Appendix: Details of the SPARK Protocol

1 Logistic Regression

Let $\boldsymbol{Y} = (Y_1, \ldots, Y_N)'$ be independent Bernoulli variables with mean $\mathrm{E}(\boldsymbol{Y}) = \boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)'$. Given an intercept and a set of covariates $\boldsymbol{X} = [\boldsymbol{1}, \boldsymbol{X}_{\bullet 1}, \ldots, \boldsymbol{X}_{\bullet v}]$, where $\boldsymbol{X}_{\bullet j} = (x_{1j}, \ldots, x_{Nj})'$ contains the values for covariate j, we define a logistic model with parameters $\boldsymbol{\beta}$ using the formula:

$$\mathrm{logit}(\boldsymbol{\mu}) = \log\left(\frac{\boldsymbol{\mu}}{1-\boldsymbol{\mu}}\right) = \boldsymbol{X\beta}$$

(we say that the logit function links the random component $\boldsymbol{\mu}$ to the systematic component $\boldsymbol{X\beta}$). The log-likelihood $l(\boldsymbol{\beta}; \boldsymbol{y})$ of the full model, which can be used to assess model fit (usually given as -2 log-likelihood), equals:

$$l(\boldsymbol{\beta}; \boldsymbol{y}) = \sum_{i=1}^{N} \left[y_i \boldsymbol{X}_{i\bullet}\boldsymbol{\beta} - \ln\left(1 + \exp\left(\boldsymbol{X}_{i\bullet}\boldsymbol{\beta}\right)\right)\right],$$

where $\boldsymbol{X}_{i\bullet}$ is row i from the design matrix $\boldsymbol{X}$.

For a set of observations $\boldsymbol{y} = (y_1, \ldots, y_N)$, we can determine parameter estimates $\boldsymbol{b}$ at which the log-likelihood $l(\boldsymbol{\beta}; \boldsymbol{y})$ of the model is maximized using the Newton-Raphson method (or, equivalently, the Fisher scoring method, since the estimated and observed information matrices are the same for a logistic model [1]). That is, we iteratively compute the estimates using $\boldsymbol{b}^{(t+1)} = \boldsymbol{b}^{(t)} - \left[\boldsymbol{\mathcal{I}}^{(t)}\right]^{-1} \boldsymbol{u}^{(t)}$, at iteration t, where $\boldsymbol{u}^{(t)} = \boldsymbol{X}'\left(\boldsymbol{y} - \boldsymbol{p}^{(t)}\right)$ is the estimated score vector with probability of success $\boldsymbol{p}^{(t)} = \mathrm{logit}^{-1}\left(\boldsymbol{X}\,\boldsymbol{b}^{(t)}\right)$, and $\boldsymbol{\mathcal{I}}^{(t)} = \boldsymbol{X}'\boldsymbol{W}^{(t)}\boldsymbol{X}$ is the estimated information matrix with weight matrix $\boldsymbol{W}^{(t)} = \mathrm{diag}\left[p_i^{(t)}\left(1 - p_i^{(t)}\right)\right]$. This fitting method can be used for any generalized linear model (with new derivations for the score vector and information matrix) [2], and has been shown to converge to a solution in fewer iterations than other optimization algorithms applied to logistic models [3].

2 Current Protocols and Systems

Assume there are k sites that are the data sources for the pooled analysis. They want to securely collaborate with each other to construct a logistic regression model on the pooled data set, but they are unable to share personal health information with each other. This kind of data is referred to as “horizontally partitioned data” because each party has a subset of records from the whole data set containing the information on all the attributes.

Let each site P_i have its own data set $\boldsymbol{D}_i$ such that $\boldsymbol{D} = \bigcup_{i=1}^{k} \boldsymbol{D}_i$ is the pooled data set. Let there be v independent attributes in each record of the dataset $\boldsymbol{D}$. We assume that the objective of the sites is to construct a logistic regression model on $\boldsymbol{D}$ (or other type of generalized linear model) without exchanging raw data. We assume that the variables represent sensitive personal health information, and should an adversary extract the raw values then this would be considered an inappropriate disclosure by the computation protocol.

A number of different distributed data network approaches have been proposed and used in the literature [4-7]. Below we review the architectures of these distributed networks and explain how they address the privacy problem.

2.1 Multi-site Regression

A number of proposals have been made to perform distributed analysis through collaborative computation among the sites. In 2004 Du et al. suggested that sites could "exchange aggregate information, not the raw data", in order to perform multiparty linear regression with horizontally partitioned data [8]. This is essentially the same approach suggested in DataSHIELD, although DataSHIELD has extended it to generalized linear models [9], and more recently in GLORE, which only considered logistic regression but added the Hosmer-Lemeshow goodness-of-fit test and area under the receiver operating characteristic (ROC) curve (AUC) [10]..

The parameter estimates $\boldsymbol{b}^{(t)}$ are common to all sites, but the score vector $\boldsymbol{u}^{(t)}$ and information matrix $\mathcal{I}^{(t)}$ are computed individually and combined later (by sharing with each other or a third party) to update the overall parameter estimates $\boldsymbol{b}^{(t+1)}$. That is, let $\mathcal{I}_i^{(t)}$ be the individual information matrix for site i, and $\boldsymbol{u}_i^{(t)}$ be the individual score vector for site i, where $i \in \{1,\ldots,k\}$. Then the overall information matrix is $\mathcal{I}^{(t)} = \sum_i \mathcal{I}_i^{(t)}$ and the overall score vector is $\boldsymbol{u}^{(t)} = \sum_i \boldsymbol{u}_i^{(t)}$, each representing a sum over all sites.

This is the approach suggested by Du et al. in 2004 [8], and implemented recently in DataSHIELD [9], and GLORE [10]. Under this approach, however, it may be possible for an adversary to extract information and values about the raw data, which would be considered an inappropriate disclosure of personal health information.

There are two possible architectures for the distributed analysis. In the first architecture a third-party would receive the component information matrix and score vector from each site. Let's call this the analysis center (AC). The AC would then perform the matrix inversion, and compute the new model parameters for the next iteration, and send the new parameters back to each of the sites. The sites would then compute their information matrix and score vector anew and the cycle repeats. However, the AC is not trusted to access or view the raw data (if the AC was trusted, then we could just send the raw data to the AC and let it create a pooled data set and construct the overall model). When the iterations are completed the AC would have the final model parameters.

In the second architecture there is no central AC, and therefore the sites collaborate in a round-robin fashion. The initiating site, say site 1, sends the initial parameters, its information matrix and score vector to the neighbouring site, say site 2. Site 2 then sums that with its matrix/vector and forward these to site 3, and so on until the final matrix and vector comes back to site 1, which inverts the matrix and computes the new parameters. Then the cycle starts again. When the iterations are completed all of the sites will have the final model parameters.

Below we show that under certain conditions neither of these two architectures prevent the disclosure of PHI if no cryptographic methods are used. Therefore, these protocols are not secure [8-10]. In the first architecture with the AC, the information it receives from the sites can lead to inappropriate disclosures. In the second architecture the information site 2 receives from site 1 can lead to inappropriate disclosures. These are described below.

For ease of presentation, and to simplify notation, we withhold the use of subscripts that index sites in the following discussions of potential inadvertent disclosures. All disclosures presented below are at the site level.

2.2 Inappropriate Disclosures from the Linear Model

A linear model is the simplest form of a generalized linear model, with normal errors and identity link function. It has score vector:

$$\boldsymbol{u} = \boldsymbol{X}'\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

and information matrix :

$$\mathcal{I} = \frac{1}{s^2} X'X \quad \text{..... (2)}$$

Where X is the design matrix, y is the outcome vector, and b is the vector of parameter estimates. Rearranging the components of the score vector, and including the information matrix, yields the equation:

$$X'y = u - X'Xb = u - s^2 \mathcal{I} b \quad \text{..... (3)}$$

If an intercept is included in the linear model (a point which would be known to all sites), then the estimated variance s^2 can be easily recovered. That is, with an intercept in the model, the first column of the design matrix will be 1's, hence the first entry of the information matrix will be $[\mathcal{I}]_{1,1} = N/s^2$, where N is the total number of observations in D.

Sharing the parameter estimates, score vector, and information matrix in a multi-site linear regression therefore leads to the disclosure of $X'y$ and $X'X$ (wherein information regarding higher-order interactions, which may be sparse, is given away in the off-diagonal entries [11]). Note also that these summary values, when computed by each site and shared, results in their disclosure at the site level.

Assume the covariates in X measure risk factors for a disease, and that the outcomes in y are a surrogate measure of the disease. Then extreme values in $X'y$ could be used to identify a population with high or low rates of the disease. For example, assume we have a covariate that records patients' blood alcohol level, and the outcome records the results of a liver enzyme test. High values from the liver enzyme test could indicate liver disease. If the summary values $X'y$ are high, then we may infer a population with high blood alcohol levels and the presence of liver disease (which could strongly suggest alcoholism). This is an example of the disclosure of personal information for participants at one site, and possibly source disclosure (i.e., if sites compare these summary values to their own participants, they could determine which site contains a group with high rates of alcoholism). We will see examples of disclosures from $X'X$ in other sections (i.e., from categorical data, or the covariance matrix).

If the estimated variance s^2 cannot be recovered for some reason, then we would state the disclosures were true up to a constant, which is still not an acceptable situation (e.g., extreme values would still be identifiable in the summary data). Concerns regarding the disclosure of these summary data have been raised many times [11-16].

2.3 Disclosures from Indicators

Indicators take on the binary values 0 or 1 in the columns of the design matrix X, indicating the absence or presence, respectively, of an attribute (e.g., gender male or female). They are also commonly found when one uses categorical variables in general (e.g., gender male, female, or unknown), which are transformed into a set of indicators in the design matrix for the sake of mathematical analysis (such that the resulting design matrix is of full rank, for matrix inversion). The structure of the design matrix would need to be defined in advance of performing multi-site regression so that the results of each site can be combined.

As previously noted, if an intercept is included in a model, then the first column of the design matrix will be 1s, and the first entry in the matrix $X^{'}X$ will indicate the total number of observations, i.e., $\left[X^{'}X\right]_{1,1} = N$. In fact, all entries $\left[X^{'}X\right]_{j,j}$ for indicator j represent the number of 1s in the column, regardless of whether an intercept is used.

When data is split among sites, it is plausible that one or more sites could have a column of 1s in their design matrix, or a large proportion of 1s. Equivalently, they could have a column of 0s, or a large proportion of 0s (i.e., a sparse column). For example, a site with data for a sensitive population, such as an abuse treatment center, may wish to participate in a study involving three or more groups, but an indicator that identifies this

population directly or indirectly would reveal which summary data belongs to them when pooling information (i.e., source identification).

Disclosures of this nature extend to the information matrix $\mathcal{I} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}$ of a generalized linear model, where $\boldsymbol{W}$ is the diagonal weight matrix. For example, if we assume an intercept is included, then $\left[\boldsymbol{X}'\boldsymbol{X}\right]_{j,j} \Big/ \left[\boldsymbol{X}'\boldsymbol{X}\right]_{1,1}$ gives the exact proportion of 1s for indicator j, and $\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{j,j} \Big/ \left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,1}$ gives an approximation. The approximation is especially good for indicators with a large number of 1s or 0s. For example, if an indicator is represented by a column of 1s then $\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{j,j} \Big/ \left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,1} = 1$; if an indicator is represented by a column of 0s then $\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{j,j} \Big/ \left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,1} = 0$ One can also use the bounds on the weights to help with the inference (e.g., for a logistic model the weights are between 0 and 1).

Consider the following design matrix $\boldsymbol{X}$, representing a data set with an intercept and three binary indicators, and an accompanying weight matrix $\boldsymbol{W}$ with off-diagonal entries of zero (created for this example from a random sample of the uniform distribution):

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \boldsymbol{W} = \text{diag} \begin{pmatrix} 0.874 \\ 0.461 \\ 0.888 \\ 0.834 \\ 0.429 \\ 0.887 \\ 0.139 \\ 0.954 \\ 0.390 \\ 0.702 \\ 0.250 \\ 0.545 \\ 0.761 \\ 0.171 \\ 0.509 \\ 0.944 \\ 0.955 \\ 0.799 \\ 0.813 \\ 0.941 \end{pmatrix}$$

Let the information matrix for the linear regression be $\mathcal{I}_{s^2}$, and for the generalized linear model $\mathcal{I}_{\boldsymbol{W}}$. Hence, for this example they are,

$$\mathcal{I}_{s^2} = \frac{1}{s^2}\begin{pmatrix} 20 & 5 & 12 & 10 \\ 5 & 5 & 2 & 2 \\ 12 & 2 & 12 & 4 \\ 10 & 2 & 4 & 10 \end{pmatrix}, \mathcal{I}_W = \begin{pmatrix} 13.25 & 3.488 & 6.616 & 7.253 \\ 3.488 & 3.488 & 1.303 & 1.335 \\ 6.616 & 1.303 & 6.616 & 2.336 \\ 7.253 & 1.335 & 2.336 & 7.253 \end{pmatrix}.$$

Notice how the diagonal entries of $s^2\mathcal{I}_{s^2} = \boldsymbol{X}'\boldsymbol{X}$ give the exact number of 1s in each column. Interactions are given in the off-diagonal entries, such as indicator 2 and 3 given by $\left[\mathcal{I}_{s^2}\right]_{2,3} = 2$, which implies that they have two rows in common with value 1. We scale the matrix $\mathcal{I}_W$ by the ratio of the number of observations to the sum of the weights $\left[\mathcal{I}_W\right]_{1,1}$, then round the resulting matrix to compare with $\mathcal{I}_{s^2}$:

$$\text{round}\left(\left(\frac{20}{\left[\mathcal{I}_W\right]_{1,1}}\right)\mathcal{I}_W\right) = \begin{pmatrix} 20 & 5 & 10 & 10 \\ 5 & 5 & 2 & 2 \\ 10 & 2 & 10 & 4 \\ 11 & 2 & 4 & 11 \end{pmatrix}$$

The results are strikingly similar to $\boldsymbol{X}'\boldsymbol{X}$. Sparse design matrices are particularly vulnerable to this form of transformation, due to the sparse interactions evident in the off-diagonal entries of the information matrix.

