An Attack on the Privacy of Sanitized Data That Fuses the Outputs of Multiple Data Miners

Michal Sramka*
Department of Computer Engineering and Maths
Rovira i Virgili University
Av. Paisos Catalans, 26, 43007 Tarragona, Spain
Email: michal@sramka.net

Reihaneh Safavi-Naini and Jörg Denzinger
Department of Computer Science
University of Calgary
2500 University Dr. NW, Calgary AB T2N 1N4, Canada
Email: rei@ucalgary.ca, denzinge@ucalgary.ca

***Abstract*—Data sanitization has been used to restrict re-identification of individuals and disclosure of sensitive information from published data. We propose an attack on the privacy of the published sanitized data that simply fuses outputs of multiple data miners that are applied to the sanitized data. That attack is practical and does not require any background or additional information. We use a number of experiments to show scenarios where an adversary can combine outputs of multiple miners using a simple fusion strategy to increase their success chance of breaching privacy of individuals whose data is stored in the database. The fusion attack provides a powerful method of breaching privacy in the form of partial disclosure, for both anonymized and perturbed data. It also provides an effective way of approximating predictions of the best miner (a miner that provides the best results among all considered miners) when this miner cannot be determined.**

***Keywords*-data privacy; privacy attacks; data mining; fusion;**

I. Introduction

Data sanitization aims to restrict privacy disclosures from published data. Although extensive research exists about sanitization methods and techniques, published sanitized data is still prone to various privacy attacks. Some of these attacks use additional information known by the adversary while others do not require any "background" or "additional" knowledge.

Using data mining over sanitized data is a natural way of learning trends and patterns of the data that can be used for prediction of previously unknown values. Data mining has been already considered for capturing and measuring the usefulness (sometimes also called the utility) of the published sanitized data for end users, such as researchers and analysts. Naturally, it can also be used for predicting values that lead to privacy breaches. For example, data mining can be used to predict those values of sanitized data that help to re-identify an individual. These data mining attacks over sanitized data do not require any background knowledge available to the adversary, although such knowledge can be included in data mining or used elsewhere during the attack. The focus of this paper is on data mining attacks without assuming any background knowledge. We leave the question of using background knowledge [1] during the data mining attacks for future research.

After sanitized data is released, an adversary may use any data miner to attack the privacy and re-identify individuals or make disclosures of private and sensitive values. Many data miners are available that can be used by the adversary. For example, the Weka package [2] contains a collection of data miners that can be used with very little knowledge about data mining and thus can be easily used by non-expert adversaries, such as reporters or noisy kids. Thus, we assume that the adversary is using not one, but a set of several data miners. These multiple data miners are all used over the sanitized data to produce predictions of values that can be a threat to the privacy of the sanitized data. However, multiple data miners produce multiple outputs – predictions that can be contradicting. The adversary using multiple data miners therefore faces the problem of how to use these multiple predictions to his/her advantage. In this paper, we consider combining or "fusing" the outputs of these miners and, in particular, identify scenarios where such an attack is advantageous.

A. Our Contribution

We describe a privacy attack that uses multiple data miners applied to the sanitized data. The attack does not assume any background knowledge. The adversary that launches the attack obtains the outputs (predictions) from multiple miners and uses a fusion strategy to combine the obtained predictions into a single prediction. The predictions are based on the sanitized values. The attack is practical – our instantiation of the attack uses publicly available data mining tools and a fusion strategy based on simple statistics.

We then propose a success measure for an adversary that uses multiple data miners and a fusion attack to breach privacy of individuals' sensitive data. The measure uses a general and flexible definition of utility function for data mining algorithms, and allows us to evaluate effectiveness

* This work was done while the first author was with the Department of Computer Science, University of Calgary, and was supported by Informatics Circle of Research Excellence, Alberta.

of the fusion attack by comparing its score with other types of data mining attacks. The data mining utility function captures the adversary's intentions, by defining interest weights for each record and bonuses and penalties based on errors in the prediction. The weights specify the adversary's interest in individuals or groups of individuals, and the bonuses and penalties specify the adversary's willingness in obtaining exact, partial or even incorrect predictions.

We evaluate the performance of the fusion attack by comparing the utility of the fusion attack with the utilities of an ideal attack, the best miner, and a random guessing attack. An ideal attack is an attack that gives perfect predictions; i.e., it always correctly predict the original values of the sanitized attributes. The "best" miner is the miner that achieves the highest utility among the set of all considered miners. In practice, the adversary does not know which miner has the best prediction (the adversary does not know the unsanitized data). However for the purpose of evaluation of fusion attack we can always find this miner. Finally a random guessing attack randomly chooses the possible values as candidates for the original values.

Our experimental evaluation is performed on the UCI Adult data [3] that is k-anonymized and perturbed using ϵ-differential privacy for various privacy parameters of k and ϵ. k-anonymity [4], [5] generalizes and suppresses data values in order to ensure that an individual is indistinguishable from at least $k-1$ other individuals in the released data. ϵ-differential privacy [6], [7] when used non-interactively for data publishing, adds random noise to data values and provides strong statistical privacy guarantees. Over 40 data miners from the Weka package [2] are used to launch the data mining attack on 5 databases. A simple most frequent value strategy is used for fusion of the resulting predictions. Using specialized mining techniques and more sophisticated fusion strategies would lead to better results. However, in our experiments we assume a non-expert adversary who only uses readily available tools.

