# Secure Statistical Analysis of Distributed Databases

Alan F. Karr, Xiaodong Lin,[*] Ashish P. Sanil[†]
National Institute of Statistical Sciences
Research Triangle Park, NC 27709–4006, USA
karr@niss.org, linxd@samsi.info, ashish@niss.org

Jerome P. Reiter
Duke University
Durham, NC 27708 USA
jerry@stat.duke.edu

July 1, 2005

## 1 Introduction

A continuing need in the contexts of homeland security, national defense and counterterrorism is for statistical analyses that "integrate" data stored in multiple, distributed databases. There is some belief, for example, that integration of data from flight schools, airlines, credit card issuers, immigration records and other sources might have prevented the terrorist attacks of September 11, 2001, or might be able to prevent recurrences.

In addition to significant technical obstacles, not the least of which is poor data quality [32, 31], proposals for large-scale integration of multiple databases have engendered significant public opposition. Indeed, the outcry has been so strong that some plans have been modified or even abandoned. The political opposition to "mining" distributed databases centers on deep, if not entirely precise, concerns about the privacy of database subjects and, to a lesser extent, database owners. The latter is an issue, for example, for databases of credit card transactions or airline ticket purchases. Integrating the data without protecting ownership could be problematic for all parties: the companies would be revealing who their customers are, and where a person is a customer would also be revealed.

For many analyses, however, it is not necessary actually to integrate the data. Instead, as we show in this paper, using techniques from computer science known generically as secure multi-party computation, the database holders can share analysis-specific sufficient statistics anonymously, but in a way that the desired analysis can be performed in a principled manner. If the sole concern is protecting the source rather than the content of data elements, it is even possible to share the data themselves, in which case *any* analysis can be performed.

The same need arises in non-security settings as well, especially scientific and policy investigations. For example, a regression analysis on integrated state databases about factors influencing student performance

[*] Now at University of Cincinnati, Cincinnati, OH.
[†] Now at Bristol-Myers Squibb, Princeton, NJ.

would be more insightful than individual analyses, or complementary to them. Yet another setting is proprietary data: pharmaceutical companies might all benefit, for example, from a statistical analysis of their combined chemical libraries, but do not wish to reveal which chemicals are in the libraries [25].

The barriers to integrating databases are numerous. One is confidentiality: the database holders—we term them "agencies"—almost always wish to protect the identities of their data subjects. Another is regulation: agencies such as the Census Bureau (Census) and Bureau of Labor Statistics (BLS) are largely forbidden by law to share their data, even with each other, let alone with a trusted third party. A third is scale: despite advances in networking technology, there are few ways to move a terabyte of data from point A today to point B tomorrow.

In this paper we focus on linear regression and related analyses. The regression setting is important because of its prediction aspect; for example, vulnerable critical infrastructure components might be identified using a regression model. We begin in §2 with background on data confidentiality and on secure multi-party computation. Linear regression is treated for "horizontally partitioned data" in §3 and for "vertically partitioned data" in §4. Two methods for secure data integration and an application to secure contingency tables appear in §5, and a concluding discussion in §6.

Various assumptions are possible about the participating parties, for example, whether they use "correct" values in the computations, follow computational protocols or collude against one another. The setting in this paper is that of agencies wishing to cooperate but to preserve the privacy of their individual databases. While each agency can "subtract" its own contribution from integrated computations, it should not be able to identify the other agencies' contributions. Thus, for example, if data are pooled, an agency can of course recognize data elements that are not its own, but should not be able to determine which other agency owns them. In addition, we assume that the agencies are "semi-honest:" each follows the agreed-on computational protocols, but may retain the results of intermediate computations.

# 2 Background

In this section we present background from statistics (§2.1) and computer science (§2.2).

## 2.1 Data Confidentiality

From a statistical perspective, the problem we treat lies historically in the domain of data confidentiality or, in the context of official statistics, statistical disclosure limitation [12, 43, 44]. The fundamental dilemma is that government statistical agencies are charged with the inherently conflicting missions of both protecting the confidentiality of their data subjects and disseminating useful information derived from their data—to Congress, other federal agencies, the public and researchers.

In broad terms, two kinds of disclosures are possible from a database of records containing attributes of individuals or establishments. An "identity disclosure" occurs when a record in the database can be associated with the individual or establishment that it describes even if the record does not contain explicit identifiers. An "attribute disclosure" occurs if the value of a sensitive attribute, such as income or health status, is disclosed. This may be an issue even without identity disclosure; for instance, if a doctor is known to specialize in treating AIDS, then attribute disclosure (AIDS) may occur for his or her patients. Attribute disclosure is often inferential in nature, and may not be entirely certain. It is also highly domain-dependent.

To prevent identity disclosures, agencies remove explicit identifiers such as name and address or social security number, as well as implicit identifiers, such as "Occupation = Mayor of New York." Often, however, this is not enough. Technology poses new threats, through the proliferation of databases and software to

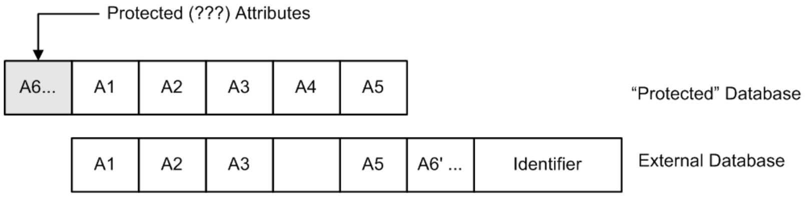

Figure 1: Pictorial representation of record linkage. The upper record, in the purported protected database is linked to a record in the external database that has the same values of attributes A1, A2, A3 and A5, but also contains an identifier. If only one record in the external database matches, then the value of A6 is known for the subject of that record. In practice, surprisingly few attributes are needed.

link records across databases. Record linkage, which is shown pictorially in Figure 1, produces identity disclosures by matching a record in the database to a record in another database containing some of the same attributes as well as identifiers. In one well-known example [41], only three attributes—date of birth, 5-digit ZIP code of residence and gender—produced identity disclosures from a medical records database by linkage to public voter registration data.

Identity disclosure can also occur by means of rare or extreme attribute values. For example, female Korean dentists in North Dakota are rare, and an intruder—the generic term for a person attempting to break confidentiality—could recognize such a record, or a family member may recognize another family member from household characteristics, or an employer could recognize an employee from salary, tenure and geography. Establishments (typically, corporations and other organizations) are especially vulnerable in data at high geographical resolution. The largest employer in a county is almost always widely known, so that county-level reporting of both numbers of employees and health benefits expenditures does not protect the latter.

There is a wealth of techniques [9, 16, 24, 43, 44] for "preventing" disclosure. In general, these techniques preserve low-dimensional statistical characteristics of the data, but distort disclosure-inducing high-dimensional characteristics. *Aggregation*—especially geographical aggregation [27, 33]—is a principal strategy to reduce identity disclosures. The Census and several other federal agencies do not release data at aggregations less than 100,000. Another is *top-coding*: for example, all incomes exceeding $10,000,000 could be lumped into a single category. *Cell suppression* is the outright refusal to release risky—usually, small count—entries in tabular data. *Data swapping* interchanges the values of one or more attributes, such as geography, between data records. *Jittering* adds random noise to values of attributes such as income. *Microaggregation* groups numerical data records into small clusters and replace all elements of each cluster by their (component-wise) average [5, 6]. Even entirely *synthetic databases* may be created, which preserve some characteristics of the original data, but whose records simply do not correspond to real individuals or establishments [13, 35, 36]. Analysis servers [19], which disseminate analyses of data rather than data themselves, are another alternative, as is the approach described in this paper.

Much current research focuses on explicit disclosure risk–data utility formulations for statistical disclosure limitation problems [7, 8, 14, 15, 20, 26, 42]. These enable agencies to make explicit tradeoffs between risk and utility.

