The Curse of Correlations for Robust Fingerprinting of Relational Databases

Tianxi Ji
txj116@case.edu
Case Western Reserve University
Cleveland, Ohio, USA

Emre Yilmaz
yilmaze@uhd.edu
University of Houston-Downtown
Houston, TX, USA

Erman Ayday
exa208@case.edu
Case Western Reserve University
Cleveland, Ohio, USA

Pan Li
lipan@case.edu
Case Western Reserve University
Cleveland, Ohio, USA

ABSTRACT

Database fingerprinting have been widely adopted to prevent unauthorized sharing of data and identify the source of data leakages. Although existing schemes are robust against common attacks, like random bit flipping and subset attack, their robustness degrades significantly if attackers utilize the inherent correlations among database entries. In this paper, we first demonstrate the vulnerability of existing database fingerprinting schemes by identifying different correlation attacks: column-wise correlation attack, row-wise correlation attack, and the integration of them. To provide robust fingerprinting against the identified correlation attacks, we then develop mitigation techniques, which can work as post-processing steps for any off-the-shelf database fingerprinting schemes. The proposed mitigation techniques also preserve the utility of the fingerprinted database considering different utility metrics. We empirically investigate the impact of the identified correlation attacks and the performance of mitigation techniques using real-world relational databases. Our results show (i) high success rates of the identified correlation attacks against existing fingerprinting schemes (e.g., the integrated correlation attack can distort 64.8% fingerprint bits by just modifying 14.2% entries in a fingerprinted database), and (ii) high robustness of the proposed mitigation techniques (e.g., with the mitigation techniques, the integrated correlation attack can only distort 3% fingerprint bits).

CCS CONCEPTS

• **Security and privacy** → **Digital rights management**; **Information accountability and usage control**; *Privacy protections.*

KEYWORDS

Robust fingerprinting; databases; correlation attacks; data sharing

ACM Reference Format:
Tianxi Ji, Emre Yilmaz, Erman Ayday, and Pan Li. 2021. The Curse of Correlations for Robust Fingerprinting of Relational Databases. In *24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '21), October 6–8, 2021, San Sebastian, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3471621.3471853

1 INTRODUCTION

Relational databases (or relations) have become the most popular database systems ever since 1970s. A relation is defined as a set of data records with the same attributes [12]. Constructing and sharing of the relations are critical to the vision of a data-driven future that benefits all human-beings. It supports broader range of tasks in real-life than just sharing database statistics or machine learning models trained from the database. For example, a relational database owner (who collects data from individuals and constructs the dataset) can benefit from outsourced computation (e.g., from service providers (SP) like Amazon Elastic Compute Cloud), let other SPs analyze its data (e.g., for personal advertisements), or exchange datasets for collaborative research after data use agreements.

Most of the time, sharing a database with an authorized SP (who is authorized to receive/use the database) is done via consent of the database owner. However, when such databases are shared or leaked beyond the authorized SPs, individuals' (people who contribute their data in the database) privacy is violated, and hence preventing unauthorized sharing of databases is of great importance. Thus, database owners want to (i) make sure that shared data is used only by the authorized parties for specified purposes and (ii) discourage such parties from releasing the received datasets to other unauthorized third parties (either intentionally or unintentionally). Such data breaches cause financial and reputational damage to database owners. For instance, it is reported that the writing site Wattpad suffered a major data breach in July 2020; over 270 million individuals' data were sold on a third party forum in the darknet [1]. Therefore, identifying the source of data breaches is crucial for database owners to hold the identified party responsible.

Digital fingerprinting is a technology that allows to identify the source of data breaches by embedding a unique mark into each shared copy of a digital object. Unlike digital watermarking, in fingerprinting, the embedded mark must be unique to detect the guilty party who is responsible for the leakage. Although the most prominent usage of fingerprinting is in the multimedia domain [14, 15, 19], fingerprinting techniques for databases have also been developed [18, 22, 24, 25]. These techniques change database entries at different positions when sharing a database copy with a SP. However, existing fingerprinting schemes for databases have been developed to embed fingerprints in continuous-valued numerical entries (floating points) in relations. On the other hand, fingerprinting discrete (or categorical) values is more challenging, since the number of possible values (or instances) for a data point is much fewer. Hence, in such databases, a small change in the value of a data point (as a fingerprint) can significantly affect the utility. In addition, existing fingerprinting schemes for databases

RAID '21, October 6–8, 2021, San Sebastian, Spain

ACM ISBN 978-1-4503-9058-3/21/10…$15.00
https://doi.org/10.1145/3471621.3471853

do not consider various inherent correlations between the data records in a database. A malicious party having a fingerprinted copy of a database can detect and distort the embedded fingerprints using its knowledge about the correlations in the data. For example, the zip codes are strongly correlated with street names in a demographic database making common fingerprinting schemes venerable to attacks utilizing such correlations. Thus, to provide robustness against correlation attacks (which utilizes the correlations between attributes and data records to infer the potentially fingerprinted entries), we need to consider such correlations when developing fingerprinting schemes for relational database.

In this work, we first identify correlation attacks against the existing database fingerprinting schemes. Namely, we present column-wise correlation attack, row-wise correlation attack, and the integration of both. To launch these attacks, a malicious SP utilizes its prior knowledge about correlations between the columns (attributes) of database, statistical relationships between the rows (data records), and the combination of both. After launching these attacks on a fingerprinted database, the malicious SP can easily distort the added fingerprint to mislead the fingerprint extraction algorithm and cause the database owner to accuse innocent parties. For example, we show that by changing 14.2% entries in a database, the integration of row- and column-wise correlation attack can distort 64.8% fingerprint bits and cause the database owner falsely accuse innocent SPs with high probability. This suggests that existing database fingerprinting schemes are vulnerable to identified correlation attacks, and mitigation techniques are in dire need.

To reduce the identified vulnerability in existing database fingerprinting schemes, we propose novel mitigation techniques to provide robust fingerprinting that can alleviate the correlation attacks. Although, we describe the proposed techniques for a specific vanilla database fingerprinting scheme [24], they can be applied to other schemes as well. In other words, the proposed mitigation techniques can work as post-processing steps for any off-the-shelf database fingerprinting schemes and make them robust against potential attacks that utilize the inherent data correlations. The proposed mitigation techniques utilize database owner's prior knowledge on the column- and row-wise correlations. In particular, to mitigate the column-wise correlation attack, the database owner modifies some of the non-fingerprinted data entries to make the post-processed fingerprinted database have column-wise correlations close to that of her prior knowledge. The data entry modification plans are determined from the solutions to a set of "optimal transportation" problems [13], each of which transports the mass of the marginal distribution of a specific attribute (column) to make it resemble the reference marginal distribution computed from database owner's prior knowledge while minimizing the transportation cost. To alleviate the row-wise correlation attack, the database owner modifies limited number of non-fingerprinted data entries by solving a combinatorial search problem to make the post-processed fingerprinted database have row-wise statistical relationships that are far away from that of her prior knowledge. We show that even if the malicious SP has access to the exactly same prior knowledge (i.e., data correlation models) with the database owner, the proposed mitigation techniques can effectively reduce the vulnerability caused by correlation attacks. The proposed mitigation techniques also maintain the utility of the post-processed fingerprinted database by (i) encoding the database entries as integers, such that the least significant bit (LSB) carries the least information, and adding the fingerprint by only changing the LSBs; and (ii) changing only a small number of database entries.

We use an real-world Census relational database to validate the effectiveness of the proposed robust fingerprinting scheme against the identified correlation attacks. In particular, we show that the malicious SP can only compromise 3% fingerprint bits, even if it launches the powerful integrated correlation attack on the Census database. Thus, it will be held as responsible for data leakage.

We summarize the main contributions of this paper as follows:

- We identify correlation attacks that can distort large portion of the fingerprint bits in the existing database fingerprinting scheme and cause the database owner to accuse innocent SPs with high probability.
- We propose robust fingerprinting scheme that involves novel mitigation techniques to alleviate the impact of the identified correlation attacks. The proposed mitigation techniques can work as post-processing steps for any off-the-shelf database fingerprinting schemes.
- We investigate the impact of the identified correlation attacks and the proposed mitigation techniques on an real-world relational database. We show that the correlation attacks are more powerful than traditional attacks, because they can distort more fingerprint bits with less utility loss. On the other hand, the mitigation techniques can effectively alleviate these attacks and maintain database utility even if the malicious SP uses data correlation models that are directly calculated from the data.

The rest of this paper is organized as follows. We review related works on existing fingerprinting schemes in Section 2, which is followed by the description on the considered vanilla fingerprinting scheme in Section 3. In Section 4, we present the system and threat models, and evaluation metrics. Section 5 introduces the identified correlation attacks. In Section 6, we develop robust fingerprinting against the identified attacks. We evaluate the impact of correlation attacks and the performance of the proposed mitigation techniques in Section 7. Finally, Section 8 concludes the paper.

2 RELATED WORK

We first briefly review the works on multimedia fingerprinting, and then focus on existing works on fingerprinting relational database.

Large volume of research on watermarking and fingerprinting have targeted multimedia, e.g., images [17, 30], audio [5, 21], videos [29], and text documents [9, 10]. Such works benefit from the high redundancy in multimedia, such that the inserted watermark or fingerprint is imperceptible for human beings. However, the aforementioned multimedia fingerprinting techniques cannot be applied to fingerprint relational databases. The reason is that a database fingerprinting scheme should be robust against common database operations, such as union, intersection, and updating, whereas multimedia fingerprinting schemes are designed to be robust against operations, like compression and formatting.

Database fingerprinting schemes are usually discussed together with database watermarking schemes [23] due to their similarity. In the seminal work [2], Agrawal et al. introduce a watermarking framework for relations with numeric attributes by assuming that the database consumer can tolerate a small amount of error in the watermarked databases. Then, based on [2], some database fingerprinting schemes have been devised. Specifically, Guo et al. [18] develop a two-stage fingerprinting scheme: the first stage is used to prove database ownership, and the second stage is designed for extracted fingerprint verification. Li et al. [24] develop a database fingerprinting scheme by extending [2] to enable the insertion and extraction of arbitrary bit-strings in relations. Furthermore, the authors provide an extensive robustness analysis (e.g., about the upper bound on the probability of detecting incorrect but valid fingerprint from the pirated database) of their scheme. Although [18, 24] pseudorandomly determine the fingerprint positions in a database, they are not robust against our identified correlation attacks. In this paper, we consider [24] as the vanilla fingerprinting scheme and corroborate its vulnerability against correlation attacks. Additionally, Liu et al. [25] propose a database fingerprinting scheme by dividing the relational database into blocks and ensuring that certain bit positions of the data at certain blocks contain specific values. In [25], since the fingerprint is embedded block-wise, it is more susceptible to attacks utilizing correlations in the data. As a result, incorporating data correlations in database fingerprint schemes is critical to provide robustness against correlation attacks.

Recently, Yilmaz et al. [33] develop a probabilistic fingerprinting scheme by explicitly considering the correlations (in terms of conditional probabilities) between data points in data record of a single individual. Ayday et al. [4] propose an optimization-based fingerprinting scheme for sharing personal sequential data by minimizing the probability of collusion attack with data correlation being one of the constraints. Our work differs from these works since we focus on developing robust fingerprint scheme for relational databases, which (i) contain large amount of data records from different individuals, (ii) include both column- and row-wise correlations, and (iii) have different utility requirements.

3 THE VANILLA FINGERPRINT SCHEME

In this work, we consider the fingerprinting scheme proposed in [24] as the vanilla scheme, for which we show the vulnerability and develop the proposed scheme. Assume a database owner shares her data with multiple service providers (SPs). The fingerprint of a specific SP is obtained using a cryptographic hash function, whose input is the concatenation of the database owner's secret key and the SP's public series number. For fingerprint insertion, the vanilla scheme pseudorandomly selects one bit position of one attribute of some data records in the database and replaces those bits with the results obtained from the exclusive or (XOR) between mask bits and fingerprint bits, both of which are also determined pseudorandomly. For fingerprint extraction, the scheme locates the exact positions of the potentially changed bits, calculates the fingerprint bits by XORing those bits with the exact mask bits, and finally recovers each bit in the fingerprint bit-string via majority voting, since each fingerprint bit can be used to mark many different positions. To preserve the utility of the fingerprinted database, we will let the vanilla scheme only change the least significant bit (LSB) of selected database entries. For completeness, we show the steps to insert fingerprint into a database, and the steps to extract fingerprint from a pirated database, in Algorithms 1 and 2, respectively. In Appendix A, we will empirically validate that only changing the LSB indeed leads to higher utility than altering one of the least k significant bits (LkSB) of selected entries.

Algorithm 1: Fingerprint insertion phase of the vanilla fingerprinting scheme [24]

```
Input  : The original relational database $\mathbf{R}$, fingerprinting ratio $\gamma$,
         database owner's secret key $\mathcal{K}$, pseudorandom number
         sequence generator $\mathcal{U}$, and the SP's series number $n$
         (which can be public).
Output : The vanilla fingerprinted relational database $\widetilde{\mathbf{R}}\left(\text{FP}, \emptyset, \emptyset\right)$.
Generate the fingerprint bit string of SP $n$, i.e., $f_{\text{SP}_n} = Hash(\mathcal{K}|n)$;
forall data record $\boldsymbol{r}_i \in \mathbf{R}$ do
    if $\mathcal{U}_1(\mathcal{K}|\boldsymbol{r}_i.\text{primary key}) \bmod \gamma = 0$ then
        //fingerprint this data record
        attribute_index $p = \mathcal{U}_2(\mathcal{K}|\boldsymbol{r}_i.\text{primary key}) \bmod |\mathcal{F}|$.
          //fingerprint this attribute ($|\mathcal{F}|$ is the
          cardinality of the attributes set)
        Set mask_bit $x = 0$, if $\mathcal{U}_3(\mathcal{K}|\boldsymbol{r}_i.\text{primary key})$ is even;
          otherwise set $x = 1$.
        fingerprint_index $l = \mathcal{U}_4(\mathcal{K}|\boldsymbol{r}_i.\text{primary key}) \bmod L$. //$L$
          is the length of the fingerprint bit-string
        fingerprint_bit $f = f_{\text{SP}_n}(l)$.
        mark_bit $m = x \oplus f$.
        Set the LSB of $\boldsymbol{r}_i.p$ to $m$.
    end
end
Return $\widetilde{\mathbf{R}}\left(\text{FP}, \emptyset, \emptyset\right)$.
```

In practice, one can choose any database fingerprinting scheme as the vanilla scheme, because our proposed mitigation techniques are independent of the adopted vanilla scheme, and they can be used as post-processing steps on top of any existing database fingerprinting schemes. The reason we choose the aforementioned vanilla scheme is because (i) it is shown to have high robustness, e.g., the probability of detecting no fingerprint as a result of random bit flipping attack (a common attack against fingerprinting schemes, as will be discussed in Section 4.2) is upper bounded by $(|SP| - 1)/2^L$, where $|SP|$ is the number of SPs who have received the fingerprinted copies and L is the length of the fingerprint bit-string, (ii) it is shown to be robust even if some fingerprinted entries are identified by a malicious SP, because it applies majority voting on all the fingerprinted entries to extract the fingerprint bit-string, and (iii) it can easily be extended to incorporate Boneh-Shaw code [8] to defend against collusion attacks. Our developed robust fingerprinting scheme inherits all the properties of the vanilla scheme because (i) it uses the vanilla scheme as the building block and (ii) it does not alter the entries that have already been changed by the vanilla scheme (due to fingerprinting insertion).

