HyObscure: Hybrid Obscuring for Privacy-Preserving Data Publishing

Xiao Han, Yuncong Yang, Junjie Wu, and Hui Xiong, Fellow, IEEE

Abstract—Minimizing privacy leakage while ensuring data utility is a critical problem in a privacy-preserving data publishing task, from which data holders can boost platform engagements or enlarge data values. Most prior research concerned only with either privacy-insensitive or exact private data and resorted to a single obscuring method to achieve a privacy-utility tradeoff, which is inadequate for real-life hybrid data especially when facing machine learning-based inference attacks. This work takes a pilot study on privacy-preserving data publishing when both widely adopted generalization and obfuscation operations are employed for privacy-heterogeneous data protection. Specifically, we first propose novel measures for privacy and utility values quantification and formulate the hybrid privacy-preserving data obscuring problem to account for the joint effect of generalization and obfuscation. We then design a novel protection mechanism called HyObscure, which decomposes the original problem into three sub-problems to cross-iteratively optimize the hybrid operations for maximum privacy protection under a certain data utility guarantee. The convergence of the iterative process and the privacy leakage bound of HyObscure are also provided in theory. Extensive experiments demonstrate that HyObscure significantly outperforms a variety of state-of-the-art baseline methods when facing various inference attacks in different scenarios.

Index Terms—Attribute inference attack, generalization, hybrid obscuring, obfuscation, privacy preserving data publishing.

Manuscript received 3 July 2022; revised 30 September 2023; accepted 26 October 2023. Date of publication 9 November 2023; date of current version 12 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 72071125, 72031001, 72242101, and 92370204, and in part by the Science and Technology Planning Project of Guangdong Province under Grant 2023A0505050111. Recommended for acceptance by X. Yi. (Corresponding author: Junjie Wu.)

Xiao Han is with the Key Laboratory of Interdisciplinary Research of Computation and Economics (Shanghai University of Finance and Economics), Ministry of Education, Shanghai 200433, China, and also with the School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China, and also with the Dishui Lake Advanced Finance Institute, Shanghai University of Finance and Economics, Shanghai 200433, China (e-mail: xiaohan@mail.shufe.edu.cn).

Yuncong Yang is with the Key Laboratory of Interdisciplinary Research of Computation and Economics (Shanghai University of Finance and Economics), Ministry of Education, Shanghai 200433, China, and also with the School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China (e-mail: yycphd@163.sufe.edu.cn).

Junjie Wu is with the Key Laboratory of Data Intelligence and Management (Beihang University), the Ministry of Industry and Information Technology, Beijing 100191, China, and also with the School of Economics and Management, Beihang University, Beijing 100191, China (e-mail: wuj@buaa.edu.cn).

Hui Xiong is with the Thrust of Artificial Intelligence, HKUST (Guangzhou), Guangzhou, Guangdong 511458, China, also with the Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China, and also with Guangzhou HK/ST Fok Ying Tung Research Institute, China (e-mail: xionghui@ust.hk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2023.3331568, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2023.3331568

I. INTRODUCTION

In the big data era, data publishing has become a popular way to facilitate data exploitation and enlarge economic values of data [1], [2], [3]. Leading data holders like Facebook and Twitter provide APIs to share data with third-parties with the purpose of increasing platform engagements [4]. More and more data holders nowadays are prone to publishing data, such as Mimic database [5], MovieLens [6], and Yelp challenges [7], to seek worldwide help in data exploitation. Indeed, data publishing as positive externality has enabled service innovation, scientific discovery, and other public benefits, which generate enormous economic values amounting to over $3 trillion annually [4].

While data publishing creates substantial benefits, privacy leakage due to improper sharing of user data may get data holders into real troubles, even the situation involving multi-billion dollar fines or lawsuits [8]. As a result, it is a critical task for data holders to protect data privacy while exploring sustainable economic benefits from data. In practice, data holders typically restrict the access to exact private data that users are reluctant to share (e.g., age or home address), and release privacy-insensitive data (e.g., movie ratings) that users agree to open in return for favorable services [5]. In recent years, generalized private data (e.g., age stage or living city) become publicly available in many published datasets [6], [7]. They cause few direct privacy breaches but are required by or helpful to services like health management or personalized recommendation. Fig. 1 visualizes a few data samples of two public data sets, in which both privacy-insensitive data and generalized private data are presented.

Recent years have witnessed the unprecedented development of artificial intelligence technology, implying that the above-mentioned data publishing routines may no longer truly protect data privacy, especially when facing the emerging attribute inference attacks [8], [9], [10], [11], [12]. For instance, users’ hidden attributes including ages, political views and locations are likely to be predicted by users’ publicly available data, such as their “likes”, posts, and relationships in online social media [13], [14], which could seriously degrade the effect of private data generalization. In this light, how to achieve privacy-preserving data

\[1\] https://mimic.mit.edu/docs/siv/

\[2\] https://netflixprize.com/

\[3\] https://www.yelp.com/dataset/challenge

\[4\] https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/open-data-unlocking-innovation-and-performance-with-liquid-information

\[5\] https://www.forbes.com/sites/mnunez/2019/07/24/ftcs-unprecedented-slap-fines-facebook-5-billion-forces-new-privacy-controls/
are the two most widely adopted privacy metrics. It decomposes the complex hybrid obscuring problem into three sub-problems, namely $I$-problem (Initialization), $O$-problem (Obfuscation) and $G$-problem (Generalization), and proposes a framework called HyObscure to optimize the obfuscation and generalization operations in a cross-iterative manner. A theoretic proof of the convergence of HyObscure and a theoretic bound for the privacy leakage are also presented rigorously. A theoretic proof of the convergence of HyObscure and a theoretic bound for the privacy leakage are also presented rigorously.

We conduct extensive experiments on two real-life datasets to validate the effectiveness of HyObscure, with the challenges from two representative attribute inference attack methods and under two attacker prior knowledge scenarios. Results demonstrate that HyObscure provides consistently much better privacy-utility tradeoffs compared with state-of-the-art baselines, scales linearly, and is robust to parameters.

The remainder of this work is organized as follows. In Section II, we review the related literature on privacy-preserving data publishing. We formulate our privacy-preserving data publishing problem in Section III and introduce HyObscure in Section IV. In Section V, we evaluate the effectiveness and efficiency of HyObscure with extensive experiments. We finally conclude the work in Section VI.

II. LITERATURE REVIEW

Privacy preservation is a critical concern from the standpoints of both data publisher and users when releasing data [5]. Traditional privacy-preserving data publishing is mainly studied in the database community, which often transform the fine-grained attribute value to the coarse one with generalization techniques. The other challenge comes from identifying suitable obfuscation and generalization functions that can jointly optimize the privacy-utility tradeoff, which was not a problem for single obscuring approaches [4], [20]. Our work overcomes the above challenges and makes the following contributions:

- It formulates a hybrid privacy-preserving data publishing problem, which aims to minimize the expected privacy leakage under a guaranteed data utility loss. To our best knowledge, this is among the earliest work to investigate hybrid obscuring techniques for privacy-preserving data publishing.
- It proposes two delicately designed obscuring functions to synchronously obfuscate users’ privacy-insensitive data and generalize their private data. New methods to quantify the levels of both privacy leakage and data utility loss for reaching a compromise are also carefully designed.