### 2.2 Secure Multi-Party Computation

The generic secure multi-party computation problem [17, 18, 45] concerns agencies $1, \ldots, K$ with values $v_1, \ldots, v_K$ that wish to compute a known function $f(v_1, \ldots, v_K)$ in such a manner that no agency $j$ learns no more about the other agencies' values than can be determined from $v_j$ and $f(v_1, \ldots, v_K)$. In practice, absolute security may not be possible, so some techniques for secure multi-party computation rely on heuristics [11] or randomization.

The simplest secure multi-party computation, and the one used in §3 for secure regression, is to sum values $v_j$ held by the agencies: $f(v_1, \ldots, v_K) = \sum_{j=1}^{K} v_j$. Let $v$ denote the sum. The secure summation protocol [2], which is depicted graphically in Figure 2, is straightforward in principle, although a "production quality" implementation presents many challenges. Number the agencies $1, \ldots, K$. Agency 1 generates a very large random integer $R$, adds $R$ to its value $v_1$, and sends the sum to agency 2. Since $R$ is random, Agency 2 learns effectively nothing about $v_1$. Agency 2 adds its value $v_2$ to $R + v_1$, sends the result to agency 3, and so on. Finally, agency 1 receives $R + v_1 + \ldots + v_K = R + v$ from agency $K$, subtracts $R$, and shares the result $v$ with the other agencies. Here is cooperation matters: agency 1 is obliged to share $v$ with the other agencies.

Figure 2 contains an extra layer of protection. Suppose that $v$ is known to lie in the range $[0, m)$, where $m$ is a very large number, say $2^{100}$, that is known to all the agencies. Then $R$ can be chosen randomly from $\{0, \ldots, m - 1\}$ and all computations performed modulo $m$.

To illustrate, suppose that the agencies have income data and wish to compute the global average income. Let $n_j$ be the number of records in agency $j$'s database and $I_j$ be the sum of their incomes. The quantity to be computed is

$$\bar{I} = \frac{\sum_j I_j}{\sum_j n_j},$$

whose numerator can be computed using secure summation on the $I_j$'s, and whose denominator can be computed using secure summation on the $n_j$'s.

This method for secure summation faces an obvious problem if, contrary to our assumption, some agencies were to collude. For example, agencies $j - 1$ and $j + 1$ can together compare the values they send and receive to determine the exact value of $v_j$. Secure summation can be extended to work for an honest majority: each agency divides $v_j$ into shares, and secure summation is used to calculate the sum for each share individually. However, the path used is altered for each share so that no agency has the same neighbor twice. To compute $v_j$, the neighbors of agency $j$ from every iteration would have to collude.

# 3 Horizontally Partitioned Data

As the name connotes, this is the case where the agencies have the same attributes on disjoint sets of data subjects [28, 30]. Examples include state-level drivers license databases and data on individuals held by their countries of citizenship.

### 3.1 The Computations

We assume that there are $K > 2$ agencies, each with the same numerical data on its own $n_j$ data subjects—$p$ predictors $X^j$ and a response $y^j$, and that the agencies wish to fit the usual linear model

$$y = X\beta + \varepsilon, \tag{1}$$

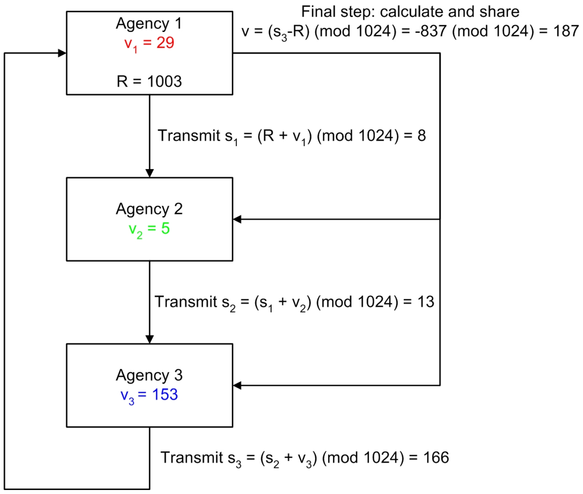
Final step: calculate and share
v = (s3-R) (mod 1024) = -837 (mod 1024) = 187
Agency 1
v1 = 29
R = 1003
Transmit s1 = (R + v1) (mod 1024) = 8
Agency 2
v2 = 5
Transmit s2 = (s1 + v2) (mod 1024) = 13
Agency 3
v3 = 153
Transmit s3 = (s2 + v3) (mod 1024) = 166

Figure 2: Values computed at each agency during secure computation of a sum initiated by Agency 1. Here $v_1 = 29$, $v_2 = 5$, $v_3 = 152$ and $v = 187$. All arithmetic is modulo $m = 1024$.

to the "global" data

$$X = \begin{bmatrix} X^1 \\ \vdots \\ X^K \end{bmatrix} \qquad \text{and} \qquad y = \begin{bmatrix} y^1 \\ \vdots \\ y^K \end{bmatrix}. \tag{2}$$

Figure 3 shows such horizontal partitioning for $K = 3$ agencies. Each $X^j$ is $n_j \times p$.

We embed the constant term of the regression in the first predictor: $X_1^j \equiv 1$ for all $j$. To illustrate the subtleties associated with distributed data, the usual strategy of centering the predictors and response at their means does not work directly, at least not without another round of secure computation. The means needed are the global—not the local—means, which are not available.[1]

Under the condition that

$$\mathrm{Cov}(\varepsilon) = \sigma^2 I, \tag{3}$$

the least squares estimator for $\beta$ is of course

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{4}$$

To compute $\hat{\beta}$ without data integration, it is necessary to compute $X^T X$ and $X^T y$. Because of the horizontal partitioning of the data in (2),

$$X^T X = \sum_{j=1}^{K} (X^j)^T X^j. \tag{5}$$

Therefore, agency $j$ simply computes its own $(X^j)^T X^j$, a local sufficient statistic that has dimensions $p \times p$, where $p$ is the number of predictors, and these are combined entrywise using secure summation. This computation is illustrated with $K = 3$ in Figure 3. Of course, because of symmetry, only $\binom{p}{2} + p$

[1] They could, of course, be computed using secure summation, as in the average income example in §2.2.

secure summations are needed. Similarly, $X^T y$ can be computed by secure, entry-wise summation of the $(X^j)^T y^j$.

Finally, each agency can calculate $\hat{\beta}$ from the shared values of $X^T X$ and $X^T y$. Note that no agency learns any other agency's $(X^j)^T X^j$ or $(X^j)^T y^j$, but only the sum of these over all the other agencies.

The least squares estimator $S^2$ of $\sigma^2$ in (3) also can be computed securely. Since

$$S^2 = \frac{(y - X\hat{\beta})^T (y - X\hat{\beta})}{n - p}, \tag{6}$$

and $X^T X$ and $\hat{\beta}$ have been computed securely, the only thing left is to compute $n$ and $y^T y$ using secure summation.

With this method for secure regression, each agency $j$ learns the global $X^T X$ and $X^T y$. This creates a unilateral incentive to "cheat:" if $j$ contributes a false $(X^j)^T X^j$ and $(X^j)^T y^j$ but every other agency uses its real data, then $j$ can recover

$$\sum_{i \neq j} (X^i)^T X^i$$

and

$$\sum_{i \neq j} (X^i)^T y^i,$$

and thereby the regression for the other agencies, correctly. Every other, agency, by contrast, ends up with an incorrect regression. Research on means of preventing this is under way at the National Institute of Statistical Sciences (NISS). Exactly what about an agency's database is learned from one regression—and whether that regression compromises individual data elements—requires additional research.

Virtually the same technique can be applied to any model for which "sufficient statistics" are additive over the agencies. One such example is generalized linear models of the form (1), but with $\Sigma = \text{Cov}(\varepsilon)$ not a diagonal matrix. The least squares estimator for $\beta$ in the GLM is

$$\beta^* = (X^T \Sigma^{-1} X)^{-1} X^T \Sigma^{-1} y,$$

which can be computed using secure summation, provided that $\Sigma$ is known to all the agencies. Exactly how $\Sigma$ would be known to all the agencies is less clear.

