RESEARCH ARTICLE

Supporting Regularized Logistic Regression Privately and Efficiently

Wenfa Li[1], Hongzhe Liu[1], Peng Yang[1], Wei Xie[2]*

1 Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, 100101, China, **2** Department of Electrical Engineering & Computer Science, Vanderbilt University, Nashville, TN 37232, United States of America

* wei.xie@vanderbilt.edu

Abstract

As one of the most popular statistical and machine learning models, logistic regression with regularization has found wide adoption in biomedicine, social sciences, information technology, and so on. These domains often involve data of human subjects that are contingent upon strict privacy regulations. Concerns over data privacy make it increasingly difficult to coordinate and conduct large-scale collaborative studies, which typically rely on cross-institution data sharing and joint analysis. Our work here focuses on safeguarding regularized logistic regression, a widely-used statistical model while at the same time has not been investigated from a data security and privacy perspective. We consider a common use scenario of multi-institution collaborative studies, such as in the form of research consortia or networks as widely seen in genetics, epidemiology, social sciences, etc. To make our privacy-enhancing solution practical, we demonstrate a non-conventional and computationally efficient method leveraging distributing computing and strong cryptography to provide comprehensive protection over individual-level and summary data. Extensive empirical evaluations on several studies validate the privacy guarantee, efficiency and scalability of our proposal. We also discuss the practical implications of our solution for large-scale studies and applications from various disciplines, including genetic and biomedical studies, smart grid, network analysis, etc.

Introduction

The ever-increasing amount of data have posed significant demand for effective analytical methods to sift through them. Logistic regression and its regularized variants [1, 2] are among the most widely-used statistical models in data analysis. It has seen a wide range of applications across various human endeavors, including genetics and genomics (e.g., genome-wide association studies, or GWAS [3], gene-gene interaction detection [4]), epidemiology (e.g., [5, 6]), social sciences [7, 8], information technology (e.g., computational advertising on the internet [9] and personalized recommender systems [10]), etc.

Many of the aforementioned disciplines and applications rely on huge numbers of data records (commonly referred to as large sample sizes in many fields) to make reliable discoveries or

predictions. The scale of data desired is often beyond the capability of any single institution, and thus depends heavily on collaboration across different institutions through data collection, data sharing and collaborative analysis.

However, data sharing and collaborative studies across different institutions bring about serious privacy concerns, as most such studies involve raw data of human subjects that are considered private and sensitive. In biomedicine, for instance, individual patient records are highly sensitive and protected under stringent regulations such as the Health Insurance Portability and Accountability Act (HIPAA) [11]; Genetic information of humans are also deemed highly sensitive [12, 13] and partially covered by the Genetic Information Nondiscrimination Act (GINA) [14]; in the education domain, students' data privacy is strictly regulated under the Family Education Rights and Privacy Act (FERPA) [15]. In other domains, ignorance of data privacy and misuse of personal information has even outraged users [16] and raised awareness of regulators [17], as in the case of targeted internet advertising. Meanwhile, various high-profile data breaches [18, 19] have exacerbated the situation, damaging the credibility of centralized data hosts and analytical centers in upholding user privacy.

A classical approach to alleviating privacy concerns is by concealing individual raw data via artificial perturbation (e.g., k-anonymity [20] or differential privacy [21]), cryptography-based methods (e.g., encrypting genetic data [22]), or distributed computing (e.g., private records residing at local institutions only [6, 13]). Increasingly, such protections prove to be insufficient, due to various privacy attacks [12, 23–26] leveraging numerous types of side channels (mostly aggregate information or summary statistics), such as allele frequencies from published GWAS studies and public reference genotypes of humans, correlation quantification between genetic variants in the form of linkage disequilibrium (LD), regression coefficients or effect size estimates, p-values, variance-covariance, etc.

Our work here studies the data privacy issues in regularized logistic regression [2]. Regularized logistic regression is widely used in various domains, and is often the preferred model of choice over standard logistic regression in practice [2, 4, 27, 28]. Despite its popularity, it has received little investigation from a data privacy and security perspective. Our work intends to bridge the gap.

Here, we focus on use scenarios where multiple disparate institutions hope to collaboratively perform joint regression analysis (ideally on their consolidated data collection). However, they do not want to disclose their respective data (either individual-level or aggregate information) to others due to privacy and/or confidentiality concerns. Such scenarios are ubiquitous in large collaborations in healthcare, genetics, epidemiology, finance, network analysis and so on (as we will elaborate later). Throughout our work, we assume the widely accepted *honest-but-curious* adversary model [29], meaning that the adversaries would perform computations as exactly specified, but may passively listen to and infer knowledge from information passed between entities in the system. Specific to our focused scenario, the adversaries may be a dishonest analysis/computation center (e.g., maybe due to ill-intentioned employees or breached servers), or curious business competitors in the collaborative study.

In this work, we show how to perform regularized logistic regression while preserving data privacy. To do so, we adapt an efficient optimization method based on distributed computing [6]. The method partitions and distributes sensitive computations such that no (private) raw individual data need to be shared beyond their owner institutions. This leads to better privacy protection on raw data and orders-of-magnitude efficiency gains over a straightforward centralized implementation. In addition, we propose highly secure and flexible protocols to protect intermediate data and computations from model fitting of regularized regression. These altogether lead to an efficient framework for safeguarding regularized logistic regression which provides comprehensive privacy protection over raw as well as intermediate data.

In summary, we consider our contributions to be three-fold:

- Firstly, we demonstrate that regularized logistic regression can be supported efficiently without violating privacy. As mentioned earlier, regularized logistic regression is widely used in practice and enjoys continued investigation from a methodological and computational perspective, yet very few efforts have been devoted to address its related privacy issues. Our work is the first to address such an important issue.

- Secondly, we present a secure and efficient method tailored for regularized logistic regression. We adapt an emerging method of distributed Newton-Raphson [6] for our problem of focus, enhance and extend its privacy protection leveraging strong cryptographic techniques [30]. Our resulting framework not only safeguards regularized logistic regression in particular, but is also relevant to the broader community of privacy-preserving regression analysis where intermediate data do not often receive sufficient protection.

- Lastly, we validate our privacy-enhanced regularized logistic regression extensively with both synthetic and real-world studies. We also demonstrate its scalability to large-scale collaborative studies, and illustrate its practical relevance to various applications from different disciplines.

Materials and Methods

We first provide a brief background introduction to regularized logistic regression and the model estimation process. Later, we will demonstrate how this process can be adapted to preserve data privacy while minimizing the computational overhead.

Preliminaries

(Regularized) Logistic Regression.   Logistic regression [1] is a probabilistic model for predicting binary or categorical outcomes through a logistic function. It is widely used in many domains such as biomedicine [4, 5, 31], social sciences [7, 8], information technology [9, 10], and so on. Briefly, logistic regression is of the form:

$$p(y = 1|\mathbf{x}; \boldsymbol{\beta}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \mathbf{x}}} \ , \tag{1}$$

where $p(.)$ denotes the probability of the response y equal to 1 (i.e., "case" or "success" depending on the scenario), $\mathbf{x}$ is the d-dimensional covariates (or features) for a specific data record, and $\boldsymbol{\beta}$ is the regression coefficients we want to estimate.

Regularized logistic regression [2, 32] shares the same model as illustrated above. However, it differs in the model estimation process (with additional regularization terms applied to the optimization objective), which leads to some desirable properties such as better model generalizability, support for feature selection, etc. As a result, regularized logistic regression is often the preferred choice for many real applications in practice [4, 33, 34].

In this work, we focus on regularized logistic regression with the ℓ_2 norm [2], i.e., with the regularization term equal to $\frac{\lambda}{2}||\boldsymbol{\beta}||_2^2$, where λ is the regularization parameter and $\boldsymbol{\beta}$ is the regression coefficients (note that incorporating other regularizations such as the ℓ_1 norm [32] is also possible).

Newton-Raphson Method.   A common way to estimate the (regularized) logistic regression model (i.e., to obtain $\boldsymbol{\beta}$ coefficients in Eq 1) is through the Newton-Raphson, or iteratively reweighted least squares (IRLS) method [35, 36]. The repeated Newton-Raphson method

adopts an iterative refinement process that eventually converges to the "true" values of the $\boldsymbol{\beta}$ coefficients.

