PRIVACY LEAKS AND EFFICIENT COUNTERMEASURES FOR HUMAN GENETICS AND MACHINE LEARNING

By

Wei Xie

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

May 11, 2018

Nashville, TN

Approved:

Professor Bradley A. Malin

Professor Douglas Fisher

Professor Aniruddha Gokhale

Professor Todd L. Edwards

Professor Nancy J. Cox

ACKNOWLEDGMENTS

Throughout my PhD, I am very fortunate to have received tremendous help from many people both professionally and personally, which I truly appreciate.

First, I feel indebted to Dr. Bradley Malin for introducing me to the field of data privacy in healthcare and genomics. His wisdom, perfectionism in writing, continued support, and patience have made my PhD much smoother and enjoyable. I would like to sincerely thank my committee for their invaluable guidance and support. In particular, Dr. Todd Edwards has been extremely helpful, knowledgeable, and kindhearted as a collaborator, mentor and friend. I always appreciate and learn from his constructive feedback in pointing me in the right direction and making high impact. I also thank Dr. Nancy Cox for sparing valuable time and serving on my committee and advising me on several research projects. I want to express my sincere gratitude to Dr. Aniruddha Gokhale for being extremely supportive and responsive in providing valuable feedback on my research. His classes are among my favorites from which I continue to benefit. Dr. Douglas Fisher has also been very helpful and inspiring to my research and education.

This dissertation would not have been possible without the help and guidance from my various collaborators and close friends. Throughout my PhD study, Dr. You Chen has been a source of support and inspiration, providing numerous opportunities for me to learn and grow both as a researcher and person. I really appreciate all his help. I also want to sincerely thank Drs. Jihun Hamm, Yang Wang, Steven Boker, Donald Brown, Yongtao Cao, Peilin Jia, Zhongming Zhao, Zhijun Yin, and Cheng Gao for their stimulating discussions and brainstorming, collaborations, and providing support for achieving many ambitious projects.

I thank Dr. Joshua Denny for various inspiring discussions and feedback on research projects and manuscripts, and providing resources support. I also thank friends and collaborators including Wenjie, Wenfa, Drs. William Bush, Dana Crawford, Bingshan Li,

Douglas Ruderfer, Jirong Long, Wei Zheng, and Murat Kantarcioglu for valuable guidance on research and providing resources. Multiple industry friends at Google and startups have also been instrumental to my research and career development, including Drs. Kun Yang, Yi Cui, Yuan Xue, Fengjun, and Michael.

I also thank lab, study group, and school mates for their help, encouragement, and valuable discussions, including Csaba Toth, Dr. Weiyi Xia, Dr. Wen Zhang, Steve Nyemba, Dr. Muqun (Rachel) Li, Yongtai Liu, Zhiyu Wan, Chao Yan, Lina Sulieman, and Dr. Daniel Fabbri.

I am truly grateful for the various funding organizations that invited and supported me with valuable opportunities to multiple cutting-edge conferences, without which I would not be able to attend. These include NSF TRUST grant, AIST Japan, NSF fundings (University of Washington, University of Memphis, Brown University, and Schloss Dagstuhl of Germany).

Last but not least, I sincerely thank my parents and family for continued love and support throughout the years. I thank my love Pan for being a personal support, making my life colorful, and being unparalleled company. It has been a long and unusual journey and I really appreciate their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

Modern scientific investigations have increasingly relied on the expanded collection and analysis of data. Big data have played a vital role in novel discoveries across various disciplines. In genomic research, decreasing costs in high-throughput sequencing technologies, in combination with large repositories of clinical information (such as electronic health records or EHR linked to biobanks) [111], has enabled many novel discoveries by examining the associations between genetic variants and disease phenotypes. Among these are several active research paradigms, such as genome-wide association studies (GWAS) [20, 151] and phenome-wide association studies (PheWAS) [35]. These achievements are facilitated by increased collection and reuse of genomic and clinical data [57], as well as broad efforts to obtain larger sample sizes (by sharing and combing data) for increased statistical power [126].

At the same time, the unique and sensitive nature of human genetic data and its close connection to clinical sensitive information has led to numerous discussions around the governance of genomic records [52, 86, 43]. Currently, policy and advisory groups recommend removing identifying information (such as personal names) to uphold the privacy of study participants [102, 130].

I.1 Privacy Concerns in Data Sharing

In genetic data sharing, the efficacy of existing policies and protections is increasingly being questioned [132, 43, 12, 75, 11]. Recent years have witnessed various demonstrations [43, 59, 95, 69, 77, 134] that successfully extracted various kinds of privacy-sensitive information of individuals, such as the identity of study participants, study participation or disease status, or even exact traits. These studies have striking and long-lasting effects on various fields involving human subjects because the shared genomic data (often summary

statistics) enabling such privacy attacks have long been considered safe for individual-level privacy and openly published. For instance, in a famous study it was shown that a male's identity (mainly surname) could be ascertained by profiling his short tandem repeats (Y-STRs) on the Y-chromosome and referencing against various public genealogy databases on the internet – even if the individual's identity was not initially tied to a DNA sequence [59]. Earlier in 2004, it was illustrated that only fewer than 80 single nucleotide polymorphisms (SNPs) of a personal genome were necessary to uniquely distinguish an individual's sequence [95]. Next, in 2008 and later, it was shown that the originally widely-available (published) allele frequencies of genetic studies could reveal study participation and potentially even disease status [69, 78, 134]. Furthermore in 2012, it was indicated that releasing even basic summary information (i.e., association effect size or its direction) of genome-wide association studies (GWAS) could lead to study (or disease) related privacy leaks [77]. And in 2013, it was also suggested that the disclosure of one individual's genome sequence could even jeopardize his relatives' privacy due to the high correlation between familial genomes [74, 81].

The series of privacy attacks have already raised concerns from scientists, policy makers, and the general public. They have also led to reduced sharing of individual-level genome sequences and even site-level summary statistics of studies. For instance, based on [69], the National Institutes of Health (NIH) and Wellcome Trust stopped sharing aggregate genomic data directly to the public [167]. These privacy breach demonstrations have also influenced proposed regulations such as [143, 44, 45].

Other fields involving human subject data are sharing similar concerns. These include healthcare [107], social sciences (especially education and psychology) [32], machine learning [79, 40, 63, 62] and statistics [124].

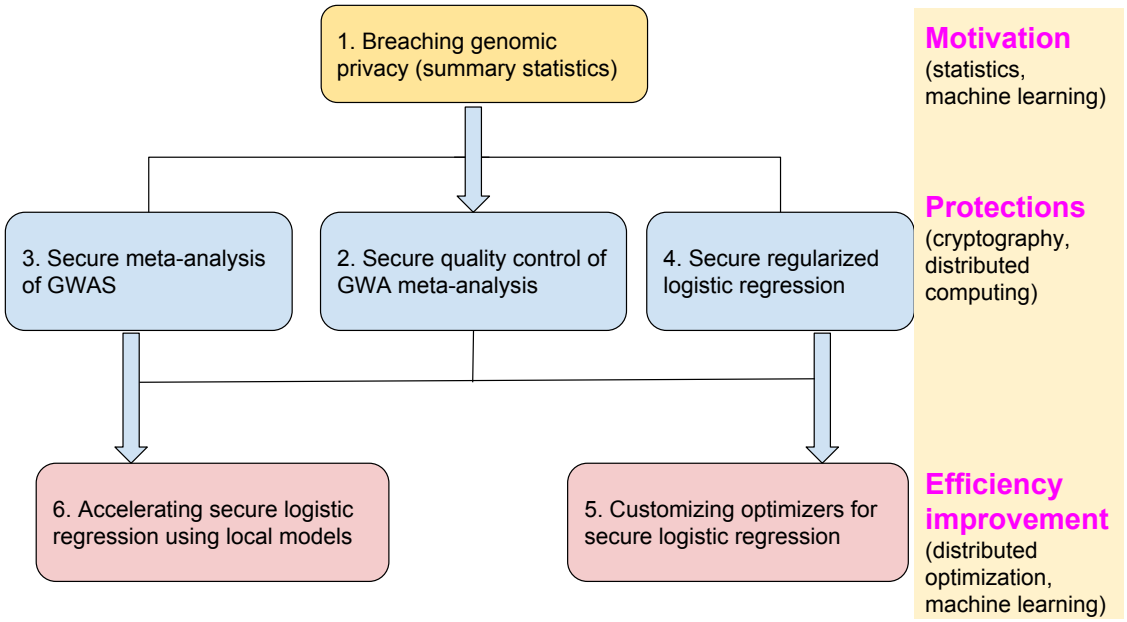


Figure I.1: An overview of this dissertation and its chapters. 1. We first propose multiple statistical inference attacks on genomic summary statistics (Chapter III), which provides novel findings as well as serving as motivations for this dissertation; 2. We then develop new methods to protect quality control of meta-analysis of genome-wide association studies (GWAS) (later in Chapter III); 3. Later, we present novel cryptographic solutions to protect meta-analysis of GWAS; 4. We also propose methods to protect distributed (regularized) logistic regression and benchmark extensively. Later on, to make cryptographic protections more practical and efficient, we propose two novel paradigms to accelerate cryptographic solutions: 5. by tailing numerical optimizers and 6. by leveraging local models.

I.2 Outlines for This Dissertation

The main focus of this dissertation is to quantify privacy risks in sharing genomic data, and provide novel and efficient methodologies for safeguarding participant privacy in statistical genetics and machine learning models. An overview of this dissertation is illustrated in Fig. I.1.

This dissertation begins by first proposing several statistical inference methods to reveal vulnerabilities of current practice of genomic data sharing including on individual- and summary statistics-level data. This will serve as the motivation for the rest of the dissertation. Specifically, in Chapter III, we introduce quality control (QC) for meta-analysis, a concrete example and very important topic in genetic research. This chapter will demon-

strate various statistical methods to breach personal privacy from genetic QC summary statistics, followed by our proposed solutions and QC pipeline based on cryptography and distributed computing to enhance privacy.

In Chapter IV, we describe a closely related and widely-used statistical method for large-scale genomic studies across institutions – meta-analysis, and propose novel cryptographic methods to safeguard this model and the underlying sensitive data.

In recognition of increasing popularity of advanced statistical and machine learning methods in genomics and related domains [94, 27, 170, 25, 166], we then cover such methodologies broadly in Chapter V and demonstrate how they can be made privacy-preserving. We choose logistic regression as a representative model throughout this chapter, due to its wide adoption in various domains such as genetics, biomedicine, and social sciences as well as being a routinely benchmarked model in machine learning and statistics research.

Later on, we observe that cryptography-based machine learning is still prohibitively slow for large-scale tasks. We thus propose novel models and algorithms from a distrbuted machine learning and optimization perspective to tackle the computational inefficiency challenge in cryptograhy, one of the major obstacles to practical secure solutions in the real-world.

In summary, Chapter V consists of three sections (each corresponding to a new contribution):

- First, in Section V.1, we present the design of a new state-of-the-art for privacy-preserving regularized logistic regression, leveraging distributed machine learning and cryptography. We also provide extensive empirical evaluations and benchmarks on performance that are lacking in existing literature.

- Second, in Section V.2, we present a contrasting perspective to privacy-preserving logistic regression and introduce the novel concept and method of tailoring numerical optimization for secure computing to drastically improve performance (in addition

to improvements of latest cryptography). Computational inefficiency is the biggest bottleneck of cryptographic protocols, and our solution provides an entirely different approach to make related protocols more practical. Moreover, it is cryptography-agnostic due to its design and can be built ontop of new (future) cryptographic schemes or systems.

- Last, in Section V.3, we introduce a novel paradigm for privacy-preserving distributed logistic regression, by leveraging local-site models to better guide numerical optimization. This paradigm improves significantly over which is common practice dominating the field of privacy-preserving distributed machine learning for over a decade. The drastic performance improvement from our new method makes cryptographic protocols more practical.

CHAPTER II

Background

In this chapter, I provide background information regarding several common genetic datasets used throughout this dissertation, and some main building blocks and methodologies for privacy protection.

II.1 Genetic Datasets

This dissertation leverages an extensive collection of datasets for demonstration and evaluation, ranging from human genome (including variants and summary statistics) and phenotypes data, various public datasets, and simulated datasets. Below, I briefly describe several datasets that are common for several studies in this dissertation. Details of application-specific datasets are postponed until their corresponding chapters.

II.1.1 eMERGE hypothyroidism study.

The first collection of datasets is from a genome-wide association study (GWAS) on hypothyroidism [36] provided by the Electronic Medical Records and Genomics (eMERGE) network [111]. This study focuses on a binary phenotype (hypothyroidism), with case/control ratio at 0.26. It consists of 6,370 study participants across five study sites/institutions who contributed data: i) the Group Health Cooperative (now Kaiser Permanente Washington), ii) the Marshfield Clinic, iii) the Mayo Clinic, iv) Northwestern University Medical Center, and v) Vanderbilt University Medical Center.

II.1.2 PAGE obesity study.

The second collection of datasets is from a genetic association study on obesity and body mass index [49] provided by the Population Architecture using Genomics and Epidemiology (PAGE) consortium [110]. This study focuses on a binary phenotype (obese or not)

and consists of 53,238 participants (37,823 European Americans and 15,415 African Americans in specific), and spans across six study sites: i) the Atherosclerosis Risk in Communities Study (ARIC), ii) the Coronary Artery Risk in Young Adults (CARDIA), iii) the Cardiovascular Health Study (CHS), iv) the Epidemiologic Architecture for Genes Linked to Environment (EAGLE) accessing the National Health and Nutrition Examination Surveys (NHANES), v) the Multiethnic Cohort (MEC), and vi) the Women's Health Initiative (WHI). We primarily used meta-analysis-related summary statistics from this dataset, thus no individual-level records are involved.

II.1.3 EAGLE diabetes study.

The third collection of datasets is from a genetic association study on type 2 diabetes provided by the EAGLE group [60], which is a sub-site of PAGE, and itself can be divided into two sub-studies associated with the National Health and Nutrition Examination Surveys (NHANES): i) NHANES III and ii) NHANES 1999-2002. This study focuses on a binary phenotype (obese or not) and contains 14,998 participants and spans several ethnicities (e.g., non-Hispanic white, non-Hispanic black, Mexican-American, and others). We used meta-analysis summary statistics (no individual-level records) from the study.

II.1.4 Other Public Genetic Data.

In addition to the aforementioned genetic datasets, there are also several widely used data repositories, including the 1000 Genomes Project [29], the International HapMap project [54], and the NCBI dbGaP database [106]. We postpone the introduction of these repositories and the data within them until they are first used in the dissertation.

II.2 Experimental Evaluation and Reproducibility

Unless otherwise specified, all experiments have been run multiple times to ensure stability and replicability of all findings throughout the dissertation. All proposed methods and models are deterministic (no randomness) in this dissertation. In addition, whenever pos-

sible and needed, we also demonstrate the replicability and generalization of the results on different datasets.

For reproducibility, I also share the main software packages and code I developed and experimental parameters when needed.

II.3 Building Blocks for Privacy Protection

II.3.1 Secure Multiparty Computation (SMC)

Secure multiparty computation (SMC) was initialized by the seminal work of Yao [164] and increasingly used in various domains and applications to allow for computations to be performed without violating the privacy of the underlying data. The use scenario is similar to ours: a group of independent organizations ("federation") hope to engage in a joint study (which can be represented as a statistical or machine learning model); they each possess their individual private datasets, and are not allowed to publicly disclose them to any external parties (including other member organizations in the collaborative study). The challenge is how to support such a collaborative study without actually sharing private data (which violates privacy otherwise).

Over the years, the original theoretical concept of SMC has been realized in working software and significantly improved to make it computational feasible (partially surveyed in [70, 98]). Currently, multiple methods (building blocks) are available and widely used to support various tasks following this privacy-preserving notation.

II.3.1.1 Yao's Garbled Circuit.

Yao's garbled circuit [164] is a popular secure two-party computation technique for securely supporting the evaluation of a joint function $f(x_1, x_2)$ from two sources of secret inputs, i.e., x_1 held by Party P_1 and x_2 held by Party P_2. In brief, function $f(.)$ is first represented in a binary circuit form. Then Party P_1 translates (called "garbles") the function circuit into a secure version (i.e., "garbled" versions of the circuit and computation table) based on its private input x_1. The resulting garbled circuit and computation table are later

sent to the opponent P_2, who initiates a 1-out-of-2 oblivious transfer protocol [164] (assisted by P_1; as mentioned above) to obliviously compute garbled values corresponding to his input x_2 and outputs the result to prescribed parties. The protocol reveals nothing to any parties other than the final result.

II.3.1.2 Additive and Linear Secret-Sharing Schemes.

The main goal of secret-sharing schemes is to split a secret value into multiple shares and let independent parties hold them collectively. In our case, the secret can be raw genetic or health-related data, or institution-level summary statistics. The secret is split such that it allows for: 1) some mathematical operations to be performed, and 2) reconstruction of secret (original or later derived).

Additive Secret-sharing.

Additive secret-sharing ([15] and the references therein) is based on simple mathematical arithmetic to split and recover secrets among several independent entities (or parties). For any secret m to be protected, the secret shares $s_1, ..., s_i, ..., s_k$ can be constructed such that: $m = \sum_{i=1}^{k} s_i$. In other words, one can first randomly generate shares for the first $(k-1)$ shares, and compose the final to satisfy the above equation. Because m is kept secret from any individual parties, the k shares are guaranteed to look random for any parties, thus providing provable security. A (more restrictive) 3-party instantiation of the scheme is commercialized and underlies the ShareMind software [15].

Linear Secret-sharing (Shamir's scheme).

Linear (or polynomial) secret-sharing (also known as Shamir's scheme) [136] is another classical and widely used scheme which is built on top of polynomial interpolation. 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 such that the only way to successfully recover the secret is through cooperation of at least t (i.e., the "threshold") share-holding parties (out of a total of w parties). 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 and modulo operation is thus required. However, for presentation simplicity, we skip the technical details):

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

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.

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 $<1,q(1)>,<2,q(2)>,...,<t,q(t)>,...,<w,q(w)>$. 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, because no more than a limited few Centers (and, in special cases, no single Center) 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 [136] to uniquely determine the polynomial $q(x)$. The secret can be derived by evaluating $q(0)$: $m = q(0)$.

II.3.2 Paillier Additively Homomorphic Encryption

There have been an increasing number of investigations into novel and efficient encryption schemes that guarantee strong security while allowing for (partial or all) computation on top of the encryptions (a property called homomorphism). Here, we begin with a widely used (partially or additive) homomorphic encryption scheme called Paillier's cryptosys-

tem [125].

The Paillier cryptosystem [125] is a public-key cryptographic scheme that is additively homomorphic (meaning that it supports additive computations as explained below). In brief, a message m ("secret") can be encrypted by:

$$Enc(m,r) = g^m r^n \mod n^2 , \tag{II.2}$$

where $n = pq$ corresponds to an RSA modulus, g is a public parameter and r is a randomization. Here the public (i.e., encryption) key would be (n,g), and the private (i.e., decryption) key would be (p,q,λ) (where λ equals the least common multiple of $p-1$ and $q-1$).

In addition to its strong security guarantees, Paillier cryptosystem also possesses a few useful partially homomorphic properties. For instance, secure addition can be computed as:

$$Enc(m_1 + m_2, r_1 r_2) = Enc(m_1, r_1) \times Enc(m_2, r_2) \mod n^2$$

and multiplication-by-constant follows (for a constant k):

$$Enc(k * m) = Enc(m,r)^k \mod n^2$$

II.3.3 Differential Privacy

Differential privacy [37] is a rigorous notion of privacy which guarantees that the output of a function is almost unchanged in the presence or absence of any specific data record. Differential privacy has received extensive investigation in recent years and is especially popular in machine learning and database domains [39]. Note that this dissertation focuses on a cryptographic notion of privacy which primarily intends to protect the intermediate results and computation process, as opposed to final result privacy (main goal of differential privacy). Regardless, many of our proposed privacy protection methods can be extended in

a straightforward way to become differentially private by injecting calibrated noise into the final output of our protected model (this is beyond the scope of this dissertation).

More formally, differential privacy has the following definition:

Definition II.3.1. (Differential Privacy) *A randomized function f (with output space Ω and well-defined probability density P) is ε-differentially private if for all adjacency data sets D, D' that differ in a single record and for all measurable sets $\omega \in \Omega$:*

$$\frac{P[f(D) \in \omega]}{P[f(D') \in \omega]} \leq e^{\varepsilon} \ , \tag{II.3}$$

Differential privacy essentially implies that even if a strong adversary knows the whole dataset D except for the target record (individual), he still cannot infer much information about the target from the function output.

II.3.3.1 Output Perturbation.

A popular way to achieve differential privacy is output perturbation, which calibrates artificial noise to the exact output of the function. The level of noise is carefully chosen based on the sensitivity of the function, which measures the maximum change in the function output when a single record in the input data is changed [37].

Definition II.3.2. (Sensitivity) *The l_2 sensitivity of function f is defined as:*

$$S(f) = \max_{D,D'} \|f(D) - f(D')\|_2 \ , \tag{II.4}$$

where $\|.\|_2$ denotes the l_2 or Euclidean norm.

It has been shown that, for a function f and the desired privacy parameter ε, the output perturbation returns $f(D) + \eta$, where noise η is generated according to density

$$P(\eta) \propto \exp(-\frac{\varepsilon}{S(f)} \|\eta\|_2) \tag{II.5}$$

CHAPTER III

Privacy Leaks in Quality Control on GWAS Meta-analysis and Effective Countermeasures

This chapter is based on our work in [160]. My contribution in this work includes conception, design and supervision of the study, implementation and experimental evaluation, analysis of results, writing the manuscript and addressing reviewer comments.

Quality control (QC) is fundamental to reliable and reproducible genome research. This is particularly the case for meta-analysis of genome-wide association studies (GWA meta-analysis), where data are contributed by disparate and often heterogeneous cohorts. Traditionally, QC for meta-analysis is enforced by sharing and contrasting summary statistics beyond their respective cohorts, under the belief that such summaries are respective of personal privacy and thus ethically safer to disclose than individual-participant data. Our investigation, however, refutes such a belief and pinpoints a series of privacy vulnerabilities of current QC practices for GWA meta-analysis, which result in the leakage of sensitive information of individual participants. Notably, empirical assessments using real GWA meta-analyses on 6370 participants indicate that our demonstrated inference attacks can reveal 1) GWAS participation status (with accuracy of 99.9% in one type of attack, and AUC of 0.78 in another), 2) disease status (i.e., case/control) for dichotomous phenotypes (with 1.00 in AUC), and 3) quantitative traits (with R^2 correlation of 0.96 against actual traits). Furthermore, we demonstrate countermeasures for mitigating privacy risks by developing novel technological protections and present a privacy-preserving QC pipeline. Empirical evaluations on several consortium studies suggest that our secure pipeline enhances participant privacy, and incurs only modest computational overhead. Our implementation is available at: http://github.com/XieConnect/SecureQC.

III.1 Introduction

Meta-analysis of genome-wide association studies (GWA meta-analysis) is a dominant method for detecting associations between genetic variants and traits [46, 151]. Increasingly, modern studies mandate large sample sizes [127, 21, 115] that can be achieved only through data sharing and result synthesis across many institutions [46]. The success of GWA meta-analysis is contingent upon the quality of source data [142], which is typically enforced by an essential process of quality control (QC). Currently, various QC procedures and toolkits for GWA meta-analysis are in practice at various consortia [51, 144, 153].

Meanwhile, plans for more extensive collection and sharing of genomic data have raised major concerns around data management, especially on privacy issues [43]. Recent years have witnessed a burgeoning number of far-reaching studies demonstrating privacy vulnerabilities due to inappropriate or unitended disclosure of genomic information ranging from individual genomic records to various summary statistics (such as those from GWAS) [43, 95, 59, 69, 77, 78, 134]. Increasing concerns over genome privacy could negatively impact study participant recruitment, wide dissemination of research results [167], data reuse and data access policies [143], all of which are fundamental for enabling novel, as well as reproducible research.

Despite the pervasive adoption of QC in multi-site meta-analysis, no investigation has considered the extent to which QC procedures are vulnerable to privak leaks. To answer this question, here we perform a systematic privacy assessment on common QC procedures and manage to pinpoint various privacy leaks throughout the process via intuitive demonstrations. Our privacy assessment covers individual-participant genomic data as well as various summary statistics, both of which are routinely shared beyond their originating institutions for QC purposes and are widely perceived to be privacy-safe in practice (such as in [153]).

Our privacy assessment, on the contrary, indicates that widely-used QC practices for GWA meta-analysis disclose much information (e.g., allele frequency and effect size of

GWAS associations) beyond their respective institutions, leading to unanticipated disclosure of individual-participant sensitive information. Notably, our empirical demonstrations on several large consortia studies suggest that we can breach genomic privacy with high success, leading to unanticipated revelation of study participation status (with accuracy as high as 99.9% $\sim$ 100%), disease case and control status of individual participants (with area under the ROC curve, or AUC, of 0.70), and accurate recovery of quantitative traits (with Pearson $R^2 = 0.96$). Surprisingly, our findings contradict with common belief that meta-analysis (and its integrative process of QC) and summary statistics-based methods are preferred approaches and considered ethnically safer for supporting large GWAS due to concealment of sensitive individual-participant genomes, over pooled analysis of directly consolidating individual-participant data [46, 152].

Later on, to mitigate the aforementioned privacy risks, we develop a privacy-enhanced QC pipeline with provable privacy guarantees. We validate our new pipeline with synthetic and real-world studies from several large consortia [36, 49, 60]. Empirical evaluations indicate that our protections can prevent these privacy attacks on human genome, while introducing little computational and monetary overhead for actual deployment.

Our privacy protection software, SecureQC, is freely available at: http://github.com/XieConnect/SecureQC

III.2 Privacy Inference Attacks and Cryptographic Protection

Here we first describe our methods to infer (quantify) private information from summary statistics in meta-analysis QC. We will then analyze QC sub-procedures in detail and propose ways to mitigate the aforementioned privacy risks, using distributed computing and cryptography.

III.2.1 Privacy Inference Attacks

The summary statistics exposed in meta-analysis QC are vulnerable to multiple types of privacy breaching attacks, which we introduce below.

III.2.1.1 Study Participation Status Inference from Allele Frequencies.

Allele frequency can be leveraged to distinguish participants from non-participants in GWAS [69]. Effect allele frequency (EAF), which is routinely shared for meta-analysis QC, is vulnerable to such risk. To illustrate this, for each target individual i, we measure the deviations of his genome (denoted Y_{i}) from two different allele frequency references: the study mixture (the EAF shared in QC; denoted *Std*), and a pubic reference panel (e.g., the 1000 Genome Project [29] or HapMap [54]; denoted as *Ref*). We then accumulate such deviations over many SNPs (indexed by j), leading to distinct distributions of attack score for study participants and non-participants. For individual i and SNP j, the distance measure is defined as [69]:

$$D(Y_{i,j}) = |Y_{i,j} - Ref_j| - |Y_{i,j} - Std_j| \ , \tag{III.1}$$

where Ref_j and Std_j denote the allele frequencies for SNP j from public reference and study mixture, respectively. To quantify the differences in distribution of the deviations, we take a one-sampled t-test across all M SNPs and derive the risk score for each target individual i [69]:

$$T(Y_i) = \frac{E(D(Y_i))}{SD(D(Y_i))/\sqrt{M}} \tag{III.2}$$

where *E(.)* denotes the expectation, and *SD(.)* is the standard deviation.

III.2.1.2 Inference of Exact Traits and Study Participation Status.

GWAS summary results, such as effect size estimates and direction of the effect, can be leveraged to breach privacy of target individuals (e.g., recovering quantitative eQTL traits [77]). This works well especially when GWAS regression overfit to the data (i.e., fitting too well). Specifically, given the regression estimates (denoted β_j for each SNP j) and the target individual's genotypes (i.e., predictor $X_{i,j}$ for SNP j), we can predict the corresponding response variable for Individual i (i.e., the sensitive trait; denoted Y_i) as averaged

across many SNPs [77]:

$$Y_i = \frac{n}{M} \sum_{j=1}^{M} \beta_j (X_{i,j} - Ref_j) \ , \tag{III.3}$$

where Ref_{j} denotes the public reference panel (for SNP j), n denotes the total number of individuals, M is the number of SNPs.

For dichotomous traits, we design a different attack score (or inferred trait; in the range between 0 and 1):

$$Y_i = 1/\{1 + \exp[-\frac{n}{M} \sum_{j=1}^{M} \beta_j (X_{i,j} - Ref_j)]\} \tag{III.4}$$

The direction of the effect can also reveal much private information, such as recovery of quantitative traits [77]. The inference approach is based on the aforementioned attacks on effect size estimates, such that:

$$S_i = \frac{1}{M} \sum_{j=1}^{M} sign(\beta) \ sign(X_{i,j} - Ref_j) \ , \tag{III.5}$$

where *sign(.)* corresponds to the sign of the data.

III.2.2 QC practice and adversarial scenarios.

We focus on the common scenario of conducting GWA meta-analysis across multiple cohorts in consortia. Such a collaborative study typically involves the following entities: i) multiple local sites who possess (private) individual-level data and contribute (summary) data to the joint study, ii) a coordinating center (i.e., the central authority) who is responsible for coordination, data consolidation and management, and joint analysis/computations (e.g., QC and meta-analysis).

In GWA meta-analysis, the primary target of protection is the privacy of individual study participants, including sensitive attributes such as disease (participation) status and quantitative traits of disease-related measurements.

By assessing privacy, our ultimate goal is to identify and protect procedures in which private information is disclosed to potentially malicious parties – it could be another partic-

ipating site in the consortia who wants to peek into others' data, or a breached coordinating center or participating site who wants to infer personal sensitive information from genome, or even a malicious insider/employee at the coordinating center who hopes to gain sensitive information about all sites and their study participants. Therefore, we consider genome privacy to be violated if: i) sensitive individual-level genome information is directly revealed to unintended parties (e.g., other sites in or outside the consortia, including the coordinating center); ii) summary statistics of studies are disclosed elsewhere (e.g., other sites or the coordinating center) which prove to be directly linkable to private information about individual participants (examples include a large body of inference attacks on genetic privacy [43, 69, 78, 134, 77, 124]).

III.2.3 Major QC procedures.

We present a high-level overview of the typical QC workflow for meta-analysis in Figure III.2. Most QC procedures deal with data quality problems at one the following three levels: i) site-specific QC which mainly checks and cleans data files from each site; ii) cross-site-level QC which contrasts diagnostic summary statistics or plots across different sites to identify site-level issues, and iii) post-study-level QC which mainly verifies meta-analysis result reliability through heterogeneity tests,.

Current QC practice relies on one centralized entity such as a Coordination Center (or other analysis organizations) to perform all the procedures, even though this entity is not fully trusted to host privacy sensitive data, such as individual genomes (protection of individual raw data is a major reason advocating meta-analysis in the community). There is a common belief that summary statistics for QC maintain individual privacy. Here we analyze the summary statistics disclosed at various levels of QC, their privacy implications, and our proposed protections.

