

Towards Robust Fingerprinting of Relational Databases by Mitigating Correlation Attacks

Tianxi Ji, Erman Ayday, Emre Yilmaz, and Pan Li

Abstract—Database fingerprinting is widely adopted to prevent unauthorized data sharing and identify the source of data leakages. Although existing schemes are robust against common attacks, their robustness degrades significantly if attackers utilize inherent correlations among database entries. In this paper, we demonstrate the vulnerability of existing schemes by identifying different correlation attacks: column-wise correlation attack, row-wise correlation attack, and their integration. We provide robust fingerprinting against these attacks by developing mitigation techniques, which can work as post-processing steps for any off-the-shelf database fingerprinting schemes and preserve the utility of databases. We investigate the impact of correlation attacks and the performance of mitigation techniques using a real-world database. Our results show (i) high success rates of correlation attacks against existing fingerprinting schemes (e.g., integrated correlation attack can distort 64.8% fingerprint bits by just modifying 14.2% entries in a fingerprinted database), and (ii) high robustness of mitigation techniques (e.g., after mitigation, integrated correlation attack can only distort 3% fingerprint bits). Additionally, the mitigation techniques effectively alleviate correlation attacks even if (i) attackers have access to correlation models directly computed from the original database, while the database owner uses inaccurate correlation models, (ii) or attackers utilizes higher order of correlations than the database owner.

Index Terms—Robust fingerprinting; relational databases; correlation attacks; privacy; data sharing.

1 INTRODUCTION

RELATIONAL databases (or relations), defined as a set of data records with the same attributes [2], have become the most widespread database systems. Sharing the full relations is beneficial to many tasks where statistics or learned models are insufficient. For instance, a relational database owner (who collects data from individuals and constructs the dataset) can benefit from outsourced computation by uploading/sharing database to service providers (SP) like Amazon Elastic Compute Cloud, let other SPs analyze its data (e.g., for personal advertisements), or exchange data 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' (database participants) privacy is violated. 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.

Digital fingerprinting is a steganography technology that allows to identify the source of data breaches by embedding a unique mark into each shared copy of a digital object. Un-

like digital watermarking, in fingerprinting, the embedded mark must be unique to distinguish all database recipients. Although the most prominent usage of fingerprinting is in the multimedia domain [3], [4], [5], fingerprinting techniques for databases have also been developed [6], [7], [8], [9], [10]. These techniques change different database entries when sharing a database copy with different SPs (see Figure 2 for a typical database fingerprinting system).

However, existing fingerprinting schemes for databases are developed to embed fingerprints in continuous-valued numerical entries (floating points). In these schemes, the database owner will modify the least significant bit (LSB) of the last digit of the selected floating numbers. However, 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 database utility.

Although, existing fingerprinting schemes are robust against common attacks, such as random bit flipping attack, subset/superset attack (discussed in Section 4.2), they do not consider various intrinsic correlations between the data entries in a database. For instance, in demographic databases, zip codes are correlated with street names. Hence, 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.

We refer to the attack that utilize the correlations between attributes and data records to infer the potentially fingerprinted entries as "correlation attacks". In our previous work [1], we have identified three correlation attacks: column-wise correlation attack, row-wise correlation attack, and the integration of them. To launch these attacks, a malicious SP utilizes its prior knowledge about correlations

- T. Ji is with Department of Computer Science, Texas Tech University, TX 79409. Email: tiji@ttu.edu.
- E. Ayday is with Department of Computer and Data Sciences, Case Western Reserve University, Cleveland, OH 44106. Email: exa208@case.edu
- E. Yilmaz is with Department of Computer Science and Engineering Technology, University of Houston-Downtown, Houston, TX 77002. Email: yilmaze@uhd.edu.
- P. Li is with Department of Electrical, Computer, and System Engineering, Case Western Reserve University, Cleveland, OH 44106. Email: lipan@case.edu.

This paper is the extended version of the conference paper [1] (<https://dl.acm.org/doi/10.1145/3471621.3471853>).

between the columns (attributes) of a database, statistical relationships between the rows (data records), and the combination of both. We show that 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 observe that by just changing 14.2% entries in a real-world relational 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 the identified correlation attacks are more powerful than traditional attacks, because they can distort more fingerprint bits with less utility loss.

To mitigate identified correlation attacks, in our previous work [1] we also proposed corresponding mitigation techniques and developed robust fingerprinting schemes for relational databases with discrete (or categorical) values. Our proposed mitigation technique can serve as a post-processing step for any vanilla¹ database fingerprinting schemes to improve their robustness against correlation attacks. In a nutshell, the mitigation techniques only introduce additional bit changes for unfingerprinted data entries in a way that the post-processed fingerprinted databases (i) present similar joint probability distributions among pairs of columns with that provided by the database owners' prior knowledge (to prevent the malicious SP from taking advantage of the discrepancy between column-wise correlations before and after fingerprinting) and (ii) make pairwise statistical similarities between rows distant from those produced by the database owner's prior knowledge (to mislead the malicious SP into changing the wrong entries without degrading the utility of the shared database). As a result, the malicious SP cannot identify and distort the fingerprinted entries without significant utility loss even if it launches the more powerful correlation attacks. Thus, it will be held as responsible for data leakage. The detailed description of mitigation techniques are provided in Section 5. The proposed mitigation techniques also maintain the utility of the shared databases by (i) encoding the database entries as integers, such that the LSB carries the least information, and adding the fingerprint by only changing the LSBs; and (ii) changing only a small number of database entries.

In this paper, we extend our previous work [1] in different aspects. In particular,

- We empirically validate the attack strength (measured in terms of confidence gain) of the identified correlation attacks, and show that such attack strength is largely constrained by our proposed mitigation techniques.
- To show the widespread vulnerability of existing fingerprinting schemes against correlation attacks and the effective remedy of our mitigation techniques as post-processing steps, we consider a new vanilla fingerprinting scheme, and show the robustness of the proposed approach using this vanilla fingerprinting scheme.
- We further deepen the investigation on the mitigation ability of the database and evaluate the performance of the mitigation techniques under (i) the prior knowledge

asymmetric setting, where database owner and malicious SP perform mitigations and attacks, respectively, based on their different prior knowledge on the data correlations; and (ii) the higher order correlation attack setting, where the database owner mitigates higher order column-wise correlation attacks using only the pairwise correlation among attributes. Experiment results show that our developed mitigation technique are still robust under the considered disadvantaged settings.

Roadmap. 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 (including the identified correlation attacks), and evaluation metrics. In Section 5, we develop robust fingerprinting against correlation attacks. We evaluate the impact of correlation attacks and the performance of the devised mitigation techniques in Section 6 under various settings. Finally, Section 7 concludes the paper.

2 RELATED WORK

The seminal work of database watermarking scheme (which inserts the same marks to all shared database copies to claim copyright) was proposed by Agrawal et al. [11] with the assumption that the database consumer can tolerate a small amount of error in the watermarked databases. Then, based on [11], some database fingerprinting schemes (that insert different customized marks into all shared copies to distinguish the recipients) have been devised [6], [7], [8]. For example, Li et al. [6] develop a database fingerprinting scheme by extending [11] to enable the insertion and extraction of arbitrary bit-strings in relations. They also 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 [6], [7], [8] pseudorandomly determine the fingerprint positions in a database, they are not robust against our identified correlation attacks. In our earlier work [1], we have considered [6] as the vanilla fingerprinting scheme. In this work, to show the widespread vulnerability of existing fingerprinting schemes against correlation attacks and the fact that the proposed mitigation techniques can work as post-processing steps for any off-the-shelf database fingerprinting schemes, we also consider another fingerprinting scheme [8] as the vanilla scheme, corroborate its vulnerability, and then show how the proposed mitigation techniques make it robust against the identified correlation attacks.

There are only a few works [12], [13] considering data correlation during watermarking and fingerprinting, but they are limited to data belonging to a single individual (one row in a relational database). To be more specific, Yilmaz et al. [12] develop a probabilistic fingerprinting scheme by considering the conditional probabilities between data points in an individual's data record. Ayday et al. [13] propose an optimization-based fingerprinting scheme for sharing sequential data by minimizing the probability of collusion attack with data correlation being one of the constraints. Our work differs from them since we focus on developing robust fingerprint scheme for relational databases, which (i)

1. We say a database fingerprinting scheme is vanilla if it is vulnerable to attacks that leverages the correlations among data entries.

contain large amount of data records from different individuals, (ii) include both column- and row-wise correlations, and (iii) have different utility requirements.

3 TWO VANILLA FINGERPRINTING SCHEMES

In this work, we consider two vanilla fingerprinting schemes, and show their vulnerability against the identified correlation attacks. In the first vanilla scheme [6] denoted as FP_1 , the fingerprint of a specific SP is obtained using a cryptographic hash function ($Hash(K|n)$), whose input is the concatenation of the database owner's secret key (K) and the SP's public series number (n). For fingerprint insertion, FP_1 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 pseudorandomly generated mask bits and fingerprint bits. For fingerprint extraction, FP_1 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.² The main reason that we consider FP_1 is because it is shown to have high robustness, e.g., the probability of detecting no fingerprint due to random bit flipping attack is negligible [6].

The second vanilla fingerprinting scheme [8] is denoted as FP_2 . The fingerprint of each SP used in FP_2 is obtained in the same fashion as FP_1 . When sharing database with different SPs, FP_2 first extracts all the fingerprintable bit positions in a 2D matrix, divides it into blocks of equal size, and then inserts fingerprint at different bit positions in different blocks determined by the database owner's secret key and the IDs of the SPs. We visualize the workflow of FP_2 via a toy example of a relational database in Figure 1. When extracting the fingerprint from a pirated database, FP_2 locates the exact positions of the potentially changed bits in each block, and recovers the fingerprint bit from the result of the XOR between the original bit value and the fingerprinted bit value.³ We consider FP_2 , because it speeds up FP_1 by adopting block-wise operations.

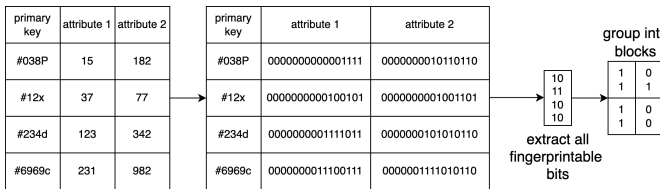


Fig. 1: Toy example of workflow of FP_2 .

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. In our previous work [1], we have empirically validated that only changing the LSB indeed leads to higher utility than altering one of the least k significant bits (LkSB) of selected entries. In

2. Please refer to Figure 1 on page 36 and Figure 2 on page 37 in [6] for the pseudocodes of fingerprint insertion and extraction of FP_1 .

3. Please refer to Tables 5 and 6 on page 460 and 461 of [8] for procedure of fingerprint insertion and extraction of FP_2 .