To illustrate the process, we use $\boldsymbol{\beta}^{old}$ and $\boldsymbol{\beta}^{new}$ to denote the $\boldsymbol{\beta}$ coefficient estimates for the current and next iterations, respectively. Each step of the Newton-Raphson method can be expressed as:

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} - \mathbf{H}^{-1}(\boldsymbol{\beta}^{old})\, \mathbf{g}(\boldsymbol{\beta}^{old})\ , \tag{2}$$

where $\mathbf{H}(\boldsymbol{\beta}^{old})$ and $\mathbf{g}(\boldsymbol{\beta}^{old})$ denote the Hessian matrix and gradient of the objective function evaluated at the current estimate of the $\boldsymbol{\beta}$ coefficients. Details of computing $\mathbf{H}(.)$ and $\mathbf{g}(.)$ will be introduced later.

Our Proposal

Here, we introduce our privacy-preserving approach for supporting ℓ_2-regularized logistic regression, based on an adapted Newton-Raphson method. Our proposal was driven by two goals: strong privacy protection and efficient computation. In below, we first provide a high-level overview of our framework; then we introduce the mathematical derivation underlying the method; later, we describe the detailed computations occurring at each stage of the framework and explain how data privacy is preserved thoroughly.

Hybrid Architecture. Our privacy-preserving method for performing ℓ_2-regularized logistic regression features a hybrid architecture combining distributed (local) computing and centralized (secure) aggregation (Fig 1). It is motivated by the observation that certain computations of model estimation could be decomposed per institution, resulting in local-institution computations and center-level aggregation. The careful partitioning and distributing of computations significantly accelerate the process compared with naïve centralized secure
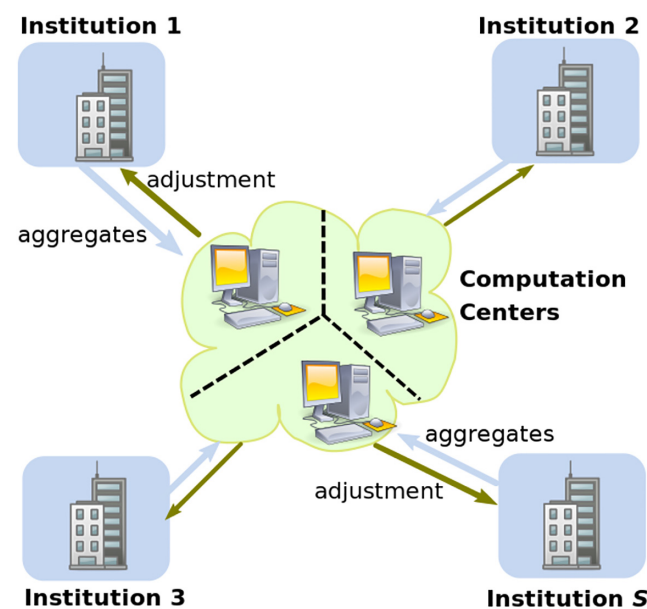

Fig 1. Overview of our secure framework for regularized logistic regression. Each institution (possessing private data) locally computes summary statistics from its own data, and submits encrypted aggregates following a strong cryptographic scheme [30]. The Computation Centers securely aggregate the encryptions and conduct model estimation, from which the model adjustment feedback will be sent back as necessary. This iterative process continues until model convergence.

doi:10.1371/journal.pone.0156479.g001

implementations of Newton-Raphson method, while still guaranteeing the same level of, if not stronger, privacy. Similar strategies of distributed computing have been explored in earlier works [6, 37] for other analytical tasks and prove successful in practice.

Without delving into technical details, we first introduce our framework as illustrated in Fig 1. The framework (and the underlying iterative procedure) consists of two classes of computations: i) the *distributed phase* for computing institution-specific summary statistics locally at individual institutions, and ii) the *centralized phase* for securely aggregating and updating regression coefficient estimates. For each iteration, individual institutions independently compute their local summary statistics (i.e., denoted as *aggregates* in Fig 1. These can be local gradient and Hessian matrix as introduced later) based on their own data, respectively. Such aggregates are then encrypted (via Shamir's secret-sharing [30] which will be explained later) and securely submitted to the Computation Centers (typically multiple independent Centers are designated to collectively hold the data for maximum security). The Computation Centers then collaborate to perform a series of secure data aggregation on the encrypted data, and perform the Newton-Raphson updating (Eq 2) to obtain a globally consistent $\boldsymbol{\beta}$. In addition, model convergence checks will also be securely performed. The new $\boldsymbol{\beta}$ (i.e., denoted as *adjustment* in Fig 1) will then be redistributed to local instituions for the next iteration. The above process of distributed and centralized computing will proceed in iterations until model convergence criteria is satisfied.

Newton-Raphson Method for ℓ_2-regularized Logistic Regression. Our framework (Fig 1) leverages an adapted Newton-Raphson method for model estimation. Here we first demonstrate how the aforementioned Newton-Raphson method applies to ℓ_2-regularized logistic regression. Then we identify the limitations of naïvely applying the method, which motivate us to derive a more efficient approach based on a hybrid architecture.

First, we reformulate the Newton-Raphson method (Eq 2) by defining a diagonal matrix W as $w_{ii} = p_i(1 - p_i)$, $\forall i = 1..N$, where p_i corresponds to the probability estimate for the ith data record (i.e., a row) and N denotes the total number of records. By expanding $\mathbf{H}(.)$ and $\mathbf{g}(.)$ for ℓ_2-regularized logistic regression, Eq 2 becomes:

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} + (\mathbf{X}\mathbf{W}\mathbf{X}^T + \lambda\mathbf{I})^{-1}(\sum_{i=1}^{N}(1 - p_i)\, y_i\mathbf{x}_i - \lambda\boldsymbol{\beta}^{old})\,, \tag{3}$$

where $\mathbf{X}$ corresponds to the design matrix (i.e., covariates) of dimension $N \times d$, λ is the regularization parameter for the ℓ_2-norm (defined *a priori* or derived via cross-validation), and $\mathbf{I}$ denotes the identity matrix.

Traditionally, the aforementioned model estimation method (Eq 3) proceeds in a centralized fashion. This indicates that all individual-level raw data are consolidated into one large (centralized) collection, on which the Hessian matrix and gradient are computed and the Newton-Raphson updating applied. Similar approaches are commonly pursued by the privacy-preserving data mining community (e.g., [38]).

We point out that such a centralized approach could suffer from severe computational inefficiency especially for large studies with privacy protection requirement. In particular, pooling raw data often results in datasets of large scale, on which secure computations can be prohibitively slow (if not infeasible) due to the complexity of supporting matrix operations in secure. Consequently, many alternative privacy-preserving proposals (e.g., [38]) do not seem practical especially for large studies. Such limitations have been illustrated in subsequent studies even on much simpler analytical tasks [39].

Distributed Model Estimation. Observing the inefficiency of the centralized Newton-Raphson method, we intend to accelerate the process by carefully partitioning the

computations to extract "safe" procedures that can be performed more efficiently without violating privacy. Such a solution leads to two anticipated benefits: First, the majority of computations could be supported without relying on expensive secure computation techniques; Second, careful partitioning of computations guarantees the same level of privacy as centralized secure alternatives. We point out it is increasingly the trend to leverage distributed computing for faster computation in privacy-preserving frameworks [39]. The partitioning of Newton-Raphson method has proven successful on other simpler tasks [6] than ours.

