathbb{C}$ to buy access to the data. At the same time, data contributors ($\mathbb{U}$) receive a fraction of the revenue. Our approach thwarts privacy concerns and helps with the pricing of contributed data.

2.4 Threat Model

In modeling security, we consider both passive and active adversaries.

Passive adversaries. Semi-honest (or honest-but-curious) passive adversaries monitor user communications and try to infer the individual contributions made by other users. For instance, users may wish to obtain attribute values of other users; similarly, data aggregators and customers may try to learn the values of the attributes from aggregated results. A passive adversary executes the protocol correctly and in the correct order, without interfering with inputs or manipulating the final result.

Active adversaries. Active (or malicious) adversaries can deviate from the intended execution of the protocol by inserting, modifying or erasing input or output data. For instance, a subset of malicious users may collude with each other in order to obtain information about other (honest) users or to bias the result of the aggregation. To achieve their goal, malicious users may also collude with either the data aggregator or with the customer. Moreover, a malicious data aggregator may collude with a customer in order to obtain private information about the user attributes.

3 Monetizing User Profiles with Privacy

We outline and formalize the data monetization framework, which consists of a protocol that is executed between users $\mathbb{U}$, a data aggregator $\mathbb{A}$ and a customer $\mathbb{C}$. We first provide an intuitive description and then detail each individual component.

3.1 High-Level Description

We propose a protocol where users trade personal attributes in a privacy-preserving way, in exchange for (possibly) monetary retributions. Intuitively, there are two possible modes of implementations: *interactive* and *batch*.

In interactive mode, a customer initiates a query about specific attributes and users. The aggregator selects users matching the query, collects encrypted replies, computes aggregates, and monetizes them according to a pricing function.

In batch mode, users send their encrypted profile, containing personal attributes, to the data broker. The aggregator combines encrypted profiles, decrypts them, obtains aggregates for each attribute, and ranks attributes based on the amount of "valuable" information they provide. A customer is then offered access to specific attributes. Without loss of generality, hereafter we describe the interactive mode.

Initialization: The data aggregator $\mathbb{A}$ and users $i \in \mathbb{U}$ engage in a secure key establishment protocol to obtain individual random secret keys s_j, where s_0 is only known to $\mathbb{A}$ and s_i ($\forall i \in \mathbb{U}$) is only known to user i, such that $s_0 + s_1 + \ldots + s_N = 0$ (this condition is required for the data aggregation described hereafter). Any secure key establishment protocol or trusted dealer can be used in this phase to distribute the secret keys, as long as the condition on their sum is respected. The initialization phase is the same as in [36]. Each user i generates its profile vector $\mathbf{p_i} \in \mathcal{D}^K$ containing personal attributes $j \in \{1, \ldots, K\}$.

1. **Customer Query**: A customer queries the aggregator. The query contains information about the type of aggregates and users. In practice, it could be formatted as an SQL query.

2. **User Selection**: The aggregator selects users based on the customer query. To do so, we consider that users shared some basic information with the aggregator, such as their demographics. Another option is for the aggregator to forward the customer query to users, and let users decide whether to participate or not.

3. **Aggregator Query**: The aggregator forwards the customer's query to the users, together with a public feature extraction function f.

4. **Feature Extraction**: Each user i can optionally execute a public feature extraction function $f : \mathcal{D}^K \rightarrow \mathcal{O}^L$ on $\mathbf{p_i}$, where L is the dimension of the output feature space $\mathcal{O}$, thus resulting in a feature vector $\mathbf{f_i}$. In our implementation, we consider a simple function that extracts the value of an attribute and its square.

5. **Encryption and Obfuscation**: Each user adds noise to $\mathbf{f_i}$, obtaining $\widehat{\mathbf{f_i}}$, and encrypts it. Encryption and obfuscation provide strong guarantees both in terms of data confidentiality and differential privacy [13]. Each user sends the encrypted vector $\mathcal{E}(\widehat{\mathbf{f_i}})$ to $\mathbb{A}$.

6. **Aggregation, Decryption, and Pricing**: $\mathbb{A}$ combines all $\mathcal{E}(\widehat{\mathbf{f_i}})$ and decrypts the result, generating a 2-tuple $(V_j, W_j) \in \mathbb{R}^2$ for each attribute j. These tuples are used to approximate the probability density function of attributes across users. $\mathbb{A}$ uses (V_j, W_j) to create a discrete sampled probability distribution function $\mathrm{d}\mathcal{N}_j$ for each attribute j. $\mathbb{A}$ then computes a distance measure $d_j = d(\mathrm{d}\mathcal{N}_j, \mathrm{d}\mathcal{U}_j) \in [0, 1]$ between $\mathrm{d}\mathcal{N}_j$ and $\mathrm{d}\mathcal{U}_j$, where $\mathrm{d}\mathcal{U}_j$ is a discrete uniform distribution in the interval $[m_j, M_j]$. A small/large distance corresponds to an attribute with low/high information "value", as described later in the text.

   $\mathbb{A}$ determines the cost *Cost*(j) of each attribute j by taking into account both the distances d_j, the number of contributing users, and the price per attribute.

7. **Answer**: $\mathbb{A}$ sends a set of 2-tuples $\{(d_{\rho_z}, Cost(\rho_z))\}_{\rho_z=1}^{K}$ to $\mathbb{C}$, which selects aggregates to purchase. After the purchase, $\mathbb{A}$ obtains a share of the total sale revenue and equally distributes the remainder to users.

3.2 Detailed Description

We detail the functions and primitives for the aggregation and monetization of user data. In this paper, we compute aggregates by estimating the probability density function (*pdf*) of user attributes. We use the Gaussian approximation to estimate *pdf*s for two reasons. First, existing work shows that this will lead to precise aggregates with few users. The CLT [23, 34] states that the arithmetic mean of a sufficiently large number of independent random variables, drawn from distributions of expected value μ and variance σ^2, will be approximately normally distributed $\mathcal{N}(\mu, \sigma^2)$. Second, a Gaussian *pdf* $\mathcal{N}$ is fully defined by two parameters and thus we do not need additional coordination among users (after the initialization phase). For information leakage ranking, we use a well-established information-theoretic distance function.