Another example is linear discriminant analysis [22]; extension to other classification techniques also remains a topic for future research.

## 3.2 Example

We illustrate the secure regression protocol of §3.1 using the "Boston housing data" [21]. There are 506 data cases, representing towns around Boston, which we partitioned, purely for illustrative purposes, among $K = 3$ agencies representing, for example, regional governmental authorities. An alternative, and more complicated partition of chemical databases occurs in Karr, *et al.* [25].

The database sizes are comparable: $n_1 = 172$, $n_2 = 182$ and $n_3 = 152$. The response $y$ is median housing value, and three predictors were selected: $X_1$ = CRIME per capita, $X_2$ = IND[USTRIALIZATION], the proportion of non-retail business acres, and $X_3$ = DIST[ANCE], a weighted sum of distances to five Boston employment centers.

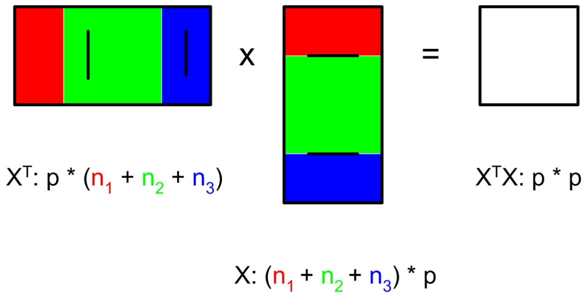

Figure 3: Pictorial representation of the secure regression protocol for horizontally partitioned data. The dimensions of various matrices are shown. The visual metaphor is that $X^T X$ (white) is the sum of red, green and blue components.

Figure 4 shows the results of the computations of their respective $(X^j)^T X^j$ and $(X^j)^T y^j$ performed by the three agencies. The agencies then use the secure regression protocol to produce the global values

$$\begin{aligned} X^T X &= (X^1)^T X^1 + (X^2)^T X^2 + (X^3)^T X^3 \\ &= \begin{bmatrix} 506.00 & 1828.44 & 5635.21 & 1920.29 \\ 1828.44 & 43970.34 & 32479.10 & 3466.28 \\ 5635.21 & 32479.10 & 86525.63 & 16220.67 \\ 1920.29 & 3466.28 & 16220.67 & 9526.77 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} X^T y &= (X^1)^T y^1 + (X^2)^T y^2 + (X^3)^T y^3 \\ &= \begin{bmatrix} 11401.60 \\ 25687.10 \\ 111564.08 \\ 45713.87 \end{bmatrix}. \end{aligned}$$

These global objects are shared among the three agencies, each of which can then calculate the estimated values of the regression coefficients.

Figure 5 contains these estimators, as well as the estimators for the three agency-specific local regressions. The intercept is $\hat{\beta}_{\text{CONST}}$, the coefficient corresponding to the constant predictor $X_1$. Each agency $j$ ends up knowing both—but only—the global coefficients and its own local coefficients. To the extent that these differ, it can infer some information about the other agencies' regressions collectively, but not individually. In this example, agency 2 can detect that its regression differs from the global one, but is not able to determine that agency 1 is the primary cause for the difference. Agency 3 is unaware that the regressions of both agency 1 and agency 2 differ from the global regression.

## 3.3 Model Diagnostics

In the absence of model diagnostics, secure regression loses appeal to statisticians. We describe briefly two strategies for producing informative diagnostics. The first is to use quantities that can be computed using

| Agency $j$ | $n_j$ | $(X^j)^T X^j$ | $(X^j)^T y^j$ |
|---|---|---|---|
| 1 | 172 | $\begin{bmatrix} 172.00 & 49.03 & 1581.19 & 781.52 \\ 49.03 & 40.42 & 556.29 & 180.95 \\ 1581.19 & 556.29 & 23448.60 & 5631.35 \\ 781.52 & 180.95 & 5631.35 & 4186.07 \end{bmatrix}$ | $\begin{bmatrix} 4057.90 \\ 909.24 \\ 32227.19 \\ 18996.12 \end{bmatrix}$ |
| 2 | 182 | $\begin{bmatrix} 182.00 & 94.47 & 1563.50 & 746.12 \\ 94.47 & 160.90 & 1433.20 & 231.87 \\ 1563.50 & 1433.20 & 18970.98 & 5224.19 \\ 746.12 & 231.87 & 5224.19 & 3882.02 \end{bmatrix}$ | $\begin{bmatrix} 4691.10 \\ 2299.13 \\ 37949.83 \\ 19193.18 \end{bmatrix}$ |
| 3 | 152 | $\begin{bmatrix} 152.00 & 1684.95 & 2490.52 & 392.64 \\ 1684.95 & 43769.02 & 30489.61 & 3053.46 \\ 2490.52 & 30489.61 & 44106.05 & 5365.14 \\ 392.64 & 3053.46 & 5365.14 & 1458.68 \end{bmatrix}$ | $\begin{bmatrix} 2652.60 \\ 22478.73 \\ 41387.06 \\ 7524.57 \end{bmatrix}$ |

Figure 4: Illustration of the secure regression protocol for horizontally partitioned data using the "Boston housing data" [21]. As discussed in the text, there are three agencies, each of which computes its local $(X^j)^T X^j$ and $(X^j)^T y^j$. These are combined entrywise using secure summation to produce shared global values $X^T X$ and $X^T y$, from which each agency calculates the global regression coefficients.

secure summation from corresponding local statistics. The second uses secure data integration protocols from §5 to share synthetic residuals.

A number of diagnostics are computable by secure summation. These include:

1. The coefficient of determination $R^2$;
2. The least squares estimate $S^2$ of the error variance $\sigma^2$, which was noted in (6);
3. Correlations between predictors and residuals;
4. The hat matrix $H = X(X^T X)^{-1} X^T$, which can be used to identify $X$-outliers.

For diagnosing some types of assumption violations, only patterns in relationships among the residuals and predictors suggestive of model mis-specification are needed, rather than exact values of the residuals

| Regression | $\hat{\beta}_{\text{CONST}}$ | $\hat{\beta}_{\text{CRIME}}$ | $\hat{\beta}_{\text{IND}}$ | $\hat{\beta}_{\text{DIST}}$ |
|---|---|---|---|---|
| Global | 35.505 | -0.273 | -0.730 | -1.016 |
| Agency 1 | 39.362 | -8.792 | -0.720 | -1.462 |
| Agency 2 | 35.611 | 2.587 | -0.896 | -0.849 |
| Agency 3 | 34.028 | -0.241 | -0.708 | -0.893 |

Figure 5: Estimated global and agency-specific regression coefficients for the partitioned Boston housing data. The intercept is $\hat{\beta}_{\text{CONST}}$.

and predictors. Such diagnostics can be produced for the global database using secure data integration protocols (§5) to share synthetic diagnostics. The synthetic diagnostics are generated in three steps [37]. First, each agency simulates values of its predictors. Second, using the global regression coefficients, each agency simulates residuals associated with these synthetic predictors in a way—and this is the hard part—that mimics the relationships between the predictors and residuals in its own data. Finally, the agencies share their synthetic predictors and residuals using secure data integration.

# 4 Vertically Partitioned Data

*Vertically partitioned* databases contain different sets of attributes for the same data subjects. For example, one government agency might have employment information, another health data, and a third information about education, but all for the same individuals.

In this section, we show to perform regression analyses on vertically partitioned data. One approach (§4.1) assumes that the database owners are willing to share sample means and covariances, which allows them to perform much richer sets of analyses than mere coefficient estimation, including inference for the coefficients, model diagnostics and model selection. The second approach (§4.2) solves directly the quadratic optimization problem associated with computation of least squares estimators. It entails less sharing of information, but requires that all agencies have access to the response attribute.