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. Our developed robust fingerprinting scheme inherits all the properties of the vanilla schemes because (i) it uses the vanilla schemes 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 naming convention 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 $\tilde{\mathbf{R}}$, and the pirated database leaked by a malicious SP as $\bar{\mathbf{R}}$, respectively. Both $\tilde{\mathbf{R}}$ and $\bar{\mathbf{R}}$ are represented using 3 input parameters showing the techniques that are adopted to generate them. 3 input parameters for $\tilde{\mathbf{R}}(\alpha, \beta, \eta)$ represent which processes have been applied to the database during fingerprinting, where α represents the vanilla fingerprinting, β is the proposed mitigation technique (Dfs_{row}) against the row-wise correlation attack, and η indicates the proposed mitigation technique (Dfs_{col}) against the column-wise correlation attack. On the other hand, 3 input parameters for $\bar{\mathbf{R}}(\alpha, \beta, \eta)$ represent which attacks have been adopted by the malicious SP to compromise the fingerprinted database, where α represents the random bit flipping attack (Atk_{rnd}), β is the row-wise correlation attack (Atk_{row}), and η denotes the column-wise correlation attack (Atk_{col}). We summarize the frequently used notations in Table 1.⁴

The mitigation techniques are equipped with the database owner's prior knowledge, i.e. the row-wise correlations (\mathcal{S}') and the column-wise correlations (\mathcal{J}'). Whereas, the correlation attacks are equipped with the malicious SP's prior knowledge on the row-wise correlations (\mathcal{S}) and the column-wise correlations (\mathcal{J}). Generally, $\mathcal{S}' \neq \mathcal{S}$ 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 has more accurate or at least equally accurate knowledge about row-wise and column-wise correlations of the database. In Section 6.4, we will investigate the impact of prior knowledge asymmetry, which is one of the new contributions of this paper.

4.1 System Model

In Figure 2, we show the fingerprint system using the vanilla fingerprinting scheme as an example. Specifically, we consider the database owner (Alice) with a relational database \mathbf{R} containing the data records of M individuals. We denote the set of attributes in \mathbf{R} as \mathcal{F} and the i th row in \mathbf{R} as \mathbf{r}_i .

4. In our previous work [1], we use $Atk_{row}(\mathcal{S}(\mathbf{R}))$ (or $Atk_{col}(\mathcal{J}(\mathbf{R}))$) to represent the row- (or column-) wise correlation attack using prior knowledge \mathcal{S} (or \mathcal{J}) directly computed from \mathbf{R} , and use $Dfs_{row}(\mathcal{S}'(\mathbf{R}))$ (or $Dfs_{col}(\mathcal{J}'(\mathbf{R}))$) to denote the row- (or column-) wise mitigation using prior knowledge \mathcal{S}' (or \mathcal{J}') obtained from \mathbf{R} . In this paper, we just use Atk_{row} , Atk_{col} , Dfs_{row} , and Dfs_{col} for notation simplicity, and only declare the specific knowledge information in Section 6.4.

\mathbf{R}	the original database owned by the database owner (Alice)
$\tilde{\mathbf{R}}$	a generic fingerprinted database shared by the database owner
\mathbf{R}_κ	a generic pirated database generated by the malicious SP
$\tilde{\mathbf{R}}(\alpha, \beta, \eta)$	a random database obtained by keeping κ percentage of data records in \mathbf{R} randomly
$\tilde{\mathbf{R}}(\alpha, \beta, \eta)$	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
$\bar{\mathbf{R}}(\alpha, \beta, \eta)$	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
$\hat{\mathcal{S}}$ and $\hat{\mathcal{J}}$	the empirical row-wise and column-wise correlations obtained from a generic fingerprinted database $\tilde{\mathbf{R}}$
Atk_{rnd}	the random bit flipping attack
Atk_{row}	the row-wise correlation attack launched by the malicious SP by using prior knowledge \mathcal{S} (see Algorithm 2)
Atk_{col}	the column-wise correlation attack launched by the malicious SP by using prior knowledge \mathcal{J} (see Algorithm 1)
Dfs_{row}	the mitigation technique using prior knowledge \mathcal{S}' to alleviate row-wise correlation attack (see Algorithm 4)
Dfs_{col}	the mitigation technique using prior knowledge \mathcal{J}' to alleviate column-wise correlation attack (see Algorithm 3)

TABLE 1: Frequently used notations in the paper.

Alice shares her data with multiple service providers (SPs) to receive specific services. To discourage 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 $\tilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$. Both f_{SP_i} and $\tilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$ 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 2). 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 $\tilde{\mathbf{R}}_{\text{SP}_i}(\text{FP}, \emptyset, \emptyset)$.

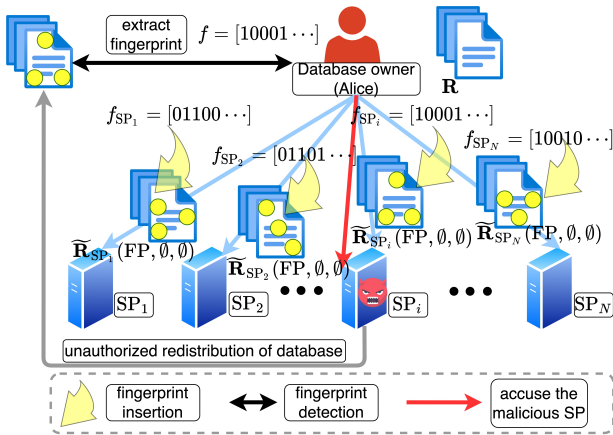


Fig. 2: The vanilla fingerprinting system, where Alice adds a unique fingerprint in each copy of her database when sharing. She is able to identify the malicious SP who pirates and redistributes her database.

4.2 Threat Model

Fingerprinted database is subject to various attacks. In this paper, we focus on “single-handed” malicious SPs, i.e., the malicious SPs that launch attacks on their own and do not

merge their individual versions of fingerprinted databases to forge a pirated copy (which is known as the collusion attack [14]). In Figure 3, we show the attacks studied in this paper. Note that in all considered attacks, a malicious SP can change/modify most of the entries in $\tilde{\mathbf{R}}$ 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 $\tilde{\mathbf{R}}$ to distort the fingerprint.

4.2.1 Random Bit Flipping Attack

In this attack, to pirate a database, a malicious SP randomly selects entries in $\tilde{\mathbf{R}}$ and flips their LSBs [11]. The vanilla fingerprint schemes are robust against this attack [6] as shown in Figure 3(i). Alice shares fingerprinted copies of her database $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ by only applying FP. If a malicious SP (SP_i) tries to distort the fingerprint in its received copy using the random bit flipping attack, and redistributes it, Alice can still detect SP_i 's fingerprint in the pirated copy with a high probability, and correctly accuse SP_i . There are also attacks which are known as subset and superset attacks. In subset attack, a malicious SP generates a pirated copy of $\tilde{\mathbf{R}}$ by randomly selecting data records from it. Superset attack (the dual of subset attack) mixes $\tilde{\mathbf{R}}$ with other databases to create a pirated one. However, these two attacks are much weaker than Atk_{rnd} [6], [12]. Thus, we do not consider them in this paper.

4.2.2 Correlation Attacks

In correlation attacks (first identified in our previous work [1]), a malicious SP utilizes the inherent correlations in the data to more accurately identify the fingerprinted entries, and hence distort the fingerprint.

Column-wise Correlation Attack (Atk_{col}). In Atk_{col} , we assume that the malicious SP has prior knowledge about the correlations among each pair of attributes (or columns

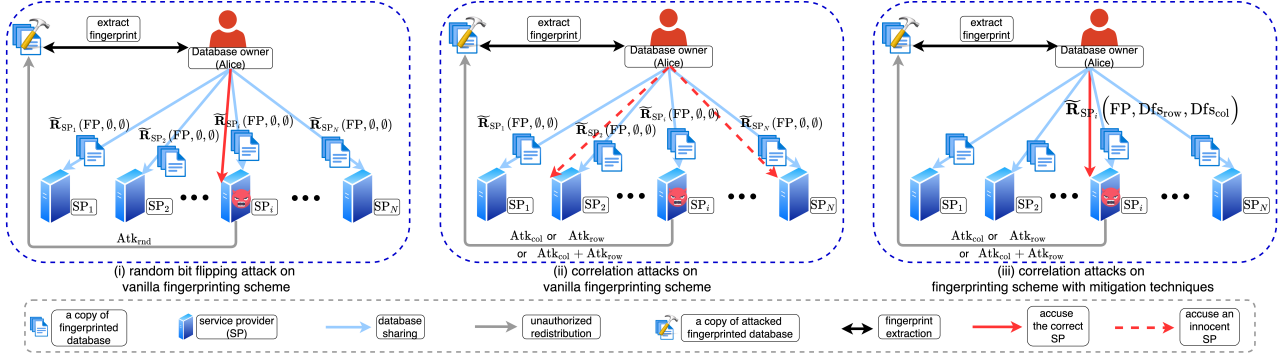


Fig. 3: Representative attacks. (i) If Alice inserts fingerprinting using the vanilla scheme, and the malicious SP_i conducts random bit flipping attack (Atk_{rnd}) on its received copy. Then, with high probability (w.h.p.), Alice can correctly accuse it for data leakage. (ii) If the malicious SP_i conducts any correlation attack, e.g., the column-wise correlation attack (Atk_{col}), the row-wise correlation attack (Atk_{row}), or the combination of them, on the vanilla fingerprinted database. Then, w.h.p., 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 (Dfs_{col}) and the row-wise correlation defense (Dfs_{row}), after the vanilla fingerprinting scheme. Then, w.h.p., she can correctly identify SP_i as the traitor even if it conducts any of the correlation attacks.

in the database) characterized by the set of joint probability distributions, i.e., $\mathcal{J} = \{J_{p,q}|p, q \in \mathcal{F}, p \neq q\}$. Once receiving the fingerprinted database $\tilde{\mathbf{R}}$, the malicious SP first calculates a new set of joint probability distributions based on $\tilde{\mathbf{R}}$. Then, it compares the new joint distributions with its prior knowledge \mathcal{J} , and flips the entries in $\tilde{\mathbf{R}}$ that causes large discrepancy between them (based on a predetermined threshold τ_{col}^{Atk}).