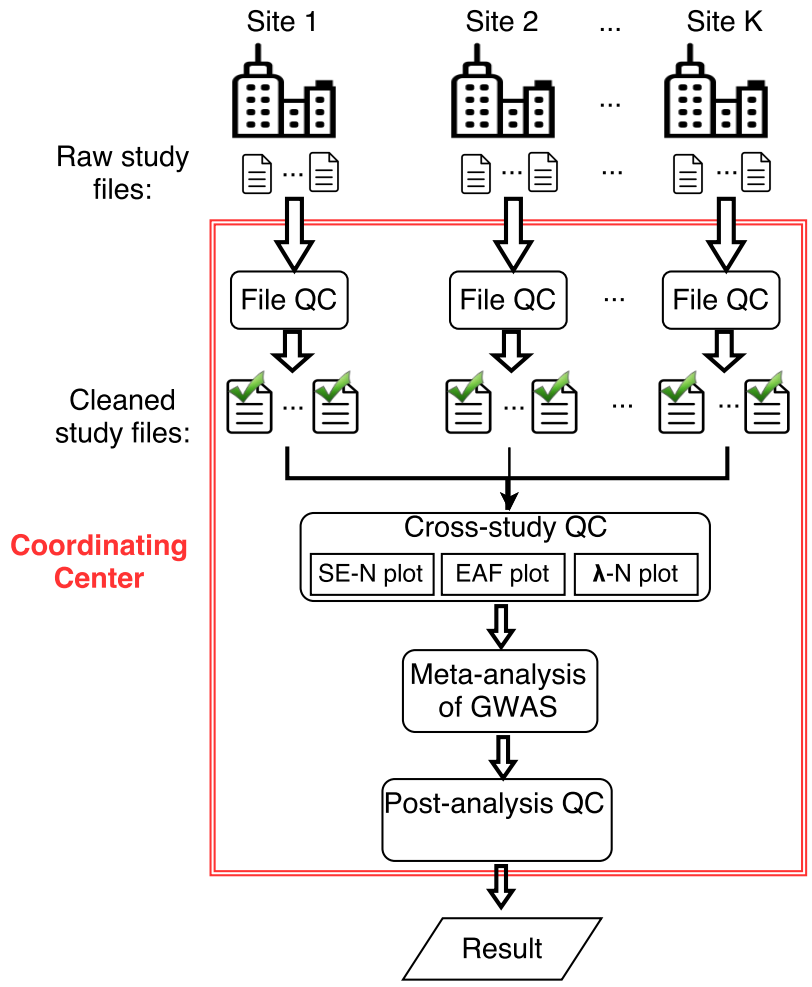


Figure III.1: Meta-analysis QC pipeline.

Figure III.2: The meta-analysis QC workflow and disclosed summary statistics. Each site performs their respective GWAS, after which result files are submitted to the Coordinating Center for QC and meta-analysis. The QC process is invoked through a series of layers: File QC, Cross-study QC, and Post-analysis QC. During the process, various summary statistics are disclosed beyond their originating sites.

III.2.3.1 Site-specific QC.

This goal of site-specific QC is mainly to perform a series of local checks on various quality issues. Common checks include: removing monomorphic SNPs or SNPs with incomplete or inaccurate information, requiring a minimum sample size per study and a minimum number of minor alleles contributing to an SNP for each participating study, filtering by imputation quality using a threshold, and so on.

III.2.3.1.1 Privacy Analysis.

To assess the privacy on site-specific QC, it is important to first determine where these QC checks are performed. If all these procedures fall within the responsibility of local sites, then this process does not incur privacy issues, since there are no data sharing beyond their owner site. The analysis below mainly concerns the other scenario where the coordinating center performs site-specific QC's.

Site-level QC requires access to highly detailed genome information, including all allele information, association effect size estimates, standard errors, etc. Sharing such information beyond the originating site could raise privacy concerns, since much of such information could lead to inference attacks on privacy, such as inference attacks based on summary statistics of association [124], linear regression coefficients [77], and disease case/control status inference on allele frequencies [69].

While it is possible for the consortia center to perform site-specific QC procedures, we point out that it may not be a wise choice, given the complexity of the task and potential privacy issues.

III.2.3.1.2 Our Protection.

We propose to offload the site-level QC checks to individual sites due to efficiency and privacy considerations. We observe that nearly all filtering criteria at the site-level can be standardized and distributed either as computer scripts or guidelines. By asking local sites to perform such tasks, the privacy issue is naturally avoided since no data sharing is necessary. As a side result, the QC procedure could be accelerated as local sites have the greatest expertise in checking and cleaning their own local data.

III.2.3.2 SE-N plot.

The general concept for the SE-N plot is as follows: For each study file, depict the inverse of the median standard error of the beta estimates across all SNPs against the square root of the sample size. High-quality data should yield a straight (diagonal) line.

Suppose we denote the sampling variance of a linear regression-derived beta estimate of a specific SNP j as SE_j, the variance of the phenotype as $Var(Y)$, the sample size as N. Then according to [153], the theory underlying the plot is that

$$c\sqrt{Var(Y)} * \frac{1}{median(SE_j)} = \sqrt{N}, \tag{III.6}$$

where constant $c = median(\frac{1}{\sqrt{Var(X_j)}}) \approx median(\frac{1}{\sqrt{2MAF_j(1-MAF_j)}})$ and is computed per study file (dependent on ethnicity, genotyping platform, imputation reference panel, and imputation quality).

For the final QC comparison, we plot the $\frac{c}{median(SE)}$ v.s. $\sqrt{N}$. So in fact the plot will indicate variance of the phenotype.

III.2.3.2.1 Privacy Analysis.

Standard practice asks each site to submit the SE-N plot with their data. So here we analyze whether the plot itself reveals sensitive information. Specifically, several statistics are implicitly contained in the composite terms (e.g., tuples of $\langle$*median(SE), sample size*$\rangle$) disclosed by the plot, including genotype frequencies or genotype dosages and imputation quality, median standard error, and the sample size. However, most of these information does not pose privacy concerns. While genotype dosage may be sensitive, it seems there is no easy way to distill such information from the revealed plot.

III.2.3.2.2 Our Protection.

The generation of the above plot is typically done locally at each site. So the generation itself does not incur privacy issues. And we just showed that the plot itself does not reveal sensitive information either. So it appears it is relatively safe to release the plot.

However, if we want to maximize protection by avoiding information disclosure, we may want to perform the SE-N plot solely at local sites. This is because comparison of SE-N plots across sites does not bring additional benefits for the judgment on site-level

quality, since the "gold standard" for SE-N is quite obvious and public knowledge (i.e., the identity line in the plot). Such quality check can be easily enforced by local sites without cross-referencing outside counterparts.

III.2.3.3 The P-Z plot.

Standard practice for generating the P-Z plot aim at comparing the reported p-value against the that derived from computed Z-statistic based on reported beta estimate and standard error. This way, potential analytical problems involving the above statistics could be observed.

III.2.3.3.1 Privacy Analysis.

By examining the above process, we note that the computation itself may be vulnerable to inference attacks based on regression beta, p-value and standard error [77, 148].

III.2.3.3.2 Our Protection.

To eliminate privacy concerns, it is advised that the computation and comparison be performed locally at each sites. This way, no sensitive data sharing is necessary and the decision and correction can be performed solely at local sites without privacy problems.

If cross-site comparison, or center-led judgment, is truly necessary, advanced secure computation technologies can be used to quantity the correlation of the two parameters across sites.

III.2.3.4 Effect allele frequency (EAF) plot.

The general concept of the EAF plot is to compare local data statistics against a reference set to visually identify quality issues (e.g., strand issues, allele miscoding, misreported ancestry, and so on)

III.2.3.4.1 Privacy Analysis.

The EAF plot requires disclosing detailed EAF information, which might be vulnerable to inference attacks based on allele frequencies [69, 134, 78].

III.2.3.4.2 Our Protection.

From a scientific perspective, cross-site comparison also seems unnecessary, since the reference set ("gold standard") is typically public. Local sites can easily perform local checks against those public references. So it seems more appropriate to retain the data and analysis at local sites to resolve privacy concerns.

However, if cross-site comparison is indeed necessary, SMC protocols can be leveraged to support the task.

III.2.3.5 The lambda-N plot.

Plotting genomic control inflation factor λ_{GC} increases with sample size can help detect population stratification [153]. The genomic inflation factor λ_{GC} is defined as the ratio of the median of the empirically observed distribution of the test statistic to the median under the null hypothesis, thus quantifying the extent of the bulk inflation and the excess false positive rate.

$$\lambda_{GC} = median(\chi^2)/0.456 \tag{III.7}$$

III.2.3.5.1 Privacy Analysis.

We note that the derivation of λ_{GC} depends on (a χ^2-test for) allele frequencies, which is know to reveal disease and study participation status [69]. Thus, the computation process should be safeguarded. Meanwhile, the factor itself does not seem to reveal sensitive information. Specifically, only one value (i.e., median of χ^2) was revealed which is not sufficient for any known privacy attacks.

III.2.3.5.2 Our Protection.

Given the sensitive nature of computing λ_{GC}, we propose to compute site-specific λ_{GC} locally at each site (in fact already a common practice by many consortia such as GIANT [153]). Since the gold standard and threshold for distinguishing good from bad statistics can be defined *a priori* (e.g., 1.0 and 1.1, respectively), local sites can perform their individual checks using computer scripts distributed by the consortia.

The site-specific factors could also be pooled by the consortia center for a global plotting and comparison, which technically does not incur privacy issues. However, such centralized pooling should be avoided to maximize privacy guarantee. Under certain situations where centralized checking is truly necessary and minimal information should be disclosed, we suggest a secure computation-based solution. Specifically, GC factors are encrypted prior to submission and the center will perform threshold-based comparison or outlier detection to pinpoint those problematic factors and sites.

III.2.3.6 Heterogeneity Tests.

The heterogeneity test QC [153] is mainly performed after the main meta-analysis task is performed. The goal of related tests is to double check the reliability of meta-analysis result by quantifying the heterogeneity in the data used. Three closely related test statistics are typically used in practice: i.e., the Q-, I^2-, and H-statistics [68, 153, 50]. Among these tests, H and I^2 are preferred measures by their original authors [68]; and in modern GWA meta-analyses, typically Q-statistic and I^2 are used [153, 116, 50]; some studies only reported I^2 [80].

Here we briefly introduce the three common heterogeneity tests and their relations.

The *Q-statistic* [68] aggregates weighted variances across sites to test heterogeneity:

$$Q = \sum_{i=1}^{k} w_i (y_i - \hat{\mu_F})^2, \tag{III.8}$$

where k is the total number of sites, w_i is the weight (e.g., the inverse of variance for fixed-

effect meta-analysis), y_i is the estimate from site i, and $\hat{\mu_F} = \sum y_i w_i / \sum w_i$ is the weighted estimate (i.e., note that it is *NOT* the Z-score from meta-analysis because $Z = \sum y_i w_i / \sqrt{\sum w_i}$).

The *H-statistic* [68] improves over Q-test by quantifying the relative excess in Q-statistic over its degrees of freedom:

$$H^2 = \frac{Q}{k-1}, \tag{III.9}$$

where k is the total number of sites, Q is the aforementioned Q-statistic. We typically report $\max\{H, 1\}$ [68]. Intuitively, $H = 1$ implies homogeneity of effects (since $E(Q) = k-1$). Larger values mean heterogeneity is present (empirical thresholds are given in [68]).

The *I^2-statistic* [68] indicates the fraction of total variation in the estimate that is due to between-study heterogeneity.

$$I^2 = \frac{H^2 - 1}{H^2} = 1 - \frac{k-1}{Q}, \tag{III.10}$$

where Q, H are the aforementioned test-statistics, and k is the total number of sites. An I^2 of 0 means no heterogeneity.

III.2.3.6.1 Privacy Analysis.

Computing the *Q-statistic* requires several potentially sensitive inputs, such as effect size estimates (y_i), standard errors (related to w_i), many of which are subject to generic inference attacks on summary statistics [124, 77]. Since H^2 is directly correlated with Q, so it is vulnerable to the same privacy threats. According to the original definition of I^2, it requires between-study heterogeneity and within-study variance, the computation of which both involve individual-SNP-level summaries which could be sensitive. So a straightforward implementation of I^2 will bring about more privacy risks. However, if we choose to compute I^2 leveraging its correlation with Q via Equation III.10, it would be vulnerable to the same set of risks as computing Q (note that the number of sites k is not considered

sensitive).

III.2.3.6.2 Our Protection.

Our goal is to securely compute the statistics defined in Equations III.8, III.9 and III.10. As shown before, these statistics are correlated and can be derived from one another easily. So here we mainly focus on how to securely compute Q-statistic.

So to summarize, our major challenge in supporting these heterogeneity tests is to securely compute the Q-statistic (Equation III.8). When designing a secure-version Q-statistic, we have the following assumptions:

1. Inputs y_i, w_i are private and should not be disclosed to parties other than their respective owners P_i.

2. The weighted estimate $\hat{\mu_F}$ is also private, and its derivation also occurs in secure.

3. The Q-statistic final result should only be disclosed to the query issuer (i.e., the user).

In Equation III.8, $\hat{\mu_F}$ has to be derived through an expensive process involving secure division.

Here we describe a simplified solution to securely compute Equation III.8, without relying on secure division. The new approach is based on the following observation:

$$\begin{aligned} Q &= \sum_{i=1}^{k} w_i y_i^2 + \sum_{i=1}^{k} w_i \hat{\mu_F}^2 - 2\sum_{i=1}^{k} w_i y_i \hat{\mu_F} \\ &= \sum_{i=1}^{k} w_i y_i^2 - Z^2 \end{aligned} \tag{III.11}$$

There can be two alternatives to implement the above computation:

1. Straightforward approach: terms $w_i y_i, Z^2$ were already provided by SecureMA [159] (to be introduced in later Chapter IV), so we only need secure multiplication (i.e., $w_i y_i * y_i$), secure addition and subtraction.

Table III.1: Privacy issues with QC summary statistics and countermeasures.

Summary statistic	QC procedure	Privacy leaks	Suggested protection
SNP identifier	File QC, Meta-analysis	Unique variants (sets)	Secure hashing
Allele frequency	File QC, Cross-study QC	Study participation [69, 134]	Distributed computing
Effect sizes, standard error	File QC, Meta-analysis, Post-analysis QC	Study participation, disease status/traits [77]	Secure computation [159]
Direction of effect	File QC, Meta-analysis, Post-analysis QC	Study participation, disease status/traits [77]	Secure computation [159]

2. Lightweight approach: We ask sites to directly submit encrypted composite terms $E(w_i y_i)$ and $E(Z^2)$. This way, we only require secure addition and subtraction to complete the above computation.

III.3 Experimental Design and Results

We conduct a systematic privacy assessment on various types of summary statistics that are routinely shared beyond their originating institutions during meta-analysis QC. We demonstrate that privacy sensitive individual-level information can be extracted with high success from different levels of QC and through various approaches. To mitigate the aforementioned privacy leaks, we develop and evaluate our privacy-enhanced QC pipeline based on cryptography and distributed computing.

Before delving into the details, we provide a high-level overview of summary statistics for meta-analysis QC, related privacy issues, and (our) proposed countermeasures in Table III.1.

Our empirical evaluations for both privacy breaching attacks and protections span several large meta-analyses from several large consortia: 1) a hypothyroidism study of 6,370 participants from 5 institutions within the eMERGE network [36], 2) an obesity study of 53,238 participants from 6 institutions in the PAGE consortium [49], and 3) a Type II Di-

abetes study with 14,998 participants from 2 sites (sub-studies) from the EAGLE consortium [60]. In addition, for comprehensive evaluation, we also simulated additional studies with pseudo quantitative traits based on real genotypes from the eMERGE network (see corresponding subsections for details), similar to earlier publications [77].

Below, we first present our emprical privacy assessment on various QC summaries. Later, we evaluate our privacy-preserving QC pipeline.

III.3.1 Site-level QC.

Currently, site-level QC is often performed by a central organization to check for problems in various site-level submission files (i.e., inputs to meta-analysis), such as formatting errors, missing values, nonsensical values, imputation quality issues, and so on. Our privacy assessment indicates that site-level QC is highly vulnerable to privacy breaches when performed at one centralized location (e.g., Coordinating Center or designated analysis organization), which is (unfortunately) often common practice [153]. This is because submission files from each site disclose very detailed genomic information such as Z-score, p-value, effect allele frequencies (EAF), and so on, which are privacy sensitive and could lead to a series of privacy leaks about individuals' sensitive disease and study participation status (many of which will be demonstrated later).

III.3.1.1 Cross-site QC.

The goal of cross-site QC is mainly to detect site-specific data problems by cross-referencing other sites. It typically involves a few widely accepted procedures, which are all conducted by a central organization.

III.3.1.1.1 Effect allele frequency (EAF) plot.

The EAF plot is utilized in QC to contrast allele frequencies across different sites to detect data anomalies. Unfortunately, disclosing allele frequency is known to be vulnerable to privacy inference attacks on participation status in (potentially sensitive) GWAS [69].

To show that disclosing effect allele frequencies from cross-site QC incurs privacy issues, we aim to distinguish between participant (i.e., in-study) and non-participant (i.e., hold-out) individuals using (site-level) EAF summaries and public reference genomes (e.g., the 1000 Genome Project [29] or HapMap project [54]). Given the genotypes of each target individual, the privacy breaching method will quantify a per-person risk score, of which the distribution may differ significantly between in-study and holdout individuals. Using 1456 in-study participants from an eMERGE study and 99 holdout samples from the 1000 Genome Project, our inference on GWAS participation status is highly successful as evidenced by the obvious pattern differences of attack scores between in-study and holdout groups (Fig. III.3). Based on the $y = 0$ threshold, almost 99.9% of the in-study and 100% of the holdout individuals can be correctly discriminated. Similar results have also been observed on other GWA meta-analysis studies from different consortia. This confirms that in cross-site QC, the revelation of EAF can disclose sensitive (disease or trait) information of individuals.

III.3.1.1.2 P-Z plot.

The p-Z plot is another useful procedure in QC to ensure the correctness of Z-score (GWAS association effect size) from per-site file submissions. Unfortunately, disclosing such information may lead to detection of (sensitive) disease status. Specifically, when coupled with public genome references (e.g., the 1000 Genome Project [29]) and the genotypes of the target individual, an adversary could infer the exact traits of the target individual (either quantitative or binary), as well as detecting his study participation status (i.e., in-study or holdout). Similar to [77], we simulated quantitative traits (e.g., blood pressure, eQTL traits, and so on) based on real genotypes from an eMERGE study and a simple additive genetic model [163]. We tested different simulations with different parameters, such as disease prevalence being from 0.01 to 0.1, along with different causal SNPs. Since our statistical attack model does not make any assumptions on these parameters and they all

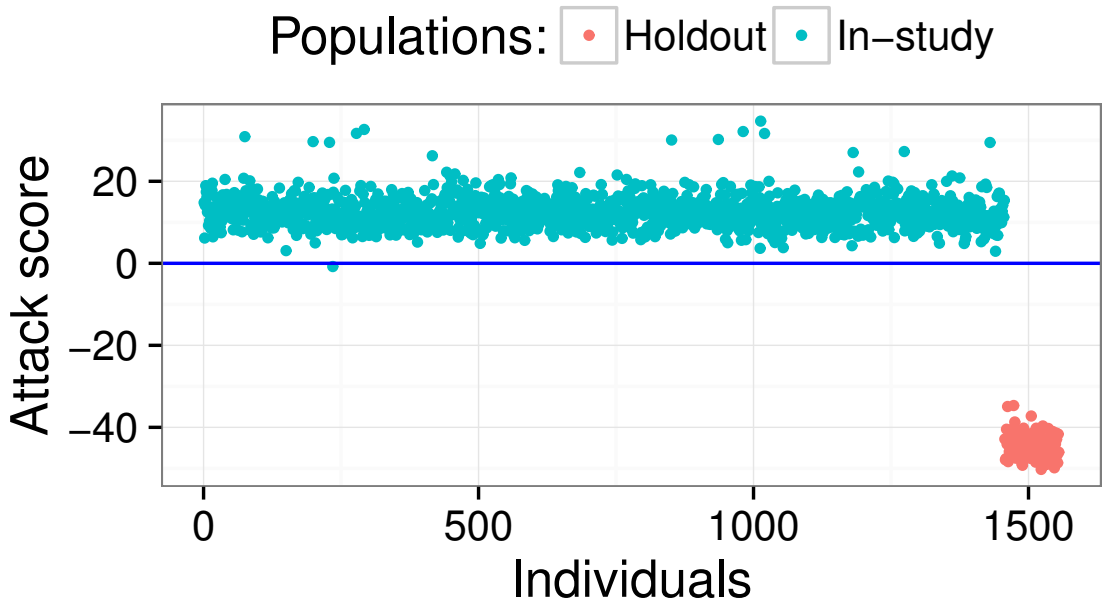
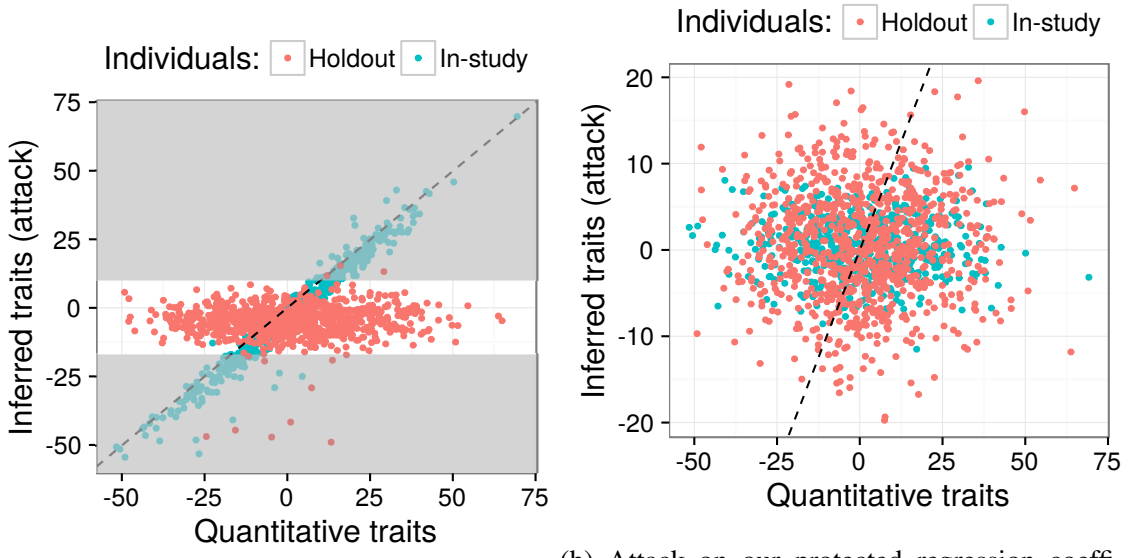


Figure III.3: Detection of GWAS participation status on target individuals using QC effect allele frequencies. Each individual (x-axis) from the eMERGE study is quantified an attack score (y-axis) per the inference attack method, and the difference in distributions of the score reveals GWAS participation status. With the $y = 0$ threshold, individuals with inferred scores above the threshold are likely to be (classified as) real participants in the released study, while individuals below the threshold are non-participants.

lead to similar attack results, we report one simulation result selected at random (disease prevalence = 0.1). In Fig. III.4a, we present our privacy inference results of recovering (simulated) quantitative traits on an eMERGE study (with real genotypes and simulated traits). It is shown that the recovery of quantitative traits is highly successful for in-study individuals (with a Pearson correlation $R^2 = 0.96$), while holdout individuals are centered around the (normal-looking) $y = 0$ line and their traits are not predictable (as expected, since they did not contribute to the study underlying the disclosed summaries). As illustrated in Fig. III.4a, the inferred traits from this method can also be used to distinguish GWAS participation status, with in-study individuals shown in the upper and lower regions (highlighted) and the holdout individuals within the middle region around $y = 0$ line. Inference on GWAS participation yields great detection power, with AUC = 0.78 (much better



(a) Attack on (raw) regression coefficients.

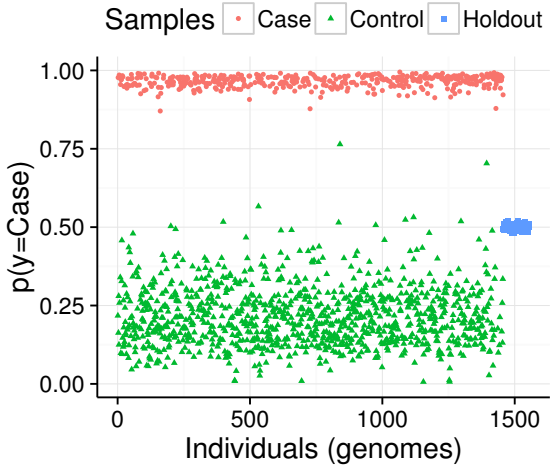
(b) Attack on our protected regression coefficients.

Figure III.4: Trait inference using regression coefficients when: (a) without protection (i.e., standard practice), and (b) after applying our protection. All results are based on eMERGE dataset (516 and 940 individuals for the in-study and holdout test samples, respectively; Both are genotyped on 368,657 SNPs.). Before protection, the inferred traits (y-axis) are in excellent alignment with the actual traits (x-axis) on the diagonal line, implying that traits are well predictable. Looking vertically, the distinction in the value ranges of inferred traits also allows us to discriminate participation status, where highlighted regions (non-white) are dominated by in-study individuals, and the middle white region by holdout individuals. After protection, trait inference is almost random and unsuccessful, and participation status is obfuscated too.

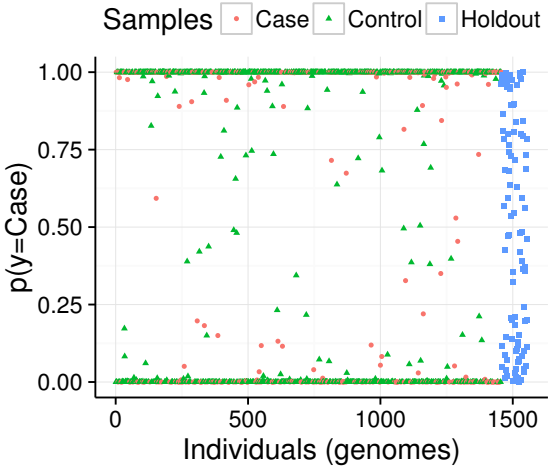
than random guessing of 0.5). Dichotomous trait-based GWAS is also vulnerable to similar attacks on Z-score or p-value, as demonstrated in Fig. III.5. Contrasting patterns are present in the distributions of our inferred scores, which allow us to differentiate in- and out-of-study (holdout) individuals (with AUC = 0.70), as well as to tell apart cases and controls (with AUC = 1.00) within the in-study sub-population.

III.3.1.2 Post-analysis QC.

Post-analysis QC mainly checks for meta-analysis output through several heterogeneity tests. To do so, it requires cross-site sharing and contrasting of summary statistics that



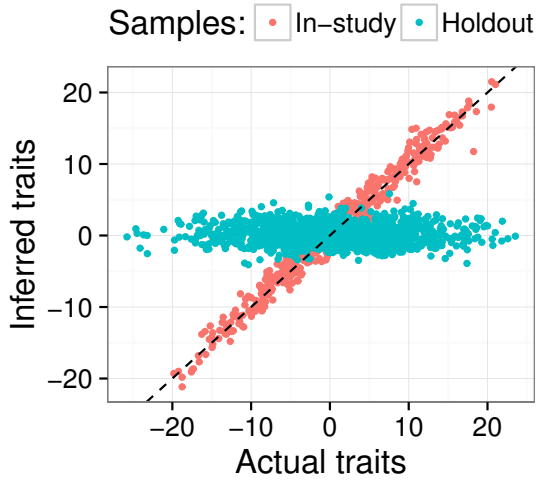
(a) Using originally disclosed effect size estimates.



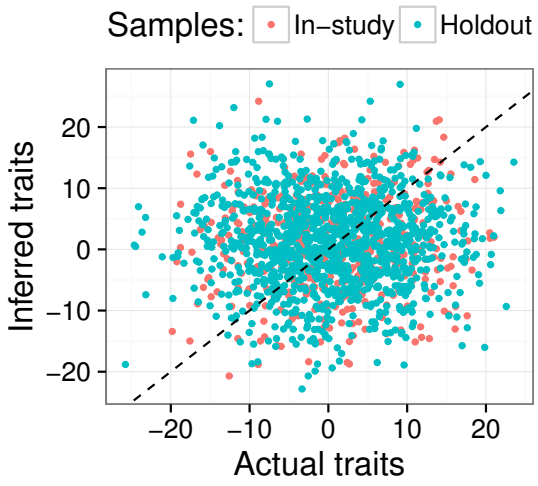
(b) Using protected effect size estimates.

Figure III.5: Inference of dichotomous traits on targeted individuals, using effect size estimates that are: a) as originally disclosed for QC; and b) protected using our proposal. In a), cases ("red") and controls ("green") reside at the extremes of the distribution. Controls and holdouts are relatively harder to distinguish because neither contribute to the GWAS by biasing the effect size estimate away from the norm. This pattern allows us to distinguish different types of participants.

seem sensitive. For instance, several heterogeneity tests rely on effect size estimates and variance of GWAS results, which can be leveraged to recover sensitive traits [124, 77, 43]. In addition to the previously demonstrated privacy breaches of inferring disease traits and study participation status from effect size estimates (Figs. III.4a and III.5a), we show that it is problematic even to reveal the direction of the effect in GWAS (i.e., positive or negative) [77] for QC purpose. Our results of inferring study participation and exact traits are demonstrated in Fig. III.6a for continuous traits. In general, one single bit of disclosed information is sufficient to achieve high inference accuracy. This evaluation on disclosure of direction of effect is notable because otherwise one may wrongfully claim that privacy could be assured by just obfuscating or truncating the numeric precision of disclosed summaries (a common practice exercised in genome research such as in [53]).



(a) Direction of effect on original GWAS summaries.



(b) Attack on our protected direction of effects.

Figure III.6: Trait inference using direction of effect: (a) without protection, and (b) after applying our protection.).

III.3.2 Privacy-enhanced QC.

To safeguard the QC without impeding the workflow, we proposed to substitute vulnerable QC components with our secured procedures. We implemented a privacy-preserving QC pipeline as computer software. The workflow of our proposal is illustrated in Fig. III.7. The QC workflow is initiated once individual sites have derived their local GWAS results, which will serve as inputs to the pipeline. The data will go through a series of multi-tiered QC procedures to ensure data quality, for instance, to locally check data quality (i.e., site-specific QC), compare data quality with other sites (i.e., cross-site QC), and quality check meta-analysis results (i.e., post-analysis QC). The pipeline will provide an informative QC report at the end. We emphasize that the whole QC workflow here proceeds securely without leaking sensitive information.