4 SYSTEM AND THREAT MODELS

First, we introduce the nomenclature for different databases obtained by applying various techniques. We denote the database owner's (i.e., Alice) original database as $\mathbf{R}$, a fingerprinted database shared by her as $\widetilde{\mathbf{R}}$, and the pirated database leaked by a malicious

Algorithm 2: Fingerprint extraction phase of the vanilla fingerprinting scheme [24]

```
Input  : The leaked relational database $\overline{\mathbf{R}}$, fingerprinting ratio $\gamma$,
         database owner's secret key $\mathcal{K}$, pseudorandom number
         sequence generator $\mathcal{U}$, and a fingerprint template
         $(?, ?, \cdots, ?)$, where ? represents unknown value.
Output : The extracted fingerprint from the leaked database.
forall $l \in [1, L]$ do
    $count[l][0] = count[l][1] = 0$. //$count[l][0]$ and
     $count[l][1]$ are number of votes for $f(l)$ to be 0 or
     1, respectively.
end
//scan all data records and obtain the counts for each
 fingerprint bit
forall data record $\boldsymbol{r}_i \in$ R do
    if $\mathcal{U}_1(\mathcal{K}|\boldsymbol{r}_i$.primary key) mod $\gamma = 0$ then
        attribute_index $p = \mathcal{U}_2(\mathcal{K}|\boldsymbol{r}_i$.primary key) mod $|\mathcal{F}|$.
        Set mark_bit $m$ as the LSB of $\boldsymbol{r}_i.p$.
        Set mask_bit $x = 0$, if $\mathcal{U}_3(\mathcal{K}|\boldsymbol{r}_i$.primary key) is even;
         otherwise set $x = 1$.
        fingerprint_bit $f = m \oplus x$.
        fingerprint_index $l = \mathcal{U}_4(\mathcal{K}|\boldsymbol{r}_i$.primary key) mod $L$.
        $count[l][f] = count[l][f] + 1$.
    end
end
//recover the fingerprint bit string
forall $l \in [1, L]$ do
    if $count[l][0] = count[l][1]$ then
        return none suspected
    end
    $f(l) = 0$ if $count[l][0] > count[l][1] = 0$.
    $f(l) = 1$ if $count[l][0] < count[l][1] = 0$.
end
Return the extracted fingerprint bit string $f$.
```

SP as $\overline{\mathbf{R}}$, respectively. Both $\widetilde{\mathbf{R}}$ and $\overline{\mathbf{R}}$ are represented using 3 input parameters showing the techniques that are adopted to generate them. 3 input parameters for $\widetilde{\mathbf{R}}(\alpha, \beta, \eta)$ represent which processes have been applied to the database during fingerprinting, where (i) α represents the vanilla fingerprinting, (ii) β represents the proposed mitigation technique against the row-wise correlation attack, and (iii) η represents the proposed mitigation technique against the column-wise correlation attack. On the other hand, 3 input parameters for $\overline{\mathbf{R}}(\alpha, \beta, \eta)$ represent which attacks have been conducted by the malicious SP on the fingerprinted database, where (i) α represents the random bit flipping attack, (ii) β represents the row-wise correlation attack, and (iii) η represents the column-wise correlation attack. We provide the details of these attacks and mitigation techniques in Sections 5 and 6, respectively. We will also use $\widetilde{\mathbf{R}}$ (or $\overline{\mathbf{R}}$) when referring to a generic fingerprinted (or pirated) database when its input parameters are clear from the context.

We summarize the frequently used notations in Table 1. For instance, $\widetilde{\mathbf{R}}\big(\text{FP}, \text{Dfs}_{\text{row}}(\mathcal{S}'), \text{Dfs}_{\text{col}}(\mathcal{J}')\big)$ represents a fingerprinted database that is generated by applying the vanilla fingerprinting scheme (FP) on the original database $\mathbf{R}$ followed by two proposed mitigation techniques $\text{Dfs}_{\text{row}}(\mathcal{S}')$ and $\text{Dfs}_{\text{col}}(\mathcal{J}')$ to alleviate the potential correlation attacks (as will be discussed in Sections 6.1 and 6.2). Here, $\mathcal{S}'$ (or $\mathcal{J}'$) is the database owner's prior knowledge on the row-wise (or column-wise) correlations in the database. Similarly, $\overline{\mathbf{R}}\big(\emptyset, \text{Atk}_{\text{row}}(\mathcal{S}), \text{Atk}_{\text{col}}(\mathcal{J})\big)$ represents a pirated database that is generated by a malicious SP by first launching the row-wise correlation attack $\text{Atk}_{\text{row}}(\mathcal{S})$, and then the column-wise correlation attack $\text{Atk}_{\text{col}}(\mathcal{J})$ (as will be discussed in Section 5.1 and 5.2, and $\emptyset$ means random bit flipping attack is not applied). Here, $\mathcal{S}$ (or $\mathcal{J}$) is the malicious SP's prior knowledge on the row-wise (or column-wise) correlations of the database. In general, $\mathcal{S}' \neq \mathcal{S}$ and $\mathcal{J}' \neq \mathcal{J}$, which is referred to as the prior knowledge asymmetry between the database owner and the malicious SP. To the advantage of the malicious SP, we assume that the malicious SP can have access to the correlation models that are directly calculated from the database, i.e., its prior knowledge is as accurate as that of the database owner. In the future work, we will also investigate the scenario where the database owner even has less accurate prior knowledge compared with the malicious SPs.

4.1 System Model

We present the vanilla fingerprint system model in Figure 1. Specifically, we consider the database owner (Alice) with a categorical relational database $\mathbf{R}$, which includes the data records of M individuals. We denote the set of attributes of the individuals as $\mathcal{F}$ and the ith row (data record) in $\mathbf{R}$ as $\boldsymbol{r}_i$. Alice shares her data with multiple service providers (SPs) to receive specific services from them. To prevent unauthorized redistribution of her database by a malicious SP, Alice includes a unique fingerprint in each copy of her database when sharing it with a SP. The fingerprint bit-string associated to SP i (SP$_i$) is denoted as f_{SP_i}, and the vanilla fingerprinted dataset received by SP$_i$ is represented as $\widetilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$. Both f_{SP_i} and $\widetilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$ are obtained using the vanilla fingerprint scheme discussed in Section 3, which changes entries of $\mathbf{R}$ at different positions (indicated by the yellow dots in Figure 1. If a malicious SP (e.g., SP$_i$) pirates and redistributes Alice's database, she is able to identify SP$_i$ as the traitor by extracting its fingerprint in $\widetilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$ as long as the data entries are not significantly modified (e.g., when less than 80% entries are changed or removed).

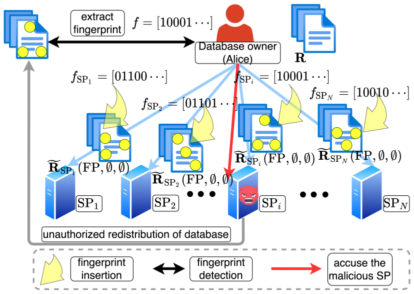
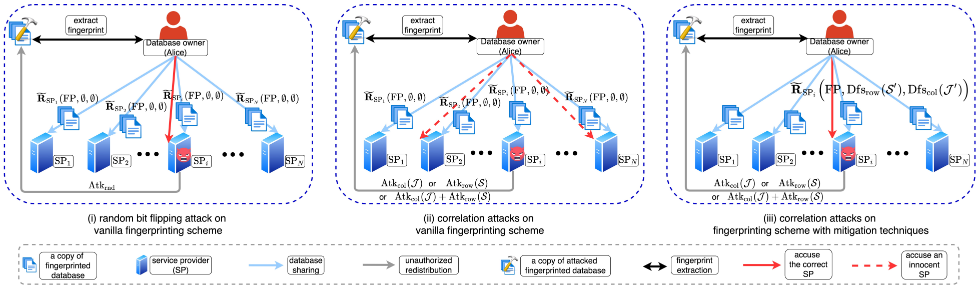

extract fingerprint
$f = [10001 \cdots]$
Database owner (Alice)
$\mathbf{R}$
$f_{\text{SP}_1} = [01100 \cdots]$
$f_{\text{SP}_2} = [01101 \cdots]$
$f_{\text{SP}_i} = [10001 \cdots]$
$f_{\text{SP}_N} = [10010 \cdots]$
$\widetilde{\mathbf{R}}_{\text{SP}_1}(\text{FP}, \emptyset, \emptyset)$
$\widetilde{\mathbf{R}}_{\text{SP}_2}(\text{FP}, \emptyset, \emptyset)$
$\widetilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$
$\widetilde{\mathbf{R}}_{\text{SP}_N}(\text{FP}, \emptyset, \emptyset)$
SP$_1$
SP$_2$
SP$_i$
SP$_N$
unauthorized redistribution of database
fingerprint insertion
fingerprint detection
accuse the malicious SP

Figure 1: The vanilla fingerprinting system, where Alice adds a unique fingerprint in each shared copy of her database. She is able to identify the malicious SP who pirates and redistributes her database as long as the data entries are not significantly modified (e.g., when less than 80% of the entries are changed or removed).

$\mathbf{R}$	the original database owned by the database owner (Alice)
$\widetilde{\mathbf{R}}$	a generic fingerprinted database shared by the database owner
$\overline{\mathbf{R}}$	a generic pirated database generated by the malicious SP
$\widetilde{\mathbf{R}}(\alpha,\beta,\eta)$	the fingerprinted database obtained by applying (i) α, the vanilla fingerprinting scheme, (ii) β, the mitigation technique against the row-wise correlation attack, and (iii) η, the mitigation technique against the column-wise correlation attack in sequence
$\overline{\mathbf{R}}(\alpha,\beta,\eta)$	the pirated database generated by the malicious SP by applying (i) the random bit flipping attack α, (ii) the row-wise correlation attack β, and (iii) the column-wise correlation attack η in sequence
$\mathcal{S}'$ and $\mathcal{J}'$	database owner's prior knowledge on the row-wise correlations and column-wise correlations
$\mathcal{S}$ and $\mathcal{J}$	the malicious SP's prior knowledge on the row-wise correlations and column-wise correlations
$\widetilde{\mathcal{S}}$ and $\widetilde{\mathcal{J}}$	the empirical row-wise and column-wise correlations obtained from a generic fingerprinted database $\widetilde{\mathbf{R}}$
$\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})$	the column-wise correlation attack launched by the malicious SP by using prior knowledge $\mathcal{J}$
$\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$	the mitigation technique using prior knowledge $\mathcal{S}'$ to alleviate row-wise correlation attack
$\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$	the mitigation technique using prior knowledge $\mathcal{J}'$ to alleviate column-wise correlation attack

Table 1: Frequently used notations in the paper.


extract fingerprint
Database owner (Alice)
$\widetilde{\mathbf{R}}_{\mathrm{SP}_1}(\mathrm{FP},\emptyset,\emptyset)$
$\widetilde{\mathbf{R}}_{\mathrm{SP}_2}(\mathrm{FP},\emptyset,\emptyset)$
$\widetilde{\mathbf{R}}_{\mathrm{SP}_i}(\mathrm{FP},\emptyset,\emptyset)$
$\widetilde{\mathbf{R}}_{\mathrm{SP}_N}(\mathrm{FP},\emptyset,\emptyset)$
$\widetilde{\mathbf{R}}_{\mathrm{SP}_i}(\mathrm{FP},\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'),\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'))$
SP_1
SP_2
SP_i
SP_N
$\mathrm{Atk}_{\mathrm{rnd}}$
$\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})$ or $\mathrm{Atk}_{\mathrm{row}}(\mathcal{S})$
or $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})+\mathrm{Atk}_{\mathrm{row}}(\mathcal{S})$
a copy of fingerprinted database
service provider (SP)
database sharing
unauthorized redistribution
a copy of attacked fingerprinted database
fingerprint extraction
accuse the correct SP
accuse an innocent SP

(i) random bit flipping attack on vanilla fingerprinting scheme

(ii) correlation attacks on vanilla fingerprinting scheme

(iii) correlation attacks on fingerprinting scheme with mitigation techniques

Figure 2: (i) If Alice inserts fingerprinting using the vanilla scheme, and the malicious SP_i conducts random bit flipping attack, i.e., $\mathrm{Atk}_{\mathrm{rnd}}$, on its received copy, i.e., $\widetilde{\mathrm{R}}_{\mathrm{SP}_i}(\mathrm{FP},\emptyset,\emptyset)$ and redistributes the data. Then, with high probability, Alice can correctly accuse it for data leakage. (ii) If the malicious SP_i conducts any correlation attack, e.g., the column-wise correlation attack ($\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})$), the row-wise correlation attack ($\mathrm{Atk}_{\mathrm{row}}(\mathcal{S})$), or the combination of them, on the vanilla fingerprinted database. Then, with high probability, Alice cannot identify it as the traitor, and she will accuse other innocent SPs. (iii) If Alice applies the mitigation techniques, i.e., the column-wise correlation defense ($\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$) and the row-wise correlation defense ($\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$), after the vanilla fingerprinting scheme, and shares $\widetilde{\mathrm{R}}(\mathrm{FP},\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'),\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'))$ with SP_i. Then, with high probability, she can correctly identify SP_i as the traitor even if it conducts any of the correlation attack on its received copy.

4.2 Threat Model

Fingerprinted database is subject to various attacks summarized in the following sections. In Figure 2, we show some representative ones that are studied in this paper. Note that in all considered attacks, a malicious SP can change/modify most of the entries in $\widetilde{\mathbf{R}}$ to distort the fingerprint (and to avoid being accused). However, such a pirated database will have significantly poor utility (as will be introduced in Section 4.4). As discussed in Section 3, we let the vanilla fingerprint scheme only change the LSBs of data entries to preserve data utility. Thus, all considered attacks also change the LSBs of the selected entries in $\widetilde{\mathbf{R}}$ to distort the fingerprint.

4.2.1 Random Bit Flipping Attack. In this attack, to pirate a database, a malicious SP selects random entries in $\widetilde{\mathbf{R}}$ and flips their LSBs [2]. The flipped entries are still in the domain of the corresponding attributes. The considered vanilla fingerprint scheme is robust against this attack [24] as shown in Figure 2(i), Alice shares fingerprinted copies of her database by only applying FP. If a malicious SP (SP_i) tries to distort the fingerprint in $\widetilde{\mathbf{R}}(\mathrm{FP},\emptyset,\emptyset)$ using the random bit flipping attack (i.e., $\mathrm{Atk}_{\mathrm{rnd}}$), and then redistributes it, Alice can still detect SP_i's fingerprint in the pirated copy with a high probability, and correctly accuse SP_i of data leakage.