For completeness, we revisit the procedure to launch Atk_{col} in Algorithm 1. In particular, the malicious SP first calculates the empirical joint distributions among pairs of attributes in $\tilde{\mathbf{R}}$, denoted as $\tilde{\mathcal{J}}$. Then, it compares each joint distribution in $\tilde{\mathcal{J}}$ (i.e., $\tilde{J}_{p,q}$) with that in \mathcal{J} (i.e., $J_{p,q}$). If the absolute difference of joint probabilities when attribute p takes value a and attribute q takes value b (i.e., $|J_{p,q}(a,b) - \tilde{J}_{p,q}(a,b)|$) is higher than a threshold τ_{col}^{Atk} , then, the malicious SP queries the row indices of the data records in $\tilde{\mathbf{R}}$ whose attributes p and q take values a and b , respectively, and collects the row indices in a set \mathcal{I} , i.e., $\mathcal{I} = \text{row_index_query}(\tilde{\mathbf{R}}.p == a \text{ and } \tilde{\mathbf{R}}.q == b)$ ($\tilde{\mathbf{R}}.p$ includes attribute p of all data record in database $\tilde{\mathbf{R}}$). 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. 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 \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 $\mathcal{H} = \mathcal{H} \cup \{i, \text{mode}(\mathcal{A}_i)\}$, where \mathcal{A}_i includes all the attributes that are paired with row index i in set \mathcal{P} , and $\text{mode}(\mathcal{A}_i)$ returns the most frequent attribute in \mathcal{A}_i . Then, the malicious SP launches the column-wise correlation attack by flipping the LSB of entries in $\tilde{\mathbf{R}}$ whose positions are in \mathcal{H} , i.e., $\tilde{\mathbf{R}}.(i, p), \forall \{i, p\} \in \mathcal{H}$ ($\tilde{\mathbf{R}}.(i, p)$ represents the value of attribute p for the i th data record in $\tilde{\mathbf{R}}$).

Algorithm 1: Column-wise Correlation Attack [1]

Input : Fingerprinted database $\tilde{\mathbf{R}}$, malicious SP's prior knowledge on the pairwise joint distributions among attributes, \mathcal{J} , and attack rounds t .

Output: $\tilde{\mathbf{R}}(\emptyset, \emptyset, Atk_{col})$.

```

1 Initialize  $cnt = 1$ , and initialize  $\mathcal{Z} = \emptyset$ ;
2 while  $cnt \leq t$  do
3   Initialize  $\mathcal{P} = \emptyset, \mathcal{H} = \emptyset$ ;
4   Update the empirical joint distributions set  $\tilde{\mathcal{J}}$  using  $\tilde{\mathbf{R}}$ ;
5   for all  $p, q \in \mathcal{F}, p \neq q$  do
6     for all  $a \in [0, k_p - 1], b \in [0, k_q - 1]$  do
7       if  $|J_{p,q}(a, b) - \tilde{J}_{p,q}(a, b)| \geq \tau_{col}^{Atk}$  then
8          $\mathcal{I} = \text{row\_index\_query}(\tilde{\mathbf{R}}.p == a \text{ and } \tilde{\mathbf{R}}.q == b)$ ;
9         for all row index  $i \in \mathcal{I}$  do
10          if  $\{i, p\} \notin \mathcal{P}$  then
11             $\mathcal{P} = \mathcal{P} \cup \{i, p\}$ ;
12          if  $\{i, q\} \notin \mathcal{P}$  then
13             $\mathcal{P} = \mathcal{P} \cup \{i, q\}$ ;
14   for all row index and attribute tuple  $\{i, p\} \in \mathcal{P}$  do
15     Collect all attributes that are paired with row index  $i$  into  $\mathcal{A}_i$ ;
16      $\mathcal{H} = \mathcal{H} \cup \{i, \text{mode}(\mathcal{A}_i)\}$ ;
17   for all highly suspicious row index and attribute tuple  $\{i, p\} \in \mathcal{H}$  do
18     if  $\{i, p\} \notin \mathcal{Z}$  then
19       Change the LSB of  $\tilde{\mathbf{R}}.(i, p)$ ;
20        $\mathcal{Z} = \mathcal{Z} \cup \{i, p\}$ ; // Append in  $\mathcal{Z}$  to avoid repeated flipping.
21    $cnt = cnt + 1$ ;
```

Row-wise Correlation Attack (Atk_{row}). 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 $\mathcal{S} = \{s_{ij}^{\text{comm}_c} | i, j \in \text{comm}_c, i \neq j, c \in [1, C]\}$, where $s_{ij}^{\text{comm}_c} = e^{-\text{dist}(\mathbf{r}_i, \mathbf{r}_j)}$ is the statistical relationship between individuals (data records) i and j in community comm_c ($\text{dist}(\mathbf{r}_i, \mathbf{r}_j)$ denotes the Hamming distance between \mathbf{r}_i and \mathbf{r}_j). Once it receives the fingerprinted database $\tilde{\mathbf{R}}$, the malicious SP first calculates a new set of statistical relationships based on $\tilde{\mathbf{R}}$, then it compares the newly computed statistical relationships with \mathcal{S} , and changes the entries that lead to large discrepancy (based on a predetermined threshold $\tau_{\text{row}}^{\text{Atk}}$) between the two sets of statistical relationships.

For completeness, we revisit the procedure of Atk_{row} in Algorithm 2. In particular, after receiving the fingerprinted database, the malicious SP computes a new set of statistical relationships among pairs of individuals in each of the communities using $\tilde{\mathbf{R}}$, i.e., $\tilde{\mathcal{S}} = \{\tilde{s}_{ij}^{\text{comm}_c} | i, j \in \text{comm}_c, i \neq j, c \in [1, C]\}$, where $\tilde{s}_{ij}^{\text{comm}_c} = e^{-\text{dist}(\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_j)}$ is the statistical relationship between the i th and j th data records (i.e., $\tilde{\mathbf{r}}_i$ and $\tilde{\mathbf{r}}_j$) in $\tilde{\mathbf{R}}$. Then, the malicious SP flips the LSBs of all attributes of a data record \mathbf{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., $\sum_{j \neq i}^{n_c} |s_{ij}^{\text{comm}_c} - \tilde{s}_{ij}^{\text{comm}_c}| \geq \tau_{\text{row}}^{\text{Atk}}, i, j \in \text{comm}_c$.

Algorithm 2: Row-wise correlation Attack [1]

Input : Fingerprinted database, $\tilde{\mathbf{R}}$, malicious SP's prior knowledge on the row-wise correlations \mathcal{S} and individuals' affiliation to the C communities.

Output: $\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \emptyset)$.

- 1 Obtain the new set of pairwise statistical relationships among individuals in each community from $\tilde{\mathbf{R}}$, i.e., $\tilde{\mathcal{S}}$;
 - 2 **forall** $\text{comm}_c, c \in [1, C]$ **do**
 - 3 **forall** individual $i \in \text{comm}_c$ **do**
 - 4 **if** $\sum_{j \neq i}^{n_c} |s_{ij}^{\text{comm}_c} - \tilde{s}_{ij}^{\text{comm}_c}| \geq \tau_{\text{row}}^{\text{Atk}}$ **then**
 - 5 Flip the LSBs of all attributes of \mathbf{r}_i in $\tilde{\mathbf{R}}$;
-

Integrated Correlation Attack. In practice, the malicious SP can also apply Atk_{row} followed by Atk_{col}. This is because (i) Atk_{row} is computationally light and modifies significantly less entries in $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ compared to Atk_{col} (as we will show in Section 6.2); and (ii) if Atk_{col} is applied first, it will change the row-wise correlations (P_{row}) significantly, yet, if Atk_{row} is applied first, it only has a small impact on the column-wise correlations P_{col} (as we will also show in Section 6.2). Figure 3(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 correlation attacks to distort the fingerprint.

Finally, Figure 3(iii) shows that if Alice uses the proposed mitigation techniques (i.e., Dfs_{row} and Dfs_{col}) (discussed in Section 5) after FP to improve the robustness of the added fingerprint and shares $\tilde{\mathbf{R}}(\text{FP}, \text{Dfs}_{\text{row}}, \text{Dfs}_{\text{col}})$, 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.

4.3 Fingerprint Robustness Metrics

The primary goal of a malicious SP is to distort the fingerprint in $\tilde{\mathbf{R}}$, thus we consider the following fingerprint robustness metrics about a pirated database $\tilde{\mathbf{R}}$.

Number of compromised fingerprint bits num_{cmp}.

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

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

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

$$r = \begin{cases} \text{"uniquely accountable"}, & \text{if } m_0 > \sum_{l=1}^L \mathbf{1}\{f_{\text{SP}_i}(l) = \bar{f}(l)\}, \forall \text{SP}_i \in \mathcal{T}, \\ \text{"top } t \text{ accountable"}, & \text{otherwise} \end{cases}$$

where $m_0 = \sum_{l=1}^L \mathbf{1}\{f_{\text{SP}_{\text{malicious}}}(l) = \bar{f}(l)\}$ 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 $t = \frac{\sum_{\text{SP}_i \in \mathcal{T}} \mathbf{1}\{(\sum_{l=1}^L \mathbf{1}\{f_{\text{SP}_i}(l) = \bar{f}(l)\}) \geq m_0\}}{|\mathcal{T}|} \times 100\%$, 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% accountable, which suggests that Alice will accuse other innocent SPs with high probability. In contrast, if $t = 1\%$, then the malicious SP's accountable ranking increases and makes it among the top 1% accountable SPs, and Alice will accuse other innocent SPs with low probability. Essentially, a high accountable rank r corresponds to either (i) a "low t " or (ii) the uniquely accountable case.

Attack strength of Atk_{col}. 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 Atk_{col}. 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 Atk_{col}, we represent such confidence as $\text{Conf}_{\text{Atk}_{\text{col}}}(\frac{1}{\gamma}; p, a)$. Here, $\gamma \in (0, 1)$ is the fingerprinting ratio and we use $\frac{1}{\gamma}$ to investigate the asymptotic behavior of the malicious SP's confidence gain, which is defined as the ratio $G_{\text{col}}(\frac{1}{\gamma}; p, a) = \text{Conf}_{\text{Atk}_{\text{col}}}(\frac{1}{\gamma}; p, a) / \text{Conf}_{\text{Atk}_{\text{rnd}}}(\frac{1}{\gamma}; p, a)$. Thus, we have the following proposition (proved in our previous work [1]).

Proposition 1. By launching $\text{Conf}_{\text{Atk}_{\text{col}}}$, 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

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

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.

In practice, we set $\tau_{col}^{Atk} \ll \frac{\gamma}{|T|} 2freq_a^p$, thus, we have $G_{col}(\frac{1}{\gamma}; p, a) = \Theta(\frac{|T|}{\gamma 2freq_a^p})$. In Section 6, we consider $|T| = 13$ and $\gamma = \frac{1}{35}$, then Atk_{col} is at least 455 times more powerful (i.e., in terms of confidence gain) than Atk_{rnd} for the considered database.

Attack strength of Atk_{row} . We analyze the impact of Atk_{row} by denoting the malicious SP's confidence that an entry (\mathbf{r}_i) is fingerprinted as $Conf_{Atk_{rnd}}(\frac{1}{\gamma}; \mathbf{r}_i)$ and $Conf_{Atk_{row}}(\frac{1}{\gamma}; \mathbf{r}_i)$, under Atk_{rnd} and Atk_{row} , respectively. Then, the confidence gain of the malicious SP is $G_{row}(\frac{1}{\gamma}; \mathbf{r}_i) = \frac{Conf_{Atk_{row}}(\frac{1}{\gamma}; \mathbf{r}_i)}{Conf_{Atk_{rnd}}(\frac{1}{\gamma}; \mathbf{r}_i)}$, calculated in the following proposition (also proved in [1]).