Our privacy-enhanced QC is based on the following observations:

Firstly, for site-level QC, we note that there is no cross-site check or dependency between QC of different sites. We thus suggest that site-specific QC be performed at their

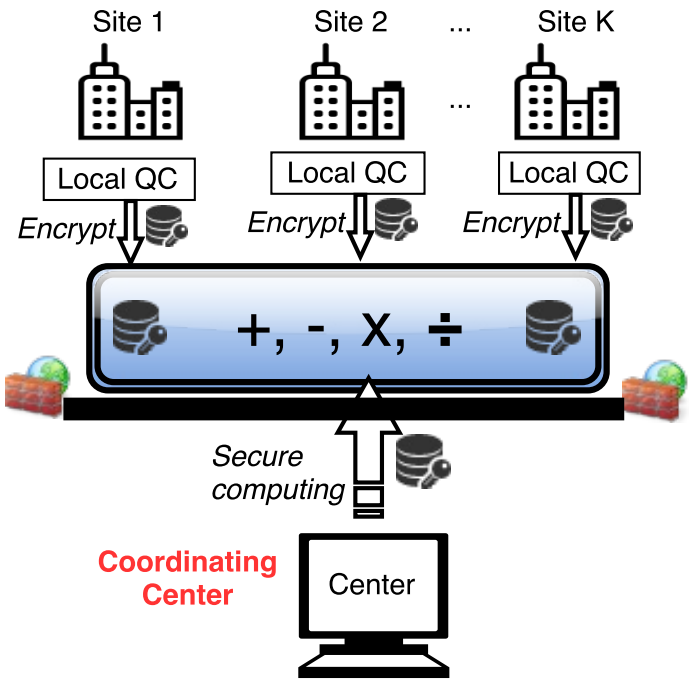


Figure III.7: Overview of our privacy-enhanced system. Each study site performs its local QC procedures, and provides encrypted diagnostic summaries to the Center. The Center performs cross-site comparison and post-analysis checks in a encrypted fashion without needing to see data content.

respective local sites where privacy issues could be circumvented. In practice, this could be achieved by disseminating standardized computer scripts for QC [153, 51, 144] to each site along with the study plan and scripts for running local GWAS.

Secondly, for cross-site QC, we note that most of the steps still do not involve cross-site contrast or dependency. The gold standards for comparison are often public information or can be easily standardized and distributed to local sites (e.g., for MAF plots, the public HapMap project acts as the public baseline; for P-Z plots the diagonal line is the gold standard). We thus suggest distributing these QC tasks to local sites for independent running.

Lastly, regarding QC results, we note that current practice does not incur privacy concerns. This is because QC seldom returns bulk of results (e.g., thousands of summaries of SNPs) to unintended entities. In QC, detailed reporting is only needed when quality issues were observed, and in those situations, only a handful few of problematic data points were

revealed to their originating sites. In addition, there are existing solutions such as differential privacy for ensuring result privacy and can be easily embedded in our system if truly necessary.

For evaluation, we apply our framework on the three multi-site meta-analyses mentioned earlier. Below we report our empirical evaluation in terms of result accuracy and system running time.

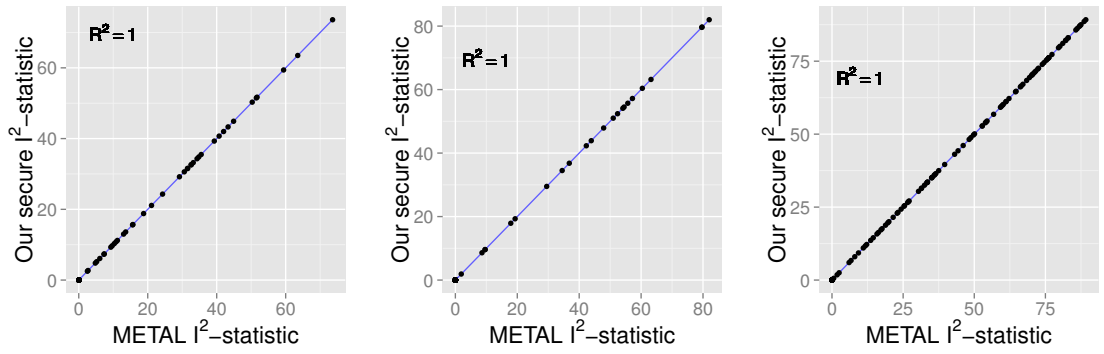
III.3.3 Accuracy of Secure Heterogeneity Tests.

Heterogeneity tests are a major component of QC and require contrasting local association summary statistics across sites. Standard practice requires sharing too detailed (and sometimes sensitive) summary statistics, which is vulnerable to privacy inference attacks. Our framework protects this process leveraging strong Cryptography-based methods (see Online Methods). Since in our secured pipeline, the protected data and computation are treated as a black box, it is essential to ensure result accuracy during evaluation so that researchers regard the results to be reliable.

We thus compare the accuracy of heterogeneity scores yielding from our secure pipeline against those from mainstream software (e.g., METAL toolkit [152]). We focus on I^2, which METAL outputs directly and can trivially translate into other heterogeneity tests. As illustrated in Fig. III.8, our results are always identical to the gold standard (with perfect correlation $R^2 = 1.0$). In fact, result accuracy is easy to prove theoretically since our secure framework replicates the exact original computation without any approximations.

III.3.4 Accuracy of Other Secure Procedures.

We also want to ensure the accuracy of other QC sub-procedures that have been upgraded and become different from existing (non-secure) practice. However, we point out that since all other safeguarded sub-procedures are still reusing previous software building blocks (albeit changes in occurence location of QC operations), correctness of QC operations is not affected.



(a) eMERGE study (b) PAGE study (c) EAGLE study

Figure III.8: Correlation of I^2-statistics between our secure result and standard METAL [152] output (numeric values represent percentages). All three empirical evaluation studies yield perfect correlations ($R^2 = 1.0$). In particular, heterogeneous SNPs (with $I^2 > 75\%$) are still identified as heterogeneous via secure implementation, and normal SNPs are also correctly labeled as normal.

Table III.2: Running time of secure heterogeneity tests.

Dataset	**# SNPs** (**# sites**)	**Local-site time** (S)	**Total time** (S)
eMERGE hypothyroidism	500k (5)	2.5	64
PAGE obesity	500k (6)	2.4	42
EAGLE diabetes	500k (2)	3.7	58

III.3.5 Computation runtime.

Naive implementation of secure systems may incur high computational overhead. Meanwhile, however, we believe careful optimizations in framework design (such as a hybrid architecture) and engineering efforts often lead to drastic efficiency improvements. In our case, we note that our cryptography-based protections introduce very minimal overhead to whole pipeline, as evidenced by running time results in Table V.6. These were evaluated on a commodity desktop with 2.4 GHz dual-core and 8 GB memory.

Also, it is important to point out that since our framework works in an automated and reusable manner (i.e., different meta-analysis studies could rely on the same infrastructure and setup), our framework often provides significant speed-up over traditional (manual) QC process. As one example, it may take months to complete even individual steps of the QC process [153].

III.4 Discussion

Our work has strong relevance to genome research for several reasons, especially given the urgency for both scientific data sharing (either for increased sample size or for reproducible research) and guaranteeing participant data privacy.

III.4.1 Implications for genome research.

With respect to participant privacy risk, we demonstrate that sharing various summary statistics that are routinely available and necessary to QC and meta-analysis tasks (both at the individual and study levels) is indeed equivalent to or only slightly safer than directly disclosing individual genomic information.

Concealment of individual-participant data and privacy protection has always been the primary, if not the sole, motivation for adopting and advocating summary statistics-based approaches towards collaborative genome research. A popular and representative example is meta-analysis (and its essential component of QC). Here instead, we prove that the promised advantage of meta-analysis is not fulfilled, and conducting meta-analysis and QC on summary statistics does not offer significant advantage over directly handling individual-participant genome information. From this perspective, meta-analysis and QC may (and should) be subject to the same regulatory frameworks governing individual-level genomic information and thus subject to the same legal risks.

Our privacy assessment has major implications for genome research, as ever-larger study consortia are being formed nationally and internationally [135, 168]. This will continue to pose significant challenges in mandating universal trust between institutions, guaranteeing sufficient maturity in security and privacy standards among all investigators, and detecting and prosecuting inappropriate disclosure of genomic information.

With respect to privacy protection, we illustrate that technological advancements can be of help for simultaneously balancing participant privacy requirements and supporting scientific workflows even as complex as QC for GWA meta-analysis. A recent work [71]

published a few years later than our current work reached similar conclusions, by showing that cryptographic methods (using Yao's garbled circuit [164]) are feasible and relatively efficient in protecting quality control pipelines. Overall this is a striking message to convey to the genetics community, as most existing works on genome privacy only demonstrate the privacy risks without providing proper solutions or advice, leaving general scientists in the misconception that privacy is dead [42] and one can only choose between data sharing or privacy.

III.4.2 Limitations.

The inference attack in our privacy assessment focused on a simplified GWAS model which did not account for covariates such as gender, age, and ethnicity. Incorporating such factors into our attack model might boost the success rate of our attacks, but only with marginal improvements since it is generally expected that the contribution of such covariates would be very small to the GWAS regression problem. So in theory, even given that correlation effects are not reported on such covariates in most published GWAS or meta-analysis, our attacks would still be successful regardless.

While we have tried our best to empirically validate our methods and claims as comprehensively as possible, we point out that an even larger-scale evaluation would still be ideal to generalize our findings and significance more broadly. Our statistical inference attacks were empirically validated using a selected collection of multi-site consortia meta-analaysis and GWAS datasets and simulated (phenotype) studies due to resource constraints. While our methods were designed to be generic and widely applicable, we yet have to benchmark our methods on a wider variety of genomic and phenotypic datasets.

Our protections and the secure QC pipeline are primarily motivated to safeguard intermediate data and analytics in the whole QC process, while incurring minimal changes to the original scientific or administrative workflows. While this makes our secure pipeline more accurate and easy to deploy in real studies, it sometimes may bring about the side

effect of privacy leaks from QC results themselves. For instance, the Q- or I^2-statistics themselves may be leveraged for privacy inferences. However, we point out that there are currently no known studies demonstrating such risks; also, since better protections for this scenario most probably would require revamping the complete scientific workflow and retaining scientifically critical information and decision making, we leave it as follow-up discussion for the general scientific community; finally, our protections could be enhanced by enforcing the concept of differential privacy on all revealed QC results. However, this would certainly deteriorate the scientific utility of the results and QC in general.

Due to computational efficiency considerations, we adopted a hybrid computing architecture by leveraging safe and faster distributed computing. We point out that in some scenarios, it may still be necessary for the consortia center to enforce central quality control on all steps, including on file-level. A a natural extension, we hope to implement a fully centralized and secure version of the pipeline.

III.4.3 Conclusion.

In this chapter, we demonstrated important privacy vulnerabilities of disclosing various summary statistics that are routine in QC for GWA meta-analysis. We further demonstrated the design and evaluation of our privacy-enhanced QC pipeline which incorporated novel and practical technical countermeasures. Empirical evaluations on various real studies confirmed the privacy vulnerabilities in traditional QC workflow. Meanwhile, our secure QC pipeline prove to support QC accurately and efficiently while guaranteeing strong privacy. We hope that our solution could alleviate privacy concerns over genome privacy and enable broader scale of collaborations and data sharing in genome research.

CHAPTER IV

SecureMA: Safeguarding Meta-analysis of Genome-wide Association Studies (GWAS)

This chapter is based on our work in [159, 158]. My contribution in this work includes conception and design of the study, implementation and experimental evaluation, analysis of results, writing the manuscript and addressing reviewer comments.

Sharing genomic data is crucial to support scientific investigation such as genome-wide association studies. However, recent investigations suggest the privacy of the individual participants in these studies can be compromised, leading to serious concerns and consequences, such as overly restricted access to data. In this chapter, we introduce a novel cryptographic strategy to securely perform meta-analysis of genome-wide association studies (GWAS) in multi-site consortia. Our methodology is useful for supporting joint studies among disparate data sites, where privacy or confidentiality is of concern. We validate our method using three multi-site association studies. Our research shows that genetic associations can be analyzed efficiently and accurately across sub-study sites, without leaking information on individual participants and site-level association summaries. In additional to the above methodology improvement, we also release our open-source software, SecureMA, for secure meta-analysis of GWAS at: http://github.com/XieConnect/SecureMA. Our customized secure computation framework is also open-source at: http://github.com/XieConnect/CircuitService.

IV.1 Introduction

Decreasing costs in sequencing technologies, in combination with large repositories of clinical information, has enabled the discovery of novel associations between genetic variants and disease. These achievements are facilitated by increased collection and reuse of genomic data [57], as well as broad efforts to obtain larger sample sizes (by sharing

and combing data) for increased statistical power [126]. Meta-analysis is a common solution for aggregating sub-study results across large consortia to achieve this goal. In fact, meta-analysis is responsible for approximately 37% of the 15,845 genome-trait associations listed in the NHGRI GWAS Catalog [151]. At the same time, the sensitive nature of genomic data has led to numerous discussions around the governance of genomic records [52, 86]. Currently, policy and advisory groups recommend removing identifying information (e.g., personal names) to uphold the privacy of study participants [102, 130].

Yet, the efficacy of such protections is increasingly being questioned [132]. Various studies demonstrate that the identity of participants, as well as sensitive information (such as disease status) can still be inferred from the shared genomic data [59, 95, 69, 78, 134, 77]. This can occur by leveraging an individual's genome sequence or the study summary statistics about associations, such as (genotype) allele frequencies and association effect sizes that would be used in meta-analysis. Most recently, it was shown that an individual's identity could be ascertained through Y-chromosome short tandem repeats (Y-STRs) using public genealogy databases on the internet [59]. Inference attacks on genetic privacy have already raised serious concerns from scientists, policy makers, and the general public. They have also led to reduced sharing of genome sequences and site-level summary statistics. For instance, based on [69], the NIH and Wellcome Trust stopped sharing aggregate genomic data directly to the public [167]. These demonstrations have also influenced proposed regulations such as [44, 45], some of which would designate all biospecimens and their derived data as identifiable [143].

To address the privacy concerns on individual genomic information as well as site-level summary statistics, we propose a practical protocol to securely perform meta-analysis of genome-wide association studies (GWAS) in large multi-site consortia (Fig. IV.1). Our protocol leverages cryptographically secure technology to provide provable security guarantees. Unlike alternative proposals [82], in our protocol, sub-study sites retain full control of their respective individual participants' data and local site analyses. This allows each

site to make appropriate adjustments to effect estimates to account for study-specific differences in design, which is pervasive in multi-site studies but not supported in [82]. Our protocol also allows sites to contribute to meta-analysis *without* exposing site-level summary statistics. Such comprehensive protections make our protocol impervious to popular privacy attacks over genomic data at both the individual- and site-level.

In this paper, we demonstrate the design and implementation of our secure meta-analysis protocol (called *SecureMA*), and provide empirical evaluations with three separate multi-site genetic association studies.

IV.2 Overview of Proposed Framework

IV.2.1 Secure Meta-analysis Protocol

The SecureMA protocol consists of two main steps: 1) Setup and 2) Secure Computation. The Setup initializes the system by: i) generating and distributing the encryption/decryption keys, ii) encrypting association statistics locally at each study site, and iii) submitting the data encryptions to the data managers (e.g., coordination centers in practice). The Secure Computation step securely performs meta-analysis over the encrypted submissions of site-level association statistics (Fig. IV.1).

IV.2.2 Setup Step of the Protocol

To setup the process, a one-time step for generating and disseminating the encryption/decryption keys is coordinated by a trusted authority who is not involved in any data management or computations (Fig. IV.2).[1] For protection purposes, the decryption key is then split into multiple shares and distributed across the participants of the protocol, as described below. By doing so, to successfully decrypt data, collaboration is required between the majority of key holders. As detailed in Section IV.4, the splitting of the key enforces an "honest-majority" to mitigate collusion for illicit decryption.

[1] Following standard practice in security for cryptographic systems, this authority generates keys and has no further interaction with any of the participants involved in SecureMA.

Optionally, to make the protocol more practical, several intermediate parties, which we call Data Managers, can be setup to host the (encrypted aggregate) data on behalf of the local sites. Following this scheme, the local sites submit encryptions of their study summary statistics (e.g., effect size and the inverse of its variance) to their entrusted data managers and can then go offline. In doing so, one manager can coordinate for several local sites, such that only a limited number of online participants are required for the protocol to proceed. And, as mentioned, enforcing an honest majority ensures no manager alone can decrypt the data. Further details on this management model can be found in Section IV.4.

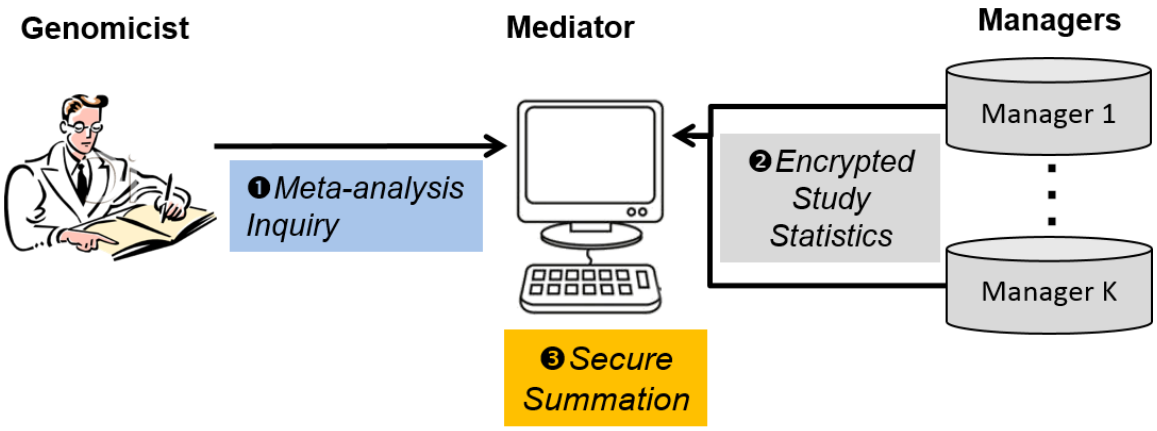
IV.2.3 Secure Computation Step of the Protocol

When a scientist issues a study inquiry to the system, encryptions of site-level association statistics are requested from the data managers and then provided to a third party responsible for coordination and computation - the Mediator - who securely sums the encrypted submissions (Fig. IV.1a).

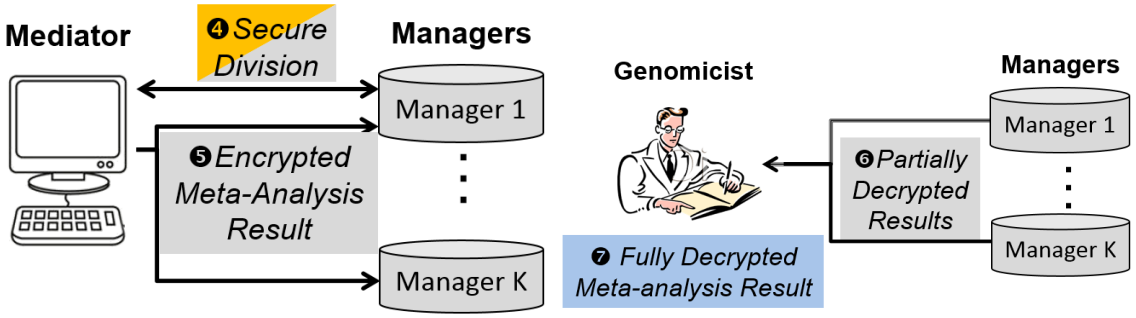
Next, the mediator coordinates with one randomly selected data manager to perform a secure division to derive the weighted average, the last operation of meta-analysis (Fig. IV.1b; details in Section IV.4.2.1).

At this point, the meta-analysis result is still in an encrypted state. The mediator is then responsible for initiating a final round of collaborative decryption by distributing the encrypted result to a majority of the trusted data managers for partial decryption (Fig. IV.1c). By collecting a sufficient number of the partially decrypted shares from the data managers, the scientist combines them to reveal the final decryption from which the final result of his study query would be derived. Thus, until the scientist requests the final decryption, no individual or site-level aggregate information is ever disclosed because all information remains encrypted throughout the protocol.

A complete activity diagram of the SecureMA protocol is provided in Fig. IV.3.



(a) Secure summation



(b) Secure division

(c) Result description

Figure IV.1: The SecureMA protocol (secure computation step). (a) The process begins when a scientist submits a meta-analysis study inquiry. Each data manager in the study submits encrypted local statistics (e.g., effect size and the inverse of its variance) to the Mediator for secure summation. (b) The Mediator then coordinates with one random data manager to securely divide the numerator by the denominator of the meta-analysis function. (c) The results of the meta-analysis are partially decrypted by the data managers, which are composed into the final full decryption of the meta-analysis p-value at the scientist's computer.

IV.3 SecureMA for privacy-preserving meta-analysis

IV.3.1 Meta-analysis

Meta-analysis [66] is a statistical technique widely-used in genetic association studies for synthesizing study results from across consortia in order to obtain larger sample sizes and gain statistical power. In this work, we focus on the fixed-effects model to perform meta-analysis [152], which yields a weighted average of the effect size (e.g., beta coefficient) using the inverse of its variance as the weight:

$$Z = \beta / se = \frac{\sum_i \beta_i w_i}{\sum_i w_i} \Big/ \sqrt{\frac{1}{\sum_i w_i}} = \sum_i \beta_i w_i \Big/ \sqrt{\sum_i w_i}\,, \tag{IV.1}$$

where β is the aggregated effect size, se is the aggregated standard error, β_i is the effect size of an association for the i^{th} sub-study (i.e., site contributing data to the meta-analysis), weight $w_i = 1/se_i^2$, and se_i corresponds to the standard error of the effect for the i^{th} sub-study.

IV.3.2 Secure Computation of Meta-analysis

To enable direct computation in a cryptographic setting, we square Equation IV.1 (i.e., Z^2) (Section IV.4.2.1). The final square root and conversion from Z-score to p-value is performed by software running on the computer of the scientist who issued the meta-analysis request.

For reference, the core (secure) computations for the proposed SecureMA protocol are summarized in Table IV.1. For each meta-analysis study, the mediator requests and receives encryptions of site-level association summaries (denoted as $E(\beta_i w_i)$, $E(w_i)$) from the data managers. Then, the mediator leverages the secure summation sub-protocol *ADD* (Section IV.4.7) to compute the sums in the numerator and denominator of Equation IV.1 without decryption (resulting in encryptions: $E(\sum_i \beta_i w_i)$ and $E(\sum_i w_i)$).

The final step of meta-analysis involves a division operation (for deriving the weighted average of effect size), where in our case, both the numerator and the denominator are

Table IV.1: The core variables and computations for SecureMA.

Notations	β_i – effect size estimate for sub-study i w_i – weight term for sub-study i $E()$ – encrypted data or secure computation
Inputs	$E(\beta_i w_i)$ – encrypted statistic for sub-study i $E(w_i)$ – encrypted statistic for sub-study i
Intermediate Computations	Summations: $E(\sum_i \beta_i w_i)$, $E(\sum_i w_i)$ Logarithms: $E(\ln \sum_i \beta_i w_i), E(\ln \sum_i w_i)$ $E(\ln Z^2) = E(2 \ln \sum_i \beta_i w_i - \ln \sum_i w_i)$ Decrypt $E(\ln Z^2)$ to obtain $\ln Z^2$
Overall Z-Score	$Z = \sqrt{\exp(\ln Z^2)}$
Overall P-value	$P = 2\Phi(-\lvert Z \rvert)$

encrypted. There is no efficient method for directly computing the division of two encryptions. Thus, we convert it into a subtraction problem which is easier to implement in cryptography, by applying a logarithmic transformation on the squared Equation IV.1 (e.g., Z^2):

$$\ln Z^2 = 2 \ln \sum_i \beta_i w_i - \ln \sum_i w_i \tag{IV.2}$$

The logarithmic transformation, $\ln x$ (where x is encrypted), is approximated using secure computation techniques and a Taylor series (Section IV.4.8). The result from this step is still in an encrypted form.

Next, secure sub-protocols for multiplication-by-constant and subtraction (e.g., the *MULC* and *SUB* sub-protocols in Section IV.4.7) are utilized to complete the rest of the operations in Equation IV.2, yielding encryption $E(\ln Z^2)$. The final Z^2 can be obtained by decrypting and computing the exponential operation at the study inquiry issuer's site.

IV.4 Technical Details and Secure Implementation

Here we provide technical details regarding the implementation of our proposed system. Section IV.4.1 provides additional figures to complement the description of the SecureMA

protocol. Section IV.4.2 introduces the details of the meta-analysis and the specific workflows associated with SecureMA. Section IV.4.3 provides additional experiments on the computational accuracy by controlling for tunable parameters associated with the protocol. Finally, Section IV.4.4 describes SecureMA in greater detail, while covering the specific technical aspects regarding how each computation is securely performed to support meta-analysis.

IV.4.1 Cryptographic Key Management and Secure Workfkow

This section provides additional figures to describe the SecureMA protocol in greater detail.

Specifically, Fig. IV.2 illustrates the Setup step around cryptographic keys in the protocol (Section IV.2.2). We emphasize that, as illustrated later in Fig. IV.3, the Key Manager who facilitates key generation and distribution is isolated from the rest of the SecureMA system and thus has no access to any data or computations. In practice, this role could be played by a semi-trusted third-party, who is outside the set of participants. For instance, the role could be assumed by a neutral organization with a good reputation in key management, a trustworthy computing module, or even a virtual party representing a distributed and secure mechanism for key generation among many protocol participants [85].

Fig. IV.3 presents the complete activity diagram of SecureMA in sequential order, including the Setup and Secure Computation steps (Section IV.2).

IV.4.2 Meta-analysis and Protocol Participants

Here, we provide additional details regarding the computation of meta-analysis, as well as the specific workflow of SecureMA.

IV.4.2.1 Meta-analysis of Genome-wide Association Studies

To simplify and increase secure computational efficiency, we try to avoid the expensive secure square root operation (in the denominator of meta-analysis equation). To do so without affecting the final output, we square aforementioned Equation IV.1 for easier implementa-

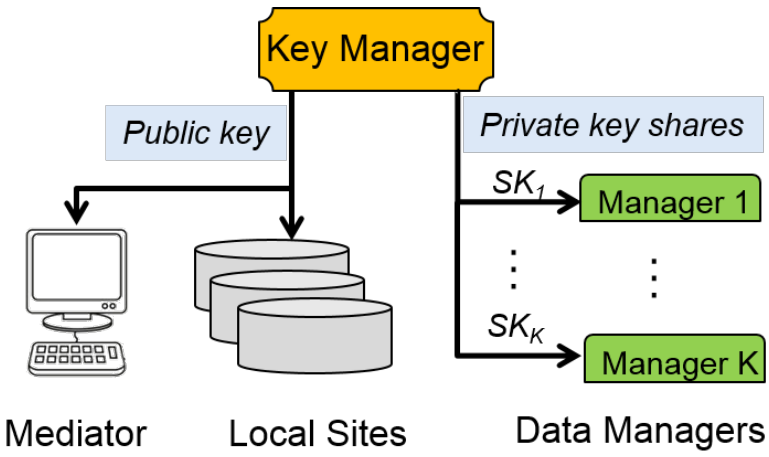


Figure IV.2: During the Setup step of the SecureMA protocol, encryption/decryption keys are generated and distributed. The public key (for encryption) is broadcast to the mediator and local sites, while the private key (for decryption) is split into secret shares (SK_1, ..., SK_K) which are securely transmitted to the respective data managers.

tion:

$$Z^2 = (\sum_i \beta_i w_i)^2 \Big/ \sum_i w_i \,, \tag{IV.3}$$

where the final square root, as well as conversion from Z-score to p-value, of the result of the meta-analysis is performed by the software running on the computer of the scientist issuing the inquiry.

IV.4.2.2 Protocol Participants

The major participants of the secure meta-analysis protocol and their roles are summarized below:

- A *Scientist* (e.g., genomicist) issues meta-analysis queries to the protocol and receives the encrypted final results which only he can fully decrypt.
- The *Local Sites* are the individual sites who collect genomic and phenotypic data, as well as conduct their local association studies.
- (Optional) The *Data Managers* (e.g., coordination centers in practice) manage the (encrypted) genomic information on behalf of *local sites*. This optional optimiza-

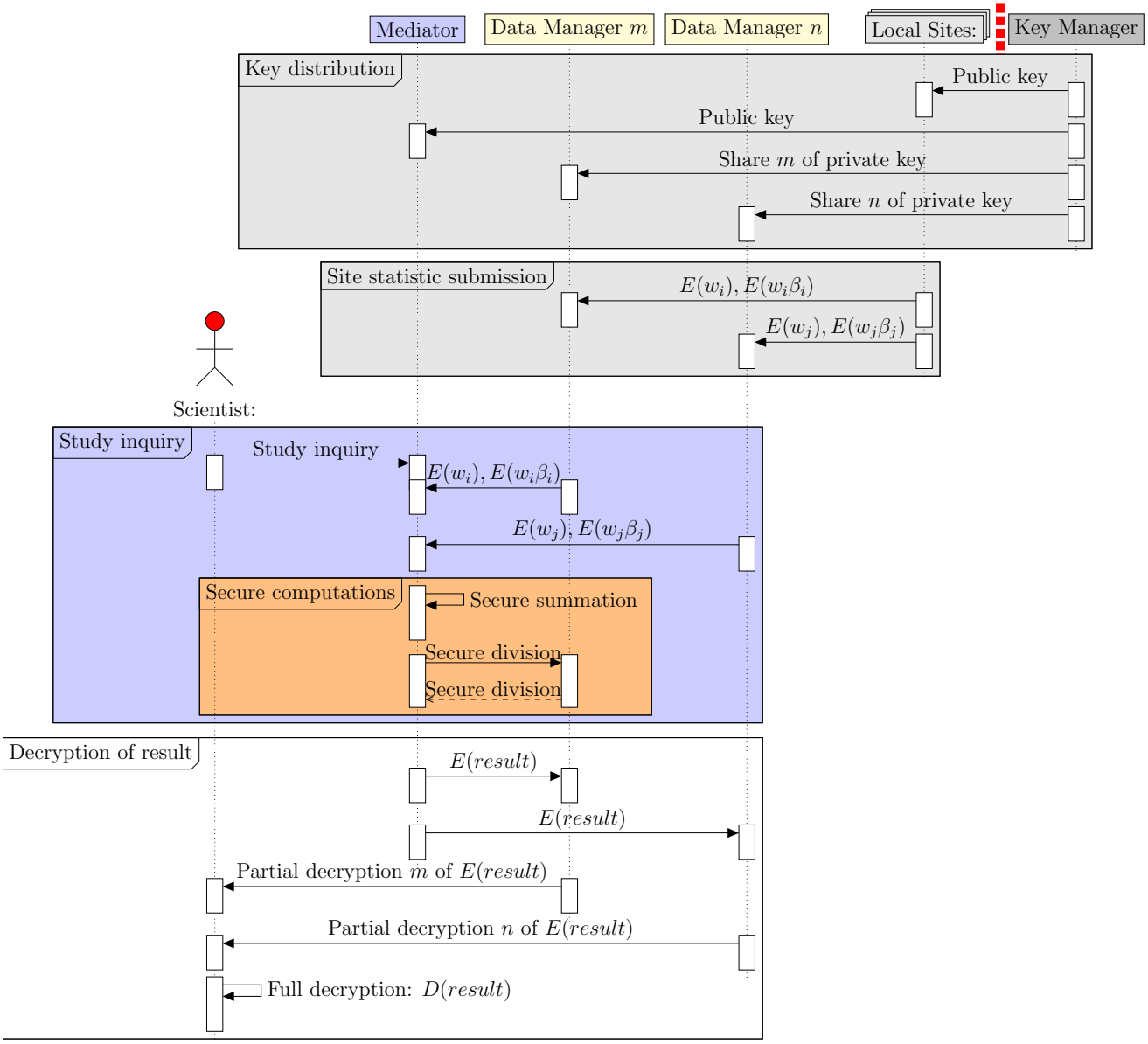


Figure IV.3: The activity diagram of the SecureMA protocol. Denoted in gray boxes is the one-time Setup step covering key distribution and submission of encrypted site statistics. In a typical running, a scientist issues a study inquiry to start the protocol, and obtains the study result in the end. In the figure, $E(data)$ and $D(data)$ correspond to the encryption and decryption of data, respectively. There can be multiple *local sites* and *data managers*. The *key manager* is isolated from the rest of the system and his only involvement is key generation and distribution.

tion makes the protocol more practical by supporting meta-analysis while reducing the number of participants required at runtime (e.g., one manager can delegate multiple local sites). The data managers only have limited decryption capabilities, as introduced later.

