Minimizing Minimality and Maximizing Utility: Analyzing Method-based attacks on Anonymized Data

Graham Cormode, Divesh Srivastava
AT&T Labs–Research
{graham,divesh}@research.att.com

Ninghui Li, Tiancheng Li
Purdue University
{ninghui,li83}@cs.purdue.edu

ABSTRACT

The principle of *anonymization* for data sharing has become a very popular paradigm for the preservation of privacy of the data subjects. Since the introduction of k-anonymity, dozens of methods and enhanced privacy definitions have been proposed. However, over-eager attempts to minimize the information lost by the anonymization potentially allow private information to be inferred. Proof-of-concept of this "minimality attack" has been demonstrated for a variety of algorithms and definitions [16].

In this paper, we provide a comprehensive analysis and study of this attack, and demonstrate that with care its effect can be almost entirely countered. The attack allows an adversary to increase his (probabilistic) belief in certain facts about individuals over the data. We show that (a) a large class of algorithms are not affected by this attack, (b) for a class of algorithms that have a "symmetric" property, the attacker's belief increases by at most a small constant, and (c) even for an algorithm chosen to be highly susceptible to the attack, the attacker's belief when using the attack increases by at most a small constant factor. We also provide a series of experiments that show in all these cases that the confidence about the sensitive value of any individual remains low in practice, while the published data is still useful for its intended purpose. From this, we conclude that the impact of such method-based attacks can be minimized.

1. INTRODUCTION

Intuitively, *anonymization* is the problem of releasing a version of a data set so that researchers can analyze the data and extract useful information from it. However, the raw data usually contains sensitive information about the individual data subjects. The goal of anonymization is to apply some masking operations to the data set to protect the privacy of the individuals in the data, while ensuring that it remains useful to researchers. This approach allows anonymized data sets to be "published" and shared with others: in some cases, the publication may be shared indiscriminately (to allow scientific study of the data by anyone); in other cases, the published data may still be shared only among trusted individuals (as in sharing customer data across departments within a company). Within the last decade, a large number of anonymization algorithms [1, 4, 8, 9, 21] have been described. To guarantee that individual privacy is protected, the algorithms ensure that various requirements hold over the published data, such as ℓ-diversity [12], (α, k)-anonymity [18], and t-closeness [10]. Recent tutorials provide more background on existing models and techniques [3, 5].

As methods are proposed, researchers study them and ask: are there vulnerabilities? Under what circumstances is it possible for the privacy of an individual to be broken? This proceeds in the vein of security research: by identifying flaws and weaknesses, new methods are designed which are not susceptible; alternatively, it may be demonstrated that any potential attacker would need resources and information that exceed the limits of plausibility, so it is safe to proceed with certain parameters.

In this spirit, we study a class of attacks based on the principle of *minimality* [15, 22]. These attacks assume that the anonymization algorithm and its parameters are known: a reasonable assumption, in line with similar assumptions made in the security world. Many anonymization methods publish a data set which implicitly encodes a number of possible worlds, each of which is a candidate for being the original input. It is assumed that each candidate is feasible, and by enforcing properties of these sets of possible worlds, the algorithm concludes that the attacker's ability to infer any facts about the original data is limited. Most algorithms aim to minimize the information loss; specifically, the algorithms never generalize or distort the data more than necessary to achieve the privacy requirement. However, the minimization allows a "minimality attack" to argue that some of these candidates are infeasible: had the algorithm been executed on these inputs, the output would have been different. Hence, by ruling out candidates, the adversary's beliefs about the input can violate the claimed privacy requirements. For instance, the simple ℓ-diversity requirement claims that the adversary cannot associate any sensitive value to any individual with confidence greater than $1/\ell$, but under minimality attack, this confidence may become larger than $1/\ell$.

This attack was proposed by Wong *et al.* in 2007 [15], where several algorithms enforcing particular privacy guarantees were shown to be susceptible to this attack. Indeed, in some cases, an attacker could conclude some supposedly hidden facts with certainty! We present examples in subsequent sections. This poses troubling questions: are all anonymization algorithms susceptible to this attack? Is every method at risk of revealing private facts with certainty?

This paper provides the first detailed analysis of this style of attack. We conclude that many algorithms are *not* susceptible to this attack, or are only marginally affected. We also study algorithms which are chosen to be highly susceptible to the attack, and show that the attacker's knowledge, expressed as a probability, may only increase by a small constant factor. As such, the minimality attack remains a concern, but its impact can often be mitigated, and made

Proceedings of the VLDB Endowment, Vol. 3, No. 1
Copyright 2010 VLDB Endowment 2150-8097/10/09... $ 10.00.

effectively zero, while still retaining utility in the anonymized data.

These results do not contradict the initial observations of [15]. Our goal is not to refute prior work, but rather to refine our understanding of this attack. The examples presented in [15] consist of small tables and small privacy parameters; there, the attack leads to total revelation of sensitive values. However, our analysis shows that for larger examples, and larger parameters, the effect is less dramatic. We identify properties of algorithms which make them particularly susceptible to this attack. It is possible to make minor modifications to some algorithms to make them less susceptible.

Contributions and Organization. Background on anonymization algorithms, privacy goals, and the nature of the minimality attack is presented in Section 2. Our contributions are as follows:

We identify three properties of algorithms that make them more susceptible to this attack: being deterministic in operation, making asymmetric grouping choices, and jointly considering the identifying and sensitive attributes. To elaborate on these criteria, we consider algorithms where these properties do not hold, and show that they are not susceptible to the attack. We show that algorithms which consider only identifying attributes or sensitive attributes, such as algorithms which focus on providing k-anonymity, or the Anatomy algorithm [19] are safe from this attack. We also show that algorithms that have a high degree of symmetry in their grouping choices are virtually invulnerable by describing a class of "even-split" algorithms that include a "symmetric" version of the Mondrian algorithm [9] (Section 3).

Next, we study the limits of the minimality attack by introducing a "Greedy Grouping" algorithm which is (deliberately) highly *susceptible* to the attack: it is deterministic, asymmetric and considers identifying and sensitive attributes together to meet privacy requirements. This algorithm is vulnerable to the minimality attack; an attacker can have a confidence of sensitive values larger than $1/\ell$ even when the table satisfies ℓ-diversity. Yet, via combinatorial analysis, we show that this confidence cannot grow too large. No matter what the input data, or value of ℓ, we show that the confidence after applying the minimality attack is at most e/ℓ, i.e. less than 3 times the intended value. We choose to analyze the greedy grouping algorithm in this paper because it is the *most vulnerable* algorithm that we have found, it demonstrates the behaviors of minimality attacks, and it is amenable to analysis. We expect that most other algorithms are *less vulnerable* to minimality attacks. We also propose a randomized variant of the grouping algorithm, which limits the attacker's confidence further (Section 4).

To illustrate the practical impact of these insights, we experimentally evaluate the methods discussed. We compute the number of tuples susceptible to minimality attack under the (deliberately bad) greedy grouping algorithm and show that the increase in confidence is in line with the analysis. We show that adding appropriate randomization is sufficient to eliminate any increase in confidence above the $1/\ell$ baseline, with negligible change in measured utility. Finally, we study the accuracy of answering a broad query workload on anonymized data, and show that the methods we have proposed achieve tangible utility gains over methods which do not try to improve utility while offering equivalent privacy properties, due to our theoretical and empirical privacy analysis (Section 5).

2. ANONYMIZATION AND MINIMALITY

The pioneering work of Samarati and Sweeney [14] was the first to alert the database community to the problems of releasing data which was supposedly stripped of identifying information, but which was vulnerable to re-identification. In their model, there is a table of data where each row contains information about an individual and each column contains an attribute; some attributes are sensitive. The goal is for the data owner to produce a modified version of the data which can be released, so that the data can be usefully studied, but no individual can be closely associated with their original sensitive values. A natural first step removes identifying information from the data, such as name or social security number. However, other parts of the data may still distinguish individuals: the non-sensitive attributes containing demographic and other information may, when taken together, uniquely identify an individual. The canonical example of health care data has patient information, attributes, and details of the disease(s) that they suffer. In this case, the "quasi-identifiers" (QI) which can identify an individual are attributes like gender, height, and postal code. With enough such attributes, many rows may be unique. Partial knowledge of a particular individual in the data could identify their record, and so recover the associated "sensitive attribute" (SA).

2.1 Anonymization Tools

The database community has expended much effort in studying this model of anonymization, and in proposing methods to limit the ability of the attacker to make such inferences. The general approach is to mask the association between any particular QI and SA, via techniques of generalization, suppression and permutation:

Generalization and Suppression reduce the specificity of attributes, within a generalization hierarchy. Dates can be generalized to month or year granularity; numeric values can be coarsened to ranges; categorical values can be replaced with (implicit) sets, such as generalizing a town to its containing state or country; and so on. The complete *suppression* of a value can be thought of as "full generalization" to a generalized value "*" meaning all possible values.

Permutation arranges rows of the data into groups, and publishes the full set of QIs and SAs in each group, but withholds the exact mapping between them. Implicitly, there is a bijection between the two sets for each group, but no further information about the mapping is published.