Our results demonstrate the effectiveness of even simple fusion strategies in improving the prediction results of the adversary, and hence increases the adversary's chance to breach privacy. The attack results in clear advantage for the adversary in obtaining partial disclosures from anonymized or perturbed data in scenarios that we outline below. We categorize these scenarios from the adversary's advantage viewpoint. The experiments show that for obtaining partial disclosure the fusion attack,

(a) is better than, or closely approximates, the success of the best miner for both anonymized and perturbed data; we note that in general the adversary has no means of identifying this miner;
(b) closely approximates the success of the ideal perfect attack in the case of perturbed data.

For obtaining exact disclosures, the success of the fusion attack in approximating the best miner or the ideal attacker deteriorates with higher privacy guarantees.

B. Related Work

In the context of data sanitization, data mining has traditionally been considered for measuring the usefulness of the released sanitized data. This has been in both perturbation/randomized response methods [8] and in anonymization methods [9]. We use data mining to show possible disclosures from the sanitized data – that is, we show possible privacy breaches resulting from publishing sanitized data.

Combining data from several measurements is referred to in the literature as data fusion, data aggregation, and data consolidation. Fusing multiple homogeneous data items into a single "best guess" value can be seen as an application of descriptive/summary statistics. More generally, multiple homogeneous data can be combined into a single value using aggregation operators [10]. See also [11]–[13].

Fusion in attacking privacy has been considered in a web-based information-fusion attack on anonymized data [14], and in a composition attack [1] that fuses multiple independent anonymized releases about overlapping populations. Fusion techniques have also been considered in privacy-preserving data re-publishing, and for analyzing knowledge merged from sequential releases of data [15], [16]. To the best of our knowledge, fusion of data mining results has not been considered before in data privacy research.

Data sanitization can be used to publish data that contains information about individuals. One approach, coming from statistical databases and interactive randomized response models, is to randomize (for example, add noise to) the values of individual records, and only release these records [8], [17], [18]. These methods have been extensively studied [19]–[23]. The amount of noise is controlled using privacy notions, such as ϵ-differential privacy [6], [7], which we use in our experiments, see Section IV-B. The other common approach, called anonymization, involves releasing records while obscuring the (quasi-)identifying attributes: k-anonymity [4], [5] and related methods [24]–[27] are well-researched [9], [28]–[33]. We use k-anonymity in our experiments, see Section IV-B.

II. The Scenario

A data owner wants to release a database that contains individuals' private data for research and analysis purposes. To protect privacy of individuals in the database, the data owner applies a sanitization algorithm to the database and publishes the sanitized data. The sanitized data can then be accessed by the end users – legitimate data users (analysts) as well as adversaries.

An adversary is an end user of the sanitized data with malicious intents, namely trying to make disclosures and breach privacy. The notions of breaching privacy or making disclosures generally depend on a concrete sanitization method

and may mean, for example, to re-identify individuals or determine relations between sensitive values and individuals.

We assume that the adversary applies a set of data miners on the sanitized data to learn private information of the individuals whose information is stored in the database. The adversary obtains predictions from various miners and faces the following questions: Which single miner provides the best prediction? Is it possible to combine the predictions into a single prediction to obtain a better result? In particular since 'the best' miner cannot be systematically found by only using the sanitized data, is it possible to combine outputs of the miners to obtain a performance close to the best miner?

The adversary can always randomly guess the private values and so a useful combined predictions must perform better than this random strategy.

A. Definitions and Assumptions

A *database* DB is a set of tuples. Each *tuple* $x = (x_1, \ldots, x_r)$ consists of *fields* x_i's defined over some finite domains (categorical data) or infinite domains (numerical data). This concept conforms to the traditional relational database definition. We refer to the database DB as the *original data* and assume it is in possession and control of a *data owner*.

A *sanitization algorithm* $\mathcal{S}$ is a randomized algorithm that transforms the database DB on input in a way that protects the relations between individuals and their sensitive information. The output of the sanitization algorithm is denoted by $DB^\star$, which is also a set of tuples with maybe a different number of fields. We refer to $DB^\star$ as the *sanitized data*, and we assume this data is public – available to *data users* (analysts) and to the adversaries. We denote by $x^\star$ the tuple of $DB^\star$ that corresponds to the tuple $x \in DB$ (assuming it exists after sanitization), and we denote by $x_i^\star$ the field that corresponds to the field x_i of $x \in DB$, again assuming it exists after the sanitization took place.

We consider only the non-interactive sanitization methods and the interactive sanitization methods that are able to operate in a non-interactive manner, for example, the ϵ-differential privacy perturbation [6].