Obscuring data to trade data utility for privacy protection is a general idea to accomplish privacy-preserving data publishing against attribute inference attacks [15]. A few recent studies focus on designing optimal obscuring methods, e.g., obfuscation [4] or noise addition [16], for privacy-insensitive data. These methods, however, did not take into account the simultaneously published generalized private data and thus might be vulnerable to attribute inference attacks. Some other studies obscure privacy-sensitive and insensitive data together by generalization methods [17]. They group users with similar attributes and represent the users in a group with the same generalized attribute values, which becomes practically impossible when facing a large number of attributes [18]. As a result, we should consider the correlations between privacy-insensitive and exact private data and design synthetic obscuring methods to obtain a joint effect of privacy protection.

It is intuitive to apply hybrid obscuring operations, e.g., the widely adopted generalization and obfuscation, to heterogeneous data where both privacy-insensitive and exact private data are present. For instance, generalization could be applied to private attributes of a small set so that the operation is efficient and no exact private information is published. Obfuscation, commonly regarded as a feasible obscuring operation to high-dimensional data [19], could be applied to privacy-insensitive data to ensure fine-grained data publishing. Along this line, we propose a hybrid obscuring solution that obfuscates privacy-insensitive data and generalizes private data synchronously for favorable privacy-preserving data publishing.

There are yet two major challenges to hybrid obscuring. The first one is to quantify the data utility loss and the privacy gain in hybrid operations so that a good tradeoff can be reached. Existing methods are mostly designed for a single obscuring operation [19], [20], which might interfere with one another if employed separately for privacy-heterogeneous data.

| User | Preferences on movies (privacy-insensitive data) | Age stage (generalized private data) |
|------|-----------------------------------------------|-------------------------------------|
|      | Hamilton  | Psycho  | Alice  | WALL-E | Coco   |        |
| U1   | Y         | N       | N      | N      | Y      | 20-29  |
| U2   | Y         | Y       | N      | N      | Y      | 20-29  |
| U3   | Y         | N       | N      | Y      | Y      | 30-39  |
| U4   | N         | Y       | Y      | N      | N      | 30-39  |

(a) Samples of the published MovieLens dataset, where users’ preferences on movies (privacy-insensitive data) are directly published and users’ ages (privacy) are generalized to age stages (generalized private data) for publishing.

| Patient | Laboratory measurements (privacy-insensitive data) | Year group of Birth (generalized private data) |
|---------|-----------------------------------------------|-------------------------------------|
|         | Heart rate | Diastolic pressure | Glucose | Respiratory rate | Creatinine |        |
| P1      | 65         | 60.7               | 7.6     | 13.5            | 88.4       | 1940-1942 |
| P2      | 71.0       | 62.1               | 6.80    | 14.5            | 70.7       | 1957-1960 |
| P3      | 76         | 78.2               | 6.90    | 19.0            | 58.9       | 1974-1975 |
| P4      | 106.9      | 54.3               | 5.91    | 16.9            | 48.6       | 1998-2000 |

(b) Samples of the published Mimic-IV database, where patients’ laboratory measurements (privacy-insensitive data) are directly published and their ages (privacy) are generalized to birth year groups (generalized private data) for publishing.

Fig. 1. Two public datasets containing both generalized private data and privacy-insensitive data.
The recent breakthrough in machine learning brings many astonishing challenges on information protection [4, 24]. The classical machine learning task uses original real data to learn the patterns in data which may cause data disclosure to attackers. Then, state-of-the-art work studies privacy-preserving machine learning schemes by allowing data patterns to be jointly learned from encrypted data without sharing any real information [25]. Moreover, a variety of machine learning based malicious attacks, such as attribute inference attack [20, 26], membership inference attack [27, 28, 29], and property inference attack [30], appear and may lead to private data leakage or data membership disclosure through the published data and the well-trained model. This work is one of those focusing on data publishing techniques against attribute inference attacks.

Delicately obscuring data before publishing is a typical idea to preserve private data from being inferred through published data. In particular, obfuscating data or adding noises to original data are common means to trade off utility and privacy for privacy-insensitive data [4, 20, 24]. Despite the proposals of optimal obscuring techniques on privacy-insensitive data, prior studies may still lead to severe privacy leakage if the generalized private data are concurrently published. While generalization operation preserves privacy by reducing data precision (i.e., making data coarse-grained), it is rarely applied to privacy-insensitive data as fine-grained privacy-insensitive data are often favorable. More seriously, generalization becomes practically impossible when it is applied to a large number of privacy-insensitive attributes (e.g., users’ preferences on thousands of movies) [18]. Therefore, this work proposes to apply hybrid data obscuring operations, i.e., obfuscating privacy-insensitive data and generalizing private data, for privacy-preserving data publishing. More specifically, our work is the pioneering study focusing on hybrid obscuring, which simultaneously optimizes obfuscation (on privacy-insensitive data) and generalization (on private data).

We also notice another rising research line that uses differential privacy (DP) mechanisms to accomplish privacy-preserving goals [31, 32]. Generally, DP aims to make the differentiable probability of any two true private data records through their obscured values be bounded within a privacy budget $\epsilon$, although it often employs the obscuring methods such as noise addition or obfuscation. For instance, DP mechanism may ensure arbitrary two locations $l_1$ and $l_2$ have a similar probability to be observed to $l_o$, so that a user’s real location at $l_1$ cannot be distinguished from $l_2$ by observing $l_o$ [32]; in this case, DP mechanism is not purposely designed to protect private location information from being inferred through other knowledge (e.g., profession or activities) about users. Comparatively, we intend to protect private data from being inferred through other relevant published information. It is clear that DP has a different goal from ours and cannot protect the private data records from attribute inference attacks [20].

Quantification of privacy leakage and utility loss is a critical process during data obscuring optimization for privacy preservation. In literature, information-theoretic metrics, including $\epsilon$-differential privacy [31, 32], mutual information [33], Kullback-Leibler divergence [34], conditional entropy [15, 23], are often used to quantify the privacy leakage, while utility loss of a privacy-preserving published data set is often measured by distance metrics (e.g., Hamming and $l_2$-norm, Jensen–Shannon divergence [34]). However, the extant privacy leakage and utility loss measurements are designed for a single obscuring operation, and thus cannot precisely measure the overall privacy or utility change when hybrid obscuring methods are applied on privacy-insensitive and private sensitive data respectively. In this vein, we propose new privacy leakage and utility loss measurements to adapt to our hybrid obscuring context.

### III. Problem Formulation

In this section, we formulate the hybrid privacy-preserving data publishing problem where both users’ privacy-insensitive data and exact private data are given. Basically, by designing an obfuscation function $\mathcal{O}_{\hat{X}|X}$ for users’ privacy-insensitive data $X$ and a generalization function $\mathcal{G}_{\hat{Y}|Y}$ for users’ private data $Y$, we expect to obtain two objectives: 1) to preserve the data utility for downstream data analytical tasks, and 2) to protect the exact private data from attribute inference attacks when the obfuscated privacy-insensitive data $\hat{X}$ and the generalized private data $\hat{Y}$ are published concurrently. In what follows, we successively define the relevant obscuring operations, quantify the levels of data utility and privacy leakage, and then formulate our problem. Table I summarizes the math notations used throughout this paper.

#### A. Obfuscation and Generalization

As an exploratory hybrid obscuring study for privacy-preserving data publishing, we apply obfuscation on privacy-insensitive data and generalization on private attributes owing to their appealing properties. Briefly, for the private attributes whose precise values are not allowed to be released, data deletion and generalization are two common ways to preserve the exact value of the original data. In practice, many data regulations suggest generalizing private data into coarse-grain values for publishing, as the generalized data can still preserve some utility but the deleted data cannot. For instance, the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule\(^6\) requires privacy-sensitive information such as age and ZIP code to be generalized before data release.