To accelerate the Newton-Raphson method (Eq 3), we observe that the computations of $\mathbf{H}(.)$ and $\mathbf{g}(.)$ in Eq 2 can be decomposed, such that some sub-procedures can be performed locally at each institutions on their own respective data where privacy is not of concern. More formally, the per-institution decomposition of computations can be expressed as:

$$\mathbf{H}(\boldsymbol{\beta}) = -\sum_{i=1}^{N} w_{ii}(t)\mathbf{x}_i\mathbf{x}_i^T - \lambda\boldsymbol{\beta} = -\sum_{j=1}^{S} \overbrace{\sum_{i=1}^{N_j} w_{ii}(t)\mathbf{x}_i\mathbf{x}_i^T}^{\text{Per-institution } \mathbf{H}_j(\boldsymbol{\beta})} \underbrace{\phantom{\sum_{j=1}^{S}\sum_{i=1}^{N_j}} -\lambda\boldsymbol{\beta}}_{\text{All institutions}} \qquad (4)$$

and

$$\mathbf{g}(\boldsymbol{\beta}) = \sum_{i=1}^{N} (1-p_i)\, y_i\mathbf{x}_i - \lambda\boldsymbol{\beta} = \sum_{j=1}^{S} \overbrace{\sum_{i=1}^{N_j} (1-p_i)\, y_i\mathbf{x}_i}^{\text{Per-institution } \mathbf{g}_j(\boldsymbol{\beta})} \underbrace{\phantom{\sum_{j=1}^{S}\sum_{i=1}^{N_j}} -\lambda\boldsymbol{\beta}}_{\text{All institutions}} \qquad (5)$$

where S denotes the total number of participating (distributed) institutions and N_j denotes the total number of data records for Institution j—it is easy to see that $N = \sum_{j=1}^{S} N_j$.

According to this decomposition, each institution can individually compute their local $\mathbf{H}_j(.)$ and $\mathbf{g}_j(.)$ on their respective data collections following their traditional practice. Later, the global Computation Centers only need to securely aggregate these (protected) intermediate results to derive the globally consistent $\mathbf{H}(.)$ and $\mathbf{g}(.)$, which would facilitate the Newton-Raphson algorithm.

In addition, the deviance test (for checking model convergence) [1] can also be decomposed similarly, since it depends on the log-likelihood which can be regarded as a series of sums.

$$Dev = -2\log L(\boldsymbol{\beta}) = -2\sum_{j=1}^{S} \overbrace{\sum_{i=1}^{N_j} (y_i \log p_i + (1-y_i)\log(1-p_i))}^{\text{Per-institution } dev_j} \underbrace{\phantom{\sum_{j=1}^{S}\sum_{i=1}^{N_j}}}_{\text{All institutions}}\,, \qquad (6)$$

where $L(\boldsymbol{\beta})$ corresponds to the likelihood.

Based on the above intuition, we introduce a hybrid architecture for supporting ℓ_2-regularized logistic regression (Algorithms 1, 2, and 3). The framework features an iterative process composed of two types of computations: distributed (local) computation (Algorithm 2) and centralized aggregation (Algorithm 3). In the following sections, we will describe these computations in greater detail.

Distributed Computation. The goal of the distributed computation phase (Algorithm 2) is for local institutions to pre-compute their respective summary statistics. During this phase, each participating institutions compute their local Hessian matrix $\mathbf{H}_j$ and gradient $\mathbf{g}_j$ (Eqs 4

and [5]) using their own data. Local deviance test dev_j can also be computed similarly ([Eq 6]). Since each institution has complete ownership over their respective data and no data sharing is involved, such local computations naturally preserve privacy without requiring computationally-expensive cryptographic protections.

Next, all intermediate summary statistics (e.g., $\mathbf{H}_j$, $\mathbf{g}_j$, dev_j) need to be synthesized and processed at the center-level to obtain a globally fitting coefficient estimate (Algorithm 3). To prevent potential privacy inference attacks on aggregate information (partially summarized in [13, 24, 25]), we require each institution to obfuscate their local summaries prior to data submission (Steps 5–6 in Algorithm 2) leveraging a strong protection mechanism known as Shamir's secret-sharing [30] (also introduced later). This mechanism ensures that all intermediate summary statistics (the "secrets") are split into multiple shares to be collectively held by many participants (e.g., one participant would possess only one piece of the secret). The actual content of the "secrets" can only be recovered if the majority of share-holding participants cooperate to decrypt. This way, even if there is collusion between a (minority) few of the secret-share holders, the system is still secure. For our use case, we designate many independent Computation Centers to be share holders.

Centralized Aggregation. Once the distributed computation is completed, the subsequent phase of centralized computation (Algorithm 1) would follow. As the first step, the Computation Centers will aggregate the respective (secret-share-protected) data submissions in a secure way. This process requires collaboration between the Centers who hold the "secrets". Once the globally adjusted $\mathbf{H}(.)$ and $\mathbf{g}(.)$ are derived, the Computation Centers will perform the Newton-Raphson updating on the $\boldsymbol{\beta}^{old}$ estimate and check for model convergence afterwards. If the model is still not converged, then the updated $\boldsymbol{\beta}^{new}$ estimate will be redistributed to local institutions to initiate the next iteration of running.

Algorithm 1 Privacy-preserving regularized logistic regression.
Input: Regression coefficient (of previous iteration) $\boldsymbol{\beta}^{old}$; Penalty parameter λ
Output: New regression coefficients $\boldsymbol{\beta}^{new}$
  1: **while** model not converged **do**
  2:    Compute summary statistics on local institutions: SecureLocal($\boldsymbol{\beta}^{old}$)
  3:    Securely aggregate on Computation Centers: $\boldsymbol{\beta}^{new}$ = SecureCenter($\boldsymbol{\beta}^{old}$, λ)
  4:    Check for model convergence
  5:    $\boldsymbol{\beta}^{old} = \boldsymbol{\beta}^{new}$
  6: **end while**
  7: Return coefficient $\boldsymbol{\beta}^{new}$

Algorithm 2 SecureLocal($\boldsymbol{\beta}^{old}$): securely compute summary statistics on local institutions.
Input: Regression coefficient (of previous iteration) $\boldsymbol{\beta}^{old}$
Output: Shamir's secret shares of $\mathbf{H}_j$, $\mathbf{g}_j$, dev_j ($\forall j \in$ institutions S)
  1: **for** Institution j = 1 **to** S **do**
  2:    Compute local Hessian matrix $\mathbf{H}_j$
  3:    Compute local gradient $\mathbf{g}_j$
  4:    Compute local deviance dev_j
  5:    Protect $\mathbf{H}_j$, $\mathbf{g}_j$, dev_j via Shamir's secret-sharing
  6:    Securely submit $\mathbf{H}_j$, $\mathbf{g}_j$, dev_j secret shares to many (independent) Computation Centers respectively
  7: **end for**

Algorithm 3 SecureCenter($\boldsymbol{\beta}^{old}$, λ): securely aggregate on Computation Centers.
Input: Secret shares of $\mathbf{H}_j$, $\mathbf{g}_j$, dev_j ($\forall j \in$ institutions S); Coefficient $\boldsymbol{\beta}^{old}$; Penalty parameter λ
Output: Updated regression coefficient $\boldsymbol{\beta}^{new}$

```
1: Securely aggregate Hessian: H = −∑_{j=1}^{s} H_j − λβ^{old}
2: Securely aggregate gradient: g = ∑_{j=1}^{s} g_j − λβ^{old}
3: Securely aggregate deviance: Dev = ∑_{j=1}^{s} dev_j
4: Securely compute β^{new} via Newton–Raphson method
5: Return coefficient β^{new}
```

Protecting Privacy

The presented framework involves various types of data and computations, many of which are sensitive or quasi-sensitive. In this section, we analyze how privacy are preserved at each level.

Privacy on Individual Data. The hybrid architecture is designed in such a way that individual raw data are fully and solely controlled by their owner institution, and no sharing of individual-level data is involved in any subsequent computations. This means that no adversarial institutions or Computation Centers would be capable of peaking into individual participants' data. As a result, individual-level privacy is maintained. We note that decoupling from raw individual data for privacy protection is a proven and increasingly popular approach in methodological development in genetics and related fields [6, 13].

Privacy on Aggregate Data. We observe that various inference attacks on privacy are only possible because of the disclosure of summary statistics. For instance, the genome-disease inference attack in [23] relies on certain genomic summaries of case/control groups; it has also been analyzed in [24–26] regarding the risks associated with disclosing summary statistics, such as covariance matrix, information matrix and score vector. Meanwhile, we note that aggregate data may also constitute confidential or proprietary information for some institutions and thus should be protected (a similar opinion was briefly communicated in [37]). This is not uncommon for joint studies in competitive scenarios, such as financial collaborations, healthcare quality comparisons, and association studies involving sensitive and rare diseases.