Coding indicators using values other than 0 and 1 does not eliminate the possibility of disclosures. For example, assume the values -1 and 1 are used, and an intercept is included. In this example, for any indicator $\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{j,j} = \left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,1}$, for a column of 1s $\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,j} = \left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,1}$, and for a column of -1s $\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,j} = -\left[\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right]_{1,1}$. In other words, although different criteria are used when we change the coding of indicators, we are still able to make inferences from the information matrix, leading to disclosures.

2.4 Disclosures from the Covariance Matrix

For highly correlated variables, knowledge of one implies knowledge of the other. Such information could therefore lead to inferential disclosures. The asymptotic covariance matrix of the parameter estimates is the inverse of the information matrix (i.e., for a generalized linear model, the estimated covariance matrix is given by $\mathcal{I}^{-1} = \left(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right)^{-1}$). Hence, in the case of multi-site regression, the information matrix shared by each individual site may put them at risk (which, combined with the site's score vector and the previous parameter estimates, can be used to derive parameter estimates specific to that site).

Reiter raises disclosure concerns with regards to standard errors of estimated model parameters [17, 18], and several researchers suggest that standard errors should not be provided if they are “too small” [11, 19]. Standard errors are derived from the diagonal entries of the estimated covariance matrix. Very small values imply a strong fit between covariates and the site’s data, giving information away regarding the strength of the model on that data.

One concern is that a model that fits data very well can lead to accurate, possibly even perfect, predictions which could be used by an adversary to match to an external dataset and re-identify individuals. Assume, for example, a model with a sensitive outcome, such as the presence of a rare or stigmatized disease (e.g., alcoholism). The parameter estimates could be used with known covariate values to predict outcomes. For observations with the same combination of covariate values (common with categorical data) and the same outcomes, the predictions on that same combination of values will be exact [17, 18].

Very small standard errors could also be used to directly or indirectly identify a sensitive population, potentially revealing information about them and the source of the data. This is similar to the concerns raised when using indicators, except that in this case we extend the measurement type to the nominal and interval scales. For example, we could have continuous variables that identify the population, such as one site that is known to have a focused age group.

A restriction on standard errors implies that confidence intervals cannot be provided, since standard errors can be uniquely derived from them. Similarly, p-values cannot be provided with parameter estimates, since standard errors can be uniquely derived from this information as well. Rather, it is suggested that levels of significance be provided (e.g., 0 < p < 0.005, 0.005 < p < 0.05, 0.05 < p < 0.1, etc.) [11, 19].

Although revealing standard errors is a concern at the site level, the purpose of multi-site regression is to pool data into a larger dataset. Therefore, restricting the disclosure of confidence intervals and p-values for the combined model may be excessive. Provided the information matrices for individual sites and the overall model are secure, there should not be a concern providing standard errors for the overall model (implying that confidence intervals and p-values can also be provided for the overall model).

2.5 Disclosures from Iterating

The iterative process of the fitting procedure implies an exchange of information between sites or a third party that can be exploited. The following example is taken from Fienberg et al. in [20], with regards to vertically partitioned data. It highlights the possibility of disclosures from shared summary data (e.g., the score vector) captured over multiple iterations, or when certain elements can be inferred from unique structures in the data, such as those we have outlined already.

Recall that sites need to share the k overall parameter estimates $\boldsymbol{b}^{(t)}$ in multiparty regression, for all iterations of $t = 1, \ldots, T$. Without loss of generality, we assume the canonical logit function of a generalized linear model (i.e., a logistic model), and that sites also share, or can access, the intermediate probability of success $p_i^{(t)} = \text{logit}^{-1}\left(\boldsymbol{X}_i \boldsymbol{b}^{(t)}\right)$ for observation i, where $\boldsymbol{X}_i$ is row i from the design matrix $\boldsymbol{X}$. To simplify notation, let $a_i^{(t)} = \text{logit}\left(p_i^{(t)}\right) = \boldsymbol{X}_i \boldsymbol{b}^{(t)}$.

Assume that there are at least as many iterations as parameters to estimate (i.e., $T \geq k$), and that we collect the parameter estimates $\boldsymbol{b}^{(t)}$ and summary value $a_i^{(t)}$ at each iteration t of the fitting procedure. Now consider the linear system $\boldsymbol{X}_i \boldsymbol{B} = \boldsymbol{a}_i^{'}$, where $\boldsymbol{B} = \left[\boldsymbol{b}^{(1)} \ldots \boldsymbol{b}^{(k)}\right]$, and $\boldsymbol{a}_i = \left(a_i^{(1)}, \ldots, a_i^{(k)}\right)^{'}$. If the columns of $\boldsymbol{B}$ are linearly independent, then we can solve the linear system such that $\boldsymbol{X}_i = \boldsymbol{a}_i^{'} \boldsymbol{B}^{-1}$, resulting in a full disclosure of row i. Note that the columns of $\boldsymbol{B}$ could be any collection of k linearly independent parameter estimates $\boldsymbol{b}^{(t)}$ during the iterative process (not simply the first k of them). Furthermore, Fienberg et al. suggest that similar concerns may appear in the calculation of the score vector $\boldsymbol{u}^{(t)} = \boldsymbol{X}^{'}\left(\boldsymbol{y} - \boldsymbol{p}^{(t)}\right)$ [20].

2.6 Disclosures from Multiple Models

Often one will build several models before deciding which provides the best fit for the purposes of explaining or predicting outcomes. For example, interactions may be included but then found not to be practically or statistically significant, and therefore dropped. The process of fitting multiple models, however, and sharing local summary data, can in itself lead to disclosures. This is a common subject of consideration in the study of remote access methods [11, 17-19].

For example, Sparks et al. in [11] demonstrate how as little as two strategically chosen linear models can be used to reconstruct the original data when the information matrix is provided. Consider the singular value decomposition of the design matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}_\lambda \boldsymbol{V}'$, where $\boldsymbol{U}^{'}\boldsymbol{U} = \boldsymbol{I}_k$, $\boldsymbol{V}^{'}\boldsymbol{V} = \boldsymbol{I}_k$ (i.e., $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary), and $\boldsymbol{D}_\lambda$ is a diagonal matrix of singular values. Then $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{D}_\lambda^{-1}$ and $\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1} = \boldsymbol{V}\boldsymbol{D}_\lambda^{-1}\boldsymbol{V}'$. If the information matrix $\boldsymbol{\mathcal{I}} = \frac{1}{s^2}\boldsymbol{X}^{'}\boldsymbol{X}$ is provided then the matrix $\boldsymbol{U}$ can be derived. A second linear model with $\boldsymbol{U}$ as the design matrix would lead to the disclosure of the linear predictor $\boldsymbol{Xb}$ (i.e., the predicted outcome for each individual). Moreover, adding the linear predictor to the residuals of the original model leads to the disclosure of the design matrix $\boldsymbol{X}$, with the original observations.

2.7 The Use of Secure Multi-Party Computation

In 2004, the use of secure multiparty computation in this context was proposed and demonstrated by Karr et al. in [12], and the literature has grown since that time.

Sharing local summary statistics with each other or a third party, such as the matrix $\boldsymbol{X}'\boldsymbol{X}$ and vector $\boldsymbol{X}'\boldsymbol{y}$, where $\boldsymbol{X}$ is the design matrix and $\boldsymbol{y}$ the outcomes, was deemed unacceptable, possibly even forbidden by law, by Karr et al. [12]. As they demonstrate, sites can calculate parameter estimates for their own data, and compare to the global results, but using secure multiparty computation ensures that they cannot determine which site may have caused deviations from their own. A detailed example of their approach for secure linear regression is given in [13].

In 2005 Karr et al. included generalized linear models to their approach using secure multiparty computation, which they acknowledge is virtually the same technique [14]. They observe that the approach has the advantage of "being resistant to source identification via attribute values" (e.g., high incomes in one data source).

Secure computation of the logistic regression has been proposed [21-23], when data is horizontally partitioned among multiple sites. However, these protocols assume that the final information matrix is known by all sites, which introduces disclosure risks in the log-linear case with two sites. If $\boldsymbol{X}'\boldsymbol{X}$ is known by a site then it can narrow down the possible values in the design matrix of the other site, especially if the original values are known to be integer (e.g., counts) and have a limited range. More generally, in the two site case one site would be able to determine the value of the information matrix for the other site, and this raises the disclosure risks noted above. Furthermore these protocols use a secure sum protocol which has known security weaknesses, as we illustrate below.

Secure sum works in a round-robin fashion. The first site selects a random value r and adds its own value to that: $x_1 + r$, and forwards that to the second site. The second site adds its own value and forwards it: $x_1 + x_2 + r$. This continues until the final sum reaches the first site: $x_1 + x_2 + \ldots + x_k + r$. The first site subtracts r from the total to obtain the sum. If there are two sites then it would be easy for the first site to determine the value x_2. If there are more than two sites, collusion among sites can also reveal the values from other sites. For example, if the second and fourth sites collude, then when the fourth site receives $x_1 + x_2 + x_3 + r$, she can send that value to the second site. The second site would subtract $x_1 + x_2 + r$ to reveal the value for the third site. Therefore, this protocol is not suitable for two parties or if all of the sites cannot be trusted. This weakness can in principle be fixed by modifying the protocol, but will increase the communication cost between each pair of parties [24].

In the general case with mixed partitions, the same assumptions discussed above were used to construct a distributed computation protocol [25]. Another protocol was proposed for linear regression on vertically partitioned data using a secure matrix product [26].

Therefore, previous attempts at constructing a protocol suitable for data would still allow some inappropriate disclosures under certain conditions or were not applicable to horizontally partitioned data.

2.8 Distributed Queries

An increasingly popular architecture for analyzing data located at multiple sites without pooling the data can be characterized as distributed data aggregation. Under this model a central user sends queries to multiple sites. Each site executes the query and sends the results back to the central user [27-30]. Such networks retain the data at the sites and only send summary information back to the central user in response to specific queries. However, it has been well established for some time that it is possible to leak sensitive information by running multiple overlapping queries on a database – this is known as the problem of inference from statistical databases [31]. The ability to extract sensitive information when multiple tables are available about the same population has been demonstrated on student data released by educational departments in the US [32] and on the re-identification of federal capital cases by combining information from multiple tables released by the Department of Justice [33].

The major problem in inference from statistical databases comes from queries that return a subset of data [34, 35]. The two main inference attacks are queries with very small or very large outputs, and query overlapping (inference chaining) [36, 37]. An attribute in a statistical database could be compromised exactly or partially. If there is no restriction on the user's queries, the exact value of a specific attribute could be deduced, especially if the user has some initial knowledge of the contents in the underlying table. Consider Table 1, named Diagnosis. Suppose that an adversary already knows that N11 is one of the patients in this data set. An adversary can run the following query:

```
SELECT COUNT(*)
FROM Diagnosis
WHERE Name='N11' AND DiagnosisCode BETWEEN 290 AND 319
```

This query would return a result set of size one, revealing the patient's diagnosis. If the adversary does not know the patient's name but the postal code instead then the following query will return the diagnosis:

```
SELECT COUNT(*)
FROM Diagnosis
WHERE PostalCode='K1L' AND DiagnosisCode BETWEEN 290 AND 319
```

These types of queries can be prevented by setting a restriction on the size of the result returned by the query. However, there are some inference methods, such as tracker queries [36], which are able to bypass this restriction by sending some related and overlapping queries, and combining the results to infer some individual data. More importantly, sometimes the number of queries needed for effective trackers can be small.

Attempts to protect against such inferences by restricting the size of the query result (query set) would not be effective. Query set size control will block any query which produces a query set smaller than some threshold, say k records. It also blocks any query set with more than $N-k$ records, where N is the total number of records in the table. This is because the complement of the query will have less than k records which violates the inference control limit. Now, suppose instead of the complement of a query with the query set of k records, we create a query which is the complement of another query, or more than one query, which in this case is not prohibited.