4.2.2 Subset and Superset Attacks. In subset attack, a malicious SP generates a pirated copy of $\widetilde{\mathbf{R}}$ by randomly selecting data records from it. Superset attack is the dual attack of subset attack. In this attack, the malicious SP mixes $\widetilde{\mathbf{R}}$ with other databases to create a pirated one. These two attacks are considered to be weak attacks. For example, for subset attack, to compromise just one specific bit in the inserted fingerprint bit-string, the malicious SP must exclude all records that are marked by that bit [24].

4.2.3 Correlation Attacks. In correlation attacks, a malicious SP utilizes the inherent correlations in the data to more accurately identify the fingerprinted entries, and hence distort the fingerprint.

Since the subset and superset attack are not as powerful as the bit flipping attack [33], we consider developing the correlation attacks based on the random bit flipping attack. In the following, we provide the high level descriptions of two main correlation attacks (details of these attacks are in Section 5).

In column-wise correlation attack, i.e., $\text{Atk}_{\text{col}}(\mathcal{J})$, we assume that the malicious SP has prior knowledge about the correlations among each pair of attributes (or columns in the database) characterized by the set of joint probability distributions $\mathcal{J}$. Once receiving the fingerprinted database $\widetilde{\mathbf{R}}$, the malicious SP first calculates a new set of joint probability distributions based on $\widetilde{\mathbf{R}}$. Then, it compares the new joint distributions with its prior knowledge $\mathcal{J}$, and flips the entries in $\widetilde{\mathbf{R}}$ that leads to large discrepancy between them.

In row-wise correlation attack, i.e., $\text{Atk}_{\text{row}}(\mathcal{S})$, we consider that the individuals belong to different communities (e.g., social circles decided by friendship, or families determined by kinship), and assume that the malicious SP has the prior knowledge $\mathcal{S}$, which contains (i) each individual's membership to the communities and (ii) the statistical relationships of pairs of individuals belonging to the same community. Once it receives the fingerprinted database $\widetilde{\mathbf{R}}$, the malicious SP first calculates a new set of statistical relationships based on $\widetilde{\mathbf{R}}$, then it compares the newly computed statistical relationships with $\mathcal{S}$, and changes the entries that leads to large discrepancy between the two sets of statistical relationships.

Figure 2(ii) shows the scenario, where Alice identifies the source of the data leakage wrong and accuses innocent SPs if she uses the vanilla fingerprinting scheme, whereas, SP_i conducts more advanced attacks to distort the fingerprint. These attacks include $\text{Atk}_{\text{col}}(\mathcal{J})$, $\text{Atk}_{\text{row}}(\mathcal{S})$, and the combination of them. Finally, Figure 2(iii) shows that if Alice uses the proposed mitigation techniques (i.e., $\text{Dfs}_{\text{row}}(\mathcal{S}')$ and $\text{Dfs}_{\text{col}}(\mathcal{J}')$ (as will be discussed in Section 6) after FP to improve the robustness of the added fingerprint and shares $\widetilde{\mathbf{R}}(\text{FP}, \text{Dfs}_{\text{row}}(\mathcal{S}'), \text{Dfs}_{\text{col}}(\mathcal{J}'))$, then, even though SP_i conducts the identified correlation attacks, Alice can still identify SP_i to be responsible for leaking the data with high probability.

4.2.4 Collusion Attack. Fingerprinted databases are also susceptible to collusion attack, where multiple malicious SPs ally together to generate a pirated database from their unique fingerprinted copies. In cryptography literature, many works have attempted to develop collusion resistant fingerprinting schemes [7, 8, 27, 32]. Our proposed mitigation techniques can also be used with a collusion-resistant vanilla fingerprinting scheme [8] to provide some level of robustness against colluding SP. In this work, we mainly focus on correlation attacks from a single-handed malicious SP. We will extend our work in the scenario of colluding SPs in future work.

4.3 Fingerprint Robustness Metrics

The primary goal of a malicious SP is to distort the fingerprint in $\widetilde{\mathbf{R}}$, thus we consider the following fingerprint robustness metrics about a pirated database $\overline{\mathbf{R}}$ generated by launching attacks on $\widetilde{\mathbf{R}}$.

4.3.1 Number of compromised fingerprint bits. We formulate the number of compromised fingerprint bits as

$$\text{num}_{\text{cmp}} = \textstyle\sum_{l=1}^{L} \mathbf{1}\{f(l) \neq \overline{f}(l)\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function, L is the length of the fingerprint bit-string, $\overline{f}$ is the extracted fingerprint bit-string from $\overline{\mathbf{R}}$, and $f(l)$ (or $\overline{f}(l)$) is the lth bit in f (or $\overline{f}$).

4.3.2 Accusable ranking of a malicious SP. We quantify the confidence of accusing the correct malicious SP by defining the accusable ranking metric (denoted as r) as follows:

$$r = \begin{cases} \text{“uniquely accusable”}, \textit{if } m_0 > \sum_{l=1}^{L} \mathbf{1}\left\{f_{\text{SP}_i}(l) = \overline{f}(l)\right\}, \forall \text{SP}_i \in \mathcal{T} \\ \text{“top } t \text{ accusable”}, \textit{otherwise} \end{cases},$$

where $m_0 = \sum_{l=1}^{L} \mathbf{1}\{f_{\text{SP}_{\text{malicious}}}(l) = \overline{f}(l)\}$ is the number of bit matches between the malicious SP's fingerprint and the extracted fingerprint from the pirated database, and $\mathcal{T}$ is the set of all innocent SPs. Specifically, if the malicious SP has the most bit matches with the extracted fingerprint, Alice will uniquely accuse it. Otherwise, we compute $t = \frac{\sum_{\text{SP}_i \in \mathcal{T}} \mathbf{1}\left\{\left(\sum_{l=1}^{L} \mathbf{1}\left\{f_{\text{SP}_i}(l) = \overline{f}(l)\right\}\right) \geq m_0\right\}}{|\mathcal{T}|} \times 100\%$, which is the fraction of innocent SPs having more bit matches with the extracted fingerprint than the malicious SP. For example, if $t = 80\%$, then the malicious SP is only top 80% accusable, which suggests that Alice will accuse other innocent SPs with high probability. In contrast, if $t = 1\%$, then the malicious SP's accusable ranking increases and makes it among the top 1% accusable SPs, and Alice will accuse other innocent SPs with low probability. Essentially, a high accusable rank r corresponds to either (i) a "low t" or (ii) the uniquely accusable case. As we will show in Section 7, for a malicious SP to avoid being accused (i.e., have low accusable rank, or high t value), it needs to distort more than half of the fingerprint bits. As we will also show via evaluations, the malicious SP can easily achieve this goal if it applies the identified correlation attacks. Whereas, if it applies the random bit flipping attack, it becomes "uniquely accusable" with high probability unless it overdistort the fingerprinted database.

According to the vanilla scheme [24], the probability of extracting a valid fingerprint from a database that does not belong to Alice (i.e., misdiagnosis false hit) is upper bounded by $|\text{SP}|/2^L$, and the probability of extracting an incorrect but valid fingerprint from the fingerprinted database (i.e., misattribution false hit) is upper bounded by $(|\text{SP}|-1)/2^L$. Since these are all negligible probabilities, we do not consider the case in which Alice does not accuse any SP when a copy of her database is leaked in the experiments.

4.4 Utility Metrics

Fingerprinting naturally changes the content of the database, and hence degrades its utility. We quantify the utility of a fingerprinted database using the following metrics.

4.4.1 Accuracy of $\widetilde{\mathbf{R}}$. We quantify the accuracy of $\widetilde{\mathbf{R}}$ as

$$Acc(\widetilde{\mathbf{R}}) = 1 - \widetilde{\mathbf{R}} \oplus \mathbf{R}/M * L,$$

where $\oplus$ is the symmetric difference operator that counts the number of different entries in the fingerprinted and the original databases. $Acc(\widetilde{\mathbf{R}})$ measures the percentage of matched entries between the fingerprinted and the original databases.

4.4.2 Preservation of column-wise correlations. We quantify the preservation of column-wise correlations in the database as

$$\mathrm{P}_{\mathrm{col}}(\widetilde{\mathbf{R}}) = 1 - \frac{\sum_{p,q\in\mathcal{F},p\neq q}\sum_{a\in p,b\in q}\mathbf{1}\{|\widetilde{J_{p,q}}(a,b) - J_{p,q}(a,b)| \geq \tau_{\mathrm{col}}\}}{\sum_{p,q\in\mathcal{F},p\neq q} k_p k_q},$$

where p and q are two attributes in the attribute set $\mathcal{F}$, k_p (or k_q) stands for the number of unique instances of attribute p (or q), and $\widetilde{J_{p,q}}(a,b)$ (or $J_{p,q}(a,b)$) is the joint probability that attribute p takes value a and attribute q takes value b in $\widetilde{\mathbf{R}}$ (or $\mathbf{R}$). $\mathrm{P}_{\mathrm{col}}$ calculates the fraction of instances of $|\widetilde{J_{p,q}}(a,b) - J_{p,q}(a,b)|$ that do not exceed a predetermined threshold τ_{col} before and after fingerprinting $\mathbf{R}$.

4.4.3 Preservation of row-wise correlations. We quantify the preservation of row-wise correlations in the database as

$$\mathrm{P}_{\mathrm{row}}(\widetilde{\mathbf{R}}) = 1 - \frac{\sum_{c=1}^{C}\sum_{i,j\in\mathrm{comm}_c,i\neq j}\mathbf{1}\{|\widetilde{s_{i,j}}^{\mathrm{comm}_c} - s_{i,j}^{\mathrm{comm}_c}| \geq \tau_{\mathrm{row}}\}}{\sum_{c=1}^{C} n_c(n_c-1)},$$

where comm_c represents the set of all individuals in a community c, $\widetilde{s_{i,j}}^{\mathrm{comm}_c}$ (or $s_{i,j}^{\mathrm{comm}_c}$) is the statistical relationship between individual i and j belonging to comm_c in $\widetilde{\mathbf{R}}$ (or $\mathbf{R}$), n_c is the number of individuals in comm_c, and C is the number of communities. In essence, $\mathrm{P}_{\mathrm{row}}(\widetilde{\mathbf{R}})$ evaluates the fraction of statistical relationship that has absolute difference less than τ_{row} in the entire population before and after fingerprinting.

4.4.4 Preservation of empirical covariance matrix. We quantify the preservation of empirical covariance matrix of the database as

$$\mathrm{P}_{\mathrm{cov}} = 1 - ||\mathrm{cov}(\widetilde{\mathbf{R}}) - \mathrm{cov}(\mathbf{R})||_F/||\mathrm{cov}(\mathbf{R})||_F,$$

where $\mathrm{cov}(\mathbf{R}) = \sum_{i=1}^{M} \boldsymbol{r}_i^T \boldsymbol{r}_i/M$ is the empirical covariance matrix of data records in $\mathbf{R}$. $\mathrm{P}_{\mathrm{cov}}$ evaluates the similarity between the covariance matrices of the database before and after fingerprinting. We consider this metric because the fingerprinted database may also be used in data analysis tasks, and empirical covariance matrix is frequently utilized to establish predictive models, e.g., regression and probability fitting [11, 20]. Besides, multivariate data analysis often involves the investigation of inter-relationships among data records which requires an accurate covariance matrix estimation.

Note that the utility of the pirated database $\overline{\mathbf{R}}$ generated by the malicious SP can also be quantified using the same metrics, i.e., $Acc(\overline{\mathbf{R}})$, $\mathrm{P}_{\mathrm{col}}(\overline{\mathbf{R}})$, $\mathrm{P}_{\mathrm{row}}(\overline{\mathbf{R}})$, and $\mathrm{P}_{\mathrm{cov}}(\overline{\mathbf{R}})$. As discussed, a malicious SP can successfully (without being accused) distort the fingerprint easily by over-distorting $\widetilde{\mathbf{R}}$, however, to preserve the data utility, a rational malicious SP will not over-distort a database.

In addition to the general utility metrics defined above, we will also consider specific statistical utilities, e.g., portion of individuals that have a particular education degree or higher, and the standard deviation of individuals' age distribution. It is noteworthy that if the general utility metrics are high, it implicitly suggests high utility for the specific statistical (or other application related) utilities.

5 IDENTIFIED CORRELATION ATTACKS

In the correlation attacks, we assume that the malicious SP has access to both column- and row-wise correlations of Alice's database, which contains (i) correlations between all pairs of attributes (columns), (ii) each individual's membership to the communities and (iii) the statistical relationships of pairs of individuals belonging to the same community. Specifically, the column-wise correlations are characterized by the set of joint distributions among pairs of attributes (columns) in the database, i.e., $\mathcal{J} = \{J_{p,q}|p,q \in \mathcal{F}, p \neq q\}$. Row-wise correlations, on the other hand, are characterized by the set of statistical relationships between pairs of individuals (rows) in a community. For instance, $\mathcal{S} = \{s_{ij}^{\mathrm{comm}_c}|i,j \in \mathrm{comm}_c, i \neq j, c \in [1,C]\}$, where $s_{ij}^{\mathrm{comm}_c} = e^{-\mathrm{dist}(\boldsymbol{r}_i,\boldsymbol{r}_j)}$ is the statistical relationship between individuals (data records) i and j in community comm_c ($\mathrm{dist}(\boldsymbol{r}_i,\boldsymbol{r}_j)$ denotes the Hamming distance between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$). Since the added fingerprint changes some entries in the original database, which will lead to the change of both joint distributions and statistical relationships, the malicious SP can utilize its auxiliary (publicly available) information about $\mathcal{J}$ and $\mathcal{S}$ to identify the positions of suspicious entries in $\widetilde{\mathbf{R}}$ that are potentially fingerprinted.

5.1 Column-wise Correlation Attack

To launch the column-wise correlation attack ($\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})$) on $\widetilde{\mathbf{R}}$, the malicious SP first calculates the empirical joint distributions among pairs of attributes in $\widetilde{\mathbf{R}}$, denoted as $\widetilde{\mathcal{J}}$. Then, it compares each joint distribution in $\widetilde{\mathcal{J}}$ (i.e., $\widetilde{J_{p,q}}$) with that in $\mathcal{J}$ (i.e., $J_{p,q}$). For instance, if the absolute difference of joint probabilities when attribute p takes value a and attribute q takes value b (i.e., $|J_{p,q}(a,b) - \widetilde{J_{p,q}}(a,b)|$) is higher than a threshold $\tau_{\mathrm{col}}^{\mathrm{Atk}}$, then, the malicious SP queries the row indices of the data records in $\widetilde{\mathbf{R}}$ whose attributes p and q take values a and b, respectively, and collects the corresponding row indices in a set $\mathcal{I}$, i.e., for the previous example, $\mathcal{I}$ = row index query$\big(\widetilde{\mathbf{R}}.p == a$ and $\widetilde{\mathbf{R}}.q == b\big)$ ($\widetilde{\mathbf{R}}.p$ includes attribute p of all data record in database $\widetilde{\mathbf{R}}$). For each row index $i \in \mathcal{I}$, either position $\{i,p\}$ or $\{i,q\}$ (i.e., the row index and attribute tuple) can be potentially fingerprinted, because they both affect the joint distribution $\widetilde{J_{p,q}}(a,b)$. Thus, the malicious SP adds each of these tuples, i.e., $\{i,p\}$ and $\{i,q\}, i \in \mathcal{I}$ into a suspicious position set denoted as $\mathcal{P}$.