A data miner or simply a *miner* $\mathcal{M}$ is an algorithm that takes the sanitized data $DB^\star$ as the input together with one sanitized tuple $x^\star$. The miner determines trends and patterns in this data and outputs a prediction for the m-th field $x_m^\star$ of the tuple $x^\star$. m is a fixed parameter of the miner $\mathcal{M}$. The adversary possesses a set of miners $M = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$, each predicting the same field m of any input tuple. The *prediction* for the m-th field of the tuple $x^\star$ obtained by the miner $\mathcal{M}_i$ is denoted by p_i, that is, $p_i \leftarrow \mathcal{M}_i(DB^\star, x^\star)$.

III. The Attack Model

The adversary has the set $M = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ of miners and uses them over the sanitized data $DB^\star$ to attack the privacy of the individuals in the data. The adversary performs the mining with all the miners and over all the tuples of the sanitized data. After performing the mining, the adversary ends up with n predictions for the m-th field of each tuple in $DB^\star$. However, the adversary is clueless with regard to which predictions are correct (if any at all) and which single miner provides the best results. In the following, we answer the questions of how to meaningfully fuse the multiple predictions into a single prediction, and whether such a fusion is of any advantage to the adversary.

A. Fusion

We use the notion of a fusion algorithm to model the adversary's strategy in combining the available predictions with respect to the value of $x_m^\star$. A data *fusion algorithm* $\mathcal{F}$ is an algorithm that combines the same type of data (homogeneous data) from multiple sources. It takes the m-th field $x_m^\star$ of the sanitized tuple $x^\star$ and n not-necessarily different predictions $p_1, \ldots, p_n$, combines the predictions with respect to $x_m^\star$ based on some internal fusion strategy, and outputs the combined prediction q. That is,

$$q \leftarrow \mathcal{F}(x_m^\star, p_1, \ldots, p_n) \ .$$

The adversary uses this fusion algorithm $\mathcal{F}$ to obtain a single prediction q. The algorithm itself may include additional information that is available to the adversary, such as the ranking of the miners, the confidence of the miners in predicting the values, or the information about the predicted field. For example, based on the database scheme and the number and quality of the available records, some miners of M could be considered better and therefore the adversary may have a higher confidence in their predictions. Or, using information about the predicted field m, an adversary trying to predict an age value may not consider negative predictions while deciding on the final value q.

The adversary's fusion strategy itself can be based on many known techniques [10], [13] as briefly outlined in Section I-B. The techniques can include statistical as well as non-statistical methods such as summary statistic, voting or mediating strategies, and can be based on different types of rankings and hierarchies. We also note that the adversary can use existing fusion validation methods [10] to asses the quality of the fusion and the reliability of the particular fusion strategy, and to estimate bias and errors. We use the most frequent value technique as an instantiation of the fusion algorithm in Section IV-D.

B. Utility of Fusion

We measure the utility of the adversary's fusion by considering the error of the combined value q with respect to the original value x_m and the sanitized value $x_m^\star$, and we do this for all tuples $x^\star \in DB^\star$.

An *error function* $E(x_m, q, x_m^\star)$ measures the correctness of the fused prediction q with respect to the sanitized value

$x^\star_m$ and the original value x_m. The value of E should be higher as the fused prediction q approximates x_m better than $x^\star_m$, and should be zero or even negative (to indicate penalties) when $x^\star_m$ approximates x_m better than q. In other words, the function value of E should provide a quantitative answer of whether q approximates x_m better than $x^\star_m$.

One natural instantiation of the error function is the concept of "nearness", which measures the distance of a (combined) prediction toward the original value with respect to the sanitized value. A prediction q is $c\%$-*nearer* to the original value x_m, if

$$\delta(q, x_m) \leq \frac{100 - c}{100} \cdot \delta(x^\star_m, x_m) \ ,$$

where δ is a distance function, for example the Euclidean distance function. We say a prediction q is *nearer*, if it is $c\%$-nearer for any $c > 0$, and it is *exact*, if it is 100%-nearer.

The adversary's interest in the individuals represented by the tuples x is modeled by the *weight function* $w(x)$. The higher the value of $w(x)$, the higher the interest of the adversary in the tuple $x^\star$, respectively, x. For example, an adversary may be interested only in obtaining disclosures of famous people. We would model this interest as $w(x) = 1$ or some other high value for the tuples x's that represent the famous people, and $w(x) = 0$ for the remaining tuples.

The *utility* $\mathcal{U}_{\mathsf{fuse}}$ of this adversary's fusion attack is computed as

$$\mathcal{U}_{\mathsf{fuse}}(DB, \mathcal{S}, M, \mathcal{F}) = \sum_{x \in DB} w(x) \cdot E(x_m, q, x^\star_m) \ .$$

Using the fore-mentioned error function E and interest weights w, high utility values of $\mathcal{U}_{\mathsf{fuse}}$ mean the fusion attack was successful, while low values mean the attack failed.