Proposition 2. By launching $Conf_{Atk_{row}}$, the malicious SP's maximum confidence gain about an entry in \mathbf{R} is fingerprinted can be shown asymptotically as

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

where n_c is the number of individuals in a specific community.

$G_{row}(\frac{1}{\gamma}; \mathbf{r}_i)$ represents the complement of the binomial cumulative distribution function evaluated at τ_{row}^{Atk} (the binomial distribution is $B(n_c - 1, 2\gamma - \gamma^2)$). In the experiment considered in Section 6.2, we set $\tau_{row}^{Atk} = 0.1$, then for a community with only 50 individuals, Atk_{row} is already approximately 32.95 times more powerful than Atk_{rnd} . In Section 6.2.3, we will experimentally validate the theoretical findings in Proposition 1 and 2 using a real-world database.

4.4 Utility Metrics

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

Accuracy of $\tilde{\mathbf{R}}$. $Acc(\tilde{\mathbf{R}}) = 1 - \tilde{\mathbf{R}} \oplus \mathbf{R} / (M * |\mathcal{F}|)$, where \oplus is the symmetric difference operator that counts the number of different entries in the fingerprinted and the original databases. $Acc(\tilde{\mathbf{R}})$ measures the percentage of matched entries between $\tilde{\mathbf{R}}$ and \mathbf{R} .

Preservation of column-wise correlations.

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

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 $\tilde{J}_{p,q}(a,b)$ (or $J_{p,q}(a,b)$) is the joint probability that attribute p takes value a and attribute q takes value b in $\tilde{\mathbf{R}}$ (or \mathbf{R}). P_{col} calculates the fraction of instances of $|\tilde{J}_{p,q}(a,b) - J_{p,q}(a,b)|$ that do not exceed a predetermined threshold τ_{col} before and after fingerprinting \mathbf{R} .

Preservation of row-wise correlations.

$$P_{row}(\tilde{\mathbf{R}}) = 1 - \frac{\sum_{c=1}^C \sum_{i,j \in comm_c, i \neq j} \mathbf{1}\{|\tilde{s}_{i,j}^{comm_c} - s_{i,j}^{comm_c}| \geq \tau_{row}\}}{\sum_{c=1}^C n_c(n_c-1)},$$

where $comm_c$ represents the set of all individuals in a community c , $\tilde{s}_{i,j}^{comm_c}$ (or $s_{i,j}^{comm_c}$) is the statistical relationship between individual i and j belonging to $comm_c$ in $\tilde{\mathbf{R}}$ (or \mathbf{R}), n_c is the number of individuals in $comm_c$, and C is the

number of communities. In essence, $P_{row}(\tilde{\mathbf{R}})$ evaluates the fraction of statistical relationship that has absolute difference less than τ_{row} in the entire population before and after fingerprinting.

Preservation of empirical covariance matrix.

$$P_{cov} = 1 - \|\text{cov}(\tilde{\mathbf{R}}) - \text{cov}(\mathbf{R})\|_F / \|\text{cov}(\mathbf{R})\|_F,$$

where $\text{cov}(\mathbf{R}) = \sum_{i=1}^M \mathbf{r}_i^T \mathbf{r}_i / M$ is the empirical covariance matrix of data records in \mathbf{R} . P_{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 often utilized to establish predictive models, e.g., regression and probability distribution fitting [15], [16]. 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 $\tilde{\mathbf{R}}$ generated by the malicious SP can also be quantified using the same metrics, i.e., $Acc(\tilde{\mathbf{R}})$, $P_{col}(\tilde{\mathbf{R}})$, $P_{row}(\tilde{\mathbf{R}})$, and $P_{cov}(\tilde{\mathbf{R}})$. As discussed, a malicious SP can successfully (without being accused) distort the fingerprint easily by over-distorting $\tilde{\mathbf{R}}$, 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 ROBUST FINGERPRINTING AGAINST IDENTIFIED CORRELATION ATTACKS

Now, we propose the mitigation techniques (that can serve as post-processing steps for any off-the-shelf (vanilla) fingerprinting schemes) against the correlation attacks. To provide robustness against column- and row-wise correlation attack, i.e., Atk_{col} and Atk_{row} , 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 $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, such that the post-processed fingerprinted database has column-wise correlations close to \mathcal{J}' and row-wise correlations far from \mathcal{S}' .

5.1 Robust Fingerprinting Against Atk_{col}

5.1.1 Mitigation via Mass Transportation

To make a vanilla fingerprinting scheme robust against column-wise correlation attack, the main goal of Dfs_{col} is to transform $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ to have column-wise joint distributions close to the reference joint distributions in \mathcal{J}' . We develop Dfs_{col} using "optimal transport" [17], which moves the probability mass of the marginal distribution of each attribute in $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ 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 $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ to obtain $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$. While doing this, the new empirical joint distributions calculated from $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$ also become close to the ones in \mathcal{J}' . Even if \mathcal{J}' is less accurate than \mathcal{J} , which is utilized by the malicious SP, we show (in Section 6.4) that Dfs_{col} can still mitigate Atk_{col} . Thus, the malicious SP can still be accused with high probability (i.e., either uniquely accusable or with a high accusable ranking r) if it leaks a copy of $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$.

In particular, for a specific attribute (column) p , we denote its marginal distribution obtained from the (vanilla) fingerprinted database as $\text{Pr}(C_{\tilde{p}})$, and that obtained from the marginalization of a reference $J'_{p,q}$ distribution in \mathcal{J}' as $\text{Pr}(C_{p'}) = J'_{p,q} \mathbf{1}^T$ (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 $\text{Pr}(C_{\tilde{p}})$ to resemble $\text{Pr}(C_{p'})$, we need to find another joint distribution (i.e., the mass transportation plan) $G_{\tilde{p},p'} \in \mathcal{R}^{k_p \times k_p}$ (k_p is the number of possible values that attribute p can take), whose marginal distributions are identical to $\text{Pr}(C_{\tilde{p}})$ and $\text{Pr}(C_{p'})$. Let a and b be two distinct values that attribute p can take ($a, b \in [0, k_p - 1]$). Then, $G_{\tilde{p},p'}(a, b)$ indicates that the database owner should change $G_{\tilde{p},p'}(a, b)$ percentage of entries in $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ 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 $\text{Pr}(C_{\tilde{p}})$ close to $\text{Pr}(C_{p'})$. In practice, such a transportation plan can be obtained by solving a regularized optimal transportation problem, i.e., the entropy regularized Sinkhorn distance minimization [18] as follows:

$$d(\text{Pr}(C_{\tilde{p}}), \text{Pr}(C_{p'}), \lambda_p) = \min_{G_{\tilde{p},p'} \in \mathcal{G}(\text{Pr}(C_{\tilde{p}}), \text{Pr}(C_{p'}))} \langle G_{\tilde{p},p'}, \Theta_{\tilde{p},p'} \rangle_F - \frac{H(G_{\tilde{p},p'})}{\lambda_p}, \quad (1)$$

where $\mathcal{G}(\text{Pr}(C_{\tilde{p}}), \text{Pr}(C_{p'})) = \{G \in \mathcal{R}^{k_p \times k_p} | G\mathbf{1} = \text{Pr}(C_{\tilde{p}}), G^T\mathbf{1} = \text{Pr}(C_{p'})\}$ is the set of all joint probability distributions whose marginal distributions are the probability mass functions of $\text{Pr}(C_{\tilde{p}})$ and $\text{Pr}(C_{p'})$. $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product of two matrices with the same size. Also, $\Theta_{\tilde{p},p'}$ is the transportation cost matrix and $\Theta_{\tilde{p},p'}(a, b) > 0$ represents the cost to move a unit percentage of mass from $\text{Pr}(C_{\tilde{p}} = a)$ to $\text{Pr}(C_{\tilde{p}} = b)$. Finally, $H(G_{\tilde{p},p'}) = -\langle G_{\tilde{p},p'}, \log G_{\tilde{p},p'} \rangle_F$ calculates the information entropy of $G_{\tilde{p},p'}$ and $\lambda_p > 0$ is a tuning parameter. In practice, (1) can be solved by iteratively rescaling rows and columns of the initialized $G_{\tilde{p},p'}$ to have desired marginal distributions. The obtained $G_{\tilde{p},p'}$ is more heterogeneous for larger values of λ_p . This suggests that the transportation plan tends to move the mass of $\text{Pr}(C_{\tilde{p}} = a)$ to the adjacent instances, i.e., $b = a - 1$ or $b = a + 1$. In contrast, the obtained $G_{\tilde{p},p'}$ is more homogeneous for smaller values of λ_p , which suggests that the transportation plan tends to move the mass of $\text{Pr}(C_{\tilde{p}} = a)$ to all other instances. A homogeneous plan makes $\text{Pr}(C_{\tilde{p}})$ much closer to $\text{Pr}(C_{p'})$ 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 $\text{Pr}(C_{\tilde{p}})$ and $\text{Pr}(C_{p'})$ after the mass transportation.

In Section 6.3, we will try different values of λ_p to strike a balance between mitigation performance and data utility.

5.1.2 Algorithm Description

In the following, we formally describe the procedure of Dfs_{col} . After Alice generates $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ using the vanilla fingerprinting scheme, she evaluates the new joint distributions of all pairs of attributes, i.e., $\tilde{J}_{p,q}, p, q \in \mathcal{F}, p \neq q$, and compares them with the reference joint distributions $J'_{p,q}, p, q \in \mathcal{F}, p \neq q$. If the discrepancy between a particular pair of joint distributions exceeds a predetermined threshold, i.e., $\|\tilde{J}_{p,q} - J'_{p,q}\|_F \geq \tau_{\text{col}}^{\text{Dfs}}$, Alice records both attributes p and q in a set \mathcal{Q} . For all the attributes in \mathcal{Q} , Alice obtains $\text{Pr}(C_{\tilde{p}})$ from $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ and calculates $\text{Pr}(C_{p'}) = J'_{p,q} \mathbf{1}^T$. Next, she gets the optimal transportation plan for attribute p by solving (1). Then, she changes the instances of $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ to other instances by following the transportation moves suggested by $G_{\tilde{p},p'}$, i.e., given $G_{\tilde{p},p'}(a, b)$, Alice randomly samples $G_{\tilde{p},p'}(a, b)$ fraction of entries (excluding the fingerprinted entries) whose attribute p takes value a and changes them to b . We summarize the procedure of Dfs_{col} in Algorithm 3.

Algorithm 3: Dfs_{col} : defense against column-wise correlation attack.

Input : Vanilla fingerprinted database $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, 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: $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$.