Since a specific suspicious row index i can be associated with multiple attributes in the suspicious position set $\mathcal{P}$, the suspicious attribute that is most frequently associated with i is considered to be **highly suspicious**. The malicious SP collects these highly suspicious combinations of row index and attribute in a set $\mathcal{H} = \mathcal{H} \cup \{i, \mathrm{mode}(\mathcal{A}_i)\}$, where $\mathcal{A}_i$ includes all the attributes that are paired with row index i in set $\mathcal{P}$, and $\mathrm{mode}(\mathcal{A}_i)$ returns the most frequent attribute in $\mathcal{A}_i$ (if there is a tie, the malicious SP randomly chooses one). Then, the malicious LSP launches the column-wise correlation attack by flipping the LSB of entries in $\widetilde{\mathbf{R}}$ whose positions are in $\mathcal{H}$, i.e., $\widetilde{\mathbf{R}}.(i,p), \forall\{i,p\} \in \mathcal{H}$ ($\widetilde{\mathbf{R}}.(i,p)$ represents the value of attribute p for the ith data record in $\widetilde{\mathbf{R}}$).

In practice, the malicious SP can launch multiple rounds of $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})$ by iteratively comparing the new joint distributions obtained from the attacked fingerprinted database in the previous round with its prior knowledge $\mathcal{J}$. In each round, a new $\mathcal{H}$ is constructed, but the malicious SP does not flip the highly suspicious positions that have already been flipped in previous rounds. This can be achieved by maintaining and updating an accumulative highly suspicious position set $\mathcal{Z}$. We summarize the steps of conducting t rounds of $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})$ in Algorithm 3.

Algorithm 3: $\text{Atk}_{\text{col}}(\mathcal{J})$: Column-wise Correlation Attack

```
Input  : Fingerprinted database $\widetilde{\mathbf{R}}$, malicious SP's prior knowledge
         on the pairwise joint distributions among attributes, $\mathcal{J}$,
         and attack rounds $t$.
Output : column-wise correlation attacked DB $\overline{\mathbf{R}}\Big(\emptyset, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J})\Big)$.
Initialize $cnt = 1$;
Initialize $\mathcal{Z} = \emptyset$;
while $cnt \leq t$ do
    Initialize $\mathcal{P} = \emptyset, \mathcal{H} = \emptyset$;
    Update the empirical joint distributions set $\widetilde{\mathcal{J}}$ using $\widetilde{\mathbf{R}}$;
    forall $p, q \in \mathcal{F}, p \neq q$ do
        forall $a \in [0, k_p - 1], b \in [0, k_q - 1]$ do
            if $\big|J_{p,q}(a, b) - \widetilde{J_{p,q}}(a, b)\big| \geq \tau_{\text{col}}^{\text{Atk}}$ then
                $\mathcal{I}$ = row index query$(\widetilde{\mathbf{R}}.p == a$ and $\widetilde{\mathbf{R}}.q == b)$;
                forall row index $i \in \mathcal{I}$ do
                    if $\{i, p\} \notin \mathcal{P}$ then
                        $\mathcal{P} = \mathcal{P} \cup \{i, p\}$;
                    end
                    if $\{i, q\} \notin \mathcal{P}$ then
                        $\mathcal{P} = \mathcal{P} \cup \{i, q\}$;
                    end
                end
            end
        end
    end
    forall row index and attribute tuple $\{i, p\} \in \mathcal{P}$ do
        Collect all attributes paired with row index $i$ into set $\mathcal{A}_i$
        $\mathcal{H} = \mathcal{H} \cup \{i, \text{mode}(\mathcal{A}_i)\}$;//the most frequent
         attribute associate with row index $i$ is
         recorded in $\mathcal{H}$.
    end
    forall highly suspicious row index and attribute tuple
     $\{i, p\} \in \mathcal{H}$ do
        if $\{i, p\} \notin \mathcal{Z}$ then
            Change the LSB of $\widetilde{\mathbf{R}}.(i, p)$;
            $\mathcal{Z} = \mathcal{Z} \cup \{i, p\}$;
        end
    end
    $cnt = cnt + 1$;
end
Return $\overline{\mathbf{R}}\Big(\emptyset, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J})\Big) = \widetilde{\mathbf{R}}$;
```

Next, we show that a malicious SP can increase its inference power (confidence) about whether a particular entry in the database is fingerprinted or not by launching $\text{Atk}_{\text{col}}(\mathcal{J})$. In Section 7, we experimentally validate this finding using a real-world database. Under Atk_{rnd}, we denote the malicious SP's confidence that an entry, whose attribute p takes value a in the original database ($\mathbf{R}$), is changed due to the fingerprinting as $\text{Conf}_{\text{Atk}_{\text{rnd}}}(\frac{1}{\gamma}; p, a)$. Likewise, under $\text{Atk}_{\text{col}}(\mathcal{J})$, we represent such confidence as $\text{Conf}_{\text{Atk}_{\text{col}}(\mathcal{J})}(\frac{1}{\gamma}; p, a)$. Here, $\gamma \in (0, 1)$ is the fingerprinting ratio and we use $\frac{1}{\gamma}$ as the decision parameter to investigate the asymptotic behavior of the malicious SP's confidence gain, which is defined as the ratio $G_{\text{col}}(\frac{1}{\gamma}; p, a) = \text{Conf}_{\text{Atk}_{\text{col}}(\mathcal{J})}(\frac{1}{\gamma}; p, a)\Big/\text{Conf}_{\text{Atk}_{\text{rnd}}}(\frac{1}{\gamma}; p, a)$. Thus, we have the following proposition.

Proposition 1. *By launching* $\text{Conf}_{\text{Atk}_{\text{col}}(\mathcal{J})}$, *the malicious SP's confidence gain about an entry, whose attribute* p *takes value* a *in* $\mathbf{R}$, *is fingerprinted can be shown in an asymptotic manner as*

$$G_{\text{col}}(\frac{1}{\gamma}; p, a) = \Theta\left(\left(1 - \prod_{q\in\mathcal{T}, q\neq p}\left(\frac{\tau_{\text{col}}^{\text{Atk}}}{\frac{\gamma}{|\mathcal{T}|}2freq_a^p}\right)^{k_q}\right)\Bigg/\left(\frac{\gamma}{|\mathcal{T}|}freq_a^p\right)\right),$$

where $freq_a^p$ *is the frequency of records with attribute* p *taking value* a *in* $\mathbf{R}$, k_q *is the number of different values for attribute* q, *and* $\Theta(\cdot)$ *is the Big-Theta notation.*

Proof Sketch. For the vanilla fingerprinting scheme, we have $\text{Conf}_{\text{Atk}_{\text{rnd}}}(p, a) = \frac{\gamma}{|\mathcal{T}|}freq_a^p$. When launching the $\text{Atk}_{\text{col}}(\mathcal{J})$, the malicious SP will add the corresponding suspicious row index and attribute tuple in $\mathcal{P}$ if $\big|J_{p,q}(a, b) - \widetilde{J_{p,q}}(a, b)\big| \geq \tau_{\text{col}}^{\text{Atk}}$. Thus, we have $\text{Conf}_{\text{Atk}_{\text{col}}(\mathcal{J})}(p, a) = 1 - \prod_{q\in\mathcal{T}, q\neq p}\prod_{b\in[0,k_q-1]}\Pr\left(\big|J_{p,q}(a, b) - \widetilde{J_{p,q}}(a, b)\big| < \tau_{\text{col}}^{\text{Atk}}\right)$. Since the inserted fingerprint will cause $J_{p,q}(a, b) - \widetilde{J_{p,q}}(a, b)$ vary in the range of $\left[-\frac{\gamma}{|\mathcal{T}|}2freq_{a,b}^{p,q}, \frac{\gamma}{|\mathcal{T}|}2freq_{a,b}^{p,q}\right]$, where $freq_{a,b}^{p,q}$ is the frequency of entries whose attributes p and q take values a and b in $\mathbf{R}$. Then, $|J_{p,q}(a, b) - \widetilde{J_{p,q}}(a, b)|$ can be shown as a random variable attributed to an uniform distribution in the support of $\left[0, \frac{\gamma}{|\mathcal{T}|}2freq_{a,b}^{p,q}\right]$, which leads to $\text{Conf}_{\text{Atk}_{\text{col}}(\mathcal{J})}(p, a) = 1 - \prod_{q\in\mathcal{T}, q\neq p}\prod_{b\in[0,k_q-1]}\tau_{\text{col}}^{\text{Atk}}\Big/\left(\frac{\gamma}{|\mathcal{T}|}2freq_a^p\right)$. By applying arithmetic-geometric mean inequality along with the fact $\sum_{b\in[0,k_q-1]}freq_{a,b}^{p,q} = freq_a^p, \forall q \neq p$, we can complete the proof. □

Remark 1. *We aim at presenting a generic confidence gain achieved from* $\text{Atk}_{\text{col}}(\mathcal{J})$, *thus we consider the potential fingerprinted entries in the suspicious set* $\mathcal{P}$ *instead of the highly suspicious set* $\mathcal{H}$. *In practice, the generation process of* $\mathcal{H}$ *from* $\mathcal{P}$ *heavily depends on the data distribution in the considered databases.*

5.2 Row-wise Correlation Attack

Since the malicious SP has access to both individuals' memberships to communities and row-wise correlations, i.e., $\mathcal{S}$, after receiving the fingerprinted database, it can compute a new set of statistical relationships among pairs of individuals in each of the communities using $\widetilde{\mathbf{R}}$, i.e., $\widetilde{\mathcal{S}} = \{\widetilde{s_{ij}}^{\text{comm}_c}|i, j \in \text{comm}_c, i \neq j, c \in [1, C]\}$, where $\widetilde{s_{ij}}^{\text{comm}_c} = e^{-\text{dist}(\widetilde{\boldsymbol{r}_i}, \widetilde{\boldsymbol{r}_j})}$ is the statistical relationship between the ith and jth data records (i.e., $\widetilde{\boldsymbol{r}_i}$ and $\widetilde{\boldsymbol{r}_j}$) in $\widetilde{\mathbf{R}}$. Then, to conduct $\text{Atk}_{\text{row}}(\mathcal{S})$, the malicious SP flips the LSBs of all attributes of a data record $\boldsymbol{r}_i$, if the cumulative absolute difference of its statistical relationships with respect to other records in the same community exceeds a predetermined threshold $\tau_{\text{row}}^{\text{Atk}}$ after fingerprinting, i.e., $\sum_{j\neq i}^{n_c}\big|s_{ij}^{\text{comm}_c} - \widetilde{s_{ij}}^{\text{comm}_c}\big| \geq \tau_{\text{row}}^{\text{Atk}}, i, j \in \text{comm}_c$. The rationale behind this is because the row-wise statistical information $\mathcal{S}$ is calculated using the entire data records, instead of individual entries between the rows. Although, this represents the strongest row-wise attack as it changes all the entries of a given data record, in practice, $\text{Atk}_{\text{row}}(\mathcal{S})$ changes only a limited number of data records, as will be shown in Section 7.2.1. We summarize the steps to launch $\text{Atk}_{\text{row}}(\mathcal{S})$ on $\widetilde{\mathbf{R}}$ in Algorithm 4.

Algorithm 4: $\text{Atk}_{\text{row}}(\mathcal{S})$: Row-wise correlation attack

```
Input  : Fingerprinted database, $\widetilde{\mathbf{R}}$, malicious SP's prior knowledge
         on the row-wise correlations $\mathcal{S}$ and individuals' affiliation
         to the $C$ communities.
Output : $\overline{\mathbf{R}}\Big(\emptyset,\ \text{Atk}_{\text{row}}(\mathcal{S}),\ \emptyset,\ \Big)$.
Obtain the new set of pairwise statistical relationships among
  individuals in each community from $\widetilde{\mathbf{R}}$, i.e., $\widetilde{\mathcal{S}}$;
forall $\text{comm}_c$, $c \in [1, C]$ do
    forall individual $i \in \text{comm}_c$ do
        if $\sum_{j\neq i}^{n_c} \left|s_{ij}^{\text{comm}_c} - \widetilde{s_{ij}}^{\text{comm}_c}\right| \geq \tau_{\text{row}}^{\text{Atk}}$ then
            Flip the LSBs of all attributes of $\boldsymbol{r}_i$ in $\widetilde{\mathbf{R}}$;
        end
    end
end
Return $\overline{\mathbf{R}}\Big(\emptyset,\ \text{Atk}_{\text{row}}(\mathcal{S}),\ \emptyset,\ \Big) = \widetilde{\mathbf{R}}$.
```

We analyze the impact of $\text{Atk}_{\text{row}}(\mathcal{S})$ by denoting the malicious SP's confidence that an entry $(\boldsymbol{r}_i)$ is fingerprinted as $\text{Conf}_{\text{Atk}_{\text{rnd}}}(\frac{1}{\gamma};\boldsymbol{r}_i)$ and $\text{Conf}_{\text{Atk}_{\text{row}}(\mathcal{S})}(\frac{1}{\gamma};\boldsymbol{r}_i)$, under Atk_{rnd} and $\text{Atk}_{\text{row}}(\mathcal{S})$, respectively. Then, the confidence gain of the malicious SP is $G_{\text{row}}(\frac{1}{\gamma};\boldsymbol{r}_i) = \frac{\text{Conf}_{\text{Atk}_{\text{row}}(\mathcal{S})}(\frac{1}{\gamma};\boldsymbol{r}_i)}{\text{Conf}_{\text{Atk}_{\text{rnd}}}(\frac{1}{\gamma};\boldsymbol{r}_i)}$, which is calculated in the following proposition.