C. Success of Fusion

There are several possibilities to evaluate the adversary's success. Here we consider and present three such possibilities in detail. The first one is to compare the utility of fusion $\mathcal{U}_{\mathsf{fuse}}$ with the hypothetical maximum utility that the fore-mentioned utility function can achieve using an idealized (perfect) prediction algorithm that always returns the original value $q = x_m$. It follows that, by design, the error function E achieves maximum at the combined prediction q that equals the original value x_m. The maximum utility $\mathcal{U}_{\mathsf{ideal}}$ is then computed as

$$\mathcal{U}_{\mathsf{ideal}}(DB, \mathcal{S}) = \sum_{x \in DB} w(x) \cdot E(x_m, x_m, x^\star_m) \ .$$

Comparing $\mathcal{U}_{\mathsf{fuse}}$ with $\mathcal{U}_{\mathsf{ideal}}$ gives us the *success* $\sigma_{\mathsf{ideal}} = \mathcal{U}_{\mathsf{fuse}}/\mathcal{U}_{\mathsf{ideal}}$, which is the fraction that shows how closely the fusion algorithm $\mathcal{F}$ approximates the ideal perfect fusion. Note that $\mathcal{U}_{\mathsf{fuse}} \leq \mathcal{U}_{\mathsf{ideal}}$, and so $\sigma_{\mathsf{ideal}} \leq 1$.

Another evaluation criterion that we consider is to compute the utility values achieved by each and every miner separately and then compare the adversary's utility of fusion $\mathcal{U}_{\mathsf{fuse}}$ with the highest such value. The single "best" miner is the miner that provides the highest utility value among all the miners. Its utility is computed as

$$\mathcal{U}_{\mathsf{best}}(DB, \mathcal{S}, M) = \max_{i \in \{1,\ldots,n\}} \sum_{x \in DB} w(x) \cdot E(x_m, p_i, x^\star_m) \ .$$

The *success* $\sigma_{\mathsf{best}} = \mathcal{U}_{\mathsf{fuse}}/\mathcal{U}_{\mathsf{best}}$ quantifies how better ($\sigma_{\mathsf{best}} \geq 1$) or how worse ($\sigma_{\mathsf{best}} < 1$) is the adversary's fusion attack from using just the predictions obtained by the best miner. Note that the adversary is not aware which is the best miner, and so this success σ_{best} also allows us to determine in which scenarios can be the fusion attack used to approximate the best miner.

Finally, we are interested in how better or worse is the fusion algorithm performing compared to a simple "guessing" adversary. Let $\mathcal{U}_{\mathsf{guess}}$ denote the utility of a miner that makes the predictions uniformly at random. For example, if a sanitized integer is an interval $x^\star_m = [20\text{–}29]$, this "guessing" miner will choose a random integer in [20–29] as a prediction. Let r denote such a random guess. This guessing method works for obtaining exact disclosures, but does not help in obtaining partial disclosures, as there is no "nearer" concept here that would evaluate how a guess from the interval is closer to the original value in that interval. For example, if $x_m = 27$ is sanitized into $x^\star_m = [20\text{–}29]$ and a random guess from this interval is $r = 23$, then it is easy to determine if the guess is correct, but hard to decide if 23 is "closer" to 27 in [20–29]. The utility of the guessing adversary is formally defined as

$$\mathcal{U}_{\mathsf{guess}}(DB, \mathcal{S}) = \sum_{x \in DB} w(x) \cdot E(x_m, r, x^\star_m) \ ,$$

and the *success* $\sigma_{\mathsf{guess}} = \mathcal{U}_{\mathsf{fuse}}/\mathcal{U}_{\mathsf{guess}}$ represents how better ($\sigma_{\mathsf{guess}} \geq 1$) or how worse ($\sigma_{\mathsf{guess}} < 1$) is the fusion algorithm performing compared to this guessing adversary.

IV. Experimental Results

Our experiments show in which scenarios it is of an advantage for an adversary to use data mining and fusion. We provide details of these scenarios and describe the advantages an adversary achieves. We conclude that, in general, employing a fusion algorithm which uses a simple fusion strategy during an attack on sanitized data by multiple data miners is *beneficial for obtaining partial disclosures*. Regarding exact disclosures, the higher the privacy guarantees of sanitized data are, the smaller the benefit of using a fusion attack is. This indicates that the simple fusion strategy alone is often not enough to provide exact disclosures from sanitized data.

The simple fusion attack allows an adversary, in several different scenarios, to

- be more successful than the guessing adversary, i.e., be more successful than an adversary who is randomly

guessing the private values, unless the adversary knows the domains of the original data;
- be able to make exact predictions, although all miners only predicted intervals and therefore could not provide exact values; and
- be better than the single best miner, or at least closely approximate the success of this single best miner, while there is no easy way for the adversary to pinpoint this best miner from the list of miners and their predictions.

The experimental results also demonstrate that, in some situations, the use of a fusion algorithm has minimal impact on the privacy. This is a bad news for the adversaries, but a good news for the data owners who are in charge of protecting the privacy of the data.

The comprehensive experiments were performed on several subsets of the UCI's Adult database. k-anonymity and ϵ-differential privacy, representing anonymization and randomized perturbation, were used to obtain the sanitized data. We simulated an adversary by attacking two attributes in the sanitized data with numerous data miners from the Weka package and using a simple fusion strategy to combine the predictions.