Two assumptions underlie this section. First, we assume that the agencies know that they have data on the same subjects, or that there is a secure method for determining which subjects are common to all their databases. The second, and stronger, assumption is that agencies can link records without error. Operationally, this requires in effect that the databases have a common primary key, such as social security number. How realistic this assumption varies by context. For administrative and financial records, it may be sensible, but it becomes problematic in situations where error-prone keys such as name or address must be used.

For the remainder of the section, we assume that the agencies have aligned their common data subjects in the same order.

## 4.1 Secure Matrix Products

This method [29], which is in the spirit of Du, Han and Chan [10], computes the off-diagonal blocks of the full data covariance matrix securely.

Since each such block involves only two agencies, we restrict attention to two database owners labeled agency $A$ and agency $B$, that possess disjoint sets of attributes for the same $n$ data subjects. Let agency $A$ possess $n$ $p$-dimensional data elements $X_1^A, \ldots, X_n^A$, and let agency $B$ possess $n$ $q$-dimensional data elements $X_1^B, \ldots, X_n^B$, so that the full data matrix is

$$[X^A \quad X^B] = \begin{bmatrix} X_{11}^A & \cdots & X_{1p}^A & X_{11}^B & \cdots & X_{1q}^B \\ \vdots & & \vdots & \vdots & & \vdots \\ X_{n1}^A & \cdots & X_{np}^A & X_{n1}^B & \cdots & X_{nq}^B \end{bmatrix}. \tag{7}$$

We assume the two data matrices are of full rank; if not, the agencies remove linearly dependent columns.

The agencies wish to compute securely and share the $p \times q$-dimensional matrix $(X^A)^T X^B$. Assuming that they also share "diagonal blocks" of the covariance matrix; as we describe below, once they have done so, each possesses the "full data" covariance matrix, and may perform a variety of statistical analyses of the integrated data.

### 4.1.1 Computation of Secure Matrix Products

An optimal computational protocol ensures that neither agency learns more about the other's data by using the protocol than it would learn if an omniscient third party were to tell it the result. From the perspective of fairness, the protocol should be symmetric in the amount of information exchanged. A protocol that achieves both of these goals, at least approximately, is:

1. Agency $A$ generates a set of $g = \lfloor (n - p)/2 \rfloor$ orthonormal vectors $Z_1, Z_2, \ldots, Z_g \in \mathbb{R}^n$ such that $Z_i^T X_j^A = 0$ for all $i$ and $j$, and sends the matrix $Z = [Z_1 Z_2 \cdots Z_g]$ to agency $B$.

2. Agency $B$ computes
$$W = (I - ZZ^T)X^B,$$
where $I$ is an identity matrix, and sends $W$ to agency $A$.

3. Agency $A$ calculates, and shares with agency B,
$$(X^A)^T W = (X^A)^T (I - ZZ^T) X^B = (X^A)^T X^B.$$
The latter equality holds since $(X_j^A)^T Z_i = 0$ for all $i$ and $j$.

A method for generating $Z$ is presented in Sanil, *et al.* [29].

It might appear that agency $B$'s data can be learned exactly since agency $A$ knows both $W$ and $Z$. However, $W$ has rank $(n - g) = (n - 2p)/2$, so that agency $A$ cannot invert it to obtain $X^B$.

To assess the degree of protection afforded by this protocol, we note that for any matrix product protocol where $(X^A)^T X^B$ is learned by both agencies, including protocols that involve trusted third parties, at a minimum each agency knows $pq$ constraints on the other's data, one for each element of $(X^A)^T X^B$. In realistic settings, the number of data subjects is much greater than the number of terms in the cross-product matrix: $n \gg pq$. Thus, the knowledge of agency $A$ about $X^B$ consists of $pq$ constraints implied by $(X^A)^T X^B$, and that the $X_i^B$ lie in the $g \approx n/2$-dimensional subspace given by $W = (I - ZZ^T)X^B$. Thus, agency $A$ has a total of $g + pq$ constraints on $X^B$. Assuming $n \gg pq$, we can say that agency $A$ knows the approximately $n/2$-dimensional subspace that the $X_i^B$ lie in. For large $n$, agency $B$'s data may be considered safe.

Correspondingly, agency $B$ knows $pq$ constraints on $X^A$ implied by $(X^A)^T X^B$, and that the $X_i$ lie in the $(n - g) \approx n/2$-dimensional subspace orthogonal to $Z$. Thus, agency $B$ has a total of $n - g + pq$ constraints on $X^A$. Assuming $n \gg pq$ and that $g \approx n/2$, we can say that agency $B$ knows the approximately $n/2$-dimensional subspace that the $X_i^A$ lie in. For large $n$, agency $A$'s data may be considered safe.

Since agency $A$ and agency $B$ can each place the other's data in an approximately $n/2$-dimensional subspace, the protocol is symmetric in the information exchanged. At higher levels, though, symmetry can break down. For example, if agency $A$ holds the response, but none of its other attributes is a good predictor, whereas the attributes held by agency $B$ are good predictors, then arguably $A$ learns more about $B$'s data than *vice versa*.

The protocol is not optimal in the sense of each agency's learning as little as possible about the other's data. From $(X^A)^T X^B$ alone, agency $A$ has only $pq$ constraints on $X^B$, rather than the approximately $n/2$ constraints described above. The symmetry, however, implies a minimax form of optimality: the total amount of information that must be exchanged is $n$ (Consider the extreme case that agency $A$ transmits its data to agency $B$, which computes $(X^A)^T X^B$ and returns the result to $A$.), and so each agency's transmitting $n/2$ constraints on its data minimizes the maximum information transferred.

Nor is the protocol immune to breaches of confidentiality if the agencies do not use their real data. Moreover, disclosures might be generated because of the values of the attributes themselves. A related problem occurs if one agency has attributes that are nearly linear combinations of the other agency's attributes. When this happens, accurate predictions of the data subjects' values can be obtained from linear regressions built from the securely computed matrix products.

### 4.1.2 Application to Secure Regression

Application of the secure matrix product protocol to perform secure linear regression analyses is straightforward. Altering notation for simplicity, let the matrix of all variables in the possession of the agencies be $D = [D_1, \ldots, D_p]$, with

$$D_i = \begin{bmatrix} d_{i1} \\ \vdots \\ d_{in} \end{bmatrix}, \quad 1 \le i \le p \ . \tag{8}$$

The data matrix $D$ is distributed among agencies $A_1, A_2, \ldots, A_K$. Each agency $A_j$ possesses its own $p_j$ columns of $D$, where $\sum_{j=1}^{K} p_j = p$.

A regression model of some response attribute, say $D_i \in D$, on a collection of the other attributes, say $D^0 \subseteq D \setminus \{D_i\}$, is of the form

$$D_i = D^0\beta + \varepsilon \tag{9}$$

where $\varepsilon \sim N(0, \sigma^2)$. As in §3, an intercept term is achieved by including a column of ones in $D^0$, which without loss of generality, we assume is owned by agency $A_1$.

The goal is to regress any $D_i$ on some arbitrary subset $D^0$ using secure computation. For simplicity, we suppress dependence of $\beta$, $\varepsilon$ and $\sigma^2$ on $D^0$. The maximum likelihood estimates of $\beta$ and $\sigma^2$, as well as the standard errors of the estimated coefficients, can be obtained from the sample covariance matrix of $D$, using for example the sweep algorithm [1, 40]. Hence, the agencies need only the elements of the sample covariance matrix of $D$ to perform the regression. Each agency computes and shares the on-diagonal blocks of the matrix corresponding to its variables, and the agencies use secure matrix computations as described above to compute the off-diagonal blocks.

