Verification Servers

Karr, Alan F.
National Institute of Statistical Sciences
Research Triangle Park, NC 27709, USA
E-mail: karr@niss.org

Oganyan, Anna
Georgia Southern University,
Statesboro, GA 30458, USA
E-mail: aoganyan@GeorgiaSouthern.edu

Reiter, Jerome
Duke University, Statistical Science
Durham, NC 27708, USA
E-mail: jerry@stat.duke.edu

1 Introduction

Many national statistical agencies disseminate microdata to the public. Doing so benefits society, enabling the research and policy communities to analyze the data in many more ways and to address many more questions than agencies can on their own. Virtually always, however, agencies cannot release microdata as collected (more precisely, as edited and/or imputed), because of confidentiality concerns. Agencies therefore limit what they release, typically by altering the collected data (Willenborg and de Waal, 2001). Most public use data sets released by agencies have undergone at least one form of statistical disclosure limitation (SDL).

Increasing the amount of alteration decreases the risk of disclosure, but also decreases the accuracy of inferences obtained from the released data. Agencies decide on the type and intensity of alterations by evaluating the disclosure risks and analytical usefulness of candidate releases. Often this process is informal, but sometimes selection of the released database represents an explicit tradeoff between quantified measures of disclosure risk and data utility (Duncan et al., 2004; Gomatam et al., 2005b; Reiter, 2005; Karr et al., 2006).

This current practice suffers from a major problem: with most disclosure limitation strategies, it is difficult or impossible for analysts using public datasets to determine how much their particular analysis has been compromised by the data alteration. This is especially problematic when details of the SDL process are suppressed (Cox et al., 2011).

In this paper, we describe *verification servers*—interactive, web-based systems that enable users of to assess the quality of inferences made on public use data subjected to SDL. Details appear in Section 2, but the conceptual nature of a verification server, which is illustrated in Figure 1, is straightforward. Operationally,

1. An analyst performs an analysis of the publicly released data.
2. The analyst submits to the verification server a description of the analysis (example: regress attribute 5 on attributes 1,2, 4 and the logarithm of attribute 6).
3. The verification server performs the analysis on both the confidential data and the publicly released data, and from the results calculates an analysis-specific measure of the fidelity of the latter (known to the analyst) to the former (known only to the agency).
4. The verification server returns to the analyst the value of the fidelity measure.

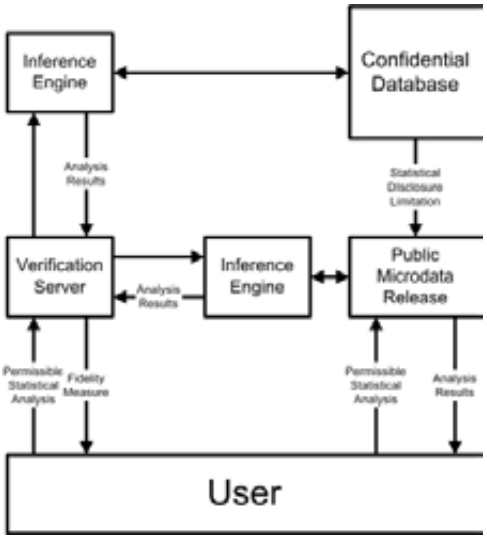

Figure 1: Pictorial depiction of a verification server. The fundamental information provided to users are measures of the fidelity of a statistical analysis performed on the publicly released, disclosure-limited data to the same analysis performed on the underlying confidential data.

Verification servers have multiple potential benefits. Most important, analysts can avoid communicating results with poor quality, or that are possibly even incorrect. For example, an analysis in which the sign of an effect were reversed in the publicly released data from the confidential data, or an effect were significant in the confidential but not the public data, could have scientific or policy consequences that are detrimental to the agency as well as the analyst. In addition, analysts can be confident about results with good quality, and agencies that have provided information about quality are not responsible for inaccurate conclusions reached by analysts. Finally, a verification server serves as a data collection mechanism for the agency, capturing what attributes and analysts care most about, enabling them to improve the quality of future data releases.

But of course verification servers are not a panacea. Improving data quality incurs risk. As we illustrate in Section 3, measures of data quality can provide ill-intentioned users, henceforth called *intruders*, with information for disclosure attacks. We describe some approaches to reducing the risks associated with verification servers in Section 4.