- The *Mediator* computes the secure meta-analysis equations and responds to the scientist's queries with encrypted results.

IV.4.3 Computational Accuracy in a Controlled Setting

As mentioned earlier, the secure computation results were close to the "true" association values (from the original publications), but not perfect. We note that in replication studies, it is not uncommon for there to be minor variability in the statistical routines performed. Thus, to present a more controlled evaluation on the computational accuracy, we performed additional comparisons with a non-secure meta-analysis as the baseline (i.e., results taken directly from the widely-used METAL software [152] instead of using the reported results from their original studies).

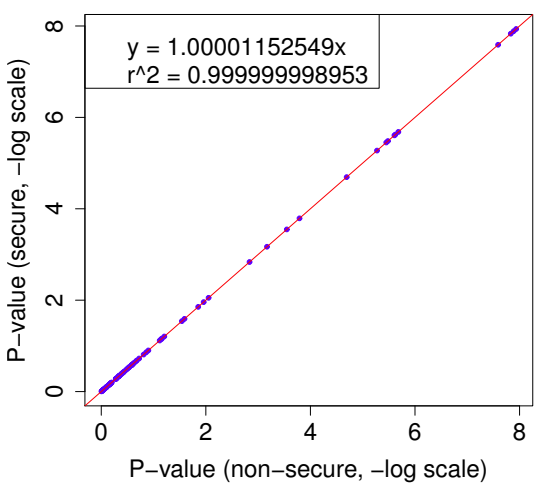
The comparisons are reported as QQ-plots on a negative logarithmic scale (Fig. IV.4). It can be seen that our secure results are extremely close to the non-secure results. Specifically, a linear regression with the y-intercept forced to zero, yielded both a slope and correlation coefficient of $\sim$1.000 for all three datasets. These results lend further evidence that our SecureMA protocol is accurate.

IV.4.4 Details on Securely Computing Meta-analysis

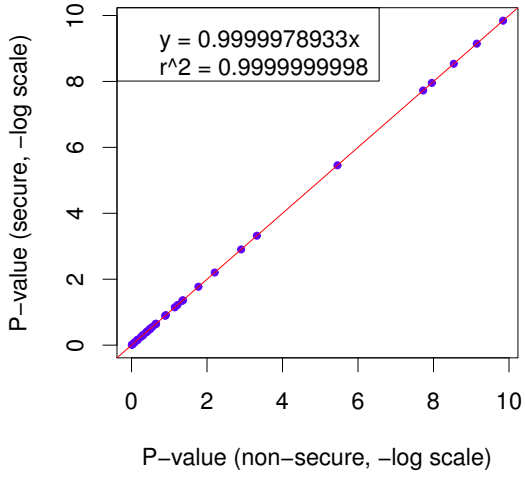
Here we provide the technical details regarding the various sub-protocols underpinning the secure meta-analysis computation.

IV.4.5 SHARES: Converting Encryptions to Secret Shares

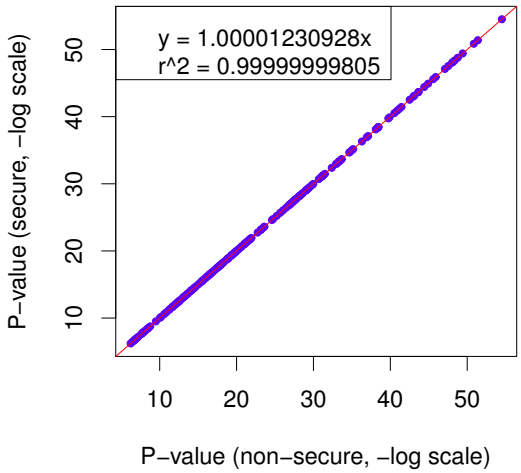
The secure logarithm protocol (a step of secure division) introduced later in Section IV.4.8 requires inputs to be in the form of secret shares, while all data in our protocol are en-



(a) eMERGE



(b) PAGE



(c) EAGLE

Figure IV.4: A controlled comparison of the P-values derived from a non-secure and secure meta-analysis protocol. These results are based on (a) 100 SNPs from eMERGE, (b) 40 SNPs from PAGE, and (c) 216 SNPs from EAGLE.

crypted using the Paillier crypto-system. We propose the following *SHARES* sub-protocol to convert Paillier encryptions into two-party secret shares (i.e., two participants collaboratively keep the secret). Given an encryption $E(x)$, the goal is to find two random values x_1 and x_2 (to be held by two participants respectively), such that $x_1 + x_2 = x$. These values are randomized to ensure it is *not* possible to predict the value of one from the other. This is accomplished as follows. First, a data manager generates a random value *rand* to obfuscate the given (encrypted) value $E(x)$ by computing $E(x + rand)$ (via the secure summation sub-protocol *ADD* introduced later). The resulting encryption $E(x + rand)$ is then transmitted to the mediator. Later, a decryption process helps obtain the mediator's data share $x_2 = x + rand$, while the data manager holds his share $x_1 = -rand$.

IV.4.6 Garbled Circuits for Secure Division

In our protocol, we leverage Yao's garbled circuit [164] to perform part of the secure division operation introduced below. As introduced in earlier (Chapter II), this approach allows two participants to collaboratively evaluate an arbitrary function on their individual data without disclosing anything other than the final output. This is enabled by implementing the function to compute as a binary circuit and the security is achieved by randomizing (garbling) the data in the circuit. We design our own function circuits and enhance the low-level FastGC framework [70] for execution. Our garbled circuit software is released open-source [157].

IV.4.7 Secure Arithmetic Operations

The Paillier crypto-system supports secure summation through an additive homomorphic property. The secure addition sub-protocol, *ADD*, is defined as follows: given two messages m_1, m_2 (and n being the Paillier field size), the encryption of sum $(m_1 + m_2)$ can be computed as:

$$E(m_1 + m_2) = E(m_1) \cdot E(m_2) \mod n^2$$

It is also straightforward to implement multiplication of an encrypted value by a known constant in the Paillier crypto-system. The multiplication-by-a-constant sub-protocol (*MULC*) proceeds as follows. Suppose we are provided with encryption $E(m)$ of message m and need to compute $E(k \cdot m)$, where k is a known constant. This can be accomplished by computing:

$$E(k \cdot m) = (E(m))^k \mod n^2$$

Secure subtraction (*SUB* sub-protocol) can be achieved by taking advantage of the multiplication-by-constant and addition protocols described above. In brief, given two encryptions $E(m_1), E(m_2)$, we can compute the subtraction as:

$$E(m_1 - m_2) = ADD(E(m_1),\ MULC(E(m_2), -1))$$

It can further be observed in Equation IV.3 that a meta-analysis requires the final division of a numerator by a denominator. However, there is no existing protocol for directly computing the division of two Paillier-encrypted numbers. We therefore choose to convert the division operation (denoted as *DIV* sub-protocol) into a subtraction problem using a secure logarithmic transformation. For simplicity, we denote: $a = \sum_i \beta_i w_i$ and $b = \sum_i w_i$. Via the logarithmic transformation, the goal in Equation IV.3 becomes:

$$\ln Z^2 = \ln \frac{a^2}{b} = 2 \ln a - \ln b \tag{IV.4}$$

We leverage the secure logarithm sub-protocol described below (Section IV.4.8) to compute $\ln a$ and $\ln b$ for the transformed division operation. The final Z^2 can be easily derived by taking the exponential, $\exp(.)$, on the final subtraction result.

IV.4.8 Secure Logarithmic Transformation

As described earlier, secure logarithmic transformation is utilized in our protocol to perform the division operation. Our $\ln x$ transformation builds upon the secure $ln(x)$ sub-

protocol in ([96]). Given input x, which is composed of secret shares x_1 and x_2 from two participants (following the *SHARES* sub-protocol), a two-phase process is applied to approximate the logarithm and output two secret shares of the result.

More specifically, x is approximated by 2^y, with a relative error of ε :

$$\ln x = \ln(2^y(1+\varepsilon)) = y\ln 2 + \ln(1+\varepsilon) \tag{IV.5}$$

Based on this representation, approximating $\ln x$ requires securely computing the two terms in Equation IV.5, which is facilitated by the two-phase process described below.

IV.4.8.1 Logarithm Phase 1: Rough Estimate via Garbled Circuits

In the first phase, $\ln x$ is approximated by 2^y using a garbled circuit evaluation to protect sensitive data. The output of this phase contains two portions, γ and α, each of which is composed of two secret shares obfuscated to prevent disclosure and is scaled up (i.e., multiplied by a power of 2 and truncated) to avoid numbers with decimals and use only integers:

$$\gamma_{true} + \gamma_{rand} = 2^N \cdot y\ln 2 \tag{IV.6}$$

$$\alpha_{true} + \alpha_{rand} = 2^N \cdot \varepsilon \tag{IV.7}$$

Equation IV.6 approximates the first term in Equation IV.5, which is a rough estimate of $\ln x$. The terms are scaled up to avoid decimal values because the computation is performed over encrypted data, which requires the operands to be integers. Here, the term 2^N is as a scaling factor, where N is the upper bound for the exponent estimate y.

Equation IV.7 denotes the scaled relative error of the approximation, and will be applied in the next phase to boost the accuracy of approximating Equation IV.5.

Since a garbled circuit evaluation involves two participants and no meaningful information should be disclosed to any single participant, we adopt random values γ_{rand} and α_{rand} contributed by one of the two participants in the computation for proper protection.

At the end of this process, one participant will hold α_{rand} and γ_{rand}, while a second participant will be in possession of α_{true} and γ_{true}, as illustrated in Equations IV.6 and IV.7.

IV.4.8.2 Logarithm Phase 2: Refined Estimate via Taylor Series

In the second phase, we further refine our $\ln x$ approximation by estimating the second term in Equation IV.5. This is accomplished via an oblivious polynomial evaluation ([117]), such that a secure polynomial from one participant is computed on the data contributed by the other participant without disclosing additional information. To perform the approximation, ε is substituted with $\frac{\alpha_{true}+\alpha_{rand}}{2^N}$ (derived from Equation IV.7). Next, we apply the following Taylor series (with proper scaling up to avoid fractional values):

$$\ln(1+\varepsilon)\cdot 2^{Nk} lcm(2,...,k) \approx \sum_{i=1}^{k} (-1)^{i-1} 2^{N(k-i)} \cdot \frac{lcm(2,...,k)}{i} \cdot (\alpha_{true}+\alpha_{rand})^i \quad \text{(IV.8)}$$

The polynomial on the right side (denoted as $Q(\alpha_{true})$) will be expanded and evaluated leveraging our *MULC* and *ADD* sub-protocols. The result at this point is still encrypted.

IV.4.8.3 Result Assembly for Logarithm

Based on the results from the previous two phases, the final result of $\ln(x)$ is obtained through an assembly process. First, the γ's in Equation IV.7 are scaled up by a factor $2^{N(k-1)} lcm(2,..,k)$:

$$(\gamma_{rand}+\gamma_{true})\cdot 2^{N(k-1)} lcm(2,\ldots,k) = y\ln 2 \times 2^{Nk} lcm(2,\ldots,k) \quad \text{(IV.9)}$$

Next, the scaled γ's are encrypted and securely summed via Equations IV.9 and IV.8:

$$\begin{aligned} &E((\ln(1+\varepsilon)+y\ln 2)\cdot 2^{Nk} lcm(2,\ldots,k)) \\ &\approx E(\ln x \cdot 2^{Nk} lcm(2,\ldots,k)) \end{aligned} \quad \text{(IV.10)}$$

After obtaining the encryptions of scaled-up $\ln a$ and $\ln b$, we can compute the scaled-

up $E(\ln Z^2)$ via Equation IV.4. The final Z-score (in decimal) can easily be derived after decryption and scaling the result back down. And the desired p-value can be obtained following the instruction in Section IV.4.2.1.

IV.5 Results

We implemented the SecureMA protocol in working software and released it open-source. To demonstrate its feasibility and practicality, we reproduced three multi-site genetic association meta-analyses. For the purposes of evaluation, we focus on the efficacy of protecting participant privacy, the computational accuracy, the running time efficiency and the sensitivity to certain protocol parameterizations.

IV.5.1 Study Data

While details of some datasets for evaluation have been introduced earlier (Section II.1). Here we re-state related information for completeness.

IV.5.1.1 The eMERGE hypothyroidism study.

The first collection of datasets is from a GWAS on hypothyroidism provided by the eMERGE consortia [36]. It consists of 6,370 study participants across five study sites, and for evaluation we analyzed 100 single nucleotide polymorphisms (SNPs) – these include the 16 statistically significant SNPs ($p < 10^{-6}$) reported in their original study and an additional 84 random SNPs for running time efficiency analysis (Section II.1). Local-site studies were adjusted for birth decade and sex following the approach described in [36].

IV.5.1.2 The PAGE obesity study.

The second collection of datasets is from a genetic association study on obesity and body mass index provided by the PAGE consortia [49]. It consists of 53,238 participants across six study sites, and for evaluation we analyzed 40 SNPs – these include the 25 statistically significant SNPs ($p < 0.05$) as identified by their original study, and an additional 15 SNPs

(Section II.1). Local-site studies were completed following the processing procedures described in [49].

IV.5.1.3 The EAGLE diabetes study.

The third collection of datasets is from a genetic association study on Type II Diabetes provided by the EAGLE group [60]. It contains 14,998 participants across two sub-studies and we analyzed 216 SNPs. The published study did not report p-values for all SNPs and, thus, for comparison, we only focus on a controlled benchmark using the standard non-secure meta-analysis as the baseline (reported in Section II.1) and running time analysis.

IV.5.2 Protection of Sensitive Information

Throughout the SecureMA protocol, the privacy of the genomic records of the individual participants is ensured. This is because the records are maintained solely at their respective local sites and are never disclosed. This resolves privacy concerns over individual genome sequences (e.g., no risk of unique identifiability based on the uniqueness of SNPs as posed by [95]).

Moreover, site-level summaries (e.g., association study statistics of each local site) are protected via strong encryption throughout the process. And the final meta-analysis results (limited to aggregate p-values only) are only made known to the inquiry issuer. Such protections make it impossible to perform inference attacks based on group statistics or allele frequencies or regression coefficients; which are features relied upon in various attacks; e.g., [69, 78, 134, 77].

IV.5.3 Accuracy of GWAS Meta-analysis Results

We compared the accuracy of our secure computations with those reported by the original studies associated with these datasets [36, 49] (EAGLE is excluded from comparison due to lack of published p-values as baseline). These results are summarized as QQ-plots of the SNP association p-values on a negative logarithmic scale (Fig. IV.5). The plots for

the eMERGE and PAGE genotype-phenotype summary statistics correspond to the 16 and 25 SNPs, respectively, that were reported as significant in the publications. To compare the secure and non-secure estimates of the p-values, we applied a linear regression with the y-intercept forced to zero. The Pearson correlation coefficient was found to be $\sim$0.998 and $\sim$1.000 for eMERGE and PAGE, respectively, implying that the secure meta-analysis yielded results directly in line with those in the original publications. The regression slopes for the PAGE and eMERGE datasets were 1.001 and 0.952 respectively, and in both cases the rank order of the significance of the SNPs was retained. These results illustrate that the secure and non-secure meta-analysis approaches produce highly consistent results.

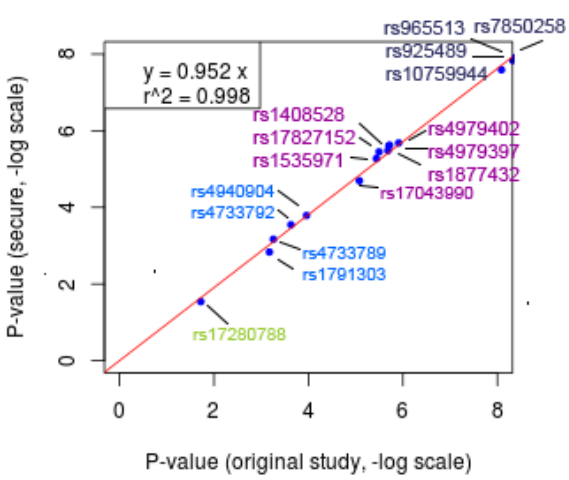
We noticed that certain original studies utilized different analysis methods (e.g., pooled analysis instead of meta-analysis) and additional data processing, which may introduce replication discrepancy. We thus performed additional controlled experiments with the standard non-secure meta-analysis as the baseline (i.e., we used the METAL software [152] to compute significance). The findings indicate our secure results are accurate, yielding both a slope and correlation coefficient of $\sim$1.000 for all datasets evaluated (Fig. IV.5c and Fig. IV.4).

Overall, these results demonstrate our secure protocol supports genetic association studies with high accuracy. Further details on how to achieve even greater accuracy can be found in the sensitivity analysis (Section IV.5.5).

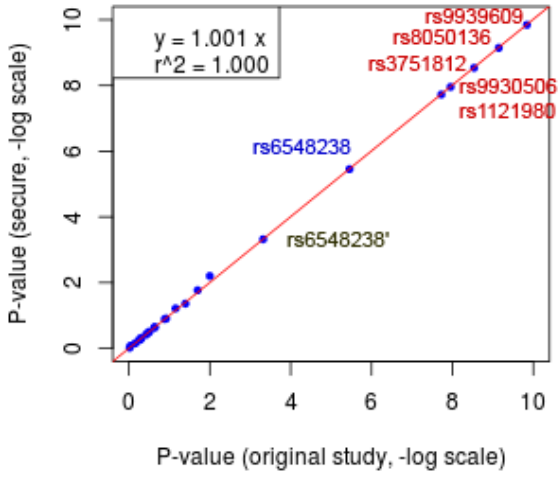
IV.5.4 Running time Efficiency

To evaluate the running time of the protocol, we performed a series of experiments on a desktop computer (2.4 GHz dual-core, 4 GB memory) running Java 1.7. We simulated the different participants of the protocol using separate system processes. All experiments were performed without parallelization to mitigate interference in the measurement of running time.

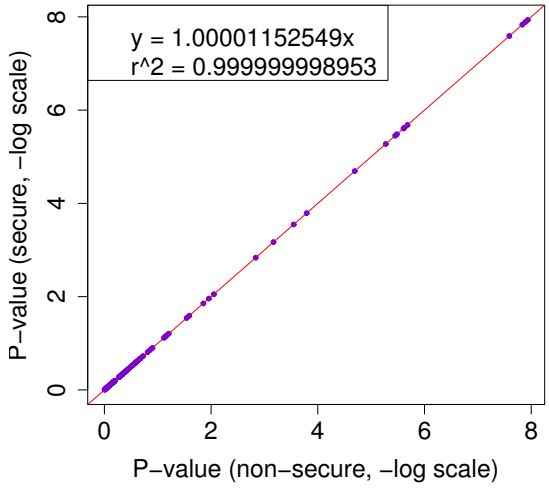
On average, the secure meta-analysis for most SNPs completed in 1.20 to 1.34 seconds



(a) eMERGE accuracy



(b) PAGE accuracy



(c) eMERGE accuracy (controlled)

Figure IV.5: Protocol accuracy. The correlation plots correspond to: (a) the p-values (secure protocol vs. original publication) based on the 16 SNPs from eMERGE; (b) the p-values (secure protocol vs. original publication) based on the 25 SNP-ethnicity pairs from PAGE (all SNPs annotated correspond to one ethnicity sub-population, except for rs6548238', which corresponds to another); and (c) the p-values (secure protocol vs. standard non-secure meta-analysis) based on a controlled comparison of 100 SNPs from eMERGE).

(with a standard deviation ≤ 0.024 seconds) and no SNP required more than 1.38 seconds (Table IV.2). In comparison to the eMERGE and PAGE datasets, the EAGLE study consumed slightly more time, due to the fact that EAGLE consists of much larger numeric values which leads to longer processing time.

Table IV.2: Per-SNP running time for SecureMA and the proportion of the time dedicated to the secure division process (mean and standard deviation in seconds).

Dataset	Total	**Division Sub-step**	**Proportion of Division**
eMERGE	1.2028 (0.0169)	1.2017 (0.0169)	0.9991 (0.0002)
PAGE	1.2148 (0.0239)	1.2136 (0.0240)	0.9990 (0.0005)
EAGLE	1.3427 (0.0164)	1.3423 (0.0165)	0.9997 (0.0003)

IV.5.4.1 Sample size.

It is important to recognize that the running time of our protocol is *weakly* dependent on the number of study participants in the study (i.e., sample sizes), because the secure computations only occur on site-level summaries[2]. This implies that our protocol can be efficient even in studies with very large sample sizes, which is common for GWAS in large consortia.

IV.5.4.2 Number of sites.

We also point out that the majority of the computation time is dedicated to the secure division of the meta-analysis (more than 99.9%), as opposed to other computations such as secure summation (Table IV.2). This indicates the protocol is scalable to a large number of data-contributing sites. Specifically, the division operation only involves the mediator and one other participant, and thus its running time is *not* dependent on the number of sites. While the running time of other computations (e.g., secure summation) may increase linearly with the number of sites, its overall running time (and increase) is negligible.

[2]Individual participant records are used by sites only for their local analyses. These are computed without encryption and, thus, the running time is negligible when compared to secure computations.

To demonstrate the scalability of our technology for large consortia, we randomly selected sites from the eMERGE dataset to simulate environments consisting of up to 100 data-contributing sites (e.g., data managers participating in the protocol). For each setting, we computed a meta-analysis for 100 SNPs (Fig. IV.6). We illustrate that even when the protocol is composed of 100 sites, the time to complete the computation is around 1.22 seconds, which is approximately the same as the initial case studies.

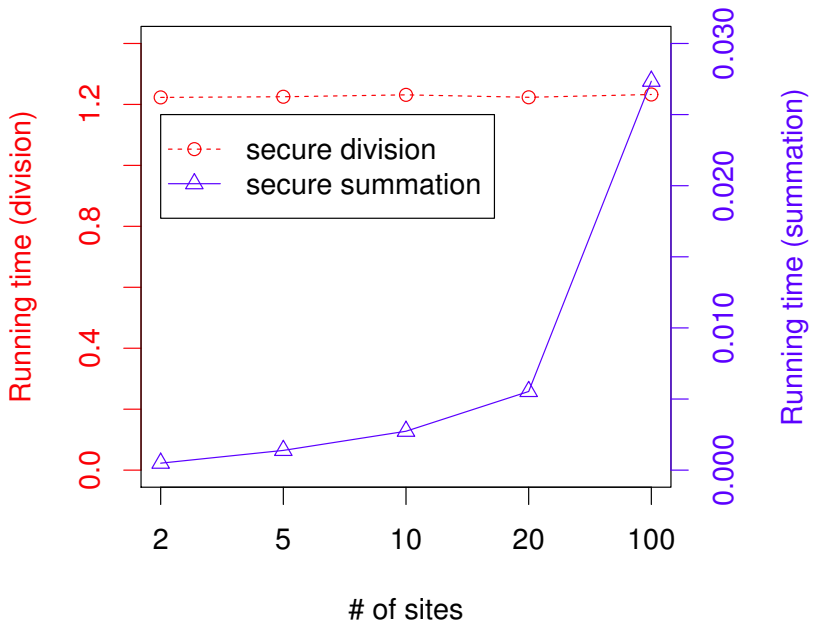


Figure IV.6: Average running time of SecureMA, per SNP, as a function of the number of sites providing data (all times reported in seconds).

IV.5.5 Sensitivity Analysis

The SecureMA protocol incorporates several tunable parameters to allow users to tune the computational accuracy and running time efficiency as necessary. These are introduced because neither decimal values, nor division over encryptions, are directly supported in cryptographic protocols. Here we demonstrate their impact both theoretically and empiri-

cally (Section IV.4.4 provides further details on these tunable parameters).

IV.5.5.1 Parameters Influencing Protocol Sensitivity.

There are three primary parameters that influence the accuracy and running time of the SecureMA protocol. These parameters were introduced due to a series of transformations and approximations to the square of Equation IV.1.

The first parameter corresponds to a scale-up factor 10^s, where the scale s is defined *a priori* by protocol participants. This is multiplied against every value submitted by the local sites. In doing so, every value is converted from a decimal to an integer.

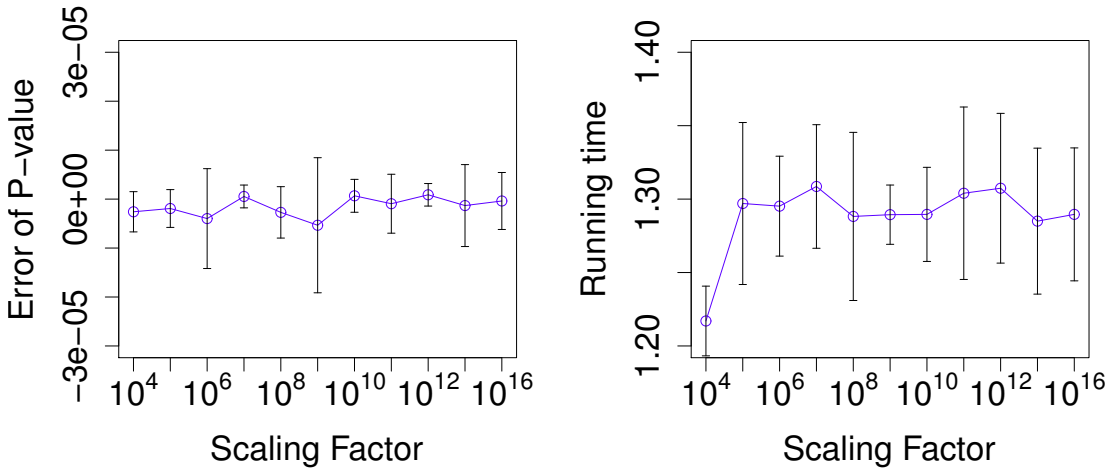
The next two parameters are associated with the approximation of secure division, which relies on the secure logarithmic transformation (Equation IV.2). Briefly, $\ln x$ can be approximated as follows:

$$\ln x \approx \frac{y\,ln\,2 \times 2^{Nk} \cdot lcm(2,\ldots,k)}{2^{Nk} \cdot lcm(2,\ldots,k)} + \frac{\sum_{i=1}^{k} (-1)^{i-1} 2^{N(k-i)} \cdot \frac{lcm(2,\ldots,k)}{i} \cdot (\alpha_{true} + \alpha_{rand})^i}{2^{Nk} \cdot lcm(2,\ldots,k)}, \tag{IV.11}$$

where integer y is a rough estimate of the exponent such that $2^y \approx x$, and additional terms such as 2^{Nk} and $lcm(2,\ldots,k)$ are for scaling purposes. The first term on the right side of Equation IV.11 obtains a rough estimate of $\ln x$ while the second term refines the previous approximation using a Taylor series.

Based on the above function, the second tunable parameter corresponds to the maximum exponent (i.e., N, or the upper bound of exponent estimate y) required to roughly estimate $\ln x$. And, the third tunable parameter corresponds to the number of expansions (i.e., k) to perform in a Taylor series when refining the accuracy of approximating $\ln x$.

For evaluation purposes, we randomly selected five significant and five non-significant SNPs from the eMERGE dataset to execute a series of secure meta-analyses.



(a) Accuracy

(b) Running time (in seconds)

Figure IV.7: Impact of the scale-up factor on (a) computational accuracy; (b) running time efficiency. Results are based on the 10 SNPs from the eMERGE dataset (mean +/- one standard deviation).

IV.5.5.2 Evaluation of the Scale-up Factor.

As mentioned, the scale-up factor 10^s is used to convert decimal values into integers. Larger factors result in the truncation of a fewer number of trailing digits and, thus, a smaller amount of information loss during computation.

Fig. IV.7 depicts how the computational error and the overall running time, respectively, of the secure meta-analysis are influenced as the factor is varied from 10^4 to 10^{16}. For context, SecureMA uses a default value of 10^8.

In Fig. IV.7a), it can be seen that, in general, the computational error of the p-value decreases (approaching 0) as the scale-up factor increases. Overall, the absolute and relative errors are always bounded within the range $[-3.0\times10^{-5}, 8.2\times10^{-6}]$ and $[-0.03\%, 0.01\%]$ respectively. However, we note there are several outlying points in the graph, such as at 10^6 and 10^9. We note that these occur because, at times, the error of the two logarithms in Equation IV.2 diverge in opposite directions, which results in a magnification of the total error.

Nonetheless, in Fig. IV.7b) it can be seen that the variance of the overall running time is relatively small as the scale-up factor increases. This is an expected result because the change of the scale-up factor has limited influence on the secure division operation, which is the most time-consuming process in the protocol.

IV.5.5.3 Evaluation of the Maximum Exponent of the Logarithm Approximation.

The secure logarithmic transformation (i.e., $\ln x$ where x is encrypted) involves two phases to the approximation. The first phase aims to find an optimal integer exponent to roughly estimate the number x. The maximum exponent we analyze in this section corresponds to the upper bound for the exponent estimate. The second step corresponds to the application of a Taylor series, which we discuss in further depth below.

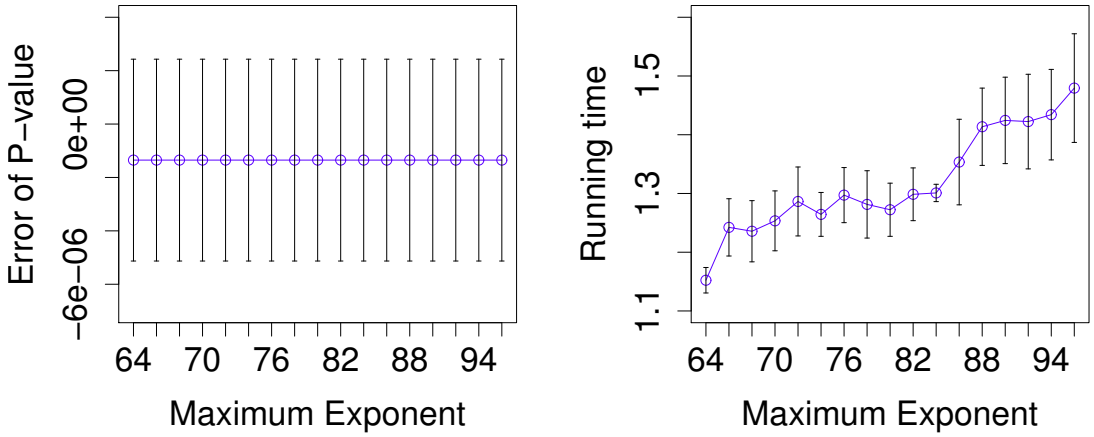
Fig. IV.8 shows how the computational error and the overall running time, respectively of the secure meta-analysis (per SNP) are affected as the exponent varies from 64 to 96. For context, SecureMA uses a default value of 80.

It was expected that a larger exponent would yield better approximation accuracy, with a trade-off in a longer running time. It is confirmed that the overall running time changes almost linearly with the increase of the maximum exponent (Fig. IV.8b). However, it can be seen that the computational accuracy is almost identical across all test cases (Fig. IV.8a). This is because, in this particular scenario, the other two protocol parameters are the dominating factors regarding computational accuracy.