Proposition 2. *By launching* $\text{Conf}_{\text{Atk}_{\text{row}}(\mathcal{S})}$, *the malicious SP's maximum confidence gain about an entry in* $\mathbf{R}$ *is fingerprinted can be shown asymptotically as*

$$G_{\text{row}}(\frac{1}{\gamma};\boldsymbol{r}_i) = \Theta\left(\left(1-\sum_{j=0}^{\lfloor \tau_{\text{row}}^{\text{Atk}} \rfloor}\binom{n_c-1}{j}(2\gamma-\gamma^2)^j(1-\gamma)^{2(n_c-1-j)}\right)\Big/\gamma\right).$$

Proof Sketch. Clearly, $\text{Conf}_{\text{Atk}_{\text{rnd}}}(\boldsymbol{r}_i) = \gamma$. According to Algorithm 4, $\text{Conf}_{\text{Atk}_{\text{row}}(\mathcal{S})}(\frac{1}{\gamma};\boldsymbol{r}_i) = \Pr(\sum_{j\neq i}^{n_c} \left|e^{-\text{dist}(\boldsymbol{r}_i,\boldsymbol{r}_j)} - e^{-\text{dist}(\widetilde{\boldsymbol{r}_i},\widetilde{\boldsymbol{r}_j})}\right| \geq \tau_{\text{row}}^{\text{Atk}}) \overset{*}{\approx} \Pr(\sum_{j\neq i}^{n_c} \left|\text{dist}(\boldsymbol{r}_i,\boldsymbol{r}_j) - \text{dist}(\widetilde{\boldsymbol{r}_i},\widetilde{\boldsymbol{r}_j})\right| \geq \tau_{\text{row}}^{\text{Atk}})$, where $*$ is due to the Taylor approximation and the assumption that the distance between individuals in the same community is small. Then, $\left|\text{dist}(\boldsymbol{r}_i,\boldsymbol{r}_j) - \text{dist}(\widetilde{\boldsymbol{r}_i},\widetilde{\boldsymbol{r}_j})\right|$ can be shown as a Bernoulli random variable, which is 0 with probability $(1-\gamma)^2$, and is nonzero with probability $2\gamma - \gamma^2$. Since the summation of Bernoulli random variable is attributed to binomial distribution, we can finish the proof. □

Remark 2. *In the above analysis, we ignored the scenario where* $\left|\text{dist}(\boldsymbol{r}_i,\boldsymbol{r}_j) - \text{dist}(\widetilde{\boldsymbol{r}_i},\widetilde{\boldsymbol{r}_j})\right|$ *is 2 with probability* γ^2 *to avoid extra heavy notations. In the experiments, we set* $\gamma = 1/35$, *thus,* γ^2 *is negligible.*

5.3 Integrated Correlation Attack

In practice, the malicious SP will apply $\text{Atk}_{\text{row}}(\mathcal{S})$ followed by $\text{Atk}_{\text{col}}(\mathcal{J})$ if it launches the integrated correlation attack. This is because (i) $\text{Atk}_{\text{row}}(\mathcal{S})$ is computationally light and modifies significantly less entries in $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ compared to $\text{Atk}_{\text{col}}(\mathcal{J})$ (as we will show in Section 7.2.1). (ii) If $\text{Atk}_{\text{col}}(\mathcal{J})$ is applied first, it will change the row-wise correlations (P_{row}) significantly, yet, if $\text{Atk}_{\text{row}}(\mathcal{S})$ is applied first, it only has a small impact on the column-wise correlations P_{col} (as we will also show in Section 7.2.1). Algorithm 5 summarizes the major steps of this integrated attack. Note that, in practice, there is no minimum distribution difference requirement to perform the proposed attacks, because a malicious SP can always reduce the value of $\tau_{\text{col}}^{\text{Atk}}$ and $\tau_{\text{row}}^{\text{Atk}}$ to obtain more potentially fingerprinted entries.

Algorithm 5: Integrated correlation attack

```
Input  : Fingerprinted database, $\widetilde{\mathbf{R}}$, malicious SP's prior knowledge
         on the row-wise correlations $\mathcal{S}$, individuals' affiliation to
         the $C$ communities, and column-wise correlations $\mathcal{J}$.
Output : $\overline{\mathbf{R}}\Big(\emptyset,\ \text{Atk}_{\text{row}}(\mathcal{S}),\ \text{Atk}_{\text{col}}(\mathcal{J})\Big)$.
Launch row-wise correlation attack $\text{Atk}_{\text{row}}(\mathcal{S})$ on $\widetilde{\mathbf{R}}$ using
  Algorithm 4, and obtain $\overline{\mathbf{R}}\Big(\emptyset,\ \text{Atk}_{\text{row}}(\mathcal{S}),\ \emptyset\Big)$;
Launch column-wise correlation attack $\text{Atk}_{\text{col}}(\mathcal{J})$ on
  $\overline{\mathbf{R}}\Big(\emptyset,\ \text{Atk}_{\text{row}}(\mathcal{S}),\ \emptyset\Big)$ using Algorithm 3, obtain and return
  $\overline{\mathbf{R}}\Big(\emptyset,\ \text{Atk}_{\text{row}}(\mathcal{S}),\ \text{Atk}_{\text{col}}(\mathcal{J})\Big)$.
```

By taking advantage of the correlation models, the identified attacks (in Sections 5.1 and 5.2) are more powerful than the traditional random bit flipping attack (in Section 4.2). As we will show in Section 7.2.1, to effectively distort the added fingerprint and cause Alice to accuse innocent SPs with high probability, a malicious SP only needs to change a small fraction of entries in the fingerprinted database if it conducts the correlation attacks on $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$. In contrast, to achieve a similar attack performance, the random bit flipping attack needs to change more than 80% of the entries in $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, which results in a significant loss in database utility. Thus, the correlation attacks not only distort the inserted fingerprint but they also maintain a high utility for the pirated database.

Due to the identified vulnerability of existing fingerprinting schemes for relations against correlation attacks, it is critical to develop defense mechanisms that can mitigate these attacks. In the next section, we discuss how to develop robust fingerprinting techniques against both column- and row-wise correlation attacks.

6 ROBUST FINGERPRINTING AGAINST IDENTIFIED CORRELATION ATTACKS

Now, we propose robust fingerprinting schemes against the identified correlation attacks that can serve as post-processing steps for any off-the-shelf (vanilla) fingerprinting schemes. To provide robustness against column- and row-wise correlation attack, i.e., $\text{Atk}_{\text{col}}(\mathcal{J})$ and $\text{Atk}_{\text{row}}(\mathcal{S})$, the database owner (Alice) utilizes her prior knowledge $\mathcal{J}'$ and $\mathcal{S}'$ as the reference column-wise joint distributions and statistical relationships, respectively. We will show that to implement the proposed mitigation techniques, Alice needs to change only a few entries (e.g., less than 3%) in $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, such that the post-processed fingerprinted database has column-wise correlation close to $\mathcal{J}'$ and row-wise correlation far from $\mathcal{S}'$.

6.1 Robust Fingerprinting Against Column-wise Correlation Attack

6.1.1 Mitigation via mass transportation. To make a vanilla fingerprinting scheme robust against column-wise correlation attack, the main goal of the proposed technique $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$ is to transform $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$ to have column-wise joint distributions close to the reference joint distributions in $\mathcal{J}'$. We develop $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$ using "optimal transportation" [13], which moves the probability mass of the marginal distribution of each attribute in $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$ to resemble the distribution obtained from the marginalization of each reference joint distribution in $\mathcal{J}'$. Then, the optimal transportation plan is used to change the entries in each attribute of $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$ to obtain $\widetilde{\mathbf{R}}\big(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\big)$. While doing this, the new empirical joint distributions calculated from $\widetilde{\mathbf{R}}\big(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\big)$ also become close to the ones in $\mathcal{J}'$.

In particular, for a specific attribute (column) p, we denote its marginal distribution obtained from the (vanilla) fingerprinted database as $\Pr(C_{\widetilde{p}})$, and that obtained from the marginalization of a reference $J'_{p,q}$ distribution in $\mathcal{J}'$ as $\Pr(C_{p'}) = J'_{p,q}\mathbf{1}^T$ (q can be any attribute that is different from p, because the marginalization with respect to p using different $J'_{p,q}$ will lead to the identical marginal distribution of p). To move the mass of $\Pr(C_{\widetilde{p}})$ to resemble $\Pr(C_{p'})$, we need to find another joint distribution (i.e., the mass transportation plan) $G_{\widetilde{p},p'} \in \mathcal{R}^{k_p \times k_p}$ (k_p is the number of possible values that attribute p can take), whose marginal distributions are identical to $\Pr(C_{\widetilde{p}})$ and $\Pr(C_{p'})$. Let a and b be two distinct values that attribute p can take ($a, b \in [0, k_p - 1]$). Then, $G_{\widetilde{p},p'}(a, b)$ indicates that the database owner should change $G_{\widetilde{p},p'}(a, b)$ percentage of entries in $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$ whose attribute p takes value a (i.e., $p = a$) to value b (i.e., change them to make $p = b$), so as to make $\Pr(C_{\widetilde{p}})$ close to $\Pr(C_{p'})$. In practice, such a transportation plan can be obtained by solving a regularized optimal transportation problem, i.e., the entropy regularized Sinkhorn distance minimization [16] as follows:

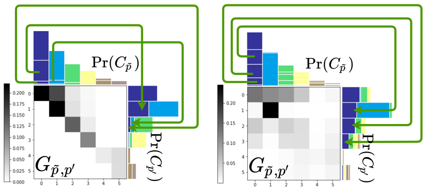
$$d\Big(\Pr(C_{\widetilde{p}}), \Pr(C_{p'}), \lambda_p\Big) = \min_{G_{\widetilde{p},p'} \in \mathcal{G}\left(\Pr(C_{\widetilde{p}}), \Pr(C_{p'})\right)} < G_{\widetilde{p},p'}, \Theta_{\widetilde{p},p'} >_F - \frac{H(G_{\widetilde{p},p'})}{\lambda_p}, \qquad (1)$$

where $\mathcal{G}\big(\Pr(C_{\widetilde{p}}), \Pr(C_{p'})\big) = \big\{G \in \mathcal{R}^{k_p \times k_p} \big| G\mathbf{1} = \Pr(C_{\widetilde{p}}), G^T\mathbf{1} = \Pr(C_{p'})\big\}$ is the set of all joint probability distributions whose marginal distributions are the probability mass functions of $\Pr(C_{\widetilde{p}})$ and $\Pr(C_{p'})$. $< \cdot, \cdot >_F$ denotes the Frobenius inner product of two matrices with the same size. Also, $\Theta_{\widetilde{p},p'}$ is the transportation cost matrix and $\Theta_{\widetilde{p},p'}(a, b) > 0$ represents the cost to move a unit percentage of mass from $\Pr(C_{\widetilde{p}} = a)$ to $\Pr(C_{\widetilde{p}} = b)$. Finally, $H(G_{\widetilde{p},p'}) = - < G_{\widetilde{p},p'}, \log G_{\widetilde{p},p'} >_F$ calculates the information entropy of $G_{\widetilde{p},p'}$ and $\lambda_p > 0$ is a tuning parameter. In practice, (1) can be solved by iteratively rescaling rows and columns of the initialized $G_{\widetilde{p},p'}$ to have desired marginal distributions. The obtained $G_{\widetilde{p},p'}$ is more heterogeneous for larger values of λ_p. This suggests that the transportation plan tends to move the mass of $\Pr(C_{\widetilde{p}} = a)$ to the adjacent instances, i.e, $b = a - 1$ or $b = a + 1$. In contrast, the obtained $G_{\widetilde{p},p'}$ is more homogeneous for smaller values of λ_p, which suggests that the transportation plan tends to move the mass of $\Pr(C_{\widetilde{p}} = a)$ to all other instances. A homogeneous plan makes $\Pr(C_{\widetilde{p}})$ much closer to $\Pr(C_{p'})$ after the mass transportation, but it causes more data entries to be changed, and results in a higher decrease in the database utility. On the other hand, a heterogeneous plan changes less data entries by tolerating a larger difference between $\Pr(C_{\widetilde{p}})$ and $\Pr(C_{p'})$ after the mass transportation. In the evaluation (in Section 7), we will try different values of λ_p to strike a balance between the mitigation performance and data utility.

6.1.2 A toy example on mass transportation. To illustrate $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$ via mass transportation of $\Pr(C_{\widetilde{p}})$ to resemble $\Pr(C_{p'})$, we use a pair of discrete probability distributions shown in Figure 3(a) as an example, and demonstrate the transportation plans obtained by solving (1) when $\lambda_p = 500$ and $\lambda_p = 50$ in Figures 3(b) and (c), respectively. In Figure 3(b), we have a heterogeneous $G_{\widetilde{p},p'}$, which often moves the mass to adjacent instances, e.g., the mass of $\Pr(C_{\widetilde{p}} = 0)$ is divided into 3 parts and a larger portion of mass is moved to $\Pr(C_{\widetilde{p}} = 1)$. $G_{\widetilde{p},p'}(0, 1) = 0.157$, thus 0.157 mass of $\Pr(C_{\widetilde{p}} = 0)$ is moved to $\Pr(C_{\widetilde{p}} = 1)$. In Figure 3(c), we obtain a homogeneous $G_{\widetilde{p},p'}$, which distributes the mass to many other instances. For example, the mass of $\Pr(C_{\widetilde{p}} = 0)$ is divided into 5 parts and 4 of them are moved to $\Pr(C_{\widetilde{p}} = 1)$, $\Pr(C_{\widetilde{p}} = 2)$, $\Pr(C_{\widetilde{p}} = 3)$, and $\Pr(C_{\widetilde{p}} = 5)$.

$C_{\widetilde{p}}$	0	1	2	3	4	5
$\Pr(C_{\widetilde{p}})$	$\frac{13193}{32561}$	$\frac{8305}{32561}$	$\frac{5068}{32561}$	$\frac{3446}{32561}$	$\frac{1568}{32561}$	$\frac{981}{32561}$
$C_{p'}$	0	1	2	3	4	5
$\Pr(C_{p'})$	$\frac{7193}{32561}$	$\frac{12305}{32561}$	$\frac{6068}{32561}$	$\frac{4446}{32561}$	$\frac{568}{32561}$	$\frac{1981}{32561}$

(a) Probability density functions of $\Pr(C_{\tilde{p}})$ and $\Pr(C_{p'})$



(b) Transportation plan $G_{\tilde{p},p}$ obtained when $\lambda_p = 500$

(c) Transportation plan $G_{\tilde{p},p}$ obtained when $\lambda_p = 50$

Figure 3: Visualization of mass transportation plans obtained by solving (1) using different λ_p values to move mass of $\Pr(C_{\widetilde{p}})$ to resemble $\Pr(C_{p'})$. (a) example discrete probability distributions. (b) if $\lambda_p = 500$, we achieve a heterogeneous plan, which tolerates more difference between $\Pr(C_{\widetilde{p}})$ and $\Pr(C_{p'})$ after the mass transportation. (c) if $\lambda_p = 50$, we achieve a homogeneous plan. which makes $\Pr(C_{\widetilde{p}})$ more closer to $\Pr(C_{p'})$ after the mass transportation.