The types of diagnostic measures available in vertically partitioned data settings depend on the amount of information the agencies are willing to share. Diagnostics based on residuals require the predicted values, $D^0\hat{\beta}$. These can be obtained using the secure matrix product protocol, since

$$D^0\hat{\beta} = D^0\left[(D^0)^T D^0\right]^{-1}(D^0)^T D_i.$$

Alternatively, once the $\hat{\beta}$ is shared, each agency could compute the portion of $D^0\hat{\beta}$ based on the attributes in its possession, and these vectors can be summed across agencies using secure summation.

Once the predicted values are known, the agency with the response $D_i$ can calculate the residuals $E_0 = D_i - D^0\hat{\beta}$. If that agency is willing to share the residuals with the other agencies, each agency can plot residuals versus its predictors and report the nature of any lack of fit to the other agencies. Sharing $E_0$ also enables all agencies to obtain Cook's distance measures, since these are functions of $E_0$ and the diagonal elements of $H = D^0[(D^0)^T D^0]^{-1}(D^0)^T$, which can be computed securely, as noted in §3.

The agency with $D_i$ may be unwilling to share $E_0$ with the other agencies, since sharing could reveal the values of $D_i$ itself. In this case, one option is to compute the correlations of the residuals with the independent variables using the secure matrix product protocol. When the model fits poorly, these correlations will be far from zero, suggesting model mis-specification. Additionally, the agency with $D_i$ can make a plot of $E_0$ versus $D^0\hat{\beta}$, and a normal quantile plot of $E_0$, and report any evidence of model violations to the other agencies. The number of residuals exceeding certain thresholds, i.e., outliers, also can be reported.

Variations of linear regression can be performed using the secure matrix product protocol. For example, to perform weighted least squares regression, the agencies first securely pre-multiply their variables by $T^{1/2}$, where $T$ is the matrix of weights, and then apply the secure matrix protocol to the transformed variables. To run semi-automatic model selection procedures such as stepwise regression, the agencies can calculate the covariance matrix securely, then select models based on criteria that are functions of it, such as the $F$-statistic or the Akaike Information Criterion.

It is also possible to perform ridge regression [23] securely. Ridge regression shrinks the estimated regression coefficients away from the maximum likelihood estimates by imposing a penalty on their magnitude. Written in matrix form, ridge regression seeks the $\hat{\beta}$ that minimizes

$$\text{Ridge}(\lambda) = (D_i - D^0\beta)^T(D_i - D^0\beta) + \lambda\beta^T\beta, \tag{10}$$

where $\lambda$ is a specified constant. The ridge regression estimate of the coefficients is

$$\hat{\beta}_R = \left[(D^0)^T D^0 + \lambda I\right]^{-1}(D^0)^T D_i. \tag{11}$$

Since $(D^0)^T D^0$ can be computed using the secure matrix product protocol, $[(D^0)^T D^0 + \lambda I]^{-1}$ can be obtained and shared among the agencies. The agencies also can share $(D^0)^T D_i$ securely, which enables calculation of the estimated ridge regression coefficients.

## 4.2 Secure Least Squares

A second approach to vertically partitioned data entails less sharing of information than for the secure matrix product protocol of §4.1, but requires that all agencies possess the response attribute $y$. If this were not the case, the agency holding $y$ would be required to share it with the others, which poses obvious disclosure risks.

We assume the model of (1), and that (3) holds. The least squares estimates $\hat{\beta}$ of (4) are, by definition, the solution of the quadratic optimization problem

$$\hat{\beta} = \arg\min_{\beta}\,(y - X\beta)^T(y - X\beta). \tag{12}$$

Denote by $I_j$ the predictors held by agency $A_j$, and assume that the $I_j$ are disjoint. If there were overlaps, the agencies would decide in advance which one "owns" shared attribute. For a vector $u$, we write $u_{I_j}$ for $\{u_i\}_{i\in I_j}$. The total number of attributes—predictors and response—remains $p$.

As in other protocols for secure multi-party computation, one agency must assume a lead role in initiating and coordinating the process. This is a purely administrative role and does not imply any information advantage or disadvantage. We assume that agency 1 is the designated leader.

### 4.2.1 Powell's Algorithm

The basis of the computational protocol is *Powell's method* [34] for solution of quadratic optimization problems with calculating—which in practice means approximating numerically—derivatives. We will use it to calculate $\hat{\beta}$ in (12) directly.

Powell's method is a derivative-free numerical minimization method that solves the multidimensional minimization problem by solving a series of one-dimensional ("line search") minimization problems. A high-level description of the algorithm is as follows:

1. Start with a set of suitably chosen set of $p$ vectors in $\mathbb{R}^p$ that serve as "search directions."
2. Start at an arbitrary starting point in $\mathbb{R}^p$ and determine the step size $\delta$ along the first search direction $s^{(1)}$ that minimizes the objective function.
3. Move distance $\delta$ along $s^{(1)}$.
4. Move an optimal step in the second search direction $s^{(2)}$, and so on until all the search directions are exhausted.
5. Make appropriate updates to the set of search directions, and continue until the minimum is obtained.

Specifically, the procedure for finding the minimizer of the function $f(\beta)$ consists of an initialization step and an iteration block as described below.

**Initialization:** Select an arbitrary[2] orthogonal basis $s^{(1)}, \ldots, s^{(p)}$ for $\mathbb{R}^p$. Also, pick an arbitrary starting point $\tilde{\beta} \in \mathbb{R}^p$.

**Iteration:** Repeat the following block of steps $p$ times.

- Set $\beta \leftarrow \tilde{\beta}$.
- For $i = 1, \ldots, p$, find $\delta$ that minimizes $f(\beta + \delta s^{(i)})$, and then set $\beta \leftarrow \beta + \delta s^{(i)}$.
- For $i = 1, \ldots, (p-1)$, set $s^{(i)} \leftarrow s^{(i+1)}$.
- Set $s^{(p)} \leftarrow \beta - \tilde{\beta}$.
- Find $\delta$ that minimizes $f(\beta + \delta s^{(p)})$, and set $\tilde{\beta} \leftarrow \beta + \delta s^{(p)}$.

Note that each iteration of the iteration block involves solving $(p+1)$ one-dimensional minimization problems, to determine the $\delta$'s.

Powell [34] established the remarkable result that if $f$ is a quadratic function, then $p$ iterations of the iteration block yield the *exact minimizer* of $f$! That is, solving $p(p+1)$ one-dimensional minimization produces the minimizer of a quadratic function.

### 4.2.2 Application to Secure Regression

The gist of our approach [39] is to apply Powell's method to

$$f(\beta) = (y - X\beta)^T (y - X\beta)$$

in order to solve (12). The complication, of course, is that no agency possesses all of the data. The details are as follows.

1. Let $s^{(1)}, \ldots, s^{(p)} \in \mathbb{R}^p$ be $p$-dimensional vectors that will serve as a set of search directions in $\mathbb{R}^p$, to be used for finding the optimal estimate $\hat{\beta}$. The $s^{(r)}$ will be initially chosen and later updated in such a manner that agency $A_j$ knows only the $s^{(r)}_{I_j}$ components of each $s^{(r)}$.

[2] Powell's original algorithm used the coordinate axis vectors as the basis, but any orthogonal basis also suffices [4].

2. Initially, $s^{(r)}$ are chosen as follows. Each $A_j$ picks an orthogonal basis $\{v^{(r)}\}_{r \in I_j}$ for $\mathbb{R}^{d_j}$. Then for $r \in I_j$ let $s_{I_j}^{(r)} = v^{(r)}$, and $s_l^{(r)} = 0$ for $l \notin I_j$. Each agency should pick its basis at random so that the other agencies cannot guess it.

3. Let $\tilde{\beta} = \left(\tilde{\beta}_{I_1}, \ldots, \tilde{\beta}_{I_k}\right) \in \mathbb{R}^p$ be the initial starting value of $\beta$ obtained by each $A_j$'s picking $\tilde{\beta}_{I_j}$ arbitrarily.

4. Perform the **Basic Iteration Block** below $p$ times. The final value of $\tilde{\beta}$ will be the least squares estimators $\hat{\beta}$.