IV.5.5.4 Evaluation of the Number of Steps in the Taylor Series.

A Taylor series is applied in the second phase of the secure logarithm sub-protocol to boost the approximation accuracy. Fig. IV.9 shows how the computational error and the overall running time, respectively, of the secure meta-analysis is affected as the number of steps in the series varies from 6 to 12. For context, SecureMA uses a default value of 10.

Fig. IV.9a illustrates that the more steps in the Taylor series, the better the computational accuracy is on average. Fig. IV.9b further demonstrates that there is a slight linear



(a) Accuracy

(b) Running time (in seconds)

Figure IV.8: The impact of the maximum exponent on (a) computational accuracy and (b) running time efficiency. The results are based on 10 SNPs from the eMERGE dataset (mean +/- one standard deviation).

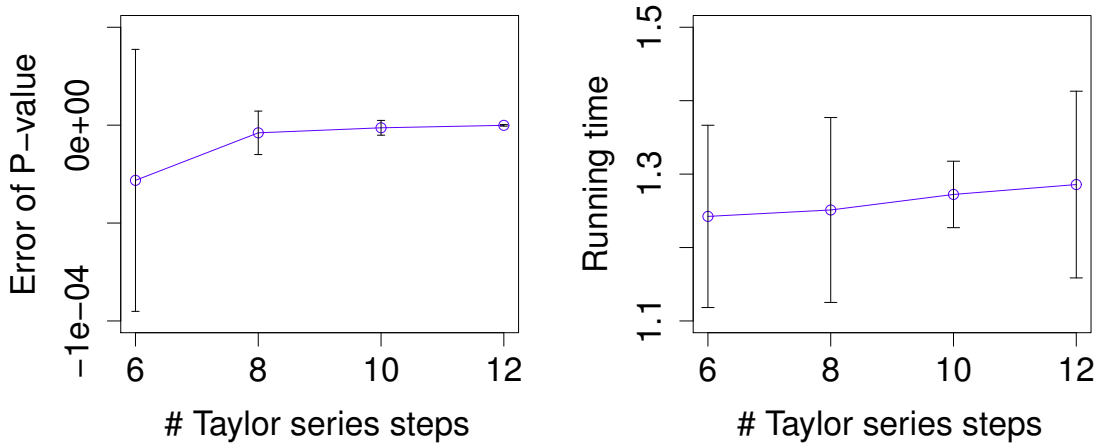
increase in the running time as the number of steps in the Taylor series grows. This result stems from the fact that the number of terms required to compute in secure computation is increasing, which causes a longer running time.

IV.6 Discussion

IV.6.1 Analysis on GWAS Scale

As discussed earlier, one of the benefits of the SecureMA protocol is that its running time has only a weak dependence on the sample size. As a result, it can be efficient for studies run over very large consortia. This is a notable improvement over alternative cryptographic proposals such as [83, 82] whose running time is positively correlated, in a linear and sometimes exponential manner, with the number of study participants and sites.

At the same time, the SecureMA protocol can be made more efficient to support analysis on a genome-wide scale (e.g., millions of association tests). First, the SecureMA protocol can easily be run in parallel on large computer clusters or cloud computing servers



(a) Accuracy

(b) Running time (in seconds)

Figure IV.9: The impact of the number of steps in the Taylor series (i.e., k in Equation IV.11) on (a) computational accuracy and (b) running time efficiency. The results are based on 10 SNPs from the eMERGE dataset (mean +/- one standard deviation).

because each SNP can be analyzed independently. Thus the total computation time for a large-scale GWAS would be inversely proportional to the computing resources allocated. As a rough estimate, a GWAS on 2,000,000 SNPs would require around 10 hours on sixteen 8-core computers without further optimization. Second, from a scientific perspective, it might be permissible to disclose the aggregate effect size of meta-analysis (i.e., the numerator in Equation IV.1). In such a scenario, the time-consuming secure division operation could be avoided entirely, reducing the overall running time per SNP to milliseconds. Third, recent advances in the optimization of secure computations such as [8, 67] may be ready to transition into practice in the near future. This could allow for certain SecureMA sub-protocols, such as secure division, to be run on parallel computing frameworks and make significant gains in efficiency.

IV.6.2 Limitations

There are several limitations to the SecureMA protocol as currently designed. First, SecureMA assumes that study data has already been carefully cleaned and subject to rigorous quality control (QC) (e.g., deposited data in dbGaP [106]). To support more dirty data in the wild, it will be necessary to embed QC processes for meta-analysis in the protocol [153]. Certain procedures may be vulnerable to attacks on privacy, but those which are based on standard algebraic computations should be translatable into secure computations. At the same time, it should be noted that many procedures can be directly applied in the clear because they do not violate privacy (e.g., file-level QC and SE-N plots in [153]). Since QC is a relatively independent and large pipeline, we leave it for future disucssion in a separate chapter (Chapter III).

Second, the current SecureMA implementation relies on a trusted authority to generate cryptographic keys, which sometimes may not be desirable (alternative solutions are in Section IV.4.1).

Third, in situations when *individual-level* genomic records need to be processed, it will be necessary to pair secure data management technologies with effective societal controls (e.g., use agreements and mandated limits on investigator behavior) that deter misuse and limit the extent to which genomic information can be abused and cause harm to people (e.g., expansion of laws to prevent utilization of genomic data in life insurance eligibility and support for long term care [6]).

IV.6.3 Alternative Methods to Maintain Genomic Privacy

To provide context for the contributions of the SecureMA protocol, we take a moment to review other recent developments in the field. There are generally two categories of data protection mechanisms that have been proposed to maintain participant privacy while supporting scientific investigations on genomic data. From a societal and regulatory perspective, it has been suggested that research participants consent to the risk of being re-identified

[105] (which may bias participant recruitment), while users of such data contractually agree not to attempt to re-identify the participants [141]. We believe such mechanisms can lower risk and, while data use agreements assign liability, they do not provide any technological deterrent and can only be enforced when violations could be detected.

On the other hand, various technological techniques have been proposed to promise genomic privacy. These include encrypting genomic sequences and supporting simple queries [83], obfuscating raw (short) genome sequences and allowing for retrieval [9], splitting regression analyses into local-site computations and center-level aggregation [154], and hosting participant-level genomic data using a cryptographic technique and facilitating genetic association studies [82]. The two approaches most similar to ours are hampered by practical limitations. First, the work [154] may leak sensitive information because local sites inappropriately disclose intermediate summary statistics during computation [41]; The other recent proposal [82] fails to account for site-specific covariates and other data preprocessing within sites, which is a common practice for multi-site genetic association studies. Their solution may also suffer from computational scalability and network communications issues in studies with large sample sizes because all individual genomic data must pass through, and be analyzed by, every server.

IV.6.4 Conclusion

This work illustrates that the privacy of individual participants, and site-level summary statistics, in genetic association meta-analysis can be guaranteed without sacrificing the ability to perform analysis that use shared data. Our proposal, SecureMA, is useful for running joint studies over disparate data sites in large consortia, where privacy or confidentiality is a concern. If appropriately implemented, our approach can prevent privacy intrusions posed by the attacks published to date. While there are opportunities to make this protocol more efficient and to incorporate quality control measures, we believe it is possible to enable much broader analytic access to genomic data for the purposes of effect

estimation and statistical association via meta-analysis.

CHAPTER V

Privacy-preserving Regression Analysis and Efficiency Optimizations in Distributed Collaborative Studies

The previous chapters mainly focused on the privacy risk and protection of genomic data and several classical statistical methods. In the following chapters, we expand our scope, and focus on more general methods of statistics and machine learning which are increasingly popular in scientific research. Such models are the foundation of many scientific disciplines, including genomics and biomedicine in general [94, 27, 170, 25, 166], social sciences [165], and physical sciences [26].

Among these, regression analysis is perhaps the most utilized statistical and machine learning methods in various domains. Representative tasks include linear and logistic regression [20, 149, 155], feature selection and regularization, and so on. It is increasing popular to conduct collaborative regression studies among disparate or federated organizations, by collectively aggregating large sample sizes and reaching reliable conclusions [87, 159, 63]. However, to enable and coordinate such multi-institution collaborative studies, serious privacy concerns around human subject data remain one of the biggest hurdles [124, 40, 64, 122]. This subfield is often referred to as privacy-preserving distributed machine learning or data mining, or secure federated machine learning [87].

Without loss of generalization, this chapter focuses on (regularized) logistic regression, a representative and relatively complex regression model with wide adoption in practice. It is also straightforward to extend our work to the generalized linear models (GLM) or apply to simpler models such as linear (ridge) regression.

We tackle this challenge of safeguarding collaborative regression analysis on two fundamental aspects: privacy/security protection and computational efficiency. In particular,

- We propose a secure and efficient framework for privacy-preserving regularized lo-

gistic regression (Section V.1) [93]. This safeguards a novel application that has not been addressed before and also provides the first practical secure implementation for logistic regression.

- We propose to tailor numerical optimization methods for privacy-preserving logistic regression, which drastically improves computational efficiency and makes secure protocols more practical (Section V.2) [161]. This provides a contrasting perspective and significantly differentiates with the common practices in data security and privacy research.

- We propose a novel paradigm for privacy-preserving logistic regression, by leveraging local-institution models (Section V.3) to accelerate the performance. This is a different approach than the traditional distributed machine learning-based formulation common in the community.

The approaches we propose here are generic and widely applicable to privacy-preserving distributed machine learning in general.

V.1 Safeguarding Regularized Logistic Regression

This section is based on our work [93]. My contribution in this work includes conception, design and supervision of the study, implementation and experimental evaluation, analysis of results, writing the manuscript and addressing reviewer comments.

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, and network analysis.

V.1.1 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 [150, 90] 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 ge-

nomics (e.g., genome-wide association studies, or GWAS [147], gene-gene interaction detection [128]), epidemiology (e.g., [33, 154]), social sciences [129, 88], and information technology (e.g., computational advertising on the internet [131] and personalized recommender systems [31]).

Many of the aforementioned disciplines and applications rely on huge numbers of data records (i.e., large sample sizes) 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) [123]; Genetic information of humans are also deemed highly sensitive [43, 159] and partially covered by the Genetic Information Nondiscrimination Act (GINA) [73]; in the education domain, students' data privacy is strictly regulated under the Family Education Rights and Privacy Act (FERPA) [32]. In other domains, failing to respect data privacy and misuse of personal information has even outraged users [18] and raised awareness of regulators [47], as in the case of targeted internet advertising. Meanwhile, various high-profile data breaches [1, 2] 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 [140] or differential privacy [38]), cryptography-based methods (e.g., encrypting genetic data [83]), or distributed computing (e.g., private records residing at local institutions only [159, 154]). Increasingly, such protections prove to be insufficient, due to various privacy attacks [69, 43, 139, 124, 41] 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, and variance-covariance.

Our work here studies the data privacy issues in regularized logistic regression [90]. Regularized logistic regression is widely used in various domains, and is often the preferred model of choice over standard logistic regression in practice [90, 10, 128, 108]. Despite its popularity, it has received little investigation from a data privacy and security perspective. The work in this chapter 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 [56], 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 [154]. 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.

V.1.1.1 Contributions.

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 [154] for our problem of focus, enhance and extend its privacy protection leveraging strong cryptographic techniques [136]. 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.

V.1.1.2 Outlines.

This section is organized as follows: in Subsection V.1.2, background information on regularized logistic regression and Newton method is provided; then, I present the method details in Section V.1.3; This is followed by experimental results in Section V.1.5; we conclude in Section V.1.6.

V.1.2 Preliminaries

V.1.2.1 (Regularized) Logistic Regression.

Logistic regression [150] is a probabilistic model for predicting binary or categorical outcomes through a logistic function. It is widely used in many domains such as biomedicine [20, 128, 33, 170, 155], social sciences [129, 88], information technology [131, 31], and so on. Briefly, logistic regression is of the form:

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

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 β is the regression coefficients we want to estimate.

In this work, we focus on regularized logistic regression with the ℓ_2 norm [90], i.e., with the regularization term equal to $\frac{\lambda}{2}||\beta||_2^2$, where λ is the regularization parameter and β is the regression coefficients (note that incorporating other regularizations such as the ℓ_1 norm is also possible).

V.1.2.2 Newton-Raphson Method.

A common way to estimate the (regularized) logistic regression model (i.e., to obtain β coefficients in Equation V.17) is through the Newton-Raphson, or iteratively reweighted least squares (IRLS) method [58, 114]. The repeated Newton-Raphson method adopts an iterative refinement process that eventually converges to the "true" values of the β coeffi-

cients.

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

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

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

V.1.3 Privacy-preserving Regularized Logistic Regression

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.

V.1.3.1 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 V.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 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 [154, 156] for other analytical tasks and prove successful in practice.

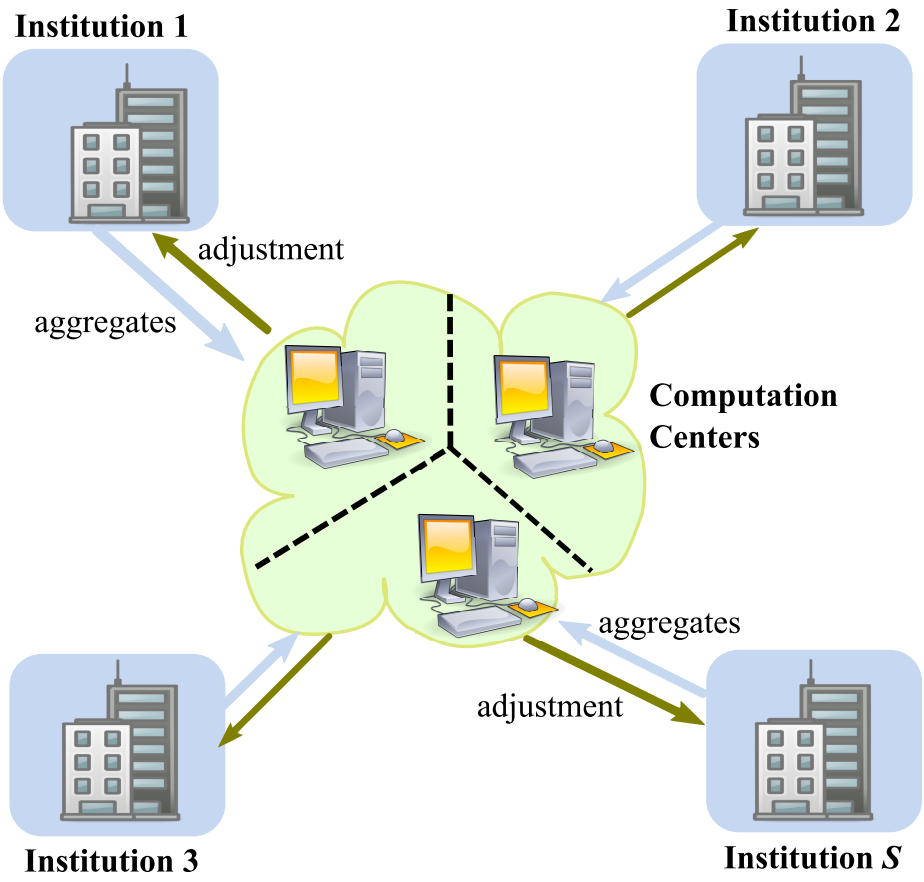


Figure V.1: Overview of our secure framework for regularized logistic regression. Each institution (possessing private data) locally computes summary statistics from its own data, and submits encrypted aggregates following a strong cryptographic scheme [136]. The Computation Centers securely aggregate 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.

Without delving into technical details, we first introduce our framework as illustrated in Fig V.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 V.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 [136] 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 (Equation V.2) to obtain a globally consistent β. In addition, model convergence checks will also be securely performed. The new β (i.e., denoted as *adjustment* in Fig V.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.

V.1.3.2 Newton-Raphson Method for ℓ_2-regularized Logistic Regression.

Our framework (Fig V.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 (Equation V.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 i^{th} 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, Equation V.2 becomes:

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

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 (Equation V.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., [118]).

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., [118]) do not seem practical especially for large studies. Such limitations have been illustrated in subsequent studies even on much simpler analytical tasks [120].

V.1.3.3 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 [120]. The partitioning of Newton-Raphson method has proven successful on other simpler tasks [154] than ours.

To accelerate the Newton-Raphson method (Equation V.3), we observe that the computations of $\mathbf{H}(.)$ and $\mathbf{g}(.)$ in Equation V.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:

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

and

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

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) [150] can also be decomposed similarly, since it depends on the log-likelihood which can be regarded as a series of sums.

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

where $L(\beta)$ 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.

V.1.3.4 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 $\mathbf{g}_j$ (Equations V.4 and V.5) using their own data. Local deviance test dev_j can also be computed similarly (Equation V.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 [139, 124, 159]), 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 [136] (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.

V.1.3.5 Centralized Aggregation.

Once the distributed computation is completed, the subsequent phase of centralized computation (Algorithm 3) would follow. As the first step, the Computation Centers will aggregate the respective (secret-share-protected) data submissions in a secure way. This pro-

cess 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 β^{old} estimate and check for model convergence afterwards. If the model is still not converged, then the updated β^{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) β^{old}; Penalty parameter λ
Output: New regression coefficients β^{new}

1: **while** model not converged **do**
2: Compute summary statistics on local institutions: $SecureLocal(\beta^{old})$
3: Securely aggregate on Computation Centers: $\beta^{new} = SecureCenter(\beta^{old}, \lambda)$
4: Check for model convergence
5: $\beta^{old} = \beta^{new}$
6: **end while**
7: Return coefficient β^{new}

Algorithm 2 $SecureLocal(\beta^{old})$: securely compute summary statistics on local institutions.

Input: Regression coefficient (of previous iteration) β^{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 $\mathbf{H}_\mathrm{j}$
3: Compute local gradient $\mathbf{g}_\mathrm{j}$
4: Compute local deviance dev_j
5: Protect $\mathbf{H}_\mathrm{j}$, $\mathbf{g}_\mathrm{j}$, dev_j via Shamir's secret-sharing
6: Securely submit $\mathbf{H}_\mathrm{j}$, $\mathbf{g}_\mathrm{j}$, dev_j secret shares to many (independent) Computation Centers respectively
7: **end for**

V.1.4 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.

Algorithm 3 *SecureCenter*(β^{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 β^{old}; Penalty parameter λ

Output: Updated regression coefficient β^{new}

1: Securely aggregate Hessian: $\mathbf{H} = -\sum_{j=1}^{S} \mathbf{H}_j - \lambda\beta^{old}$
2: Securely aggregate gradient: $\mathbf{g} = \sum_{j=1}^{S} \mathbf{g}_j - \lambda\beta^{old}$
3: Securely aggregate deviance: $Dev = \sum_{j=1}^{S} dev_j$
4: Securely compute β^{new} via Newton-Raphson method
5: Return coefficient β^{new}

V.1.4.1 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 [154, 159].

V.1.4.2 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 [69] relies on certain genomic summaries of case/control groups; it has also been analyzed in [139, 124, 41] 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 [156]) . 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 [139, 124, 159].

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 [136] (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 [120, 156, 41], however, mostly only on simpler tasks (e.g., ridge linear regression, standard logistic regression, etc) than ours. In a more related work [156], 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.

V.1.4.3 Shamir's Secret-Sharing for Protecting Data.

In our protocol, we leverage Shamir's secret-sharing [136] 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):

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

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 $(1,q(1)),(2,q(2)),...,(t,q(t)),...,(w,q(w))$. 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 [136] 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.

V.1.4.4 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 [139, 124, 159]. Feasibility of existing attacks rely on both Hessian and gradient. 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" strat-

egy 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 [120]. A fully secure version is implemented as the baseline in our later work [161].

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 [13] for a detailed security proof.

V.1.4.5 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 $\mathbf{X}_j$, responses $\mathbf{y}_j$ for Institution $j \in S$.

1: Generate coefficients $\beta \in \mathbb{R}^d$ 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 $\mathbf{X}_j = \left[\begin{array}{cc} 1 & \mathbf{cov}_j \end{array} \right] \in \mathbb{R}^{N_j \times d}$
5: Calculate probabilities $\mathbf{p}_j = 1/(1+e^{-\beta^T \mathbf{X}_j}) \in \mathbb{R}^{N_j}$
6: Generate and output response variables $\mathbf{y}_j \in \mathbb{R}^{N_j}$ from $Bernoulli(\mathbf{p}_j)$
7: **end for**

V.1.5 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, computa-

tional efficiency, as well as scalability to large studies.

V.1.5.1 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. While specific simulation parameters do not matter in our case, for demonstration purpose, I generated coefficients uniformly from range -5 to 5, and covariates from Gaussian distribution with mean of 0 and standard deviation of 1. This dataset consists of 1 million records spanning 6 features from 6 institutions, which is quite representative for most real-world use cases.

- The **Insurance dataset** [145] 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 [97]. 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.

V.1.5.2 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 V.2,

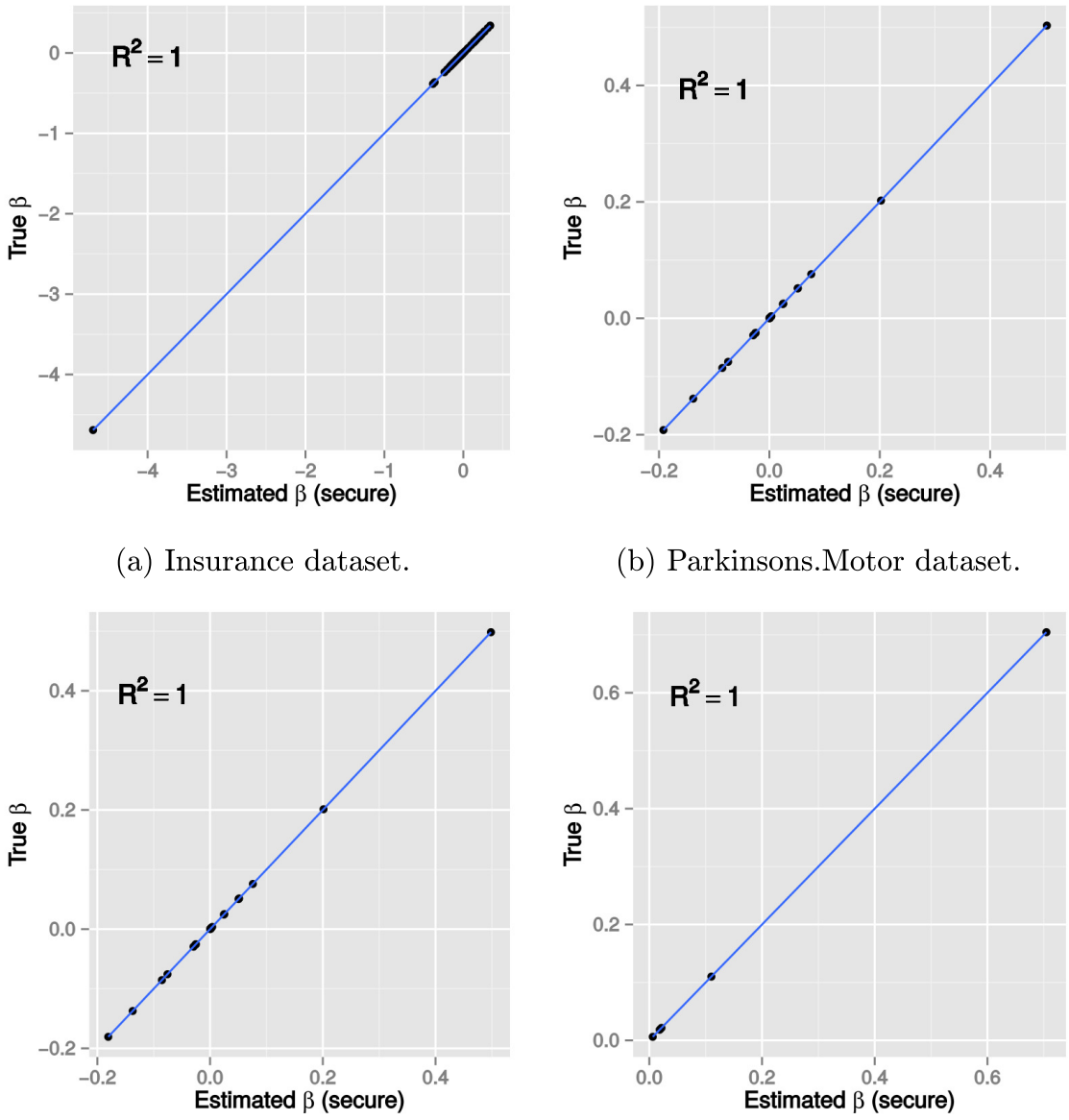
our framework yields identical results to the expected ground truth across all evaluations (with correlation $R^2 = 1.00$). The result accuracy is also evidenced by the mathematical proof explained earlier, where we have shown that our distributed model estimation method follows an exact derivation and no approximation is involved in the secure computation procedures.

V.1.5.3 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 V.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 [61] 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 [120] (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



(a) Insurance dataset.

(b) Parkinsons.Motor dataset.

(c) Parkinsons.Total dataset.

(d) Synthetic dataset.

Figure V.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 $R^2 = 1.00$.

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

Table V.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

iterations, as evidenced by Fig V.3. Across all evaluation datasets, the models converged within $6 \sim 8$ 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. We might see minor variance in the iterations to converge, depending on the difference of the input datasets and data simulation parameters. However, this is out of scope for this work since these are agnostic of our cryptographic protections and our conclusions are not affected.

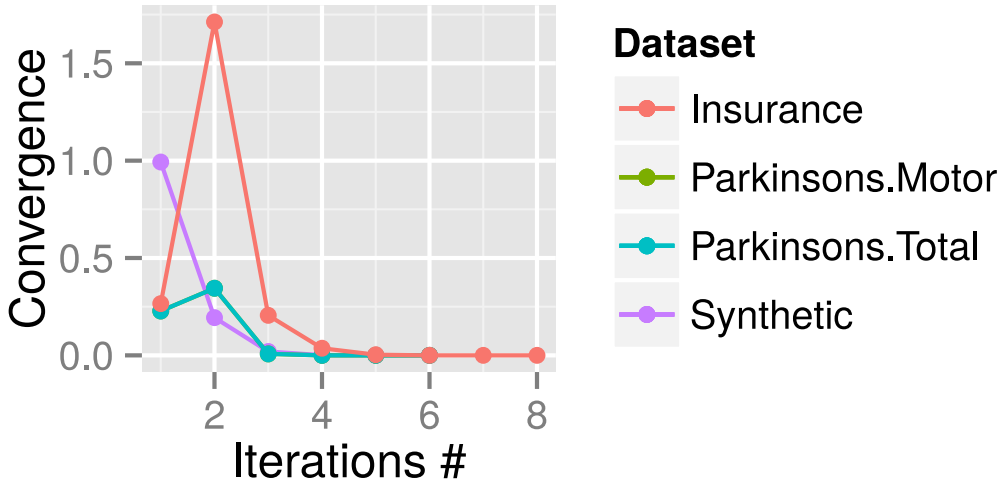


Figure V.3: Model convergence (i.e., deviance) for all datasets (deviance smaller than the threshold indicates convergence). All models converged within $6 \sim 8$ iterations. Note that the convergence scores for the Parkinsons.Motor and Parkinsons.Total studies almost overlap due to their high similarity in the plot.

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 V.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.

V.1.5.4 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 [73], the Patient-Centered Outcomes Research Institute (PCORI) [135], and financial systematic risk protection [3].

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 first generate a large-scale synthetic dataset (Algorithm 5, and then simulated multi-insitutional studies with up to 100 institutions by randomly partitioning the dataset by rows (thus each subset rows blong to an institution). We reported the results in Fig V.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).

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)

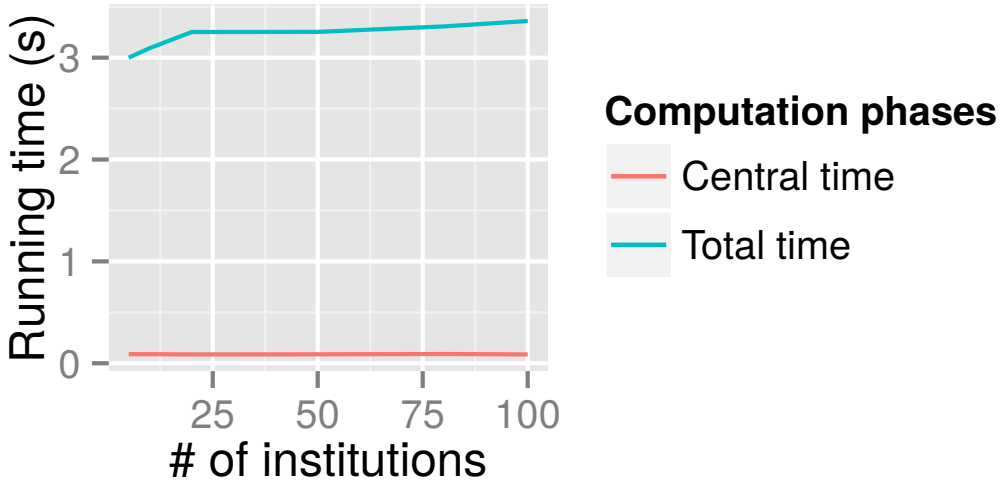


Figure V.4: Running time (in seconds) for the central phase and total computation respectively, as the number of participating institutions increases. Negligible time fluctuation is present, especially for the central (secure) computation.

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.

V.1.6 Discussion

The proposal presented in this chapter works even when data are imbalanced among different institutions. This is because our model updating/fitting are (securely) conducted at a global central server, ensuring that minor data (imbalance) noise will still be dominated by desirable true signals. 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 [89]); 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 [136], 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 [159, 120, 41]). 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 [120], which directly solves the linear system in secure centrally. Other secure solutions [61, 118, 41], 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 [154, 156, 159] 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 [154] directly expose summary data from model fitting, leading to serious privacy concerns over inference attacks on intermediate data [139, 124, 159, 41]. While preliminary efforts have started to gather around protecting institution-level summary information (especially regarding logistic re-

gression), existing protections seem quite weak. For instance, the obfuscation protection in [156] 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. In addition to cryptographic solutions, there have been recent works (some developed for different machine learning models than ours) based on information-theoretically secure methods [34] or secure hardware [24]. However, since this is out of scope for our work and these works were published much later than ours, we ommit further discussion here. Another popular research direction in privacy-preserving logistic regression leverages non-cryptographic approaches, such as the classical k-anonymity model [140] or differential privacy [38]. One notable example is the ε-differentially private logistic regression [22], 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 [120] 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 [154]. The distributed process makes our secure protocol for regularized logistic regression highly efficient compared to a straightforward centralized implementation [41]. Thirdly, we protect

intermediate data and computations with stronger cryptographic schemes [136], 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 [159], ours is the first to safeguard regularized logistic regression regarding intermediate data. Among the two closely related works, [154] failed to provide any protection over summaries; And [156] 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 [140] or differential privacy [38]) on the data or computations, thus maintaining accuracy of the predictive model.

V.1.6.1 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.

V.1.6.1.1 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 [10, 138], gene-gene interactions [128], GWAS [101], etc. Other biomedical studies such as prediction of adverse drug reactions [100] 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 [159]. We envision that our framework can provide an automated and privacy-preserving solution for supporting such collaborative investigations.

V.1.6.1.2 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 [112] 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.

V.1.6.1.3 Large-scale Network Analysis.