A. Data

We performed our experiments on the *Adult data* from the UCI Machine Learning Repository [3]. We prepared the data as proposed in [9]: we removed the records with unknown values and kept the following attributes: age, work class, education (edu), marital status, occupation, race, gender, native country, and salary. The only two numeric attributes were: *Age* (integers 17–99) and *edu* (integers 1–16). For brevity we do not describe the other domains, but they can be found in [9]. For the evaluation, we created 5 independent databases, each by randomly selecting 20,000 records from the original 45,222 records.

B. Sanitization

We used k-anonymization and ϵ-differential privacy with several choices of the privacy parameters k and ϵ to obtain sanitized data from the 5 databases. Both methods are widely accepted methods for sanitization.

1) k-anonymization.: k-anonymity [4], [5] ensures that an individual is indistinguishable from at least $k-1$ other individuals in the released data. It does it by removing identifying attributes, and splitting the remaining attributes into quasi-identifying attributes (QIAs) (combinations of which can still uniquely identify an individual) and sensitive attributes (those that individuals consider private).

Our sanitization algorithm $\mathcal{S}$ was the greedy k-clustering algorithm [32, Figure 5]. Our QIAs were all the attributes except *salary*, which was considered to be a sensitive attribute. Selecting all but the sensitive attribute as QIAs is common, because it provides the maximum protection against any linking attack [5] with any subset of the QIAs.

The values of the attribute *age* were generalized into intervals of length 5, 10, 20, then 50, and finally suppressed into the interval of length 100. Similarly, *edu* was generalized into intervals [1–3], [4–7], [8–12], [13–16], and suppressed into [1–16]. The taxonomy trees that were used for generalizations of the other (categorical) attributes are omitted for brevity. We performed k-anonymization for $k = 2,\ 10,\ 50$, and 100.

2) ϵ-differential privacy.: The ϵ-differential privacy notion [6], [7] has been originally proposed to protect statistical databases and aggregate data. It is achieved by perturbation of the data (namely by noise addition to numerical attributes), and perturbation is a recognized [8] sanitization method for data publishing.

We used leakage $\epsilon = 0.5,\ 0.1,\ 0.05$, and 0.01. Then the ϵ-differential privacy for the numerical attributes was achieved by a sanitization mechanism $\mathcal{S}$ that added to the original numerical values noise chosen randomly from the Laplace distribution $\mathrm{Lap}(0, \Delta f/\epsilon)$ with the probability density function $h(y) = (\epsilon/2)\cdot\exp(-\epsilon|y|/\Delta f)$, where Δf is the sensitivity [6] of the query function f, and was $\Delta f = 82$ for the attribute *age* and $\Delta f = 16$ for the attribute *edu*.

C. Set of Miners

In practice, the selection of miners is a choice of the adversary. We elected to choose miners from the Weka 3.6.0 package [2] because of their availability and ease of use, and therefore a likelihood that an adversary would do the same. However, the selection of the miners can be influenced by the type of data and the ease of use. Of course, using more miners, refined parameters, pre-processing (such as subsampling and/or discretization), and predicting over more attributes than just *age* and *edu* can lead to different results. From this point of view, our results are dependent on the choice of a database, miners, and a fusion strategy.

1) Numerical.: Our set $M_{\text{numerical}}$ of miners, used to try to predict nearer numerical values, consisted of the following 10 miners from the Weka 3.6.0 package [2]: functions (RBFNetwork), rules (ConjunctiveRule, DecisionTable, M5Rules, ZeroR), trees (DecisionStump, M5P, REPTree), functions (LeastMedSq, LinearRegression).

2) Categorical.: Our set $M_{\text{categorical}}$ of miners used to try to predict nearer categorical values consisted of the following 32 miners: bayes (AODE, AODEsr, BayesNet, HNB, NaiveBayes, NaiveBayesSimple, NaiveBayesUpdateable, WAODE), functions (RBFNetwork, SMO), lazy (IB1, IBk, KStar, LWL), rules (ConjunctiveRule, DTNB, DecisionTable, OneR, Ridor, ZeroR), trees (DecisionStump, Id3, J48, J48graft, REPTree, RandomForest, RandomTree, SimpleCart), misc (HyperPipes, MinMaxExtension, OLM, VFI).

D. Fusion Strategy

We used a simple fusion strategy to simulate the adversary. Even with this simple fusion strategy, we were able

to demonstrate the advantage of using it over just using (multiple) miners without fusion. We performed the experiments on the attributes *age* and *edu*, but here we describe the fusion strategy using only *age*. The same strategy, steps and procedures apply to the attribute *edu*.

1) Numerical.: We simulated the adversary by using the 10 miners from the set $M_{\text{numerical}}$ and obtaining 10 predictions $p_1, \ldots, p_{10}$ for the field $x^\star_{\text{age}}$ of each sanitized tuple $x^\star$. All the obtained predictions and also the sanitized values are real numbers. The original value x_{age} is an integer. Our simple fusion algorithm $q \leftarrow \mathcal{F}_{\text{numerical}}(x^\star_{\text{age}}, p_1, \ldots, p_{10})$ does not take the sanitized value into account. In contrast, we show how to use the sanitized value during fusion while defining a fusion algorithm for categorical values, in the next section. $\mathcal{F}_{\text{numerical}}$ determines the most frequent predictions and returns the average of them as the combined value q. Let most-freq$(p_1, \ldots, p_{10})$ be a function that returns the most frequent values from among the arguments. Then

$$q := \frac{\sum \text{most-freq}(p_1, \ldots, p_{10})}{\#\text{ of most-freq}(p_1, \ldots, p_{10})} \ .$$

To determine whether q is exact or nearer, we used the nearer distance function described in Section III-B.