Generalization is usually used to generate multiple rows which have the same set of (generalized) quasi-identifiers. Whichever "recoding" method is used, most anonymization algorithms implicitly partition the input data into *groups*, and then apply recoding to make individuals in these groups indistinguishable. Indeed, it is possible to first choose a grouping and only then choose which recoding method to apply afterwards. Most existing algorithms can be thought of as such *group-and-recode* methods. Formally, let T be the original dataset so that a tuple $t \in T$ can be represented as $t = (t[QI], t[SA])$ where $t[QI]$ is the QI value of t and $t[SA]$ is the SA value of t. Then:

DEFINITION 1 (GROUP-AND-RECODE METHOD). *Given a dataset T, a* grouping *of T partitions T into m* groups *$\{G_1, G_2, \cdots, G_m\}$, so that $\cup_{i=1}^{m} G_i = T$ and $\forall 1 \leq i_1 \neq i_2 \leq m$, $G_{i_1} \cap G_{i_2} = \emptyset$. A* recoding *of T applies a method such as generalization or permutation to each tuple t_i to generate t_i', with the intent of masking the mapping from QI to SA values within each group.*

In addition, we say a recoding is **SA-Intact** if $\forall i : t_i'[SA] = t_i[SA]$. In other words, it keeps the SA attribute intact and only modifies the QI attributes. In this paper, we consider SA-Intact group-and-recode methods (for arbitrary SAs i.e. categoric or numeric). This is not a limitation, since the vast majority of tabular anonymization methods (including all those considered in [15]) are in this category. Our analysis is independent of the final choice of recoding, so it applies to *all* anonymization methods in this class: the attack and our study focus on the grouping only.

2.2 Privacy Guarantees

We describe some privacy guarantees that have been commonly used. All view the anonymized data as a collection of groups.

k**-anonymization** [14] requires each group to have at least k members, to give "safety in numbers". It means that over the possible worlds, there should be only a $1/k$ probability that a particular individual in the group is matched with a particular instance of a sensitive attribute in the group. However, if the same SA value occurs many times in the same group, the overall likelihood of any individual in the group having that value is higher than $1/k$, leading to more stringent requirements being posed [13, Footnote 2].

(simple) ℓ**-diversity** [12] tries to overcome this limitation by insisting that in each group no distinct SA value occurs more than a $1/\ell$ fraction of the time. This special case of the definition proposed by [12] is equivalent to the (α, k) definition due to [18] with $\alpha = 1/\ell$. In some cases, only certain values of the SA are deemed "sensitive", and only these values must obey the $1/\ell$ fraction. An extreme case is "binary" ℓ-diversity, when the sensitive attribute encodes either positive or negative (e.g. whether or not someone has a particular disease), and each group should have at most a $1/\ell$ fraction of positives. Negative values are not considered to be sensitive. This case is considered extensively in [15]. Variants based on the entropy of the SA distribution behave similarly in practice.

t**-closeness** [10] captures the insight that the earlier methods are ultimately concerned with ensuring that the SAs in each group are not significantly divergent from the overall distribution of the SAs. So t-closeness requires that the statistical difference between the SA distribution in a group and the global SA distribution should not be more than some parameter t. The difference may be measured with a metric like Earth Mover's Distance (EMD) which incorporates the similarity of distinct SA values.

2.3 Anonymization Methods

Many SA-Intact group-and-recode methods for anonymizing data have been proposed [3, 5]. Here, we present examples to focus our study. Further background and context is provided in Appendix A.

Mondrian [9] treats the d QIs for each row in the table as defining a point in d-dimensional space. Groups are formed by hierarchically partitioning this space. Starting from the whole dataset as a single group, the algorithm recursively picks a dimension to split the current group along, such as the widest dimension. The median along that dimension of the points in the group is used a divider, so approximately half the items fall in each new subgroup. In the *strict partitioning* case, all points from the parent group on one side of the divider are placed in one (sub)group and the rest in the other; in the *relaxed partitioning* case, points lying on the median are assigned so subgroup sizes differ by at most 1. A recursive branch halts when a group cannot be divided to achieve the privacy guarantee. [9] achieved k-anonymity by halting group splitting when it contains k to $2k-1$ points. Subsequent adaptations adopt different conditions, such as requiring ℓ-diversity of each proposed split.

Anatomy [19] chooses groups that meet the ℓ diversity measure. It forms groups by repeatedly identifying the current top-ℓ (i.e. most frequent) SA values among the ungrouped rows. For each top-ℓ value, it picks an ungrouped row with this value at random, and forms the ℓ picked rows into a new group. The process terminates when there are fewer than ℓ remaining ungrouped rows, and these are assigned to existing groups while maintaining ℓ-diversity. Clearly each group meets the privacy guarantee; it is proved in [19] that provided the global distribution is ℓ-diverse, the remaining ungrouped rows will always have at least ℓ distinct values.

Greedy Grouping (GG) is a simple anonymization algorithm that we introduce here to exemplify the minimality attack and to study its power. GG first sorts the data based on the quasi-identifiers in some fashion, chosen to make similar QIs adjacent in the ordering. The sort can be as simple as choosing a prioritization of the attributes, and doing a lexicographic sort; or more complex, such as some clustering. Starting from the first ungrouped row, the algorithm keeps adding rows to the current group until the resulting group meets a given privacy requirement (such as ℓ-diversity), at which point the group is complete, and a new group is started[1]. GG abstracts the algorithms studied in the work of Wong et al. [15].

2.4 Minimality Attacks

Next we explain the rationale of minimality attack and show some examples. Given a desired privacy requirement, this requirement determines the sizes of the resulting groups, and the distribution of the sensitive values within the group. The *utility* of the resulting anonymized data then depends on how much information remains: clearly, if a lot of generalization has been performed on the data, then there is little information present in the published data. One anonymization paradigm is to try to perform the minimal amount of masking needed to provide a desired privacy goal, and hence minimize the "loss" of utility.

This minimization is precisely what the minimality attack aims to exploit. That is, the attack proceeds by using knowledge of the algorithm to infer properties of the original data, and hence of individuals in the data. This can be understood probabilistically. There are many possible unanonymized tables consistent with an anonymized table. Without further information, the statistically best strategy is to treat each of these "possible worlds" as equally likely. Minimality attacks, and other "method-based attacks", use knowledge of the algorithm to eliminate (or reduce the likelihood of) some of these possible worlds: essentially, they can say "running the algorithm on this input would not give the same output, therefore it could not be the original data". The elimination of possibilities can raise the attacker's belief in certain events (e.g. some individual having a particular disease), which may break the desired privacy requirements. Examples of the attack are shown in Appendix B and [15]: these show the vulnerability of GG in particular to minimality attack for a variety of privacy guarantees.

These examples suggest that the minimality attack can be very powerful: for some entities, the anonymization could be completely undone. Are many anonymization algorithms hopelessly vulnerable to this attack? In fact, through our subsequent analysis, we will show that this is *not* the case, and that we can quantify the (limits of) the strength of the minimality attack. In the examples in Appendix B, the damage comes in part because the chosen parameter $\ell = 2$ was so small. In more realistic examples, larger values of ℓ are used. There, the attack results not in complete revelation of sensitive values, but rather in elevated probabilities of certain values, which we are able to bound mathematically and empirically.

Formally, in the minimality attack, the attacker is pessimistically assumed to have knowledge of (1) the QI values of all tuples in the data, (2) the anonymization method used $\mathcal{A}$ (including the parameters), and (3) the anonymized data $\mathcal{D}$. The goal of the adversary is to infer SA values for a QI value, and the effectiveness of the attack is measured based on this ability.

DEFINITION 2 (MINIMALITY ATTACK). *The minimality attack occurs when conditioning on $\mathcal{A}$ increases the posterior belief in a particular QI value being associated with a particular SA value, i.e.* $\Pr[t[SA] = s|\mathcal{A}, \mathcal{D}] > \Pr[t[SA] = s|\mathcal{D}]$.

[1] For simplicity, we choose to ignore any tuples which remain after this process, which only makes the GG more vulnerable to attack.

3. ANALYSIS OF MINIMALITY ATTACK

Studying the minimality attack in different settings, we abstract three properties of methods which are vulnerable to this attack:

1) *Deterministic Behavior.* The action of the anonymization algorithm is primarily deterministic, which allows the attacker to work back from the published data and reason about which inputs were possible, as in the examples of Appendix B.

2) *Asymmetric Group Choices.* A key step in the attacker's inference is often to look at a group, and reason that the decision to merge several smaller groups (or not to split a larger group) was much more likely to have been caused due to the violation of a diversity constraint in a particular group. For the example in Figure 6(b), it was argued that the larger group could not have violated diversity no matter how the sensitive values were allocated, so this violation must have taken place in the smaller group.

3) *Consideration of QIs and SAs together.* The goal of publishing the data is typically to separate the QIs and SAs so that the attacker is unable to restore the exact mapping between any of them with high confidence. Algorithms which consider these values together potentially leak information about the original mapping by their choices in what to group together.

3.1 SA-only methods

First, we study the impact that the minimality attack can have on methods which choose a grouping of the rows of the data as a function of the sensitive attributes *only*. Formally,

DEFINITION 3 (SA-ONLY METHOD). *An anonymization method is* SA-only *iff given two datasets D and D' with $|D| = |D'|$ that have the same SA values for each tuple, the method (with the same random choices) outputs the same grouping for D and D'.*

CLAIM 4. *Methods which choose an SA-Intact grouping based on sensitive attributes alone are safe from the minimality attack.*

This and all other proofs are deferred to the Appendix. To understand this claim, consider such a method: the Anatomy algorithm as described in Section 2 forms the ungrouped rows into groups based on their sensitive attributes alone, ensure that there is a diverse mix of SA values in each group. In particular, among rows which share the same SA value, the algorithm picks one at random. So there is no "minimizing" at work here, and as a result, there is nothing for the attacker to rule out.