For conciseness, we focus on the description of privacy-preserving aggregation and pricing (phases 4 to 6, i.e., feature extraction, encryption, aggregation and ranking). With respect to the initialization and query forwarding phases (1-3), our method is general enough and can be adapted to any specific implementation.

3.2.1 Phase 4-5: Feature Extraction and Encryption.

Each user i generates a profile vector $\mathbf{p_i} = [x_{i,1}, \ldots, x_{i,K}]$. Each attribute j takes value $x_{i,j} \in \{m_j, \ldots, M_j\}$, where $m_j, M_j \in \mathbb{Z}_p$ are the minimum and maximum value. Note that as in [36], computations are in cyclic group $\mathbb{Z}_p$ of prime order p. The aggregator also chooses a generator g at random, such that $g \in \mathbb{Z}_p$, and H is a Hash function. Remember that in practice, a user can derive $\mathbf{p_i}$ either from an existing online profile (e.g., Facebook) or by manually entering values $x_{i,j}$. In our evaluation, we use values from the U.S. Census Bureau [43, 44].

We consider a simple feature extraction f that consists in providing x_j and computing x_j^2. Obviously, other feature extraction method may contribute higher-order moments or simply combine attributes together to obtain richer x_j's.

To guarantee (ϵ, δ)-Differential Privacy, each user i adds noise $r_{i,j}, o_{i,j}$ to attribute values sampled from a symmetric Geometric distribution according to Algorithm 1 in [36]. In particular, in the following we add noise to both $x_{i,j}$ and $x_{i,j}^2$, as they will be subsequently combined to obliviously compute the parameters of the model that underlies the actual data:

$$\widehat{x_{i,j}} = x_{i,j} + r_{i,j} \mod p$$

and

$$\widehat{x_{i,j}}^{(2)} = x_{i,j}^2 + o_{i,j} \mod p$$

where p is the prime order [36].

With $\widehat{x_{i,j}}$ and $\widehat{x_{i,j}}^{(2)}$, each user generates the following encrypted vectors $(\mathbf{c_i}, \mathbf{b_i})$:

$$\mathbf{c_i} = \begin{pmatrix} c_{i,1} \\ c_{i,2} \\ \vdots \\ c_{i,K} \end{pmatrix} = \begin{pmatrix} g^{\widehat{x_{i,1}}} H(t)^{s_i} \\ g^{\widehat{x_{i,2}}} H(t)^{s_i} \\ \vdots \\ g^{\widehat{x_{i,K}}} H(t)^{s_i} \end{pmatrix}, \qquad \mathbf{b_i} = \begin{pmatrix} b_{i,1} \\ b_{i,2} \\ \vdots \\ b_{i,K} \end{pmatrix} = \begin{pmatrix} g^{\widehat{x_{i,1}}^{(2)}} H(t)^{s_i} \\ g^{\widehat{x_{i,2}}^{(2)}} H(t)^{s_i} \\ \vdots \\ g^{\widehat{x_{i,K}}^{(2)}} H(t)^{s_i} \end{pmatrix}$$

Each user i then sends $(\mathbf{c_i}, \mathbf{b_i})$ to $\mathbb{A}$. Note that the encryption scheme guarantees that $\mathbb{A}$ is unable to decrypt the vectors $(\mathbf{c_i}, \mathbf{b_i})$. However, thanks to its own secret share s_0, $\mathbb{A}$ can decrypt aggregates as explained hereafter.

3.2.2 Phase 6: Privacy-Preserving Aggregation and Pricing.

To compute the sample mean $\widehat{\mu_j}$ and variance $\widehat{\sigma_j^2}$ without having access to the individual values $\widehat{x_{i,j}}, \widehat{x_{i,j}}^{(2)}$ of any user i, $\mathbb{A}$ first computes intermediate 2-tuple (V_j, W_j):

$$V_j = H(t)^{s_0} \Pi_{i=1}^{N} c_{i,j} = H(t)^{\sum_{k=0}^{N} s_k} g^{\sum_{i=1}^{N} \widehat{x_{i,j}}} = g^{\sum_{i=1}^{N} \widehat{x_{i,j}}}$$

$$W_j = H(t)^{s_0} \Pi_{i=1}^{N} b_{i,j} = H(t)^{\sum_{k=0}^{N} s_k} g^{\sum_{i=1}^{N} \widehat{x_{i,j}}^{(2)}} = g^{\sum_{i=1}^{N} \widehat{x_{i,j}}^{(2)}}$$

To obtain $(\widehat{\mu_j}, \widehat{\sigma_j^2})$, $\mathbb{A}$ takes the discrete logarithm base g of (V_j, W_j):

$$\widehat{\mu_j} = \frac{\log_g(V_j)}{N} = \frac{\sum_{i=1}^{N} \widehat{x_{i,j}}}{N}$$

$$\widehat{\sigma_j^2} = \frac{\log_g(W_j)}{N} - \widehat{\mu_j}^2 = \frac{\sum_{i=1}^{N} \widehat{x_{i,j}}^{(2)}}{N} - \widehat{\mu_j}^2$$

Finally, using the derived $(\widehat{\mu_j}, \widehat{\sigma_j^2})$, $\mathbb{A}$ computes the Normal *pdf* approximation $\mathcal{N}_j \sim \mathcal{N}(\widehat{\mu_j}, \widehat{\sigma_j^2})$ for each attribute j.

Ranking. In order to estimate the amount of "valuable" information (i.e., sensitivity) that each attribute leaks, we propose to measure the *distance* (i.e., divergence) between the Normal approximation $\mathcal{N}_j$ and the Uniform distribution $\mathcal{U}$. This makes sense because divergence measures distance between distributions: By comparing $\mathcal{N}_j$ to the Uniform, we measure how much information $\mathcal{N}_j$ leaks compared to the distribution $\mathcal{U}$ that leaks the least amount of information [20]. This approach applies to a variety of computing scenarios. For example, a related concept was studied in [15, 21] for measuring the "interestingness" of textual data by comparing it to an expected model, usually with the Kullback-Liebler (KL) divergence.