2 Defining Verification Servers

Here we define verification servers and describe an illustrative fidelity measure.

2.1 Formal Description

Consider an agency holding an *original*, confidential database $\mathcal{O}$, from which it constructs a *masked* database $\mathcal{M}$ to be released to the public, using one or more SDL strategies. We assume limited transparency (Cox et al., 2011): the agency reveals at most broad information about the SDL strategy.

Consider an analyst seeking to make inferences about some quantity Q, such as a confidence interval for a regression coefficient. Let $Q(\mathcal{M})$ be the estimate of Q obtained from using $\mathcal{M}$, and let $Q(\mathcal{O})$ be the estimate of Q obtained from using $\mathcal{O}$. The analyst can compute only $Q(\mathcal{M})$. Because of the SDL, we expect that $Q(\mathcal{M}) \neq Q(\mathcal{O})$ for most Q, although this is not necessarily the case.

The agency wants to inform analysts about the differences between $Q(\mathcal{M})$ and $Q(\mathcal{O})$. Given instructions from the analyst, the agency could manually compute $Q(\mathcal{M})$ and $Q(\mathcal{O})$ and describe the

differences to the analyst. However, this process is both prohibitively labor-intensive, as well as fail to address central issues such as the risks arising from query interaction.

A verification server is a web-based system that analysts query for measures of the quality of their particular $Q(\mathcal{M})$. Let $\mathbf{FM}(a, b)$ represent a fidelity measure: a numerical summary comparing a to b, where a and b are the results of the "same" statistical analysis on different databases. In general, a, b and $\mathbf{FM}$ may be vector-valued, although here we focus on scalar fidelity measures. The verification server reports $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$, or an approximation to it, to the analyst.

One may ask why the server does not report $Q(\mathcal{O})$ directly, as is done, for example, by remote analysis servers (Karr et al., 2001; Dobra et al., 2002, 2003; Reiter, 2003). Verification servers have advantages over remote analysis servers. As argued in Section 3, providing infinitely precise information about analyses of $\mathcal{O}$, whether in the form of $Q(\mathcal{O})$ or $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$, in combination with $Q(\mathcal{M})$, can lead to high disclosure risks, especially when multiple, nearly identical queries are permitted. Arguably, it is easier to coarsen $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ than $Q(\mathcal{O})$: fidelity measures may in some sense be qualitative. By contrast, altering $Q(\mathcal{O})$ defeats the purpose of providing it. Additionally, remote analysis servers must limit what and how much output is released, since clever queries can reveal individual data values (Gomatam et al., 2005a).

2.2 Examplar Fidelity Measure

Existing measures of data utility in SDL contexts typically fall into one of two categories. The first are global but blunt measures of the difference between $\mathcal{M}$ and $\mathcal{O}$, such as Hellinger distances (Gomatam et al., 2005b), Kullback–Liebler distances (Karr et al., 2006) and model-based propensity scores (Woo et al., 2009). As the distance between the distributions grows, the overall quality of $\mathcal{M}$ drops, although the consequences may vary dramatically for specific analyses of the data. The second set of measures are comparisons of specific analyses—in most cases, fitting models—between the original and released data. We focus on these, using the confidence interval overlap and confidence ellipsoid overlap for multiple model coefficients measures of Karr et al. (2006), as well as measures associated with categorical/tabular data (Dobra et al., 2002, 2003).

Let e be, for simplicity, a scalar estimand. The verification server first calculates 95% confidence intervals for e from the masked data $\mathcal{M}$: $Q(\mathcal{M}) = (L_{\mathcal{M}}, U_{\mathcal{M}})$ and the confidential data: $Q(\mathcal{O}) = (L_{\mathcal{O}}, U_{\mathcal{O}})$. The verification server calculates the intersection of these two intervals, which we denote by (L_i, U_i). The fidelity measure is then

$$\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) = \frac{U_i - L_i}{2(U_{\mathcal{O}} - L_{\mathcal{O}})} + \frac{U_i - L_i}{2(U_{\mathcal{M}} - L_{\mathcal{M}})}. \quad (1)$$

When the intervals are nearly identical, corresponding to high utility, $\mathbf{FM} \approx 1$. When the intervals do not overlap, corresponding to low utility, $\mathbf{FM} = 0$. The second term in (1) is included to differentiate between intervals with $(U_i - L_i)/(U_{\mathcal{O}} - L_{\mathcal{O}}) = 1$, but different lengths. For example, for two masked data intervals that fully contain the collected data interval, the measure (1) favors the shorter interval.