Specific to our task of regularized logistic regression (and logistic regression in general), the vulnerable summaries are the hessian and gradient, which collectively could result in inference attacks on private response variables and model recovery [13, 24, 25].

To prevent potential attacks or confidentiality breaches, our framework encrypts summary statistics from participating institutions (prior to data submission to Computation Centers) leveraging a strong Cryptographic mechanism known as Shamir's secret-sharing [30](to be introduced in the following section). Due to encryption, neither the potentially adversarial institutions nor Computer Centers could access aggregate information, which is the prerequisite to any aforementioned attacks. The idea of protecting intermediate data has been explored before [26, 37, 39], however, mostly only on simpler tasks (e.g., ridge linear regression, standard logistic regression, etc) than ours; in a more related work [37], summaries from distributed Newton method have been obfuscated with simple tricks, however, the protection is insufficient and easily vulnerable to collusion attacks as we will discuss later.

Shamir's Secret-Sharing for Protecting Data. In our protocol, we leverage Shamir's secret-sharing [30] to protect intermediate data (including summary statistics from institutions). The general idea underlying Shamir's secret sharing is that for a t-dimensional Cartesian plane, at least t independent coordinate pairs are necessary to uniquely determine a polynomial curve. Formally, a t-out-of-w secret-share scheme is defined as follows: we intend to protect a secret m (e.g., certain institution-specific summary statistic in our case) such that the only way to successfully recover the secret is through cooperation of at least t (i.e., the "threshold") share-holding participants (out of a total of w). To achieve the goal, we construct a random polynomial $q(x)$ of degree $(t − 1)$ with the secret m embedded (we point out that the calculations actually occur in a finite integer field. However, for presentation simplicity, we skip the

technical details):

$$q(x) = m + \sum_{i=1}^{t-1} a_i x^i \,, \tag{7}$$

where m is the secret we want to protect, and a_i's are randomly generated polynomial coefficients. Note that the polynomial itself will be kept secret.

In order to split and "share" the secret, we proceed to evaluate $q(x)$ and derive t or more distinct values from the polynomial, yielding coordinate pairs $(1, q(1)), (2, q(2)), \ldots, (t, q(t)), \ldots, (w, q(w))$. Due to the inherent randomness in the specified polynomial, the coordinate pairs we obtain here are random and reveal nothing meaningful about the secret. These pairs, each of which constitutes a share of the secret, are then distributed to t or more Computation Centers, respectively (i.e., each participant only receives one piece of the secret). Under this mechanism, we can claim that the secret is successfully protected, since no single Center or a limited few are capable of inferring anything about the polynomial or the embedded secret. When it is necessary to recover the original secret, t or more share holders will collectively perform Lagrange polynomial interpolation [30] to uniquely determine the polynomial $q(x)$. The secret will naturally emerge by evaluating $q(0)$: $m = q(0)$. To facilitate complex data and computations in our framework, we have extended the scheme to support matrices and vectors.

Privacy on Computations.   Since all data in our framework are in encrypted form, special care must be taken to support analytical procedures. Here we introduce several secure primitives for supporting necessary computations without decrypting intermediate data. We focus on secure addition and secure multiplication by a public value, which are necessary for our task under question.

Secure addition is a fundamental building block for the central aggregation phase (Algorithm 3). Briefly, the primitive helps securely derive the sum $A+B$ without knowing the actual content of A and B, since both of which are encrypted via Shamir's secret-sharing. As illustrated in Algorithm 4, the general idea of the secure addition primitive is to ask each share holders to locally aggregate original shares of the two secret addends in order to derive new shares, which will serve as the shares for their sum.

Algorithm 4 Secure addition (aggregation).
Input: Secret-shared data A and B (among w institutions)
Output: Sum $sum = A+B$ in secret-shared form (among w institutions)
  1: **for** institution j := 1 **to** w **do**
  2:     [At Institution j]
  3:     Compute and store new share: $sum_j = A_j+B_j$
  4: **end for**

To show that the secure addition primitive is correct, we assume the (secret-sharing) polynomials to be $q_A(x)$, $q_B(x)$, respectively, for the two secrets A, B. In other words: $A = q_A(0)$, $B = q_B(0)$. Since both polynomials share the same covariates and degrees, we have: $q_A(0)+q_B(0) = (q_A+q_B)(0)$. This indicates that, the aggregated coordinate pairs satisfy the newly defined polynomial $(q_A+q_B)(.)$ and thus represent the new shares of the to-be-computed sum $A+B$.

Next, we show how secure multiplication-by-a-constant can be implemented, which is required by the Newton-Raphson method. In particular, we consider multiplying a secret value (in secret-shared form) by a known constant value. The primitive is surprisingly simple: share holders only need to locally multiply their shares (of the secret value) by the public constant to derive the new shares for the product of the two values. The proof for this method is

straightforward, since multiplication by a constant can be reformulated as a series of secure additions by the secret value itself.

Note that in our current implementation, we take a pragmatic approach to security for better computational efficiency without degrading privacy. Specifically, the primary reason why protecting intermediate data is necessary in regularized logistic regression is due to privacy inference attacks [13, 24, 25]. Existing attacks rely on both hessian and gradient to be feasible. Our protection thus only needs to protect one of the summaries to prevent such attacks. This can lead to significant speedup as compared to an "encrypting-all" strategy and our privacy protection goal is still achieved. Extending our current implementation to a fully encrypted setting is also straightforward, as the additional secure primitives (e.g., secure matrix inversion) have already been demonstrated before [39].

Secure matrix inversion can be useful if we want to fully secure intermediate computations (e.g., inverting the Hessian matrix). Several existing secure solutions [39–41] serve as our reference that leverage methods such as LU-decomposition, Gaussian elimination, etc. Due to the focus of this work, we leave it as future extension.

Since none of our aforementioned primitives change the original Shamir's scheme, the information-theoretical security still holds in our protocol. Interested readers are kindly referred to relevant literature [42] for a detailed security proof.

Generating synthetic data.   To allow for comprehensive evaluation on our framework, we also generate synthetic datasets (in addition to other real datasets as introduced later) according to Algorithm 5. We first generate coefficients and covariates at random (according to uniform and Gaussian distributions, respectively). Later, based on the calculated probabilities, we generate the response variables from the Bernoulli distribution. The resulting synthetic dataset is partitioned per institution.

Algorithm 5 Generate synthetic data
Input: Covariate dimensionality d
Output: Covariates $\mathbf{X}$, responses $\mathbf{y}$ (both partitioned per-institution)
  1: Generate coefficients $\boldsymbol{\beta} \in \mathbb{R}^d$ at random
  2: **for** institution $j := 1$ **to** S **do**
  3:   Generate covariates $\mathbf{cov}_j \in \mathbb{R}^{N_j \times (d-1)}$ from $\mathcal{N}(\mu, \sigma^2)$
  4:   Output concatenated covariates $\mathbf{X}_j = \begin{bmatrix} 1 & \mathbf{cov}_j \end{bmatrix} \in \mathbb{R}^{N_j \times d}$
  5:   Calculate probabilities $\mathbf{p}_j = 1/(1 + e^{-\boldsymbol{\beta}^T \mathbf{X}_j}) \in \mathbb{R}^{N_j}$
  6:   Generate and output response variables $\mathbf{y}_j \in \mathbb{R}^{N_j}$ from $Bernoulli(\mathbf{p}_j)$
  7: **end for**

Results

We have implemented our privacy-preserving framework for ℓ_2-regularized logistic regression. To validate our proposal, we perform extensive empirical evaluation on both synthetic and real-world studies. We report on the evaluations in terms of result accuracy, computational efficiency, as well as scalability to large studies.

Evaluation Datasets

Included in our empirical evaluation are four studies, which represent a wide spectrum of applications from different domains and data scales. In specific,

- The **Synthetic dataset** is a large-scale dataset we generated at random according to Algorithm 5. It consists of 1 million records spanning 6 features from 6 institutions, which is quite representative for most real-world use cases.

- The **Insurance dataset** [43] is a dataset from an insurance company with the goal of predicting users' insurance policy status based on socio-demographic features. It contains 9,822 records and 84 features, and we simulated 5 institutions by randomly partitioning the dataset horizontally.