	Name	Age	Gender	Postal Code	Language	Diagnosis Code
1	N1	72	F	K1K	En	305
2	N2	63	F	K1N	En	294
3	N3	58	M	K1K	Fr	153
4	N4	56	F	K2K	En	231
5	N5	51	M	K1P	Fr	294
6	N6	48	M	K1R	It	282
7	N7	39	M	K2K	Fr	745
8	N8	38	F	K1P	En	523
9	N9	31	F	K1M	En	310
10	N10	29	M	K1K	En	312
11	N11	29	F	K1L	En	300
12	N12	23	M	K1J	It	482

Table 1: Example table to illustrate tracker queries.

Even with restrictions on the result set size, the adversary can run the following two queries:

```
SELECT COUNT (*)
FROM Diagnosis
```

```
WHERE Gender='F' AND Language='En' AND DiagnosisCode BETWEEN 290 AND 319
```

and:

```
SELECT COUNT (*)
FROM Diagnosis
WHERE Name < > 'N11' AND
        Gender='F' AND Language='En' AND DiagnosisCode BETWEEN 290 AND 319
```

If we assume that $k = 2$, both of the above queries have query sets between k and $N - k$ records, and therefore would not be blocked. However, by subtracting the results of the above two queries the adversary will obtain the singleton query set equal to the result of the prohibited query, and can figure out the range of N11's diagnosis code. This and other examples will lead us to set another stricter condition on the query set size. For instance, the allowable size for a query set has to be between $2k$ and $N - 2k$. However, consider the following two queries which are both allowed because of the size of the query sets, 7 and 5 records respectively:

```
SELECT COUNT (*)
FROM Diagnosis
WHERE Gender='M' OR Language='Fr'
```

and:

```
SELECT COUNT (*)
FROM Diagnosis
WHERE Gender < > 'M' AND Language < > 'Fr'
```

The adversary can pad a condition to each of the above queries as follows:

```
SELECT COUNT (*)
FROM Diagnosis
WHERE (Name < > 'N11' AND  DiagnosisCode BETWEEN 290 AND 319)
        OR (Gender='M' OR Language='Fr')
```

and:

```
SELECT COUNT (*)
FROM Diagnosis
WHERE (Name < > 'N11' AND  DiagnosisCode BETWEEN 290 AND 319)
        OR (Gender < > 'M' AND Language < > 'Fr')
```

The first query set has 8 records and the second one has 5. Now by adding up these two and subtracting from the total of the previous two queries, N11's diagnosis code will be confirmed.

Queries that reveal information are not limited to 'count' queries. Suppose we also know that N11's age is 72, then we can construct the following 'max' query:

```
SELECT MAX (Age)
FROM Diagnosis
WHERE DiagnosisCode BETWEEN 290 AND 319
```

By returning 72, the adversary can conclude that N11's diagnosis code is in a specific range. Similar conclusions are possible using 'min' queries. Also, 'sum' queries can be used. As a simple example, suppose S is a sensitive numeric field and we want to have the result from the following query:

```
SELECT SUM(S)
FROM Diagnosis
WHERE Name='N11'
```

This query has the query set size problem and cannot be processed. However, the following two eligible queries could be executed to get the private value:

```
SELECT SUM(S)
FROM Diagnosis
WHERE Name<>'N11' and Language='En'
```

and:

```
SELECT SUM(S)
FROM Diagnosis
WHERE Language='En'
```

By subtracting the results from the two above queries, the value of S for N11 will be disclosed.

These examples illustrate the ineffectiveness of inference control using query set size. The experimental results show that an adversary, who has some prior information about the distribution of attribute values in the underlying data sets, is able to find a general tracker with as few as one or two queries.

Alternatives to query set size restrictions have been proposed, but these require the perturbation of the data [38, 39]. Such perturbative techniques will reduce the accuracy of analyses based on the data, and are not known to be used in the distributed networks cited above.

2.9 Meta-analytic Approaches

One approach to combine models across multiple sites providing data is for each site to develop a model and then a central unit would perform a meta-analysis of the models to construct the final model.

Using Individual Patient (or Participant) Data (IPD) in a meta-analysis is widely considered the gold standard way in which to combine results from multiple studies or sites [40]. This means getting access to "raw data" on all patients from multiple studies or sites, so that investigators can build models on the original data. Privacy restrictions or concerns would prohibit the use of IPD in a meta-analysis as the raw data would need to be disclosed.

Restricting investigations to aggregated data limits regression modeling to summary data that may or may not be complete, and limited to study-level covariates [41, 42]. In general, IPD reviews are preferred because one can: include unpublished data (e.g., due to non-significant effects, or privacy concerns); analyse common outcomes and patient subgroups; analyse by time to event; and conduct a more flexible analysis [43]. Furthermore, it has been shown through simulation [44] and with real medical data [45, 46] that IPD analysis is generally necessary to relate patient characteristics to treatment. Using aggregated data in a meta-analytic regression with study-level covariates will typically have very low statistical power compared to an equivalent analysis using IPD, with different parameter estimates between the two [44].

Furthermore, as noted earlier in Section 2.4, sharing standard errors from site-specific models may reveal sensitive information about the population at that site under certain circumstances. The standard errors would need to be shared for regression model meta-analysis.

2.10 Other Methods

Alternative methods have been proposed to perform a pooled analysis in the context of drug and vaccine safety surveillance. A straight forward approach to perform a pooled analysis from multiple sites is to minimize or de-identify the data at source and then send the de-identified data to an AC for analysis [47-49]. De-identification always results in a loss of precision, and one must be careful that the de-identification applied is consistent across all sites *and* ensure that the risk of re-identification is acceptably low in the different jurisdictions of the sites.

Another proposed approach is the creation of propensity score models at each site, and then send these to the AC which would construct a final model with the propensity scores and other non-private variables [50, 51]. The authors show that pooling propensity scores provides equivalent results to a meta-analysis of model parameters. It is not clear what the loss in power and precision would be compared to an ideal pooled analysis with the original raw data.

2.11 Summary

The sharing of de-identified data to create a pooled data set and meta-analytics methods result in a loss of precision and power. Methods for multi-site regression would retain the precision and power. However, as noted above, current multi-site regression approaches are prone to inappropriate disclosure. Distributed networks that send queries to sites are prone to tracker queries that can reveal personal information.

Our objective was therefore to develop a multi-site regression protocol that addresses the disclosure risks by: (a) not revealing the individual site information matrix and score vectors, (b) avoiding inference channels through multiple overlapping queries, and (c) retaining the precision of a raw data pooled analysis. Our protocol uses secure multi-party computation methods. This is the SPARK protocol described below.

3 SPARK Protocol

3.1 Existing Building Blocks

Below we summarize the inputs and outputs for each of the secure building blocks that were already known and that we use in our protocol. A summary of the inputs and outputs is also provided in Table 2.

3.1.1 Secure Dot Product

For the multi-party case, secure dot product can be performed using secure multi-party multiplication.

- Number of parties: $2, \{P_1, P_2\}$
- Inputs: two private vectors of length m
  - $P_1 : V_1 = \langle v_{1,1}, v_{1,2}, \ldots, v_{1,m} \rangle$
  - $P_2 : V_2 = \langle v_{2,1}, v_{2,2}, \ldots, v_{2,m} \rangle$
- Outputs: two private numbers
  - $P_1 : a_1$
  - $P_2 : a_2$

  such that: $V_1 \cdot V_2 = a_1 + a_2$

3.1.2 Secure Matrix Multiplication for Two Parties

- Number of parties: $2, \{P_1, P_2\}$
- Inputs: two private matrices
  - $P_1 : A_1 = \begin{bmatrix} a_{1,1,1} & a_{1,1,2} & \ldots & a_{1,1,m} \\ a_{1,2,1} & a_{1,2,2} & \ldots & a_{1,2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,s,1} & a_{1,s,2} & \ldots & a_{1,s,m} \end{bmatrix}_{s \times m}$

- $P_2 : A_2 = \begin{bmatrix} a_{2,1,1} & a_{2,1,2} & \dots & a_{2,1,u} \\ a_{2,2,1} & a_{2,2,2} & \dots & a_{2,2,u} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2,m,1} & a_{2,m,2} & \dots & a_{2,m,u} \end{bmatrix}_{m \times u}$

o Outputs: two private matrices

- $P_1 : B_1 = \begin{bmatrix} b_{1,1,1} & b_{1,1,2} & \dots & b_{1,1,u} \\ b_{1,2,1} & b_{1,2,2} & \dots & b_{1,2,u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1,s,1} & b_{1,s,2} & \dots & b_{1,s,u} \end{bmatrix}_{s \times u}$
- $P_2 : B_2 = \begin{bmatrix} b_{2,1,1} & b_{2,1,2} & \dots & b_{2,1,u} \\ b_{2,2,1} & b_{2,2,2} & \dots & b_{2,2,u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{2,s,1} & b_{2,s,2} & \dots & b_{2,s,u} \end{bmatrix}_{s \times u}$

such that: $A_1 \times A_2 = B_1 + B_2$

3.1.3 Secure Multiparty Addition

o Number of parties: $k, \{P_1, P_2, \dots, P_k\}$

o Inputs: k private numbers

- $P_i : x_i, i \in \{1, 2, \dots, k\}$

o Outputs: k private numbers

- $P_i : y_i, i \in \{1, 2, \dots, k\}$

such that: $\sum_{i=1}^{k} x_i = \prod_{i=1}^{k} y_i$

3.1.4 Secure Multiparty Multiplication

o Number of parties: $k, \{P_1, P_2, \dots, P_k\}$

o Inputs: k private numbers

- $P_i : x_i, i \in \{1, 2, \dots, k\}$

o Outputs: k private numbers

- $P_i : y_i, i \in \{1, 2, \dots, k\}$

such that: $\prod_{i=1}^{k} x_i = \sum_{i=1}^{k} y_i$

3.1.5 Secure Matrix Sum Inverse for Two Parties

o Number of parties: $2, \{P_1, P_2\}$

o Inputs: 2 private matrices

- $P_1 : A_1 = \begin{bmatrix} a_{1,1,1} & a_{1,1,2} & \dots & a_{1,1,t} \\ a_{1,2,1} & a_{1,2,2} & \dots & a_{1,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,s,1} & a_{1,s,2} & \dots & a_{1,s,t} \end{bmatrix}_{s \times t}$
- $P_2 : A_2 = \begin{bmatrix} a_{2,1,1} & a_{2,1,2} & \dots & a_{2,1,t} \\ a_{2,2,1} & a_{2,2,2} & \dots & a_{2,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2,m,1} & a_{2,m,2} & \dots & a_{2,m,t} \end{bmatrix}_{s \times t}$

o Outputs: 2 private matrices

- $P_1 : B_1 = \begin{bmatrix} b_{1,1,1} & b_{1,1,2} & \dots & b_{1,1,t} \\ b_{1,2,1} & b_{1,2,2} & \dots & b_{1,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1,s,1} & b_{1,s,2} & \dots & b_{1,s,t} \end{bmatrix}_{s \times t}$
- $P_2 : B_2 = \begin{bmatrix} b_{2,1,1} & b_{2,1,2} & \dots & b_{2,1,t} \\ b_{2,2,1} & b_{2,2,2} & \dots & b_{2,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ b_{2,s,1} & b_{2,s,2} & \dots & b_{2,s,t} \end{bmatrix}_{s \times t}$

such that: $\left(A_1 + A_2\right)^{-1} = B_1 + B_2$

Secure Building Block	Number of Parties	Inputs	Outputs	Equation
Secure Dot Product	$2, \{P_1, P_2\}$	$P_i : V_i = \langle v_{i,1}, v_{i,2}, \ldots, v_{i,m} \rangle$	$P_i : a_i$	$V_1 \cdot V_2 = a_1 + a_2$
Secure Matrix Multiplication	$2, \{P_1, P_2\}$	$P_1 : A_{1(s \times t)}, P_2 : A_{2(t \times u)}$	$P_1 : B_{1(s \times u)}, P_2 : B_{2(s \times u)}$	$A_1 \times A_2 = B_1 + B_2$
Secure Multiparty Addition	$k, \{P_1, P_2, \ldots, P_k\}$	$P_i : x_i, i \in \{1, 2, \ldots, k\}$	$P_i : y_i, i \in \{1, 2, \ldots, k\}$	$\sum_{i=1}^{n} x_i = \prod_{i=1}^{n} y_i$
Secure Multiparty Multiplication	$k, \{P_1, P_2, \ldots, P_k\}$	$P_i : x_i, i \in \{1, 2, \ldots, k\}$	$P_i : y_i, i \in \{1, 2, \ldots, k\}$	$\prod_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$
Secure Matrix Sum Inverse	$2, \{P_1, P_2\}$	$P_1 : A_{1(s \times t)}, P_2 : A_{2(s \times t)}$	$P_1 : B_{1(s \times t)}, P_2 : B_{2(s \times t)}$	$(A_1 + A_2)^{-1} = B_1 + B_2$

Table 2: Summary of inputs and outputs of existing secure building blocks.