3 Risks Associated with Verification Servers

We have noted already that $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ provides intruders with additional information to support disclosure attacks. In this section, we describe how intruders might utilize such information when $\mathcal{M}$ is constructed by means of common SDL strategies.

For concreteness, consider a survey such as the US National Health Interview Survey (National Center for Health Statistics, 2008) in which data on individuals are collected. Let X_j represent the vector of quasi-identifiers for unit j; examples are age, race, gender, marital status, and geography.

We assume that these are known without error by the intruder for selected records in the database, either because the information is "public" or because it can be purchased from the owner of some external database. Let Y_j be the vector of sensitive attributes collected in the survey for unit j, such as health or other personal variables, which are not known by the intruder.

The scenarios that follow illustrate attacks that can defeat common SDL strategies completely, given a verification server permitting all queries and responding with infinite precision. Some of these attacks, however, entail significant computational burdens.

3.1 Data Swapping

Suppose that for a small percentage of records the X_j are swapped as a group, but the Y_j are left alone. The agency does not reveal which or even how many records were swapped. Given infinitely precise fidelity measures, the intruder can undo many swapping protections. Note that in this case, $\mathcal{M}$ and $\mathcal{O}$ contain the same number of records, and each record in $\mathcal{M}$ has a unique "parent" in $\mathcal{O}$.

Identifying which records were swapped. The intruder can use a trial and error approach to determine which records underwent swapping. The intruder submits a query for the confidence interval for the slope in the regression of one variable in Y on a subset of the records. If $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) = 1$, with very high probability the values of X_j for the records involved in the query are the original values. When $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) \neq 1$, at least one record involved in the query was swapped. The intruder can isolate the swapped records by submitting multiple queries based on different subsets of records.

Determining original values for swapped records. Once the set of swapped records is identified, intruders can determine the original values for those records. To illustrate, suppose the intruder seeks the actual value of marital status for a target record with swapped value of marital status equal to "married." First, the intruder selects a set of unswapped records with sufficient numbers in each marital status category. Second, the intruder appends the target record to this unswapped set. Third, using the appended data in the query, the intruder asks for the fidelity measures for the proportion of people in each marital status category. Any marital status category for which $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) = 1$ is eliminated as a candidate for the target's true marital status. Only two categories have $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) \neq 1$. The target's true marital status is the remaining marital status not equal to the swapped value, i.e., the one other than "married" in our example.

As another and generalizable, albeit not very scalable, attack strategy, the intruder could construct every possible value $\mathcal{O}^*$ for $\mathcal{O}$, in this case simply by permuting X_j. Assuming that the intruder knows how the fidelity measure **FM** is calculated—otherwise the whole concept of a verification server is undermined—the intruder can compute $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*))$ for all $\mathcal{O}^*$ and an enormous number of analyses Q, and compare these to values of $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ from the verification server. If the number of analyses is large enough, there will be only one $\mathcal{O}^*$ for which

$$\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*)) = \mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) \quad (2)$$

for all Q, which must then be $\mathcal{O}$.

3.2 Top-coding

Suppose that one numerical component of Y_j is protected by top-coding: large values of this variable are reported only as exceeding a threshold t. Infinitely precise fidelity measure values allow the intruder to undo many of the top-coding protections. In our examples, we assume that the intruder computes $Q(\mathcal{M})$ by setting the top-coded values equal to t.

Rank order top-coded values. The intruder can utilize the fidelity measure to order the top-coded records from smallest to largest values of the unobserved Y. First, the intruder identifies

a subset of n records not subject to the top-coding. Second, for some record j subject to top-coding, the intruder appends the record to this subset. Third, the intruder asks for fidelity measures for the mean of Y based on the $n+1$ records in this query. The intruder repeats the second and third steps of this process for each top-coded record, each time using the same n records not subject to top-coding. Finally, the intruder orders the values of $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ from smallest to largest. This ordering matches the ranking of the unobserved Y, since $Q(\mathcal{M})$ declines more in quality the greater the "degree" of top-coding.