---

\(^6\)https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html

---

| Table I  | Summary of Math Notations |
|---------|---------------------------|
| Notation | Description               |
| $X$ ($\hat{X}$) | Original (obfuscated) insensitive data. |
| $Y$ ($\hat{Y}$) | Original (generalized) private data. |
| $\mathcal{O}_{X|X}$ | Obfuscation function. |
| $\mathcal{G}_{\hat{Y}|Y}$ | Generalization function. |
| $f(\cdot)$ | Mutual information. |

---

| $X$ ($\hat{X}$) | Original (obfuscated) insensitive data. |
| $Y$ ($\hat{Y}$) | Original (generalized) private data. |
| $\mathcal{O}_{X|X}$ | Obfuscation function. |
| $\mathcal{G}_{\hat{Y}|Y}$ | Generalization function. |
| $f(\cdot)$ | Mutual information. |
As for the privacy-insensitive data that can be released publicly, data obfuscation and noise addition are preferably adopted to keep the fineness of data [4], [16]. In this study, we opt for using obfuscation as it can hold data actuality by selecting proper data substitutions from the raw data, whereas the noise addition method will produce unreal values that never exist in the original data. Nevertheless, we believe our hybrid obscuring work with obfuscation and generalization can shed light on the privacy-utility tradeoff with other types of obscuring methods. Next, we respectively define the obfuscation, generalization, and our proposed hybrid operations.

**Definition 1 (Obfuscation):** By designing a probabilistic obfuscation function \( O_{X|\hat{X}} \) given an obfuscation budget, obfuscation techniques replace users’ privacy-insensitive data \( X \) with others’ data \( \hat{X} \) by certain probabilities, so as to reduce the disclosure probability of users’ private information under a guarantee of data utility [4], [24].

**Definition 2 (Generalization):** A generalization function \( G_{Y|\hat{Y}} \) maps an arbitrary exact value of a private attribute \( y \in Y \) into a generalized value \( \hat{y} \in \hat{Y} \). Each generalized value \( \hat{y} \in \hat{Y} \) associates with a group of exact values, denoted as \( Y_{\hat{y}} \). For instance, the exact age 38 may be generalized to an age stage of 30–40. The obscured groups of exact private values are usually required to satisfy users’ privacy requirements such as \( K \)-anonymity [21] and \( L \)-diversity [22].

**Definition 3 (Hybrid Obfuscation and Generalization):** Given users’ privacy-insensitive data \( X \) and private data \( Y \), \( X \) is obfuscated and \( Y \) is generalized for publishing, so that the exact value of \( Y \) is preserved and the fineness of \( X \) is maintained. In particular, to keep the privacy-insensitive value distribution \( X_{\hat{y}} \) invariant after obfuscation for \( \hat{y} \in \hat{Y} \), the obfuscation of \( X \) is only allowed between users with the same \( \hat{y} \in \hat{Y} \) [35].

### B. Privacy Leakage Computation

If the obfuscated privacy-insensitive data \( \hat{X} \) and generalized private data \( \hat{Y} \) are published, attackers may perform attribute inference attacks to correctly infer users’ private data \( Y \) based on the published data, which will incur privacy leakage on \( Y \).

**Definition 4 (Attribute Inference Attack):** Given \([\hat{X}, \hat{Y}]\) as the published data of a set of users \( U \), an attribute inference attack is to infer users’ exact values of \( Y \) with the objective of minimizing the expected inference loss:

\[
q^* = \arg \min_q E_{Y|\hat{X}}[L(Y, q) | \hat{X}, \hat{Y}],
\]

where \( L \) denotes the loss function, and \( q^* \) is the preferable inference method obtained by solving the attribute inference attack problem.

In practice, adversaries may utilize any inference method \( q \) to optimize their attack by minimizing the expected loss. A stronger \( q \) that produces a smaller expected loss will enable more accurate inference of private information, thus exacerbating privacy leakage. Consequently, we formally define privacy leakage as the difference between the expected losses before and after the

adversaries obtain the public information \([\hat{X}, \hat{Y}]\):

\[
\Delta \text{loss}_{\text{privacy}} = \min_q E_Y[L(Y, q)] - \min_{q, \hat{X}} E_{Y|\hat{X}}[L(Y, q) | \hat{X}, \hat{Y}].
\]

By quantifying how much additional inference power the public data grants adversaries, this formulation provides a principled metric for privacy leakage that applies for any inference method \( q \) and associated loss function \( L \).

In practice, it hardly enumerates all the potential inference methods and loss functions for privacy leakage computation; therefore, we first adopt the log-loss function as \( L \) in (2), which has been widely used in general inference tasks and is deemed consistent with “rational” adversaries’ goals [33]. The expected loss of adversaries without any public information using log-loss can be rewritten as:

\[
E_Y[L(Y, q)] = \sum_{y \in Y} -p(y) \log p_q(y) \geq \sum_{y \in Y} -p(y) \log p(y) = H(Y).
\]

where \( y \in Y \) is an exact value of a private attribute, \( p(y) \) is the actual data distribution regarding \( y \), and \( p_q(y) \) denotes the predicted data distribution regarding \( y \) using inference method \( q \). Similarly, the expected loss of adversaries observing the public information \( \hat{X} \) and \( \hat{Y} \) equals:

\[
E_{Y|\hat{X}}[L(Y, q) | \hat{X}, \hat{Y}] = \sum_{y \in Y} -p(y|\hat{X}, \hat{Y}) \log p_q(y|\hat{X}, \hat{Y}) \geq \sum_{y \in Y} -p(y|\hat{X}, \hat{Y}) \log (p(y|\hat{X}, \hat{Y})) = H(Y|\hat{X}, \hat{Y}).
\]

The conditional entropy \( H(Y|\hat{X}, \hat{Y}) \) is the minimal expected inference loss \( \min_q E_{Y|\hat{X}, \hat{Y}}[L(Y, q) | \hat{X}, \hat{Y}] \) that the strongest adversaries with public information \([\hat{X}, \hat{Y}]\) can reach. According to (2)–(4), we have:

\[
\Delta \text{loss}_{\text{privacy}} = \min_{q, \hat{X}} E_Y[L(Y, q)] - \min_{q, \hat{X}} E_{Y|\hat{X}}[L(Y, q) | \hat{X}, \hat{Y}] = H(Y) - H(Y|\hat{X}, \hat{Y}) = I(Y; \hat{X}, \hat{Y}).
\]

In brief, when the log-loss function is used, \( \Delta \text{loss}_{\text{privacy}} \) equals the mutual information between the published data \([\hat{X}, \hat{Y}]\) and the exact values of private attributes \( Y \), which can be computed following Theorem 1 below (we leave proofs to the supplemental document for concision).

**Theorem 1:** The expected privacy leakage on \( Y \) caused by the publicly released \([\hat{X}, \hat{Y}]\) is a function of \( O_{X|\hat{X}} \) and \( G_{Y|\hat{Y}} \), which is given by

\[
\Delta \text{loss}_{\text{privacy}} = I(Y; \hat{X}, \hat{Y}) = \sum_{y \in Y} G_{Y|\hat{Y}}(y|y') p_Y(y') \sum_{\hat{x} \in \hat{X}} O_{X|\hat{X}}(\hat{x}|x, \hat{y}) p_{XY}(x, y|\hat{y})
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[
\log \sum_{x' \in X_X} \mathcal{O}_{X|X}(\hat{x}, \hat{y}) p_{XY}(x', y) \left( \frac{1}{p_Y(y)} \right) - \sum_{y \in Y} p_Y(y) \log p_Y(y),
\]

(6)

where \( p_Y(y') \), \( p_{XY}(x, y) \) and \( \sum_{y \in Y} p_Y(y) \log p_Y(y) \) can be calculated given \( \mathcal{O}_{X|Y} \) and \( \mathcal{G}_{Y|Y} \).