To the best of our knowledge, we are the first to explore this approach in the context of information privacy. Instead of the KL divergence, we rely on the Jensen-Shannon (JS) divergence for two reasons: (1) JS is a symmetric and (2) bounded equivalent of the KL divergence. It is defined as:

$$JS(u, q) = \frac{1}{2} KL(u, m) + \frac{1}{2} KL(q, m) = H(\frac{1}{2} u + \frac{1}{2} q) - \frac{1}{2} H(u) - \frac{1}{2} H(q)$$

where $m = u/2 + q/2$ and H is the Shannon entropy. As JS is in $[0, 1]$ (when using the logarithm base 2), it quantifies the relative distance between $\mathcal{N}_j$ and $\mathcal{U}_j$, and also provides absolute comparisons with distributions different from the uniform.

As JS operates on discrete values, $\mathbb{A}$ must first discretize distributions $\mathcal{N}_j$ and $\mathcal{U}_j$. Given the knowledge of intervals $\{m_j, \ldots, M_j\}$ for each attribute j, we can use Riemann's centered sum to approximate a definite integral, where the number of approximation bins is related to the accuracy of the approximation. We choose the number of bins to be $M_j - m_j$, and thus guarantee a bin width of 1. We approximate $\mathcal{N}_j$ by the discrete random variable d$\mathcal{N}_j$ with the following probability mass function:

$$\mathbf{Pr}(\mathrm{d}\mathcal{N}_j) = \begin{pmatrix} Pr(x_j = m_j) \\ Pr(x_j = m_j + 1) \\ \vdots \\ Pr(x_j = M_j) \end{pmatrix} = \begin{pmatrix} pdf_j(\frac{1}{2}(m_j + m_j - 1)) \\ pdf_j(\frac{1}{2}(m_j + 1 + m_j)) \\ \vdots \\ pdf_j(\frac{1}{2}(M_j + M_j - 1)) \end{pmatrix}$$

where pdf_j is the probability density function $\mathcal{N}_j$ and $x_j \in \{m_j, \ldots, M_j\}$. We then normalize $\mathbf{Pr}(\mathrm{d}\mathcal{N}_j)$ such that $\sum_k Pr(x_j = k) = 1$, for each j. For the uniform distribution $\mathcal{U}_j$, the discretization to $\mathrm{d}\mathcal{U}_j$ is straightforward, i.e., $\mathbf{Pr}(\mathrm{d}\mathcal{U}_j) = (1/(M_j - m_j), \ldots, 1/(M_j - m_j))^T$, where $\dim(\mathrm{d}\mathcal{U}_j) = M_j - m_j$.

$\mathbb{A}$ can now compute distances $d_j = JS(\mathrm{d}\mathcal{N}_j, \mathrm{d}\mathcal{U}_j) \in [0, 1]$ and rank attributes in increasing order of information leakage such that $d_{\rho_1} \leq d_{\rho_2} \leq \ldots \leq d_{\rho_K}$, where $\rho_1 = \arg\min_j d_j$ and ρ_z (for $2 \leq z \leq K$) are defined as $\rho_z = \arg\min_{j \neq \{\rho_k\}_{k=1}^{z-1}}(d_j)$

At this point, $\mathbb{A}$ computed the 3-tuple $(d_{\rho_j}, \widehat{\mu_j}, \widehat{\sigma_j^2})$ for each attribute j. Each user i can now decide whether it is comfortable sharing attribute j given distance d_j and privacy sensitivity $\lambda_{i,j}$. To do so, each user i sends $\lambda_{i,j}$ to $\mathbb{A}$ for comparison. $\mathbb{A}$ then checks which users are willing to share each attribute j and updates the ratio $\gamma_j = S_j/N$, where S_j is the number of users that are comfortable sharing, i.e., $S_j = |\{i \in \mathbb{U} \text{ s.t. } d_j \leq 1 - \lambda_{i,j}\}|$. In practice, $\mathbb{A}$ could then use the majority rule to decide whether or not to monetize attribute j.

Pricing. After this ranking phase, the data broker $\mathbb{A}$ concludes the process with the pricing and revenue phases. Prior work shows that users assign unique monetary value to different types of attributes depending on several factors, such as offline/online activities [7], type of third-parties involved [7], privacy sensitivity [3], amount of details and fairness [26].

We measure the value of aggregates depending on their sensitivity, the number of contributing users, and the cost of each attribute. Without loss of generality, we estimate the value of an aggregate j using the following linear model:

$$Cost(j) = Price(j) \cdot d_j \cdot N$$

where $Price(j)$ is the monetary value that users assign to attribute j. Without loss of generality, we assume in our pricing scheme a relative value of 1 for each attribute. Existing work discussed the value of user attributes, and estimated a large range from \$ 0.0005 to \$33 [7, 31] highlighting the difficulty in determining a fixed price. In practice, this is likely to change depending on the monetization scenario.

$\mathbb{A}$ then sends the set of 2-tuples $\{(d_{\rho_z}, Cost(\rho_z))\}_{\rho_z=1}^{K}$ to $\mathbb{C}$. Based on the tuples, $\mathbb{C}$ selects the set $\mathbb{P}$ of attributes it wishes to purchase. After the purchase is complete, $\mathbb{A}$ re-distributes revenue R among users and itself, according to the agreement stipulated with the users upon their first registration with $\mathbb{A}$.

We consider a standard revenue sharing monetization scheme, where the revenue is split among users and the data aggregator (i.e., aggregator takes commissions):

$$R(\mathbb{A}) = \sum_{j \in \mathbb{P}} \omega_j \cdot Cost(j), \qquad R(i) = \frac{1}{N} \sum_{j \in \mathbb{P}} (1 - \omega_j) \cdot Cost(j), \qquad \forall i \in \mathbb{U}$$

where ω_j is the commission percentage of $\mathbb{A}$. This system is popular in existing aggregating schemes [12], credit-card payments, and online stores (e.g., iOS App Store). We assume a fixed ω_j for each attribute j.