A second example is the permutation based method of Koudas et al. [7]: this method forms groupings of data with a single numeric SA, and tries to ensure that all rows in the group have similar SA values, but not too narrow a range. Here, there is some attempt at minimizing, to ensure that the range of the SAs is not too much larger than the minimum required. However, the minimality attack does not apply here either, since the choice is over the SAs only, and the same argument about interchanging the QIs above still applies.

Given this observation, one might ask, why not simply stick to algorithms such as Anatomy which are invulnerable to minimality attack? The reason is that in many settings, the utility of anonymized data is limited. In particular, the Anatomy approach alone, which makes no effort to provide utility sometimes yields unsatisfactory results. There is no correlation between QI values that are chosen to form a group, which leaves more uncertainty for query answering. In contrast, the GG algorithm gives groups with QIs that are similar to each other under the initial sort ordering. Thus a query which, say, selects a subset of rows based on QI, is likely to select many groups in their entirety. This reduces uncertainty in the query answer, making such groupings much more useful.

SA-only methods can improve the utility of the published data by dividing large groups into smaller ones. A method which performs an initial grouping based on both QI and SA information may result in some large groups to ensure a guarantee such as ℓ-diversity. Executing an SA-only method on these large groups can guarantee that the privacy guarantee is upheld, and that the resulting groups are as small as possible. For example, provided a group is ℓ-diverse, executing the Anatomy algorithm on the group partitions it further in groups of size (approximately) ℓ, each of which is ℓ-diverse. We refer to this technique as "Anatomization of large groups".

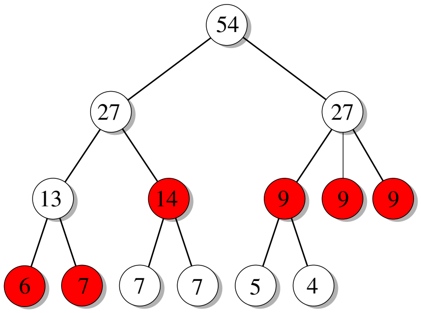


Figure 1: Example even-split partitioning

This discussion of SA-only methods applies equally to the case of QI-only methods—that is, methods which only inspect the QI attributes to determine the grouping. These are not susceptible to minimality attacks for the same reasons: we can permute the SAs anyhow and still achieve the same grouping. However, such methods can only give weak privacy guarantees such as k-anonymity (which depend only on the QIs)—they cannot guarantee ℓ-diversity or t-closeness, since that would require inspecting the SAs. As such, they may be of less interest.

3.2 Symmetric Methods

The two examples of the minimality attack in Appendix B both relied on an *asymmetry* in the group formation. For greedy grouping, the lexicographical ordering of the data meant that the presence of a larger group could be attributed to the *first* group that was considered; for Mondrian, the strict partitioning resulted in asymmetric group sizes, so only one of them could have prevented the split into smaller groups. Next, we show that by enforcing stronger symmetry conditions on the set of groups considered by an anonymization algorithm, it is possible to effectively prevent minimality attacks.

We define an *even-split algorithm* as one that considers a set of possible groupings over a given data set based on recursively partitioning the data into groups based on properties of the QIs. The requirement on the group sizes is that all child groups of any group must differ in size by at most 1 item. The algorithm searches over this space of possible groupings, and outputs a partitioning of the input data into groups from this hierarchy which satisfies a chosen privacy requirement (k-anonymity, ℓ-diversity, t-closeness etc.). Figure 1 shows an example recursive partitioning that might be considered by an even-split algorithm. Nodes denote possible groups, and the number in each node denotes the number of entities within the possible group. Hence, the whole data set consists of 54 entities, which at the top-level can be split into two groups of size 27. The highlighted nodes indicate a possible grouping that may be output by the algorithm: the input is published as six groups, of size 6, 7, 14, 9, 9 and 9 respectively.

We argue that however the grouping is chosen, and whatever method is used to search for the groupings, any such even-split algorithm is not vulnerable to a minimality attack. The insight is that the resulting grouping is sufficiently symmetrical to eliminate

```
Input: a microdata table $T$ and a parameter $\ell$
Output: a set of groups $\{G_1, ..., G_p\}$ so that $\{G_1, ..., G_p\}$
partitions of $T$ and each $G_i$ $(1 \leq i \leq p)$ satisfies $\ell$-diversity.
1. Order tuples in $T$ and partition them into buckets of size $\ell$.
2. Let $T = \bigcup_{i=1}^{n} b_i$ where each bucket $b_i$ has $\ell$ tuples.
3. $Q = \{b_1, b_2, \cdots, b_n\}$, $i = 0$.
4. while $Q \neq \emptyset$
5.     $\leftarrow i + 1$, remove the first bucket $b$ from $Q$, $G_i = b$
6.     while $G_i$ does not satisfy $\ell$-diversity
7.         remove the first bucket $b'$ from $Q$, $G_i = G_i \cup b'$
```

Figure 2: Greedy Grouping Algorithm with ℓ-diversity

any possibility of inference. For k-anonymity, this follows immediately from the discussion above, since the grouping considers only QIs. We prove this claim for (simple) ℓ-diversity; other variations of ℓ-diversity have similar analyses.

THEOREM 5. *Given the output of an even-split algorithm, knowledge of the algorithm used and possible QI groups considered, the adversary's belief in the probability of any item in* any *possible group taking a particular sensitive value is at most* $1/(\ell - \frac{2}{3})$.

Application to Mondrian. The Mondrian algorithm fits the pattern of considering hierarchical binary splits of the input data and accepting splits which meet a privacy requirement. As observed, when the partitions are "strict", the algorithm is vulnerable to the minimality attack (see the example in Figure 6(b) and also [16]). But when the partitions are "relaxed", the even-split property can be enforced. Following the above analysis, this version of the algorithm is effectively immune to the attack! This gives an unexpected dichotomy: a seemingly minor implementation choice has significant impact on the susceptibility to attack.

Symmetric Grouping Algorithm (SG). To compare to the greedy grouping algorithm, we define an alternate approach based on the even-split paradigm. The SG algorithm is described in detail in Appendix C.3. It is also effectively immune to the minimality attack.

4. ANALYSIS OF GREEDY GROUPING

We analyze the impact of minimality attacks on the permutation-based algorithm, Greedy Grouping (GG) with binary ℓ-diversity.

4.1 Formal Definition of Greedy Grouping

We comment on the operation of the greedy algorithm. From the ℓ-diversity requirement, each group must have size at least ℓ. Moreover, since each sensitive attribute must have an integral number of occurrences, the size of each group must be a multiple of ℓ. Suppose to the contrary that the algorithm produced a group of size $c\ell + j$ for some $j < \ell$. Then there can be at most c occurrences of any sensitive value in the group. But then the algorithm could output only the $c\ell$-sized prefix of the group and still meet the diversity requirement. As a result, we break the input relation into "buckets" of size ℓ, as each group is formed from the union of such buckets.

Formally, GG is shown in Figure 2, using this terminology. The algorithm takes a microdata table T and a parameter ℓ as input, and outputs a set of groups $G_1, G_2, \cdots, G_p$ such that $\{G_1, G_2, \cdots, G_p\}$ is a partition of table T, the sensitive values in each group G_i $(1 \leq i \leq p)$ are randomly permuted, and each sensitive value in each group occurs with relative frequency at most $1/\ell$.

The algorithm starts by partitioning the tuples in T into buckets of size ℓ (line 1-2). Let the buckets be $\{b_1, b_2, ...\}$. The initial grouping preserves data utility by grouping tuples with similar QI values in the same bucket. Then the algorithm iteratively generates the groups $\{G_1, G_2, ..., G_p\}$ (line 4-8). Specifically, to generate G_i $(1 \leq i \leq p)$, starting from the first remaining bucket, the algorithm chooses to merge it with the next bucket until the merged group satisfies ℓ-diversity (line 5-8). There may not be enough buckets and the last group may not be ℓ-diverse; if so, we choose to remove it from consideration for simplicity.

4.2 Properties of Greedy Grouping

We analyze the increased probability with which the attacker can associate particular SAs with particular QIs. As each group is generated independently from the others, it is sufficient to analyze a single group at a time. If the group contains only ℓ items, there is no information for the attacker to use, so instead suppose that this group has been formed by merging together $m > 1$ consecutive buckets under the initial ordering. We write $G_{1,i} = \bigcup_{j=1}^{i} b_j$ for $1 \leq i \leq m$ to denote the concatenation of the first i buckets. Thus group G can be written as $G_{1,m} = \bigcup_{j=1}^{m} b_j$. Since each bucket is of size ℓ, the size of G is $m\ell$. Let $f(G)$ be the fraction of (positive) sensitive values in a group G. Because $G_{1,m}$ satisfies binary ℓ-diversity, we have

$$f(G_{1,m}) \leq \frac{1}{\ell} \tag{1}$$

By minimality, the adversary infers that for $1 \leq i \leq m - 1$,

$$f(G_{1,i}) > \frac{1}{\ell} \tag{2}$$

In particular, since the first bucket of size ℓ is not sufficiently diverse, $f(G_{1,1}) \geq 2/\ell$. Meanwhile, by the greedy nature of the algorithm, the last bucket b_m must be ℓ-diverse, i.e. it contains at most a $1/\ell$ fraction of positive sensitive values. Without additional knowledge, the adversary has to believe that all records in the same bucket share the same probability of having a given sensitive value. Let $p(b)$ be the probability that the adversary associates with records in bucket b having a positive sensitive value.