We will also discuss in Section III-E that minimizing mutual information could defend against adversary inference algorithms with other loss functions.

C. Data Utility Maintenance

Obfuscation on privacy-insensitive data \( X \) may diminish the utility of \( X \), while generalization on private data \( Y \) often lowers the utility of \( Y \). Data utility maintenance thus becomes an associated and important problem of privacy protection.

1) Maintaining Utility for Obfuscation: Data utility loss caused by obfuscation is typically measured by the expected difference between the obfuscated and original data, i.e., \( \mathbb{E}_{X \hat{X}} [d(X, \hat{X})] \), where \( d \) is a difference calibration function, such as \( l_2 \)-norm, Kullback-Leibler divergence [36], and Jensen-Shannon divergence [34].

In our case of hybrid obfuscation and generalization, as the obfuscation of \( X \) is only allowed between users with the same generalized private value \( \hat{y} \in \hat{Y} \), the utility loss of obfuscation can be computed by the expected distance between \( X \) and \( \hat{X} \) given \( Y \), which is given by

\[
\mathbb{E}_{X \hat{X}} [d(X, \hat{X})] = \mathbb{E}_{\hat{y} \in \hat{Y}} \mathbb{E}_{x, \hat{x} | \hat{y}} [d(x, \hat{x})].
\]

(7)

In this vein, we can control the data utility loss while minimizing the expected privacy leakage.

2) Maintaining Utility for Generalization: Concerning that data utility loss due to generalization closely relates to the degree of generalization, we define two measures for maintaining data utility against generalization as follows.

Definition 5 ((k, \( \alpha \))-uniqueness): It controls the number of users \( |U_\hat{y}| \) given a generalized private attribute value \( \hat{y} \in \hat{Y} \), and is achieved if \( k \leq |U_\hat{y}| \leq \alpha, \forall \hat{y} \in \hat{Y} \).

Definition 6 ((l, \( \beta \))-variety): It maintains the size of exact values \( |Y_\hat{y}| \) regarding a generalized private attribute value \( \hat{y} \in \hat{Y} \), and is achieved if \( l \leq |Y_\hat{y}| \leq \beta, \forall \hat{y} \in \hat{Y} \).

In general, data utility decreases if more users or more exact private attribute values are clustered regarding a generalized value. As a result, \( \alpha \) and \( \beta \) respectively limit the numbers of users and exact private attribute values in a generalized group to retain data utility. However, the privacy is likely violated if the size of users or exact private values in a group is too small. Therefore, \( k \) and \( l \) are set to ensure enough users and exact private attribute values in a generalized group for privacy protection [21], [22].

D. Problem Definition

Given the above definitions, we formulate our hybrid privacy-preserving data obscuring problem as follows:

\[
\min_{\mathcal{O}_{X|Y}, \mathcal{G}_{Y|Y}} I(Y; \hat{X}, \hat{Y})
\]

s.t. \( \mathbb{E}_{X \hat{X}} [d(X, \hat{X})] \leq \Delta Q_X \),

(9)

\( k \leq |U_\hat{y}| \leq \alpha, \forall \hat{y} \in \hat{Y} \),

(10)

\( l \leq |Y_\hat{y}| \leq \beta, \forall \hat{y} \in \hat{Y} \),

(11)

In general, we aim to strike a balance between privacy protection and utility maintenance by taking the minimization of privacy loss due to data publishing as the objective function and taking data utility loss as the constraints of the optimization. The obfuscation function \( \mathcal{O}_{X|X} \) and the generalization function \( \mathcal{G}_{Y|Y} \) are two key optimum operators we seek for solving the constrained optimization problem. In more detail, (9) restricts data utility loss in publishing \( \hat{X} \) instead of \( X \), where \( \Delta Q_X \) denotes the obfuscation budget. (10) and (11) control the utility and meanwhile preserve the privacy of generalized private data \( \hat{Y} \); that is, higher data utility will be retained with smaller \( |U_\hat{y}| \) and \( |Y_\hat{y}| \).

E. Discussion

While our problem assumes that adversaries apply the log-loss function in their attack methods, in fact they may use any inference methods with alternative loss functions. Nevertheless, the utilization of the log-loss function is sensible. On one hand, it is impossible to enumerate all the inference methods as well as loss functions when quantifying the precise privacy leakage. On the other, prior studies have proved that the inference success of any method indeed relates to the mutual information measure [33]. In particular, by Fano’s Inequality, the error probability \( p(Y \neq q^\ast(\hat{X}, \hat{Y})) \) of any inference algorithm \( q^\ast \) that infers \( Y \) given data \( [\hat{X}, \hat{Y}] \) is lower-bounded by

\[
p(Y \neq q^\ast(\hat{X}, \hat{Y})) \geq \frac{H(Y) - I(Y; \hat{X}, \hat{Y}) - 1}{\log |\hat{Y}|}.
\]

(12)

Note that in (12) \( H(Y) \) and \( |\hat{Y}| \) are fixed given certain private information \( Y \). In this light, the lower bound will increase as \( I(Y; \hat{X}, \hat{Y}) \) decreases. In other words, while our problem aims at minimizing \( I(Y; \hat{X}, \hat{Y}) \), it tends to raise the inference error by maximizing its lower bound no matter which inference methods are applied. We will also empirically show that the proposed method can defend against attribute inference attacks using various machine learning models in practice.

IV. HYBRID OBSCURING METHOD

A. Overview

The obfuscation function \( \mathcal{O}_{X|X} \) and generalization function \( \mathcal{G}_{Y|Y} \) are two interleaved sets of decision variables in our objective function ((8)). Since the objective function is not necessarily linear or convex, solving it directly is intractable. We then resort to problem decomposition for a feasible solution. Specifically,
we propose HyObscure to systematically solve the problem in four steps: 1) decompose the hybrid privacy-preserving data obscuring problem into three sub-problems, i.e., I-problem (Initialization), O-problem (Obfuscation) and G-problem (Generalization); 2) solve I-problem to initialize $G_{Y|\tilde{Y}}$; 3) solve O-problem and G-problem in a cross-iterative manner until convergence to obtain $G_{\tilde{Y}|X}$ and $G_{Y|\tilde{Y}}$; 4) conduct obfuscation and generalization on $X$ and $Y$ respectively by $G_{\tilde{Y}|X}$ and $G_{Y|\tilde{Y}}$, and generate $\tilde{X}$ and $\tilde{Y}$ for publishing. Fig. 2 gives an overview of HyObscure.