- 1 Initialize $\mathcal{Q} = \emptyset$;
 - 2 Obtain the empirical joint distributions set $\tilde{\mathcal{J}}$ using $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$;
 - 3 **forall** $p, q \in \mathcal{F}, p \neq q$ **do**
 - 4 **if** $\|J'_{p,q} - \tilde{J}_{p,q}\|_F > \tau_{\text{col}}^{\text{Dfs}}$ **then**
 - 5 $\mathcal{Q} = \mathcal{Q} \cup p \cup q$;
 - 6 **forall** $p \in \mathcal{Q}$ **do**
 - 7 Initialize the mass movement cost matrix $\Theta_{\tilde{p},p'}$ and tuning parameter λ_p ;
 - 8 Obtain empirical marginal distribution $\text{Pr}(C_{\tilde{p}})$ from $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$;
 - 9 Initialize $G_{\tilde{p},p'} = e^{-\lambda_p \Theta_{\tilde{p},p'}}$;
 - 10 **while not converge do**
 - 11 Scale the rows of $G_{\tilde{p},p'}$ to make the rows sum to the marginal distribution $\text{Pr}(C_{\tilde{p}})$;
 - 12 Scale the columns of $G_{\tilde{p},p'}$ to make the columns sum to the marginal distribution $\text{Pr}(C_{p'})$;
 - 13 **forall** $a \in [0, k_p - 1]$ **do**
 - 14 **forall** $b \in [0, k_p - 1], b \neq a$ **do**
 - 15 Sample $G_{\tilde{p},p'}(a, b)$ percentage of entries from $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ (excluding the vanilla fingerprinted entries) whose attribute p takes value a , and change their value to b ;
 - 16 **Return** $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$.
-

5.1.3 Mitigating Atk_{col} When Malicious SP Has Access to Higher Order Column-wise Correlations

In a column-wise correlation attack, a malicious SP may also take advantage of higher-order correlations, e.g., the joint

distributions of three-tuples of attributes ($J_{p,q,f}, p, q, f \in \mathcal{F}$). The proposed technique is also robust against such an attack by just moving the mass of $\Pr(C_{\tilde{p}})$ to resemble the marginalization of joint distribution. We will empirically corroborate this in Section 6.5. Since in practice, Alice does not have the knowledge about which order of column-wise correlations the malicious SP will use, and she cannot explicitly model all potential correlations, the proposed mitigation technique, which only considers the marginal distributions (the most general one) can be considered as an “universal prescription” to mitigate the impact of other column-wise correlation attacks.

5.2 Robust Fingerprinting Against Atk_{row}

To make a vanilla fingerprinting scheme also robust against row-wise correlation attack (in Section 4.2.2), we develop another mitigation technique, i.e., Dfs_{row} . The main goal of Dfs_{row} 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 Dfs_{col} , which makes the newly obtained joint distributions resemble the prior knowledge, we design Dfs_{row} by changing selected entries of non-fingerprinted data records to make the newly obtained statistical relationships as contradictory to 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 6.2), 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 contradictory to 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 Atk_{row} , the malicious SP will identify wrong data records (r_i), which causes $\sum_{j \neq i} |s_{ij}^{\text{comm}_c} - \tilde{s}_{ij}^{\text{comm}_c}| \geq \tau_{\text{row}}^{\text{Atk}}$, and hence change the non-fingerprinted records.

In Dfs_{row} , Alice selects a subset of non-fingerprinted data records in a comm_c , i.e., $\mathcal{E}_c \subset \text{comm}_c$, and changes their value to $\hat{r}_i, i \in \mathcal{E}_c$, 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 Dfs_{row} . This is formulated as the following optimization problem:

$$\begin{aligned} \max_{\mathcal{E}_c, \hat{r}_i} \quad & d(\mathcal{E}_c) = \left| \sum_{j \in \text{comm}_c / \mathcal{E}_c} \sum_{i \in \mathcal{E}_c} \left| s_{ij}^{\text{comm}_c} - \tilde{s}_{ij}^{\text{comm}_c} \right| \right. \\ & \left. - \sum_{j \in \text{comm}_c / \mathcal{E}_c} \sum_{i \in \mathcal{E}_c} \left| s_{ij}^{\text{comm}_c} - \hat{s}_{ij}^{\text{comm}_c} \right| \right| \\ \text{s.t.} \quad & \mathcal{E}_c \subset \text{comm}_c / Q_c, \\ & \hat{s}_{ij}^{\text{comm}_c} = e^{-\text{dist}(\hat{r}_i, r_j)}, i \in \mathcal{E}_c, j \in \text{comm}_c / \mathcal{E}_c, \\ & \hat{r}_i = \text{value change}(\tilde{r}_i), i \in \mathcal{E}_c, \\ & |\mathcal{E}_c| \leq \lceil n_c \gamma \rceil, \end{aligned} \quad (2)$$

$\forall c \in [1, C]$. Q_c is the set of fingerprinted records in community c , $s_{ij}^{\text{comm}_c}$ denotes Alice's prior knowledge on the statistical relationship between individuals i and j in community c , $\tilde{s}_{ij}^{\text{comm}_c}$ is the statistical relationship between individuals i and j in community c in $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, whose i th data record is denoted as \tilde{r}_i , and $\hat{s}_{ij}^{\text{comm}_c}$ is such information obtained from $\tilde{\mathbf{R}}(\text{FP}, \text{Dfs}_{\text{row}}, \emptyset)$, whose i th data record is represented as \hat{r}_i . Also, $\text{value change}(\cdot)$ is the function that changes each attribute of \tilde{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. Since (2) is an NP-hard combinatorial search problem [19], we develop a heuristic approach to solve it (refer to our previous work [1] for details). We describe the steps of Dfs_{row} in Algorithm 4.

Algorithm 4: Dfs_{row} : defense against row-wise correlation attack.

Input : Vanilla fingerprinted database, $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$, fingerprinting ratio γ , database owner's prior knowledge on the row-wise correlations \mathcal{S}' and individuals' affiliation to the C communities.

Output: $\tilde{\mathbf{R}}(\text{FP}, \text{Dfs}_{\text{row}}, \emptyset, \cdot)$.

- 1 Obtain $\tilde{\mathcal{S}}$, i.e., the set of pairwise statistical relationships among individuals in each community, from the vanilla fingerprinted database $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$;
 - 2 **forall** $\text{comm}_c, c \in [1, C]$ **do**
 - 3 **forall** non-fingerprinted individual $i \in \text{comm}_c / Q_c$ **do**
 - 4 Calculate

$$e_i = \sum_{j \in \text{comm}_c, j \neq i} |s_{ij}^{\text{comm}_c} - \tilde{s}_{ij}^{\text{comm}_c}|, i \in \text{comm}_c / Q_c;$$
 - 5 Obtain the largest $\lceil n_c \gamma \rceil$ e_i 's, and collect these row index i in set \mathcal{E}_c ;
 - 6 **forall** row index $i \in \mathcal{E}_c$ **do**
 - 7 $\hat{r}_i = \text{value change}(\tilde{r}_i)$; // change the value of each attribute of \tilde{r}_i to the most frequently occurred instance of that attribute in comm_c .
 - 8 Return $\tilde{\mathbf{R}}(\text{FP}, \text{Dfs}_{\text{row}}, \emptyset, \cdot)$.
-

5.3 Integrated Robust Fingerprinting

Although after applying Dfs_{row} , the malicious SP may still identify (and distort) some fingerprinted data records using Atk_{row} , 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 6.2, we validate that Algorithm 4 can successfully mitigate the row-wise correlation attack in a real-world database. Since Dfs_{row} changes less number of entries than Dfs_{col} , database owner will apply Dfs_{row} followed by Dfs_{col} to conduct the integrated mitigation.

6 EVALUATION

Now, we show the vulnerability of the existing fingerprinting schemes against the correlation attacks, evaluate the performance of the proposed mitigation techniques, investigate their impact on database utility, and empirically study the

effect of knowledge asymmetry between the database owner and a malicious SP.

6.1 Experiment Setup

We consider a Census database [20],⁵ which records 14 discrete/categorical attributes of 32561 individuals. As discussed in Section 3, we choose the state-of-the-art schemes developed in [6] and [8] as the vanilla mechanisms, both of which are robust against common attacks (such as random bit flipping, subset, and superset attacks). We use 128-bits fingerprint string ($L = 128$) for them, because when considering N SPs, as long as $L > \ln N$, they can thwart exhaustive search and various types of attacks.

6.2 Vulnerability Against Identified Correlation Attacks

To add fingerprint to the Census 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 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, we encode the instances of the attribute in such a way that instances with close semantic meaning are represented using integers that are also close to each other. Take the “marital-status” attribute as an example, its instances are first mapped to a high dimensional space via the word embedding technique [21]. 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. For example, Figure 4 shows the dendrogram visualizing the hierarchically clustered instances of the “marital-status” attribute. Besides, we use K-means algorithm to group the individuals in the Census database into non-overlapping communities, and according to the Schwarz’s Bayesian inference criterion (BIC) [22], the optimal number of communities is $C = 10$.

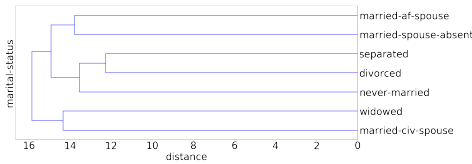


Fig. 4: The dendrogram obtained by clustering instances of “marital-status” attribute in a hierarchical structure.

5. This is a generic relational database. The proposed robust fingerprinting scheme can be applied on any relational databases. For example, when fingerprinting genomic databases, the malicious SP can utilize the correlations determined by Mendel’s law and linkage disequilibrium to compromise the inserted fingerprint bits. In our recent work [10], we have shown that an adaptation of the proposed work can also achieve robustness in genomic database fingerprinting.

6.2.1 Vulnerability of FP_1

We first show the vulnerability of the first vanilla fingerprinting scheme, i.e., FP_1 [6]. 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 on $\tilde{\mathbf{R}}(FP_1, \emptyset, \emptyset)$, the malicious SP generates pirated database $\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \emptyset)$, $\tilde{\mathbf{R}}(\emptyset, \emptyset, \text{Atk}_{\text{col}})$, and $\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \text{Atk}_{\text{col}})$, respectively. We obtain the database utility and fingerprint robustness achieved by these pirated databases and compare them with the database that is only subject to the random flipping attack, i.e., $\tilde{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$.

metrics	utility				robustness	
Database	Acc	P _{col}	P _{row}	P _{cov}	num _{cmp}	r
$\tilde{\mathbf{R}}(FP_1, \emptyset, \emptyset)$	98.5%	95.2%	100%	99.4%	N/A	N/A
$\tilde{\mathbf{R}}(\emptyset, \emptyset, \text{Atk}_{\text{col}})$	73.3%	75.8%	88.6%	93.0%	82	top 91.4%
$\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \emptyset)$	98.1%	90.4%	95.1%	97.2%	78	top 82.9%
$\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \text{Atk}_{\text{col}})$	72.9%	75.0%	88.4%	93.6%	83	top 93.7%
$\tilde{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$	72.9%	67.4%	65.4%	91.2%	4	uniquely

TABLE 2: Database utilities and fingerprint robustness obtained from $\tilde{\mathbf{R}}(FP_1, \emptyset, \emptyset)$ and pirated databases generated by launching various attacks on it.