- The **Parkinsons.Motor and Parkinsons.Total datasets** both relate to one dataset targeted for predicting parkinson's tele-monitoring quantities, with 5,875 samples spanning 20 features [44]. Since there are two distinct target predictions in the original dataset, we partition the dataset into two sub-studies which we denote as **Parkinsons.Motor** (for predicting motor UPDRS) and **Parkinsons.Total** (for total UPDRS). They share the same covariates but with different response variables. We randomly partitioned the records among 5 institutions.

Regression Result Accuracy

The first question we consider in validating our framework is whether the regression result is accurate and reliable. To answer this question, we compare our estimated regression coefficients with that obtained from standard software packages. As illustrated in Fig 2, our framework yields identical results to the expected ground truth across all evaluations (with correlation $R^2 = 1.00$). The result accuracy is also evidenced by the mathematical proof explained earlier, where we have shown that our distributed model estimation method follows an exact derivation and no approximation is involved in the secure computation procedures.

Running Time

We implemented the prototype in R and Scala, a Java Virtual Machine-based programming language. Experiments were performed on a quad-core computer with 2.4GHz CPU and 8GB memory, running Ubuntu 13.04. To eliminate network latency effects, we simulated distributed computing nodes on a single computer and report the network data exchanged. We performed each experiment several times and reported the mean of the running time.

Empirical evaluation indicates that our protocol is highly efficient, as demonstrated in Table 1. For datasets with as many as 1 million records, our protocol completed in less than 12 seconds. For datasets of more modest sizes as typically found in everyday applications, our protocol took only around $2 \sim 4$ seconds or less.

Since our framework is focused on a novel analytical application that is not addressed in the privacy/security domain, technically we do not have any alternatives to compare against. We do however, try to provide brief comparisons against similar secure approaches in related problems—mostly from linear (ridge) regression which also considered regularization and adopted a similar hybrid architecture. Our evaluation indicate that our protocol is more efficient than other related secure proposals (even though they focused on much simpler regression models). For instance, as a rough comparison, secure linear regression in [40] on 51,016 samples with 22 covariates took two days. Our framework is also competitive compared with the state-of-the-art secure solution for the ridge (linear) regression [39] (a much simpler model), which took 55 seconds on a smaller-scale Insurance dataset (with only 14 features). We do acknowledgment that such comparisons are not very fair, as our proposal solves a different and more complicated regression model; also some alternatives implemented additional features. Nevertheless, the results demonstrate that our secure framework for regularized logistic regression is efficient and competitive.
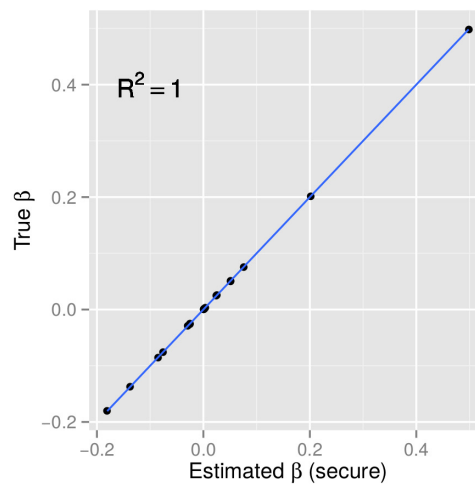
Overall, the repeated Newton-Raphson process converged within a limited number of iterations, as evidenced by Fig 3. Across all evaluation datasets, the models converged within $6 \sim 8$
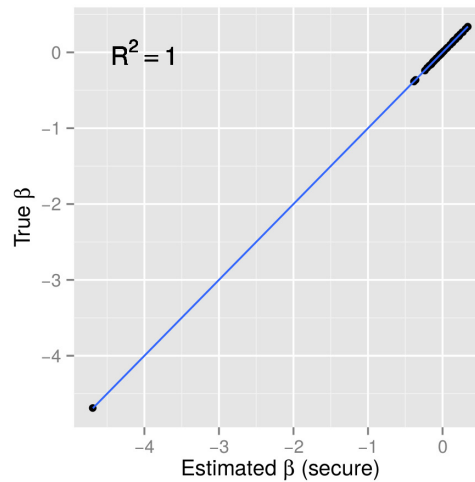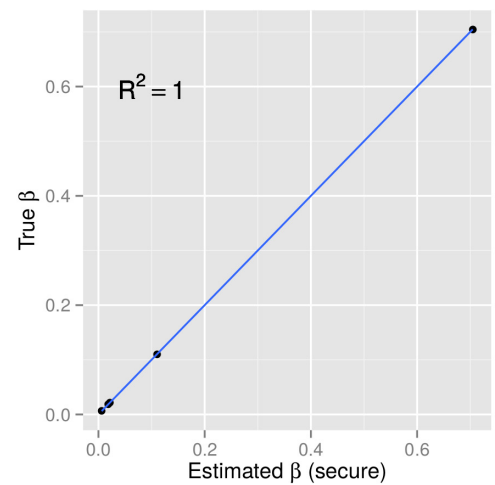
(a) Insurance dataset.
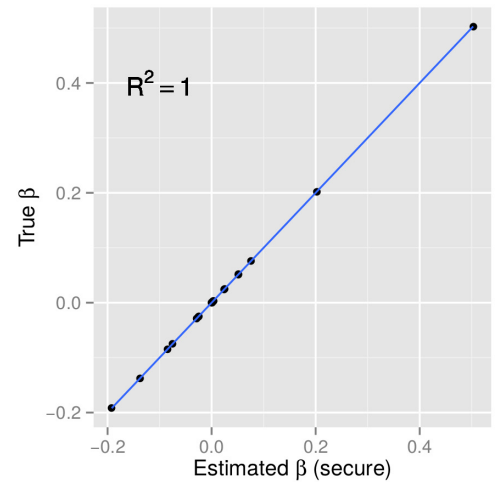
(b) Parkinsons.Motor dataset.

(c) Parkinsons.Total dataset.

(d) Synthetic dataset.

Fig 2. Model accuracy of our securely estimated β against the gold standard for four evaluation datasets. As illustrated, the regression coefficients estimated via our secure framework are identical to the gold standards, with correlation $R^2 = 1.00$.

doi:10.1371/journal.pone.0156479.g002

Table 1. Computational efficiency on evaluation datasets.

Dataset	Insurance	Parkinsons.Motor	Parkinsons.Total	Synthetic
# samples	9,822	5,875	5,875	1,000,000
# features	84	20	20	6
# iterations	8	6	6	6
Central runtime (S)	0.42	0.264	0.236	0.076
Total runtime (S)	3.77	2.017	2.352	12.76
Data transmitted (MB)	80	492	492	612