**B. Problem Decomposition**

The decomposition technique is a general solution to optimization problems with complicated variables. The basic idea is to first break up the problem into a set of sub-problems and then separately solve each one of them. Intrinsically, the objective equation is convex regarding the obfuscation function $O_{X|\tilde{X}}$ when the generalized private data $\tilde{Y}$ are certain; it is also solvable regarding the generalization function $G_{Y|\tilde{Y}}$ if the obfuscated privacy-insensitive data $\tilde{X}$ are fixed. Along this line, we first carefully decompose the intractable problem into three solvable sub-problems, namely O-problem for optimizing the obfuscation function $O_{X|\tilde{X}}$ given $\tilde{Y}$, G-problem for optimizing the generalization function $G_{Y|\tilde{Y}}$ given $\tilde{X}$, and I-problem for initiating the optimization. O-problem and G-problem are listed as follows:

**O – problem**:

$$\min_{O_{X|\tilde{X}}} I(\tilde{X}, \tilde{Y}; Y)$$

s.t. $E_{\hat{y} \in \hat{Y}} E_{\tilde{X}} [d(X, \tilde{X}|\hat{y})] \leq \Delta Q_X$, \hspace{1cm} (13)

**G – problem**:

$$\min_{G_{\tilde{Y}|X}} I(\tilde{X}, \tilde{Y}; Y)$$

s.t. $E_{\hat{y} \in \hat{Y}} E_{\tilde{X}} [d(X, \tilde{X}|\hat{y})] \leq \Delta Q_X$, \hspace{1cm} (14)

$$k \leq |U_\tilde{g}| \leq \alpha, \forall \hat{y} \in \hat{Y},$$

$$l \leq |Y_\hat{g}| \leq \beta, \forall \hat{y} \in \hat{Y},$$

where the objective ((19)) is to minimize the expected sum of within-group distances between an arbitrary element inside each group $\hat{y} \in \hat{Y}$ and the centroid of the group $c_{\hat{y}}$ [38]; (20) and (21) guarantee $(k, \alpha)$ -uniqueness and $(l, \beta)$ -variety, respectively. By solving the I-problem, the derived generalization function $G_{\tilde{Y}|X}$ can be used to generate a good initial generalized private data $\tilde{Y}^0$.

In brief, we decompose our hybrid data obscuring problem into three sub-problems. Next, we successively address these sub-problems.

**C. Generalization Initialization**

This section addresses the I-problem for initializing the generalization function. As aforementioned, I-problem can be regarded as a $K$-means clustering problem of users’ private data with two generalization constraints. The original $K$-means clustering problem (without any constraint) is already NP-hard in a euclidean space [39], so our I-problem is also NP-hard. To this end, we modify a widely-used heuristic $K$-means algorithm [38] to satisfy both $(k, \alpha)$-uniqueness and $(l, \beta)$-variety given a cluster number $K$. Note that $K$ should be set to $\frac{|U|}{\alpha k} \leq K \leq \frac{|U|}{\beta}$, where $|U|$ is the number of users.
Algorithm 1: Initialization ($I$-Problem).

**Input:** $Y$: private attribute values; $K$: number of generalization groups; $k$ and $(k, \alpha)$-uniqueness constraint; $l$ and $(l, \beta)$-variety constraint;

**Output:** $G^0_{Y|Y^*}$: initial generalization function;

1. $\bar{Y}_c \leftarrow \emptyset$;
2. for $g$ in $\{1, K\}$ do
3.     $y_g^* \leftarrow \text{Rand}(Y)$; //randomly select a center for group $y_g$
4.     $\bar{Y}_c \leftarrow \bar{Y}_c \cup \{y_g^*\}$; //the center set of initial generalized groups
5. end
6. $d^* \leftarrow \infty$; $\bar{Y}^* \leftarrow \emptyset$;
7. while true do
8.     $\bar{Y} \leftarrow \emptyset$; $Y^- \leftarrow \emptyset$
9.     for $g$ in $\{1, K\}$ do
10.        $y_g \leftarrow (y_g^*)$; //initialize group $y_g$ with the center $y_g$
11.        $Y^- \leftarrow Y^- \\setminus \{y_g\}$; //the remaining ungrouped private values
12.        $\bar{Y} \leftarrow \bar{Y} \cup \{y_g\}$; //initial generalized groups
13.     end
14.     $S_{k,t} \leftarrow \bar{Y}$; $S_{o,s} \leftarrow \bar{Y}$;
15.     while $Y^- \neq \emptyset$ do
16.         $d_{min}(Y^-, \bar{Y}_c, S)$ searches a pair of an exact private value $y$ and a group $y_g$ from $S$ that attains the smallest distance and satisfies $|U_y| + |U_{y_g}| \leq \alpha$
17.         if $S_{k,t} \neq \emptyset$ then
18.             $y, y_g \leftarrow d_{min}(Y^-, \bar{Y}_c, S_{k,t})$
19.         else if $S_{o,s} \neq \emptyset$ then
20.             $y, y_g \leftarrow d_{min}(Y^-, \bar{Y}_c, S_{o,s})$
21.         else
22.             set $K \leftarrow K + 1$ and restart the algorithm;
23.             return;
24.     end
25.     $y_g \leftarrow y \cup \{y\}$; $Y^- \leftarrow Y^- \setminus \{y\}$
26.     if $|U_{y_g}| \geq k$ and $|y_g| \geq l$ then
27.         $S_{k,t} \leftarrow S_{k,t} \setminus \{y_g\}$
28.     end
29.     if $|U_{y_g}| = \alpha$ or $|y_g| = \beta$ then
30.         $S_{o,s} \leftarrow S_{o,s} \setminus \{y_g\}$
31.     end
32. end
33. $\bar{Y}_c \leftarrow \emptyset$;
34. for $g$ in $\{1, K\}$ do
35.     $y_g^* \leftarrow \text{Center}(y_g)$; //re-identify the group center
36.     $\bar{Y}_c \leftarrow \bar{Y}_c \cup \{y_g^*\}$; //update group center set
37. end
38. $d \leftarrow d_{sum}(Y, \bar{Y}, \bar{Y}_c)$; //calculates the sum of distances between each private attribute value and its group center
39. if $d \geq d^*$ then
40.     break; //if new generalized groups incur the same or a larger sum of distances, stop the algorithm
41. else
42.     $d^* \leftarrow d$; $\bar{Y}^* \leftarrow \bar{Y}$;
43. end
44. $G^0_{Y|Y^*} \leftarrow$ the initial generalization function that maps $Y$ to $\bar{Y}^*$;
45. return $G^0_{Y|Y^*}$.

The $K$-means algorithm [38] is initiated by selecting a center for each cluster randomly; it then repeats the following two steps until a stopping criterion is achieved: 1) assign users’ private data to their closest clusters according to the euclidean distances between the data and the clusters’ centers; 2) re-identify the center for each cluster obtained by step one. To include the constraints of $(k, \alpha)$-uniqueness and $(l, \beta)$-variety, we design an adapted constraint-aware $K$-means method. Basically, when assigning data samples to clusters, our method prioritizes the low-bound requirements that each cluster at least includes $k$ users and $l$ exact private attribute values. In other words, instead of assigning a data sample to the closest cluster directly as the $K$-means method, our method adds a data sample to the closest cluster that has not yet possessed $k$ users or $l$ exact private attribute values. Once a cluster includes enough data samples that satisfy the lower-bound requirements, it will not be assigned more samples until all the clusters satisfy the lower-bound requirements. Furthermore, for maintaining data utility, our method refuses to assign a data sample to a cluster if the cluster would exceed $\alpha$ users or $\beta$ exact private attribute values. Once a data sample is assigned to a cluster, we adjust the cluster center for the next round of data sample assignments.

Algorithm 1 depicts the detailed initial generalization process. Concerning that the proposed generalization process mainly repeats the distance computation between each user and each cluster center in every iteration, its time complexity equals $O(K \cdot t \cdot |U|)$, where $K$ is the number of generalized groups, $t$ denotes the number of iterations, and $|U|$ represents the number of users. In brief, the initial generalization process scales linearly with the user size.