Without taking the minimality attack into consideration, the adversary's knowledge is limited to (1). In this case, for any bucket $b \in G_{1,m}$, we have $p(b) = f(G_{1,m}) \leq 1/\ell$ and the confidence of making any association between a record and a sensitive value is thus bounded by $1/\ell$. But when the minimality principle is applied, in addition to (1), the adversary also has (2) for all i. These inequations imply more about $p(b)$, and when $p(b) > 1/\ell$ for some bucket b, the minimality attack has succeeded. The goal of our analysis is to *calculate* the adversary's belief $p(b)$ and examine how much larger $p(b)$ can become with this knowledge.

Our main theorem guarantees that $p(b_i) < e/\ell$ for any $1 \leq i \leq m$. In other words, the adversary's confidence in associating any sensitive value to any record is bounded by e/ℓ. The full analysis is presented in Appendix D, from which we conclude:

THEOREM 6. *For any output group $G_{1,m}$, $p(b_i) < \frac{e}{\ell}$.*

This bound is somewhat tight: for a group with $m = 2$ we have that $p(b_1) = 2/\ell$, and as m increases this approaches e/ℓ in the limit.

Extension to other privacy guarantees. Theorem 6 applies the specific case of binary ℓ-diversity. However, it is plausible that the same guarantee holds for related formulations of ℓ-diversity: we argue that the worst case for simple ℓ-diversity is when there is a single frequent sensitive value within a group, and all other values are unique or non-sensitive. If this is the case, then the same analysis applies to this case, and leads us to the same conclusion, that the worst case is e/ℓ confidence. Indeed, our experimental study shows that this e/ℓ guarantee holds in practice for such a privacy

	Attribute	Type	# of values
1	Age	Numeric	74
2	Workclass	Categorical	8
3	Education	Categorical	16
4	Marital_Status	Categorical	7
5	Race	Categorical	5
6	Gender	Categorical	2
7	Occupation	Sensitive	14

Table 1: Description of the *Adult* dataset.

guarantee. Similarly, we could ask how an algorithm like GG performs when the privacy guarantee is t-closeness. This requirement is less amenable to analysis; hence we study it experimentally.

4.3 Randomized Choice Methods

We have argued that deterministic operation is an important factor in allowing a minimality attack (also assumed in the definitions of [22]). In this section, we study how random choices affect the level of disclosure. Merely incorporating arbitrary randomization is not sufficient to prevent attacks. Consider augmenting an existing (vulnerable) anonymization method by tossing a fair coin to determine whether to publish the output of the algorithm, or to publish nothing. Then, conditioned on some results being published, the adversary can still perform the attack on them. More realistically, consider a method which merges two groups together into a larger group either when it is forced to do so to satisfy a privacy guarantee, or also when a low-probability event occurs. The intention is that the adversary should be unable to deduce whether the merger was "forced", or if it was "voluntary". However, if the random event is very low probability, then an attacker's belief may be that it was much more likely to be a forced merger.

Consequently, we still need to carefully analyze randomized algorithms to bound the overall vulnerability. In particular, we define and study a randomized version of GG. The analysis shows that while indiscriminate randomization is not a universal cure, applied carefully it can reduce the effect of minimality attacks.

Randomized Greedy Grouping Method (RGG). Informally, the main difference is that given a partial group G_i, we may randomly choose to add the next bucket b to the group, rather than always closing the group as soon as it meets the privacy guarantee. More formally, we modify the algorithm presented in Figure 2, by changing the condition in line 6 to be if either G_i does not satisfy ℓ-diversity *or* if a random event with probability p occurs. The output of the algorithm is the same as before: a collection of groups, where the QI values and SA values in each group are presented, without any further description of how they are related. As in GG, we can optionally apply "anatomization" to reduce the group sizes.

THEOREM 7. *For any output group G,* $p(b_1) \in (1/\ell, e/\ell)$.

For $p = 0$, this gives the upper bound of e/ℓ. As we increase p, the aim is to reduce this bound progressively. However, we cannot guarantee that this will always work: as noted above, the attacker's belief depends on the likelihood of merges being forced. In some cases, even with $p = 1$, the (worst-case) bound remains close to e/ℓ. Hence, we study the power of this method empirically.

5. EXPERIMENTS

We now empirically evaluate the various methods that we have discussed, including (1) SA-only methods (Anatomy); (2) symmetric methods (SG); (3) asymmetric methods (GG); and (4) randomized methods (RGG). We first study the power of minimality attacks on the vulnerable methods, and then go on to compare the utility provided by each method for answering queries. Throughout, we use permutation as the recoding method, as this is observed to have better utility than generalization/supression.

As with prior work on anonymization, we used the Adult dataset from the UC Irvine machine learning repository [2], comprised of data collected from the US census[2]. Tuples with missing values are eliminated, leaving a total of 45222 valid tuples. We use seven attributes of the fifteen attributes in the data, as described in Table 1. We pick "Occupation" as the sensitive attribute, which contains 14 values. We designate 2 of the 14 values ("Tech-Support" and "Craft-Repair") as the (positive) sensitive values.

5.1 Privacy Risks

We ran experiments with GG and RGG to establish the increase in the attacker's confidence in practice. The privacy risks of the published grouping are computed via Monte Carlo sampling, as follows. Given the published data, unifomly at random create an assignment of the sensitive values to the tuples[3]. Reject those assignments on which the (deterministic) algorithm would not produce the published grouping: (i.e. use the minimality principle). Over non-rejected assignments, the privacy risk is the largest probability of a sensitive value in any of the buckets forming any group.

Figure 3(a) shows the fraction of tuples that are vulnerable to minimality attacks with ℓ-diversity as the privacy model in GG (i.e., they have a probability of above $1/\ell$ under minimality attacks). The two bars show the effect of setting "Tech-Support" and "Craft-Repair" as the SA in turn. In all cases, the fraction of vulnerable tuples is quite small (in most cases below 10%), and tends to increase with ℓ: this may be because when a group is deemed vulnerable, more tuples are involved.

Figure 3(b) shows the maximum privacy risk among all vulnerable groups, shown as a multiple of $1/\ell$. Our analysis argues that this should be at most e; and at least 2 if any bucket is unsafe. Figure 3(b) shows that indeed this multiple ranges between 2 and e: when $\ell = 6$, for instance, the maximum privacy risks for the two sensitive values are 2.52 and 2.70, respectively.

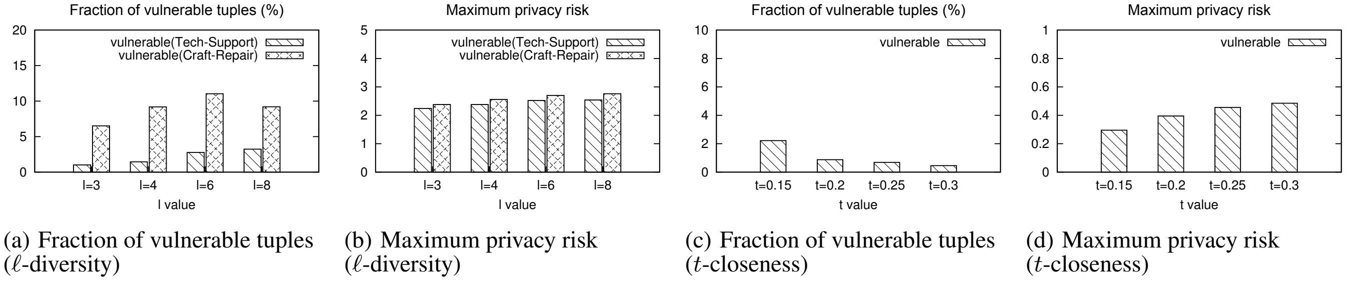
Figure 3(c) and Figure 3(d) show the results when the privacy guarantee used in the algorithm is t-closeness. Figure 3(c) shows that the fraction of tuples vulnerable to minimality is low: about 2% at most in our experiments. The trend is increasing as t decreases (also corresponding to a stronger privacy guarantee and larger groups). Figure 3(d) shows the maximum privacy risk in terms of the actual distance between the probability distribution in a group and that in the overall table. The actual distance is at most twice the t value in all experiments, suggesting that the privacy breach for minimality attacks for this measure is also bounded.

5.2 Randomized Methods

Our next experiments show that RGG, which adds randomization, can effectively prevent minimality attacks (the number of vulnerable tuples reduces significantly) without much reduction of data utility. We evaluate the privacy risks of the randomized choice algorithms using different probability values of $p = 0, 0.2, 0.4, 0.6, 0.65$. Figure 4 shows the results for RGG under ℓ-diversity with $\ell = 6$. Figure 4(a) shows that the fraction of vulnerable tuples decreases quickly as p increases. Setting $p = 0.6$ is sufficient to ensure that no groups are vulnerable with respect to "Tech-Support"; increasing to $p = 0.65$ makes all groups safe for both sensitive values. Figure 4(b) shows that the maximum privacy risk (as a multiple of $1/\ell$) reduces quickly as p increases. When $p = 0.6$, no group

[2] Results on other data were similar, and are omitted for brevity

[3] This effectively applies a uniform prior to the possible worlds; other priors are of course possible [6]



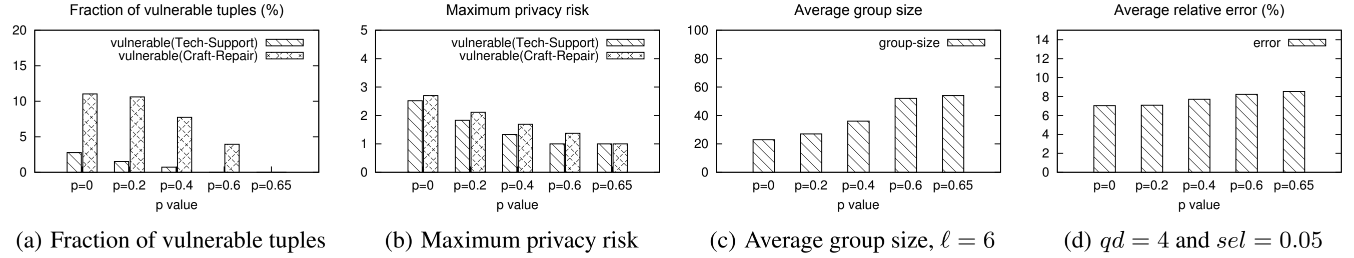
(a) Fraction of vulnerable tuples (ℓ-diversity)

(b) Maximum privacy risk (ℓ-diversity)

(c) Fraction of vulnerable tuples (t-closeness)

(d) Maximum privacy risk (t-closeness)

Figure 3: Privacy risks of minimality attacks



(a) Fraction of vulnerable tuples

(b) Maximum privacy risk

(c) Average group size, $\ell = 6$

(d) $qd = 4$ and $sel = 0.05$

Figure 4: Effectiveness of random choice methods

is vulnerable to "Tech-Support" and the maximum privacy risk for "Craft-Repair" is reduced to $1.37/\ell$. When $p = 0.65$, the maximum privacy risk is reduced to $1/\ell$, i.e. the anonymized data is safe from the minimality attack.