2) Categorical.: We simulated the adversary by using the 32 miners from the set $M_{\text{categorical}}$ and obtaining 32 predictions $p_1, \ldots, p_{32}$ for the field $x^\star_{\text{age}}$ of each sanitized tuple $x^\star$. The sanitized value $x^\star_{\text{age}}$ in this case is an interval obtained during k-anonymization. Similarly, the miners from $M_{\text{categorical}}$ working over k-anonymized data $DB^\star$ can only predict intervals, and so all p_i's are intervals.

We now describe the fusion algorithm $q \leftarrow \mathcal{F}_{\text{categorical}}(x^\star_{\text{age}}, p_1, \ldots, p_{32})$. The knowledge of the sanitized interval $x^\star_{\text{age}}$ and its structure allows us and a possible adversary to use this information during fusion, unlike the previous case, where $\mathcal{F}_{\text{numerical}}$ does not use the value of $x^\star_{\text{age}}$. The fusion algorithm firstly determines which predictions p_i's can possibly better approximate the sanitized value $x^\star_{\text{age}}$. We do this by determining whether each p_i $(i = 1, \ldots, 32)$ is a proper sub-interval of $x^\star_{\text{age}}$. The algorithm next considers only those predictions that pass this test. In other words, it is possible for the adversary to filter out bad predictions. For example, if the original value $x_{\text{age}} = 43$ has been sanitized to the interval $x^\star_{\text{age}} = [0 - 49]$, then the prediction $p_u = [20 - 29]$ would be considered by the adversary although it is not a "nearer" prediction, but a prediction of $p_v = [60 - 65]$ would be filtered out because it cannot possibly better estimate the value in $x^\star_{\text{age}} = [0 - 49]$.

The algorithm $\mathcal{F}_{\text{categorical}}$ depends on whether exact or nearer predictions are being sought. For exact predictions, frequencies of all integers (ages) from the remaining sub-intervals are computed. The rounded average of the most frequent integers is returned as the fused value q. If there is no prediction, the algorithm returns the rounded average of the integers from the sanitized interval $x^\star_{\text{age}}$ as the value q. Formally, let sub-int$(x^\star_{\text{age}}; p_1, \ldots, p_{32})$ be a function that returns those intervals p_i's that are sub-intervals of $x^\star_{\text{age}} = [x^\star_{min} - x^\star_{max}]$. If sub-int$(x^\star_{\text{age}}; p_1, \ldots, p_{32}) = \emptyset$, then

$$q := \frac{x^\star_{min} + x^\star_{max}}{2} \ ,$$

otherwise

$$q := \frac{\sum \text{most-freq}(\text{sub-int}(x^\star_{\text{age}}; p_1, \ldots, p_{32}))}{\#\text{ of most-freq}(\text{sub-int}(x^\star_{\text{age}}; p_1, \ldots, p_{32}))} \ .$$

Then we use rounding to obtain an integer. The fusion q is exact if $x_{\text{age}} = q$. For nearer predictions, the algorithm returns the set of all ages that are most frequent in all the better approximating predictions. Simply,

$$q := \text{most-freq}(\text{sub-int}(x^\star_{\text{age}}; p_1, \ldots, p_{32})) \ ,$$

and then the fusion q is nearer if the original value $x_{\text{age}} \in q$.

E. Fusion Attacks

1) Scenarios.: We used the following two scenarios that model an adversary's intentions: The adversary is interested in breaching privacy for (A) all the individuals equally, obtaining any partial disclosure, that is, obtaining any nearer prediction; and for (B) all the individuals equally, obtaining exact disclosures, that is, obtaining 100%-nearer predictions.

These scenarios are modeled using the interest weight function $w(x)$ and using the nearness concept as the error function E. In both scenarios, the interest weight function was constant $w(x) = 1$, which represented equal interest in all the tuples. In scenario (A), the error function E returned 1 for all predictions p_i's or the fused value q that were better approximating (were nearer to) the true value x_{age} than the sanitized value $x^\star_{\text{age}}$. In scenario (B), the error function E returned 1 for those predictions p_i's or the fused value q that exactly matched the original value $x^\star_{\text{age}}$. The error functions in both scenarios returned 0 for all other cases. The same functions were used for the attribute *edu*.