D. Generalization-Aware Obfuscation

This section addresses $O$-problem by proposing a generalization-aware obfuscation approach. The goal is to learn an optimal $\hat{O}_{X|Y}$ for privacy-insensitive data $X$ by minimizing privacy leakage conditioned on a given $G^0_{Y|Y^*}$. While the objective in $O$-problem is convex given $\tilde{Y}$, it cannot be easily solved when the user size is larger than a few hundred. Therefore, instead of learning the obfuscation function based on individual users, we turn to optimize the obfuscation function on user clusters.

Specifically, we first cluster users based on their privacy-insensitive data. Each user is mapped into a user cluster and our individual user-based obfuscation function $O^0_{X|X,Y}$ becomes a user cluster-based obfuscation function $O_{C|C,Y}$. In principle, such a convex optimization problem with several constraints can be solved by many solvers (e.g., CVX [37]) with a computational complexity of $O(|C|^2)$, where $|C|$ is the number of user clusters (see Algorithm 2).

E. Generalization With Stochastic Privacy-Utility Boosting

To address the $G$-problem and minimize the expected privacy leakage in (15) given the optimized $\hat{X}$ of the $O$-problem, we propose a stochastic privacy-utility boosting algorithm to further lower the expected privacy leakage by searching for a better $\tilde{Y}^*$. Because the objective function of the $G$-problem is not convex, we cannot solve it directly. Instead, we first adjust the prior generalization function $G^0_{Y|Y^*}$ by modifying a stochastically selected generalized value. A new candidate generalization function $\hat{G}^0_{Y|Y^*}$ is generated if the generalized private data given $G^0_{Y|Y^*}$ still satisfy $(k, \alpha)$-uniqueness and $(l, \beta)$-variety. Then we
Algorithm 2: Obfuscation (O-problem).

Input: \( G_{Y|Y} \): generalization function; 
\( \Delta Q \): utility loss budget; 
Output: \( O_{C|C,Y} \): obfuscation function; 
1. \( C \leftarrow \) cluster users based on their privacy-insensitive data; 
2. \( \hat{Y} \leftarrow \) generalize \( Y \) according to \( G_{Y|Y} \); 
3. \( O_{C|C,Y} \leftarrow \) solve the following optimization problem:

\[
\min_{O_{C|C,Y}} I(\hat{C}, \hat{Y})
\]
\[
\text{s.t. } \mathbb{E}_{C \in \hat{C}}[d(C, \hat{g})] \leq \Delta Q \\
O_{C|C}(\hat{c}) \in [0, 1], \forall c, \hat{c} \in C \\
\sum_{\hat{c}} O_{C|C}(\hat{c}|c) = 1, \forall c \in C
\]

4. return \( O_{C|C,Y} \).

Algorithm 3: Generalization (G-Problem).

Input: \( G_{Y|Y} \): the prior generalization function; 
\( O_{\hat{X}|X} \): the optimal obfuscation function given \( G_{Y|Y} \); 
k and \( \alpha \): \((k, \alpha)\)-uniqueness constraint; 
l and \( \beta \): \((l, \beta)\)-variety constraint; 
\( N \): maximum iteration number; 
Output: \( G_{\hat{X}|X} \): better generalization function for \( O_{\hat{X}|X} \); 
1. \( G_{0}|\hat{Y} \leftarrow \) calculate privacy leakage given \( O_{\hat{X}|X} \) and \( G_{Y|Y} \); 
2. \( G_{0}|\hat{Y} \leftarrow \) calculate data utility loss given \( O_{\hat{X}|X} \) and \( G_{Y|Y} \); 
3. for \( n \in [1, N] \) do 
4. \( G_{n}|\hat{Y} \leftarrow \) find a new candidate generalization function that can meet \((k, \alpha)\)-uniqueness and \((l, \beta)\)-variety constraints; 
5. if \( G_{n}|\hat{Y} \) not exists then 
6. break; 
7. else 
8. \( G_{n+1}|\hat{Y} \leftarrow \) calculate privacy leakage given \( O_{\hat{X}|X} \) and \( G_{n}|\hat{Y} \); 
9. \( G_{n+1}|\hat{Y} \leftarrow \) calculate data utility loss given \( O_{\hat{X}|X} \) and \( G_{n}|\hat{Y} \); 
10. if \( I_{n+1} < I^0 \) and \( Q_{n+1} < Q^0 \) then 
11. \( G_{n+1}|\hat{Y} \leftarrow G_{n}|\hat{Y} \); 
12. end 
13. end 
14. end

return \( G_{\hat{X}|X} \).

Algorithm 4: HyObscure.

Input: privacy-insensitive data \( X \) and private data \( Y \); 
Output: \( X, \hat{Y} \): obfuscated \( X, \hat{Y} \); 
1. \( G_{0}|\hat{Y} \leftarrow \) obtain \( G_{0}|\hat{Y} \) by solving I-problem; 
2. while not converged do 
3. \( G_{\hat{X}|X} \leftarrow \) solve O-problem given \( G_{\hat{X}|X} \); 
4. \( G_{\hat{X}|X} \leftarrow \) solve G-problem given \( G_{\hat{X}|X} \); 
5. end 
6. \( X, \hat{Y} \leftarrow \) obfuscate \( X \) by \( O_{\hat{X}|X} \), generalize \( Y \) by \( G_{Y|Y} \); 

where \( O^{*}_{\hat{X}|X} \) is the optimized function when only obfuscation is concerned, and \( c \) is a constant.

V. EVALUATION

In this section, we conduct extensive experiments to evaluate HyObscure. All experiments are run on a Windows 10 desktop server with Intel Core i7-8700 CPU@3.2 GHz and 16 GB RAM.

A. Experimental Setup

1) Data Sets: Two real-world datasets are leveraged for the experiments as follows.

MovieLens [6]: The MovieLens100 K data set contains 943 users with their demographic information and ratings on 1,682 movies. Each user has rated at least 20 movies, and the total number of ratings is more than 100,000. We take movie ratings as privacy-insensitive data and age as private data for the experiments.

Foursquare [4]: We use a Foursquare data set from New York City with 3,669 users and 893,722 check-ins at 1,861 POIs. Note that the home locations are sparsely distributed in New York City. Hence, we divide the area of New York City into 1 km \( \times \) 1 km grids, and all the location points in one grid are equally
represented by the center of the grid. In total, we have 2,640 grids. We employ users’ daily check-in activities as privacy-insensitive data and their home addresses as private information.

2) Evaluation Scenarios: We specify two real-life attribute inference attack scenarios used in our experiments:

**Scenario I:** We assume that the attackers derive a small number of users’ real data including both privacy-insensitive and private information (e.g., real movie ratings and exact age), and train their attribute inference models based on the real data. The attribute inference models are then leveraged to predict other users’ private data when their obscured data are publicly released.

**Scenario II:** We suppose that the attackers trade a set of obscured data with the data holders and meanwhile harvest a small number of users’ real private data (e.g., obfuscated movie ratings and exact age). Note that the small number of users with real private data is a subset of users with obscured data. Then the attribute inference model can be learned on the small number of users whose obscured data and private data are both available to the attackers. It can be used to infer private data about users whose obscured data are accessible.

3) Attribute Inference Attacks and Privacy Leakage: In general, the goal of attackers is to infer exact values about users’ private data with their prior knowledge. For the sake of generality and efficiency, we adopt two representative and effective methods, i.e., Random Forest (RF) [40] and XGBoost [41], to train the inference models for two different private attributes.

**Age inference.** Age is regarded as a private attribute in the MovieLens data set. We exploit users’ movie ratings, either original (Scenario I) or obfuscated (Scenario II), and age stage (generalized age value) as inputs to train age inference attack models. We adopt the squared-loss function for age inference model training as it is more widely used in regression tasks. We consider the age inference as a regression problem and measure the age inference performance by mean absolute error (MAE).