Figures 4(c) and 4(d) show that adding randomization does not significantly affect data utility. Figure 4(c) shows that the average group size does not increase by very much. Note that if a random event occurs with probability p, the expected number of times before a failure is expected is $1/(1 - p)$. Hence, if a group remains safe as more buckets are randomly added, then we expect $1/(1-p)$ new buckets to be added after it has met the privacy requirement. More critically, the average relative error (shown in Figure 4(d)) does not grow much as p is increased; (more details on this evaluation are in the next section). Taken together, these results suggest that careful use of randomization can defuse the minimality attack without sacrificing utility.

5.3 Aggregate Query Answering

Our final experiments address utility by measuring the accuracy of query answering on the anonymized data. All methods are configured to offer the same level of privacy after accounting for minimality, equivalent to ℓ-diversity. We evaluated utility of the anonymized data for aggregate query answering, as in prior work [19, 11]. We show results for "COUNT" queries where the selection predicate involves the SA, as in [19], of the form:

SELECT COUNT(*) FROM Table
WHERE $v_{i_1} \in V_{i_1}$ AND ... $v_{i_d} \in V_{i_d}$ AND $s \in V_s$

where $v_{i_j} (1 \leq j \leq d)$ is the QI value for attribute A_{i_j}, $V_{i_j} \subseteq D_{i_j}$ where D_{i_j} is the domain for attribute A_{i_j}, s is the SA value and $V_s \subseteq D_s$ where D_s is the domain for the sensitive attribute S.

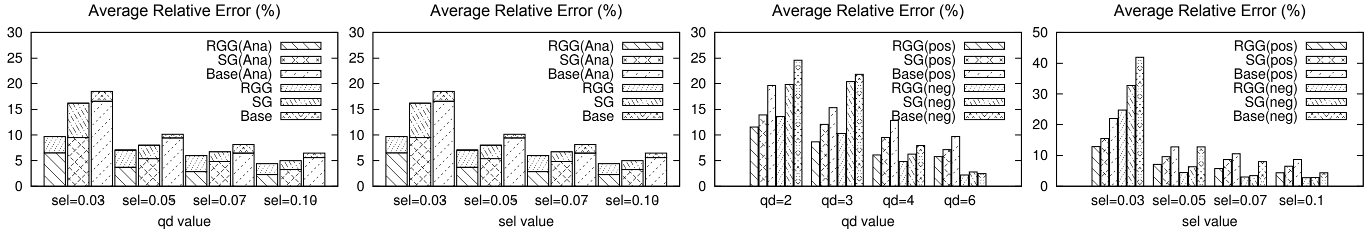
A query predicate is characterized by: (1) the predicate dimension qd and (2) the query selectivity sel. The dimension qd indicates the number of quasi-identifiers involved in the predicate. The selectivity sel indicates the number of values in each $V_{i_j}, (1 \leq j \leq qd)$, where the size of $V_{i_j}, (1 \leq j \leq qd)$ is randomly chosen from $\{0, 1, ..., sel|D_{i_j}|\}$. For each parameter setting, we tested a set of 1000 queries. Relative error is computed for each query in the standard way, as the absolute difference between the true and estimated values for each query, scaled by the true value; we show the average relative error (ARE) over the set of queries.

We compare the methods GG, RGG, SG, and a baseline method which simply releases two independent tables, one containing all QI attributes and one containing the sensitive attribute. Since the RGG method with p chosen to eliminate the minimality attack always performed better than GG with the group size set to $e\ell$ (also sufficient to eliminate minimality), we report results for RGG only. We show the impacts of "anatomization" on all methods: using Anatomy to partition a large group into smaller groups, each of which satisfies ℓ-diversity. Therefore, we have RGG(Ana), SG(Ana), Base(Ana) corresponding to the three anatomized methods [4].

Figure 5(a) shows ARE as a function of query dimension and Figure 5(b) shows ARE as a function of query selectivity. In all experiments, RGG has smaller ARE than SG and the baseline, and this relative ordering holds whether or not we perform anatomization. The experiments also show that ARE decreases when query dimension increases and when query selectivity increases (both of which effectively eliminate more tuples from consideration). Anatomization reduces ARE: breaking a large group into smaller ones still satisfies privacy with a substantial improvement on data utility.

Two characteristics that a grouping method should satisfy in order to have better data utility are: (1) the average group size should be small and (2) the quasi-identifier values in a group should be similar. When both conditions hold, it is more likely that either all tuples or none of the tuples in a group satisfy the query predicate. Thus there is less uncertainty in the answer, reducing estimation error and so improving utility. As groups grow larger or more disparate, it is less likely that we have this "all or nothing" feature.

[4] Base(Ana) is equivalent to the original Anatomy algorithm [19].



(a) $\ell = 6$ and $sel = 0.05$

(b) $\ell = 6$ and $qd = 4$

(c) $\ell = 6$ and $sel = 0.05$

(d) $\ell = 6$ and $qd = 4$

Figure 5: Aggregate Query Answering

Before anatomization SG has a somewhat larger average group size than RGG. The experiments in Figures 4(c) and 4(d) showed that although the RGG group size is somewhat larger than ℓ, the clustering of QI values in the group is effective at keeping the relative error low. But after anatomization, the baseline and RGG methods have similar group sizes, very close to ℓ, indicating that group size alone is insufficient to determine utility. SG has slightly larger group size, but still less than 2ℓ.

Intuitively, if each group is representative of the overall table (i.e., the SA distribution in each group and in the overall table are similar), then the baseline method "Base" would work well. We partition the 1000 queries into two sets of the 100 queries which show the most correlation. That is, we take queries where the true count is much larger (smaller) than if we treat the attributes as uncorrelated and simply estimate based on multiplying selectivities, as the negative (positive) queries. Figure 5(c) and Figure 5(d) show the results for varying qd values and sel values, respectively. They show that both errors decrease with increasing qd and sel values. Moreover, we see that RG often achieves half the error of the baseline methods, indicating that we can defeat minimality and still obtain greater utility.

6. CONCLUSIONS AND FUTURE WORK

We have seen that with careful analysis it is possible to bound or even eliminate the impact of minimality attacks which exploit knowledge of the anonymization method. This is not to say that this attack can be brushed aside entirely: for example, we saw an algorithm (Mondrian) where a seemingly minor implementation choice (whether to use a strict or relaxed grouping) makes all the difference in whether the attacker has a higher chance of finding the sensitive value of some tuples. Other algorithms which share the danger signs identified (lack of symmetry, determinism, choices based on both QI and SA values) remain vulnerable. For example, the well-known Incognito algorithm [8], when combined with an ℓ-diversity condition, is easily shown to be susceptible for examples like those shown here and in [15].

Nevertheless, our understanding of this style of attack is much advanced. It remains to bring this insight to bear on other combinations of algorithms and privacy requirements. Our focus, in line with the bulk of the anonymization literature, has been on methods dealing with data represented within a table. However, there is much recent interest in also handling other forms of data, such as set-valued (representing transactions) and graph-structured (representing social-networks) [3]. Algorithms in these domains are also concerned with minimizing information loss, and so are potentially vulnerable to attack; we believe that similar analytic methodologies will quantify exactly how damaging these attacks may be.

7. REFERENCES

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, pages 246–258, 2005.

[2] A. Asuncion and D. Newman. UCI ML repository, 2007.

[3] G. Cormode and D. Srivastava. Anonymized data: generation, models, usage. In *SIGMOD*, 2009.

[4] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, pages 205–216, 2005.

[5] J. Gehrke and A. Machanavajjhala. Privacy in data publishing. In *S&P*, 2009.

[6] D. Kifer. Attacks on privacy and deFinetti's theorem. In *SIGMOD*, 2009.

[7] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang. Aggregate query answering on anonymized tables. In *ICDE*, pages 116–125, 2007.

[8] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD*, pages 49–60, 2005.

[9] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, page 25, 2006.

[10] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *ICDE*, pages 106–115, 2007.

[11] T. Li and N. Li. Injector: Mining background knowledge for data anonymization. In *ICDE*, 2008.

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

[13] P. Samarati. Protecting respondent's privacy in microdata release. *TKDE*, 13(6):1010–1027, 2001.

[14] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. SRI-CSL-98-04, SRI, 1998.

[15] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei. Minimality attack in privacy preserving data publishing. In *VLDB*, pages 543–554, 2007.