2) Results.: Table I shows the obtained results of utility and success of the adversary's fusion attack on two attributes (*age* and *edu*), for the two scenarios (A and B) described above, for two different sanitization algorithms (k-anonymity and ϵ-differential privacy), while using the above mentioned sets of miners $M_{\text{categorical}}$ and $M_{\text{numerical}}$, and the above mentioned fusion algorithms $\mathcal{F}_{\text{categorical}}$ and $\mathcal{F}_{\text{numerical}}$. $\mathcal{U}_{\mathsf{ideal}}$, representing the utility of the perfect ideal attack, is 20,000 in all cases. The other utility values, namely $\mathcal{U}_{\mathsf{best}}$ and $\mathcal{U}_{\mathsf{guess}}$, can be computed from the table if needed. The numbers in Table I represent averages over 5 databases and 5 repetitions of the guessing attack.

3) Discussion – scenario A.: $\sigma_{\mathsf{best}} \geq 96\%$, which means that the fusion attack closely approximates or is better than the best miner for obtaining partial disclosures. Note that the adversary has no easy way of determining the best miner. Our result effectively shows that using the fusion of multiple

attr.	scen.	$\mathcal{S}$an:	$k=2$	$k=10$	$k=50$	$k=100$	$\epsilon=.5$	$\epsilon=.1$	$\epsilon=.05$	$\epsilon=.01$
age	**A**	$\mathcal{U}_{\mathsf{fuse}}$	1,794	4,046	4,635	4,325	19,007	19,104	19,068	19,082
		σ_{ideal}	9%	20%	23%	22%	95%	96%	95%	95%
		σ_{best}	206%	207%	134%	127%	97%	97%	96%	96%
		σ_{guess}	n/a	n/a	n/a	n/a	106%	97%	96%	96%
	B	$\mathcal{U}_{\mathsf{fuse}}$	3,204	1,456	682	585	468	156	75	5
		σ_{ideal}	16%	7%	3%	3%	2%	1%	0%	0%
		σ_{best}	∞	∞	∞	∞	70%	34%	23%	33%
		σ_{guess}	103%	116%	149%	147%	190%	64%	31%	2%
edu	**A**	$\mathcal{U}_{\mathsf{fuse}}$	104	1,784	5,159	5,900	14,650	14,375	13,992	13,725
		σ_{ideal}	1%	9%	26%	30%	73%	72%	70%	69%
		σ_{best}	104%	106%	126%	114%	97%	100%	99%	99%
		σ_{guess}	n/a	n/a	n/a	n/a	91%	73%	71%	69%
	B	$\mathcal{U}_{\mathsf{fuse}}$	1,766	1,520	1,510	1,447	1,383	147	148	54
		σ_{ideal}	9%	8%	8%	7%	7%	1%	1%	0%
		σ_{best}	∞	∞	∞	∞	83%	62%	86%	110%
		σ_{guess}	38%	38%	56%	67%	843%	88%	93%	33%

Table I

ADVERSARY'S SUCCESS IN THE FUSION ATTACK ON ATTRIBUTES *age* AND *edu*, FOR DATA SETS SANITIZED WITH k-ANONYMITY AND ϵ-DIFFERENTIAL PRIVACY, IN TWO SCENARIOS: A (NEARER PREDICTIONS) AND B (EXACT PREDICTIONS). $\mathcal{U}_{\mathsf{fuse}}$ IS THE UTILITY OF FUSION. THE SIGMAS PRESENT HOW BETTER OR WORSE IS THE FUSION ATTACK COMPARED TO: THE PERFECT IDEAL ADVERSARY (σ_{ideal}), THE SINGLE BEST MINER (σ_{best}), AND JUST GUESSING THE ORIGINAL VALUES (σ_{guess}). THE NUMBERS ARE AVERAGES OVER 5 DATABASES AND 5 RUNS OF THE GUESSING ATTACK.

miners' results allows the adversary to achieve better or nearly the same utility as the one of the best miner.

For the ϵ-differential privacy perturbed data, the fusion attack also closely approximates the ideal attack, especially for the attribute *age*, while the approximation is slightly lower for *edu*. And, as k is increased, the fusion attack better approximates the ideal attack for the k-anonymized data.

The guessing attack was not applicable for the k-anonymized data, because of the lack of "nearer" measure for two values (guessed and original) in the sanitized interval. See Section III-C for an example. In the case of the perturbed data, the guessing attack is seemingly better than the fusion attack ($\sigma_{\mathsf{best}} < 100\%$ up to one exception), but this is because knowledge of the original domain was used during the guessing attack, knowledge which is not always available to the adversary.

Overall, an adversary using this fusion attack would have a clear advantage over an adversary who uses just data mining.

4) Discussion – scenario B.: The success of the fusion attack deteriorates as higher privacy is applied (higher k or smaller ϵ). Declining σ_{ideal}, ending at $\leq 7\%$ for the highest privacy, indicates that the sanitization wards off the fusion attack.

For the k-anonymized data, $\sigma_{\mathsf{best}} = \infty$, because the fusion attack predicts numbers and hence can produce exact disclosures, while any miner individually can only predict intervals. For perturbed data, σ_{best} is way below 100% (up to one exception), which means that the fusion attack is much worse than the best miner.

Regarding the guessing attacks, we observe the same patterns for perturbed data as in the scenario A. The advantage of the guessing attack over the fusion attack ($\sigma_{\mathsf{guess}} < 100\%$) can be explained by the fact that the guessing adversary was given knowledge of the original domains. Thus the guessing adversary worked over the original domain, while the fusion attack worked over the sanitized domain – perturbed data values that were of several orders of magnitude larger.