Determine values of top-coded records. The strategy used in Section 3.2 can be modified to learn exact values of top-coded records. As before, the intruder obtains $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ for a subset of n records with $Y < t$ and one top-coded record j. Then, the intruder guesses a plausible value of the true Y_j for that record, using that guess to make a proposed true database, $\mathcal{O}^*$, for those $n+1$ records. The intruder computes $FM(Q(\mathcal{M}), Q(\mathcal{O}^*))$. The intruder repeats this process many times using different initial guesses of the true Y_j. The guess that results in $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*)) = \mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ and exceeds t is the true value for that record.

3.3 Additive Noise

With clever transformations, the intruder can estimate the actual values of variables protected by additive noise. For example, the intruder submits a query based on n records in which all values but the one for record j are transformed to equal the same number. The intruder obtains $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$ for that query. Then, the intruder guesses a plausible value of the true Y_j, using that guess to make a proposed true database, $\mathcal{O}^*$, for those n records. The intruder computes $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*))$. The intruder repeats this process many times using different initial guesses of the true Y_j. The guesses that satisfy (2) are candidates for the true value. Some of these may be more plausible than the others, given other information released in the data.

3.4 Synthetic Data

The attack strategies in Sections 3.1–3.3 can be applied to partially synthetic data, for which there is a one-to-one correspondence between $\mathcal{O}$ and each version of $\mathcal{M}$. As an example, to learn the original value Y_j for a record with Y synthesized, the analyst appends this record to a set of n records whose values of Y are not synthesized. The intruder guesses a plausible value of the true Y_j, using that guess to make a proposed true data base, $\mathcal{O}^*$, for those $n+1$ records. The intruder computes $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*))$. The intruder repeats this process many times using different initial guesses of the true Y_j. The guesses that satisfy (2) are candidates for the true value. This attack is not possible when all values of the variable are synthesized.

4 Reducing Risk

The factors driving the risks described in Section 3 are:

Subsetting of data, which in effect allows individual records to be moved into and out of queries Q, and attribute values to be deduced from differences in $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$.

Infinite precision fidelity measures, which make attacks based on (2) possible.

Transformations of attributes, which are already known to be a major threat (Gomatam et al., 2005a; Karr et al., 2006).

In this section we present approaches for reducing risk by limiting the query space of the verification server and restricting transformations of attributes (Section 4.1), and by reducing the precision

of fidelity measures (Section 4.2).

4.1 Limiting the Query Space

A verification server need not respond to all conceivable queries. Certain attacks can be made more difficult with query restrictions that have minimal impact on legitimate data users.

Require well-defined target populations. The verification server can require that records in any queries comprise well-defined sub-populations rather than arbitrary subsets. For example, the verification server can require selection criteria to be functions with limited complexity, or can force the analyst to specify the subset selection criteria rather than a collection of record identifiers.

Disallowing certain transformations. Some transformations, such as those described in Section 3, are not part of many legitimate applications, but are powerful tools for disclosure attacks. It may be possible to prevent some of these transformations, for instance by allowing only a set of standard transformations, such as polynomials and low powers, and disallowing all others. However, because such restrictions may also preclude legitimate uses.

Query interaction. The collective risk of a set queries may exceed the "sum" of their individual risks. Issues of query interaction are understood for some classes of remote analysis servers—specifically, table servers (Dobra et al., 2002, 2003) and, to a lesser extent, regression servers (Gomatam et al., 2005a). They exist as well for verification servers; indeed, some of the intruder strategies described in Section 3 specifically exploit query interaction. Strategies developed for remote analysis servers may be adaptable to verification servers, but this requires further investigation.

4.2 Reducing Precision of Fidelity Measures

Coarsening fidelity measures creates uncertainty in all of the attacks, because the true fidelity measure is not available for use in attacks based on (2). Here we describe approaches to coarsening fidelity measures. The goal is provide the analyst with sufficient information to decide whether $Q(\mathcal{M})$ is "good enough" compared to $Q(\mathcal{O})$ for such purposes as publication or policy.