6.1.3 Algorithm description. In the following, we formally describe the procedure of $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$. After Alice generates $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$ using the vanilla fingerprinting scheme, she evaluates the new joint distributions of all pairs of attributes, i.e., $\widetilde{J_{p,q}}, p, q \in \mathcal{F}, p \neq q$, and compares them with the reference joint distributions $\mathcal{J}'_{p,q}, p, q \in \mathcal{F}, p \neq q$. If the discrepancy between a particular pair of joint distributions exceeds a predetermined threshold, i.e., $||\widetilde{J_{p,q}} - J'_{p,q}||_F \geq \tau^{\mathrm{Dfs}}_{\mathrm{col}}$, Alice records both attributes p and q in a set Q. For all the attributes in Q, Alice obtains $\Pr(C_{\widetilde{p}})$ from $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset).p$ and calculates $\Pr(C_{p'}) = J'_{p,q}\mathbf{1}^T$. Next, she gets the optimal transportation plan for attribute p by solving (1). Then, she changes the instances of $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset).p$ to other instances by following the transportation moves suggested by $G_{\widetilde{p},p'}$, i.e., given $G_{\widetilde{p},p'}(a, b)$, Alice randomly samples $G_{\widetilde{p},p'}(a,b)$ fraction of entries (excluding the fingerprinted entries) whose attribute p takes value a and changes them to b. We summarize the procedure of $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$ in Algorithm 6, where lines 9-?? solves (1) to obtain the optimal mass transportation plan for attribute p, and lines 16-?? change the values of entries in $\widetilde{\mathbf{R}}(\mathrm{FP},\emptyset,\emptyset).p$ according to $G_{\widetilde{p},p'}$.

Algorithm 6: $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$: defense against column-wise correlation attack.

```
Input  : Vanilla fingerprinted database $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$, locations of
         entries changed by the vanilla fingerprinting scheme, and
         Alice's prior knowledge on the joint distributions of the
         pairwise attributes, i.e., $\mathcal{J}'$.
Output : $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\Big)$.
1  Initialize $Q = \emptyset$;
2  Obtain the empirical joint distributions set $\widetilde{\mathcal{J}}$ using $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$;
3  forall $p, q \in \mathcal{F}, p \neq q$ do
4      if $||J'_{p,q} - \widetilde{J}_{p,q}||_F > \tau_{\mathrm{col}}^{\mathrm{Dfs}}$ then
5          $Q = Q \cup p \cup q$;
6      end
7  end
8  forall $p \in Q$ do
9      Initialize the mass movement cost matrix $\Theta_{\widetilde{p},p'}$ and tuning
         parameter $\lambda_p$;
10     Obtain empirical marginal distribution $\Pr(C_{\widetilde{p}})$ from
         $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset).p$;
11     Initialize $G_{\widetilde{p},p'} = e^{-\lambda_p \Theta_{\widetilde{p},p'}}$;
12     while not converge do
13         Scale the rows of $G_{\widetilde{p},p'}$ to make the rows sum to the
             marginal distribution $\Pr(C_{\widetilde{p}})$;
14         Scale the columns of $G_{\widetilde{p},p'}$ to make the columns sum to
             the marginal distribution $\Pr(C_{p'})$;
15     end
16     forall $a \in [0, k_p - 1]$ do
17         forall $b \in [0, k_p - 1], b \neq a$ do
18             Sample $G_{\widetilde{p},p'}(a, b)$ percentage of entries from
                 $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset).p$ (excluding the vanilla fingerprinted
                 entries) whose attribute $p$ takes value $a$, and change
                 their value to $b$;
19         end
20     end
21 end
22 Return $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\Big)$.
```

6.1.4 Design details of $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$. We do not apply the optimal transportation technique to directly move the mass of the joint distributions obtained from $\widetilde{\mathbf{R}}(\mathrm{FP},\emptyset,\emptyset)$ to resemble the joint distributions in $\mathcal{J}'$. One reason is that, to do so, the database owner (Alice) needs to solve (1) for $\frac{|\mathcal{F}|(|\mathcal{F}|-1)}{2}$ joint distributions. This is computationally expensive if the database includes a large number of attributes. Thus, by considering the mass transportation in marginal distributions, the developed mitigation technique becomes more efficient. Furthermore, by only considering the marginal distributions, Alice can arrange $\widetilde{\mathbf{R}}\Big(\mathrm{FP},\emptyset,\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\Big)$ to have Pearson's correlations among attribute pairs that are close to those obtained from $\overline{\mathbf{R}}\Big(\emptyset,\emptyset,\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})\Big)$ if $\mathcal{J}'$ is close to $\mathcal{J}$. For instance, denote the Pearson's correlation between attributes p and q calculated from $\widetilde{\mathbf{R}}\Big(\mathrm{FP},\emptyset,\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\Big)$ and $\overline{\mathbf{R}}\Big(\emptyset,\emptyset,\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})\Big)$ as $\rho_{p',q'}$ and $\rho_{p,q}$, respectively. Then, we have $\rho_{p',q'} = \frac{\sum_{a,b}(a-\mu_{C_{p'}})(b-\mu_{C_{q'}})J'_{p,q}(a,b)}{\sigma_{C_{p'}}\sigma_{C_{q'}}}$, where $\mu_{C_{p'}}$ (or $\mu_{C_{q'}}$) and $\sigma_{C_{p'}}$ (or $\sigma_{C_{q'}}$) is the expected value and the standard deviation of attribute p (or q) obtained after applying the vanilla fingerprinting scheme followed by $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$, respectively. Also, $J'_{p,q}(a,b)$ is the database owner's prior knowledge on the joint probability distribution of attribute p taking value a and attribute q taking value b. Likewise, $\rho_{p,q} = \frac{\sum_{a,b}(a-\mu_{C_p})(b-\mu_{C_q})J_{p,q}(a,b)}{\sigma_{C_p}\sigma_{C_q}}$, where μ_{C_p} (or μ_{C_q}) and σ_{C_p} (or σ_{C_q}) is the expected value and the standard deviation of attribute p (or q) in $\overline{\mathbf{R}}\Big(\emptyset,\emptyset,\mathrm{Atk}_{\mathrm{col}}(\mathcal{J})\Big)$. Also, $J_{p,q}(a,b)$ is the malicious SP's prior knowledge on the joint probability distribution of attribute p taking value a and attribute q taking value b. If $\mathcal{J}'$ is close to $\mathcal{J}$, then $\mu_{C_{p'}}$ (or $\mu_{C_{q'}}$) is also close to μ_{C_p} (or μ_{C_q}), because of the marginalization of the similar joint distributions. Similar discussion also holds for $\sigma_{C_{p'}}$ (or $\sigma_{C_{q'}}$) and σ_{C_p} (or σ_{C_q}). As a result, $\rho_{p',q'}$ also becomes close to $\rho_{p,q}$, which improves the robustness of the fingerprint (against correlation attacks by a malicious SP), and hence prevents a malicious SP from distorting the potentially fingerprinted positions.

6.2 Robust Fingerprinting Against Row-wise Correlation Attack

To make a vanilla fingerprinting scheme also robust against row-wise correlation attack (in Section 5.2), we develop another mitigation technique, i.e., $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$. The main goal of $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$ is to avoid a malicious SP from distorting the fingerprint due to discrepancies in the expected statistical relationships between data records. Different from the design principle of $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$, which makes the newly obtained joint distributions resemble the prior knowledge, we design $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$ by changing selected entries of non-fingerprinted data records to make the newly obtained statistical relationships as far away from Alice's prior knowledge $\mathcal{S}'$ as possible. This is because the row-wise correlation attack usually changes limited number of entries in the vanilla fingerprinted database (as we validate in Section 7.2.1), thus, to make the newly obtained statistical relationships resemble $\mathcal{S}'$, one needs to change all non-fingerprinted data records and this will significantly compromise the database utility. Instead, by making the new statistical relationships far away from her prior knowledge, Alice can make additional (non-fingerprinted) data records that have cumulative absolute difference (with respect to the other records in the same community) exceeding a predetermined threshold. As a result, when launching $\mathrm{Atk}_{\mathrm{row}}(\mathcal{J})$, the malicious SP will identify wrong data records ($\boldsymbol{r}_i$), which causes $\sum_{j\neq i}^{n_c}\Big|s_{ij}^{\mathrm{comm}_c} - \widetilde{s_{ij}}^{\mathrm{comm}_c}\Big| \geq \tau_{\mathrm{row}}^{\mathrm{Atk}}$, and hence change the non-fingerprinted records.

In $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$, Alice selects a subset of non-fingerprinted data records in a community c, i.e., $\mathcal{E}_c \subset \mathrm{comm}_c$, $c \in [1,C]$, and changes their value to $\widetilde{\widetilde{\boldsymbol{r}}}_i, i \in \mathcal{E}_c$, such that the cumulative absolute difference between statistical relationships in her prior knowledge and those obtained from the fingerprinted database achieves the maximum difference after applying $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$. This can be formulated as the following optimization problem:

$$\begin{aligned} \max_{\mathcal{E}_c, \widetilde{\widetilde{\boldsymbol{r}}}_i} \quad & d(\mathcal{E}_c) = \Bigg| \sum_{j \in \mathrm{comm}_c/\mathcal{E}_c} \sum_{i \in \mathcal{E}_c} \left| {s'_{ij}}^{\mathrm{comm}_c} - {\widetilde{\widetilde{s_{ij}}}}^{\mathrm{comm}_c} \right| \\ & \qquad - \sum_{j \in \mathrm{comm}_c/\mathcal{E}_c} \sum_{i \in \mathcal{E}_c} \left| {s'_{ij}}^{\mathrm{comm}_c} - {\widetilde{s_{ij}}}^{\mathrm{comm}_c} \right| \Bigg| \\ \text{s.t.} \quad & \mathcal{E}_c \subset \mathrm{comm}_c/\mathcal{Q}_c, \\ & {\widetilde{\widetilde{s_{ij}}}}^{\mathrm{comm}_c} = e^{-\mathrm{dist}(\widetilde{\widetilde{\boldsymbol{r}}}_i, \boldsymbol{r}_j)}, i \in \mathcal{E}_c, j \in \mathrm{comm}_c/\mathcal{E}_c, \\ & \widetilde{\widetilde{\boldsymbol{r}}}_i = \text{value change}(\widetilde{\boldsymbol{r}}_i), i \in \mathcal{E}_c, \\ & |\mathcal{E}_c| \leq \lceil n_c \gamma \rceil, \end{aligned} \tag{2}$$

$\forall c \in [1, C]$. $\mathcal{Q}_c$ is the set of fingerprinted records in community c, ${s'_{ij}}^{\mathrm{comm}_c}$ denotes Alice's prior knowledge on the statistical relationship between individuals i and j in community c, ${\widetilde{s_{ij}}}^{\mathrm{comm}_c}$ is the statistical relationship between individuals i and j in community c in $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$, whose ith data record is denoted as $\widetilde{\boldsymbol{r}}_i$, and ${\widetilde{\widetilde{s_{ij}}}}^{\mathrm{comm}_c}$ is such information obtained from $\widetilde{\mathbf{R}}(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \emptyset)$, whose ith data record is represented as $\widetilde{\widetilde{\boldsymbol{r}}}_i$. Also, value change$(\cdot)$ is the function that changes each attribute of $\widetilde{\boldsymbol{r}}_i$, and it will be elaborated later. In (2), we let the cardinality of $\mathcal{E}_c$ to be smaller than $\lceil n_c \gamma \rceil$ (γ is the percentage of fingerprinted records) to restrict the number of selected non-fingerprinted records to maintain database utility.

(2) is an NP-hard combinatorial search problem [6]. Thus, we use a greedy algorithm to determine $\mathcal{E}_c$ and a heuristic approach to obtain $\widetilde{\widetilde{\boldsymbol{r}}}_i, i \in \mathcal{E}_c$. In fact, (2) also belongs to the problems of set function maximization, which can be connected to submodular optimization [31], and greedy algorithms are widely used for selecting candidate sets. Specifically, Alice constructs $\mathcal{E}_c$ by greedily choosing up to $\lceil n_c \gamma \rceil$ non-fingerprinted data records (in $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$) that have the maximum cumulative absolute difference (i.e., $\sum_{j \in \mathrm{comm}_c, j \neq i}^{n_c} \left| {s'_{ij}}^{\mathrm{comm}_c} - {\widetilde{s_{ij}}}^{\mathrm{comm}_c} \right|, i \in \mathrm{comm}_c/\mathcal{Q}_c$) with Alice's prior knowledge ($\mathcal{S}'$). Next, she changes the value of each attribute of the selected data records in $\mathcal{E}_c$ to the most frequent occurring instance of that attribute to obtain $\widetilde{\widetilde{\boldsymbol{r}}}_i$ (i.e., $\widetilde{\widetilde{\boldsymbol{r}}}_i$ = value change$(\widetilde{\boldsymbol{r}}_i)$). We describe the steps to apply $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$ in Algorithm 7.

The solution to (2) depends on the database and the distribution of data entries, thus, it is infeasible to derive a generic closed-form expression to quantify the mitigation performance of $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$. However, in Section 7.2.2, we will empirically show that the fraction of the fingerprinted entries inferred by $\mathrm{Atk}_{\mathrm{row}}(\mathcal{S})$ will decrease significantly if Alice applies the post-processing step $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$.

6.3 Integrated Robust Fingerprinting

Although after applying $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$, the malicious SP may still identify (and distort) some fingerprinted data records using $\mathrm{Atk}_{\mathrm{row}}(\mathcal{J})$, the amount of distortion in the fingerprint will not be enough to compromise the fingerprint bit-string due to the majority voting considered in the vanilla scheme. In Section 7.2.1, we validate that Algorithm 7 can successfully mitigate the row-wise correlation attack in a real-world database. Since $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$ changes less number of entries than $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$, database owner will apply $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$ first after the vanilla fingerprinting. In Algorithm 8, we summarize the main steps of our integrated robust fingerprinting scheme against the identified correlation attacks.

Algorithm 7: $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$: defense against row-wise correlation attack.

```
Input  : Vanilla fingerprinted database, $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$, fingerprinting
         ratio $\gamma$, database owner's prior knowledge on the
         row-wise correlations $\mathcal{S}'$ and individuals' affiliation to the
         $C$ communities.
Output : $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \emptyset, \Big)$.
Obtain $\widetilde{\mathcal{S}}$, i.e., the set of pairwise statistical relationships among
  individuals in each community, from the vanilla fingerprinted
  database $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$;
forall $\mathrm{comm}_c$, $c \in [1, C]$ do
    forall non-fingerprinted individual $i \in \mathrm{comm}_c/\mathcal{Q}_c$ do
        Calculate $e_i = \sum_{j \in \mathrm{comm}_c, j \neq i}^{n_c} \left| {s'_{ij}}^{\mathrm{comm}_c} - {\widetilde{s_{ij}}}^{\mathrm{comm}_c} \right|$, $i \in$
          $\mathrm{comm}_c/\mathcal{Q}_c$;
    end
    Obtain the largest $\lceil n_c \gamma \rceil$ $e_i$'s, and collect these row index $i$ in
      set $\mathcal{E}_c$;
    forall row index $i \in \mathcal{E}_c$ do
        $\widetilde{\widetilde{\boldsymbol{r}}}_i$ = value change($\widetilde{\boldsymbol{r}}_i$); //change the value of each
          attribute of $\widetilde{\boldsymbol{r}}_i$ to the most frequently
          occurred instance of that attribute in $\mathrm{comm}_c$.
    end
end
Return $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \emptyset, \Big)$.
```

Algorithm 8: Robust fingerprinting against correlation attacks.

```
Input  : A database $\mathbf{R}$, a vanilla fingerprinting scheme FP, database
         owner's prior knowledge on the column-wise and
         row-wise correlation, i.e., $\mathcal{J}'$ and $\mathcal{S}'$, and individuals'
         affiliation to the $C$ communities.
Output : $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\Big)$.
Apply the vanilla fingerprinting scheme on $\mathbf{R}$ and obtain
  $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \emptyset, \emptyset\Big)$;
Apply $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}')$ on $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \emptyset, \emptyset\Big)$ using Algorithm 7 and obtain
  $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \emptyset\Big)$;
Apply $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')$ on $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \emptyset\Big)$ using Algorithm 6 and
  obtain $\widetilde{\mathbf{R}}\Big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}')\Big)$;
```

7 EVALUATION

Now, we evaluate the correlation attacks and the robust fingerprinting mechanisms, investigate their impact on fingerprint robustness and database utility, and empirically study the effect of knowledge asymmetry between the database owner and a malicious SP.