A larger MAE indicates less privacy leakage and better privacy protection.

**Home address inference.** With the Foursquare data set, we take users’ check-in activities and areas of users’ home addresses (generalized private value) as inputs to learn inference attack models regarding users’ exact home addresses. We take home location inference as a multi-class classification problem, thus the log-loss function is applied for training and accuracy is employed to assess its inference performance. A lower inference accuracy represents less privacy leakage. In this vein, we use 1-Accuracy (1-acc) to assess the privacy protection ability on users’ home addresses (larger values mean better protection).

4) Data Utility in Applications: We exploit the published (obscured) data to carry out movie rating prediction and activity recommendation for data utility evaluation.

**Movie rating prediction.** With the MovieLens data set, we rely on the obfuscated movie ratings and age stage to perform user-based collaborative filtering for movie rating prediction [42], [43]. Specifically, we predict movie ratings for users according to the movie rating similarities of the users with the same age stage. We use root mean square error (RMSE) to assess the prediction performance. Lower RMSE indicates better prediction performance and higher data utility given the same prediction method.

**Activity recommendation.** Activity recommendation is implemented based on the analysis of users’ personal check-in patterns and activity preferences [44]. In particular, we calculate the POI similarities of users whose home addresses are in the same area, and make recommendations based on users’ POI similarities. Mean average precision (mAP) is often employed to estimate the recommendation performance. To make the value changing trend the same as RMSE for movie rating prediction, we use 1-mAP to measure activity recommendation performance and data utility. In general, data with larger utility will lead to lower 1-mAP and better recommendation performance.

5) Baselines: Existing privacy-preserving data publishing approaches focus on either obfuscating or generalizing data, but rarely consider hybrid obfuscating for both privacy-insensitive and private data. Hence, we fix either one of the two obfuscating operations and replace the other one with existing methods to implement the baselines without considering the data correlation. On the one hand, we use our initial generalization function for private data and perform state-of-the-art obfuscation methods for privacy-insensitive data. We employ the following methods to obfuscate privacy-insensitive data as one group of baselines.

- **Random** [19]. It randomly selects some users for obfuscation given a default obfuscation budget and then obfuscates the users with another randomly selected user.
- **Frapp** [45]. It performs data obfuscation with a higher ratio to users themselves but with a lower likelihood to others.
- **Simp** [19]. It obfuscates data between users who are more similar (dissimilar) to each other with a larger (smaller) probability.
- **DP** [32]. It obfuscates user $u$ to $v$ by a probability decreasing exponentially with their distance $d(u, v)$, i.e., $p_{DP}(v|u) \propto \exp(-\beta d(u, v))$. It satisfies $2\beta d_{\max}$-differential privacy, where $d_{\max} = \max_{u,v\in U} d(u, v)$.
- **PrivCheck** [19]. It optimizes an obfuscation function for privacy-insensitive data to achieve minimal privacy leakage given a distortion budget.

On the other hand, we replace our initial generalization function by two traditional generalization methods and use our obfuscation method to implement another group of baselines. These methods are tuned to produce $\hat{Y}$ generalized private values for fair comparisons.

- **NCPGen** [46]. It performs generalization by optimizing the normalized certainty penalty (NCP) metric.
- **kNNGen** [47], [48]. It employs a k-nearest neighbor (kNN) clustering technique to achieve $K$-anonymity for private data generalization.

In addition, we implement two variants of HyObscure to verify the importance of cross-iterative obscuring:

- **XObf**. Given the initial generalization approach $\mathcal{O}_{Y\sim Y'}^{\mathcal{O}}$, $XObf$ carries out the generalization-aware obfuscation and aims to identify an optimal obfuscation function.
- **YGen**. Given an obfuscation function $\mathcal{O}_{X\sim X'}^{\mathcal{O}}$, $YGen$ runs the stochastic privacy-utility boosting approach to find a better generalization solution. PrivCheck is set to the default $\mathcal{O}_{X\sim X}$ as it performs the best among the baselines.
Fig. 3. Tradeoff: privacy protection on age vs. movie rating prediction performance (MovieLens).

Fig. 4. Tradeoff: privacy protection on home address vs. activity recommendation performance (Foursquare).

B. Experimental Results

We report the experimental results of HyObscure from the following aspects: 1) privacy-utility tradeoff, 2) effectiveness of cross-iterative obscuring, 3) impact of key parameters, 4) convergence, and 5) scalability. The default parameters set for the first two experiments are listed in Table II. Since all the experimental results are similar for the RF and XGBoost inference attack models, we report the privacy-utility tradeoff using both RF and XGBoost models, and only report the results of RF for the rest experiments due to the page limit.

1) Privacy-Utility Tradeoff: We here conduct experiments using two different inference attack methods (RF and XGBoost) under two attack scenarios where attackers have different prior knowledge. Figs. 3 and 4 show the trend of privacy protection effectiveness of our method as well as various baselines given varying utility loss on MovieLens and Foursquare data sets, respectively. Note that 1) the increase of the number on x-axis (RMSE or 1-mAP) represents the decrease of utility, 2) a larger number on y-axis (MAE or 1-acc) indicates attackers make more errors in predicting private data and thus better privacy protection, and 3) the bottom-left most point in each figure is the reference point, indicating the “worst” privacy protection and the “best” utility with the unobscured original data.

Figs. 3 and 4 show the same trend that the privacy protection performance is continuously improved with the decrease of utility, demonstrating that the privacy protection indeed can be augmented by carefully obscuring data at the expense of utility reduction. More importantly, HyObscure achieves a more efficient privacy-utility tradeoff by using lower utility loss to pursue a better privacy protection effect. For instance, in Fig. 3(a), when the customized utility on movie rating prediction performance decreases around 0.2% compared to the reference point, HyObscure can raise the privacy protection performance on age by 7.2%, whereas the privacy protection increments by other baselines are only around 1%. Fig. 3(b) shows that when the privacy protection performance is around 0.817 in terms of 1-acc, the data utility loss caused by HyObscure is lower than the baseline approaches by more than 35%. Recall that all the baselines do not consider the correlation between privacy-insensitive and generalized private data when applying data obscuring, the results verify the importance of considering data correlation in hybrid data obscuring.

2) Importance of Cross-Iterative Optimization: Fig. 5 compares HyObscure with two variants on the privacy protection effect of the age and location data given certain levels of application utility. Note that both XObf and YGen subsequently optimize two obscuring functions without the cross-iterative process. In Fig. 5, HyObscure delivers obviously higher privacy protection than that of XObf and YGen given the same level of utility. The results demonstrate that cross-iteratively optimizing the obfuscation function on privacy-insensitive data and the generalization function on private data can effectively strengthen the privacy protection capability of HyObscure.

3) Impact of Key Parameters: Three groups of parameters are examined to verify the reliability of HyObscure.
Generalized private value size ($|\tilde{Y}|$). We vary the generalized private value size to conduct a robustness check on HyObscure. To focus on the impact of $|\tilde{Y}|$, we fix a certain utility budget and inspect whether HyObscure can perform better in terms of privacy protection at different levels of $|\tilde{Y}|$. Fig. 6 displays the results where the RMSE of movie rating prediction is set to 1.038 for MovieLens and the 1-mAP of activity recommendation is set to 0.371 for Foursquare. The other parameters are set to default values as listed in Table II. These results verify that HyObscure consistently outperforms the baseline methods with varied $|\tilde{Y}|$.