4 Evaluation

To test the relevance and the practicality of our privacy-preserving monetization solution, we measure the quality of aggregates, the overhead, and generated revenue. In particular, we study how the number of protocol participants and their privacy sensitivities affect the accuracy of the Gaussian approximations, the computational performance, the amount of information leaked for each attribute, and revenue.

		Number of randomly selected users in the dataset					
		10	**100**	**1k**	**10k**	**50k**	**100k**
Income	μ	6.50	9.72	10.30	10.87	10.83	10.89
	σ	19.17	18.70	20.04	17.05	16.72	16.52
	(m_j, M_j)	(1, 10)	(1, 15)	(1, 15)	(1, 16)	(1, 16)	(1, 16)
Educ.	μ	5.70	7.23	10.29	10.38	10.21	10.18
	σ	15.57	7.07	7.96	7.68	7.73	7.63
	(m_j, M_j)	(1, 9)	(1, 12)	(1, 15)	(1, 16)	(1, 16)	(1, 16)
Age	μ	38.10	35.40	41.91	42.44	41.49	39.79
	σ	252.54	502.79	563.32	546.40	553.68	539.60
	(m_j, M_j)	(11, 67)	(1, 85)	(0, 85)	(0, 85)	(0, 85)	(0, 85)

Table 1: Summary of the U.S. Census dataset used for the evaluation. We considered three types of attributes (level of income, education and age), which reflect different types of sample distributions (as shown in Fig. 2).

4.1 Setup

We consider secret shares in $\mathbb{Z}_p$ where p is a 1024 bits modulus, the number of users $N \in [10, 100000]$, and each user i with profile $\mathbf{p_i}$. We implemented our privacy-preserving protocol in Java, and rely on public libraries for secret key initialization, for multi-threading decryption, and on the MALLET [27] package for computation of the JS divergence.

We run our experiments on a machine equipped with Mac OSX 10.8.3, dual-core Core i5 processor, 2.53 GHz, and 8 GB RAM. Measurements up to 100 users are averaged over 300 iterations, and the rest (from 1k to 100k users) are averaged over 3 iterations due to large simulation times.

We populate user profiles with U.S. Census Bureau information [43, 44]: We obtained anonymized offline and online attributes about 100,000 people. We pre-processed the acquired data by removing incomplete profiles (i.e., some respondents prefer not to reveal specific attributes).

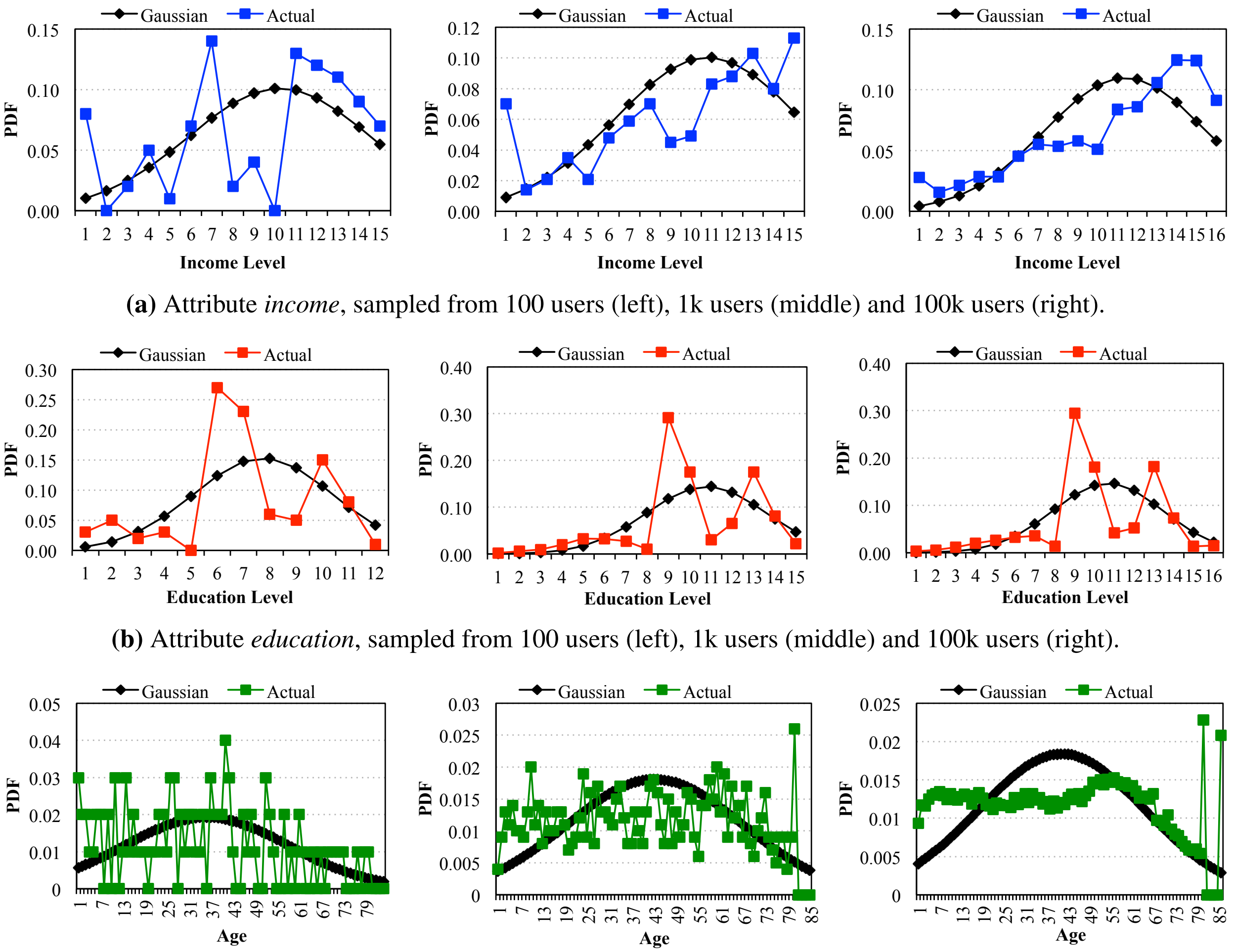
Without loss of generality, we focus on three types of offline attributes: Yearly *income* level, *education* level and *age*. We selected these attributes because (1) a recent study [7] shows that these attributes have high monetary value (and thus privacy sensitivity), and (2) they have significantly different distributions across users. This allows us to compare retribution models, and measure the accuracy of the Gaussian approximation for a variety of distributions.