Many important innovations involve analysis of social network data, such as [109, 88, 5]. 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.

V.1.7 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.

V.2 PrivLogit: Efficient Privacy-preserving Logistic Regression by Tailoring Numerical Optimizers

The section is based on our work [161]. My contribution in this work includes conception, design and supervision of the study, implementation and experimental evaluation, analysis of results, writing the manuscript and addressing reviewer comments.

Safeguarding privacy in machine learning is highly desirable, especially in collaborative studies across many organizations. Despite popularity, existing cryptographic solutions for privacy-preserving distributed machine learning incur excess computational overhead, partially due to naive adoption of mainstream model estimation algorithms (such as the Newton method) and failing to tailor for secure computing-specific characteristics. Here, we present a contrasting perspective on designing numerical optimization method for cryptographically secure settings. We introduce a seemingly less-favorable optimization method that can in fact significantly accelerate privacy-preserving logistic regression. Leveraging this new method, which we call PrivLogit, we propose two new secure protocols for conducting logistic regression in a privacy-preserving and distributed manner. Extensive theoretical and empirical evaluations prove the competitive performance of our two secure proposals while ensuring accuracy and privacy: with speedup up to 2.3x and 8.1x, respectively, over state-of-the-art; and even faster as data scales up. Our drastic improvement makes privacy-preserving logistic regression more scalable and practical to large-scale studies which are common for modern science. In addition, our proposal of the PrivLogit optimizer is agnostic of and parallel to existing and future performance innovations from cryptography alone, thus can serve as a drop-in replacement for any privacy-preserving (distributed) logistic regression protocols.

V.2.1 Introduction

Logistic regression is a fundamental statistical model with wide adoption in various domains, such as in computer science, biomedical and social sciences (e.g., healthcare, ge-

netics, psychology, education, etc), etc. To reach powerful and reliable statistical conclusions, it is increasingly popular for these disciplines to perform collaborative regression through data sharing and joint analysis across a federation of organizations [111]. Such a trend, however, is often hampered by serious privacy concerns as human subject data underlying these studies are typically considered sensitive and strictly protected by various privacy laws and regulations [123, 72, 32]. Meanwhile, many organizations are also reluctant to reveal their data content to external entities (due to concerns around privacy and business secrets), even though they still want to contribute to collaborative studies. This is increasingly common in areas such as healthcare, business, finance, etc.

More formally, we are interested in the following common scenario: multiple independent organizations (e.g., different institutions, medical centers, etc) want to conduct joint analytics (e.g., logistic regression). They each possess their respective private data of a sub-population (e.g., patient health records or human genomes), but are not willing or permitted to disclose the data beyond their respective organizations due to privacy and proprietary reasons. We focus on the horizontally partitioned setting [4]. In such a collaborative study, potential adversaries include: distrustful aggregation center (e.g., due to breached servers or malicious employees), distrustful member organizations (due to curiosity about other organizations' secrets or business competition), and external curious people or hackers. The adversary's goal is to learn privacy-sensitive information of individual data records or organizations by peeking into raw and summary-level data. The challenge here is on how to support such a collaborative study while preserving privacy, especially when it is difficult or economically impractical to find a fully entrusted central authority.

Cryptography (secure multi-party computation or SMC in particular) and distributed computing are classical and reviving solutions for tackling the challenge [4]. Numerous efforts have attempted to support data mining without disclosing raw and intermediate data [4, 155, 118, 121, 159, 17, 93] (known as privacy-preserving distributed data mining). Among these, significant attention is devoted to logistic regression [154, 155, 41,

118, 93, 7].

Despite encouraging progress, few proposals have seen wide adoption in real world for privacy-preserving logistic regression. A major reason seems to concern the excess computational overhead of cryptographic protocols. While it is generally expected for secure computation to be slower than non-secure counterparts, we also make a surprising observation: much of the computational overhead indeed traces back to the sub-optimal technical decisions made by humans experts (e.g., authors of secure protocols) and could have been avoided. For instance, nearly all existing secure protocols [155, 41, 93] directly apply mainstream (distributed) model estimation algorithms (e.g., the popular Newton method for logistic regression [65]), failing to account for secure computing-specific characteristics and thus missing valuable opportunities for performance improvement.

In this work, we present a contrasting perspective on privacy-preserving logistic regression, and propose an improved model estimation method tailored for secure computing which significantly accelerates the computation while guaranteeing privacy and accuracy. In our proposal (termed PrivLogit), we derive a constant approximation for the second-order curvature information (i.e., Hessian) in the Newton method for logistic regression. This adapted optimizer seems counter-intuitive and "unfavorable" due to its elongated convergence and increased network interactions, but surprisingly turns out to be highly competitive in performance.

Following PrivLogit, we propose and evaluate two highly-efficient cryptographic protocols for privacy-preserving distributed logistic regression, i.e., PrivLogit-Hessian and PrivLogit-Local.

V.2.1.0.1 Contributions

Our contributions are as follows:

- We propose a secure computing-centric perspective for selecting model estimation methods, and introduce a counter-intuitive but surprisingly better approach (i.e.,

PrivLogit) for privacy-preserving logistic regression.

- We propose two highly-efficient secure protocols (i.e., PrivLogit-Hessian and PrivLogit-Local) for privacy-preserving logistic regression.
- We provide detailed theoretical analysis on our proposals.
- We extensively evaluate our proposals on various simulated and real-world studies of large scale.

V.2.1.0.2 Outline

To set context, we first provide background on logistic regression and model estimation methods in Section V.2.2. In Sections V.2.3 and V.2.4, we describe our improved optimization method PrivLogit, and two secure implementations. In Section V.2.5, we elaborate on theoretical details regarding security guarantees, computational complexity, model convergence of our proposals. This is followed by experimental results in Section V.2.6. In Section V.2.7, we survey related works. We discuss and conclude in Section V.2.8.

V.2.2 Logistic Regression and Newton Method

Before introducing logistic regression and the model estimation Newton method, we first list the main notations in this work in Table V.4.

Table V.2: Notations.

Notations	Meaning
$\mathbf{X} \in \mathbb{R}^{n \times p}$	Regression covariates: n samples, p features
$\mathbf{y} \in \mathbb{R}^{n}$	Regression response vector: n samples
$\beta \in \mathbb{R}^{p}$	Regression coefficients
$\mathbf{H}$, $\tilde{\mathbf{H}} \in \mathbb{R}^{p \times p}$	Hessian, approximate Hessian matrices
$\mathbf{g} \in \mathbb{R}^{p}$	Gradient
$\lambda \in \mathbb{R}$	Regression regularization parameter
$l_2(\beta)$	Log-likelihood (objective)
$Enc(data)$	Encryption of $data$
$\oplus, \ominus, \otimes, \oslash$	Secure arithmetics for $+, -, \times, \div$
$E_{sqrt}(data)$	Secure square root of $data$

V.2.2.1 Logistic Regression

This work concerns conducting logistic regression in a collaborative (distributed) environment. Logistic regression is a probabilistic model that can be used for predicting binary (i.e., categorical) outcomes [65]. It is among the most utilized statistical models in practice, with wide adoption in biomedicine [103], genetics [92], economics [65], online advertising [113], and so on. Briefly, the logistic regression model is defined as:

$$p(y = 1|\mathbf{x};\beta) = \frac{1}{1+e^{-\beta^T \mathbf{x}}} , \tag{V.8}$$

where $p(.)$ denotes the probability of the binary response variable y equal to 1 (i.e., "case" or "success" in practice), $\mathbf{x}$ is the p-dimensional covariates for a specific data record, and β is the p-dimensional regression coefficients we want to estimate.

In practice, regularization is often applied to the model estimation process to aid feature selection and prevent overfitting by penalizing extreme parameters [119]. Here we consider the popular ℓ_2-regularization (or ridge) for logistic regression [119] to make our work generically applicable. The standard logistic regression can be derived by simply setting the regularization to 0. The ℓ_2-regularized logistic regression imposes an additional regularization term, $-\frac{\lambda}{2}\beta^T\beta$, to the optimization objective during model estimation. For a dataset $(\mathbf{X} \in \mathbb{R}^{n\times p}, \mathbf{y} \in \mathbb{R}^n) = [(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)]$ with n independent samples and p features, the log-likelihood (i.e., optimization objective) of ℓ_2-regularized logistic regression is:

$$l_2(\beta) = \sum_{i=1}^{n} [y_i(\beta^T \mathbf{x_i}) - \log(1 + e^{\beta^T \mathbf{x_i}})] - \frac{\lambda}{2}\beta^T\beta , \tag{V.9}$$

where λ is the predefined penalty parameter to tune the regularization.

V.2.2.2 Distributed Newton Method

When fitting a (regularized) logistic regression, the goal is to estimate the coefficients β from existing training data $(\mathbf{X}, \mathbf{y})$. Since logistic regression does not have a closed form,

model estimation is often accomplished by numerical optimization over the objective $l_2(\beta)$. The *de facto* approach for estimating the (regularized) logistic regression coefficient β (Equation V.8) is the Newton method (or iteratively reweighted least squares, known as IRLS) [65]. Newton method iteratively approaches the optimal coefficients, and for each iteration, the coefficient estimates are updated by:

$$\beta^{(t+1)} = \beta^{(t)} - \mathbf{H}^{-1}(\beta^{(t)})\, \mathbf{g}(\beta^{(t)}) \,, \tag{V.10}$$

where $\mathbf{H}(\beta^{(t)})$ and $\mathbf{g}(\beta^{(t)})$ denote the Hessian and gradient of the objective $l_2(\beta)$ (Equation V.18) evaluated at the current $\beta^{(t)}$ coefficient estimate. The superscripts $(t), (t+1)$ denote the $t^{\texttt{th}}, (t+1)^{\texttt{th}}$ iterations, respectively. This updating process iterates until model convergence.

Based on Equation V.18, the gradient and Hessian for ℓ_2-regularized logistic regression can be computed as follows (setting $\lambda = 0$ will skip regularization and yield the standard logistic regression):

$$\mathbf{g}(\beta) = \nabla_\beta l_2(\beta) = \mathbf{X}^T(\mathbf{y} - \mathbf{p}) - \lambda\beta = \sum_{j=1}^{S} \mathbf{g_j}(\beta) - \lambda\beta \,, \tag{V.11}$$

$$\mathbf{H}(\beta) = \frac{d^2 l_2(\beta)}{d\beta d\beta^T} = -\mathbf{X}^T\mathbf{A}\mathbf{X} - \lambda\mathbf{I} = \sum_{j=1}^{S} \mathbf{H_j}(\beta) - \lambda\mathbf{I} \,, \tag{V.12}$$

where $\mathbf{X}$ represents covariates of n samples and p features; $\mathbf{y}$ denotes the response vector of n data records; $\mathbf{p} \in \mathbb{R}^n$ is the vector of logistic regression probabilities for n records; $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with elements defined as $a_{i,i} = p_i(1-p_i)$; and $\mathbf{g_j}(\beta)$ and $\mathbf{H_j}(\beta)$ are the per-organization gradient and Hessian, respectively, that will be introduced afterwards in the distributed version; S is the total number of organizations contributing data to the collaborative study.

As is also manifested in the last equalities of Equations V.21 and V.22, the computation of both $\mathbf{g}(\beta)$ and $\mathbf{H}(\beta)$ can be decomposed per participating organizations (who can freely

access their respective private data such as $\mathbf{X}_j, \mathbf{y}_j$), and thus need not invoke expensive cryptographic computation (except for the final summation across organizations).

The (distributed) Newton method is widely implemented in statistical software and also underlies almost all existing solutions for privacy-preserving logistic regression [154, 155, 41, 93].

V.2.3 PrivLogit: A Novel Optimizer Tailored for Fast Logistic Regression

Here, we first point out problems with mainstream secure Newton method, which motivates us to design a better optimization method (PrivLogit) tailored for secure computing. We later analyze the attractive properties of PrivLogit.

V.2.3.1 Limitations of Newton Method.

To estimate regression coefficients via the aforementioned Newton method (Equations V.19), the evaluation and inversion of the Hessian have to be repeated for every iteration until model convergence. These two operations can be prohibitively expensive in computation and network communication especially when implemented using cryptography.

For (distributed) Newton method in general (e.g., privacy-free applications), it has been well acknowledged that the evaluation and inversion of the Hessian matrix are the overall computational bottleneck due to large data sizes, inherent complexity and repetitive nature of these operations [28, 99]. This in fact has motivated numerous improved optimizers referred to as Quasi-Newton or Hessian-free optimization in machine learning and optimization [28, 99] (unfortunately, most such enhancements do not seem amenable to efficient and data-agnostic secure implementation and thus are not covered in our work).

In data security and privacy research, the issue of expensive Newton method is exacerbated as secure inversion of Hessian matrix requires complex operations (e.g., secure division and square root) which have to resort to expensive primitives and approximations from secure multi-party computation (SMC) [118, 41]. As a result, almost all existing secure logistic regression proposals have to compromise privacy protection or result accuracy to

increase performance (e.g., to selectively reveal intermediate data/computations [155, 93] or to use approximations [118, 7]).

In addition, the lack of model convergence guarantee in Newton method is also a known issue, when poor initialization (initial guess of coefficients) is provided [16].

V.2.3.2 PrivLogit for Fast Privacy-preserving Logistic Regression.

We are motivated to design a tailored optimizer for secure computing by addressing the aforementioned limitations of Newton method. Our proposal is inspired by a classical work on quadratic function approximation [16] and with new theoretical analysis. In brief, we propose to use one carefully-chosen constant matrix as a surrogate for the exact Hessian matrices across all iterations. Specifically, the following approximate Hessian (denoted $\tilde{\mathbf{H}}$) is proposed:

$$\tilde{\mathbf{H}} = -\frac{1}{4}\mathbf{X}^T\mathbf{X} - \lambda\mathbf{I}\,, \tag{V.13}$$

here $\tilde{\mathbf{H}}$ is a tight lower bound because for all $p_i \in [0,1]$ (the probability in logistic regression), we have that: $\max\{a_{i,i} = p_i(1-p_i)\} = \frac{1}{4}$ (where $a_{i,i}$ denotes elements of the diagonal matrix $\mathbf{A}$ defined in Equation V.22 for Hessian). We highlight that this approximation guarantees exact model convergence and do not affect accuracy (with theoretical proof later).

The calculation of approximate Hessian $\tilde{\mathbf{H}}$ can be decomposed per-organization (horizontally partitioned) and computed in a distributed manner among many organizations:

$$\tilde{\mathbf{H}} = -\frac{1}{4}\sum_{j=1}^{S}\mathbf{X_j}^T\mathbf{X_j} - \lambda\mathbf{I} = \sum_{j=1}^{S}\tilde{\mathbf{H}}_j - \lambda\mathbf{I} \tag{V.14}$$

where $\mathbf{X}_j$ is the (privacy-sensitive) raw data stored locally at Organization j, S is the total number of organizations contributing data, and $\tilde{\mathbf{H}}_j = -\frac{1}{4}\mathbf{X}_j^T\mathbf{X}_j$ denotes the approximate Hessian for Organization j.

Substituting this approximate Hessian into the Newton method (Equation V.19), along

with the distributed evaluation of gradient, the iterative updating formula for our new optimizer (denoted as PrivLogit) follows:

$$\beta^{(t+1)} = \beta^{(t)} - [\sum_{j=1}^{S} \tilde{\mathbf{H}}_j - \lambda \mathbf{I}]^{-1} \, [\sum_{j=1}^{S} \mathbf{g}_j(\beta^{(t)}) - \lambda \beta^{(t)}] \tag{V.15}$$

The above iterative process continues until model convergence. Convergence can be measured by the relative change of log-likelihood and compared against a predefined threshold (e.g., 10^{-6}):

$$\frac{|l_2^{(t+1)} - l_2^{(t)}|}{|l_2^{(t)}|} < 10^{-6} \, , \tag{V.16}$$

where $l_2^{(t+1)}, l_2^{(t)}$ correspond to the log-likelihood of logistic regression for Iterations $(t+1)$ and (t), respectively.

V.2.3.3 Advantages of PrivLogit.

Our new PrivLogit optimizer enables a few attractive properties, which seem highly promising for efficient privacy-preserving logistic regression.

V.2.3.3.1 Asymmetric Computational Complexity in Secure Distributed Settings

The PrivLogit adaption comes at the cost of more iterations required for convergence (and also increased local-organization computation), which seems counter-intuitive and less favorable because more iterations mean slower convergence. However, this view fails to consider computational cost as a whole and the different computational characteristics of distributed model estimation with and without cryptographic protections. In secure implementations, the local computation at each organization is essentially "free" because organizations have full control of their respective private data and fast non-secure computations are applicable; but secure computation at the aggregation center is usually orders of magnitudes slower than non-secure counterparts (due to expensive cryptographic protections against an adversarial center). This implies that eliminating complexity of center-based

secure computation (current bottleneck) can potentially lead to significant speedup (as is the case in PrivLogit).

V.2.3.3.2 Constant Hessian

Our proposed Hessian approximation stays constant and independent of the varying $\beta^{(t)}$'s coefficients across all iterations. This indicates that it only needs to be evaluated and inverted once during preprocessing and can then be reused across all iterations, leading to dramatic reduction in computation compared with traditional Newton method.

V.2.3.3.3 Decomposition of Computation

The new optimizer allows for easy decomposition the computation among participating organizations, which can be leveraged to achieve significant speedup. For instance, the approximate Hessian can be computed in a distributed manner via a series of aggregations, as demonstrated in Equation V.14. So is the gradient.

In addition, further reduction in computation is possible after the approximate Hessian is securely inverted and properly protected. As will be introduced later in our second implementation PrivLogit-Local (Section V.2.4.2), partial Newton update direction can be computed locally by each local nodes (who has privacy-free access to their respective private data and thus local gradient need not be encrypted). The center only needs to securely aggregate these local Newton steps, which is highly efficient.

V.2.3.3.4 Guaranteed Model Quality

Despite the approximation to Hessian, the PrivLogit optimizer is guaranteed to converge to accurate model estimates. We will demonstrate this property both analytically and empirically later.

V.2.3.3.5 Guaranteed Model Convergence

Finally, our adapted optimizer has better convergence guarantee than standard Newton. In particular, PrivLogit will generate a sequence of parameter estimates which monotonically increase the value of the objective function, leading to guaranteed convergence to the optimal solution of the convex objective no matter what initializations the algorithm adopts. In contrast, Newton method may fail to converge with "poor" initializations (as demonstrated later). Moreover, unlike gradient descent and Newton method, there is no need for complicated and expensive line searching for best step size.

V.2.4 Safeguarding PrivLogit

Based on our new PrivLogit optimizer, we propose two secure protocols for preserving privacy in logistic regression. The first is called PrivLogit-Hessian and is a straightforward cryptographic implementation of PrivLogit. Our second proposal, called PrivLogit-Local, further offsets some expensive matrix operations to local organizations and take advantage of their fast and privacy-free computing power.

Both our secure protocols adopt the distributed architecture consisting of local Nodes (organizations) and an aggregation Center (semi-honest), as illustrated in Figure V.5. In brief, participating organizations (i.e., Nodes) are responsible for protecting their respective data and only generating (safe) summary-level data, which would be encrypted and securely consumed by the Center for model estimation. In a strongly protected system such as ours, all data and computations at the Center are encrypted and not visible even to the Center itself. The role of Center is typically played by two or more mutually independent semi-trusted authorities (denoted as different Servers in Figure V.5), as is common for secure multi-party computation applications [4, 121, 159, 93]. As long as there is no major collusion between the authorities, the security of the system is guaranteed. For practical deployment such as in biomedical or social sciences, the role of Center could be assumed by the coordinating center (of a consortium, federation or association) in addition to a

third-party authority (e.g., audit organizations or even a respectful member organization).

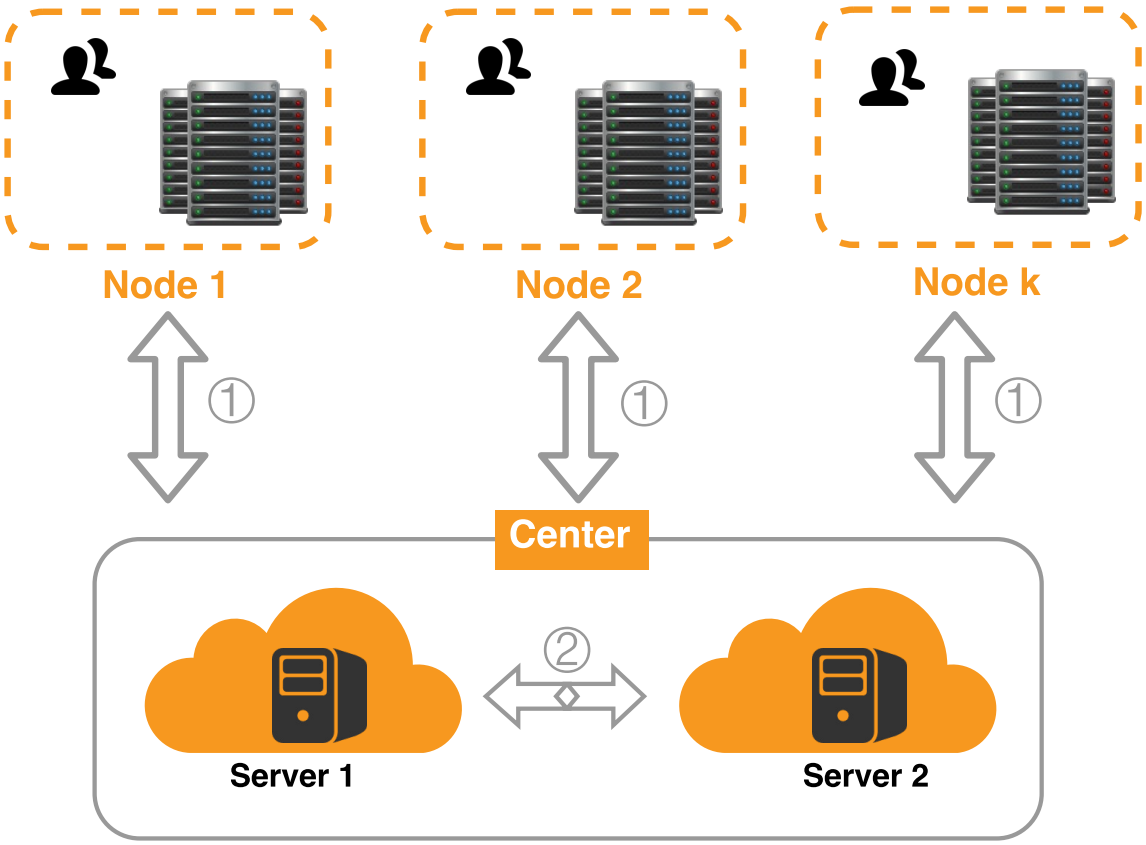


Figure V.5: Distributed architecture for privacy-preserving logistic regression. Two main types of computations are involved between: 1) local Nodes and the Center; 2) different Servers/authorities at the Center.

Our proposals are agnostic of specific choices of cryptographic schemes and many existing or new cryptographic sub-protocols can be leveraged. Since the focus of our work is not on specific cryptographic protocols and due to space constraint, we avoid specific cryptographic details (e.g., cryptographic key management) that are common knowledge in privacy-preserving (distributed) data mining [4, 121]. For demonstration purpose, we build on a hybrid of two popular schemes, i.e., Yao's garbled circuit [164] (mainly for Type 2 computations between independent Center servers as depicted in Figure V.5) and the Paillier cryptosystem [125] (mainly for Type 1 computations between local Nodes and the Center as depicted in Figure V.5). Such hybrid schemes also underlie various state-of-the-art protocols for privacy-preserving logistic regression and other machine learning

models [4, 41, 121].

Note that our protocols do not address output privacy (i.e., privacy issues of releasing regression coefficients), which is typically covered by an independent topic called differential privacy [37] (also briefly discussed earlier in Section II.3.3). This complies with security guarantees and primary focus of SMC-based secure protocols. For simplicity, we use intuitive symbols to denote a few common secure mathematical arithmetics. Each of these operations take encrypted operands as inputs, and securely compute without decryption to output an encrypted result. As before, encrypted data are represented as $Enc(.)$. And we denote secure addition, subtraction, multiplication, division and square root as: $\oplus, \ominus, \otimes, \oslash, E_{sqrt}(.)$, respectively.

V.2.4.1 PrivLogit-Hessian: Secure Distributed Approximate Hessian

PrivLogit-Hessian is our straightforward secure and distributed implementation of the PrivLogit optimizer, as presented in Algorithm 6. In Algorithm 6, we flag computations by their location of occurrence in accordance with the distributed architecture in Figure V.5 (i.e., whether the computation is conducted by local Nodes or by SMC servers at the Center).

The secure PrivLogit-Hessian protocol consists of two phases of computation: a one-time setup phase of securely approximating and inverting the Hessian, and a repeated (iterative) secure model estimation phase.

The first phase (Step 1 in Algorithm 6 or *SetupOnce()* function in Algorithm 7) focuses on securely approximating and inverting Hessian. Specifically, based on Algorithm 7, each local organizations compute their local Hessian approximation $\tilde{\mathbf{H}}_j$ (based on covariance matrix $\mathbf{X}_j^T\mathbf{X}_j$) and encrypt it before sharing with the Center (Steps 1 to 4 in Algorithm 7). The Center securely aggregates these encrypted per-organization Hessians (and the regularization term as necessary), yielding an encrypted global Hessian approximation $Enc(\tilde{\mathbf{H}})$ (Step 5 in Algorithm 7 and Equation V.14). Later, the Center needs to securely invert the Hessian, which is typically achieved by secure Cholesky decomposition (see Appendix)

Algorithm 6 PrivLogit-Hessian: Fast and Secure Logistic Regression.

Input: Random initial $\beta^{(0)}$; Regularization parameter λ
Output: Globally fit coefficient estimate β

[At local organizations and Center] :

Securely approximate and Cholesky-decompose (negated) Hessian: $Enc(\mathbf{L}) = SetupOnce()$ (where $Enc(\mathbf{L}\mathbf{L}^T) = Enc(-\tilde{\mathbf{H}})$)

while regression model not converged **do**

[At local organizations] :

for each organization j = 1 **to** S **do**

Compute local gradient $\mathbf{g}_j$ and encrypt

Compute local log-likelihood l_{sj} and encrypt

Securely transmit encryptions $Enc(\mathbf{g}_j), Enc(l_{sj})$ to Center

end for

[At Center] :

Securely aggregate gradients across organizations: $Enc(\mathbf{g}) = Enc(\mathbf{g}_1) \oplus ... \oplus Enc(\mathbf{g}_j) \oplus ... \oplus Enc(\mathbf{g}_S) \ominus Enc(\lambda\beta^{(t)})$

Secure back-substitution: $Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}) \leftarrow Enc(\mathbf{L})$, $Enc(\mathbf{g})$

Securely update coefficient estimates via PrivLogit: $\beta^{(t+1)} \leftarrow \beta^{(t)}$ (Equation V.15)

Securely aggregate log-likelihood across organizations: $Enc(l_2) = Enc(l_{s1}) \oplus ... \oplus Enc(l_{sj}) \oplus ... \oplus Enc(l_{sS}) \ominus Enc(\frac{\lambda}{2}[\beta^{(t)}]^T\beta^{(t)})$

Securely check model convergence

Securely disseminate new coefficient estimates to each local organizations: $Enc(\beta^{(t+1)})$

end while

return $\beta^{(t)}$ (final converged estimate)

Algorithm 7 *SetupOnce()* for securely approximating and inverting Hessian.

Input: Local organizations with their respective data
Output: Encrypted triangular matrix $Enc(\mathbf{L})$ from Cholesky decomposition (where $Enc(\mathbf{L}\mathbf{L}^T) = Enc(-\tilde{\mathbf{H}})$)

[At local organizations] :

for each organization j = 1 **to** S **do**

Approximate local Hessian $\tilde{\mathbf{H}}_j$

Encrypt and securely transmit $Enc(\tilde{\mathbf{H}}_j)$ to Center

end for

[At Center] :

Securely aggregate Hessians across organizations: $Enc(\tilde{\mathbf{H}}) = Enc(\tilde{\mathbf{H}}_1) \oplus ... \oplus Enc(\tilde{\mathbf{H}}_j) \oplus ... \oplus Enc(\tilde{\mathbf{H}}_S) \ominus Enc(\lambda\mathbf{I})$

Secure Cholesky decomposition to obtain: $Enc(\mathbf{L})$ (where $Enc(\mathbf{L}\mathbf{L}^T) = Enc(-\tilde{\mathbf{H}})$)

return encryption $Enc(\mathbf{L})$

on the protected (negated) Hessian and obtains its encrypted "inversion" (the encrypted Cholesky triangular matrix $Enc(\mathbf{L})$ to be precise), such that $Enc(\mathbf{L}\mathbf{L}^T) = Enc(-\tilde{\mathbf{H}})$. Note that the whole phase only needs to occur once, which is a significant improvement over Newton method-based protocols.

The second phase (Steps 2 to 14 in Algorithm 6) of PrivLogit-Hessian resembles that of the widely-used privacy-preserving distributed Newton method, except for the substitution of repeated Hessian evaluation and inversion. Model estimation proceeds in a secure and iterative process. Model convergence is checked at each iteration (Step 12 in Algorithm 6). For each iteration, local organizations only need to compute their local gradient $\mathbf{g}_j$ and log-likelihood l_{sj} (where j indexes each organization), and securely transmit their encryptions to the Center (Steps 3 to 7). The Center securely aggregates the gradient and log-likelihood submissions, and compose the encrypted global gradient (Step 8) and log-likelihood (Step 11). Later on in Step 9, back-substitution is securely performed to derive the encrypted product $Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g})$ from previously derived encryptions $Enc(\mathbf{L})$ and $Enc(\mathbf{g})$. The Center then updates current coefficient estimates following the PrivLogit updating formula (Step 10 and Equation V.15). This iterative process continues until model converges.

V.2.4.1.1 Secure Cholesky Decomposition

Secure Cholesky decomposition (Algorithm 8; also used by [121]) is used in our protocol to help "invert" the (negated) Hessian, which is the main computation in PrivLogit (and the bottleneck in Newton method). We denote the input matrix to be $\mathbf{B} \in \mathbb{R}^{p \times p}$, and each elements of it as $L_{i,j}$, where i, j index the row and column positions, respectively.

V.2.4.2 PrivLogit-Local: Further Offsetting Computations to Local Nodes.

Our second and even faster secure protocol, PrivLogit-Local, is presented in Algorithm 9. This protocol takes advantage of the fact that the centrally aggregated approximate Hessian $\tilde{\mathbf{H}}^{-1}$ (or encryption $Enc(\tilde{\mathbf{H}}^{-1})$) can be regarded as a (private) constant value. We note that for each local Nodes, local gradient $\mathbf{g}_j$ is privacy-free and essentially a public constant.

Algorithm 8 Secure Cholesky decomposition of matrix $\mathbf{B}$.