Altering responses to queries is an SDL strategy largely absent from the statistical literature, but has been studied by computer scientists and cryptographers, for example under the rubric of differential privacy (Dwork, 2007; McSherry and Talwar, 2007).

Interval reporting. Rather than report precise values of fidelity measures, a verification server can report measures by intervals, for example, confidence interval overlap of between 90% and 100% or between 80% and 90%. Such reporting may be sufficient to enable analysts to evaluate quality, yet provide enough uncertainty to mask true values. In particular, if the verification server never reports that $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O})) = 1$ precisely, then some attacks described in Section 3 can be thwarted.

However, interval reporting does not prevent all attacks. For example, when seeking to order top-coded records, an intruder can determine which record owns the largest value when the interval with the least overlap (e.g., 10% to 20% has smaller overlap than 30% to 40%) is unique. Moreover, intruders can use trial and error attacks to obtain ranges of possible values for unknown Y. For example, to get a range for a particular top-coded value Y_j, the intruder appends that record to a collection of n other records that are not top-coded. The intruder submits queries about the mean of Y based on those $n + 1$ records, obtaining the reported interval $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}))$. The intruder then proceeds as follows:

A1. Set f equal to the lower bound of the reported interval for $Q(\mathcal{M})$.

A2. Find the value Y_j^* that, when used to make a candidate $\mathcal{O}^*$ for $\mathcal{O}$, produces $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*))^* = f$, where $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*))^*$ is a single number measurement of the fidelity measure. Store

this value of Y_j^*.

A3. Set $f = f + c$, where c is some small number.

A4. Repeat Step 2 and 3 until f equals the upper bound of the reported interval for $Q(\mathcal{M})$.

The stored values Y_j^* are the plausible values for the original Y_j. When this distribution does not have sufficient variability, an inferential disclosure occurs.

To illustrate, we use data from the 1995 US Current Population Survey (CPS). The variables include adjusted gross income (AGI) and interest income (INTVAL). The intruder submits a query for the fidelity measure for the intercept in a regression of AGI on INTVAL. The regression is based on a set of 31 records, one of which has AGI top-coded. The intruder seeks to estimate the original value of the top-coded AGI. Figure 2 displays graphically the output from the attack protocol described above. The curved line connects the values of $\mathbf{FM}(Q(\mathcal{M}), Q(\mathcal{O}^*))^*$ for several guesses at the original AGI. If the top-coded record is such that the verification server reports an interval of (0.8, 0.9), the intruder knows that the original AGI is between \$152,082 and \$217,318. If the top-coded record is such that the server reports (0.6, 0.7), the original AGI is between \$299,950 and \$421,723.

Adding noise. An alternative approach is to add random noise to fidelity measures before reporting them. The amount and distribution of the noise are hidden from intruders in order to provide less information for backsolving attacks. The noise value is the same for all requests for the same $Q(\mathcal{M})$, in order to eliminate intruders' ability to sharpen estimates by averaging the results of multiple queries for the same analysis. But, the noise must differ by analysis in order to reduce the chance that the intruder can guess the value of the added noise for some queries, for example, if the intruder knows $Q(\mathcal{O})$ and submits $Q(\mathcal{M})$ anyway. These two criteria can be met by linking the random seed to some function of $Q(\mathcal{O})$ that provides unique seeds for almost all different queries.

Computing on different databases. As yet another approach, a verification server can report fidelity measures based on databases that differ slightly from both $\mathcal{O}$ and $\mathcal{M}$, in ways not revealed to users. As an example, the verification server can do the following.

B1. Delete k randomly sampled records from the data used to compute $Q(D)$. Let r_d and r_n be the row numbers of the deleted records (r_d) and the not deleted records (r_n). Let $\mathcal{O}_{r_n}$ and $\mathcal{M}_{r_n}$ be the data in $\mathcal{O}$ and $\mathcal{M}$ for the records in r_n.

B2. Sample k row numbers from r_d, with replacement. Let r_s be the sampled row numbers. Let $\mathcal{O}_{r_s}$ and $\mathcal{M}_{r_s}$ be the data in $\mathcal{O}$ and $\mathcal{M}$ for the records in r_s.