Table 1 shows the mean and standard deviation for the three considered attributes with a varying number of users. Note that the provided values for *income* and *education* use a specific scale defined by the Census Bureau. For example, a value of 1 and 16 for *education* correspond to “Less than 1st grade” and “Doctorate”, respectively.

We could consider other types of attributes as well, such as internet, music and video preferences from alternative sources, such as Yahoo Webscope [47]. Although an exhaustive comparison of the monetization of all different attributes is an exciting perspective, it is out of the scope of this paper and we leave this for future work.

4.2 Results

We evaluate four aspects of our privacy-preserving scheme: model accuracy, information leakage, overhead and pricing.



(a) Attribute *income*, sampled from 100 users (left), 1k users (middle) and 100k users (right).

(b) Attribute *education*, sampled from 100 users (left), 1k users (middle) and 100k users (right).

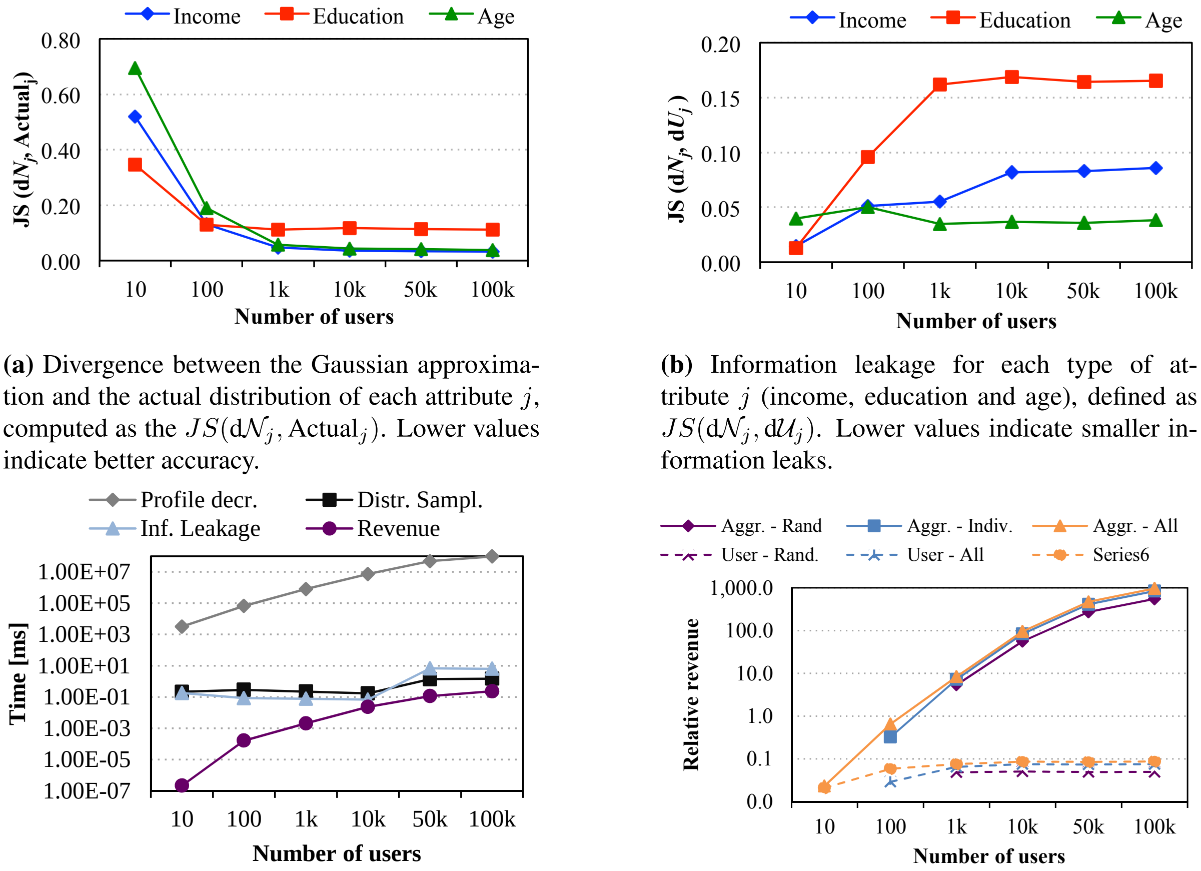
(c) Attribute *age*, sampled from 100 users (left), 1k users (middle) and 100k users (right).

Figure 2: Gaussian approximation vs. actual distribution for each considered attribute.

4.2.1 Model Accuracy

In our proposal, we approximate empirical probability density functions with Gaussian distributions. The accuracy of approximations is important to assess the relevance of derived data models. In Fig. 2, we compare the actual distribution of each attribute with their respective Gaussian approximation and vary the number of users from 100 to 100,000. Note that in order to compare probabilities over the domain $[m_j, M_j]$, we scaled both the actual distribution and the Gaussian approximation such that their respective sums over that domain are equal to one. We observe that, visually, the Gaussian approximation captures general trends in the actual data.

We measure the accuracy of the Gaussian approximation in more details with the JS divergence (Fig. 3a). We observe that with 100 users, the approximation reaches a plateau for *education*, whereas *income* and *age* require 1k users to converge. For the two latter attributes, the approximation accuracy triples when increasing from 100 to 1k users. Moreover, as the number of user increases, the fit of the Gaussian model for *income* and *age* is two times better (JS of 0.05 bits) than for *education* (JS of 0.1 bits). The main reason is that *education* has more data points with large differences between actual and approximated distributions



(a) Divergence between the Gaussian approximation and the actual distribution of each attribute j, computed as the $JS(\mathrm{d}\mathcal{N}_j, \text{Actual}_j)$. Lower values indicate better accuracy.