3.2 Secure Multi-party Matrix Sum Inverse

Suppose each of the sites has a $d \times d$ matrix and they jointly and securely want to compute the inverse matrix of the summation of their matrices in the form of the summation of private matrix shares:

$$\left(L_1 + \cdots + L_k\right)^{-1} = M_1 + \cdots + M_k \quad , \quad L_i, M_i \in P_i \qquad \text{..... (4)}$$

To simplify the presentation and without loss of generality, here we consider that $k = 3$. The protocol is similar in the case of $k > 3$. Note that it is implicitly assumed that the inverse exists for the summation matrix. The following are the major steps of this building block protocol:

1. Party P_3 randomly generates a non-singular $d \times d$ matrix, R_3.
2. Parties P_1 and P_3 perform secure matrix multiplication for their input matrices (see Section 3.1.2), L_1 and R_3, respectively, such that:

$$L_1 \times R_3 = N_1 + N_{3,1} \quad , \quad N_1 \to P_1 \ , \ N_{3,1} \to P_3 \qquad \text{..... (5)}$$

3. Parties P_2 and P_3 perform secure matrix multiplication for their input matrices (see Section 3.1.2), L_2 and R_3, respectively, such that:

$$L_2 \times R_3 = N_2 + N_{3,2} \quad , \quad N_2 \to P_2 \ , \ N_{3,2} \to P_3 \qquad \text{..... (6)}$$

4. P_3 selects a random number $0 < r < 1$ and sends $N_{3,1} + r\left(L_3 \times R_3\right)$ and $N_{3,2} + \left(1-r\right)\left(L_3 \times R_3\right)$ to P_1 and P_2, respectively.
5. P_1 locally computes $Q_1 = N_1 + N_{3,1} + r\left(L_3 \times R_3\right)$ which is equal to $L_1 \times R_3 + r\left(L_3 \times R_3\right)$, using equation (5).
6. P_2 locally computes $Q_2 = N_2 + N_{3,2} + \left(1-r\right)\left(L_3 \times R_3\right)$ which is equal to $L_2 \times R_3 + \left(1-r\right)\left(L_3 \times R_3\right)$, using equation (6).

Note that according to the equations in the steps 5 and 6 the following equations are satisfied:

$$Q_1 + Q_2 = L_1 \times R_3 + r\left(L_3 \times R_3\right) + L_2 \times R_3 + \left(1-r\right)\left(L_3 \times R_3\right) = \left(L_1 + L_2 + L_3\right) \times R_3$$

$$\Rightarrow \left(Q_1 + Q_2\right)^{-1} = \left(\left(L_1 + L_2 + L_3\right) \times R_3\right)^{-1} = R_3^{-1} \times \left(L_1 + L_2 + L_3\right)^{-1}.$$

7. P_1 and P_2 run secure matrix sum inverse for two parties on their private matrices, Q_1 and Q_2, respectively (see Section 3.1.5), such that:

$$\left(Q_1 + Q_2\right)^{-1} = T_1 + T_2 \quad , \quad T_1 \to P_1 \ , \ T_2 \to P_2 \qquad \text{..... (7)}$$

8. Parties P_1 and P_2 perform secure matrix multiplication for their matrices, T_1 and R_3, respectively, such that:

$$R_3 \times T_1 = M_{3,1} + M_1 \quad , \quad M_1 \to P_1 \ , \ M_{3,1} \to P_3 \qquad \text{..... (8)}$$

9. Parties P_2 and P_3 perform secure matrix multiplication for their matrices, T_2 and R_3, respectively, such that:

$$R_3 \times T_2 = M_{3,2} + M_2 \quad , \quad M_2 \to P_2 \ , \ M_{3,2} \to P_3 \qquad \text{..... (9)}$$

10. Then we have:

$$\begin{aligned}\left(L_1 + L_2 + L_3\right)^{-1} &= \left(R_3 \times R_3^{-1}\right) \times \left(L_1 + L_2 + L_3\right)^{-1} = R_3 \times \left(R_3^{-1} \times \left(L_1 + L_2 + L_3\right)^{-1}\right) \\ &= R_3 \times \left(Q_1 + Q_2\right)^{-1} = R_3 \times \left(T_1 + T_2\right) = \left(R_3 \times T_1\right) + \left(R_3 \times T_2\right) \\ &= M_{3,1} + M_1 + M_{3,2} + M_2 = M_1 + M_2 + M_3\end{aligned}$$

Note, that we set $M_3 = M_{3,1} + M_{3,2}$, such that $M_3 \to P_3$. Therefore, M_i is the final private output matrix share of $P_i, i \in \{1,2,3\}$.

3.3 Secure Logistic Function

Inside the secure logistic regression algorithm, the logistic function $\frac{e^x}{1+e^x}$ has to be computed, while x is privately shared among k parties, such that $x = \sum_{i=1}^{k} x_i$. To jointly and securely compute this value such that at the end each party P_i receives their private output share y_i, secure addition and multiplication building blocks are used as follows:

$$\frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-\sum_{i=1}^{k} x_i}} = \frac{1}{1+\prod_{i=1}^{k} e^{-x_i}} = \frac{1}{1+\sum_{i=1}^{k} b_i} = \prod_{i=1}^{k} c_i = \sum_{i=1}^{k} y_i \qquad \text{..... (10)}$$

The following are the steps for this secure building block:

1. Parties perform secure multi-party multiplication for their private inputs, x_i's, such that:

$$\prod_{i=1}^{k} e^{-x_i} = \sum_{i=1}^{k} b_i$$

2. They then execute secure inverse addition for their inputs b_i's such that:

$$\frac{1}{1+\sum_{i=1}^{k} b_i} = \prod_{i=1}^{k} c_i$$

3. Finally, they run secure multiplication for their private inputs, c_i's, as follows:

$$\prod_{i=1}^{k} c_i = \sum_{i=1}^{k} y_i$$

Note that the same solution could be applied on $\frac{e^x}{\left(1+e^x\right)^2} = \frac{1}{2+e^x+e^{-x}}$ to create separate private shares for the parties.

The computation of inverse addition is an extension of secure addition. We illustrate the protocol here for two parties, P_1 and P_2. Assume each party has a value b_1 and b_2 and we need to compute:

$$\frac{1}{\left(b_1+b_2\right)} = c_1 \times c_2$$

Then we execute the following steps:

1. Party P_1 computes $E\left(b_1\right)$ and a random number k_1 and sends these to P_2.
2. P_2 generates a pair of random numbers r_2 and k_2, and computes:

$$x_2 = r_2 \times 10^{k_2}$$

3. P_2 computes $\left[E\left(b_1\right) \times_h E\left(b_2\right)\right]^{x_2}$ and send that to P_1.
4. P_1 decrypts the value it receives.

At the end of the computation we have:

$$c_1 = \frac{10^{k_1}}{\left(b_1+b_2\right) \times x_2}$$

$$c_2 = r_2 \times 10^{\left(k_2-k_1\right)} = \frac{x_2}{10^{k_1}}$$

3.4 Secure multi-party 2-norm distance

There is an existing secure protocol in [52] for secure 2-norm distance. However, in that protocol, which is restricted to two parties, each party owns a vector and they want to jointly and securely compute the distance between their vectors. In our context, there is no restriction on the number of parties and each element in each vector is privately shared among the parties, such that each element is the summation of corresponding shares. Therefore we propose a protocol suitable for our context below.

In this building block the 2-norm distance of two vectors, α and β, is securely computed. Each item of these vectors is the summation of the private values belonging to k parties, as follows:

$$\alpha = \left(\alpha_0 \quad \cdots \quad \alpha_v\right) = \left(\sum_{i=1}^{k} \alpha_{i,0} \quad \cdots \quad \sum_{i=1}^{k} \alpha_{i,v}\right) \quad , \quad \beta = \left(\beta_0 \quad \cdots \quad \beta_v\right) = \left(\sum_{i=1}^{k} \beta_{i,0} \quad \cdots \quad \sum_{i=1}^{k} \beta_{i,v}\right).$$

where v is the number of items in the vector. At the end each party has their private output share such that the summation of the shares would be the 2-norm distance of the given vectors. Thus, the following computation has to be jointly and securely done by the parties:

$$\|\alpha,\beta\| = \sqrt{\sum_{i=0}^{v}(\alpha_i-\beta_i)^2} = \sqrt{\sum_{i=0}^{v}\left((\alpha_{1,i}+\cdots+\alpha_{k,i})-(\beta_{1,i}+\cdots+\beta_{k,i})\right)^2}$$

$$= \sqrt{\sum_{i=0}^{v}\left((\alpha_{1,i}-\beta_{1,i})+\cdots+(\alpha_{k,i}-\beta_{k,i})\right)^2} = \sqrt{\sum_{i=0}^{v}(\delta_{1,i}+\cdots+\delta_{k,i})^2}\ .$$

By assuming $\delta_{j,i}=\alpha_{j,i}-\beta_{j,i}$. Because we only need the distance to compare with a constant value in our protocol, we could ignore the square root and simply continue as follows:

$$\sum_{i=0}^{v}(\delta_{1,i}+\cdots+\delta_{k,i})^2 = \sum_{i=0}^{v}\delta_{1,i}^2+\cdots+\sum_{i=0}^{v}\delta_{k,i}^2+2\left(\sum_{i=0}^{v}\delta_{1,i}\delta_{2,i}+\cdots+\sum_{i=0}^{v}\delta_{k-1,i}\delta_{k,i}\right).$$

Therefore, the following are the major steps of this sub-protocol:

1. For each $\sum_{i=0}^{v}\delta_{j,i}\delta_{l,i}$, $j<l$, the two parties involved have to perform secure dot product to obtain their own private shares as follows:

$$\sum_{i=0}^{v}\delta_{j,i}\delta_{l,i} = \lambda_j+\lambda_l\ .$$

2. Each party P_j sets their private output share as $\varphi_j = 2\lambda_j+\sum_{i=0}^{v}\delta_{j,i}^2$. Therefore:

$$\|\alpha,\beta\|^2 = \sum_{i=0}^{v}(\alpha_i-\beta_i)^2 = \sum_{i=0}^{v}(\delta_{1,i}+\cdots+\delta_{k,i})^2 = \sum_{i=0}^{v}\delta_{1,i}^2+\cdots+\sum_{i=0}^{v}\delta_{k,i}^2+2\left(\sum_{i=0}^{v}\delta_{1,i}\delta_{2,i}+\cdots+\sum_{i=0}^{v}\delta_{k-1,i}\delta_{k,i}\right)=\sum_{j=1}^{k}\varphi_j$$

where the final squared distance is the sum of the φ_j distributed among the k parties.

3.5 Secure Multi-party Comparison

Secure comparison has been proposed in previous papers, such as in [53, 54]. However, those protocols are suitable in the case where we want to compare the two private values between two parties. In our context we wish to compare the summation of private shares among multiple parties with a constant value.

In this sub-protocol k parties will securely compare the summation of their private shares with a constant value τ, by using a central unit as a commodity server, without revealing their input values to each other and the central unit. The central unit remains unaware about the final result of the comparison, and only the parties will receive the final result.

1. The central unit establishes a pair of encryption keys using the Paillier cryptosystem and broadcasts the public key to all the parties.
2. The central unit generates one random number r_j for each party P_j, encrypts it, and privately sends $E(r_j)$ to that party (i.e., this random number is not shared among the parties).
3. Starting from the first party, each P_j encrypts their private input φ_j, computes $E(\varphi_j)\times_h E(r_j)$, and multiplies the result by the received value from the previous party, and sends the result to the next party. This means, for instance P_j will send $\prod_{i=1}^{j}E(\varphi_i)\times_h E(r_i)$ to P_{j+1}.

4. The last party, P_k, after performing the same operations, multiplies the final value by $E(\tau)^{-1}$, which produces the following encrypted value:

$$E(\tau)^{-1} \times_h \prod_{i=1}^{k} \left(E(\varphi_i) \times_h E(r_i) \right) = E\left(\left(\sum_{i=1}^{k} \varphi_i \right) + \left(\sum_{i=1}^{k} r_i \right) - \tau \right)$$

Then, P_k generates a large random list, inserts the above value to the list, performs a random permutation, and sends the list to the central unit for decryption.

5. The central unit decrypts each item of the list, subtracts $\sum_{j=1}^{k} r_j$ from each item, and returns the decrypted list back to P_k.

6. Party P_k, after applying the permutation on the received list, retrieves the decrypted item which corresponds to the encrypted value in the sending list, and obtains the comparison result.

7. Party P_k broadcasts the result to the other parties.

3.6 Description of Protocol

The SPARK protocol assumes a central analysis center (the AC) that would coordinate and combine the analysis results from all of the sites. It is not necessary for the AC to be trusted as it would not be able to view any of the individual sites' data nor able to draw inferences from information it receives.

The pattern of communications would be between each site and the AC, and also among the sites. The latter is required to implement some of the building block protocols, such as secure comparison.