7.1 Experiment Setup

We consider a Census database [3] as the study case. As discussed in Section 3, we choose the state-of-the-art scheme developed in

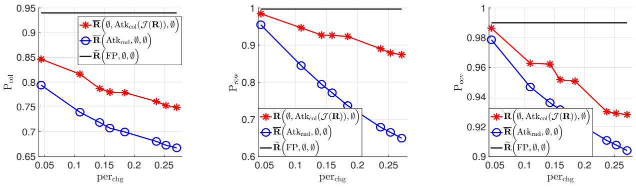
[24] as the vanilla mechanism, because it is shown to be robust against common attacks (such as random bit flipping, subset, and superset attacks). We use 128-bits fingerprint string ($L = 128$) for the vanilla scheme, because when considering N SPs, as long as $L > \ln N$, the vanilla scheme can thwart exhaustive search and various types of attacks [24], and in most cases a 64-bits fingerprint string is shown to provide high robustness.

In different experiments, to distinguish different instances of the row-wise and column-wise correlations, we also parametrize $\mathcal{J}'$, $\mathcal{S}'$, $\mathcal{J}$, and $\mathcal{S}$ when specifying their resources. For instance, $\mathcal{J}'(\mathbf{R})$ indicates Alice's prior knowledge on column-wise correlations are calculated directly from the original database.

7.2 Evaluations on Census Database

Census database [3] records 14 discrete or categorical attributes of 32561 individuals. To add fingerprint to this database, Alice first encodes the values of each attribute as integers in a way that the LSB carries the least information. Recall that to achieve high database utility, we let the vanilla scheme only fingerprint the LSBs (in Appendix A we validate that fingerprinting the other bits reduces database utility). In particular, for a discrete numerical attribute (e.g., age), the values are first sorted in an ascending order and then divided into non-overlapping ranges, which are then encoded as ascending integers starting from 0. For a categorical attribute (e.g., marital-status), the instances are first mapped to a high dimensional space via the word embedding technique [26]. Words having similar meanings appear roughly in the same area of the space. After mapping, these vectors are clustered into a hierarchical tree structure, where each leaf node represents an instance of that attribute and is encoded by an integer and the adjacent leaf nodes differ in the LSB. Besides, we use K-means to group the individuals in the Census database into non-overlapping communities, and according to the Schwarz's Bayesian inference criterion (BIC) [28], the optimal number of communities is $C = 10$.

7.2.1 Impact of Correlation Attacks on Census Database. We first study the impact of $\text{Atk}_{\text{row}}(\mathcal{S})$ and $\text{Atk}_{\text{col}}(\mathcal{J})$, and then present the impact of the integration of them. In this experiment, we assume that the malicious SP has the ground truth knowledge about the row- and column-wise correlations, i.e., it has access to $\mathcal{S}$ and $\mathcal{J}$ that are directly computed from $\mathbf{R}$. As a result, we represent its prior knowledge as $\mathcal{S}(\mathbf{R})$ and $\mathcal{J}(\mathbf{R})$. By launching the row-wise, column-wise, and integrated correlation attack, the malicious SP generates pirated database $\overline{\mathbf{R}}\Big(\text{FP}, \text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R})), \emptyset\Big)$, $\overline{\mathbf{R}}\Big(\text{FP}, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))\Big)$, and $\overline{\mathbf{R}}\Big(\text{FP}, \text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R})), \text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))\Big)$, respectively.



(a) P_{col}

(b) P_{row}

(c) P_{cov}

Figure 4: Comparison of (a) P_{col}, (b) P_{row}, and (c) P_{cov} achieved by $\overline{\mathbf{R}}(\emptyset, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R})))$ and $\overline{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$ when per_{chg} are set as the values highlighted in gray in Table 2.

Impact of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$. First, we validate that $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ is more powerful than the random bit flipping attack Atk_{rnd} discussed in Section 4. We set the threshold $\tau_{\text{col}}^{\text{Atk}} = 0.0001$ when comparing $|J_{p,q}(a,b) - \widetilde{J_{p,q}}(a,b)|$.[1] As a result, it takes 8 iterations (attack rounds) for $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ to converge (i.e., stop including new suspicious fingerprinted positions in $\mathcal{P}$). In Table 2, we record the fingerprint robustness (i.e., num_{cmp} and r) and utility loss of the malicious SP (fraction of modified entries as a result of the attack, i.e., $\text{per}_{\text{chg}} = 1 - Acc(\overline{\mathbf{R}})$) when launching increasing rounds of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ on the vanilla fingerprinted Census database. We observe that with more attack rounds, more fingerprint bits are compromised, and the accusable ranking of the malicious SP also decreases, which suggests that Alice may accuse innocent SP with increasing probability. In Table 3, we present the performance of Atk_{rnd} on the vanilla fingerprinted database. Specifically, by setting the fraction of entries changed (per_{chg}) due to Atk_{rnd} equal to that of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ with increasing rounds (i.e., the cells highlighted in gray in Table 2), we calculated num_{cmp} and r achieved by Atk_{rnd}.

Combining Tables 2 and 3, we observe that if per_{chg} is below 14.2%, Atk_{rnd} cannot compromise any fingerprint bits, whereas $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ compromises 28 fingerprint bits (out of 128). Even when $\text{per}_{\text{chg}} = 27.1\%$, Atk_{rnd} can only distort 4 fingerprint bits. As a result, if the malicious SP launches Atk_{rnd}, it will be uniquely accusable for pirating the database. Whereas, when $\text{per}_{\text{chg}} = 27.1\%$ $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ distorts 82 bits, which makes the malicious SP only rank top 91.4% accusable and will cause Alice accuse innocent SP with very high probability (the cells highlighted in green in Table 2). In fact, for Atk_{rnd} to compromise enough fingerprint bits so as to cause Alice to accuse innocent SPs, it needs to flip more than 83% of the entries in the fingerprinted Census database. Clearly, the vanilla fingerprint scheme is robust against Atk_{rnd}, however, its robustness significantly degrades against $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$.

In Figure 4, by setting per_{chg} to the values highlighted in gray in Table 2, we compute and compare the utility of the pirated database (i.e, $\text{P}_{\text{col}}(\overline{\mathbf{R}})$ for $\tau_{\text{col}} = 0.0001$, $\text{P}_{\text{row}}(\overline{\mathbf{R}})$ for $\tau_{\text{row}} = 10$, and $\text{P}_{\text{cov}}(\overline{\mathbf{R}})$) obtained from the vanilla fingerprinted database after Atk_{rnd} and $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$, i.e., $\overline{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$ and $\overline{\mathbf{R}}(\emptyset, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R})))$. We also plot the utility of the vanilla fingerprinted database using black lines as the benchmark ($\text{P}_{\text{col}}(\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)) = 0.95$, $\text{P}_{\text{row}}(\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)) = 1.00$, and $\text{P}_{\text{cov}}(\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)) = 0.99$). Clearly, $\overline{\mathbf{R}}(\emptyset, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R})))$ always achieves higher utility values than $\overline{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$, and it has similar utility values compared to $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ when per_{chg} is small, e.g., if $\text{per}_{\text{chg}} \leq 20\%$, $\text{P}_{\text{col}}\big(\overline{\mathbf{R}}(\emptyset, \emptyset, \text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R})))\big) \geq 0.92$. Combining Table 2, 3, and Figure 4, we conclude that $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ is not only more powerful (in terms of distorting the fingerprint bit-string), but it also preserves more database utility compared to

[1] In all experiments, we choose a small value for $\tau_{\text{col}}^{\text{Atk}}$, $\tau_{\text{col}}^{\text{Dfs}}$, and τ_{col}, because a database usually contains thousands of data records and the addition of fingerprint changes a small fraction of entries, which does not cause large changes in the joint distributions. On the contrary, we choose a large value for $\tau_{\text{row}}^{\text{Atk}}$ and τ_{row}, because the statistical relationship is defined as an exponentially decay function, which ranges from 0 to 1, and the added fingerprint results in a larger change for this statistical relationship.

Attack on	robustness &	using	rounds of $\text{Atk}_{\text{col}}\big(\mathcal{J}(\mathbf{R})\big)$							
$\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$	utility loss	$\text{Atk}_{\text{row}}\big(\mathcal{S}(\mathbf{R})\big)$	1	2	3	4	5	6	7	8
using	num_{cmp}		28	43	55	58	63	74	77	82
$\text{Atk}_{\text{col}}\big(\mathcal{J}(\mathbf{R})\big)$	r	N/A	u	u	u	< 0.08%	< 0.73%	< 53.2%	< 71.8%	< 91.4%
	per_{chg}		4.4%	10.9%	14.2%	15.9%	18.4%	23.7%	25.3%	27.1%
using $\text{Atk}_{\text{row}}\big(\mathcal{S}(\mathbf{R})\big)$	num_{cmp}	78	78	79	80	81	82	83	83	83
and $\text{Atk}_{\text{col}}\big(\mathcal{J}(\mathbf{R})\big)$	r	< 82.9%	< 82.9%	< 89.1%	< 89.4%	< 90.1%	< 91.4%	< 93.7%	< 93.7%	< 93.7%
	per_{chg}	2.9%	8.9%	10.5%	11.3%	11.5%	11.9%	12.6%	13.7%	14.2%

Table 2: Fingerprint robustness and utility loss of different correlation attacks on the vanilla fingerprinted Census database $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$. The fingerprint robustness metrics are the number of compromised fingerprint bits, i.e., num_{cmp} and i.e., accusable ranking r. The utility loss of the malicious SP is the fraction of modified entries as a result of the attack, i.e., $\text{per}_{\text{chg}} = 1 - Acc(\widetilde{\mathbf{R}})$. 'u' stands for uniquely accusable. '< r' means top r accusable.

per_{chg}	≤ 14.2%	15.9%	18.4%	23.7%	25.3%	27.1%
num_{cmp}	0	1	1	2	3	4
r	u	u	u	u	u	u

Table 3: Performance and cost of Atk_{rnd} on the vanilla fingerprinted Census database. per_{chg} values are set to be equal to that of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ (cells highlighted in gray in Table 2). r = u means uniquely accusable.

Atk_{rnd}. In addition to the generic utility metrics defined in Section 4.4, we also calculate and compare the utility some specific statistical computations on the pirated database. For example, under the same attack performance (i.e., compromising exactly 63 fingerprinting bits) $\text{Atk}_{\text{col}}(\mathcal{J})$ only causes 0.3% change in the frequency of individuals having bachelor degree or higher and 0.01 change for the standard deviation of individuals' age, whereas, the same values for Atk_{rnd} are 1.4% and 0.12, respectively.

Impact of $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}))$. By setting the threshold $\tau_{\text{row}}^{\text{Atk}} = 0.1$ when comparing $\sum_{j \neq i}^{n_c} |s_{ij}^{\text{comm}_c} - \widetilde{s_{ij}}^{\text{comm}_c}|$ in Section 5.2 we show the impact of $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}))$ in the blue cells of Table 2. After launching row-wise correlation attack on $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, 78 fingerprint bits are distorted at the cost of only 2.9% utility loss. It makes the malicious SP only rank top 82.9% accusable, and may cause Alice accuse innocent SP with high probability. In particular, we have $\text{P}_{\text{col}}\big(\overline{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}}(\mathcal{S}(\mathbf{R})), \emptyset)\big) = 0.90$, $\text{P}_{\text{row}}\big(\overline{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}}(\mathcal{S}(\mathbf{R})), \emptyset)\big) = 0.95$, and $\text{P}_{\text{cov}}\big(\overline{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}}(\mathcal{S}(\mathbf{R})), \emptyset)\big) = 0.97$, which are all closer to that of $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$. This suggests again that the identified correlation attacks are powerful than the conventional attacks and they can maintain the utility of database.

Impact of integrated correlation attack. By launching $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}))$ on $\widetilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ followed by 8 rounds of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$, the integrated correlation attack can distort more fingerprint bits, i.e., 83 bits, which makes the malicious SP's accusable ranking drops to top 93.7% (the cells highlighted in red in Table 2). This suggests again that the vanilla fingerprint scheme is not capable of identifying the guilty SP that is liable for pirating the database if the malicious SP utilizes data correlations to distort the fingerprint.

Note that, although $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ has similar attack performance compared to the integrated attack, its utility loss is higher, i.e., 27.1% entries are modified by the attacker. Besides, at the early stages of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$, the malicious SP cannot distort more than half of the fingerprint bits (e.g., at the end of the 5th round, only 63 bits are compromised by modifying 15.9% of the entries), which is inadequate to cause Alice accuse innocent SPs and also makes the malicious SP uniquely accusable. Since $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}))$ can distort sufficient fingerprint bits and cause Alice to accuse innocent SPs with high probability at a much lower utility loss (measured using both generic utility metrics and specific statistical utilities, like the change in frequencies of data records and standard deviations), we conclude that it is more powerful than $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$. This suggests that in real-world integrated correlation attacks, the malicious SP can conduct $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}))$ followed by a few rounds of $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}))$ to simultaneously distort a large number of fingerprint bits and preserve data utility when generating the pirated database.

7.2.2 *Performance of Mitigation Techniques on Census Database.*

We have shown that correlation attacks can distort the fingerprint bit-string and may make the database owner accuse innocent SPs by resulting in low degradation in terms of database utility. In this section, we first evaluate the proposed mitigation techniques against correlation attacks separately, and then consider the integrated mitigation technique against the integrated correlation attack, i.e., the row-wise correlation attack followed by the column-wise correlation attack. In this experiment, we also assume that Alice has access to $\mathcal{S}'$ and $\mathcal{J}'$that are directly computed from $\mathbf{R}$. Thus, we represent her prior knowledge as $\mathcal{S}'(\mathbf{R})$ and $\mathcal{J}'(\mathbf{R})$. As a result, we have $\mathcal{S}' = \mathcal{S}$ and $\mathcal{J}' = \mathcal{J}$.