(b) Information leakage for each type of attribute j (income, education and age), defined as $JS(\mathrm{d}\mathcal{N}_j, \mathrm{d}\mathcal{U}_j)$. Lower values indicate smaller information leaks.

(c) Performance measurements for each of the four phases of the protocol performed by the data broker.

(d) Relative revenue (per attribute) for each user $i \in \mathbb{U}$ and the data aggregator $\mathbb{A}$, assuming that an attribute is valued at 1.

Figure 3: Results of the evaluation of the proposed framework on the U.S. Census dataset.

than *income* and *age* (as shown in Fig. 2).

These results indicate that, for non-uniform distributions, the Gaussian approximation is accurate with a relatively small number of users (about 100). It is interesting to study this result in light of the Central Limit Theorem (CLT). Remember that the CLT states that the arithmetic mean of a sufficiently large number of variables will tend to be normally distributed. In other words, a Gaussian approximation quickly converges to the original distribution and this confirms the validity of our experiments. This also means that $\mathbb{C}$ can obtain accurate models even if it requests aggregates about small groups of users. In other words, collecting data about more than 1k users does not significantly improve the accuracy of approximations, even for more extreme distributions.

4.2.2 Information Leakage

We compare the divergence between Gaussian approximations and uniform distributions to measure the information leakage of different attributes. Fig. 3b shows the sensitivity for each attribute with a varying number of users. We observe that the amount of information leakage stabilizes for all attributes after a given number of participants. In particular, *education* and *age* reach a maximum information leakage with 1k users, whereas 10k users are required for *income* to achieve the same leakage.

Overall, we observe that *education* is by far the attribute with the largest distance to the uniform distribution, and therefore arguably the most valuable one. In comparison, *Income* and *age* are 50% and 75% less "revealing". Information leakage for *age* decreases from 100 to 1k users, as age distribution in our dataset tends towards a uniform distribution. In contrast, *education* and *income* are significantly different from a uniform distribution. An important observation is that the amount of valuable information does not increase monotonically with the number of users: For *age*, it decreases by 30% when the number of users increases from 100 to 1k, and for *education* it decreases by 3% when transitioning from 10k to 50k users.

These findings show that larger user samples do not necessarily provide better discriminating features. This also shows that users should not decide whether to participate in our protocol solely based on a fixed threshold over total participants, as this may prove to leak slightly more private information.

4.2.3 Overhead

We measure the computation overhead for both users and the data broker. For each user, we find that one execution of the protocol requires 0.284 ms (excluding communication delays), out of which 0.01 ms are spent for the profile generation, 0.024 ms for the feature extraction, 0.026 ms for the differential-privacy noise addition, and 0.224 ms for encryption of the noisy attribute. In general, user profiles are not subject to change within short time intervals, thus suggesting that user-side operations could be executed on resource-constrained devices such as mobile phones.

From Fig. 3c, observe that the data broker requires about one second to complete its phases when there are only 10 users, 1.5 min with 100 users, 15 min with 1k users, and 27.7 h for 100k users. Note, however, that running times can be remarkably reduced using algorithmic optimization and parallelization, which is part of our future work. In our results, decryption is the most time-consuming operation for the data broker as it incurs ($O(N \cdot M_j)$): this could be reduced to $O(\sqrt{N \cdot M_j})$ by using the Pollard's Rho method for computing the discrete logarithm [33]. Also, decryption can be speedup up by splitting decryption operations across multiple machines (i.e., the underlying algorithm is highly-parallelizable).

4.2.4 Pricing

Recall that the price of an attribute aggregate depends on the number of contributing users, the amount of information leakage, and the cost of the attribute. We consider that each attribute j has a unit cost of 1 and the data broker takes a commission ω_j. We consider three types of privacy sensitivities λ: (i) a uniform random distribution of privacy sensitivities $\lambda_{i,j}$ for each user i and for each attribute j, (ii) an individual privacy sensitivity λ_i for each user (same across different attributes), and (iii) an all-share scenario ($\lambda_i = 0$ and all users contribute). The commission percentage is set to $\omega_j = \omega = 0.1$.

Fig. 3d shows the average revenue generated from one attribute by the data broker and by users. We observe that user revenue is small and does not increase with the number of participants. In contrast, the data broker revenue increases linearly with the number of participants. In terms of privacy sensitivities, we observe that with higher privacy sensitivities ($\lambda_i > 0$), fewer users contribute, thus generating lower revenue overall and per user. For example, users start earning revenue with 10 participants in the all-share scenario, but more users are required to start generating revenue if users adopt higher privacy sensitivities.

We observe that users are incentivized to participate as they earn some revenue (rather than not benefiting at all), but the generated revenue does not generate significant income, thus, it might encourage user participation from "biased" demographics (e.g., similar to Amazon Mechanical Turk). In contrast, the data broker has incentives to attract more users, as it revenue increases with the number of participants. However, customers are incentivized to select fewer users because cost increases with the number of users, and 100 users provide as good an aggregate as 1000 users. This is an intriguing result, as it encourages customers to focus on small groups of users representative of a certain population category.

4.3 Security

Passive adversaries. To ensure privacy of the personal user attributes, our framework relies on the security of the underlying encryption and differential-privacy methods presented in [36]. Hence, no passive adversary (a user participating in the monetization protocol, the data aggregator or an external party not involved in the protocol) can learn any of the user attributes, assuming that the key setup phase has been performed correctly and that a suitable algebraic group (satisfying the DDH assumption) with a large enough prime order (1024 bits or more) has been chosen.