The **Basic Iteration Block** is:

1. Each $A_j$ sets $\beta_{I_j} \leftarrow \tilde{\beta}_{I_j}$.

2. For $r = 1, \ldots, p$:

   (a) Each $A_j$ computes $X_{I_j}\beta_{I_j}$ and $X_{I_j}s_{I_j}^{(r)}$.

   (b) The agencies use secure summation to compute

   $$z = y - X\beta = y - \sum_{j=1}^{K} X_{I_j}\beta_{I_j}$$

   and

   $$w = Xs^{(r)} = \sum_{j=1}^{K} X_{I_j}s_{I_j}^{(r)}.$$

   In (only) the first iteration of this block, for a given $r$, $X_{I_j}s_{I_j}^{(r)}$ is non-zero only for the agency that owns $x_r$. Revealing this to all agencies would too risky, so only that particular agency, say $A_r$, will compute $w$, but not reveal it to the others.

   (c) All agencies compute

   $$\delta = z^T w / w^T w.$$

   In the first iteration, $A_r$ computes this and announces it to the other agencies.

   (d) Each $A_j$ updates $\beta_{I_j} \leftarrow \beta_{I_j} + \delta \cdot s_{I_j}^{(r)}$.

3. For $r = 1, \ldots, (p-1)$, each $A_j$ updates $s_{I_j}^{(r)} \leftarrow s_{I_j}^{(r+1)}$.

4. Each $A_j$ updates $s_{I_j}^{(p)} \leftarrow \beta_{I_j} - \tilde{\beta}_{I_j}$.

5. $z$, $w$ and $\delta$ are computed as before, and each $A_j$ updates $\beta_{I_j} \leftarrow \beta_{I_j} + \delta \cdot s_{I_j}^{(p)}$.

After the regression coefficients are calculated and shared, the agencies learn at least three useful quantities. The first of these, of course, is the global coefficients $\hat{\beta}$, enabling each agency to assess the effect of its variables on the response variable after accounting for the effects of the other agencies' variables. Agencies can also assess the size of effects of the other agencies' variables. If an agency obtains a complete record for some individual, the global regression equation can also be used for prediction of the response value. A comparison of the globally obtained coefficients with the coefficients of the local regression (i.e., the regression of $y$ on $X_{I_j}$) could also be informative.

Agencies also learn the vector of residuals $e = y - X\hat{\beta}$, which is equal to the final $z$ in our iterative procedure. The residuals permit agencies to perform diagnostic tests to determine if the linear regression model is appropriate. The agencies can perform formal statistical tests or use simple visual diagnostics [39]. Finally, agencies can compute the coefficient of determination

$$R^2 = \frac{y^T y - e^T e}{y^T y}. \tag{13}$$

To assess what is revealed by this protocol, consider any one step of the iteration: the only information exchanged by the agencies are the $z$ and $w$ vectors. The actual risk to the data $x$ is less since there is some masking with components of the $s$ vectors. Specifically, the vulnerability is highest in the first step of the iteration since—because of the way we have chosen the initial $s$—only one agency contributes to the sum $w$ at each round of the basic iteration block. We can reduce risk of disclosure by having the contributing agency compute $\delta$ privately and announce it to the others. If we assume that the agencies select their initial bases randomly, so that it is impossible for the others to guess them, and if the summation is performed using the secure summation protocol, then no private information is revealed if only $z$ and $w$ are common knowledge.

If iterations were independent, then clearly the procedure would be secure. However, the values that each agency contributes to the sum are functionally related from one iteration to the next. The relationship is complex and difficult to express, however, so that this complexity combined with the nature of the secure sum protocol will make it impossible in practice for malicious agencies to exploit the iteration-to-iteration dependency of the values to compromise data privacy.

Whether the approach is feasible computationally has not been established.

## 5 Secure Data Integration

The procedures described in §3 and 4 are tailored to regressions, or more generally to statistical analyses for which there exist sufficient statistics that are additive over the agencies. This makes the protocols efficient, but obviously every time a new kind of analysis is needed, so are new algorithms.

If the agencies are concerned primarily with protecting which one holds which data elements, then it is possible to construct an integrated database that can be shared among the agencies, and on which any kind of analysis is possible. There are, however, at least two problematic aspects of this. First, it requires sharing individual data values, with attendant disclosure risks to the data subjects. Second, secure data integration does not work in situations when data values themselves are informative about their source. For instance, it would not work with state-held databases containing ZIP codes. Nor would it work when, for example, for hospital databases containing income when the patient populations have drastically different incomes.

Consider $K > 2$ agencies wishing to share the integrated data among themselves without revealing the origin of any record, and without use of mechanisms such as a trusted third party. We present two algorithms for doing this, neither of which provides any confidentiality protection for data subjects beyond what may already have been imposed by the agencies.

### 5.1 Algorithm 1

Algorithm 1 passes a continually growing integrated database among the agencies in a known round-robin order, and in this sense is similar to secure summation, although multiple rounds are required. To protect the sources of individual records, agencies are allowed or required to insert both real and "synthetic" records. The synthetic data may be produced by procedures similar to those for construction of synthetic residuals (see §3.3), by drawing from predictive distributions fit to the data [30], or by some other means. Once all real data have been placed in the integrated database, each agency recognizes and removes its synthetic data, leaving the integrated database.

The steps in Algorithm 1 are:

1. **Initialization:** Order the agencies by number 1 through $K$.

2. **Round 1:** Agency 1 initiates the integrated database by adding *only synthetic data*. Every other agency puts in a mixture of at least 5% of its real data and—optionally—synthetic data, and then randomly permutes the current set of records. The value of 5% is arbitrary, and serves to ensure that the process terminates in at most 21 rounds. Permutation thwarts attempts to identify the source of records from their position in the database.

3. **Rounds** $2, \ldots, 20$**:** Each agency puts in at least 5% of its real data or all real data that it has left, and then randomly permutes the current set of records.

4. **Round 21:** the Agency 1, if it has data left, adds them, and removes its synthetic records. In turn, each other agency $2, \ldots, K$ removes its synthetic data.

5. **Sharing:** The integrated data are shared after all synthetic data have been removed.

The role of synthetic data is analogous to that of the random number $R$ in secure summation: without it, agency 2 would receive only real data from agency 1 in round 1. However, synthetic data do not protect the agencies completely. In round 1, agency 3 receives a combination of synthetic data from agency 1 and a mixture of synthetic and real data from agency 2. By retaining this intermediate version of the integrated database, which semi-honesty allows, and comparing it with the final version, which contains only real data, agency 2 can determine which records are synthetic—they are absent from the final version—and thus identify agency 2 as the source of some real records. The problem propagates, but with decreasing severity. For example, what agency 4 receives in round 1 is a mixture of synthetic data from agency 1, synthetic and real data from agency 2, and synthetic and real data from agency 3. By *ex post facto* removal of the synthetic data, agency 4 is left with real data that it knows to have come from either agency 2 or agency 3, although it does not know which. There are also corresponding vulnerabilities in the last round.

Algorithm 1 is rather clearly vulnerable to poorly synthesized data. For example, if the synthetic data produced by agencies 1 and 2 are readily detectable, then even without retaining intermediate versions of the database, agency 3 can identify the real data received from agency 2 in round 1. There is no guaranteed way to eliminate risks associated with retained intermediate computations in Algorithm 1, other than the agencies' agreeing not to retain intermediate versions of the integrated database. Alternatively, the agencies may simply accept the risks, since only a controllably small fraction of the data is compromised. Given the "at least 5% of real data" requirement in Algorithm 1, agency 2 would be revealing 5% of its data to agency 3, agencies 2 and 3 would reveal collectively 5% of their data to agency 4, and so on. Reducing 5% to a smaller value would reduce this risk, but at the expense of requiring more rounds.