Input: Encryption $Enc(\mathbf{B})$, where $\mathbf{B} = (L_{i,j}) \in R^{p \times p}$
Output: Encrypted triangular matrix $Enc(\mathbf{L})$, such that $\mathbf{L}\mathbf{L}^T = \mathbf{B}$

[At Center] :

1: **for** j = 1 **to** p **do**
2: **for** k = 1 **to** j-1 **do**
3: **for** i = j **to** p **do**
4: $Enc(L_{i,j}) = Enc(L_{i,j}) \ominus Enc(L_{i,k}\, L_{j,k})$
5: **end for**
6: **end for**
7: $Enc(L_{i,j}) = E_{sqrt}(L_{j,j})$
8: **for** k = j+1 **to** p **do**
9: $Enc(L_{k,j}) = Enc(L_{k,j}) \oslash Enc(L_{j,j})$
10: **end for**
11: **end for**
12: **return** $Enc(\mathbf{L}) = Enc(updated\ \mathbf{B})$

This means that we can further distribute the expensive (center-based) matrix-vector multiplication to local Nodes by leveraging cheap secure multiplication-by-constant locally: i.e., to locally compute $Enc(\tilde{\mathbf{H}}^{-1}) \otimes \mathbf{g}_j$ (which can be centrally aggregated efficiently in secure later).

In greater detail, the first step of PrivLogit-Local still involves the local organizations and Center securely approximating and "inverting" the Hessian (Step 1 in Algorithm 9; or *SetupOnce()* in Algorithm 7), similar to Phase 1 of PrivLogit-Hessian. Next, we directly materialize the inversion of approximate Hessian in encrypted form, i.e., $Enc(\tilde{\mathbf{H}}^{-1})$. After that, this encrypted inversion is disseminated to each local organizations where local computation of gradients only involves privacy-free operations.

Later on, at each iteration, local organizations derive their local summaries, such as log-likelihood (Step 5) and gradient (Step 6). Then they compute their respective versions of (partial) Newton updating step, by using efficient secure multiplication primitives. Since the local gradients $\mathbf{g}_j$ do not involve privacy concerns at their respective local organizations (thus can be regarded as a public constant value), the computation is greatly simplified to highly efficient secure multiplication-by-constant primitives. Afterwards, local organiza-

Algorithm 9 PrivLogit-Local: offsetting partial Newton update step to local organizations.

Input: Random initial $\beta^{(0)}$; regularization parameter λ
Output: Globally fit coefficient estimate β

[At local organizations and Center] :

1: Securely approximate and Cholesky-decompose Hessian: $Enc(\mathbf{L}) = SetupOnce()$ (where $Enc(\mathbf{L}\mathbf{L}^T) = Enc(-\tilde{\mathbf{H}})$)
2: Securely invert Hessian: $Enc(\tilde{\mathbf{H}}^{-1}) \leftarrow Enc(\mathbf{L})$

3: **while** regression model not converged **do**
[At local organizations] :
4: **for** each organization j = 1 **to** S **do**
5: Compute local log-likelihood l_{sj} and encrypt
6: Compute local gradient $\mathbf{g}_j$
7: Secure multiplication: $Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}_j) \leftarrow Enc(\tilde{\mathbf{H}}^{-1}),\ \mathbf{g}_j$;
8: Securely send encryptions $Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}_j),\ Enc(l_{sj})$ to Center
9: **end for**

[At Center] :
10: Securely compose global numerical updating step: $Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}) = Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}_1) \oplus ... \oplus Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}_j) \oplus ... \oplus Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}_S) \ominus Enc(\lambda\tilde{\mathbf{H}}^{-1}\beta^{(t)})$
11: Securely update coefficient estimates via PrivLogit: $\beta^{(t+1)} \leftarrow \beta^{(t)}$ (Equation V.15)
12: Securely aggregate log-likelihood across organizations: $Enc(l_2) = Enc(l_{s1}) \oplus Enc(l_{s2}) \oplus ... \oplus Enc(l_{sj}) \ominus Enc(\frac{\lambda}{2}[\beta^{(t)}]^T\beta^{(t)})$
13: Securely check model convergence
14: Securely disseminate new coefficient estimates to each local organiztions: $Enc(\beta^{(t+1)})$
15: **end while**
16: **return** $\beta^{(t)}$ (last converged estimate)

tions send their encrypted summaries $Enc(\tilde{\mathbf{H}}^{-1}\mathbf{g}_j), Enc(l_{sj})$ back to the Center (Step 8). For regularized logistic regression, the regularization term also needs to be securely composed, which can be prepared by the local organizations and then aggregated centrally, i.e., $Enc(\lambda\tilde{\mathbf{H}}^{-1}\beta^{(t)}) = Enc(\tilde{\mathbf{H}}^{-1}\sum_{j=1}^{S}\lambda\beta_j^{(t)})$. Finally, the Center only needs to perform trivial secure aggregation to complete the Newton updating process and convergence check (Steps 10 to 13).

The correctness of Algorithm 9 is evident. Briefly, $\tilde{\mathbf{H}}^{-1}\mathbf{g} = \tilde{\mathbf{H}}^{-1}(\sum_j \mathbf{g}_j - \lambda\beta) = \sum_j \tilde{\mathbf{H}}^{-1}\mathbf{g}_j - \lambda\tilde{\mathbf{H}}^{-1}\beta$.

A slight variant to this protocol would be to directly use the Cholesky triangular matrix $Enc(\mathbf{L})$ (as is generally recommended for privacy-free scenarios) instead of the actual Hessian inversion. Our evaluation indicates that they have equivalent performance in secure. Thus we opt for the implementation introduced above.

In addition to earlier improvements from PrivLogit-Hessian, our second protocol further avoids expensive secure matrix multiplication (between encryptions), which leads to significantly less computation than PrivLogit-Hessian and Newton.

V.2.5 Theoretical Analysis and Proof

In this section, we present theoretical analysis and proof for our proposals regarding computational complexity, and model convergence.

V.2.5.1 Complexity Analysis

Here we roughly analyze the computational complexity of the operations involved, loosely following the Big-O notation. Since cryptographic operations are dominating the total computation of secure protocols introduced, we thus focus on cryptography-related procedures only.

For gradient $\mathbf{g} \in \mathbb{R}^p$ and Hessian $\mathbf{H} \in \mathbb{R}^{p \times p}$, the main operations concerning privacy protection are: matrix-vector multiplication ($O(p^2)$ complexity), matrix inversion ($O(p^3)$), Cholesky decomposition ($O(p^3)$), and back-substitution ($O(p^2)$).

State-of-the-art privacy-preserving Newton method requires repeatedly decomposing Hessians and matrix multiplication, with total complexity of $O(p^3 \times \texttt{Newton iterations})$.

PrivLogit in general requires one step of Hessian inversion, and many iterations of matrix-vector multiplication, with total complexity of $O(p^3 + p^2 \times \texttt{PrivLogit iterations})$. Note that specifically for PrivLogit-Local, the second complexity term can much lower (than PrivLogit-Hessian) since multiplication-by-constant (the main computation involved) is much more efficient than secure multiplication of two encryptions (as in PrivLogit-Hessian).

Since the relationship between p and iteration numbers (of Newton and PrivLogit) is not determined, performance improvement is not strictly guaranteed for (directly applying) PrivLogit over Newton method. This is a limitation of one related work [118]. In practice, we show that PrivLogit tends to have lower amortized cost, since PrivLogit iterations are of low cost. And this advantage grows with data dimensionality p. Our second adaption PrivLogit-Local should guarantee to outperform Newton and the speedup is significant. This is because statistics and optimization theory suggests that the iteration numbers (for Newton or PrivLogit) grows slower than dimensionality (p). So the dominant factor for complexity is the first term involving p^3, where PrivLogit-Local manifests obvious improvement over Newton empirically.

V.2.5.2 Security Guarantees

Our work considers the *honest-but-curious* adversary model [55], where the adversary always follows the prescribed protocol, but may attempt to learn additional knowledge from the information flowed by. Since the focus of our work is not on specific cryptographic implementations and for demonstration we use standard secure primitives (e.g., Yao's garbled circuit [164], Paillier cryptosystem [125]) whose security are well established, we only provide concise security analysis for brevity.

In both PrivLogit-Hessian and PrivLogit-Local, local-Node summaries are encrypted prior to submission to guarantee privacy. In PrivLogit-Local, the inverted approximate Hessian is also encrypted before being shared with local Nodes. At the aggregation Center, all incoming inputs are encrypted in Paillier or Yao's garbled circuit shares. All data, computations and results are also encrypted. Based on the composition theorem of security [55], the composition of these secure sub-protocols also yield a secure protocol overall.

The only information disclosure is the regression coefficients (shared with local nodes), which share the same privacy properties as the final output (e.g., final regression coefficients). By definition, cryptographic protocols do not guarantee security on final output

itself (as mentioned earlier), so this practice does not violate security. The only potential way to breach security here is for local nodes to form a linear equation system using many regression coefficients from all iterations [124, 41]. However, since the number of iterations is small and (private) data size/dimensionality is huge ($n \times p$), the system is severely undetermined and such attacks are not possible.

V.2.5.3 Convergence Proof for PrivLogit

Since our PrivLogit introduced approximation to Hessian, the convergence properties of standard Newton no longer apply. We thus present theoretical proof regarding the convergence of PrivLogit, which is based on quadratic function approximation [16]. We show that our PrivLogit optimizer is guaranteed to converge to the optimum, and at a linear convergence rate. Specifically, we prove the following proposition:

Proposition 1. *Assume the optimal solution β^* to the objective function $l_2(\beta)$ (Equation V.18) exists and is unique. Let $\{\beta^{(t)}\}$ be a sequence generated by PrivLogit with the update formula in Equation V.15. The sequence has the following properties:*

(a) $l_2(\beta^{(t+1)}) > l_2(\beta^{(t)})$ and $\beta^{(t)}$ will converge to the optimal solution β^*.

(b) The rate of convergence of PrivLogit method is linear.

Proof.

(a) By using the negative definiteness of $\tilde{\mathbf{H}}$ and the second-order Taylor expansion of $l_2(\beta)$, we have,

$$
\begin{aligned}
& l_2(\beta^{(t+1)}) - l_2(\beta^{(t)}) \\
&= -\mathbf{g}(\beta^{(t)})^\intercal \tilde{\mathbf{H}}^{-1} \mathbf{g}(\beta^{(t)}) + \frac{1}{2} \mathbf{g}(\beta^{(t)})^\intercal \tilde{\mathbf{H}}^{-1} \mathbf{H}(\hat{\beta}) \tilde{\mathbf{H}}^{-1} \mathbf{g}(\beta^{(t)}) \\
&> -\mathbf{g}(\beta^{(t)})^\intercal \tilde{\mathbf{H}}^{-1} \mathbf{g}(\beta^{(t)}) + \frac{1}{2} \mathbf{g}(\beta^{(t)})^\intercal \tilde{\mathbf{H}}^{-1} \tilde{\mathbf{H}} \tilde{\mathbf{H}}^{-1} \mathbf{g}(\beta^{(t)}) \\
&= -\frac{1}{2} \mathbf{g}(\beta^{(t)})^\intercal \tilde{\mathbf{H}}^{-1} \mathbf{g}(\beta^{(t)}) > 0
\end{aligned}
$$

where $\hat{\beta}$ is between $\beta^{(t)}$ and $\beta^{(t+1)}$.

The objective function $l_2(\beta)$ is strictly concave with a negative definite Hessian matrix and therefore is maximized at the optimal solution β^*. From the previous derivation, we obtain the lower bound of the increment of the objective function at each iteration. If $\mathbf{g}(\beta^{(t)})$ is bounded away from 0 for all t, in other words, $||\mathbf{g}(\beta^{(t)})|| > \varepsilon$ for some positive constant ε, then the increment of each iteration is also bounded above 0, which contradicts the upper boundedness of the objective function. Therefore, $\mathbf{g}(\beta^{(t)}) \to 0$ as $t \to \infty$, which means the sequence $\{\beta^{(t)}\}$ converges to the optimal solution β^*.

(b) Since $\mathbf{X}^\intercal\mathbf{X}$ is positive semi-definite, its eigenvalues are all non-negative. Denote the biggest eigenvalue of $\mathbf{X}^\intercal\mathbf{X}$ as λ_{max}. Furthermore, we also assume $\mathbf{X}^\intercal\mathbf{A}\mathbf{X}$ is positive definite at every iteration, with the smallest eigenvalue $\lambda^{min} > 0$. Then we have

$$-\frac{1}{\frac{1}{4}\lambda_{max}+\lambda}\mathbf{I} \succeq \tilde{\mathbf{H}}^{-1}$$

and

$$-(\lambda^{min}+\lambda)\mathbf{I} \succeq \mathbf{H}(\beta) \succeq -(\frac{1}{4}\lambda_{max}+\lambda)\mathbf{I}$$

Let $M = \frac{1}{4}\lambda_{max}+\lambda$ and $m = \lambda^{min}+\lambda$. By the strong concavity assumption and the second-order Taylor expansion of l_2, we have for any υ and ω in the parameter space,

$$\begin{aligned} l_2(\omega) &< l_2(\upsilon) + \mathbf{g}(\upsilon)^\intercal(\omega-\upsilon) - \frac{1}{2}m||\omega-\upsilon||_2^2 \\ &< l_2(\upsilon) + \frac{||\mathbf{g}(\upsilon)||_2^2}{2m} \end{aligned}$$

Since the inequality holds everywhere in the parameter space, we have $||g(\upsilon)||_2^2 > 2m(l_2(\beta^*) - l_2(\upsilon))$ for any υ. Next we need to investigate the relation between $l_2(\beta^*) - l_2(\beta^{(t+1)})$ and

$l_2(\beta^*) - l_2(\beta^{(t)})$ for all t. From part(a), we have

$$\begin{aligned} l_2(\beta^{(t+1)}) &> l_2(\beta^{(t)}) - \frac{1}{2}\mathbf{g}(\beta^{(t)})^\intercal \tilde{\mathbf{H}}^{-1}\mathbf{g}(\beta^{(t)}) \\ &> l_2(\beta^{(t)}) + \frac{1}{2M}||\mathbf{g}(\beta^{(t)})||_2^2 \end{aligned}$$

Subtracting both sides from $l_2(\beta^*)$, we get

$$\begin{aligned} l_2(\beta^*) - l_2(\beta^{(t+1)}) &< l_2(\beta^*) - l_2(\beta^{(t)}) - \frac{1}{2M}||\mathbf{g}(\beta^{(t)})||_2^2 \\ &< (1 - \frac{m}{M})(l_2(\beta^*) - l_2(\beta^{(t)})) \\ &< (1 - \frac{m}{M})^t(l_2(\beta^*) - l_2(\beta^{(1)})) \end{aligned}$$

where the factor $1 - \frac{m}{M} < 1$. It shows that $l_2(\beta^{(t)})$ converges in a linear rate to β^* as $t \to \infty$. □

V.2.6 Experiments

We implement both PrivLogit-Hessian and PrivLogit-Local in the Java and Julia [14] programming languages. At a lower-level, our Yao's garbled circuit evaluation is building on top of state-of-the-art framework ObliVM-GC [98]. Numerical values are denoted in floating-point representations. We use the recommended 2048-bit security parameter for encryption. Other security parameters follow standards or default values from NIST and ObliVM-GC. Since no open-source code is available as baseline, we also implement state-of-the-art privacy-preserving distributed Newton method. We run all experiments between two commodity PC with 2.5 GHz quad-core CPU and 16 GB memory, connected via ethernet.

Our empirical evaluations focus on the following criteria: 1) Model estimation quality (the accuracy of estimated coefficients) (in Section V.2.6.2); 2) Model convergence performance (in Section V.2.6.3); 3) Guarantee in model convergence (in Section V.2.6.4).

In our experiments concerning numerical optimizers (i.e., PrivLogit and Newton method),

we randomly initialize first coefficient estimates as commonly suggested (e.g., 0 as initial guess). We use 10^{-6} as our stopping threshold when checking model convergence (i.e., relative change of likelihood). Other thresholds have also been tested, such as 10^{-7} and 10^{-8}, which do not affect our main results and conclusions and thus are not reported.

V.2.6.1 Datasets

Our empirical evaluation includes a series of simulated and real-world studies, covering a wide spectrum of applications from different domains and of different scales.

Among these, we have compiled four real-world studies, including: 1) the *Wine* quality study (with 6,497 samples and 12 features) [30] for predicting wine quality from physico-chemical tests, 2) online *Loans* data (with 122, 578 samples and 33 features) from Lending Club [91] for studying loan default status from loan application data, 3) company *Insurance* study (of dimension: $9,882 \times 38$) for predicting caravan issurance from demographic information and personal finance attributes, and 4) *News* dataset (of dimension $39,082 \times 52$) [48] for predicting the popularity of Mashable.com news from article features.

To make our evaluations more comprehensive, we have also simulated a series of studies with varying data scales, including: *SimuX10* (of dimension $50,000 \times 10$), *SimuX12* ($1,000,000 \times 12$), *SimuX50* ($1,000,000 \times 50$), *SimuX100* ($3,000,000 \times 100$), *SimuX150* ($4,000,000 \times 150$), *SimuX200* ($5,000,000 \times 200$), *SimuX400* ($50,000,000 \times 400$), etc. We also evaluated on additional studies with various data sizes and numbers of participating organizations. Since these factors do not have direct influence on the secure computation process (which primarily concerns summary data) both theoretically and empirically, we do not report on them separately. We following standard data simulation approach, by randomly generating covariates $\mathbf{X} \in R^{n \times p}$ and coefficients $\beta \in R^{p \times 1}$, and then deriving responses $\mathbf{y} \in R^{n \times p}$ according to Bernoulli distribution.

These evaluation datasets should be representative for most large-scale studies in our focused domains in the foreable future. We also randomly partition datasets into subsets

(by row or horizontally) in order to emulate different organizations in collaborative studies.

V.2.6.2 Model Accuracy

First and foremost, we want to ensure that our proposals are scientifically sound and reliable. To do so, we examine the model quality (as measured by accuracy of coefficient estimates and subsequent predictions) estimated from PrivLogit-Hessian and PrivLogit-Local. The standard non-secure distributed Newton method serves as the ground truth. Our hypothesis is that despite the significant change in our numerical optimizer and reliance on cryptographic operations, the accuracy of our model estimation (measured in coefficients β) should still be guaranteed.

Numerical results have confirmed our hypothesis, as is illustrated in the QQ-plots in Figure V.6. Specifically, our β coefficient estimates from PrivLogit-Hessian and PrivLogit-Local are in perfect alignment with the ground-truth Newton across all studies, with correlation $R^2 = 1.00$ (perfect correlation). Since there is deterministic mapping between β and prediction target $\mathbf{y}$ for given input data $\mathbf{X}$ and our coefficient estimates are exactly the same as the ground-truth, we omit the comparison result in terms of model prediction performance for brevity.

This implies that the approximate Hessian adaption we introduced in PrivLogit does not affect model quality and is scientifically reliable. Moreover, it also confirms that the various cryptographic protections underlying PrivLogit-Hessian and PrivLogit-Local have no influence on the model quality. All such observations are in accordance with our earlier analytical evidence.

V.2.6.3 Computational Performance

Next, we evaluate the computational performance of PrivLogit-Hessian and PrivLogit-Local in terms of model convergence with respect to iterations count and total runtime. We partition each evaluation datasets into 4$\sim$20 blocks horizontally (i.e., by rows) to emulate different data-contributing organizations. As it has been demonstrated both analytically

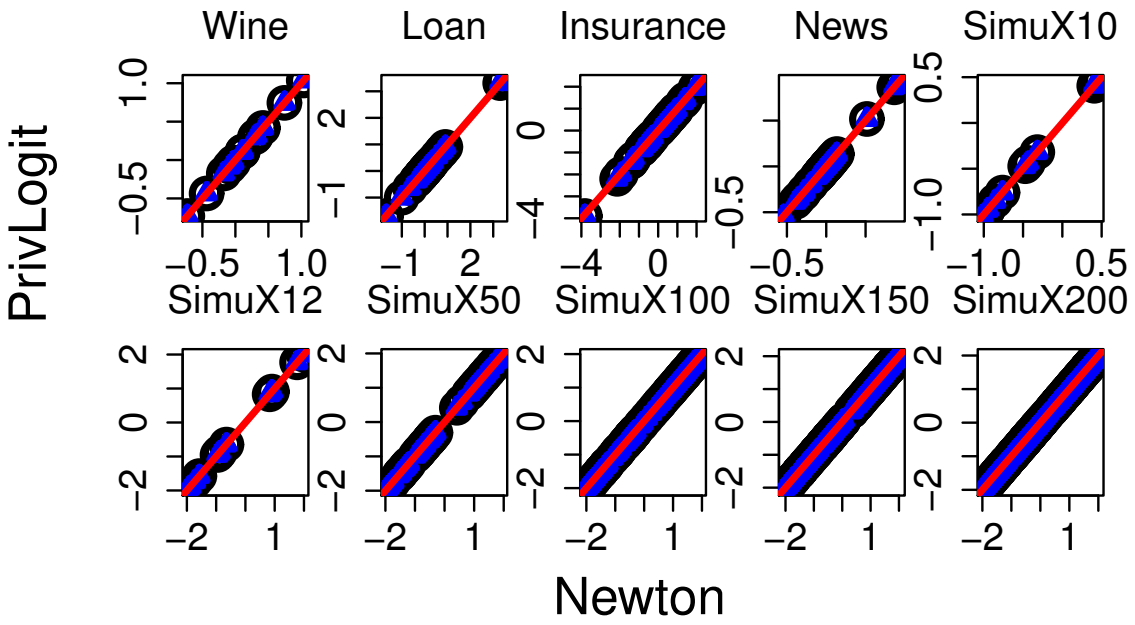
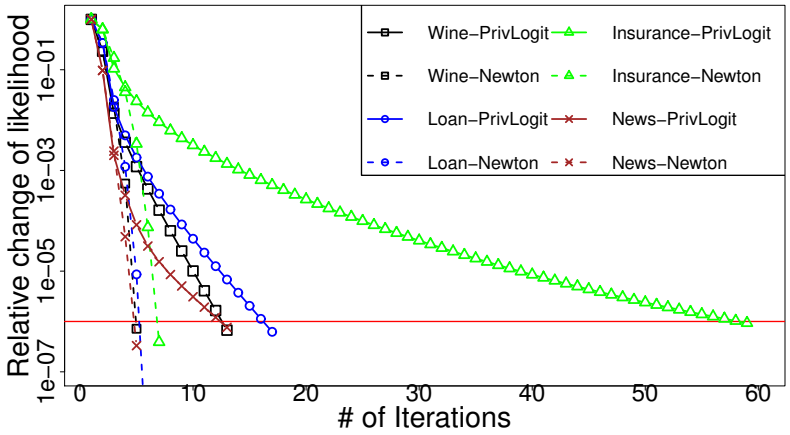


Figure V.6: QQ-plot comparison of coefficients estimated by PrivLogit-Hessian and PrivLogit-Local vs. that by baseline Newton, across various datasets. PrivLogit-Hessian (in black) and PrivLogit-Local (in blue) points overlap significantly.

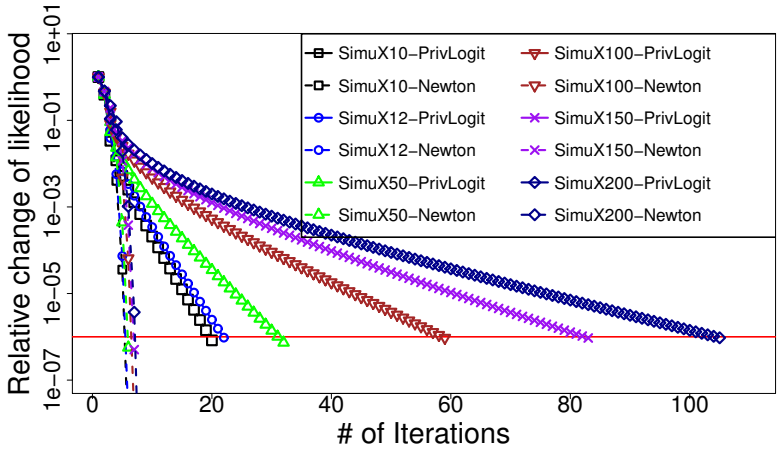
and empirically that cryptographic protections do not affect the accuracy of computation in our case, we refer to our two secure protocols as PrivLogit in general for simplicity. Our model convergence threshold is set at 10^{-6}, as mentioned earlier.

V.2.6.3.1 Iterations to convergence

As is illustrated in Figure V.7, all protocols managed to converge within a reasonable number of iterations. For instance, the *Loans* study (of dimension: $122,578 \times 33$) requires 6 and 17 iterations, respectively, to converge for the Newton and our PrivLogit-based secure proposals. For the smaller *Insurance* study, it takes 7 (for Newton) and 59 (for PrivLogit) iterations, respectively. As the data size (especially dimensionality) increases, we observe increases in the number of iterations both for Newton and PrivLogit, with the former growing slower. For instance, *SimuX150* (with 4 millions samples and 150 features) requires 7 iterations for Newton (only 17% increase over *Loans*) and 83 iterations for PrivLogit



(a) Real-world datasets.



(b) Simulated datasets.

Figure V.7: Convergence iterations of PrivLogit and the Newton method baseline on real-world (upper panel) and simulated (lower panel) datasets. Red horizontal line denotes the stopping threshold.

(388% increase over *Loans*).

Judging from model convergence iterations, PrivLogit seems "unfavorable" to Newton, as PrivLogit often requires a few tens of or more iterations, while the latter seems significantly faster with only single-digit number of iterations. The elongated convergence rate is perhaps the main reason why methods similar to PrivLogit have never been considered in the data security and privacy community. However, we will soon refute such a misconception by comparing the total runtime.

V.2.6.3.2 Convergence runtime

Surprisingly, detailed runtime benchmark in Table V.3 manifests that both our secure protocols, i.e., PrivLogit-Hessian and PrivLogit-Local, turn out to be quite competitive in computational performance. For instance, in the *Loans* study, while Newton method takes only 6 iterations, its actual runtime reaches as much as 492 seconds (because of expensive per-iteration computation); On the other hand, despite requiring substantially more iterations (i.e., 17), our PrivLogit-Hessian and PrivLogit-Local protocols only take around 260 and 104 seconds, respectively, leading to 1.9x and 4.7x speedup, respectively. For even larger-scale studies such as *SimuX150*, Newton method takes 42,951 seconds or roughly 12 hours. PrivLogit-Hessian and PrivLogit-Local are respectively 1.7x and 7.1x times faster than Newton in this case.

One interesting observation is that in rare occasions, PrivLogit-Hessian can be slightly slower than Newton. For instance, the *Insurance* study requires around 843 seconds for Newton (for 7 iterations), but 978 seconds (1.16x slower) for PrivLogit-Hessian. This indicates that *directly* applying PrivLogit (i.e., PrivLogit-Hessian) does *not* guarantee improvement. Our second protocol, PrivLogit-Local, however, always outperforms Newton with dramatic speedup: requiring only 144 seconds (5.9x speedup).

Overall, PrivLogit-Local constantly outperforms other methods with significant speedup, while PrivLogit-Hessian is generally faster than Newton most of the time.

Furthermore, we also test on datasets with dimensions as high as 400, a scale that has never been tested before for privacy-preserving logistic regression. Unfortunately, only PrivLogit-Local converges within reasonable time (110,598 seconds or roughly 1.28 days; for 206 iterations). The other two protocols still did not complete after 4 days. While PrivLogit-Hessian did not complete, its convergence iterations is expected to be the same as PrivLogit-Local (i.e., 206 iterations). For Newton method, a non-secure implementation requires 8 iterations.

Table V.3: Model convergence iterations (*Iter.*) and runtime (in seconds) benchmark for Newton (*Ntn.*), PrivLogit (*Priv.*), PrivLogit-Hessian (*-Hessian*), PrivLogit-Local (*-Local*).

Dataset	Iterations Ntn. (Priv.)	Time Ntn.	Time -Hessian	Time -Local
Wine	5 (13)	32	24	17
Loans	6 (17)	492	260	104
Insurance	7 (59)	843	978	144
News	5 (13)	1442	621	313
SimuX10	6 (20)	26	24	13
SimuX12	6 (22)	38	37	17
SimuX50	6 (32)	1549	1052	383
SimuX100	7 (59)	13138	7817	1807
SimuX150	7 (83)	42951	25030	6055
SimuX200	8 (105)	114522	56917	14105
SimuX400	8 (206)	N/A	N/A	110598

V.2.6.3.3 Relative speedup

To better demonstrate the relative performance of PrivLogit-Hessian and PrivLogit-Local over existing secure Newton methods, we extensively benchmark the relative speedup of our methods over the baseline Newton. As illustrated in Figure V.8, PrivLogit-Hessian outperforms Newton most of the time (except for one occurrence of *Insurance*), and the speedup is between 1.03x$\sim$2.32x. For PrivLogit-Local, the speedup is even more striking, with a speedup of up to 8.1x. For small datasets, PrivLogit-Local is around 2x faster than Newton; for medium datasets such as *Loans, Insurance, News*, its speedup is around 4x$\sim$6x. The largest increase in relative performance is from PrivLogit-Local on the *SimuX200* dataset, with 8.1x speedup. PrivLogit-Hessian also performs well, with 2x speedup. In general, as data dimension increases, we see much more relative efficiency gain for both PrivLogit-Hessian and PrivLogit-Local.

Overall, this provides further evidence that our secure PrivLogit proposals have better performance compared to state-of-the-art privacy-preserving distributed Newton method, and our relative competitive advantage increases along with data scale. This indicates that our methods hold much better potential for large-scale studies in the big data era.

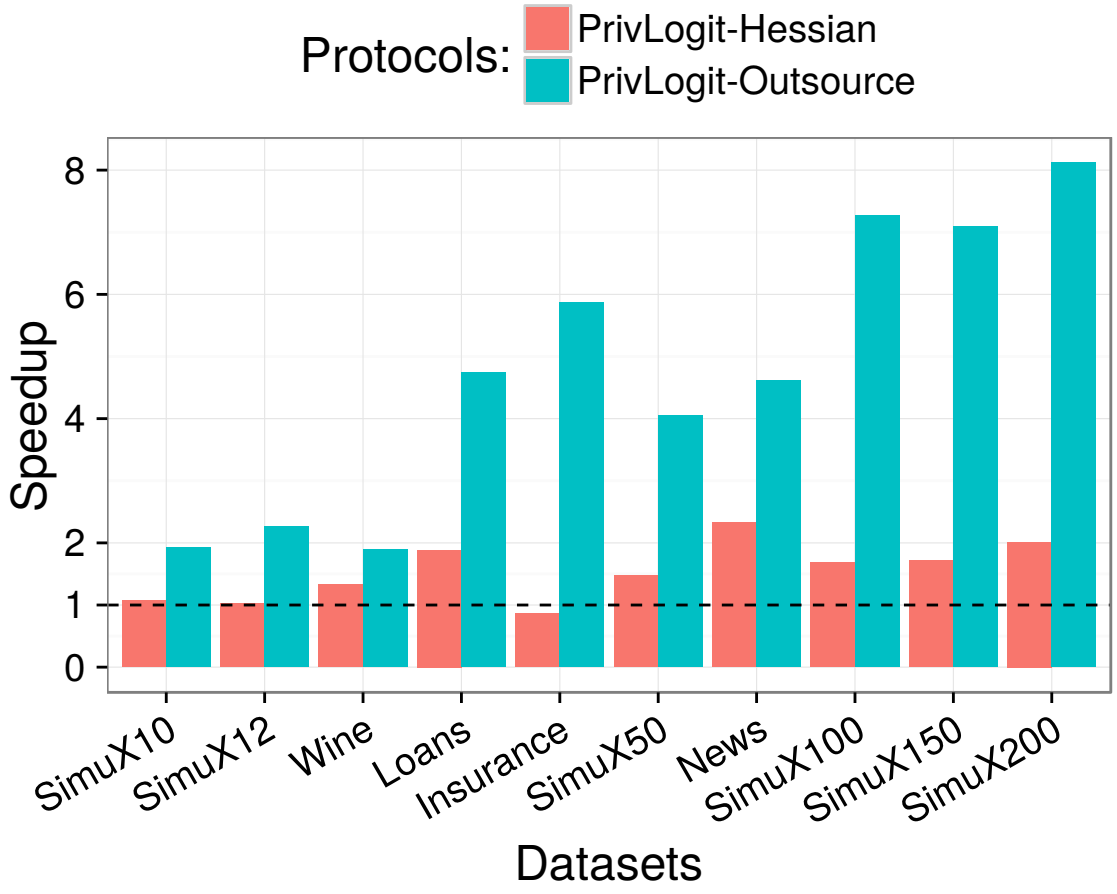


Figure V.8: Relative speedup of PrivLogit-Hessian and PrivLogit-Local over the secure distributed Newton baseline (the $y = 1$ line), across various datasets. Our protocols can speed-up the computation by up to 2.32x and 8.1x times, respectively.