In Table 2, we show the experimental results. In particular, the cells highlighted in red are the benchmark database utilities obtained using the first vanilla fingerprinting scheme (note that the robustness metrics are not applicable, because $\tilde{\mathbf{R}}(FP_1, \emptyset, \emptyset)$ has not been compromised yet). During Atk_{col} , we set the threshold $\tau_{\text{col}}^{\text{Atk}} = 0.0001$ when comparing with $|J_{p,q}(a, b) - \tilde{J}_{p,q}(a, b)|$.⁶ On the contrary, we choose a large value for $\tau_{\text{row}}^{\text{Atk}}$ and τ_{cov} , 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. We observe that at some cost of the database utility (i.e., decrease of Acc, P_{col}, P_{row}, and P_{cov}), Atk_{col} is able to compromise 82 (out of 128) fingerprint bits and makes the malicious SP only top 91.4% accusable, which suggests that Alice will accuse innocent SPs with a high probability.

In Atk_{row} , we set the threshold $\tau_{\text{row}}^{\text{Atk}} = 0.1$ when comparing with $\sum_{j \neq i}^{n_c} |s_{ij}^{\text{comm}_c} - \tilde{s}_{ij}^{\text{comm}_c}|$. After launching row-wise correlation attack on $\tilde{\mathbf{R}}(FP_1, \emptyset, \emptyset)$, 78 fingerprint bits are distorted at the cost of only 2.9% database utility loss. This makes the malicious SP only rank top 82.9% accusable, and may also cause Alice accuse innocent SP with a high probability. In particular, we have $P_{\text{col}}(\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}}, \emptyset)) = 0.904$, $P_{\text{row}}(\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}}, \emptyset)) = 0.951$, and $P_{\text{cov}}(\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}}, \emptyset)) = 0.972$, which are all close to that of $\tilde{\mathbf{R}}(FP_1, \emptyset, \emptyset)$. Since Atk_{row} can distort sufficient fingerprint bits and cause Alice

6. 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. For example, one can set $\tau_{\text{col}}^{\text{Atk}}$ and $\tau_{\text{col}}^{\text{Dfs}}$ as $\min_{p,q,a,b} \frac{\gamma}{|\mathcal{T}|} \text{freq}_{a,b}^{p,q}$, which is approximately the minimum expected value of absolute difference of the pairwise joint probability before and after fingerprint insertion (here γ is the fingerprinting density, $|\mathcal{T}|$ is the number of attributes, and $\text{freq}_{a,b}^{p,q}$ is the frequency of entries whose attributes p and q take values a and b in the original database).

to accuse innocent SPs with a high probability at a much lower utility loss, we conclude that it is more powerful than Atk_{col} . This suggests that in real-world integrated correlation attacks, the malicious SP can conduct Atk_{row} followed by Atk_{col} to simultaneously distort a large number of fingerprint bits and preserve data utility when generating the pirated database. For example, via the integration of both correlation attacks, i.e., $\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \text{Atk}_{\text{col}})$ a malicious SP can distort 83 fingerprint bits and further reduce the accusable ranking of the malicious SP, which now becomes only the top 93.7% accusable.

To conduct a fair comparison with the conventional random flipping attack, we let the database compromised by Atk_{rnd} (i.e., $\tilde{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$) have the same database utility (in terms of accuracy, i.e., $\text{Acc}(\tilde{\mathbf{R}})$) as $\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \text{Atk}_{\text{col}})$, and then compare the other utility metrics and fingerprint robustness. This can be achieved by letting Atk_{rnd} only change $1 - \text{Acc} = 29.1\%$ entries of $\tilde{\mathbf{R}}(\text{FP}_1, \emptyset, \emptyset)$. The cells highlighted in gray in Table 2 show the corresponding results. In particular, Atk_{rnd} significantly reduces the database utility and only distort 4 (out of 128) fingerprint bits. As a result, Alice uniquely accuses the correct malicious SP for data leakage. As we have shown in our previous work [1], to avoid being uniquely accusable, the malicious SP needs to change at least 80% of the data entries if it applies random bit flipping attack, which inevitably leads to poor utility of the pirated database.

6.2.2 Vulnerability of FP_2

Here, we show the vulnerability of the second vanilla fingerprinting scheme [8]. The experiment setup is the same as Section 6.2.1 and the results are summarized in Table 3. As shown, all column-wise, row-wise, and the integrated correlation attacks can distort a significant portion of the fingerprint bits and reduce the accusable ranking of the malicious SP. In contrast, the random bit flipping attack still causes the malicious SP to be uniquely accusable even though it has same percentage of distorted entries as $\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \text{Atk}_{\text{col}})$.

metrics	utility				robustness	
	Acc	P _{col}	P _{row}	P _{cov}	num _{cmp}	r
Database						
$\tilde{\mathbf{R}}(\text{FP}_2, \emptyset, \emptyset)$	98.6%	97.5%	100%	99.5%	N/A	N/A
$\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{col}})$	77.9%	81.6%	92.3%	95.4%	83	top 93.7%
$\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \emptyset)$	98.3%	94.7%	96.2%	97.8%	78	top 82.9%
$\tilde{\mathbf{R}}(\emptyset, \text{Atk}_{\text{row}}, \text{Atk}_{\text{col}})$	77.2%	81.2%	91.6%	94.7%	84	top 94.8%
$\tilde{\mathbf{R}}(\text{Atk}_{\text{rnd}}, \emptyset, \emptyset)$	77.2%	67.4%	65.5%	91.2%	4	uniquely

TABLE 3: Database utilities and fingerprint robustness obtained from $\tilde{\mathbf{R}}(\text{FP}_2, \emptyset, \emptyset)$ and the pirated databases generated by launching various attacks on it.

Comparing Table 2 with 3, we observe that if FP_2 is adopted as the vanilla fingerprinting scheme, the resulted databases have higher utility than the one that is obtained using FP_1 as the vanilla scheme. This is because FP_2 first groups all the fingerprintable bits into non-overlapping blocks, and then inserts fingerprint block-wise, which leads to lower utility loss. However, FP_2 is more vulnerable to the correlation attacks, as Atk_{row} and Atk_{col} can distort more fingerprint bits by changing less entries in $\tilde{\mathbf{R}}(\text{FP}_2, \emptyset, \emptyset)$.

6.2.3 Empirical Validation of the Attack Strength of Identified Correlation Attacks

Here, we validate the strengths of Atk_{col} and Atk_{row} introduced in Section 4.3. Since FP_2 is more vulnerable to the correlation attacks than FP_1 , and the strength of the attack is a generic metric that is independent of the adopted vanilla fingerprint scheme, we just show the validation results using FP_1 . In particular, given different fingerprinting ratios γ , we investigate the malicious SP's confidence gain, i.e., $G_{\text{col}}(\frac{1}{\gamma}; p, a)$ and $G_{\text{row}}(\frac{1}{\gamma}; \mathbf{r}_i)$ (see Proposition 1 and 2). For this experiment, we let p be the "age" attribute and $a = 3$ (after encoding).

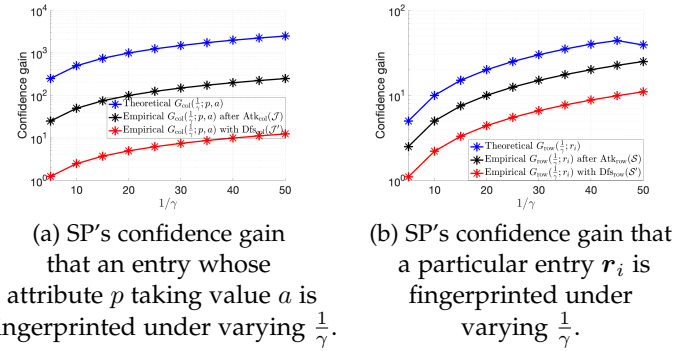


Fig. 5: Evaluation of the confidence gain of fingerprinting under different correlation attacks with and without the corresponding mitigation techniques.

In Figures 5(a) and (b), we show the theoretical values of $G_{\text{col}}(\frac{1}{\gamma}; p, a)$ and $G_{\text{row}}(\frac{1}{\gamma}; \mathbf{r}_i)$ under varying $\frac{1}{\gamma}$, and compare these with the empirical values obtained after the correlation attacks without (black curves) and with (red curves) the corresponding mitigation techniques. In particular, the empirical value of $G_{\text{col}}(\frac{1}{\gamma}; p, a)$ is the frequency of entries whose attribute p takes value a is fingerprinted, and at the same time, is included in the constructed suspicious set \mathcal{P} . The empirical value of $G_{\text{row}}(\frac{1}{\gamma}; \mathbf{r}_i)$ the frequency of rows identified as fingerprinted by $\text{Atk}_{\text{row}}(\mathcal{S})$. We observe that the confidence gain after the mitigation techniques is always smaller than that the one without the mitigation techniques, and it is close to 1 when the fingerprint ratio is high, i.e., $\frac{1}{\gamma}$ is small. The experiment results are consistent with our theoretical findings in Proposition 1 and 2, and this validates the effectiveness of the developed mitigation techniques by showing that the proposed mitigation techniques make the confidence gain of a malicious SP only slightly better than Atk_{rnd} . The gaps between the blue curves (theoretical values) and black curves are due to the applying of inequalities during derivations [1].

6.3 Evaluation of Mitigation Techniques

We have shown that correlation attacks can distort the fingerprint bit-string inserted by both vanilla fingerprinting schemes, and they may make the database owner accuse innocent SPs with high probabilities. In this section, we evaluate the proposed mitigation techniques and show that they can serve as post-processing steps of any vanilla fingerprinting schemes to establish robust fingerprinting schemes.

6.3.1 Making FP_1 Robust against Correlation Attacks

In this experiment, we also assume that Alice has access to S' and J' that are directly computed from R . Thus, we represent her prior knowledge as $S'(R)$ and $J'(R)$. As a result, we have $S' = S$ and $J' = J$.

Performance of Dfs_{col} . As discussed in Section 5.1, 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 Dfs_{col} , we calculate the utility of $\tilde{R}(FP, \emptyset, Dfs_{col})$ by setting $\lambda_p \in \{100, \dots, 1000\}$, $\forall p \in \mathcal{F}$, and show the results in Figure 6. 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, and hence fewer number of entries are modified by Dfs_{col} .

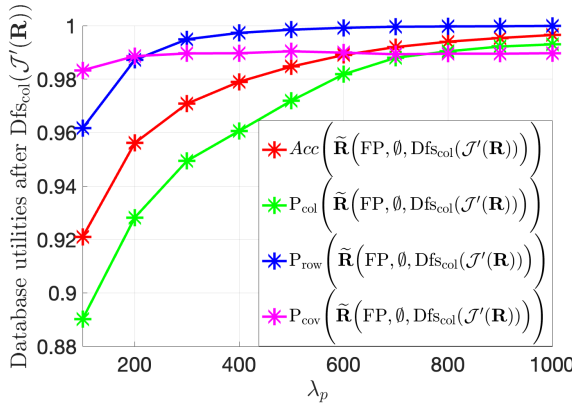


Fig. 6: Utilities of $\tilde{R}(FP, \emptyset, Dfs_{col})$ under varying λ_p .