[16] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei. Anonymization-based attacks in privacy-preserving data publishing. *ACM Trans. Database Syst.*, 34(2), 2009.

[17] R. C.-W. Wong, A. W.-C. Fu, K. Wang, Y. Xu, and P. S. Yu. Can the utility of anonymized data be used for privacy breaches? Technical Report abs/0905.1755, arXiv, 2009.

[18] 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*, pages 754–759, 2006.

[19] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In *VLDB*, pages 139–150, 2006.

[20] X. Xiao, Y. Tao, and N. Koudas. Transparent anonymization: Thwarting adversaries who know the algorithm. *ACM Trans. Database Syst.*, 35(2):1–48, 2010.

[21] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-based anonymization using local recoding. In *KDD*, pages 785–790, 2006.

[22] L. Zhang, S. Jajodia, and A. Brodsky. Information disclosure under realistic assumptions: Privacy versus optimality. In *ACM Conference on Computer and Communications Security*, 2007.

APPENDIX

A. A BRIEF HISTORY OF ATTACKS

Research on anonymization is driven by studying potential attacks on proposed methods, leading to improved anonymization methods and new paradigms. Indeed, the groundbreaking work on k-anonymization can be seen as a response to trivial methods which merely remove uniquely identifying information [14]. That first work showed how quasi-identifiers in the data could be linked with external sources to reidentify individuals in the data and discover their sensitive values. Data which satisfies k-anonymity is still susceptible to attacks: if many individuals in a group have the same or similar sensitive values, the attacker's belief in a particular value can exceed the desired $1/k$ bound. Such "homogeneity attacks" (identified as early as [13]) led to stronger requirements in the form of ℓ-diversity (based on the frequency of sensitive values in a group) and t-closeness (based on the similarity of distributions in the group to the global distribution).

The minimality attack can be seen as another step in this process of attack and strengthening: algorithms which try too eagerly to enforce a condition such as ℓ-diversity with the minimum of generalization or modification end up failing to meet their promises when the action of the algorithm is considered. The attack was first proposed by [15]; similar ideas were also discussed in [22] around the same time. Wong et al. [15] showed there exist algorithms that attempt to enforce measures such as ℓ-diversity and t-closeness that are susceptible. The journal version of the paper showed in particular that when algorithms such Incognito [8], Mondrian [9] and Zhang's algorithm [22] are used to enforce ℓ-diversity, the result can be attacked. Wong *et al.* then proposed a "Mask" algorithm that avoided minimality by modifying sensitive values, a paradigm which most anonymization algorithms explicitly avoid due to the distortion of basic statistics that results.

Recently, Xiao *et al.* [20] independently identified algorithms that can achieve ℓ-diversity against an adversary who knows the algorithm. Compared to their algorithms, ours add less restrictive constraints and thus potentially have better data utility. We also demonstrated that the effects of the minimality attack are limited.

New attacks and countermeasures continue to be proposed. A limitation of our approach is that we focus on the case where the attacker has limited background knowledge, and so makes a uniformity assumption over the possible worlds implied by the anonymized data. However, recent work has considered extracting information about correlations between QIs and SAs in the anonymized data itself, and used this to modify beliefs about the likelihood of possible worlds [6, 17]. These so-called "foreground knowledge" or "deFinetti" attacks allow an attacker to again raise their confidence above the $1/\ell$ level. The observed effects are most pronounced on small groups (of size 2 to 4 tuples). An open problem for the community is to exhibit algorithms where the effect of such attacks is bounded or removed, just as we have for the minimality attack.

B. EXAMPLES OF MINIMALITY ATTACK

QI	SA
a	Negative
b	Negative
c	Positive
d	Positive

(a) Example for GG algorithm

QI	SA
a	Negative
b	Negative
c	Negative
c	Negative
c	Positive
d	Positive

(b) Example for Mondrian

Figure 6: Examples of minimality attack on anonymized data

The anonymized table shown in Figure 6(a) shows a group of 4 items that has been formed by GG with a binary 2-diversity requirement. The grouping is shown prior to recoding. The left column shows the set of QIs present in the group in lexicographic order (stylized as values a, b, c, d) and the right column shows that there are two "positive" and two "negative" values associated with the data. The intent is that any bijection between the QIs and SAs in this group should be possible, and since $\ell = 2$, no QI should be associated with a positive value with probability greater than $\frac{1}{2}$.

However, knowing that the GG algorithm was used, an attacker can learn more. The algorithm considered the input in the order a, b, c, d, and must have decided not to release the grouping $\{a, b\}$—therefore, this must not meet ℓ-diversity. The only way that this could happen given the tuples present in the group is if both a and b were positive, violating the privacy requirement.

In Figure 6(b), the data is grouped (again, prior to recoding) to meet binary 2-diversity using Mondrian with strict partitioning. The QI value c is the median of the group, and in this implementation of Mondrian, the two groups considered were $\{a, b\}$ and $\{c, c, c, d\}$. If both positive values were in the larger group $\{c, c, c, d\}$, this would have met the diversity requirement. Similarly, if only one positive value was in each group, the result would also meet the diversity requirement. So it must be that both a and b were positive, again violating the privacy requirement.

By contrast, if the Anatomy algorithm had been used to obtain binary 2-diversity, no such inference would be possible: in Figure 6(a), the SAs would be split into two groups with a positive and negative value in each, but the attacker would have no further information with which to identify the corresponding QIs. Likewise, in Figure 6(b) the split would be into three groups with at most one positive value in each, and the attacker could make no further inference. □

C. DETAILED TECHNICAL MATERIAL

C.1 Proof of Claim 4

CLAIM 4. *Methods which choose an SA-Intact grouping based on sensitive attributes alone are safe from the minimality attack.*

PROOF. Consider such a method which does not examine the quasi-identifiers at all during the formation of groups. Then the QI values for each input item could be altered completely, and the algorithm would be just as likely to find the same grouping. In particular, for any given group, the QIs could be interchanged so that any possible mapping of QIs to rows would be possible, and the same grouping would still result from the anonymization. But this means that the attacker has no way of establishing which QI belonged to which original row (beyond what is published), and hence to which original SA, within a group. □

C.2 Proof of Theorem 5

THEOREM 5. *Given the output of an even-split algorithm, knowledge of the algorithm used and possible QI groups considered, the adversary's belief in the probability of any item in* any *possible group taking a particular sensitive value is at most* $1/(\ell - \frac{2}{3})$.

PROOF. The output of the algorithm consists of a set of groups of QIs and the corresponding set of groups of sensitive values. These form a partition of the original input to the algorithm. Clearly, the output groups give more information about individuals than any groups corresponding to ancestors in the tree. So consider a particular output group, and the child groups which it could have been partitioned into. Assume that the algorithm considered partitioning this group into its children, but did not since this would have violated the diversity requirement. We treat this as a deterministic decision; allowing a random choice only decreases the adversary's confidence in the probability of various outcomes.

Let the group under consideration be denoted G, and consider any sensitive value x present in G. Suppose this value occurs r times within G, and that G is formed from the union of $k \geq 2$ subgroups. We reason about the probability of x in each of these subgroups. If all subgroups have the same size, then by the symmetry, any possible assignment of the a copies of x to each group is equally likely. Since G is ℓ-diverse, the probability of x in each group is also at most $1/\ell$.

Therefore, the case to consider is when the k subgroups are not equal in size. By the conditions on the grouping, some are size $s-1$ and the rest are size s. Since we assume that some subgroups is not diverse, let r be the smallest value so that $r/(s-1) > 1/\ell$. Consider the possible worlds that have various numbers of instances of x in each of the k subgroups. We can have at least r copies of x in the smallest group, and the remaining copies divided between the other groups. Let v be a vector of k values which records how many copies of x are in each group: v_1 indicates the number in the first group, and so on. Due to the similarity of sizes of the other groups, if there are $r+1$ or more copies of x present, then they cannot satisfy the diversity condition: since $s \geq r+1$ (otherwise, it is not possible to have this many copies of x present in a group), it follows that $rs+s-r-1 \geq rs$, and hence $(r+1)/s \geq r/(s-1) > 1/\ell$. The other case to analyze is whether r copies of x are placed in a group of size s is diverse or not. If it is not, then $r/s \leq 1/\ell$ while $r/(s-1) > 1/\ell$, and so $s = r\ell$. Consequently, with r copies of x in a group of size $s-1$ gives the adversary a probability of $r/(s-1) = s/(s-1) \cdot 1/\ell$.

We analyze all possible vectors v which have at least one group violating the diversity requirement. First, consider all vectors which have one entry at least $r+1$. For each such vector, consider all vectors which are permutations of it: observe that they all correspond to assignments of copies of x to groups so that at least one group has more than $r+1$ copies, and hence cannot be diverse. Across all these vectors then the average number of copies of x in each group is $\sum_{i=1}^{k} v_1/k = r/k$. Each possible assignment generates almost the same set of possible worlds: there are slightly more possible worlds corresponding to cases with the lack of diversity due to groups of size s than size $s-1$, but this only weakens the adversary's knowledge. Hence, for groups of size $s-1$, the adversary's belief is bounded by

$$\frac{r}{k(s-1)} \leq \frac{1}{\ell} \cdot \frac{|G|}{ks-k} \leq \frac{1}{\ell} \cdot \frac{ks-1}{ks-k} \leq \frac{s}{s-1} \cdot \frac{1}{\ell}$$

for $s \geq 1$.