3.6.1 Secure Logistic Regression

We let the main design matrix be denoted $\boldsymbol{X}$, containing v independent attributes in each record, and the vector of the values of the dependent attribute, $\boldsymbol{y}$, are as follows:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_k \end{pmatrix}, \ \boldsymbol{X}_i = \begin{pmatrix} X_{i,1} \\ X_{i,2} \\ \vdots \\ X_{i,N_i} \end{pmatrix} \rightarrow P_i, \ X_{ij} = \begin{pmatrix} 1 & x_{ij1} & x_{ij2} & \cdots & x_{ijv} \end{pmatrix}, \ \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix}, \ \boldsymbol{y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{iN_i} \end{pmatrix}.$$

Note, that $N = \sum_{i=1}^{k} N_i$, and each $\boldsymbol{X}_i$ is a $N_i \times (v+1)$ matrix.

The regression coefficient vector, $\mathbf{b}$, will be kept private and each party will obtain a private share of this vector. Then, this vector would be reconstructed by the AC to produce the final vector and perform the required analyses. Each distributed coefficient value is indexed as b_{is}, where i indexes the site, and s indexes the attribute. Therefore, we use the following notation:

$$\boldsymbol{b} = \begin{pmatrix} b_{+0} & b_{+1} & \cdots & b_{+v} \end{pmatrix}' \quad , \quad b_{+s} = \sum_{i=1}^{k} b_{is} \quad , \quad b_{ij} \rightarrow P_i .$$

This means that every item of $\boldsymbol{b}$ is the summation of the private shares belonging to the sites. The steps of the protocol are as follows:

1. Each party P_i initiates their shares for the regression coefficients, a zero vector, such that $\boldsymbol{b} = \begin{pmatrix} 0 & 0 & \cdots & 0 \end{pmatrix}'$.

2. Compute the fitted probabilities vector $\boldsymbol{p}$. We define: $\boldsymbol{p} = \begin{pmatrix} \frac{e^{X_{i1}\times\boldsymbol{b}}}{1+e^{X_{i1}\times\boldsymbol{b}}} & \frac{e^{X_{i2}\times\boldsymbol{b}}}{1+e^{X_{i2}\times\boldsymbol{b}}} & \cdots & \frac{e^{X_{iN_i}\times\boldsymbol{b}}}{1+e^{X_{iN_i}\times\boldsymbol{b}}} \end{pmatrix}'$

First, for each record $X_{ij} \rightarrow P_i$ all the sites jointly compute $X_{ij} \times \boldsymbol{b}$. These computations are not local and the sites have to securely collaborate to create their private shares. According to the distribution of the vector $\boldsymbol{b}$, we have:

$$
\begin{aligned}
X_{ij} \times \boldsymbol{b} &= \begin{pmatrix} 1 & x_{ij1} & x_{ij2} & \cdots & x_{ijv} \end{pmatrix} \times \begin{pmatrix} b_{+0} & b_{+1} & \cdots & b_{+v} \end{pmatrix}' \\
&= \begin{pmatrix} 1 & x_{ij1} & x_{ij2} & \cdots & x_{ijv} \end{pmatrix} \times \begin{pmatrix} \sum_{i=1}^{k} b_{i0} & \sum_{i=1}^{k} b_{i1} & \cdots & \sum_{i=1}^{k} b_{iv} \end{pmatrix}' \\
&= \sum_{i=1}^{k} b_{i0} + x_{ij1} \sum_{i=1}^{k} b_{i1} + \cdots + x_{ijv} \sum_{i=1}^{k} b_{iv} \\
&= \left(b_{10} + \cdots + b_{k0} \right) + \left(x_{ij1} b_{11} + \cdots + x_{ijv} b_{1v} \right) + \cdots + \left(x_{ij1} b_{k1} + \cdots + x_{ijv} b_{kv} \right) \qquad \text{..... (11)}
\end{aligned}
$$

Each $\left(x_{ij1} b_{l1} + \cdots + x_{ijv} b_{lv} \right)$ is securely computed using secure dot product between the two sites P_i and P_j. To compute equation (11) a total of $(k-1)$ secure dot products must be performed on vectors of size $(v-1)$. Then $X_{ij} \times \boldsymbol{b}$ is converted to the summation of private output shares:

$$X_{ij} \times \boldsymbol{b} = a_{1j} + \cdots + a_{kj} \quad , \qquad a_{ij} \rightarrow P_i$$

By applying the secure logistic function presented in Section 3.3 on the above results as inputs each site will receive their private share for the vector $\boldsymbol{p}$.

3. Computing the weights matrix $\boldsymbol{W}$. The weight matrix is an $N_i \times N_i$ diagonal square matrix, denoted as follows:

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{W}_k \end{pmatrix}_{k \times k} .$$

By using the secure logistic function described in Section 3.3, each element of this matrix will be the summation of the private shares:

$$\boldsymbol{W}_i = \begin{pmatrix} \sum_{i=1}^{k} w_{i1} & & 0 \\ & \ddots & \\ 0 & & \sum_{i=1}^{k} w_{iN_i} \end{pmatrix}_{N_i \times N_i}$$

$$\sum_{i=1}^{k} w_{il} = \frac{e^{X_{il}\times \boldsymbol{b}}}{\left(1+e^{X_{il}\times \boldsymbol{b}}\right)^2} \quad , \qquad l \in \{1,\cdots,N_i\} .$$

4. Computation of $\boldsymbol{X}'\times\boldsymbol{W}$. In the following matrix multiplication, the product can be separated into k matrix multiplications.

$$\boldsymbol{X}'\times\boldsymbol{W} = \begin{pmatrix} \boldsymbol{X}_1' & \cdots & \boldsymbol{X}_k' \end{pmatrix} \times \begin{pmatrix} \boldsymbol{W}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{W}_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1'\times\boldsymbol{W}_1 & \cdots & \boldsymbol{X}_k'\times\boldsymbol{W}_k \end{pmatrix}$$

$$= \left(\begin{pmatrix} 1 & \cdots & 1 \\ x_{1,1} & \cdots & x_{N_1,1} \\ \vdots & \ddots & \vdots \\ x_{1,v} & \cdots & x_{N_1,v} \end{pmatrix} \times \begin{pmatrix} \sum_{j=1}^{k} w_{1,j,1} & & 0 \\ & \ddots & \\ 0 & & \sum_{j=1}^{k} w_{1,j,N_i} \end{pmatrix} \cdots \begin{pmatrix} 1 & \cdots & 1 \\ x_{k,1} & \cdots & x_{N_k,1} \\ \vdots & \ddots & \vdots \\ x_{k,v} & \cdots & x_{N_k,v} \end{pmatrix} \times \begin{pmatrix} \sum_{j=1}^{k} w_{k,j,1} & & 0 \\ & \ddots & \\ 0 & & \sum_{j=1}^{k} w_{k,j,N_k} \end{pmatrix} \right)$$

$$= \left(\begin{pmatrix} \sum_{j=1}^{k} w_{1,j,1} & \cdots & \sum_{j=1}^{k} w_{1,j,N_1} \\ x_{1,1}\sum_{j=1}^{k} w_{1,j,1} & \cdots & x_{N_1,1}\sum_{j=1}^{k} w_{1,j,N_1} \\ \vdots & \ddots & \vdots \\ x_{1,v}\sum_{j=1}^{k} w_{1,j,1} & \cdots & x_{N_1,v}\sum_{j=1}^{k} w_{1,j,N_1} \end{pmatrix} \cdots \begin{pmatrix} \sum_{j=1}^{k} w_{k,j,1} & \cdots & \sum_{j=1}^{k} w_{k,j,N_k} \\ x_{k,1}\sum_{j=1}^{k} w_{k,j,1} & \cdots & x_{N_k,1}\sum_{j=1}^{k} w_{k,j,N_k} \\ \vdots & \ddots & \vdots \\ x_{k,v}\sum_{j=1}^{k} w_{k,j,1} & \cdots & x_{N_k,v}\sum_{j=1}^{k} w_{k,j,N_k} \end{pmatrix} \right) .$$

The first row of the above matrix is already the summation of the private shares. Each of the items in the rest of the matrix could be transformed to a summation of the shares using the secure multiplication building block. Therefore, at the end, $\boldsymbol{X}'\times\boldsymbol{W}$ will be the following matrix:

$$\boldsymbol{X}'\times\boldsymbol{W} = \begin{pmatrix} \sum_{j=1}^{k} w'_{j,0,1} & \cdots & \sum_{j=1}^{k} w'_{j,0,N} \\ \sum_{j=1}^{k} w'_{j,1,1} & \cdots & \sum_{j=1}^{k} w'_{j,1,N} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{k} w'_{j,v,1} & \cdots & \sum_{j=1}^{k} w'_{j,v,N} \end{pmatrix}_{(v+1)\times N} = \boldsymbol{W}_1' + \cdots + \boldsymbol{W}_k'$$

Each $\boldsymbol{W}_i'$ is a $(v+1)\times N$ matrix.

5. Computation of $\boldsymbol{X}'\times\boldsymbol{W}\times\boldsymbol{X}$: This matrix multiplication is also the same as the previous step, and at the end, each item of the matrix would be the summation of the private shares:

$$\boldsymbol{X}' \times \boldsymbol{W} \times \boldsymbol{X} = \begin{pmatrix} \sum_{j=1}^{k} x'_{j,0,0} & \cdots & \sum_{j=1}^{k} x'_{j,0,v} \\ \sum_{j=1}^{k} x'_{j,1,0} & \cdots & \sum_{j=1}^{k} x'_{j,1,v} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{k} x'_{j,v,0} & \cdots & \sum_{j=1}^{k} x'_{j,v,v} \end{pmatrix}_{(v+1)\times(v+1)} = \boldsymbol{Z}_1 + \cdots + \boldsymbol{Z}_k$$

in which each $\boldsymbol{Z}_i$ is a $(v+1)\times(v+1)$ matrix.

6. Computation of $\left(\boldsymbol{X}' \times \boldsymbol{W} \times \boldsymbol{X}\right)^{-1}$. The secure computation of this step is based on the building block described in Section 3.2. Each matrix of the above summation belongs to one party and the inverse of this matrix summation has to be calculated. Suppose we denote each $\left(\boldsymbol{X}_i' \times \boldsymbol{W}_i \times \boldsymbol{X}_i\right)$ by $\boldsymbol{Z}_i$. Therefore, by using the secure multi-party matrix sum inverse, the following equation would be obtained: $\left(\boldsymbol{Z}_1 + \cdots + \boldsymbol{Z}_k\right)^{-1} = \boldsymbol{T}_1 + \cdots + \boldsymbol{T}_k$ in which each $\boldsymbol{T}_i \to P_i$, and is a $(v+1)\times(v+1)$ matrix.

7. Computation of $\boldsymbol{y} - \boldsymbol{p}$. Because each item in $\boldsymbol{p}$ is shared among the parties, this matrix subtraction could be converted to the summation of the private shares as follows:

$$\boldsymbol{y}_i - \boldsymbol{p}_i = \left(y_{i,1} - \sum_{j=1}^{k} b_{i,j,1} \quad \cdots \quad y_{i,k} - \sum_{j=1}^{k} b_{i,j,N_i} \right)' = \left(\sum_{j=1}^{k} b'_{i,j,1} \quad \cdots \quad \sum_{j=1}^{k} b'_{i,j,N_i} \right)' .$$

8. Computation of $\boldsymbol{X}' \times \left(\boldsymbol{y} - \boldsymbol{p}\right)$: Similar to the step 5 , this matrix-vector product will be computed to generate the following matrix:

$$\boldsymbol{X}' \times \left(\boldsymbol{y} - \boldsymbol{p}\right) = \left(\boldsymbol{X}_1' \times \left(\sum_{j=1}^{k} b'_{1,j,1} \quad \cdots \quad \sum_{j=1}^{k} b'_{1,j,N_i} \right)' \right) + \cdots + \left(\boldsymbol{X}_k' \times \left(\sum_{j=1}^{k} b'_{k,j,1} \quad \cdots \quad \sum_{j=1}^{k} b'_{k,j,N_i} \right)' \right)$$

$$= \left(\sum_{j=1}^{k} b''_{i,j,0} \quad \cdots \quad \sum_{j=1}^{k} b''_{i,j,v} \right)' = \boldsymbol{U}_1 + \cdots + \boldsymbol{U}_k \ .$$

Each $\boldsymbol{U}_i$ is a $(v+1)$-dimension vector.

9. Computation of $\left(\boldsymbol{X}' \times \boldsymbol{W} \times \boldsymbol{X}\right)^{-1} \times \boldsymbol{X}' \times \left(\boldsymbol{y} - \boldsymbol{p}\right)$. In this step, a matrix-vector multiplication has to be done, in which the matrix is the summation of k matrices, and each item of the vector is the summation of k items, as follows:

$$\left(\boldsymbol{X}' \times \boldsymbol{W} \times \boldsymbol{X}\right)^{-1} \times \boldsymbol{X}' \times \left(\boldsymbol{y} - \boldsymbol{p}\right) = \left(\boldsymbol{T}_1 + \cdots + \boldsymbol{T}_k\right) \times \left(\boldsymbol{U}_1 + \cdots + \boldsymbol{U}_k\right) = \boldsymbol{V}_1 + \cdots + \boldsymbol{V}_k \ .$$

In the above equation, $\boldsymbol{V}_1 + \cdots + \boldsymbol{V}_k$ is a $(v+1)$-dimension vector, and the matrix-vector product is jointly and securely computed by the k parties using $\frac{k(k-1)}{2} SDP(2 \times v)$, in which $SDP(a)$ is the secure dot product of two a-dimension vectors.