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 Atk_{col} on $\tilde{R}(FP, \emptyset, Dfs_{col})$. In Table 4(a), we observe that after launching Atk_{col} , the malicious SP can only compromise 18 (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 Atk_{col} on the vanilla fingerprinted database $\tilde{R}(FP, \emptyset, \emptyset)$, the malicious SP can compromise 82 bits and make itself only rank top 91.4% accusable. This suggests that proposed Dfs_{col} significantly mitigates the column-wise correlation attack. It is noteworthy that the column-wise mitigation technique preserves the database utilities, as Acc , P_{col} , P_{row} , and P_{cov} are also close to that of $\tilde{R}(FP_1, \emptyset, \emptyset)$ shown in Table 2.

On the other hand, we observe that Atk_{col} also degrades the utilities of the vanilla fingerprinted database post-processed by Dfs_{col} . In particular, the accuracy drops to 0.76 and the preservation of column-wise correlation drops to 0.67. Thus, we conclude that as a post-processing step, our column-wise correlation mitigation technique provides robust fingerprint against column-wise correlation attack and preserves database utility.

	Acc	P_{col}	P_{row}	P_{cov}	num_{cmp}	r
$\tilde{R}(FP_1, \emptyset, Dfs_{col})$	0.95	0.92	0.97	0.98	N/A	N/A
$\tilde{R}(\emptyset, \emptyset, Atk_{col})$	0.76	0.67	0.89	0.94	18	uniquely accusable

(a) Impact of Dfs_{col} before and after Atk_{col} when FP_1 is chosen as the vanilla scheme.

	Acc	P_{col}	P_{row}	P_{cov}	num_{cmp}	r
$\tilde{R}(FP_1, Dfs_{row}, \emptyset)$	0.97	0.94	0.99	0.99	N/A	N/A
$\tilde{R}(\emptyset, Atk_{row}, \emptyset)$	0.93	0.92	0.94	0.98	13	uniquely accusable

(b) Impact of Dfs_{row} before and after Atk_{row} when FP_1 is chosen as the vanilla scheme.

	Acc	P_{col}	P_{row}	P_{cov}	num_{cmp}	r
$\tilde{R}(FP_1, Dfs_{row}, Dfs_{col})$	0.94	0.91	0.96	0.97	N/A	N/A
$\tilde{R}(\emptyset, Atk_{row}, Atk_{col})$	0.77	0.82	0.86	0.94	4	uniquely accusable

(c) Impact of integrated mitigation before and after integrated correlation attack when FP_1 is chosen as the vanilla scheme.

TABLE 4: Robust fingerprinting achieved by post-processing $\tilde{R}(FP_1, \emptyset, \emptyset)$ using the mitigation techniques.

Performance of Dfs_{row} . In Table 4(b), we evaluate the performance of the robust fingerprinted database against row-wise attack, i.e., $\tilde{R}(FP_1, Dfs_{row}, \emptyset)$, along with the pirated database obtained by launching Atk_{row} on it. Clearly, Dfs_{row} successfully defends against Atk_{row} , since the pirated database only distorts 13 fingerprint bits and makes the malicious SP uniquely accusable. Combining this result with Table 2, we conclude that Dfs_{row} not only mitigates the row-wise correlation attack but it also preserves the database utility.

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 $\tilde{R}(FP, Dfs_{row}, Dfs_{col})$ before and after it is subject to the integrated attack, i.e., Atk_{row} followed by Atk_{col} . We show the results in Table 4(c). 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.

6.3.2 Making FP_2 Robust against Correlation Attacks

In this section, we show that the proposed mitigation techniques can also improve FP_2 and make it robust against the correlation attacks. By adopting the same experiment setup with Section 6.3.1, we show the performance of the mitigation techniques applied after FP_2 in Table 5. As shown, the proposed mitigation techniques improve the robustness of FP_2 : When the malicious SP conducts Atk_{col} , Atk_{row} , or the integration of both, it can only distort 17, 11, and 2 fingerprint bits, respectively, which makes it uniquely accusable by the database owner under all these attacks. This suggests again that our developed mitigation techniques can work as post-processing steps to improve the robustness of any existing fingerprinting scheme against correlation attacks.

6.4 Investigation of Asymmetric Prior Knowledge

Now, we investigate the impact of asymmetric prior knowledge on correlation attacks and mitigation techniques.

	Acc	P_{col}	P_{row}	P_{cov}	num_{cmp}	r
$\tilde{\mathbf{R}}(\mathbf{FP}_2, \emptyset, \mathbf{Dfs}_{col})$	0.96	0.94	0.97	0.98	N/A	N/A
$\tilde{\mathbf{R}}(\emptyset, \emptyset, \mathbf{Atk}_{col})$	0.78	0.83	0.93	0.95	17	uniquely accusable

(a) Impact of \mathbf{Dfs}_{col} before and after \mathbf{Atk}_{col} when \mathbf{FP}_2 is chosen as the vanilla scheme.

	Acc	P_{col}	P_{row}	P_{cov}	num_{cmp}	r
$\tilde{\mathbf{R}}(\mathbf{FP}_2, \mathbf{Dfs}_{row}, \emptyset)$	0.98	0.96	0.99	0.99	N/A	N/A
$\tilde{\mathbf{R}}(\emptyset, \mathbf{Atk}_{row}, \emptyset)$	0.94	0.91	0.95	0.98	11	uniquely accusable

(b) Impact of \mathbf{Dfs}_{row} before and after \mathbf{Atk}_{row} when \mathbf{FP}_2 is chosen as the vanilla scheme.

	Acc	P_{col}	P_{row}	P_{cov}	num_{cmp}	r
$\tilde{\mathbf{R}}(\mathbf{FP}_2, \mathbf{Dfs}_{row}, \mathbf{Dfs}_{col})$	0.96	0.93	0.96	0.97	N/A	N/A
$\tilde{\mathbf{R}}(\emptyset, \mathbf{Atk}_{row}, \mathbf{Atk}_{col})$	0.78	0.82	0.90	0.94	2	uniquely accusable

(c) Impact of integrated mitigation before and after integrated correlation attack when \mathbf{FP}_2 is chosen as the vanilla scheme.

TABLE 5: Robust fingerprinting achieved by post-processing $\tilde{\mathbf{R}}(\mathbf{FP}_2, \emptyset, \emptyset)$ using the mitigation techniques.

6.4.1 Mitigation Techniques with Inaccurate Prior Knowledge vs. Correlation Attacks with Accurate Prior Knowledge

As discussed in Section 4, to the advantage of the malicious SP, we assume that the malicious SP has more accurate or at least equally accurate knowledge (compared to the database owner Alice) about row-wise and column-wise correlations in the database. Here, we further investigate the scenario, in which the malicious SP uses the accurate knowledge, whereas Alice uses inaccurate prior knowledge to implement the proposed mitigation techniques. Without loss of generality, in this section, we conduct experiments using \mathbf{FP}_1 as the vanilla scheme.

To generate inaccurate column- and row-wise correlations, we assume Alice's knowledge on \mathcal{J}' and \mathcal{S}' is computed from \mathbf{R}_{κ} , which is obtained by randomly removing κ percentage of data records from \mathbf{R} . We represent the mitigation techniques using inaccurate knowledge as $\mathbf{Dfs}_{col}(\mathcal{J}'(\mathbf{R}_{\kappa}))$ and $\mathbf{Dfs}_{row}(\mathcal{S}'(\mathbf{R}_{\kappa}))$.

Scenario 1: $\mathbf{Dfs}_{col}(\mathcal{J}'(\mathbf{R}_{\kappa}))$ vs. \mathbf{Atk}_{col} . By varying κ from 1 to 15, we evaluate the fingerprint robustness and database utility before and after $\tilde{\mathbf{R}}(\mathbf{FP}, \emptyset, \mathbf{Dfs}_{col}(\mathcal{J}'(\mathbf{R}_{\kappa})))$ is attacked by \mathbf{Atk}_{col} and show the results in Figure 7. In particular, Figure 7(a) shows that as κ increases, i.e., $\mathcal{J}'(\mathbf{R}_{\kappa})$ becomes less accurate, the number of distorted fingerprint bits by the malicious SP increases from 18 to 35, however, even for the most inaccurate $\mathcal{J}'(\mathbf{R}_{\kappa})$ (when $\kappa = 15$), we observe that the malicious SP cannot distort more than half of the fingerprint bits, and thus it will be uniquely accusable. Figures 7(b)-(e) compare different utility metrics for $\tilde{\mathbf{R}}(\mathbf{FP}, \emptyset, \mathbf{Dfs}_{col}(\mathcal{J}'(\mathbf{R}_{\kappa})))$ and that attacked by \mathbf{Atk}_{col} , i.e., $\tilde{\mathbf{R}}(\emptyset, \emptyset, \mathbf{Atk}_{col}(\mathbf{R}))$. We observe that even with inaccurate column-wise correlations, Alice can still achieve high utilities when generating the fingerprinted database. If the malicious SP attacks using the accurate column-wise correlations, it causes large utility losses in the pirated database, i.e., Acc drops to 0.758 when $\kappa = 15$. This validates that the proposed column-wise mitigation technique is robust even with inaccurate knowledge of the database owner.