B3. Construct $\mathcal{O}' = (\mathcal{O}_{r_n}, \mathcal{O}_{r_s})$ and $\mathcal{M}' = (\mathcal{M}_{r_n}, \mathcal{M}_{r_s})$.

B4. Report $\mathbf{FM}(Q(\mathcal{M}'), Q(\mathcal{O}'))$ to users.

With large k, this creates a combinatorial explosion of possible true values for the records in $\mathcal{O}$, potentially rendering attacks based on (2) infeasible computationally.

This approach is not immune to disclosure risks in verification servers, although the intruder must work hard to guess original values sensibly. First, the intruder proposes a possible value of $\mathcal{O}$, say $\mathcal{O}^*$. Second, the intruder proposes possible values of $\mathcal{O}'$ and $\mathcal{M}'$, say $\mathcal{O}'^*$ and $\mathcal{M}'^*$, obtained by mimicking steps B1–B3 on $\mathcal{O}^*$ and $\mathcal{M}$. Third, the intruder computes $\mathbf{FM}(Q(\mathcal{M}'^*), Q(\mathcal{O}'^*))$, repeating this process many times, collecting all $\mathcal{O}'^*$ for which $\mathbf{FM}(Q(\mathcal{M}'^*), Q(\mathcal{O}'^*)) = \mathbf{FM}(Q(\mathcal{M}'), Q(\mathcal{O}'))$. The values of the targeted Y_j among the $\mathcal{O}^*$ meeting this criterion are plausible values of the original Y_j. If the distribution does not have sufficient variability, or if some obvious function such as the mean/median of the plausible values is too close to the truth, there may be a disclosure risk.

Figure 3 illustrates this attack strategy when there is top-coded value of AGI. The intruder's query is based on n records with $Y < t$ and one record subject to top-coding. To draw any one

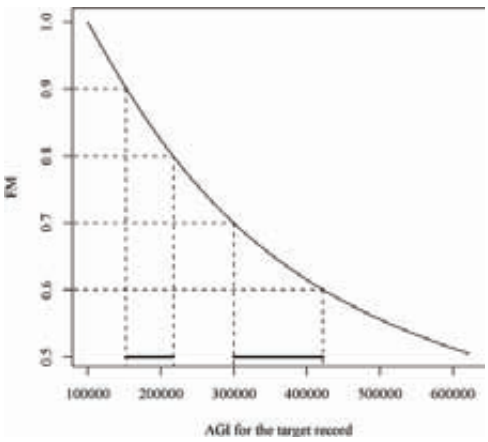

Figure 2: Illustration of intruder's bounding procedure when the verification server reports interval fidelity measures. The solid horizontal lines are ranges of plausible values for original AGIs when the reported interval is (0.6, 0.7) or (0.8, 0.9).

curve, we assume that the intruder knows k and follows steps B1–B3 only with $\mathcal{M}$ to get $\mathcal{M}'^*$. For all top-coded records in $\mathcal{M}'^*$, we replace t with a guess of the original AGI, thus obtaining a $\mathcal{O}'^*$. We then compute $\mathbf{FM}(Q(\mathcal{M}'^*), Q(\mathcal{O}'^*))$. We repeat the last two steps for different guesses, and connect values of $\mathbf{FM}(Q(\mathcal{M}'^*), Q(\mathcal{O}'^*))$ to make the curve. The figure shows 50 such curves derived from different records in $\mathcal{M}'^*$. The intruder draws a horizontal line at the reported $\mathbf{FM}(Q(\mathcal{M}'), Q(\mathcal{O}'))$. Its intersections with the curves gives the distribution of plausible values for the original datum. For one realization of D' we obtained $\mathbf{FM}(Q(\mathcal{M}'), Q(\mathcal{O}')) = 0.87$. For this D', the intruder can average the plausible AGI values to get very close to the original value. However, for another realization of $\mathcal{O}'$ we obtained $\mathbf{FM}(Q(\mathcal{M}'), Q(\mathcal{O}')) = 0.77$, for which the average of the plausible AGI values is not a close estimate of the original value. Although 0.87 and 0.77 do not seem to differ significantly in terms of data quality, Figure 3 indicates that protection is sensitive to which units are in $\mathcal{O}'$.