The number of user clusters ($|C|$). As aforementioned, HyObscure reduces computational complexity by clustering users based on their privacy-insensitive data. However, the number of clusters $|C|$ may impact the privacy protection effects. Here, we investigate the impacts of $|C|$ by varying its value over a wide range (from 10 to 160) on MovieLens data set. The results shown in Fig. 7 demonstrate that the privacy protection effect continues to improve as $|C|$ increases. This aligns with our expectation, since a smaller $|C|$ leads to a smaller search space for solving the O-problem, and degrades the privacy protection effect. Nevertheless, a small $|C|$ does accelerate the computation. Therefore, we suggest developers carefully selecting a $|C|$ to balance the privacy protection effect and computation time.

4) Convergence of HyObscure: While we have proved the convergence of HyObscure in theory in Section IV-F, Fig. 9 empirically examines the convergence of the optimization process of an instance. It shows that the minimized privacy leakage decreases monotonically by the increase of optimization iterations with an apparent convergent trend. HyObscure stops when it cannot find a better solution to further reduce the objective value by more than the pre-defined threshold, which is set to 0.0001 in the experiment.

5) Algorithm Scalability: We examine the computational efficiency of HyObscure by varying the sizes of users and movies/POIs, respectively, on both data sets. Fig. 10(a) and (b) show that HyObscure scales linearly with the size of the user and that of the movie/POI.
VI. CONCLUSION

In this paper, we addressed a novel privacy-preserving data publishing problem that aims to employ hybrid obscuring operations on heterogeneous features for effective privacy-preserving data publishing. To that end, we first proposed new privacy and utility quantification measurements for published data by considering the joint effect of generalization and obfuscation. We also proposed a method called HyObscure to cross-iteratively optimize the data generalization and obfuscation functions for the best privacy protection effect under the utility guarantee. Both theoretical and empirical studies validated the effectiveness of HyObscure.

Our study has some limitations. HyObscure in theory can work for multiple private features by treating them as a multi-dimensional feature. For example, Location in our experiments is a two-dimensional feature with both latitude and longitude information. However, the searching space of generalization functions increases exponentially with the dimension, so the computational cost will be extremely high for very high-dimensional data. We leave the solution as future work. Moreover, while we take the widely adopted obfuscation and generalization operations as an example for hybrid obscuring of heterogeneous features, other obscuring measures like adding noise [20] could also be integrated into HyObscure to resist more complicated inference attacks or attackers with more prior knowledge. We also leave this direction to future work.
[23] X.-B. Li and S. Sarkar, “Against classification attacks: A decision tree pruning approach to privacy protection in data mining,” *Operations Res.*, vol. 57, no. 6, pp. 1496–1509, 2009.

[24] S. Salamatian et al., “Managing your private and public data: Bringing down inference attacks against your privacy,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1240–1255, Oct. 2015.

[25] P. Mohassel and Y. Zhang, “SecureML: A system for scalable privacy-preserving machine learning,” in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 19–38.

[26] N. Z. Gong and B. Liu, “You are who you know and how you behave: Attribute inference attacks via users’ social friends and behaviors,” in *Proc. USENIX Symp. Secur.*, 2016, pp. 979–995.

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 3–18.

[28] M. Nasr, R. Shokri, and A. Housmansadr, “Machine learning with membership privacy using adversarial regularization,” in *Proc. ACM SIGSAC Conf. Comput. Commun., Secur.*, 2018, pp. 634–646.

[29] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, “MemGuard: Defending against black-box membership inference attacks via adversarial examples,” 2019, arXiv: 1909.10594.

[30] J. Zhou, Y. Chen, C. Shen, and Y. Zhang, “Property inference attacks against GANs,” in *Proc. Int. Conf. Softw. Eng.*, 2021.

[31] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *Proc. IEEE Annul. Symp. Found. Comput. Sci.*, 2007, pp. 94–103.

[32] L. Wang, G. Qin, D. Yang, X. Han, and X. Ma, “Geographic differential privacy for mobile crowd coverage maximization,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 200–207.

[33] F. du Pin Calmon and N. Fawaz, “Random forests,” *Mach. Learn.*, vol. 102, no. 1, pp. 46–50.

[34] J. A. Hartigan and M. C. Wong, “Algorithm AS 136: A K-means clustering algorithm,” *J. Roy. Statist. Soc.*, vol. 28, no. 1, pp. 100–108, 1979.

[35] H. Wang and M. Song, “Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming,” *R J.*, vol. 32, pp. 29–33, 2011.

[36] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[37] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[38] J. L. Herlocker, J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, “GroupLens: Applying collaborative filtering to usenet newsgroups,” *Commun. ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[39] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 230–237.

[40] D. Yang, D. Zhang, V. W. Zheng, and J. Yu, “Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[41] S. Agrawal and J. R. Haritsa, “A framework for high-accuracy privacy-preserving mining,” in *Proc. IEEE Int. Conf. Data Eng.*, 2005, pp. 193–204.

[42] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, “Utility-based anonymization using local recoding,” in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 785–790.

[43] D. Slijepčević, M. Henzl, L. D. Klausner, T. Dam, P. Kieseberg, and M. Zeppelzauer, “k-anonymity in practice: How generalisation and suppression affect machine learning classifiers,” *Comput. Secur.*, vol. 111, 2021, Art. no. 102488.

[44] J. Li and M.-C. Wei, “An efficient clustering method for k-anonymization,” in *Proc. Int. Workshop Privacy Anonymity Inf. Sec.*, 2008, pp. 46–50.

HAN et al.: HYOBSCURE: HYBRID OBSCURING FOR PRIVACY-PRESERVING DATA PUBLISHING 3905

Xiao Han received the PhD degree in computer science from Pierre and Marie Curie University and Institut Mines-Telecom/Telecom SudParis. She is currently a full professor with the School of Information Management and Engineering at Shanghai University of Finance and Economics in China. Her research interests include data mining, information security, and privacy-preserving computing.

Yuncong Yang received the BS and MS degrees in management science and engineering from the Shanghai University of Finance and Economics, in 2018 and 2020, respectively. Currently, he is working toward the PhD degree with the School of Information Management and Engineering, Shanghai University of Finance and Economics. His research interests include graph neural networks and privacy protection in machine learning.

Junjie Wu is currently a full professor with the School of Economics and Management, Beihang University. He has been engaged in the interdisciplinary research of management science, computer science, and social science. His research interests include data science and artificial intelligence with intense interests in smart city, fintech, and business intelligence. His work has been published prolifically on international journals including *Information Systems Research*, *Management Information Systems Quarterly*, *The Journal of Organic Chemistry*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Information Systems*, etc., as well as leading technical conferences such as KDD, SIGIR, AAAI, IJCAI, etc. He holds more than 25 national invention patents and 3 first-class provincial science and technology awards. He is the recipient of the NSFC Distinguished Young Scholars grant and the PI of five national key projects granted by NSFC and MOST.

Hui Xiong (Fellow, IEEE) received the PhD degree in computer science from the University of Minnesota, USA. He is a chair professor, associate vice president (Knowledge Transfer), and head of the AI Thrust with the Hong Kong University of Science and Technology (Guzhou). His research interests include Artificial Intelligence, data mining, and mobile computing. He has served on numerous organization and program committees for conferences, including as program co-chair for the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), program co-chair for the 2013 International Conference on Data Mining (ICDM), general co-chair for the 2015 IEEE International Conference on Data Mining (ICDM), and program co-chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He received several awards, such as the 2021 AAAI Best Paper Award and the 2011 IEEE ICDM Best Research Paper Award. For his significant contributions to data mining and mobile computing, he was elected as a fellow of AAAAS in 2020.