Scenario 2: $\mathbf{Dfs}_{row}(\mathcal{S}'(\mathbf{R}_{\kappa}))$ vs. \mathbf{Atk}_{row} . In Figure 8, we show the experiment results when $\tilde{\mathbf{R}}(\mathbf{FP}, \mathbf{Dfs}_{row}(\mathcal{S}'(\mathbf{R}_{\kappa})), \emptyset)$ is subject to \mathbf{Atk}_{row} . Similar to before, as $\mathbf{Dfs}_{row}(\mathcal{S}'(\mathbf{R}_{\kappa}))$

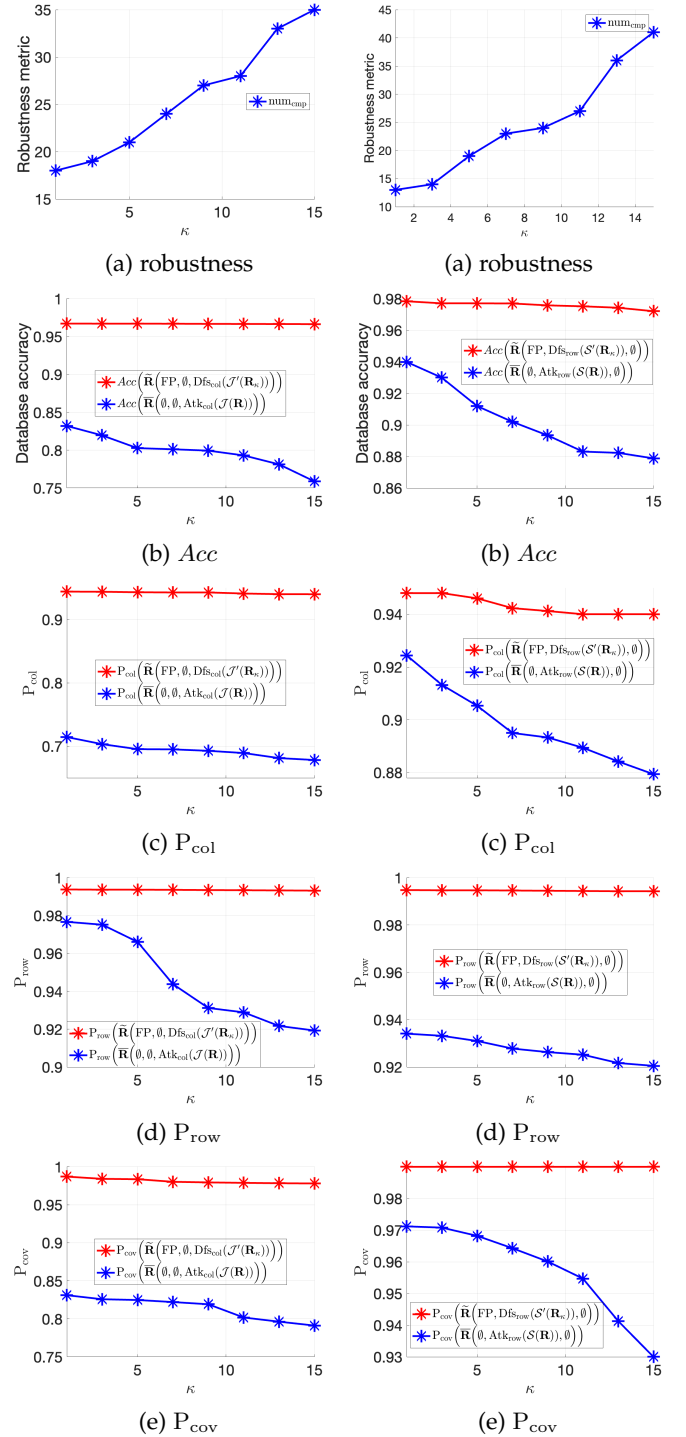


Fig. 7: Evaluation of fingerprint robustness and database utility considering database utility considering $\mathbf{Dfs}_{col}(\mathcal{J}'(\mathbf{R}_{\kappa}))$ and \mathbf{Atk}_{col} . $\mathbf{Dfs}_{row}(\mathcal{S}'(\mathbf{R}_{\kappa}))$ and \mathbf{Atk}_{row} .

becomes less accurate, the malicious SP can distort more fingerprint bits by launching Atk_{row} . However, as shown in Figure 8(a), the malicious SP is still unable to distort more than half of the bits to avoid being identified by Alice. Furthermore, Figures 8(b)-(e) show that row-wise mitigation technique with inaccurate knowledge still preserves the database utility. These results corroborate that the proposed mitigation techniques with inaccurate knowledge can alleviate the correlation attacks even if the malicious SP has access to accurate correlations.

6.4.2 Mitigation Techniques with Accurate Prior Knowledge vs. Correlation Attacks with Inaccurate Prior Knowledge

Here, we consider the opposite case, in which the malicious SP launches correlation attacks using inaccurate prior knowledge, and the database owner uses accurate correlations to perform the mitigation. Note that this case is the most realistic case in real-world applications. We denote the correlation attacks using inaccurate knowledge as $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}_\kappa))$ and $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}_\kappa))$.

Scenario 1: $\text{Dfs}_{\text{col}}(\mathcal{J}'(\mathbf{R}))$ Vs $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}_\kappa))$. In Figure 9, we vary κ from 1 to 15 and evaluate the fingerprint robustness when $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}}(\mathcal{J}'(\mathbf{R})))$ is attacked by $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}_\kappa))$. We observe that the malicious SP can compromise less fingerprint bits as $\mathcal{J}(\mathbf{R}_\kappa)$ becomes less accurate and will make itself uniquely accusable.

Scenario 2: $\text{Dfs}_{\text{row}}(\mathcal{S}'(\mathbf{R}))$ Vs $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}_\kappa))$. In Figure 10, we evaluate the fingerprint robustness when $\tilde{\mathbf{R}}(\text{FP}, \text{Dfs}_{\text{row}}(\mathcal{S}'(\mathbf{R})), \emptyset)$ is attacked by $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}_\kappa))$. As κ increases, num_{cmp} drops from 62 to 39, which also causes the malicious SP to be uniquely accusable.

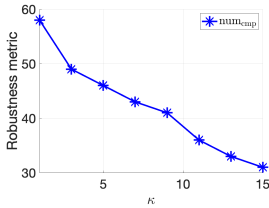


Fig. 9: $\text{Dfs}_{\text{col}}(\mathcal{J}'(\mathbf{R}))$ vs. $\text{Atk}_{\text{col}}(\mathcal{J}(\mathbf{R}_\kappa))$.

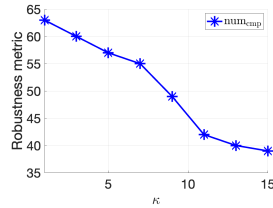


Fig. 10: $\text{Dfs}_{\text{row}}(\mathcal{S}'(\mathbf{R}))$ vs. $\text{Atk}_{\text{row}}(\mathcal{S}(\mathbf{R}_\kappa))$.

From these results, we conclude that although inaccurate knowledge degrades the attack performance, the identified correlation attacks are still significantly more powerful than the conventional random bit flipping attacks in terms of distorting the fingerprint bits.

6.5 Dfs_{col} vs. Atk_{col} with Higher-Order Correlations

In Section 6.2, we have evaluated the mitigation performance of Dfs_{col} against Atk_{col} , which utilizes the joint distributions between pairs of attributes. Here, we validate that the proposed column-wise mitigation technique also alleviates Atk_{col} if a malicious SP uses higher-order correlations in the data. Specifically, we consider the third-order correlations as an example, where the malicious SP computes $|J_{p,q,f}(a,b,c) - \tilde{J}_{p,q,f}(a,b,c)|$ and includes the position tuples $\{i,p\}$, $\{i,q\}$ and $\{i,f\}$ into set \mathcal{P} (see Algorithm 1) if the result exceeds the predetermined threshold

$\tau_{\text{col}}^{\text{Atk}}$. In Figure 11, we show the number of compromised fingerprint bits (num_{cmp}) in the fingerprinted database with and without Dfs_{col} when Atk_{col} uses third-order correlations in the data. We observe that the malicious SP can distort 81 fingerprint bits in the fingerprinted database without Dfs_{col} , i.e., $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$. Thus, Alice may accuse some other innocent SPs. However, the malicious SP can only distort 18 fingerprint bits in the fingerprinted database with Dfs_{col} , i.e., $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$, and as a result the malicious SP will be uniquely accusable.

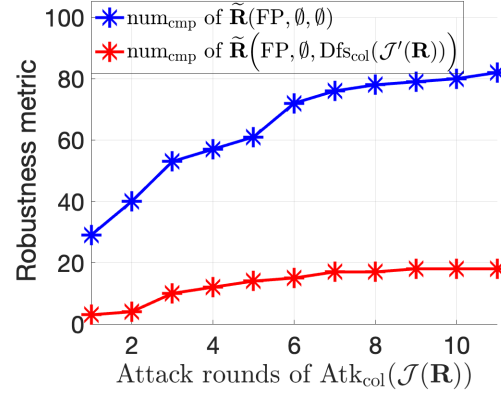


Fig. 11: Number of compromised fingerprint bits when the malicious SP launches Atk_{col} using the third-order correlations on $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \emptyset)$ and $\tilde{\mathbf{R}}(\text{FP}, \emptyset, \text{Dfs}_{\text{col}})$.

7 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 extended our previous work [1] by (i) showing applicability of the proposed techniques to different vanilla fingerprinting schemes, (ii) empirically validating our theoretical findings about the strength of the correlation attacks, (iii) investigating the impact of asymmetric prior knowledge between the mitigation techniques and correlation attacks, and (iv) demonstrating the robustness of the proposed column-wise mitigation technique against higher-order of column-wise correlation attacks.

ACKNOWLEDGEMENT

Research reported in this publication was supported by the National Library of Medicine of the National Institutes of

Health under Award Number R01LM013429 and by the National Science Foundation (NSF) under grant numbers 2050410 and OAC-2112606.

REFERENCES

- [1] T. Ji, E. Yilmaz, E. Ayday, and P. Li, "The curse of correlations for robust fingerprinting of relational databases," in *24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2021)*, 2021.
- [2] E. F. Codd, "A relational model of data for large shared data banks," in *Software pioneers*. Springer, 2002, pp. 263–294.
- [3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE transactions on image processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [4] I. J. Cox, M. L. Miller, J. A. Bloom, and C. Honsinger, *Digital watermarking*. Springer, 2002, vol. 53.
- [5] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures: Steganography and Watermarking: Attacks and Countermeasures*. Springer Science & Business Media, 2001, vol. 1.
- [6] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: Schemes and specialties," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 34–45, 2005.
- [7] F. Guo, J. Wang, and D. Li, "Fingerprinting relational databases," in *Proceedings of the 2006 ACM symposium on Applied computing*, 2006, pp. 487–492.
- [8] S. Liu, S. Wang, R. H. Deng, and W. Shao, "A block oriented fingerprinting scheme in relational database," in *International conference on information security and cryptology*. Springer, 2004, pp. 455–466.
- [9] J. Lafaye, D. Gross-Amblard, C. Constantin, and M. Guerrouani, "Watermill: An optimized fingerprinting system for databases under constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, pp. 532–546, 2008.
- [10] T. Ji, E. Ayday, E. Yilmaz, and P. Li, "Robust fingerprinting of genomic databases," *Bioinformatics*, vol. 38, no. Supplement 1, pp. i143–i152, 06 2022. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btac243>
- [11] R. Agrawal, P. J. Haas, and J. Kiernan, "Watermarking relational data: framework, algorithms and analysis," *The VLDB journal*, vol. 12, no. 2, pp. 157–169, 2003.
- [12] E. Yilmaz and E. Ayday, "Collusion-resilient probabilistic fingerprinting scheme for correlated data," *arXiv preprint arXiv:2001.09555*, 2020.
- [13] E. Ayday, E. Yilmaz, and A. Yilmaz, "Robust optimization-based watermarking scheme for sequential data," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, 2019, pp. 323–336.
- [14] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.
- [15] I. T. Jolliffe, "Springer series in statistics," *Principal component analysis*, vol. 29, 2002.
- [16] M. W. Browne and R. Cudeck, "Alternative ways of assessing model fit," *Sociological methods & research*, vol. 21, no. 2, pp. 230–258, 1992.
- [17] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.
- [18] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in neural information processing systems*, 2013, pp. 2292–2300.
- [19] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.
- [20] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.
- [21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, pp. 3111–3119, 2013.
- [22] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.