## 5.2 Algorithm 2

Algorithm 2 is more secure than Algorithm 1, but it is also much more complex. In particular, while the algorithm will terminate in a finite number of stages, there is no fixed upper bound on this number. By randomizing the order in which agencies add data not only are the risks reduced but also the need for synthetic data is almost obviated. In addition to a growing integrated database, Algorithm 2 requires transmission of a binary vector $d = (d_1, \dots, d_K)$, in which $d_j = 1$ indicates that agency $j$ has not yet contributed all of its data and $d_j = 0$ indicates that it has.

Steps in Algorithm 2 are:

1. **Initialization:** A randomly chosen agency is designated as the *stage 1 agency* $a_1$.

2. **Stage 1:** The stage 1 agency $a_1$ initializes the integrated database with some synthetic data and at least one real data record, and permutes the order of the records. If $a_1$ has exhausted its data, it sets $d_{a_1} = 0$. Then, $a_1$ picks a *stage 2 agency* $a_2$ randomly from the set of agencies $j$, other than itself, for which $d_j = 1$, and sends the integrated database and the vector $d$ to $a_2$.

3. **Intermediate stages** $2, \dots$**:** As long as more than two agencies have data left, the stage $\ell$ agency $a_\ell$ adds at least one real data record and, optionally, as many synthetic data records as it wishes, to the integrated database, and then permutes the order of the records. If its own data are exhausted, it sets $d_{a_\ell} = 0$. It then selects the stage $\ell + 1$ agency $a_{\ell+1}$ randomly from the set of agencies $j$, other than itself, for which $d_j = 1$ and sends the integrated database and the vector $d$ to $a_{\ell+1}$.

4. **Final round:** Each agency removes its synthetic data.

The attractive feature of Algorithm 2 is that because of the randomization of the "next stage agency," no agency can be sure which other agencies other than possibly the agency from which it received the current integrated database has contributed real data to it. The number and order of previous contributors to the growing integrated database cannot be determined. Nor—it if comes from the stage 1 agency—is there even certainty that the database contains real data.

In fact, to a significant extent, Algorithm 2 does not even need synthetic data. The one possible exception is stage 1. If only real data were used, an agency that receives data from the stage 1 agency knows that with probability $1/(K-1)$ that it is the stage 2 agency, and would, even with this low probability, be able to associate them with the stage 1 agency, which is presumed to be known to all agencies. The variant of Algorithm 2 that uses synthetic data at stage 1 and only real data thereafter seems completely workable.

## 5.3 Application: Secure Contingency Tables

The algorithms for secure data integration have both direct uses—to do data integration—and indirect applications. Here we illustrate the latter, using secure data integration to construct contingency tables containing counts.

Let $\mathcal{D}$ be a database containing only categorical variables $V_1, \dots, V_J$. The associated contingency table is the $J$-dimensional array $T$ defined by

$$T(v_1, \dots, v_J) = \#\{r \in \mathcal{D} : r_1 = v_1, \dots, r_J = v_J\}, \tag{14}$$

where each $v_i$ is a possible value of the categorical variable $V_i$[3], $\#\{\cdot\}$ denotes "cardinality of $\cdot$" and $r_i$ is the $i$th attribute of record $i$. The $J$-tuple $(v_1, \dots, v_J)$ is called the cell coordinates. More generally, contingency

[3] For example, if $V_1$ is gender, then possible values of $v_1$ are "female" and "male."

tables may contain sums of numerical variables rather than counts; in fact the procedure described below works in either case. The table $T$ is a near-universal sufficient statistic, for example for fitting log-linear models [3].

While (14) defines a table as an array, this is not a feasible data structure for large tables—with many cells, which are invariably sparse—with relatively few cells having non-zero counts. For example, the table associated with the Census "long form," which contains 52 questions, has more than $10^{15}$ cells (1 gigabyte = $10^9$) but at most approximately $10^8$ (the number of households in the US) of these are non-zero. The *sparse representation* of a table is the data structure of (cell coordinate, cell count) pairs

$$\Big(v_1, \ldots, v_J, T(v_1, \ldots, v_J)\Big)$$

for only those cells for which $T(v_1, \ldots, v_J) \neq 0$. Algorithms that use the sparse representation data structure have been developed for virtually all important table operations.

Consider now the problem of securely building a contingency table from agency databases $\mathcal{D}_1, \ldots, \mathcal{D}_K$ containing the same categorical attributes for disjoint sets of data subjects. Given the tools described in §3, 5.1 and 5.2, this process is straightforward. The steps:

1. **List of Non-Zero Cells:** Use secure data integration (either protocol) to build the list $\mathcal{L}$ of cells with non-zero counts. The "databases" being integrated in this case are the agencies' individual lists of cells with non-zero counts. The protocols in §5.1 and 5.2 allow each agency not to reveal in which cells it has data in.

2. **Non-Zero Cell Counts:** For each cell in $\mathcal{L}$, use secure summation to determine the associated count (or sum).

# 6 Discussion

In this paper we have presented a framework for secure linear regression and other statistical analyses in a cooperative environment, under various forms of data partitioning.

A huge number of variations is possible. For example, in the case of horizontally partitioned data, in order to give the agencies flexibility, it may be important to allow them to withdraw from the computation when the perceived risk becomes too great. Ideally, this should be possible without first performing the regression. To illustrate, agency $j$ may wish to withdraw if its sample size $n_j$ is too large relative to the global sample size $n = \sum_{i=1}^{K} n_i$, which is the classical $p$-rule in the statistical disclosure limitation literature [44]. But, $n$ can be computed using secure summation, and so agencies may "opt out" according to whatever criteria they wish to employ, prior to any other computations. It is even possible, under a scenario that the process does not proceed if any one of the agencies opts out, to allow the opting out itself to be anonymous. Opting out in the case of vertically partitioned data does not make sense, however.

There are also more complex partitioning schemes. For example, initial approaches for databases that combine features of the horizontally and vertically partitioned cases are outlined in Reiter, *et al.* [38]. Both data subjects and attributes may be spread among agencies, and there may be many missing data elements, necessitating EM-algorithm-like methods. Additional issues arise, however, that require both new abstractions and new methods. For example, is there a way to protect the knowledge of which agencies hold which attributes on which data subjects? This information may be very important in the context of counterterrorism if it would compromise sources of information or reveal that data subjects are survey respondents.

Perhaps the most important issue is that the techniques discussed in this paper protect database holders, but not necessarily database subjects. Even when only data summaries are shared, there may be substantial disclosure risks. Consequently, privacy concerns about data mining in the name of counterterrorism might be attenuated, but would not be eliminated, by use of the techniques described here. Indeed, while it seems almost self-evident that disclosure risk is reduced by our techniques, this is not guaranteed, especially for vertically partitioned data. Nor is there any clear way to assess disclosure risk without actually performing the analyses, at which point it is arguably "too late." Research on techniques such as those in §3–5 from this "traditional" statistical disclosure limitation perspective is currently underway at NISS.

## Acknowledgements

This research was supported by NSF grant EIA–0131884 to the National Institute of Statistical Sciences (NISS). 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. We thank Max Buot, Christine Kohnen and Michael Larsen for useful discussion and comments.

## References

[1] BEATON, A. E., The use of special matrix operations in statistical calculus, Research Bulletin RB-64-51, Educational Testing Service, Princeton, NJ, 1964.

[2] BENALOH, J., Secret sharing homomorphisms: Keeping shares of a secret sharing, in Odlyzko, A. M., editor, *CRYPTO86*, Lecture Notes in Computer Science No. 263, pp. 251–260, Springer–Verlag, New York, 1987.

[3] BISHOP, Y. M. M., FIENBERG, S. E., AND HOLLAND, P. W., *Discrete Multivariate Analysis: Theory and Practice*, MIT Press, Cambridge, MA, 1975.

[4] BRENT, R. P., *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[5] DEFAYS, D. AND ANWAR, N., Micro-aggregation: A generic method, in *Proceedings of the 2nd International Symposium on Statistical Confidentiality*, pp. 69–78, Office for Official Publications of the European Community, Loxembourg, 1995.