Active adversaries. As per [36], our framework is resistant to collusion attacks among users and between a subset of users and the data broker, as each user i encrypts its attribute values with a unique and secret key s_i. However, pollution attacks, which try to manipulate the aggregated result by encrypting out-of-scope values, can affect the aggregate result of our protocol. Nevertheless, such attacks can be mitigated by including, in addition to encryption, range checks based on efficient (non-interactive) zero-knowledge proofs of knowledge [5, 6, 25]: each user could submit, in addition to the encrypted values, a proof that such values are indeed in the plausible range specified by the data aggregator. However, even within a specific range, a user can manipulate its contributed value and thus affect the aggregate. Although nudging users to reveal their true attribute value is an important challenge, it is outside of the scope of this paper.

5 Related Work

Our work builds upon two main domains, in order to provide the privacy and incentives for the users and data aggregators: (1) privacy-preserving aggregation [14, 36, 37, 46], and (2) privacy-preserving monetization of user profiles [4, 19, 35, 41]. Hereafter we discuss these two sets of works.

5.1 Privacy-Preserving Aggregation

Erkin and Tsudik [14] design a method to perform privacy-preserving data aggregation in the smart grid. Smart meters jointly establish secret keys without having to rely on a trusted third party, and mask individual readings using a modified version of the Paillier encryption scheme [32]. The aggregator then computes the sum of all readings without seeing individual values. Smart meters must communicate with each other, thus limiting this proposal to online settings. Shi et al. [37] compute the sum of different inputs based on data slicing and mixing with other users, but have the same limitation: all participants must actively communicate with each other during the aggregation.

Another line of work [9, 10] introduces privacy-preserving aggregation by combining homomorphic encryption and differential privacy, i.e., users encrypt their data with the customer public key and send it to a trusted aggregator. The aggregator adds differential noise to encrypted values (using the homomorphic property), and forwards the result to the customer. The customer decrypts contributions and computes desired aggregates. These proposals, however, suffer from a number of shortcomings as: (i) they rely on a trusted third party for differential privacy; (ii) they require at least one public key operation per single bit of user input, and one kilobit of data per single bit of user answer, or rely on XOR encryption; and (iii) contributions are linkable to users as aggregation occurs after decryption.

The work by Shi et al. [36] supports computing the sum of different inputs in a privacy-preserving fashion, without requiring communication among users, nor repeated interactions with a third party. It also provides differential privacy guarantees in presence of malicious users, and establishes an upper bound on the error induced by the additive noise. This work formally shows that a Geometric distribution provides (ϵ, δ)-differential privacy (DD) in $\mathbb{Z}_p$. We extend the construction in [36] to support the privacy-preserving computation of probability distributions (in addition to sums). Intuitively, we use the proposed technique to compute the parameters of Gaussian approximations in a privacy-preserving way. As we maintain the same security assumptions, our framework preserves provable privacy properties. As part of future work, we intend to explore the properties of regression modeling and privacy-preserving computation of regression parameters [1, 46], in addition to distributions.

5.2 Privacy-Preserving Monetization

Previous work investigated two main approaches to privacy-preserving Online Behavioral Advertisement (OBA). The first approach minimizes the data shared with third parties, by introducing local user profile generation, categorization, and ad selection [2, 19, 28, 41]. The second approach relies on anonymizing proxies to shield users' behavioral data from third parties, until users agree to sell their data [4, 35].

Toubiana et al. [41] propose to let users maintain browsing profiles on their device and match ads with user profiles, based on a cosine-similarity measure between visited websites meta-data (title, URL, tags) and ad categories. Users receive a large number of ads, select appropriate ones, and share selected ads with ad providers (not revealing visited websites nor user details). Guha et al. [19] propose to do the ad matching with an anonymization proxy instead. Although the cost of such system is estimated at $0.01/user per year, such solution demands significant changes from web browser vendors and online advertisers. Akkus et al. propose to let users rely on the website publisher to anonymize their browsing patterns *vis-à-vis* the ad-provider. Their protocol introduces significant overhead: The website publisher must repeatedly interact with each visitor and forward encrypted messages to the ad-provider.

Instead of local profiles, Riederer et al. [35] propose a fully centralized approach, where an anonymization proxy mediates interactions between users and website publishers. The proxy releases the mapping between IP addresses and long-term user identifiers only after users agree to sell their data to a customer, thus allowing the customer to link different visits by the same users. However, users have to entrust a third party with their personal information.

In contrast, our framework does not rely on any additional user-side software, does not impose computationally expensive cryptographic computation on user devices, and prevents the customer from learning individual user data.

6 Conclusion

As the amount and sensitivity of personal data shared with service providers increases, so do privacy concerns. Users usually have little control over what information is processed by service providers and how it is monetized with advertisers. Our work offers a privacy-preserving alternative where users only disclose an aggregate model of their profiles, by means of encrypted and differentially private contributions. Our solution tackles trust and incentive challenges: rather than selling data *as-is*, users trade a *model* of their data. Users also monetize their profiles by dynamically assessing the value of data aggregates. To this end, we use an information-theoretic measure to compute the amount of valuable information provided to advertisers.

We evaluate our framework on a real and anonymized dataset with more than 100,000 users (obtained from the U.S. Census Bureau) and show, with an experimental evaluation, that our solution (i) provides accurate aggregates with as little as 100 users, (ii) introduces low overhead for both users (less than 1ms on commodity hardware) and data aggregators, and (iii) generates revenue for both users and aggregators.

As part of future work, we plan to enhance our scheme with new features, including fault-tolerant aggregation [8], which can be integrated in order to allow users to join/leave dynamically without disrupting the scheme. Also, range checks for the encrypted user attributes, based on efficient zero-knowledge proofs, could thwart active pollution attacks. Users could also contribute higher order moments (e.g., x^3 or x^4) for the aggregator to obtain more precise approximations using moment-generating functions (an alternative to pdfs). Finally, we intend to investigate schemes for targeting ads to users contributing data to the aggregation, by allowing the aggregator to select specific subgroups of users according to the customer's target population.

References

[1] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han. Privacy-aware regression modeling of participatory sensing data.