doi:10.1371/journal.pone.0156479.t001
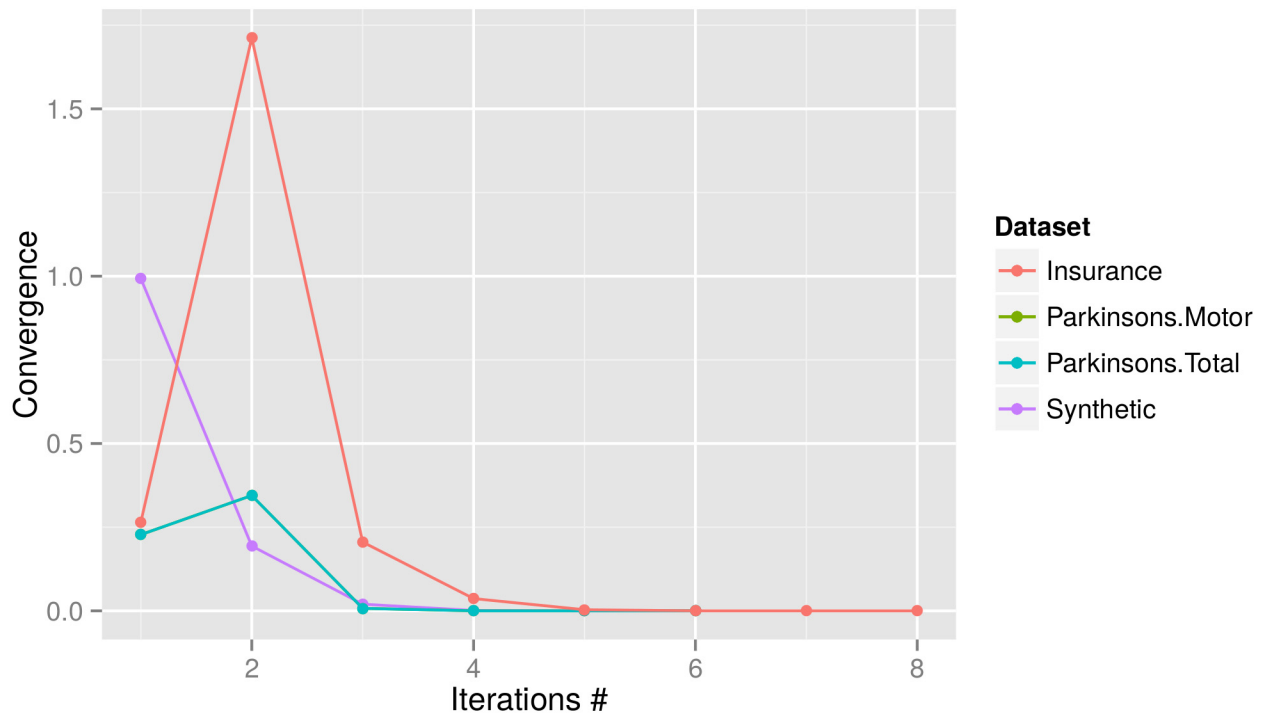
Fig 3. Model convergence (i.e., deviance) for all datasets (deviance smaller than the threshold indicates convergence). All models converged within 6 ∼ 8 iterations. Note that the convergence scores for the Parkinsons.Motor and Parkinsons.Total studies almost overlap due to their high similarity in the plot.

doi:10.1371/journal.pone.0156479.g003

iterations. As common in statistics, we set the convergence criteria to be 10^{-10}. Also, the amount of data to be exchanged during computation is also modest. As an example, for the Synthetic dataset with 1 million records, only around 612 megabytes of data are transmitted over the network.

To further demonstrate the efficiency of our method, we report on the time efficiency of its major procedures (i.e., the central phase and the total runtime) in Table 1. We emphasize that the vast majority of runtime is spent at individual local institutions (on conventional computations), and secure computation at the Computation Centers only consumes around 11.14%, 13.09%, 10.03%, and 0.60% of the total time for the datasets evaluated, respectively.

Scalability to Large Studies

With the advent of the big data era, large-scale collaborative studies are becoming ubiquitous in many domains. A few notable examples include the International Cancer Genome Consortium [14], the Patient-Centered Outcomes Research Institute (PCORI) [45], and financial systematic risk protection [46].

To meet the demand of large-scale cross-institution studies, we also demonstrate the scalability of our framework. Since regression accuracy is not affected by the increase of participating institutions, we mainly focus on evaluating the running time. To do so, we simulated studies with up to 100 institutions, and reported the results in Fig 4 (we simplified the scenario by assuming that each institution contributes 10000 records. So in fact, our evaluation reflects the running time affected by the increase of both the number of institutions and the total number of data records).
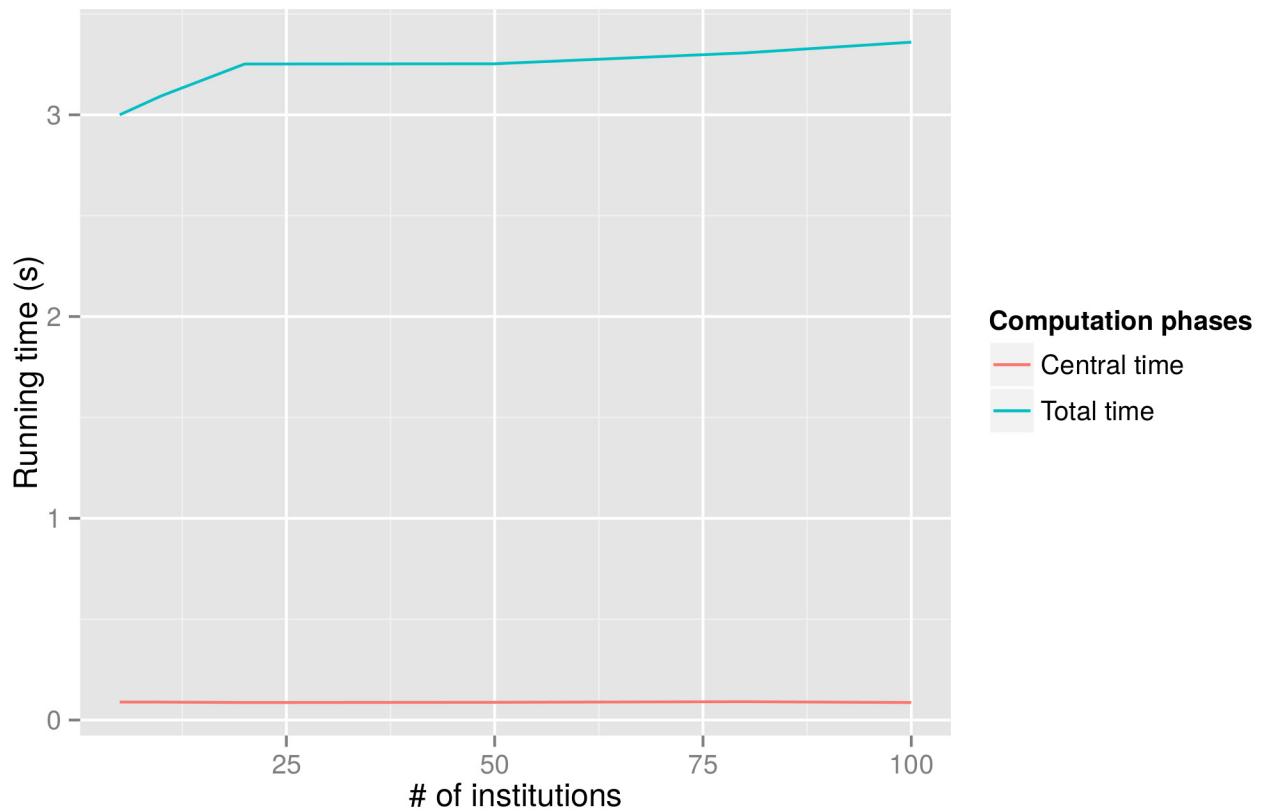
Fig 4. Running time (in seconds) for the central phase and total computation respectively, as the number of participating institutions increases. Negligible time fluctuation is present, especially for the central (secure) computation.

It can be seen that the total time is always between $3.0 \sim 3.3$ seconds, exhibiting minimal fluctuation as the number of participating institutions increases. This is especially the case for the secure-computation-based centralized phase, which consistently takes only around 0.088 seconds.

Such a trend is well explained from a theoretical perspective (as evident in the computation details in Algorithm 1, as individual institutions perform their local (distributed) computations simultaneously without interacting with (or waiting for) other participants. As a result, local computations are relatively stable from the change. The increase of the number of institutions does slightly influence the centralized aggregation of institution-level summary statistics, as more summaries need to be transmitted and aggregated. But the effect is minimal, since the summary data size is relatively small and the majority of computations for aggregating secret shares occur locally at each Computation Center (as explained earlier regarding secure addition and multiplication).

Overall, the evaluation has demonstrated that our secure framework could support large-scale studies with hundreds of institutions and millions of data records.

Discussion

While the prototype implementation has already demonstrated impressive efficiency, we point out that further speed-ups can be obtained for production systems. For instance, local data can be cached in computer memory to greatly streamline and accelerate subsequent iterations of computations; further acceleration can be gained locally by adopting high-performance programming

languages (e.g., C/C++) and libraries (e.g., BLAS/LAPACK [47]); as for the central computation, it can also greatly benefit from multi-core parallelism, since many secure operations can be parallelized naturally. In addition, the cryptography community continues to improve efficiency of secure primitives which could be useful to us in future. In addition to Shamir's secret-sharing we used here [30], there are also several alternative schemes that prove to be useful on many tasks, such as Paillier encryption and Yao's garbled circuit (as used by [13, 26, 39]). Due to space constraint, we intend to explore other potential schemes for related tasks in future.