10. The new value of the vector $\boldsymbol{b}$ will be computed as:

$$\boldsymbol{b}+\left(\boldsymbol{X}'\times\boldsymbol{W}\times\boldsymbol{X}\right)^{-1}\times\boldsymbol{X}'\times\left(\boldsymbol{y}-\boldsymbol{p}\right)=\boldsymbol{b}+\boldsymbol{V}_1+\cdots+\boldsymbol{V}_k$$

$$=\left(\sum_{i=1}^{k}b_{i,0}\quad\sum_{i=1}^{k}b_{i,1}\quad\cdots\quad\sum_{i=1}^{k}b_{i,v}\right)^T+\left(\sum_{i=1}^{k}V_{i,0}\quad\sum_{i=1}^{k}V_{i,1}\quad\cdots\quad\sum_{i=1}^{k}V_{i,v}\right)^T$$

$$=\left(\sum_{i=1}^{k}b'_{i,0}\quad\sum_{i=1}^{k}b'_{i,1}\quad\cdots\quad\sum_{i=1}^{k}b'_{i,v}\right)'$$

in which each $b'_{i,j}=b_{i,j}+V_{i,j}$.

11. Loop termination. At the end of the loop the distance of the old and new $\boldsymbol{b}$ has to be compared with the specified threshold, say τ. For this comparison, different distance functions could be used. Here, we use 2-norm distance. Therefore, first by using the secure building block presented in Section 3.4, secure multi-party 2-norm distance of the two vectors, $\boldsymbol{\alpha}=\boldsymbol{b}^{New}$ and $\boldsymbol{\gamma}=\boldsymbol{b}^{Old}$, will be computed, such that:

$$\boldsymbol{\alpha}=\left(\alpha_0\quad\cdots\quad\alpha_v\right)=\left(\sum_{i=1}^{k}\alpha_{i,0}\quad\cdots\quad\sum_{i=1}^{k}\alpha_{i,v}\right)\quad,\quad\boldsymbol{\gamma}=\left(\gamma_0\quad\cdots\quad\gamma_v\right)=\left(\sum_{i=1}^{k}\gamma_{i,0}\quad\cdots\quad\sum_{i=1}^{k}\gamma_{i,v}\right)$$

and at the end of the sub-protocol:

$$\left\|\boldsymbol{\alpha},\boldsymbol{\gamma}\right\|=\sum_{j=1}^{k}\varphi_j\qquad,\qquad\varphi_j\rightarrow P_j\ .$$

For the comparison, the secure sub-protocol presented in Section 3.5 is used to compare $\sum_{j=1}^{k}\varphi_j$ and τ. This comparison is done jointly by the parties and the AC.

The main protocol is terminated according to the comparison result in step 11, if the distance between the new and old regression vectors is smaller than the threshold, and otherwise the protocol is repeated from step 2.

4 Security Analysis

In our proposed protocol the sites perform computations on their distributed and secure data. It is therefore a reasonable assumption that they correctly follow the protocol steps and provide valid data in their data communications with other sites. However, any site may also use the intermediate results received from the others to compute or infer private information. This means that all of the parties are considered *semi-honest*.

Because the protocols are usually complex and different operations are used during their steps, the composition theorem [55, 56] is used for their security proof. The base sub-protocols in the algorithms presented in this paper are secure dot product, secure addition and multiplication, and secure comparison. Security proofs for these building blocks could be found in their corresponding papers. The exception is secure comparison, which is discussed in this section. Also, collusion attacks between two or more sites, or one or more sites and the central unit are examined.

4.1 Secure Multi-party Comparison:

To prove the security of this building block, we use the Simulation paradigm [55, 57]. Using this paradigm, a protocol is considered secure if for any party involved in the protocol all the information that can be gained by the party could also be acquired from their own inputs and outputs. Thus, for each party P a simulator S has to be found such that the party's view using the secure protocol is computationally indistinguishable [55] from the output of that simulator.

- Central unit: The first two steps are done by the central unit. Thus, the first part of the simulator S_{CU} for this party would be:

  Input: Nothing
  Process:
  - Establishing the encryption keys
  - Selecting random numbers $r_1, \cdots, r_k$
  - Encrypting the random numbers

  Output: $\{PK, E(r_1), \cdots, E(r_k)\}$

  In the output, PK is the public key.

- $P_1, \cdots, P_{k-1}$: In step 3 each party, except the first one, receives an encrypted message from the previous party, performs an encryption and a multiplication, and sends the result to the next party. Note that an encrypted message received by a party is considered a random number, because they do not know the private key and therefore are not able to decrypt the message. Thus, the simulator S_{P_1} for the first party is:

  Input: $\{PK, E(r_1)\}$
  Process:
  - Encrypting her private input φ_1
  - Performing $E(\varphi_1) \times_h E(r_1)$

  Output: $\{E(\varphi_1) \times_h E(r_1)\}$

  Also, the simulator S_{P_j} for each party $P_j, j \in \{2, \cdots, k-1\}$ will be as follows:

  Input: $\left\{PK, E(r_j), \prod_{l=1}^{j-1} \left(E(\varphi_l) \times_h E(r_l)\right)\right\}$
  Process:
  - Encrypting their private input φ_j
  - Performing $E(\varphi_j) \times_h E(r_j) \times_h \prod_{l=1}^{j-1} \left(E(\varphi_l) \times_h E(r_l)\right)$

  Output: $\left\{\prod_{l=1}^{j} \left(E(\varphi_l) \times_h E(r_l)\right)\right\}$

- P_k : In step 4 the last party performs the same operations as the previous parties have done. The last party also hides the result into a random list and sends it to the central unit. The first part of the simulator S_{P_k} for this party would be:

  Input: $\left\{PK, E(r_k), \prod_{l=1}^{k-1} \left(E(\varphi_l) \times_h E(r_l)\right)\right\}$
  Process:
  - Encrypting their private input φ_k
  - Performing $E(\tau)^{-1} \times_h E(\varphi_k) \times_h E(r_k) \times_h \left(\prod_{l=1}^{k-1} \left(E(\varphi_l) \times_h E(r_l)\right)\right)$
  - Generating a random list

- Hiding $\prod_{l=1}^{k} E(\varphi_l) \times_h E(r_l)$ in the random list

Output: A random list of numbers

- Central unit: The second part of the simulator for this party is as follows:

Input: A random list of numbers
Process:
- Decrypting all the items in the list
- Subtracting $\sum_{j=1}^{k} r_j$ from each decrypted item
- Sending back the result to the last party

Output: A random list of numbers

- P_k: The second part of the simulator S_{P_k} is:

Input: A random list of numbers
Process:
- Selecting the corresponding item from the list

Output: The comparison result

4.2 Collusion Attacks

In the protocols that involve more than two parties, collusion between two or more parties could be used to extract one or more other parties' private data. Here we discuss the possibility of such attacks applied to the sub-protocols presented in this paper, along with the main protocol.

4.2.1 Secure Multi-party Comparison

- Collusion between the central unit and one party: if the first party, P_1, colludes with the central unit, no private data from the other parties will be accessed because no information is sent from them to the first party. In case of collusion of the central unit with a party, P_j, other than the first one, the only information revealed to them is $\sum_{l=1}^{j-1} \varphi_l$, but no individual data, except in the case of $j = 2$.

  If the central unit colludes with P_k, then the comparison result is also revealed to the central unit.

- Collusion between two parties P_i and P_j: No site knows the private key of the cryptosystem which is generated by the central analysis unit. Therefore, even by collusion, no useful and meaningful information about the other parties' private data could be revealed to the colluding parties.

4.2.2 Secure Matrix Sum Inverse

- P_1 and P_2: By colluding these two parties do not extract private information from P_3, because R_3 is a random value known only by P_3, and all the rest of the intermediate values are based on that value as well as L_3, which is also unknown to those two colluded parties.
- P_1 and P_3: In equation (6), both L_2 and N_2 are known only to P_2, and therefore collusion of P_1 and P_3, will not be revealed these values. Following the same reasoning, Q_2, M_2 and T_2 could not be accessed by these two parties.

- P_2 and P_3 : The same reasoning as the previous case applies here.

4.2.3 Secure 2-norm Distance

In this sub-protocol, we only use the secure dot product between each pair of parties. This building block is applied to only two parties and therefore there is no collusion attack risk for this secure building block.

4.2.4 Main protocol

For evaluating collusion attacks, we consider three different scenarios:

i. Collusion between (k-1) parties: Suppose there are k parties who keep the data and run the protocol to produce their own private shares of the output values of the regression coefficient vector. If k-1 parties collude, we go through the steps of the protocol to investigate any possible security issues:
   1. There is no security problem in this step, because each party just initializes their own private shares for the regression coefficient vector by 0's.
   2. For each record $X_{i,j} \in P_i$ all the parties jointly compute $X_{i,j} \times \boldsymbol{b}$. If $X_{i,j}$ belongs to one of the colluding parties, at the end of this step, they only know their own output shares, and will not know the output share of that single party, because of the use of the secure dot product. In the case that $X_{i,j}$ belongs to that single party, again none of their output shares from the secure dot products between that party and other parties will be revealed to those parties. However, this is not correct for the first iteration, in which all the private share values of the vector $\boldsymbol{b}$ are zero. Therefore, to prevent any possible collusion attack, each party randomly selects very small and non-zero values for their input shares of the coefficient vector instead of zeros. In this way, even in the first iteration, private shares of the single party from the vector are unknown to the colluding parties.
   3. To compute the weight matrix, the secure logistic function is used by the parties. In this building block, secure addition and multiplication are used to convert the logistic function to a summation of output private shares. Again, because the colluding parties do not know the private input of the single party, they are not able to figure out the final output share of the single party because of the resistance of those building blocks to collusion attacks.
   4,5. In these steps, secure matrix multiplication is only used, which is safe against collusion attacks.
   6. Secure matrix sum inverse is used in this step, which has already been investigated for collusion attacks above, and found to be safe.
   7. There is no joint operation in this step and therefore, there is no collusion attack.
   8. Same as step 4, and safe.
   9. In this step, secure dot product is used between every two parties. If this is done between two of the colluding parties, they will only know their own shares, and if it is done between one of those parties and the single party, at the end, the private share of the single party is kept unknown to the colluding parties because of the secure dot product is safe to collusion attack.
   10. There is no joint operation in this step and therefore, there is no collusion attack.
   11. In this step, secure 2-norm distance and secure comparison are used between the parties and their resistance to collusion attacks has been already investigated above and found to be safe.

ii. Collusion between all the parties: If all the parties collude, they can jointly compute the regression coefficient vector from their private data. This is actually the non-privacy-preserving protocol.

iii. Collusion between one or more parties and the central unit: If there are only two parties who have data and want to run the protocol to find their private shares of the regression vector, then collusion of the central unit with one of those parties will reveal the private shares of the other party to the central unit. In general, this is the case if the central unit colludes with all the parties except one. However, if there are less than (k-1) parties who collude with the central unit, then the maximum information the central unit will know is the summation of the private shares of the non-colluding parties, which will not help the central unit to extract their separate private shares.

5 Complexity Analysis

In this section we analyze the complexity of each sub-protocol and building block used in them. We first show the complexity of the building blocks and then use those as the basis for the complexity analysis of the main protocol. The building blocks used in this work are secure addition (SA), multiplication (SM), dot product (SDP), matrix product (SMP), matrix sum inverse (SMSI), logistic function (SLF), 2-norm distance (SND), and comparison (SC). The complexity of each protocol is shown in terms of the number of sent messages, encryptions, and decryptions.

In Table 3, k is the number of parties involved, v is the number of independent variables, d is the dimension of a square matrix, and l is the number of items in the random list inside the secure comparison protocol. We only apply secure matrix product in the secure multi-party sum inverse sub-protocol, and therefore we only deal with the square matrix, which is why we consider only one dimension for the matrix multiplication.

Secure Building Blocks	Sent Messages	Encryptions	Decryptions
Addition $\left(SA(k)\right)$	$k(k-1)$	$\frac{(k+2)(k-1)}{2}$	$\frac{k(k-1)}{2}$
Multiplication $\left(SM(k)\right)$	$k(k-1)$	$k(k-1)$	$\frac{k(k-1)}{2}$
Dot Product $\left(SDP(k)\right)$	$k+1$	$k+1$	1
Matrix Product $\left(SMP(d)\right)$	$2\times d^2$	$2\times d^2$	d^2
Matrix Sum Inverse $\left(SMSI(d)\right)$	$14\times d^2$	$12\times d^2$	$6\times d^2$
Logistic Function $\left(SLF(k)\right)$	$3k(k-1)$	$5k^2-3k-2$	$\frac{3k(k-1)}{2}$
2-Norm Distance $\left(SND(k,v)\right)$	$\frac{k(k-1)(v+2)}{2}$	$\frac{k(k-1)(v+2)}{2}$	$\frac{k(k-1)}{2}$
Comparison $\left(SC(k)\right)$	$(2\times k)-1+2l$	$2\times k$	l

Table 3: Complexity analyses of the secure building blocks.