[2] I. E. Akkus, R. Chen, M. Hardt, P. Francis, and J. Gehrke. Non-tracking Web Analytics. In *ACM CCS*, 2012.

[3] C. Aperjis and B. A. Huberman. A market for unbiased private data: Paying individuals according to their privacy attitudes. *ArXiv Report 1205.0030*, 2012.

[4] M. Backes, A. Kate, M. Maffei, and K. Pecina. Obliviad: Provably secure and practical online behavioral advertising. In *IEEE Security and Privacy*, 2012.

[5] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *ACM STOC*, 1988.

[6] F. Boudot. Efficient proofs that a committed number lies in an interval. In *Eurocrypt*, 2000.

[7] J. P. Carrascal, C. Riederer, V. Erramilli, M. Cherubini, and R. de Oliveira. Your Browsing Behavior for a Big Mac: Economics of Personal Information Online. In *WWW*, 2013.

[8] T.-H. H. Chan, E. Shi, and D. Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography and Data Security*, 2012.

[9] R. Chen, I. E. Akkus, and P. Francis. SplitX: High-performance Private Analytics. In *ACM SIGCOMM*, 2013.

[10] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke. Towards statistical queries over distributed private user data. In *NSDI*, 2012.

[11] ComRes. Big Brother Watch Online Privacy Survey. http://www.comres.co.uk/polls/Big_Brother_Watch_Online_Privacy_Survey.pdf, 2013.

[12] DataCommons. Partner Organizations Helping to Advance Healthcare. http://mydatacommons.org, 2014.

[13] C. Dwork. Differential privacy. In *ICALP*, 2006.

[14] Z. Erkin and G. Tsudik. Private computation of spatial and temporal power consumption with smart meters. In *ACNS*, 2012.

[15] R. Feldman and I. Dagan. Knowledge Discovery in Textual Databases. In *KDD*, 1995.

[16] G. Flood. Online Privacy Worries Increasing Worldwide. InformationWeek, http://www.informationweek.co.uk/security/privacy/online-privacy-worries-increasing-worldw/240153200, 2013.

[17] B. Gellman and L. Poitras. US Intelligence Mines Data from Internet Firms in Secret Program. The Washington Post, http://wapo.st/J2gkLY, 2013.

[18] G. Greenwald and E. MacAskill. NSA Prism program taps in to user data of Apple, Google and others. The Guardian, http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data, 2013.

[19] S. Guha, B. Cheng, and P. Francis. Privad: practical privacy in online advertising. In *NSDI*, 2011.

[20] H. J. Hamilton, R. J. Hilderman, and N. Cercone. Attribute-oriented induction using domain generalization graphs. In *IEEE ICTAI*, 1996.

[21] R. J. Hilderman, H. J. Hamilton, and B. Barber. Ranking the interestingness of summaries from data mining systems. In *FLAIRS Conference*, 1999.

[22] P. Kumaraguru and L. F. Cranor. Privacy Indexes: A Survey of Westins Studies. *Institute for Software Research International*, 2005.

[23] P. S. Laplace. *Mémoire sur les approximations des formules qui sont fonctions de très-grands nombres, et sur leur application aux probabilités*. Baudouin, 1810.

[24] J. Lin. Divergence measures based on the shannon entropy. *IEEE TIT*, 37(1), 1991.

[25] H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *Asiacrypt*, 2003.

[26] M. Malheiros, S. Preibusch, and M. A. Sasse. “Fairly truthful”: The impact of Perceived Effort, Fairness, Relevance, and Sensitivity on Personal Data Disclosure. In *TRUST*, 2013.

[27] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[28] P. Mohan, S. Nath, and O. Riva. Prefetching Mobile Ads: Can Advertising Systems Afford It? In *EuroSys*, 2013.

[29] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Security and Privacy*, 2008.

[30] Natasha. Congress to examine data sellers. The New York Times, http://nyti.ms/Pewbq1, 2012.

[31] L. Olejnik, T. Minh-Dung, and C. Castelluccia. Selling Off Privacy at Auction. In *NDSS*, 2014.

[32] P. Paillier. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT*, 1999.

[33] J. M. Pollard. Monte carlo methods for index computation. *Mathematics of computation*, 32(143), 1978.

[34] J. A. Rice. *Mathematical statistics and data analysis*. Wadsworth & Brooks/Cole, 1988.

[35] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez. For Sale: Your Data: By: You. In *HotNets*, 2011.

[36] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-Preserving Aggregation of Time-Series Data. In *NDSS*, 2011.

[37] J. Shi, R. Zhang, Y. Liu, and Y. Zhang. Prisense: Privacy-preserving Data Aggregation in People-Centric Urban Sensing Systems. In *IEEE INFOCOM*, 2010.

[38] R. Singel. Encrypted E-Mail Company Hushmail Spills to Feds. http://www.wired.com/threatlevel/2007/11/encrypted-e-mai/, 2007.

[39] N. Singer. Mapping, and sharing, the consumer genome. The New York Times, http://nyti.ms/LcBw0g, 2012.

[40] E. W. Tanzina Vega. U.s. agency seeks tougher consumer privacy rules. The New York Times, http://nyti.ms/GQQCrY, 2012.

[41] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *NDSS*, 2010.

[42] A. Tunner. Bizarro World Of Hilarious Mistakes Revealed In Long Secret Personal Data Files Just Opened. Forbes, http://onforb.es/1rZ5PZQ, 2013.

[43] U.S. Census Bureau. DataFerrett Analysis and Extraction Tool. http://dataferrett.census.gov.

[44] U.S. Government. The home of the U.S. Government's open data. www.data.gov.

[45] E. Wyatt. U.S. Penalizes Online Company in Sale of Personal Data. The New York Times, http://nyti.ms/OsDrgI, 2012.

[46] K. Xing, Z. Wan, P. Hu, H. Zhu, Y. Wang, X. Chen, Y. Wang, and L. Huang. Mutual privacy-preserving regression modeling in participatory sensing. In *IEEE INFOCOM*, 2013.

[47] Yahoo Labs. Webscope. http://webscope.sandbox.yahoo.com.