There have been various alternative proposals for protecting privacy while supporting regression analysis. Most of them only focused on much simpler regression models, such as linear (ridge) regression, or standard logistic regression without regularization. And typically there is no or only weak protection over summary statistics during the computation process. One line of research that is directly relevant to our proposal is cryptography-based approaches. For instance, a privacy-preserving method was proposed for (linear) ridge regression [39], which directly solves the linear system in secure centrally. Other secure solutions [26, 38, 40], for linear or logistic regression relied on some expensive cryotpgraphic primitives and approximations, which add significant computational overhead and do not seem scalable to modest or large sample sizes. Increasingly, distributed-computing-based solutions [6, 13, 37] emerged as promising solutions for linear/logistic regression and related analytics. However, none of these support regularized regression which is a more widely used model in practice. Many related proposals [6] directly expose summary data from model fitting, leading to serious privacy concerns over inference attacks on intermediate data [13, 24–26]. While preliminary efforts have started to gather around protecting institution-level summary information (especially regarding logistic regression), existing protections seem quite weak. For instance, the obfuscation protection in [37] is vulnerable to collusion attacks by the center (who generates the randomization noise) and any of the institutions, causing single points of failure or breach from a security perspective. Another popular research direction in privacy-preserving logistic regression leverages non-cryptographic approaches, such as the classical k-anonymity model [20] or differential privacy [21]. One notable example is the ϵ-differentially private logistic regression [48], which adds artificial noise to the result or perturbs the optimization objective function to make the regression result private. Such methods, however, distort the computation or output, often rendering the result inaccurate and scientifically not useful to domain experts. In addition, such methods do not protect intermediate computations.

Our framework demonstrated here for regularized logistic regression differentiates in several ways. Firstly, we focus on an important and (more) widely-used statistical model that has not been addressed by the data security/privacy community. While there is recent privacy-preserving work [39] specifically targeted for ridge (linear) regression (i.e., with ℓ_2-regularization), it focused on a much simpler regression model (i.e., linear regression) and the model estimation process is completely different from regularized logistic regression (the focus of our work). None of the other related works have considered regularization, despite its wide adoption and popularity in various application domains as well as methodological development in statistics and machine learning. Secondly, for efficient model estimation on regularized logistic regression, we adapted a distributed Newton method that previously has only been validated on simpler analytical models [6]. The distributed process makes our secure protocol for regularized logistic regression highly efficient compared to a straightforward centralized implementation [26]. Thirdly, we protect intermediate data and computations with stronger cryptographic schemes [30], providing strong security guarantees thanks to decentralization of trust while still allowing for efficient and flexible computation. While privacy protection on summary statistics has been explored for other tasks [13], ours is the first to safeguard regularized logistic regression regarding intermediate data. Among the two closely related works, [6] failed to

provide any protection over summaries; And [37] had very weak protection as discussed earlier. Lastly, our model does not involve approximation or artificial perturbation (contrary to solutions based on classical *k*-anonymity [20] or differential privacy [21]) on the data or computations, thus maintaining accuracy of the predictive model.

Application Scenarios

We believe the proposed privacy-preserving framework is applicable to a wide range of domains where the privacy/confidentiality of study participants and/or institutions is of concern. Here we briefly describe a few representative application scenarios.

Genetic and Biomedical Studies. Genetic studies have enjoyed continued investigation efforts with the ultimate goal of uncovering connections between genes and human traits (e.g., diseases). Regularized logistic regression is an increasingly important tool for related applications, including for genomic selection [27, 49], gene-gene interactions [4], GWAS [34], etc. Other biomedical studies such as prediction of adverse drug reactions [50] are also potential application domains.

Many such studies rely on large-scale data sharing across institutions, while at the same time, many such data involve sensitive data such as genome information, or participant phenotypes [13]. We envision that our framework can provide an automated and privacy-preserving solution for supporting such collaborative investigations.

Analytics for Smart Grid. Smart electrical grid is a transformative technology that provides detailed data pertaining to the monitoring and management of energy consumption of individual households. Data sharing and analytics on such data have raised serious privacy concerns from both everyday consumers and governmental regulators [51] due to various privacy inference attacks on energy monitoring data. We believe that our distributed-computing-based technology can support some useful analytics on smart grid data, such that household privacy could be maintained.

Large-scale Network Analysis. Many important innovations involve analysis of social network data, such as [8, 52, 53]. These include anomaly detection, novel discoveries in online social networks (such as personalization and link prediction), etc. Social networks data often involve person-level private information, making them inappropriate to share across institutions in large collaborative studies. Our framework could serve the purpose by allowing for joint network analysis without disclosing private information.

Conclusion

In this work, we propose new cryptographic methods for preserving privacy in regularized logistic regression, a widely-used statistical model in various domains. To make the model efficient in a secure setting, we adapted a distributed method for model estimation. To further enhance privacy and prevent inference attacks over intermediate data during model estimation, we introduced strong cryptographic protections. These lead to an efficient framework for supporting regularized logistic regression across different institutions while guaranteeing strong privacy both for individual study participants and institutions. Extensive empirical evaluations have demonstrated the efficacy of the framework in guaranteeing privacy with modest computational overhead. We hope that careful implementation of our framework could enable a wider range of cross-institution joint analytics, which would otherwise be impossible due to privacy or confidentiality concerns.

Acknowledgments

Author Contributions

Conceived and designed the experiments: WL WX. Performed the experiments: WL WX. Analyzed the data: WL HL PY WX. Contributed reagents/materials/analysis tools: WL HL PY WX. Wrote the paper: WL HL PY WX.

References

1. Wasserman L. All of statistics. Springer Science & Business Media; 2011.

2. Le Cessie S, Van Houwelingen JC. Ridge estimators in logistic regression. Applied statistics. 1992;p. 191–201. doi: 10.2307/2347628

3. Visscher PM, Brown MA, McCarthy MI, Yang J. Five years of GWAS discovery. The American Journal of Human Genetics. 2012; 90(1):7–24. doi: 10.1016/j.ajhg.2011.11.029 PMID: 22243964

4. Park MY, Hastie T. Penalized logistic regression for detecting gene interactions. Biostatistics. 2008; 9 (1):30–50. doi: 10.1093/biostatistics/kxm010 PMID: 17429103

5. Dasgupta A, Sun YV, König IR, Bailey-Wilson JE, Malley JD. Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience. Genetic epidemiology. 2011; 35(S1):S5–S11. doi: 10.1002/gepi.20642 PMID: 22128059

6. Wolfson M, Wallace SE, Masca N, Rowe G, Sheehan NA, Ferretti V, et al. DataSHIELD: resolving a conflict in contemporary bioscience—performing a pooled analysis of individual-level data without sharing the data. International journal of epidemiology. 2010;p. dyq111.

7. Peng CYJ, Lee KL, Ingersoll GM. An introduction to logistic regression analysis and reporting. The Journal of Educational Research. 2002; 96(1):3–14. doi: 10.1080/00220670209598786

8. Kosinski M, Stillwell D, Graepel T. Private traits and attributes are predictable from digital records of human behavior. Proceedings of the National Academy of Sciences. 2013; 110(15):5802–5805. doi: 10.1073/pnas.1218772110

9. Richardson M, Dominowska E, Ragno R. Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on World Wide Web. ACM; 2007. p. 521–530.