Performance of $\text{Dfs}_{\text{col}}(\mathcal{J}'(\mathbf{R}))$. As discussed in Section 6.1, the mitigation strategy is determined by the marginal probability mass transportation plan, which is heterogeneous for higher λ_p (a tuning parameter controlling the entropy of the transportation plan) and homogeneous for lower λ_p. To evaluate the utility loss due to $\text{Dfs}_{\text{col}}(\mathcal{J}'(\mathbf{R}))$, we calculate the utility of $\widetilde{\mathbf{R}}\big(\text{FP}, \emptyset, \text{Dfs}_{\text{col}}(\mathcal{J}'(\mathbf{R}))\big)$ by setting $\lambda_p \in \{100, \cdots, 1000\}, \forall p \in \mathcal{F}$, and show the results in Figure 5. We see that all utilities monotonically increase as the mass transportation plans transform from homogeneous to heterogeneous (i.e., as λ_p increases). This is because, as the transportation plans become more heterogeneous, the mitigation technique can tolerate more discrepancy between two marginal distributions (Section 6.1), and hence fewer number of entries are modified by $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathbf{R}))$.

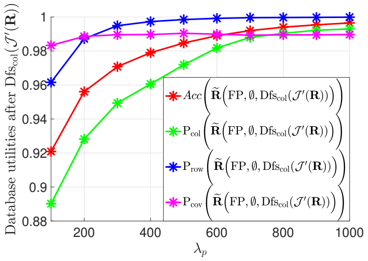
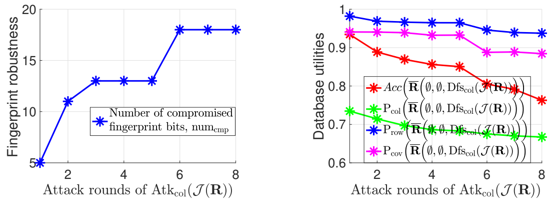


Figure 5: Utilities of $\widetilde{\mathrm{R}}(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathrm{R})))$ under varying λ_p.

Next, we fix $\lambda_p = 500, \forall p \in \mathcal{F}$, evaluate the performance (in terms of both fingerprint robustness and database utility) of launching $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathbf{R}))$ on $\widetilde{\mathbf{R}}\big(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathbf{R}))\big)$ with increasing attack rounds. In Figure 6(a), we observe that at then end of 8 rounds of $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathbf{R}))$, the malicious SP can only compromise 24 (out of 128) fingerprint bits, which is not enough to cause Alice accuse innocent SPs and will make itself uniquely accusable. In contrast, as shown in Table 2, when launching $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathbf{R}))$ on the vanilla fingerprinted database $\widetilde{\mathbf{R}}(\mathrm{FP}, \emptyset, \emptyset)$, the malicious SP can compromise 82 bits and make itself only rank top 91.4% accusable. This suggests that proposed $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathbf{R}))$ significantly mitigates the column-wise correlation attack.



(a) Fingerprinting robustness for increasing attack rounds.

(b) Database utilities for increasing attack rounds.

Figure 6: Fingerprint robustness and database utilities when launching $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathrm{R})))$ on $\widetilde{\mathrm{R}}(\mathrm{FP}, \emptyset, \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathrm{R})))$.

Furthermore, in Figure 6(b) we observe that $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathbf{R}))$ also degrades the utilities of the vanilla fingerprinted database post-processed by $\mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathbf{R}))$. In particular, the accuracy drops to 0.76 and the preservation of column-wise correlation drops to 0.67 at the end of 8 rounds of $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathbf{R}))$. Combining Figures 5 and 6, we conclude that, as a post-processing step, the proposed column-wise correlation mitigation technique provides robust fingerprint against column-wise correlation attack and preserves database utility.

Performance of $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'(\mathbf{R}))$. In Table 4, we evaluate the performance of the robust fingerprinted database against row-wise attack, i.e., $\widetilde{\mathbf{R}}(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'(\mathbf{R}), \emptyset)$, along with the pirated database obtained by launching $\mathrm{Atk}_{\mathrm{row}}(\mathcal{S}(\mathbf{R})$ on it. Clearly, $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'(\mathbf{R}))$ successfully defends against $\mathrm{Atk}_{\mathrm{row}}(\mathcal{S}(\mathbf{R}))$, since the pirated database only distorts 13 fingerprint bits and makes the malicious SP uniquely accusable. Combining this result with Table 2 (cells in blue), we conclude that $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'(\mathbf{R}))$ not only mitigates the row-wise correlation attack but it also preserves the database utility.

	Acc	$\mathrm{P}_{\mathrm{col}}$	$\mathrm{P}_{\mathrm{row}}$	$\mathrm{P}_{\mathrm{cov}}$	$\mathrm{num}_{\mathrm{cmp}}$	r
$\widetilde{\mathbf{R}}(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'), \emptyset)$	0.97	0.94	0.99	0.99	N/A	N/A
$\widetilde{\mathbf{R}}(\emptyset, \mathrm{Atk}_{\mathrm{row}}(\mathcal{S}), \emptyset)$	0.93	0.92	0.94	0.98	13	u

Table 4: Impact of $\mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'(\mathrm{R}))$ before and after $\mathrm{Atk}_{\mathrm{row}}(\mathcal{S}(\mathrm{R}))$. r = u means uniquely accusable.

Performance of integrated mitigation. Here, we investigate the performance of the integrated mitigation against the integrated correlation attacks. By setting $\lambda_p = 500, \forall p \in \mathcal{F}$, we evaluate the utility of $\widetilde{\mathbf{R}}\big(\mathrm{FP}, \mathrm{Dfs}_{\mathrm{row}}(\mathcal{S}'(\mathbf{R})), \mathrm{Dfs}_{\mathrm{col}}(\mathcal{J}'(\mathbf{R}))\big)$ before and after it is subject to the integrated attack, i.e., $\mathrm{Atk}_{\mathrm{row}}(\mathcal{S}(\mathbf{R}))$ followed by $\mathrm{Atk}_{\mathrm{col}}(\mathcal{J}(\mathbf{R}))$. We show the results in Table 5. Clearly, after integrated mitigation, the fingerprinted database still maintains high utilities. Even if the malicious SP launches integrated correlation attack, it can only compromise 4 fingerprint bits and makes itself uniquely accusable. It suggests that the proposed mitigation techniques provide high robustness against integrated correlated attacks.

	Acc	$\mathrm{P}_{\mathrm{col}}$	$\mathrm{P}_{\mathrm{row}}$	$\mathrm{P}_{\mathrm{cov}}$	$\mathrm{num}_{\mathrm{cmp}}$	r
after int. mitigation	0.94	0.91	0.96	0.97	N/A	N/A
after int. attack	0.77	0.82	0.86	0.94	4	u

Table 5: Impact of integrated mitigation before and after integrated correlation attack. r = u means uniquely accusable.

8 CONCLUSION

In this paper, we have proposed robust fingerprinting for relational databases. First, we have validated the vulnerability of existing database fingerprinting schemes by identifying different correlation attacks: column-wise correlation attack (which utilizes the joint distributions among attributes), row-wise correlation attack (which utilizes the statistical relationships among the rows), and integration of them. Next, to defend against the identified attacks, we have developed mitigation techniques that can work as post-processing steps for any off-the-shelf database fingerprinting schemes. Specifically, the column-wise mitigation technique modifies limited entries in the fingerprinted database by solving a set of optimal mass transportation problems concerning pairs of marginal distributions. On the other hand, the row-wise mitigation technique modifies a small fraction of the fingerprinted database entries by solving a combinatorial search problem. We have also empirically investigated the impact of the identified correlation attacks and the performance of proposed mitigation techniques on an real-world relational database. Experimental results show high success rates for the correlation attacks and high robustness for the proposed mitigation techniques, which alleviate the attacks having access to correlation models directly calculated from the data.

ACKNOWLEDGMENTS

Research reported in this publication was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R01LM013429.

REFERENCES

[1] [n.d.]. Wattpad data breach exposes account info for millions of users. https://www.bleepingcomputer.com/news/security/wattpad-data-breach-exposes-account-info-for-millions-of-users/. (Accessed on 01/20/2021).

[2] Rakesh Agrawal, Peter J Haas, and Jerry Kiernan. 2003. Watermarking relational data: framework, algorithms and analysis. *The VLDB journal* 12, 2 (2003), 157–169.

[3] Arthur Asuncion and David Newman. 2007. UCI machine learning repository.

[4] Erman Ayday, Emre Yilmaz, and Arif Yilmaz. 2019. Robust Optimization-Based Watermarking Scheme for Sequential Data. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*. 323–336.

[5] Paraskevi Bassia, Ioannis Pitas, and Nikos Nikolaidis. 2001. Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia* 3, 2 (2001), 232–241.

[6] Dimitris Bertsimas and John N Tsitsiklis. 1997. *Introduction to linear optimization*. Vol. 6. Athena Scientific Belmont, MA.

[7] Dan Boneh and James Shaw. 1995. Collusion-secure fingerprinting for digital data. In *Annual International Cryptology Conference*. Springer, 452–465.

[8] Dan Boneh and James Shaw. 1998. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory* 44, 5 (1998), 1897–1905.

[9] J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O'Gorman. 1995. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications* 13, 8 (1995), 1495–1504.

[10] J. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman. 1994. Hiding Information in Document Images. *Proceedings of Conference on Information Sciences and Systems* (1994).

[11] Michael W Browne and Robert Cudeck. 1992. Alternative ways of assessing model fit. *Sociological methods & research* 21, 2 (1992), 230–258.

[12] Edgar F Codd. 2002. A relational model of data for large shared data banks. In *Software pioneers*. Springer, 263–294.

[13] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2016. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence* 39, 9 (2016), 1853–1865.

[14] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. 1997. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing* 6, 12 (1997), 1673–1687.

[15] Ingemar J Cox, Matthew L Miller, Jeffrey Adam Bloom, and Chris Honsinger. 2002. *Digital watermarking*. Vol. 53. Springer.

[16] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*. 2292–2300.

[17] Sudhanshu S Gonge and Jagdish W Bakal. 2013. Robust Digital Watermarking Techniques by Using DCT and Spread Spectrum. *International Journal of Electrical, Electronics and Data Communication* 1, 2 (2013), 111–124.

[18] Fei Guo, Jianmin Wang, and Deyi Li. 2006. Fingerprinting relational databases. In *Proceedings of the 2006 ACM symposium on Applied computing*. 487–492.

[19] Neil F Johnson, Zoran Duric, and Sushil Jajodia. 2001. *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures: Steganography and Watermarking: Attacks and Countermeasures*. Vol. 1. Springer Science & Business Media.

[20] Ian T Jolliffe. 2002. Springer series in statistics. *Principal component analysis* 29 (2002).

[21] Darko Kirovski and Henrique Malvar. 2001. Robust spread-spectrum audio watermarking. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 1345–1348.

[22] Julien Lafaye, David Gross-Amblard, Camelia Constantin, and Meryem Guerrouani. 2008. Watermill: An optimized fingerprinting system for databases under constraints. *IEEE Transactions on Knowledge and Data Engineering* 20, 4 (2008), 532–546.

[23] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. 2003. Constructing a virtual primary key for fingerprinting relational data. In *Proceedings of the 3rd ACM workshop on Digital rights management*. 133–141.

[24] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. 2005. Fingerprinting relational databases: Schemes and specialties. *IEEE Transactions on Dependable and Secure Computing* 2, 1 (2005), 34–45.

[25] Siyuan Liu, Shuhong Wang, Robert H Deng, and Weizhong Shao. 2004. A block oriented fingerprinting scheme in relational database. In *International conference on information security and cryptology*. Springer, 455–466.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013), 3111–3119.

[27] Birgit Pfitzmann and Michael Waidner. 1997. Asymmetric fingerprinting for larger collusions. In *Proceedings of the 4th ACM conference on Computer and communications security*. 151–160.

[28] Gideon Schwarz et al. 1978. Estimating the dimension of a model. *The annals of statistics* 6, 2 (1978), 461–464.

[29] Mitchell D Swanson, Bin Zhu, and Ahmed H Tewfik. 1998. Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on selected areas in Communications* 16, 4 (1998), 540–550.

[30] Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. 1994. A digital watermark. In *Proceedings of IEEE International Conference on Image Processing*, Vol. 2. IEEE, 86–90.

[31] Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*. PMLR, 1954–1963.

[32] Yacov Yacobi. 2001. Improved boneh-shaw content fingerprinting. In *Cryptographers' Track at the RSA Conference*. Springer, 378–391.

[33] Emre Yilmaz and Erman Ayday. 2020. Collusion-Resilient Probabilistic Fingerprinting Scheme for Correlated Data. *arXiv preprint arXiv:2001.09555* (2020).

A TRADEOFF BETWEEN FINGERPRINT ROBUSTNESS AND DATABASE UTILITY

As discussed in Section 3, to preserve database utility, the added fingerprint only changes the LSB of database entries. In this experiment, we show that if the fingerprint bits are embedded into other bits of entries, some utility metrics will decrease. Specifically, by fixing the fingerprinting ratio to 1/30, we evaluate the utility (e.g., preservation of correlations and statistics metrics) of the fingerprinted Census database obtained by using the vanilla fingerprinting scheme and changing one of the least k ($k \geq 2$) significant bits (i.e., LkSB) of database entries (to add the fingerprint). We show the results in Table 6.

Utilities	LSB	L2SB	L3SB	L4SB
$Acc\left(\widetilde{\mathrm{R}}(\mathrm{FP}, \emptyset, \emptyset)\right)$	0.98	0.98	0.98	0.98
$\mathrm{P}_{\mathrm{col}}\left(\widetilde{\mathrm{R}}(\mathrm{FP}, \emptyset, \emptyset)\right)$	0.95	0.90	0.88	0.86
$\mathrm{P}_{\mathrm{row}}\left(\widetilde{\mathrm{R}}(\mathrm{FP}, \emptyset, \emptyset)\right)$	1.00	0.98	0.98	0.98
$\mathrm{P}_{\mathrm{cov}}\left(\widetilde{\mathrm{R}}(\mathrm{FP}, \emptyset, \emptyset)\right)$	0.99	0.96	0.95	0.94

Table 6: Different database utility values obtained when the insertion of fingerprint changes one of the least k significant bits of database entries.

We observe that although all fingerprinted databases achieve the same accuracy when the fingerprinting ratio is set to be 1/30, other utilities decrease if the added fingerprint changes LkSB ($k \geq 2$) of data entries. Especially, the preservation of column-wise correlation degrades the most as k increases. The reason is that some pairs of attributes are highly correlated and changing one of the LkSB may create statistical unlikely pairs. For example, Masters education degree corresponds to education of 14 years, if the L4SB of 14 ("1110") is flipped, we end up with an individual who has a master degree with only 6 ("0110") years of education, which compromise the correlation between "education" and "education-num".