To find the complexity of the main protocol, we go through its steps in Section 3.2 to compute the cost for each iteration of the algorithm. The initial computations in step 1, and local computations in steps 7 and 10 are negligible compared to the other steps and therefore are ignored in the overall cost of the protocol.

- Step 2: $N(k-1)SDP(v-1)$ to compute the secure product of the vectors with $v-1$ items among each pair of the parties.
- Step 3: $SLF(k)$ to apply secure logistic function on the private shares of k parties.
- Step 3: $SLF(k)$ to apply secure logistic function on the private shares of k parties.
- Steps 4, 5, and 8: $\left(v\times N(k-1)+k(v+1)(v+2)\right)SM(k)$ to perform secure matrix multiplications, when each item of the matrices is the summation of private shares.

- Step 6: $SMSI(v+1)$ to do secure matrix sum inverse on matrices with $(v+1)\times(v+1)$ dimensions.
- Step 9: $\frac{k(k-1)}{2}SDP(2v)$ to perform secure matrix-vector multiplication, in which the matrix is the summation of k matrices, and each item of the vector is the summation of k items.
- Step 11: $SND(k,v)+SC(k)$ to apply secure 2-norm distance on k parties sharing two vectors with v items.

Therefore, the overall complexity of the main SPARK protocol based on the computations in a single iteration is given by:

$$\frac{k(k-1)}{2}SDP(2\times v)+SMSI(v+1)+N(k-1)SDP(v-1)+2SLF(k)+SND(k,v)$$
$$+\left(v\times N(k-1)+k(v+1)(v+2)\right)SM(k)+SC(k)$$

As can be seen, the order of the communication costs in both protocols mostly depends on the number of parties involved, k, and the number of independent variables, v. Also, because a portion of the computation depends on the number of data records, we could use parallel operations between different parties to make the protocol faster in real-world applications with a very large number of records.

6 Extensions to Other Generalized Linear Models

The basic protocol we have presented here can be extended to other Generalized Linear Models (GLM) [1]. The link functions for other GLMs are simpler than the logit function, as illustrated in Table 4. Secure computation of the link functions could be applied using the secure building blocks in this paper. In the Poisson log function, for example, we only need to compute the exponent of the product of regression vector and design matrix, which we already have in our protocol.

Name	Function
Identity	μ
Reciprocal	$1/\mu$
Reciprocal squared	$1/\mu^2$
Square root	$\sqrt{\mu}$
Log	$\ln(\mu)$
Complementary log-log	$\ln(-\ln(\mu))$
Logit	$\ln(\mu/(1-\mu))$

Table 4: Examples of link functions.

As another example consider the reciprocal function, which is used with the exponential gamma distribution. With this link function, the reciprocal of each item of the $\boldsymbol{Xb}$ vector has to be computed. Each item of the vector is the summation of the private shares:

$$\boldsymbol{X}_{ij}\times\boldsymbol{b}=a_{i1}+\cdots+a_{ik}$$

Thus, we have to securely compute the private shares of the parties, b_{ij}'s, such that:

$$\left(a_{i1}+\cdots+a_{ik}\right)^{-1}=\frac{1}{a_{i1}+\cdots+a_{ik}}=b_{i1}\times\cdots\times b_{ik}$$

This could be done using the secure inverse addition, which is actually a subset of the secure logistic function proposed in this paper as a secure building block.

For other link functions and distributions the secure computation protocol depends on the explicit operations we have to perform, and could be done on a case by case basis.

For the weight matrix, $\boldsymbol{W}$, we have the same situation. That is, there is a different function to compute the items of the diagonal weight matrix depending on the link function and variance of the distribution. The diagonal weight matrix is derived from equation:

$$w_i=\frac{1}{\mathrm{var}\left(\mu_i\right)\left(g'\left(\mu_i\right)\right)^2}. \qquad \text{..... (12)}$$

Applying the link function to different distributions, we get Table 5.

Distribution	Link	Variance	Weight
Bernoulli	$\ln\left(\mu/(1-\mu)\right)$	$\mu(1-\mu)$	$\mu(1-\mu)$
	$\ln(\mu)$	$\mu(1-\mu)$	$\mu/(1-\mu)$
	$\ln\left(-\ln(\mu)\right)$	$\mu(1-\mu)$	$\ln(\mu)/(1-\mu)$
Binomial (k)	$\ln\left(\mu/(k-\mu)\right)$	$\mu(1-\mu/k)$	$\mu(k-\mu)/k$
Poisson	$\ln(\mu)$	μ	μ
	μ	μ	$1/\mu$
	$\sqrt{\mu}$	μ	$2\mu^2$
Negative binomial (k)	$\ln\left(\mu/(k+\mu)\right)$	$\mu+\mu^2/k$	$\mu(k+\mu)/k$
	$\ln(\mu)$	$\mu+\mu^2/k$	$(k+\mu)/(k\mu)$
Normal	μ	1	1
	$\ln(\mu)$	1	μ^2
	$1/\mu$	1	μ^4
Gamma	$1/\mu$	μ^2	μ^2
	μ	μ^2	$1/\mu^2$
	$\ln(\mu)$	μ^2	1
Inverse Gaussian	$1/\mu^2$	μ^3	$\mu^2/2$
	$1/\mu$	μ^3	μ
	$\ln(\mu)$	μ^3	$1/\mu$

Table 5: The link functions applied to GLM distributions.

7 Extension to Generalized Estimating Equations

7.1 Overview of GEEs

We consider the case where there are repeated measures on a subject (which could be a site, facility, or an individual), such that each subject i has $j = 1, \cdots, N_i$ observations, $\sum_{i=1}^{k} N_i = N$. In this case Generalized Estimating Equations (GEE's) are used to adjust for the clustering by subject, resulting in a modified iterative procedure [50]. For subjects $i = 1, \cdots, k$ we have:

$$\boldsymbol{b}_z = \boldsymbol{b}_{z-1} - \left(\sum_{i=1}^{k} D_i^T \widehat{V}_i^{-1} D_i \right)^{-1} \left(\sum_{i=1}^{k} D_i^T \widehat{V}_i^{-1} \left(y_i - \boldsymbol{p}_i \right) \right) \quad \text{..... (13)}$$

where $\boldsymbol{D}_i = \partial \boldsymbol{\mu}_i / \partial \boldsymbol{\beta} = \boldsymbol{W}_i \Delta_i \boldsymbol{X}_i$, and $\boldsymbol{V}_i = \phi \boldsymbol{W}_i^{1/2} \boldsymbol{R}_i(\boldsymbol{\alpha}) \boldsymbol{W}_i^{1/2}$. Further, $\boldsymbol{W}_i = \mathrm{diag}_j \left(\nu \left(\boldsymbol{p}_{ij} \right) \right)$ is a $N_i \times N_i$ diagonal matrix of the variance function (based on the distribution), and $\Delta_i = \mathrm{diag}\left(\partial \theta_{ij} / \partial \eta_{ij} \right)$ is a $N_i \times N_i$ matrix, although $\Delta_i = \boldsymbol{I}_i$ for canonical link functions [58]. The dispersion parameter ϕ and correlation parameters $\boldsymbol{\alpha}$, which fully characterize the "working" correlation matrix $\boldsymbol{R}_i(\boldsymbol{\alpha})$, can be estimated from the standardized (or Pearson) residuals,

$$e_{ij} = \frac{\left(y_{ij} - \hat{\boldsymbol{p}}_{ij} \right)}{\sqrt{\nu \left(\hat{\boldsymbol{p}}_{ij} \right)}}.$$

Given the standardized residuals, we can estimate the dispersion parameter using

$$\hat{\phi} = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{N_i} \sum_{j=1}^{N_i} e_{ij}^2.$$

The correlation parameters depend on the selected correlation structure. They can be found using the estimates provided in the following table [59].

Structure	*$\mathrm{Corr}\left(Y_{ij}, Y_{ik} \right)$	$\hat{\alpha}$
Independence	0	Not required
Exchangeable	α	$\frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i (1 - n_i)} \sum_{j \neq k} e_{ij} e_{ik}$
AR(1)	$\alpha^{\lvert j-k \rvert}$	$\frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i - 1} \sum_{j \leq n_i - 1} e_{ij} e_{i, j+1}$
Unstructured	α_{jk}	$\frac{1}{N} \sum_{i=1}^{N} e_{ij} e_{ik}$

*Correlation along main diagonal is $\mathrm{Corr}\left(Y_{it}, Y_{it} \right) = 1.$

Table 6: Structure definition for working correlation and estimates.

We assume a canonical link function, and use logistic as our primary example. Our variance function will therefore be $\nu(\boldsymbol{p}_i) = \boldsymbol{p}_i(1-\boldsymbol{p}_i)$ for the binomial distribution.

1. Use Newton-Raphson method to get initial parameter estimates $\boldsymbol{b}_0$ (i.e., assuming independence correlation structure).
2. Calculate the standardized residuals $\boldsymbol{e}$, then the dispersion parameter $\hat{\phi}$ and the correlation parameters $\hat{\boldsymbol{\alpha}}$ from Table 6 (based on the selected structural definition).
3. Calculate the working covariance matrix $\hat{\boldsymbol{V}}_i = \hat{\phi}\boldsymbol{W}_i^{1/2}\boldsymbol{R}_i(\hat{\boldsymbol{\alpha}})\boldsymbol{W}_i^{1/2}$, where $\boldsymbol{R}_i(\hat{\boldsymbol{\alpha}})$ is determined by the selected structural definition, and $\boldsymbol{W}_i = \mathrm{diag}_j\left(\nu(\hat{\boldsymbol{p}}_{ij})\right) = \mathrm{diag}\left(\hat{\boldsymbol{p}}_i(1-\hat{\boldsymbol{p}}_i)\right)$.
4. Update the parameter estimates $\boldsymbol{b}_z$, where $\boldsymbol{D}_i = \boldsymbol{W}_i\boldsymbol{X}_i$ (since we assume a canonical link function).
5. Iterate steps 2 to 4 until convergence.
6. Calculate the estimated covariance matrix of the parameter estimates $\boldsymbol{\beta}_z$ using the "sandwich" estimator given by

$$\mathrm{Var}(\boldsymbol{b}_z) = \left(\sum_{i=1}^{k}\boldsymbol{D}_i^T\hat{\boldsymbol{V}}_i^{-1}\boldsymbol{D}_i\right)^{-1}\left(\sum_{i=1}^{k}\boldsymbol{D}_i^T\hat{\boldsymbol{V}}_i^{-1}(\boldsymbol{y}_i-\boldsymbol{p}_i)(\boldsymbol{y}_i-\boldsymbol{p}_i)^T\hat{\boldsymbol{V}}_i^{-1}\boldsymbol{D}_i\right)\left(\sum_{i=1}^{k}\boldsymbol{D}_i^T\hat{\boldsymbol{V}}_i^{-1}\boldsymbol{D}_i\right)^{-1}.$$

This empirically-adjusted covariance uses the data $\boldsymbol{y}_i$ sandwiched by the model-based covariance to adjust standard errors in case the true covariance is very different from the working assumptions. The standard errors for the parameter estimates are the diagonal entries of the above equation.

7.2 Secure Generalized Estimating Equations

There are some differences between the GEE's method and logistic regression method presented in the previous section for the computation of the items in the coefficient vector. In GEE's method, in each iteration, there is first an inner computation for each cluster, and then using those computed values the new values of coefficient vector's item are calculated. Another difference is that in the independence correlation structure, $\boldsymbol{R}(\boldsymbol{\alpha}) = I$ and the dispersion parameter is assumed 1, i.e. $\phi = 1$.

Because of the horizontally partitioned data amongst the parties, each $D_i^T\widehat{V_i}^{-1}D_i$ could be locally computed. The same situation is satisfied for $D_i^T\widehat{V_i}^{-1}(y_i-\mu_i)$. Therefore, equation (13) will be converted to

$$\boldsymbol{b}_z = \boldsymbol{b}_{z-1} - \left(\sum_{i=1}^{k}G_i\right)^{-1}\left(\sum_{i=1}^{k}H_i\right),$$

in which $G_i = D_i^T\widehat{V_i}^{-1}D_i$ and $H_i = D_i^T\widehat{V_i}^{-1}(y_i-\boldsymbol{p}_i)$. Now, the computation has been reduced to a secure multi-party matrix sum inverse, explained in the Section 3.2 for $\left(\sum_{i=1}^{k}G_i\right)^{-1}$ such that $\left(\sum_{i=1}^{k}G_i\right)^{-1} = \sum_{i=1}^{k}L_i$, and then a multiplication of a matrix and a vector, $\sum_{i=1}^{k}L_i$ and $\sum_{i=1}^{k}H_i$, each of which are the summation of private shares. This multiplication could be securely converted to the summation of private vectors using secure dot product for the calculation of each item. Thus, the main equation will be transformed to

$$\boldsymbol{b}_z = \boldsymbol{b}_{z-1} - \sum_{i=1}^{k}M_i$$

Note, that $\sum_{i=1}^{k} L_i$ is a $(v+1)\times(v+1)$ matrix and, $\sum_{i=1}^{k} H_i$ is a vector with $(v+1)$ elements, and therefore $\sum_{i=1}^{k} M_i$ would be a vector with $(v+1)$ elements, and the equation will be transformed to the summation of k vectors that separately belong to the parties.