10. Dai C, Qian F, Jiang W, Wang Z, Wu Z. A Personalized Recommendation System for NetEase Dating Site. Proceedings of the VLDB Endowment. 2014; 7(13). doi: 10.14778/2733004.2733080

11. Office for Civil Rights H. Standards for privacy of individually identifiable health information. Final rule. Federal Register. 2002; 67(157):53181. PMID: 12180470

12. Erlich Y, Narayanan A. Routes for breaching and protecting genetic privacy. Nature Reviews Genetics. 2014; 15(6):409–421. doi: 10.1038/nrg3723 PMID: 24805122

13. Xie W, Kantarcioglu M, Bush WS, Crawford D, Denny JC, Heatherly R, et al. SecureMA: protecting participant privacy in genetic association meta-analysis. Bioinformatics. 2014; 30(23):3334–3341. doi: 10.1093/bioinformatics/btu561 PMID: 25147357

14. Hudson TJ, Anderson W, Aretz A, Barker AD, Bell C, Bernabé RR, et al. International network of cancer genome projects. Nature. 2010; 464(7291):993–998. doi: 10.1038/nature08987 PMID: 20393554

15. Daries JP, Reich J, Waldo J, Young EM, Whittinghill J, Ho AD, et al. Privacy, anonymity, and big data in the social sciences. Communications of the ACM. 2014; 57(9):56–63. doi: 10.1145/2643132

16. Boyd D. Facebook's Privacy Trainwreck. Convergence: The International Journal of Research into New Media Technologies. 2008; 14(1):13–20.

17. Federal Trade Commission. A Preliminary FTC Staff Report on Protecting Consumer Privacy in an Era of Rapid Change: A Proposed Framework for Businesses and Policymakers;. Accessed: 2015-01-10. http://www.ftc.gov/sites/default/files/documents/reports/federal-trade-commission-bureau-consumer-protection-preliminary-ftc-staff-report-protecting-consumer/101201privacyreport.pdf

18. Home Depot's 56 Million Card Breach Bigger Than Target's.;. Accessed: 2015-02-25. http://www.wsj.com/articles/home-depot-breach-bigger-than-targets-1411073571

19. Massive breach at health care company Anthem Inc.;. Accessed: 2015-02-25. http://www.usatoday.com/story/tech/2015/02/04/health-care-anthem-hacked/22900925/

20. Sweeney L. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2002; 10(05):557–570.

21. Dwork C. Differential privacy. In: Encyclopedia of Cryptography and Security. Springer; 2011. p. 338–340.

22. Kantarcioglu M, Jiang W, Liu Y, Malin B. A cryptographic approach to securely share and query genomic sequences. Information Technology in Biomedicine, IEEE Transactions on. 2008; 12(5):606–617. doi: 10.1109/TITB.2007.908465

23. Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS genetics. 2008; 4(8):e1000167. doi: 10.1371/journal.pgen.1000167 PMID: 18769715

24. Sparks R, Carter C, Donnelly JB, O'Keefe CM, Duncan J, Keighley T, et al. Remote access methods for exploratory data analysis and statistical modelling: Privacy-Preserving Analytics. Computer methods and programs in biomedicine. 2008; 91(3):208–222. doi: 10.1016/j.cmpb.2008.04.001 PMID: 18495290

25. O'Keefe CM, Chipperfield JO. A summary of attack methods and confidentiality protection measures for fully automated remote analysis systems. International Statistical Review. 2013; 81(3):426–455. doi: 10.1111/insr.12021

26. El Emam K, Samet S, Arbuckle L, Tamblyn R, Earle C, Kantarcioglu M. A secure distributed logistic regression protocol for the detection of rare adverse drug events. Journal of the American Medical Informatics Association. 2013; 20(3):453–461. doi: 10.1136/amiajnl-2011-000735 PMID: 22871397

27. Ayers KL, Cordell HJ. SNP Selection in genome-wide and candidate gene studies via penalized logistic regression. Genetic epidemiology. 2010; 34(8):879–891. doi: 10.1002/gepi.20543 PMID: 21104890

28. Malo N, Libiger O, Schork NJ. Accommodating linkage disequilibrium in genetic-association analyses via ridge regression. The American Journal of Human Genetics. 2008; 82(2):375–385. doi: 10.1016/j.ajhg.2007.10.012 PMID: 18252218

29. Goldreich O. Foundations of cryptography: volume 2, basic applications. Cambridge university press; 2009.

30. Shamir A. How to share a secret. Communications of the ACM. 1979; 22(11):612–613. doi: 10.1145/359168.359176

31. Bush WS, Moore JH. Genome-wide association studies. PLoS computational biology. 2012; 8(12): e1002822. doi: 10.1371/journal.pcbi.1002822 PMID: 23300413

32. Lee SI, Lee H, Abbeel P, Ng AY. Efficient L1 regularized logistic regression. In: Proceedings of the 21st national conference on Artificial intelligence-Volume 1. AAAI Press; 2006. p. 401–408.

33. Zhu J, Hastie T. Classification of gene microarrays by penalized logistic regression. Biostatistics. 2004; 5(3):427–443. doi: 10.1093/biostatistics/kxg046 PMID: 15208204

34. Liu Z, Shen Y, Ott J. Multilocus association mapping using generalized ridge logistic regression. BMC bioinformatics. 2011; 12(1):384. doi: 10.1186/1471-2105-12-384 PMID: 21958005

35. Green PJ. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. Journal of the Royal Statistical Society Series B (Methodological). 1984;p. 149–192.

36. Minka TP. A comparison of numerical optimizers for logistic regression; 2003. http://research.microsoft.com/en-us/um/people/minka/papers/logreg/.

37. Wu Y, Jiang X, Ohno-Machado L. Preserving institutional privacy in distributed binary logistic regression. In: AMIA Annual Symposium Proceedings. vol. 2012. American Medical Informatics Association; 2012. p. 1450.

38. Nardi Y, Fienberg SE, Hall RJ. Achieving both valid and secure logistic regression analysis on aggregated data from different private sources. Journal of Privacy and Confidentiality. 2012; 4(1):9.

39. Nikolaenko V, Weinsberg U, Ioannidis S, Joye M, Boneh D, Taft N. Privacy-preserving ridge regression on hundreds of millions of records. In: Security and Privacy (SP), 2013 IEEE Symposium on. IEEE; 2013. p. 334–348.

40. Hall R, Fienberg SE, Nardi Y. Secure multiple linear regression based on homomorphic encryption. Journal of Official Statistics. 2011; 27(4):669.

41. Du W, Han YS, Chen S. Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. In: SDM. vol. 4. SIAM; 2004. p. 222–233.

42. Beimel A. Secret-sharing schemes: a survey. In: Coding and cryptology. Springer; 2011. p. 11–46.

43. Van Der Putten P. CoIL challenge 2000: The insurance company case. 2000;.

44. Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. Biomedical Engineering, IEEE Transactions on. 2009; 56 (4):1015–1022. doi: 10.1109/TBME.2008.2005954

45. Selby JV, Beal AC, Frank L. The Patient-Centered Outcomes Research Institute (PCORI) national priorities for research and initial research agenda. Jama. 2012; 307(15):1583–1584. doi: 10.1001/jama.2012.500 PMID: 22511682

46. Abbe EA, Khandani AE, Lo AW. Privacy-Preserving Methods for Sharing Financial Risk Exposures. American Economic Review. 2012; 102(3):65–70. doi: 10.1257/aer.102.3.65

47. LAPACK—Linear Algebra PACKage;. Accessed: 2010-09-30. http://www.netlib.org/lapack.

48. Chaudhuri K, Monteleoni C. Privacy-preserving logistic regression. In: Advances in Neural Information Processing Systems; 2009. p. 289–296.

49. Shen X, Alam M, Fikse F, Rönnegård L. A novel generalized ridge regression method for quantitative genetics. Genetics. 2013; 193(4):1255–1268. doi: 10.1534/genetics.112.146720 PMID: 23335338

50. Liu M, Wu Y, Chen Y, Sun J, Zhao Z, Chen Xw, et al. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. Journal of the American Medical Informatics Association. 2012; 19(e1):e28–e35. doi: 10.1136/amiajnl-2011-000699 PMID: 22718037

51. McDaniel P, McLaughlin S. Security and Privacy Challenges in the Smart Grid. Security & Privacy, IEEE. 2009; 7(3):75–77. doi: 10.1109/MSP.2009.76

52. Masuda N, Kurahashi I, Onari H. Suicide ideation of individuals in online social networks. PloS one. 2013; 8(4):e62262. doi: 10.1371/journal.pone.0062262 PMID: 23638019

53. Almquist ZW, Butts CT. Logistic Network Regression for Scalable Analysis of Networks with Joint Edge/Vertex Dynamics. Sociological Methodology. 2014; 44(1):273–321. doi: 10.1177/0081175013520159 PMID: 26120218