The only remaining cases are those where there are r copies of x associated with groups of size $s-1$. The confidence on just this subset is also at most $s/(s-1) \cdot 1/\ell$, so no matter how much weight is placed on either of these two cases, the attacker's confidence can be at most this much. Since $s-1 \geq \ell$ (else the algorithm would not have considered a group of this size, as ℓ-diversity requires the group to be size at least ℓ), the bound can be written as $1/(\ell - \frac{\ell}{\ell+1})$. For any reasonable ℓ, this is only marginally worse than $1/\ell$: in the worst case, $\ell \geq 2$ so the final guarantee on the adversary's knowledge is essentially equivalent to $(\ell - \frac{2}{3})$-diversity. □

C.3 The Symmetric Grouping Algorithm.

Consider the input to the greedy grouping, which is the data sorted based on some chosen ordering. We apply a natural even-split algorithm to this: starting with the full data, consider the split that divides the current group under consideration into two almost equal halves, say by always breaking ties with the larger group consisting of the left half of the items under the given ordering (hence the algorithm remains quite deterministic; it is the symmetric nature which avoids minimality attack). If both new groups satisfy the privacy requirement, then we recurse on each group in turn, else we keep the current group, and terminate this branch of the recursion. Pseudocode for a realization of SG is given in Figure 7 in the appendix. We note that although described in a top-down fashion, the algorithm can also be thought of as bottom-up: starting with the leaf-level groups, merge a node with its sibling if it does not meet the diversity requirement. Given the same hierarchy of groups over an input, the bottom-up merging reaches the same final grouping as the top-down approach. The bottom-up method looks similar to the greedy grouping algorithm, but made symmetric: instead of growing groups left-to-right, the algorithm grows groups respecting the hierarchy, and merges a pair if either neighbor is unsafe. However, by Theorem 5, there is very little information that an adversary can deduce from the output of SG, whereas we have seen several examples where GG reveals more information.

```
Input: a microdata table $T$ and a parameter $\ell$
Output: a set of groups $\{G_1, G_2, ..., G_p\}$ such that
$\{G_1, G_2, ..., G_p\}$ is a partition of $T$
and each group $G_i$ $(1 \leq i \leq p)$ satisfies $\ell$-diversity.
1. Initialize $G_1 = T$.
2. Split the current group $G = t_l \dots t_r$ into $G' = t_l \dots t_{\lfloor (l+r)/2 \rfloor}$
      and $G'' = t_{\lfloor (l+r)/2 \rfloor + 1} \dots t_r$.
3. If both $G'$ and $G''$ satisfy $\ell$-diversity, recurse on each in turn.
```

Figure 7: Symmetric Grouping Algorithm with ℓ-diversity

C.4 Proof of Theorem 7

THEOREM 7. *For any output group G, $p(b_1) \in (1/\ell, e/\ell)$.*

PROOF. As for the deterministic version of GG, it suffices to consider a single group G alone. By definition, G is formed from the union of a set of buckets, and the QI values of the buckets are known. However, the associations between the QI values and the sensitive attribute values are not known.

We consider all possible assignments of the sensitive values to the QI values. Each assignment is viewed as a possible world, W. In some possible worlds, the first bucket b_1 meets the privacy guarantee and so the merge is voluntary; we call the set of such possible worlds WS. We call the set of worlds where this does not hold WN. Note that the relative sizes of these two sets is data dependent. In one extreme case, all sensitive values in G are distinct (or negative), $|\mathsf{WN}| = 0$. Here, all possible worlds are safe, so if G consists of more than one bucket, the merges must all have been voluntary. Another extreme case is when each sensitive value in G occurs exactly a $1/\ell$ fraction of the time. Here, $|\mathsf{WN}| \gg |\mathsf{WS}|$, so it is much more likely that the merges were forced.

Without additional knowledge, we must treat each of the possible worlds in WS as equally likely (we write this probability as $\Pr[W \in \mathsf{WS}]$); similarly, each possible world in WN is also equally possible (let the probability be $\Pr[W \in \mathsf{WN}]$). When minimality attacks are not considered, $\Pr[W \in \mathsf{WS}] = \Pr[W \in \mathsf{WN}]$ and therefore the probability that a record $r \in g$ takes a positive sensitive value is the fraction of positive values in G. When applying the minimality attack to the (deterministic) GG algorithm, $\Pr[W \in \mathsf{WS}] = 0$ and $\Pr[W \in \mathsf{WN}] = \frac{1}{|\mathsf{WN}|}$.

For the binary ℓ-diversity guarantee, the probability that a record in the first bucket b_1 is sensitive assuming $W \in \mathsf{WS}$ is given by $p(b_1 | W \in \mathsf{WS})$. The probability $p(b_1 | W \in \mathsf{WN})$ is defined similarly for $W \in \mathsf{WN}$. Then $p(b_1)$, the probability that a record in b_1 is sensitive is given by

$$\begin{aligned} p(b_1) =& |\mathsf{WS}| \Pr[W \in \mathsf{WS}] p(b_1 | W \in \mathsf{WS}) \\ &+ |\mathsf{WN}| \Pr[W \in \mathsf{WN}] p(b_1 | W \in \mathsf{WN}) \end{aligned}$$

The probability that a uniformly chosen world W belongs to WS is given by $|\mathsf{WS}|/(|\mathsf{WS}| + |\mathsf{WN}|)$, but the probability that this results in a merger is $p|\mathsf{WS}|/(|\mathsf{WS}| + |\mathsf{WN}|)$. Given that a merger took place, we can write $\Pr[W \in \mathsf{WS}] = \frac{p}{p|\mathsf{WS}|+|\mathsf{WN}|}$ and $\Pr[W \in \mathsf{WN}] = \frac{1}{p|\mathsf{WS}|+|\mathsf{WN}|}$. Note that when $p = 0$, the bound reduces to the simpler case where all possible worlds belong to WN. The bound on $p(b_1)$ is now

$$p(b_1) = \frac{p|\mathsf{WS}| p_x(b_1 | W \in \mathsf{WS}) + |\mathsf{WN}| p_x(b_1 | W \in \mathsf{WN})}{p|\mathsf{WS}| + |\mathsf{WN}|}$$

From our analysis in Section 3), we have that $p(b_1 | W \in \mathsf{WS}) \leq 1/\ell$ and $p(b_1 | W \in \mathsf{WN}) \leq e/\ell$. Thus,

$$p(b_1) \leq \frac{p|\mathsf{WS}| + |\mathsf{WN}| e}{\ell(p|\mathsf{WS}| + |\mathsf{WN}|)} \in (1/\ell, e/\ell]$$

□

D. ANALYSIS OF GREEDY GROUPING

D.1 Reducing to the first bucket b_1

In the next theorem, we show that the greatest privacy risk occurs within the first bucket b_1.

THEOREM 8. *Given an output group $G_{1,m} = \bigcup_{j=1}^{m} b_j$ where the buckets are in the order of $b_1, b_2, \cdots, b_m$, we have*

$$\forall 1 \leq i < m : p(b_i) \geq p(b_{i+1}).$$

PROOF. Any group $G_{1,m}$ output by GG could have resulted from any one of many possible input relations. Each possible world corresponds to a permutation of the sensitive values in $G_{1,m}$. We analyze these possible worlds by grouping together all worlds which share the same number of positive values in each bucket. Each such grouping of possible worlds can be mapped to a sequence of m integers from $[\ell + 1]^m$, recording the number of positive values in each of the m buckets in turn. A possible world could have been the input relation if and only if the corresponding sequence is valid, i.e., it satisfies the constraints given by Inequations (1) and (2). Each valid sequence implies a set of valid worlds, although note that the number of worlds generated by different sequences varies.

Consider such a sequence $(n_1, n_2, \ldots, n_m)$. Observe that, for it to be valid, we must have $\sum_{i=1}^{m} n_i = m$: it cannot be more, else the full sequence is not ℓ-diverse, and it cannot be less, else some prefix is ℓ-diverse. To analyze the likelihoods of different numbers of positive values within the buckets, we hold all elements of the sequence fixed, except for n_i and n_{i+1}. Then the value $r = n_i + n_{i+1}$ is also fixed, by the above observation. We consider all possible assignments of n_i and n_{i+1} that generate valid sequences. Certainly, if assigning $(n_i = c, n_{i+1} = r - c)$ generates a valid sequence, then $(n_i = c + 1, n_{i+1} = r - c + 1)$ is also a valid sequence. Let the minimum value for n_i that generates a valid sequence be t, then $n_i = j, n_{i+1} = r - j$ is valid, provided $t \leq j \leq r$; all other assignments are invalid.

Let the number of valid worlds corresponding to this sequence with $(n_i = j, n_{i+1} = r - j)$ be N_j. Then we have $N_j = N_{r-j}$, since we can establish a bijection between the worlds in each case: essentially, any world with $(n_i = j, n_{i+1} = r - j)$ becomes a world with $(n_i = r - j, n_{i+1} = j)$ by exchanging buckets i and $i + 1$. Now we can calculate

$$p(b_i) = \frac{\sum_{j=t}^{r} j N_j}{\sum_{j=t}^{r} N_j} \text{ and } p(b_{i+1}) = \frac{\sum_{j=t}^{r} (r-j) N_j}{\sum_{j=t}^{r} N_j}.$$

Therefore, $p(b_i) - p(b_{i+1}) = \dfrac{\sum_{j=t}^{r} (2j - r) N_j}{\sum_{j=t}^{r} N_j}$.

To prove the theorem, we show that the numerator is never negative. When $t \geq r/2$, we have $(2j - r)N_j \geq 0$ for all $t \leq j \leq r$. Then $p(b_i) - p(b_{i+1}) \geq 0$.