The computation of items in $\boldsymbol{p}$, $\boldsymbol{W}$ and $\boldsymbol{y}-\boldsymbol{p}$ are the same as those in the logistic regression protocol.

The overall complexity of GEE protocol based on the computations in a single iteration is given by

$$\frac{k(k-1)}{2} SDP(2\times v) + SMSI(v+1) + 2SLF(k) + SND(k,v) + SC(k)$$

8 References

1. Agresti A. Categorical Data Analysis. 2nd ed. 2002; New York: Wiley.
2. Agresti A. Categorical Data Analysis. Wiley Series in Probability and Statistics. 2002, Wiley: Hoboken, New Jersey.
3. Aman Goel JJ. Comparing Various Optimization Algorithms for Binary Logistic Regression. Machine Learning Course Project Paper. 2010, University of Southern California: Los Angeles. p. 5.
4. Maro J, Platt R, Holmes J, Strom B, Hennessy S, Lazarus R, Brown J. Design of a national dsitributed health data network. Annals of Internal Medicine, 2009; 151:341-344.
5. Go A, Magid D, Wells B, Sung S, Cassidy-Bushrow A, Greenlee R, Langer R, Lieu T, Margolis K, Masoudi F, McNeal C, Murata G, Newton K, Novotny R, Reynolds K, Roblin D, Smith D, Vupputuri S, White R, Olson J, Rumsfeld J, Gurwitz J. The Cardiovascular Research Network: a new paradigm for cardiovascular quality and outcomes research. Circulation: Cardiovascular quality and outcomes, 2008; 1(2):138-147.
6. Davis R, Kolczak M, Lewis E, Nordin J, Goodman M, Shay D, Platt R, Black S, Shinefield H, Chen R. Active surveillance of vaccine safety: a system to detect early signs of adverse events. Epidemiology, 2005; 16(3):336-341.
7. Chen R, Glasser J, Rhodes P, Davis R, Barlow W, Thompson R, Mullooly J, Black S, Shinefield H, Vadheim C, Marcy S, Ward J, Wise R, Wassilak S, Hadler S. Vaccine Safety Datalink project: a new tool for improving vaccine safety monitoring in the United States. The Vaccine Safety Datalink Team. Pediatrics, 1997; 99(6):765-773.
8. Du W, Han Y, Chen S. Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. Proceedings of the Fourth SIAM International Conference on Data Mining. 2004.
9. Wolfson M, Wallace S, Masca N, Rowe G, Sheehan N, Ferretti V, LaFlamme P, Tobin4 M, Macleod J, Little J, Fortier I, Knoppers B, Burton PR. DataSHIELD: Resolving a Conflict in Contemporary Bioscience—Performing a Pooled Analysis of Individual-Level Data Without Sharing the Data. International Journal of Epidemiology, 2010; 39(5):1372-1382.
10. Wu Y, Jiang X, Kim J, Ohno-Machado L. Grid Binary LOgistic REgression (GLORE): Building Shared Models Without Sharing Data. Journal of the American Medical Informatics Association, 2012.
11. Sparks R, Carter C, Donnelly J, O'Keefe C, Duncan J, Keighley T, McAullay D. Remote Access Methods for Exploratory Data Analysis and Statistical Modelling: Privacy-Preserving Analytics. Computer Methods and Programs in Biomedicine, 2008; 91(3):208-222.
12. Karr A, Lin X, Sanil A, Reiter J. Analysis of Integrated Data without Data Integration. Chance, 2004; 17(3):27-30.
13. Karr A, Feng J, Lin X, Sanil A, Young S, Reiter J. Secure Analysis of Distributed Chemical Databases Without Data Integration. Journal of Computer-Aided Molecular Design, 2005; 19(9-10):739-747.
14. Karr A, Lin X, Sanil A, Reiter J. Secure Regression on Distributed Databases. Journal of Computational and Graphical Statistics, 2005; 14(2):263-279.
15. Fulp SFW, Slavković A, Wrobel T. "Secure" Log-linear and Logistic Regression Analysis of Distributed Databases. Privacy in Statistical Databases. 2006.

16. Slavković A, Nardi Y, Tibbits M. “Secure” Logistic Regression of Horizontally and Vertically Partitioned Distributed Databases. IEEE International Conference on Data Mining. 2007.
17. Reiter J. New Approaches to Data Dissemination: A Glimpse into the Future. Chance, 2004; 17(3):12-16.
18. Reiter J, Kohnen C. Categorical Data Regression Diagnostics for Remote Access Servers. Journal of Statistical Computation and Simulation, 2005; 75(11):889-903.
19. O’Keefe C, Good N. Regression Output From a Remote Server. Data & Knowledge Engineering, 2009; 68(11):1175-1186.
20. Fienberg S, Nardi Y, Slavković A. Valid Statistical Analysis for Logistic Regression with Multiple Sources, in *Protecting Persons While Protecting the People*, G. Cecilia, P. Kantor, and M. Lesk, Editors. 2009; Springer-Verlag: Berlin. p. 82-94.
21. Fienberg SE, Fulp WJ, Slavkovic AB, Wrobel TA. "Secure" log-linear and logistic regression analysis of distributed databases. PSD 2006, J. Domingo-Ferrer, Franconi, L., Editor. 2006, Springer, Heidelberg. p. 277-290.
22. Karr AF, Fulp WJ, Vera F, Young SS, Lin X, Reiter JP. Secure, privacy-preserving analysis of distributed databases. Technometrics, 2007; 49(3):335-345.
23. Karr AF. Secure Statistical Analysis of Distributed Databases, Emphasizing What We Don't Know. Journal of Privacy and Confidentiality, 2009; 1(2):197-211.
24. Kantarcioglu M, Clifton C. Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Transactions on Knowledge and Data Engineering, 2004; 16(9):1026-1037.
25. Fienberg SE, Karr AF, Nardi Y, Slavkovic AB. Secure Logistic Regression with Multi-Party Distributed Databases. Joint Statistical Meetings. 2007: Salt Lake City, Utah, USA. p. 3506-3513.
26. Karr AF, Lin X, Reiter JP, Sanil AP. privacy-preserving analysis of vertically partitioned data using secure matrix products. Journal of Official Statistics, 2009; 25(1):125–138.
27. Brown J, Holmes J, Shah K, Hall K, Lazarus R, Platt R. Distributed health networks: A practical and preferred approach to multi-institutional evaluations of comparative effectiveness, safety, and quality of care. 48, 2010; 6 Suppl 1(S45-S51).
28. Behrman R, Benner J, Brown J, McClellan M, Woodcock J, Platt R. Developing the Sentinel System - A national resource for evidence development. New England Journal of Medicine, 2011; 364(6):498-499.
29. Platt R, Wilson M, Chan K, Benner J, Marchibroda J, McClellan M. The new sentinel network - Improving the evidence of medical-product safety. New England Journal of Medicine, 2009; 361(7):645-647.
30. Platt R, Davis R, Finkelstein J, Go A, Gurwitz J, Roblin D, Soumerai S, Ross-Degnan D, Andrade S, Goodman M, Martinson B, Raebel M, Smith D, Ulcickas-Yood M, Chan K. Multicenter epidemiologic and health services research on therapeutics in the HMO Research Network Center for Education and Research on Therapeutics. Pharmacoepidemiology and drug safety, 2001; 10(5):373-377.
31. Adam N, Wortman J. Security-control methods for statistical databases: A comparative study. ACM Computing Surveys, 1989; 21(4):515-556.
32. Muralidhar K, Sarathy R. Privacy Violations in Accountability Data Released to the Public by State Educational Agencies. Federal Committee on Statistical Methodology Research Conference. 2009.
33. Algranati D, Kadane J. Extracting Confidential Information from Public Documents: The 2000 Department of Justice Report on the Federal Use of the Death Penalty in the United States. Journal of Official Statistics, 2004; 20(1):97-113.
34. Chin F. Security problems on inference control for SUM, MAX, and MIN queries. ACM, 1986; 33(3):451–464.
35. Chin FY, GZSOYO~LU G. Auditing and inference control in statistical databases. IEEE Transaction in Software Engineering, 1982; 8(6):574-582.
36. Denning DE, Denning PJ, Schwartz MD. The tracker: a threat to statistical database security. ACM Transactions on Database Systems (TODS), 1979; 4(1):76-96.
37. Domingo-Ferrer J. Inference Control in Statistical Databases: From Theory to Practice. Lecture Notes in Computer Science, 2002; 2316.
38. Schatz JM. Survey of Techniques for Securing Statistical Databases. 1998; University of California at Davis.
39. Gopal R, Garfinkel R, Goes P. Confidentiality via Camouflage: The CVC Approach to Disclosure Limitation When Answering Queries to Databases. Operational Research, 2002; 50(3):501-516.
40. Sutton AJ, Higgins JPT. Recent Developments in Meta-analysis. Statistics in Medicine, 2008; 27(5):625-650.

41. Lyman G, Kuderer N. The strengths and limitations of meta-analyses based on aggregate data. BMC Medical Research Methodology, 2005; 5(1).
42. Friedenreich C. Methods for pooled analyses of epidemilogic studies. Epidemiology, 1993; 4(4):295-302.
43. Clarke MJ, Stewart LA. Obtaining Individual Patient Data from Randomised Controlled Trials, in *Systematic Reviews in Health Care: Meta-analysis in Context*, G.D.S. Matthias Egger, Douglas G Altman, Editor. 2001; BMJ Publishing Group: London. p. 109-121.
44. Lambert PC, Sutton AJ, Abrams KR, Jones DR. A Comparison of Summary Patient-level Covariates in Meta-regression with Individual Patient Data Meta-analysis. Journal of Clinical Epidemiology, 2002; 55(1):86–94.
45. Berlin JA, Santanna J, Schmid CH, Szczech LA, Feldman HI. Individual Patient- Versus Group-level Data Meta-regressions for the Investigation of Treatment Effect Modifiers: Ecological Bias Rears its Ugly Head. Statistics in Medicine, 2002; 21(3):371-387.
46. Teramukai S, Matsuyama Y, Mizuno S, Sakamoto J. Individual Patient-level and Study-level Meta-analysis for Investigating Modifiers of Treatment Effect. Japanese Journal of Clinical Oncology, 2004; 34(12):717–721.
47. Coloma P, Schuemie M, Trifiro G, Gini R, Herings R, Hippsley-Cox J, Mazzaglia G, Giaquinto C, Corrao G, Pedersen L, Jvan der Lei J, Sturkenboom N. Combining electronic healthcare databases in Europe to allow for large-scale drug safety monitoring: the EU-ADR project. Pharmacoepidemiology and drug safety, 2011; 20:1-11.
48. Velentgas P, Bohn R, Brown J, Chan A, Gladowski P, Holick C, Kramer J, Nakasato C, Spettell C, Walker A, Zhang F, Platt R. A distributed research network model for post-marketing safety studies: the Meningococcal Vaccine Study. Pharmacoepidemiology and drug safety, 2008; 17:1226-1234.
49. Magid D, Gurwitz J, Rumsfeld J, Go A. Creating a research data network for cardiovascular disease: the CVRN. Expert Review of Cardiovascular Therapy, 2008; 6(8):1043-1045.
50. Rassen J, Avorn J, Schneeweiss S. Multivariare-adjusted pharmacoepidemiologic analyses of confirntial information pooled from multiple health care utilization databases. Pharmacoepidemiology and drug safety, 2010.
51. Rassen J, Solomon D, Curtis J, Herrinton L, Schneeweiss S. Privacy-maintaining propepensity score-based pooling of multiple databases applied to a study of biologics. Medical Care, 2010; 48(6 Suppl):S83-S89.
52. Inan A, Kantarcioglu M, Bertino E, Scannapieco M. A Hybrid Approach to Private Record Linkage. IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008. 2008. 2008.
53. Yao A. Protocols for secure computations (extended abstract). Foundations of Computer Science. 1982.
54. Malkhi D, Nisan N, Pinkas B, Sella Y. Fairplay—a secure two-party computation system. Proceedings of the 13th conference on USENIX Security Symposium. 2004.
55. Goldreich O. Foundations of Cryptography: Volume 2, Basic Applications. 2004; New York, NY, USA: Cambridge University Press.
56. Canetti R. Security and Composition of Multiparty Cryptographic Protocols. Journal of Cryptology, 2000; 13(1):143-202.
57. Goldwasser S, Micali S, Rackoff C. The Knowledge Complexity of Interactive Proof Systems. SIAM Journal on Computing, 1989; 18(1):186-208.
58. Agresti A. Categorical Data Analysis. 2nd ed. 2002; Hoboken, New Jersey: Wiley.
59. Molenberghs G, Verbeke G. Models for Discrete Longitudinal Data. Springer, 2010.