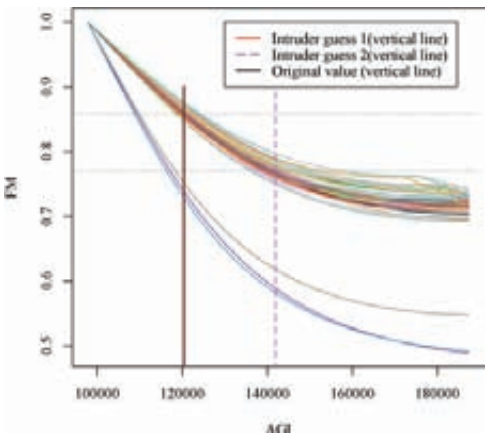

Figure 3: **FM** curves constructed by the intruder to defeat the *delete-add* strategy of the verification server.

5 Concluding Remarks

Verification servers could have enormous benefit for statistical agencies and consumers of their data. Arguably, with increasing amounts of protection applied to microdata before release, verification servers are crucial for the future existence of useful microdata. However, as we demonstrated, releasing precise quality measures can pose disclosure risks.

Acknowledgements

This research was supported by NSF grant IIS–0131884 to the National Institute of Statistical Sciences. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES (RÉFERENCES)

References

Cox, L. H., Karr, A. F., and Kinney, S. K. (2011). Risk-utility paradigms for statistical disclosure limitation: How to think, but not how to act (with discussion). *Int. Statist. Rev.* To appear.

Dobra, A., Fienberg, S. E., Karr, A. F., and Sanil, A. P. (2002). Software systems for tabular data releases. *Int. J. Uncertainty, Fuzziness and Knowledge Based Systems*, 10(5):529–544.

Dobra, A., Karr, A. F., and Sanil, A. P. (2003). Preserving confidentiality of high-dimensional tabular data: Statistical and computational issues. *Statist. and Computing*, 13(4):363–370.

Duncan, G. T., Keller-McNulty, S. A., and Stokes, S. L. (2004). Disclosure risk vs. data utility: The R–U confidentiality map. *Management Sci.* Under revision.

Dwork, C. (2007). Differential privacy. Available on-line at research.microsoft.com/research/sv/DatabasePrivacy/dwork.pdf.

Gomatam, S., Karr, A. F., Reiter, J. P., and Sanil, A. P. (2005a). Data dissemination and disclosure limitation in a world without microdata: A risk-utility framework for remote access analysis servers. *Statist. Sci.*, 20(2):163–177.

Gomatam, S., Karr, A. F., and Sanil, A. P. (2005b). Data swapping as a decision problem. *J. Official Statist.*, 21(4):635–656.

Karr, A. F., Kohnen, C. N., Oganian, A., Reiter, J. P., and Sanil, A. P. (2006). A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician*, 60(3):224–232.

Karr, A. F., Lee, J., Sanil, A. P., Hernandez, J., Karimi, S., and Litwin, K. (2001). Disseminating information but protecting confidentiality. *IEEE Computer*, 34(2):36–37.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *Foundations of Computer Science, 2007*, pages 94–103, Providence, RI. IEEE.

National Center for Health Statistics (2008). National Health Interview Survey (NHIS). Information available on-line at www.cdc.gov/nchs/nhis.htm.

Reiter, J. P. (2003). Model diagnostics for remote access regression servers. *Statistics and Computing*, 13:371–380.

Reiter, J. P. (2005). Estimating identification risks in microdata. *J. Amer. Statist. Assoc.*, 100:1101–1113.

Willenborg, L. C. R. J. and de Waal, T. (2001). *Elements of Statistical Disclosure Control*. Springer–Verlag, New York.

Woo, M.-J., Reiter, J. P., Oganian, A., and Karr, A. F. (2009). Global measures of data utility for microdata masked for disclosure limitation. *J. Privacy and Confidentiality*, 1(1):111–124.

RÉSUMÉ (ABSTRACT)

To protect confidentiality, statistical agencies typically alter data before releasing them to the public. Only rarely do agencies also provide a way for data analysts to assess the quality of inferences obtained from the released data. We propose an interactive, web-based system that analysts can query for measures of inferential quality. We illustrate potential disclosure risks of the system and propose methods for limiting these risks.