When $t < r/2$, write $T = \displaystyle\sum_{j=r-t+1}^{r} (2j - r)N_j$, so we have

$$\begin{aligned} &\sum_{j=t}^{r} (2j - r)N_j \\ &= T + \sum_{j=t}^{\lfloor r/2 \rfloor} (2j - r)N_j + \sum_{j=\lfloor r/2 \rfloor + 1}^{r-t} (2j - r)N_j \\ &= T + \sum_{j=r-\lfloor r/2 \rfloor}^{r-t} (2(r - j) - r)N_{r-j} + \sum_{j=\lfloor r/2 \rfloor + 1}^{r-t} (2j - r)N_j \\ &= T + \sum_{j=\lfloor r/2 \rfloor + 1}^{r-t} (r - 2j)N_j + \sum_{j=\lfloor r/2 \rfloor + 1}^{r-t} (2j - r)N_j \\ &= T \geq 0 \end{aligned}$$

We have $p(b_i) \geq p(b_{i+1})$ for any sequence $(k_1, k_2, \cdots, k_m)$. Therefore, $p(b_i) \geq p(b_{i+1})$. □

D.2 Calculation of $p(b_1)$.

Following the analysis in Theorem 8, in a given world we have n_i occurrences of positive sensitive values in bucket b_i, and

$$\sum_{i=1}^{m} n_i = m \qquad (3)$$

Note that Equation (3) guarantees that Inequation (1) holds since

$$f(G_{1,m}) = \tfrac{m}{m\ell} = \tfrac{1}{\ell}$$

The constraints given by Inequation (2) are equivalent to

$$\forall 1 \leq j < m : \textstyle\sum_{i=1}^{j} n_i \geq j + 1 \qquad (4)$$

We calculate $p(b_1)$ for b_1 by enumerating all possible worlds consistent with the output group $G_{1,m}$. Each possible world of $G_{1,m}$ corresponds to a permutation of the sensitive values in G_{1m} so that Equation (3) and Inequations (4) are satisfied. Based on the random worlds assumption, we must consider each possible world to be equally likely. The probability $p(b_1)$ is calculated simply by dividing the expected number of positive values in b_1 (written as $\mathsf{E}[n_1]$) by the size of b_1, i.e.:

$$p(b_1) = \mathsf{E}\left[\tfrac{n_1}{|b_1|}\right] \qquad (5)$$

Let $m^* = \min\{m, \ell\}$, an upper bound on any n_i, and let N_k be the number of possible worlds where $n_1 = k$. Then, we have

$$\mathsf{E}(n_1) = \frac{\sum_{k=2}^{m^*} k N_k}{\sum_{k=2}^{m^*} N_k} \qquad (6)$$

By Inequation (2), $n_1 \geq 2$, i.e., bucket b_1 must contain at least 2 positive values. Since $|b_1| = \ell$, combining Equations (6) and (5) yields

$$p(b_1) = \frac{\sum_{k=2}^{m^*} kN_k}{\sum_{k=2}^{m^*} \ell N_k} \tag{7}$$

Equation (7) is our formula for calculating the adversary's belief $p(b_1)$. We separately give a closed form for the numerator and the denominator (given by Lemmas 10 and 11). First, we compute N_k.

LEMMA 9. $N_k = \frac{k-1}{m-1}\binom{\ell}{k}\binom{m\ell-\ell}{m-k} m!(m\ell-m)!$

PROOF. Recall the notion of a sequence, which counts the number of occurrences of positive sensitive values in each bucket. N_k is found using sequences which have $n_1 = k$, and obey Inequation (4) and (3). Such sequences are called "valid" sequences.

To prove the lemma, we show a stronger result: among all sequences of length m that start with $1 \leq k \leq m$ and then with permutations of $(n_2, n_3, \cdots, n_m)$, the fraction of valid sequences is $\frac{k-1}{m-1}$. When we apply this result to all possible subsequences of length $m-1$ for $(n_2, n_3, \cdots, n_m)$, we obtain the lemma.

We prove the above result by induction on m. For $m = 1$, the only possible sequence is (1) which is trivially not valid. For $m = 2$, there are three possible sequences: $(2, 0)$ is valid, $(1, 1)$ is not valid, and $(0, 2)$ is not valid.

Assume as the induction hypothesis that the lemma holds for m. In other words, among all sequences of length m that start with $1 \leq k \leq m$ and then with permutations of $(n_2, n_3, \cdots, n_m)$, a $\frac{k-1}{m-1}$ fraction of the sequences are valid. We show that the result also holds for the case of $m+1$. Consider all sequences of length $m+1$ that start with $1 \leq k \leq m+1$: $(k, n_1, n_2, \cdots, n_m)$. For $k = m+1$, there is only one possibility, $(m+1, 0, 0, \cdots, 0)$, which is always a valid sequence.

For $1 \leq k \leq m$, we notice that $(k, n_1, n_2, \cdots, n_m)$ is valid iff $(k+n_1-1, n_2, \cdots, n_m)$ is valid. By the induction hypothesis, the fraction of such sequences that are valid is $\frac{k+n_1-2}{m-1}$. We repeat this analysis for all possible permutations of $(n_1, n_2, \cdots, n_m)$. The overall fraction of valid sequences is the average, given by

$$\begin{aligned}
&\frac{1}{m}\sum_{i=1}^{m} \frac{k+n_i-2}{m-1} \\
&= \frac{km + (m+1-k) - 2m}{m(m-1)} \\
&= \frac{k-1}{m}
\end{aligned}$$

Therefore, the result also holds for $m+1$. By induction, we have shown that the fraction of valid sequences is $\frac{k-1}{m-1}$ among all sequences of length m that start with k. the total number of all these sequences is

$$\binom{\ell}{k}\binom{m\ell-\ell}{m-k} m!(m\ell-m)!$$

Therefore, $N_k = \frac{k-1}{m-1}\binom{\ell}{k}\binom{m\ell-\ell}{m-k} m!(m\ell-m)!$

□

Lemma 9 allows us to bound the numerator and denominator of (7).

LEMMA 10. $\sum_{k=2}^{m^*} kN_k = \frac{\ell(\ell-1)}{m-1}\binom{m\ell-2}{m-2} m!(m\ell-m)!$

PROOF. Using Lemma 9, we have

$$\begin{aligned}
\frac{\sum_{k=2}^{m^*} kN_k}{m!(m\ell-m)!} &= \sum_{k=2}^{m^*} \frac{k(k-1)}{m-1}\binom{\ell}{k}\binom{m\ell-\ell}{m-k} \\
&= \sum_{k=2}^{m^*} \frac{\ell(\ell-1)}{m-1}\binom{\ell-2}{k-2}\binom{m\ell-\ell}{m-k} \\
&= \frac{\ell(\ell-1)}{m-1}\binom{m\ell-2}{m-2}
\end{aligned}$$

□

LEMMA 11. $\sum_{k=2}^{m^*} \ell N_k = \frac{\ell}{m-1}\binom{m\ell-\ell}{m} m!(m\ell-m)!$

PROOF. Using Lemma 9, we have

$$\begin{aligned}
\frac{\sum_{k=2}^{m^*} N_k}{m!(m\ell-m)!} &= \sum_{k=2}^{m^*} \frac{k-1}{m-1}\binom{\ell}{k}\binom{m\ell-\ell}{m-k} \\
&= \sum_{k=1}^{m^*} \frac{k-1}{m-1}\binom{\ell}{k}\binom{m\ell-\ell}{m-k} \\
&= \sum_{k=1}^{m^*} \frac{k\binom{\ell}{k}\binom{m\ell-\ell}{m-k}}{m-1} - \sum_{k=1}^{m^*} \frac{\binom{\ell}{k}\binom{m\ell-\ell}{m-k}}{m-1} \\
&= \sum_{k=1}^{m^*} \frac{\ell\binom{\ell-1}{k-1}\binom{m\ell-\ell}{m-k}}{m-1} - \frac{\binom{m\ell}{m} - \binom{m\ell-\ell}{m}}{m-1} \\
&= \frac{\ell\binom{m\ell-1}{m-1} - \binom{m\ell}{m} + \binom{m\ell-\ell}{m}}{m-1} \\
&= \frac{1}{m-1}\binom{m\ell-\ell}{m}
\end{aligned}$$

□

THEOREM 6. *For any output group $G_{1,m}$, $p(b_i) < \frac{e}{\ell}$.*

PROOF. From Theorem 8, it suffices to analyze $p(b_1)$. We compute $p(b_1)$ from Equation (7) by combining Lemmas 10 11.

$$\begin{aligned}
\ell p(b_1) &= \frac{\sum_{k=2}^{m^*} kN_k}{\sum_{k=2}^{m^*} N_k} = \frac{\ell(\ell-1)\binom{m\ell-2}{m-2}}{\binom{m\ell-\ell}{m}} \\
&= \frac{m(m-1)\ell(\ell-1)(m\ell-2)!(m\ell-\ell-m)!}{(m\ell-m)!(m\ell-\ell)!} \\
&= \frac{(m\ell-2)!(m\ell-\ell-m)!}{(m\ell-m-1)!(m\ell-\ell-1)!} \\
&= \prod_{j=1}^{\ell-1} \frac{m\ell-1-j}{m\ell-m-j} \\
&= \prod_{j=1}^{\ell-1} (1 + \frac{m-1}{m\ell-m-j}) \\
&\leq \prod_{j=1}^{\ell-1} (1 + \frac{m-1}{m\ell-m-(\ell-1)}) \\
&= (1 + \frac{1}{\ell-1})^{\ell-1} < e
\end{aligned}$$

Therefore, we have $p(b_1) < e/\ell$. □