The different patterns ($\sigma_{\mathsf{guess}} \geq 103\%$ for *age* and $\sigma_{\mathsf{guess}} \leq 67\%$ for *edu*) of the guessing attack for anonymized data are likely due to the different domain sizes and different generalization hierarchies of *age* and *edu*. Nevertheless, for both attributes, the fusion attack is getting better versus the guessing attack as k is increased.

In summary, the fusion attack is lacking usefulness in scenario B, up to the fact that exact values can be obtained from the fusion attack, while no individual miner was able to predict them.

V. CONCLUSIONS

The fusion attack is better or closely approximates the attack using the single best miner (that is unknown to the adversary), and it also successfully approximates the ideal perfect attack. The adversary would clearly gain in using the fusion attack for obtaining partial disclosures. The fusion attack turned out not to be useful for prediction of exact values, i.e., for prediction of values leading to exact disclosures. In other words, both sanitization techniques are immune to the simple fusion attack for exact disclosures, which is good news for data owners performing sanitization. We believe that if an attack using multiple data miners is launched to obtain exact disclosures, it is better to develop (if possible) a method to identify the best miner – the miner that provides the highest utility from all the used miners – than to use a fusion. This strategy as well as the methodology to identify the best miner remains an open research problem.

There are several other possibilities to extend our work. While we used a simple fusion strategy that returned the most frequent value, there can be more advanced fusion strategies that include in calculations the ranking of the data miners and the confidence of the data miners in predictions, as well as the confidence of the data fusion algorithm in the fused value. It is also possible to consider voting or mediating strategies on the fused value, based on different voting hierarchies (equal, distributed, centralized).

REFERENCES

[1] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," in *KDD 2008*.

[2] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[3] A. Asuncion and D. J. Newman, "UCI Machine Learning Repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[4] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," SRI Computer Science Laboratory, Tech. Rep. SRI-CSL-98-04, 1998.

[5] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int J Uncertainty, Fuzziness and Knowl-based Syst*, vol. 10, no. 5, pp. 557–570, 2002.

[6] C. Dwork, "Differential Privacy," in *ICALP 2006*.

[7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *TCC 2006*.

[8] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," in *SIGMOD 2000*.

[9] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *KDD 2002*.

[10] V. Torra and Y. Narukawa, *Modeling Decisions: Information Fusion and Aggregation Operators*. Springer, 2007.

[11] V. Torra, Ed., *Information Fusion in Data Mining*. Springer, 2003.

[12] J. Bleiholder and F. Naumann, "Data fusion," *ACM Comput Surv*, vol. 41, no. 1, pp. 1–41, 2008.

[13] I. R. Goodman, R. P. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*. Kluwer, 1997.

[14] S. R. Ganta and R. Acharya, "On breaching enterprise data privacy through adversarial information fusion," in *ICDEW-IIMAS 2008*.

[15] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure Anonymization for Incremental Datasets," in *SDM 2006*.

[16] G. Wang, Z. Zhu, W. Du, and Z. Teng, "Inference Analysis in Privacy-Preserving Data Re-publishing," in *ICDM 2008*.

[17] N. A. Adam and J. C. Wortman, "Security-control methods for statistical databases," *ACM Comput Surv*, vol. 21, no. 4, pp. 515–556, 1989.

[18] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *PODS 2001*.

[19] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *PODS 2003*.

[20] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *SIGMOD Record*, vol. 33, no. 1, 2004.

[21] L. Liu, M. Kantarcioglu, and B. M. Thuraisingham, "The applicability of the perturbation based privacy preserving data mining for real-world data," *Data Knowl. Eng.*, vol. 65, no. 1, pp. 5–21, 2008.

[22] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *PODS 2003*.

[23] C. Dwork and S. Yekhanin, "New Efficient Attacks on Statistical Disclosure Control Mechanisms," in *CRYPTO 2008*.

[24] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ-diversity: Privacy beyond k-anonymity," in *ICDE 2006*.

[25] T. M. Truta and B. Vinay, "Privacy Protection: p-Sensitive k-Anonymity Property," in *PDM 2006*.

[26] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing," in *KDD 2006*.

[27] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity," in *ICDE 2007*.

[28] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain K-Anonymity," in *SIGMOD 2005*.

[29] V. Ciriani, S. D. C. di Vimercati, S. Foresti, and P. Samarati, "k-Anonymity," in *Secure Data Management in Decentralized Systems*. Springer, 2007.

[30] M. E. Nergiz and C. Clifton, "Thoughts on k-Anonymization," in *PDM 2006*.

[31] M. E. Nergiz, C. Clifton, and A. E. Nergiz, "MultiRelational k-Anonymity," in *ICDE 2007*.

[32] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k -Anonymization Using Clustering Techniques," in *DASFAA 2007*.

[33] J. Domingo-Ferrer and V. Torra, "A Critique of k-Anonymity and Some of Its Enhancements," in *ARES 2008*.