[6] DEFAYS, D. AND NANOPOULOS, P., Panels of enterprises and confidentiality: The small aggregates method, in *Proceedings of the 92 Symposium on Design and Analysis of Longitudinal Surveys*, pp. 195–204, Statistics Canada, Ottawa, 1993.

[7] DOBRA, A., FIENBERG, S. E., KARR, A. F., AND SANIL, A. P., Software systems for tabular data releases, *Int. J. Uncertainty, Fuzziness and Knowledge Based Systems*, 10(5) (2002), pp. 529–544.

[8] DOBRA, A., KARR, A. F., AND SANIL, A. P., Preserving confidentiality of high-dimensional tabular data: Statistical and computational issues, *Statist. and Computing*, 13(4) (2003), pp. 363–370.

[9] DOYLE, P., LANE, J., THEEUWES, J. J. M., AND ZAYATZ, L. V., *Confidentiality, Disclosure and Data Access: Theory and Practical Application for Statistical Agencies*, Elsevier, Amsterdam, 2001.

[10] Du, W., Han, Y., and Chen, S., Privacy-preserving multivariate statistical analysis: Linear regression and classification, in *Proc. 4th SIAM Internat. Conf. on Data Mining*, pp. 222–233, 2004.

[11] Du, W. and Zhan, Z., A practical approach to solve secure multi-party computation problems, in *New Security Paradigms Workshop*, pp. 127–135, ACM Press, New York, 2002.

[12] Duncan, G. T., Jabine, T. B., and de Wolf, V. A., editors, *Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics*, Report of a Panel on Confidentiality and Data Access, Committee on National Statistics, National Academy Press, Washington, 1993.

[13] Duncan, G. T. and Keller-McNulty, S. A., Mask or impute? Proceedings of ISBA 2000, 2001.

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

[15] Duncan, G. T. and Stokes, L., Disclosure risk vs. data utility: The R–U confidentiality map as applied to topcoding, *Chance*, 17(3) (2004), pp. 16–20.

[16] Federal Committee on Statistical Methodology, *Report on Statistical Disclosure Limitation Methodology*, US Office of Management and Budget, Washington, 1994.

[17] Goldreich, O., Micali, S., and Wigderson, A., How to play any mental game, in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pp. 218–229, 1987.

[18] Goldwasser, S., Multi-party computations: Past and present, in *Proc. 16th Annual ACM Symposium on Principles of Distributed Computing*, pp. 1–6, ACM Press, New York, 1997.

[19] Gomatam, S., Karr, A. F., Reiter, J. P., and Sanil, A. P., Data dissemination and disclosure limitation in a world without microdata: A risk-utility framework for remote access analysis servers, *Statist. Sci.* (2005, to appear). Available on-line at www.niss.org/dgii/technicalreports.html.

[20] Gomatam, S., Karr, A. F., and Sanil, A. P., Data swapping as a decision problem, *J. Official Statist.* (2005, to appear). Available on-line at www.niss.org/dgii/technicalreports.html.

[21] Harrison, D. and Rubinfeld, D. L., Hedonic prices and the demand for clean air, *J. Environ. Econ. Mgt.*, 5 (1978), pp. 81–102.

[22] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer–Verlag, New York, 2001.

[23] Hoerl, A. E. and Kennard, R., Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics*, 12 (1970), pp. 55–67.

[24] Journal of Official Statistics, Special issue on disclosure limitation methods for protecting the confidentiality of statistical data. Volume 14, Number 4, edited by S. E. Fienberg and L. C. R. J. Willenborg, 1998.

[25] Karr, A. F., Feng, J., Lin, X., Reiter, J. P., Sanil, A. P., and Young, S. S., Secure analysis of distributed chemical databases without data integration, *J. Computer-Aided Molecular Design* (2005, to appear). Available on-line at www.niss.org/dgii/technicalreports.html.

[26] KARR, A. F., KOHNEN, C. N., OGANIAN, A., REITER, J. P., AND SANIL, A. P., A framework for evaluating the utility of data altered to protect confidentiality, *The American Statistician* (2005, submitted for publication). Available on-line at www.niss.org/dgii/technicalreports.html.

[27] KARR, A. F., LEE, J., SANIL, A. P., HERNANDEZ, J., KARIMI, S., AND LITWIN, K., Disseminating information but protecting confidentiality, *IEEE Computer*, 34(2) (2001), pp. 36–37.

[28] KARR, A. F., LIN, X., REITER, J. P., AND SANIL, A. P., Analysis of integrated data without data integration, *Chance*, 17(3) (2004), pp. 26–29.

[29] KARR, A. F., LIN, X., REITER, J. P., AND SANIL, A. P., Privacy preserving analysis of vertically partitioned data using secure matrix products, *J. Official Statist.* (2004, submitted for publication). Available on-line at www.niss.org/dgii/technicalreports.html.

[30] KARR, A. F., LIN, X., REITER, J. P., AND SANIL, A. P., Secure regression on distributed databases, *J. Computational and Graphical Statist.* (2005, to appear). Available on-line at www.niss.org/dgii/technicalreports.html.

[31] KARR, A. F., SANIL, A. P., AND BANKS, D. L., Data quality: A statistical perspective, *Statistical Methodology* (2005, to appear). Available on-line at www.niss.org/dgii/technicalreports.html.

[32] KARR, A. F., SANIL, A. P., SACKS, J., AND ELMAGARMID, E., Workshop Report: Affiliates Workshop on Data Quality, technical report, National Institute of Statistical Sciences, 2001. Available on-line at www.niss.org/affiliates/dqworkshop/report/dq-report.pdf.

[33] LEE, J., HOLLOMAN, C., KARR, A. F., AND SANIL, A. P., Analysis of aggregated data in survey sampling with application to fertilizer/pesticide usage surveys, *Res. Official Statist.*, 4 (2001), pp. 101–116.

[34] POWELL, M. J. D., An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Computer J.*, 7 (1964), pp. 152–162.

[35] RAGHUNATHAN, T. E., REITER, J. P., AND RUBIN, D. B., Multiple imputation for statistical disclosure limitation, *J. Official Statist.*, 19 (2003), pp. 1–16.

[36] REITER, J. P., Inference for partially synthetic, public use microdata sets, *Survey Methodology*, 29 (2003), pp. 181–188.

[37] REITER, J. P., Model diagnostics for remote access regression servers, *Statist. and Computing*, 13 (2003), pp. 371–380.

[38] REITER, J. P., KARR, A. F., KOHNEN, C. N., LIN, X., AND SANIL, A. P., Secure regression for vertically partitioned, partially overlapping data, *ASA Proceedings* (2004, to appear).

[39] SANIL, A. P., KARR, A. F., LIN, X., AND REITER, J. P., Privacy preserving regression modelling via distributed computation, in *Proc. Tenth ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining*, pp. 677–682, ACM Press, New York, 2004.

[40] SCHAFER, J. L., *Analysis of Incomplete Multivariate Data*, Chapman & Hall, London, 2003.

[41] SWEENEY, L., Computational disclosure control for medical microdata: The Datafly system, in *Record Linkage Techniques 1997: Proceedings of an International Workshop and Exposition*, pp. 442–453, 1997.

[42] TROTTINI, M., *Decision Models for Data Disclosure Limitation*, Ph.D. thesis, Carnegie Mellon University, 2003. Available on-line at www.niss.org/dgii/TR/Thesis-Trottini-final.pdf.

[43] WILLENBORG, L. C. R. J. AND DE WAAL, T., *Statistical Disclosure Control in Practice*, Springer–Verlag, New York, 1996.

[44] WILLENBORG, L. C. R. J. AND DE WAAL, T., *Elements of Statistical Disclosure Control*, Springer–Verlag, New York, 2001.

[45] YAO, A. C., Protocols for secure computations, in *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 160–164, ACM Press, New York, 1982.