V.2.6.4 Model Convergence Guarantee

Another advantage of PrivLogit is its guaranteed convergence to the optimum, a highly desirable property that is absent from Newton method. Newton method is widely known to be sensitive to initial β values. Certain sub-optimal choice of initialization may cause divergence of the method, leading to indefinite iterations.

To compare the convergence of our PrivLogit optimizer and Newton method, we use the *SimuX50* dataset and run a series of random initializations by setting all (dimensions of) $\beta^{(0)} = 0.8, 1, 1.5, or\ 2$, respectively (the converged version of Newton reported before uses 0 as initialization). We report on the Euclidean distance between the coefficient estimates

at each iteration and the ground-truth β. The larger the distance is, the less accurate the model estimation is and less likely to converge.

The superior convergence guarantee of PrivLogit is manifested in Figure V.9, where PrivLogit always converges within a reasonable number of iterations. Newton method, however, diverges significantly starting from the first few iterations and its coefficient estimation is getting worse and worse. Our extensive evaluations on other datasets indicate that the divergence of Newton is not uncommon in practice.

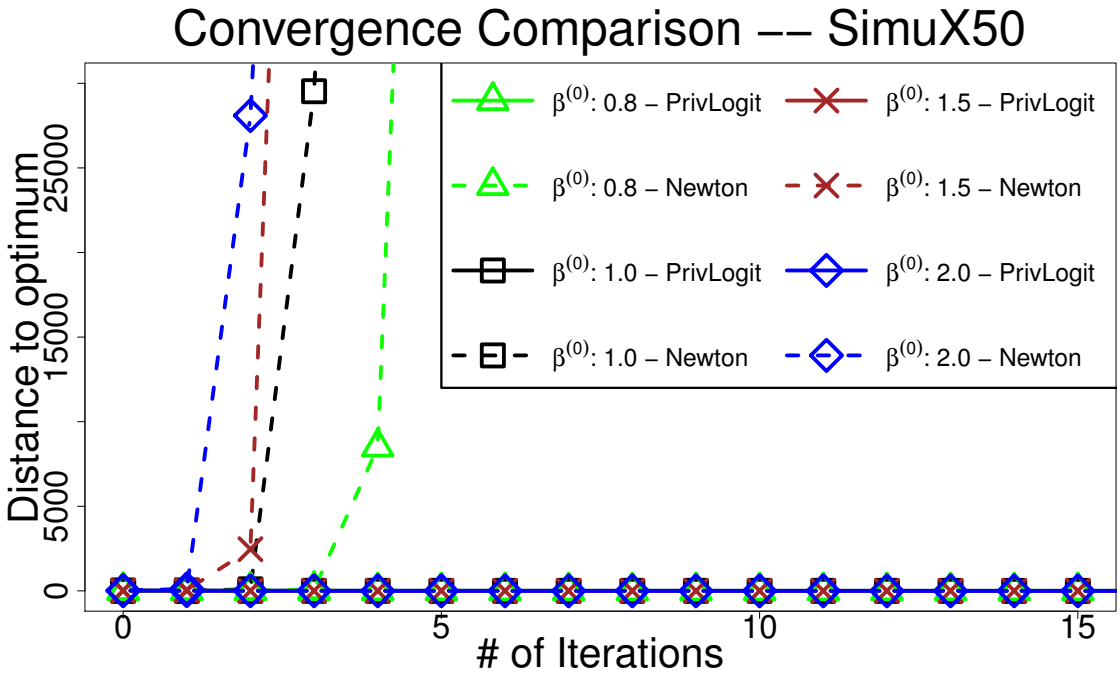


Figure V.9: Model convergence guarantees for PrivLogit and Newton methods under different coefficient initializations. Distance between per-iteration coefficient estimates and the ground-truth is reported (large distance implies inaccurate estimation and thus poor convergence).

V.2.7 Related Works

Privacy-preserving regression analysis and machine learning in general is actively investigated. Here we discuss several closely related lines of research.

V.2.7.1 Cryptographic Protections on Logistic Regression and Other Models.

Extensive efforts have focused on protecting privacy in logistic regression, from centralized solutions [41] to distributed architecture [84, 154, 118, 155]. Due to complexity of securely computing logistic function, many existing proposals compromise on security guarantee by providing no or only weak protections over intermediate summary data [154, 155, 93], which can be problematic given various inference attacks [124, 159, 41]. Other works approximate the logistic function, resulting in accuracy loss [118, 41, 7]. Nearly all existing works directly apply mainstream model estimation algorithms (i.e., Newton method) without customization. Our proposal, however, provides a secure computing-centric perspective, and proposes an optimizer that significantly outperforms alternatives while guaranteeing accuracy.

Hessian approximation was briefly explored by [118], but without justification or even (comparative) performance evaluation. Our results show that direct application of the method does not necessarily lead to better performance, and even when it does, the improvement is modest. In addition, for datasets of size $n \times p$, Newton method has per-iteration complexity $O(np^2 + p^3)$ (where the first term is dominating the cost). And the main improvement of approximate Hessian is by limiting the first term np^2 to one occurrence only (as in [118]). However, our use case is different as our local-organization computation is privacy-free (i.e., independent from sample size n) and total cost is only determined by the second term $O(p^3)$, making it not obvious of the benefits of approximate Hessian. In fact, there is no performance guarantee if *directly* adopting approximate Hessian in our situation.

Cryptography is also widely used to safeguard linear regression, association rule mining, and other data mining tasks. It is known as privacy-preserving (distributed) data mining (partially reviewed by [4]).

V.2.7.2 Perturbation-based Privacy Protection.

Perturbing data via artificial noise is also a popular technique for privacy preservation (e.g., $k-$anonymity, differential privacy). However, since such methods inherently change the data and computation, their results may no longer be scientifically valid and thus are not widely accepted in practice. In addition, they do not protect the computation process.

V.2.7.3 Improved Numerical Optimization for Regression.

Numerical optimization for regression analytics is under extensive investigation. These include various efforts to approximate or eliminate the Hessian from Newton-style optimizers, such as the Quasi-Newton or Hessian-free optimization (e.g., BFGS and L-BFGS [99]). However, none of them have seen adoption in data security and privacy research, partially because they are heavily tailored for privacy-free scenarios and often data-dependent and thus difficult for cryptographic implementation. Hessian approximation was described in the 1980s for maximum likelihood (in privacy-free applications) [16], but only with limited adoption in practice maybe due to their not-obvious efficiency improvement for privacy-free settings.

V.2.8 Discussion

In PrivLogit-Hessian and PrivLogit-Local, the network bandwidth and transmission cost is small, since the encrypted summary information exchanged has very minimal size even for large studies, especially given that Hessian only needs to be preprocessed once. Since these factors are already accounted for in the total runtime benchmark, we omit detailed discussion for brevity.

The PrivLogit optimizer is designed for secure computing in general and agonostic of specific cryptographic schemes. PrivLogit-Hessian can be further accelerated using more efficient schemes, given that the computation is simplified. However, since we aim to provide a direct comparison with state-of-the-art based on the same secure primitives, we leave it as future work to explore alternative schemes.

While our work focuses on logistic regression, our proposal of tailoring optimizers for secure computing is widely applicable to privacy-preserving machine learning, as mainstream (distributed) optimizers are not necessarily competitive for secure computing despite their wide adoption in data security and privacy. We consider extending this novel approach to other statistical models such as other regression problems and classification.

V.2.8.1 Conclusion

We have introduced an improved numerical optimizer (i.e., PrivLogit) and demonstrate its obscure but surprisingly competitive performance for privacy-preserving logistic regression. This contrasts to common wisdom in privacy-preserving data mining which naively applies mainstream numerical optimization methods, which often disregards secure computing-specific characteristics and thus misses valuable opportunities for significant performance boost. Based on PrivLogit, we also propose two secure and highly-efficient protocols for privacy-preserving logistic regression. We validate our proposals extensively using both analytical and empirical evaluations. Results indicate that our proposals outperform alternatives by a significant margin while ensuring privacy and accuracy. Our methods should be helpful for making privacy-preserving logistic regression more scalable and practical for large collaborative studies. And our generic perspective on tailoring optimizers for secure computing should also inspire other research in privacy-preserving machine learning in general.

V.3 QuickLogit: A Novel Paradigm for Efficient Privacy-preserving Logistic Regression

The following section is based on our work in [162]. My contribution in this work includes conception, design and supervision of the study, implementation and experimental evaluation, analysis of results, writing the manuscript and addressing reviewer comments.

Protecting privacy while supporting machine learning on human subject data is highly desirable in basic sciences, especially when data are naturally distributed or decentralized among different organizations (such as in multi-site consortia-based investigations). Privacy-preserving machine learning has benefited significantly from distributed machine learning in general. Nearly all state-of-the-art methods for privacy-preserving distributed data mining directly apply distributed machine learning algorithms. While progress is impressive in this domain, we question the common practices of privacy-preserving distributed machine learning, and ask a fundamental question: is state-of-the-art performance guaranteed in the secure setting by directly building on mainstream distributed machine learning algorithms? Our work answers the question in the negative. We provide contrasting insights to the problem, and propose a novel paradigm (called QuickLogit) to privacy-preserving logistic regression in the multi-site distributed learning setting, which drastically improves performance (faster by 3x to 6x or more) over standard practices of directly building on distributed algorithms. We show our our superior performance both theoretically and empirically over multiple large-scale studies.

V.3.1 Introduction

Data sharing and joint statistical analytics in a distributed system consisting of various independent databases (belonging to different institutions) is widely used in many "small data" domains, such as biomedical and social sciences. The goal of multi-institution collaborative studies is to accumulate large sample sizes across institutions and reach more powerful and generalizable statistical conclusions from bigger databases. However, often times such

databases involve privacy-sensitive human subject data, sharing of which across different institutions often require complicated and time-confusing legal and ethical reviews.

Privacy-preserving distributed machine learning (or data mining) [4] is a popular research endeavor to solve the challenge, which leverages distributed algorithms and cryptography such as secure multi-party computation (SMC) to support machine learning while protecting privacy. Recently, there is a resurgence of privacy-preserving machine learning research, mainly due to their increasing adoption in various practical applications of multi-institution investigations where there is increasing tension between privacy and social good (such as in human genetics [43, 76], smart grid, and healthcare).

Despite encouraging progress, privacy-preserving (distributed) machine learning is frequently criticized for its significant computational overhead. Inefficiency remain the biggest bottleneck in wide-spread adoption of privacy-preserving machine learning protocols.

In this work, we propose a novel and generic paradigm for privacy-preserving distributed machine learning, which deviates significantly from common practices of the community which are directly based on distributed machine learning formulations. The main merit of our proposal is to significantly accelerate privacy-preserving distributed machine learning (the main hurdle of the field), by leveraging local models from distributed nodes. To keep our presentation concrete, this work will primarily focus on logistic regression as a representative machine learning and statistical model throughout this work. Logistic regression is widely adopted in various domains and often the primary model for biomedical and social sciences. The model is also fairly complex and its model estimation process (second-order optimization) and cryptographic implementation are representative for privacy-preserving machine learning in general.

Our work begins by questioning the common practice of heavily relying on (privacy-free) distributed machine learning formulations that has dominated the field of privacy-preserving distributed machine learning for over a decade. In particular, we pose a fundamental question: is competitive performance guaranteed by directly following distributed

machine learning formulations? Unfortunately, our works answers the question in the negative.

Our results indicate that we have constantly neglected performance shortcuts that are unique only to privacy-preserving distributed machine learning, but not present in generic privacy-free distributed settings. This means that by deviating from standard distributed machine learning formulations and customizing protocols/workflows specifically for *privacy-preserving* distributed machine learning, we can gain unexpected (and significant) performance.

V.3.1.1 Contributions and Outlines

Our main contributions are as follows:

- We propose a new paradigm for privacy-preserving distributed machine learning, by leveraging the unique performance shortcuts in the secure setting.
- We propose a novel and significantly accelerated method, called QuickLogit, for privacy-preserving logistic regression as demonstration of the aforementioned paradigm.
- We provide extensive theoretical and empirical support.

This manuscript is organized as follows: background information about logistic regression is introduced in Section V.3.2. We then introduce the theories and secure implementation in Section V.3.3. Later it is followed by in-depth theoretical analysis regarding the convergence performance of QuickLogit in Section V.3.4. We empirically evaluate our proposal in Section V.3.5. Lastly, we discuss related works in Section V.3.6 and conclude in Section V.3.7.

V.3.2 Preliminaries

We briefly review logistic regression, our representative model for machine learning in this work.

Throughout this work, we follow the main notations summarized below in Table V.4.

Table V.4: Main notations.

Notations	Description
$X \in \mathbb{R}^{n \times p}$	Regression covariates: n samples, p features
$y \in \mathbb{R}^{n \times 1}$	Regression responses: n samples
$\beta \in \mathbb{R}^{p \times 1}$	Regression coefficients: p features
$\mathbf{H} \in \mathbb{R}^{p \times p}$	Hessian matrix
$\mathbf{g} \in \mathbb{R}^{p \times 1}$	Gradient
$\ell(\beta)$	Likelihood (of logistic regression)
$E(.)$	Encryption of data

V.3.2.1 Logistic Regression

Logistic regression is a probabilistic model popular for binary (i.e., categorical) outcomes classification. It is one of the most adopted statistical models in various applications, including biomedicine and genetics, psychology and other social sciences, and internet industry. For a single record, logistic regression model is represented by:

$$p(y = 1|\mathbf{x}; \beta) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}} , \tag{V.17}$$

where $p(.)$ denotes the probability of the binary response variable y equal to 1 (or a nominal class label), $\mathbf{x}$ is a d-dimensional covariates (or features) for a specific data record/sample, and β is the regression coefficients we want to estimate during model training.

Because logistic regression does not have closed form solution, in order to estimate regression coefficients β, we need to perform (iterative) numerical optimization on the objective function of the model. The optimization objective of logistic regression is of the form:

$$\ell(\beta) = \sum_{i=1}^{n} [y_i(\beta^T \mathbf{x_i}) - \log(1 + e^{\beta^T \mathbf{x_i}})] \tag{V.18}$$

V.3.2.2 Distributed Newton Method

Newton method is an iterative optimization method and is the standard model estimation approach for logistic regression, with implementations in various software packages. It

is often the default optimizer whenever it is applicable because of its fast quadratic convergence rate. Our proposal also utilizes the distributed version of Newton method as a building block, similar to various privacy-preserving logistic regression protocols [154, 155, 93].

At each Iteration $(t+1)$ of Newton, the new model estimate is updated by:

$$\beta^{(t+1)} = \beta^{(t)} - \mathbf{H}^{-1}(\beta^{(t)})\, \mathbf{g}(\beta^{(t)}) \,, \tag{V.19}$$

where $\mathbf{H}(\beta^{(t)})$ and $\mathbf{g}(\beta^{(t)})$ denote the Hessian and gradient of the objective $\ell(\beta)$ (Equation V.18) evaluated at the current β coefficient estimate. And superscripts (t) and $(t+1)$ index the t^{th} and $(t+1)^{\text{th}}$ iterations, respectively. This updating process iterates until model convergence, as determined by the relative change of log-likelihood against a predefined threshold (e.g., 10^{-6}):

$$\frac{|\ell^{(t+1)} - \ell^{(t)}|}{|\ell^{(t)}|} < 10^{-6} \,, \tag{V.20}$$

where $\ell^{(t+1)}, \ell^{(t)}$ correspond to the log-likelihood of logistic regression for Iterations $(t+1), (t)$, respectively.

Following Equation V.18, the gradient and Hessian for logistic regression can be computed following a distributed formulation (partitioned by S institutions):

$$\mathbf{g}(\beta) = \nabla_\beta \ell(\beta) = \mathbf{X}^T(\mathbf{y} - \mathbf{p}) = \sum_{j=1}^{S} \mathbf{g_j}(\beta) \,, \tag{V.21}$$

$$\mathbf{H}(\beta) = \nabla^2_\beta \ell(\beta) = -\mathbf{X}^T\mathbf{A}\mathbf{X} = \sum_{j=1}^{S} \mathbf{H_j}(\beta) \,, \tag{V.22}$$

where $\mathbf{A} \in R^{n \times n}$ is a predefined diagonal matrix with elements $a_{i,i} = p_i(1-p_i)$; $\mathbf{g_j}(\beta)$ and $\mathbf{H_j}(\beta)$ denote the per-institution gradient and Hessian, respectively, computed using local-institution data $(\mathbf{x_j}, \mathbf{y_j})$; S is the total number of organizations contributing data to the collaborative study.

Note that the log-likelihood (optimization objective) V.18 can also be decomposed per-institution (we use ll_j to denote per-institution likelihood):

$$\ell(\beta) = \sum_{j=1}^{S} ll_j(\beta) \tag{V.23}$$

V.3.2.3 Common Workflow of Privacy-preserving Distributed Machine Learning

Privacy-preserving distributed machine learning has greatly benefited from, and is thus heavily influenced by, (generic privacy-free) distributed machine learning. As a result, nearly all existing proposals in the field directly adopt distributed machine learning formulations and apply cryptographic protection on them [118, 4]. This pervasive workflow for privacy-preserving distributed machine learning (and logistic regression) can be summarized as follows:

(a) Given some model estimates, local institutions (resembling distributed nodes in distributed machine learning) compute (sufficient) summary statistics from their respective private databases. In logistic regression, these include local gradient and Hessian summaries.

(b) Local institutions apply cryptographic protection respectively and share encrypted summary statistics with a semi-honest Analysis Center;

(c) Analysis Center securely 1) aggregates all per-institution summaries, and 2) performs global model fitting and updates model estimate. In Newton method for logistic regression, this means securely deriving global gradient and Hessian, and performing Newton updating on β.

(d) Use the updated model estimates as the new initialization, and repeat previous Steps 1 to 3 in iterations (for iterative numerical optimization), until model is converged and final estimation derived.

This is also the workflow of the popular privacy-preserving (distributed) Newton method, which underlies various logistic regression protocols [93, 118].

V.3.3 QuickLogit: accelerating performance using local models

This section will focus on our novel paradigm for efficient privacy-preserving (distributed) logistic regression. We begin with discussion of the limitations of common practice of privacy-preserving machine learning that relies heavily on distributed algorithms. This motivates us to design drastically different approaches that can significantly improve performance.

V.3.3.1 Problems of Traditional Approaches

The common practice of coupling distributed machine learning with cryptography has worked well for privacy-preserving distributed machine learning, and is the standard approach for the past decade (Section V.3.2.3).

However, despite its popularity, this common practice is not necessarily the optimal approach to the problem in terms of computational efficiency (perhaps the biggest obstacle to practical deployment of privacy-preserving protocols). This is because distributed algorithms have been primarily customized for generic privacy-free settings, in which computations exhibit similar complexity patterns across all computing nodes (despite distributed nodes or center) and all computing servers are of comparable computational power. In cryptography-based secure settings (e.g., multiple local institutions and a semi-honest center), however, the scenario is drastically different. Here, the center is not trusted (for any privacy-sensitive computation or data, including summary statistics). Thus all its data and computations need to occur using cryptography. This incurs orders of magnitudes more computational overhead at the center (compared with privacy-free scenarios), whereas local-institution-based computing is extremely fast and simply generic privacy-free computations (because they have unrestricted access to their local data). This unique computational asymmetry in secure settings between the center and local nodes is not present in

generic privacy-free scenarios, thus giving rise to a long missed opportunity to significantly improve performance by going beyond traditional distributed machine learning-based formulations.

V.3.3.2 Our Novel Paradigm Leveraging Local Models

Our novel paradigm to privacy-preserving distributed machine learning is customized to leverage the aforementioned computational asymmetry between local nodes and the center in the secure settings. The ultimate goal is to significantly improve performance without affecting model accuracy, making related protocols more practical.

On a high level, we propose a two-phase computation process in our novel paradigm:

- **Phase 1 (Local Models)**: Local institutions independently estimate their *complete* models (instead of intermediate summary statistics) based on their private databases, respectively.

- **Phase 2 (Global Refinement)**: The Analysis Center securely aggregates these local models, and use globally aggregated model to initialize the iterative numerical optimizer (such as the secure distributed Newton method). The center and local institutions then follow the traditional approach of privacy-preserving distributed machine learning to iteratively refine the global model.

Our proposal guarantees exact model accuracy as with standard Newton method (i.e., without approximations), while significantly reducing the total computation by several times.

The main merit of our new paradigm stems from utilizing many local (sub-optimal) models to better initialize and guide the subsequent iterative numerical optimization. As we will show later both theoretically and empirically, this provides significantly better-informed initialization to the optimizer, and can shortcut the vast majority of optimization iterations. As a comparison, traditional approaches in privacy-preserving distributed machine learning often start with some random initialization (e.g., 0 as recommended) to the

model estimates, which may turn out to be arbitrarily far way from the optimal model and a lot of trial-and-error iterations are necessary.

V.3.3.2.1 A Geometric Intuition

Our novel paradigm has very intuitive geometric explanation, as illustrated in the optimization contour Figure V.10. Traditional Newton method (denoted in red) starts with some random model initialization x_0 (which may be far away from the optimum at the core), and make iterated progress towards the optimum.

Our proposal (denoted in green), however, aggregates various local-institution models to guide our initialization. It has been proven later that our aggregated initialization (q_0 in Figure V.10) can quickly reach the vicinity of the optimum, thus circumventing a large number of intermediate iterations and significantly accelerating the overall computation.

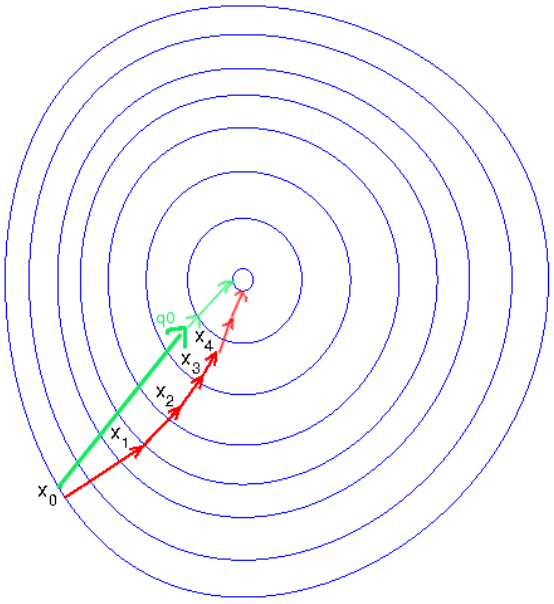


Figure V.10: Convergence path of Our Approach (green) vs. Newton (red). Ours can directly reach the vicinity around optimum in the very first iteration and quickly converge to optimum afterwards.

V.3.3.3 QuickLogit: A Novel Approach to Privacy-preserving Logistic Regression

Our novel proposal, QuickLogit, for efficient privacy-preserving distributed logistic regression is based on our two-phase paradigm just introduced (Section V.3.3.2). It is designed with two goals in mind: 1) Significant performance improvement; 2) Same model accuracy as traditional and non-secure solutions.

A high-level description of QuickLogit is presented in Algorithm 10. The two main phases of the algorithm is detailed later in Algorithms 11 and 12 and also in the following two Sections V.3.3.4 and V.3.3.5.

Overall, as demonstrated in Algorithm 10, our QuickLogit protocol closely follows the aforementioned two-phase paradigm, with the first phase a one-shot attempt and the second typically an iterative process. In the first phase (*Local Models*), the goal is to derive a good central initialization to model estimate, by securely leveraging multiple local-institution models (Step 1 in Algorithm 10). This is jointly completed by the local institutions and the computation center using one single interaction. This approximate initialization is denoted as $E(\bar{\beta})$ in its encrypted form (encryption is performed by local institutions and decryption is not accessible to the center). We point out that this step occurs only once.

Algorithm 10 QuickLogit: efficient privacy-preserving logistic regression.

Output: Globally fit coefficient estimate β

[At local Nodes and Center; one-step only]

1: $E(\bar{\beta}) = BetterInitialization()$

[At local Nodes and Center; a few iterations]

2: $E(\beta) = NewtonRefinement(E(\bar{\beta}))$

3: **return** $E(\beta)$

The second phase of QuickLogit, as listed in Algorithm 10, aims to refine model estimation centrally and securely. This essentially means that we start with the (better-informed) initialization provided by the previous phase, and simply invoke involves additional interactions and perhaps more local-institution information, and takes similar approaches as traditional privacy-preserving distributed logistic regression protocols. We emphasize,

however, that this process in QuickLogit requires far fewer iterations and is thus significantly faster, as will be demonstrated later both theoretically and empirically.

Below, we will introduce the detailed steps for both phases.

V.3.3.4 Phase 1: Local Models

As mentioned before, the goal of the first phase is to locally estimate models independently, so that they can be aggregated centrally later to approximate the global model. As demonstrated in Algorithm 10, we first have each local institutions estimate their locally optimal models β_j (where $j \in S$ institutions) (Step 2). Since there is no privacy concern (institutions can freely access their own data), any standard statistical software is applicable. These local models will be encrypted locally using advanced cryptography (e.g., Paillier partially homomorphic encryption [125] in our implementation) before sharing with the Analysis Center. The encryptions are denoted as: $E(\beta_j)$

Later, local model encryptions will be securely transmitted to the Center. The Center will construct an improved global model approximation by securely aggregating all local models (encryptions) $E(\beta_j)$. There are various approaches for model aggregation, and in our case, simple linear averaging (over S institutions) suffices (Step 5). We denote this globally aggregated model estimate as (in encrypted form): $E(\bar{\beta})$, which will serve as a good initialization estimate in the second phase.

Algorithm 11 *BetterInitialization()*: leveraging local models for better Newton initialization.

Output: Globally aggregated model (encryption) $E(\beta)$

[At local institutions]:

1: **for** each institution j = 1 **to** S **do**
2: Learn local model: β_j
3: Encrypt and securely transmit to Center: $E(\beta_j)$
4: **end for**

[At Center]:

5: Securely compose initialization: $E(\bar{\beta}) = E(\frac{1}{S}\sum_i^S \beta_j)$
6: **return** $E(\bar{\beta})$

V.3.3.5 Phase 2: Global Model Refinement.

The second phase of QuickLogit is similar to the widely-used privacy-preserving distributed Newton method, except for the use of our better-informed model initialization instead of random initialization as commonly pursued. As discussed in Section V.3.2.2, the privacy-preserving distributed Newton method is widely-adopted in the security/privacy community.

Algorithm 12 illustrates how we apply Newton method in QuickLogit. The first step highlights the main difference: in the first Newton iteration (i.e., when $t = 0$), we initialize our model estimation with our centrally aggregated model $E(\bar{\beta})$ from the Phase 1 (Section V.3.3.4). All subsequent Newton iterations are following exactly the traditional approaches (Section V.3.2.2). For completeness, we describe the workflow of privacy-preserving distributed Newton method in our case.

Algorithm 12 *NewtonRefinement($E(\bar{\beta})$)*: refining model estimation via Newton method.

Input: Model initialization $E(\bar{\beta})$ (for Newton)
Output: Globally optimal model β

Initialize Newton: $E(\beta^{(0)}) = E(\bar{\beta})$
while regression model not converged (Iteration t) **do**
  [At local institutions] :
  for each institution j = 1 **to** S **do**
    Compute summary statistics: $\mathbf{g}_j$, $\mathbf{H}_j$, ll_j
    Send encrypted summaries to Center: $E(\mathbf{g}_j)$, $E(\mathbf{H}_j)$, $E(ll_j)$
  end for

  [At Center] :
  Securely aggregate gradient: $E(\mathbf{g}) = E(\sum_j \mathbf{g}_j)$
  Securely aggregate Hessian: $E(\mathbf{H}) = E(\sum_j \mathbf{H}_j)$
  Securely aggregate likelihood: $E(ll) = E(\sum_j ll_j)$
  Secure Cholesky decomposition (for inversion): $E(\mathbf{L}) = SecureCholeskyDecomposition(E(-\mathbf{H}))$
  Secure back-substitution (for inversion): $E(\mathbf{H}^{-1}\,\mathbf{g}) \leftarrow E(\mathbf{L}), E(\mathbf{g})$
  Secure model updating (Newton): $E(\beta^{(t+1)}) \leftarrow E(\beta^{(t)})$ (Equation V.19)
  Secure comparison to check model convergence (Equation V.20)
end while
Return β (latest $\beta^{(t+1)}$)

As illustrated in Algorithm 12, at each Newton iteration (Steps 3 to 13), two main types of computations occur (in sequential): 1) local-institution-based preparation of summary statistics (privacy-free computation; Steps 3 to 6); 2) Center-based secure aggregation of summaries and Newton model updating (Steps 7 to 13). This will iterate until model convergence is reached (Step 13; Section V.20).

V.3.3.5.1 Local-institution Summary Statistics in Newton

At each Newton iteration, for each local institution j (out of a total of S institutions), it needs to compute its local gradient $\mathbf{g_j}$, Hessian $\mathbf{H_j}$, and likelihood ll_j (for model convergence check) (Step 4 in Algorithm 12). Since local institutions have unrestricted access to their respective private data $(\mathbf{x_j}, \mathbf{y_j})$, these local computations simply follow generic arithmetics without cryptography involved (thus are extremely fast).

Such summaries will then get encrypted, and shared with the untrusted Analysis Center (Step 5). We emphasize that only the local institutions hold the decryption (private) key, thus only them (not the Center) have decryption capabilities.

V.3.3.5.2 Central Aggregation and Model Fitting in Newton

Once the Analysis Center receives summary statistic encryptions from local institutions, it will securely aggregate them (mostly leveraging secure summation primitives) to construct global summaries for gradient $E(\mathbf{g})$ (Step 7), Hessian $E(\mathbf{H})$ (Step 8), and likelihood $E(ll)$ (Step 9).

According to the main step of Newton method (Equation V.19), to derive $\mathbf{H^{-1}g}$, we need to securely perform matrix inversion and multiplication. This is often achieved (more efficiently) by Cholesky decomposition on $(-\mathbf{H})$, followed by back-substitution. Both are standard textbook algorithms and have been used in the security/privacy community. Specifically, in Step 10, we securely perform Cholesky decomposition with encrypted input $E(-\mathbf{H})$, such that: $\mathbf{L}\mathbf{L}^T = -\mathbf{H}$.

Later, secure back-substitution is performed to obtain $E(\mathbf{H}^{-1}\mathbf{g})$, using Cholesky decomposition result $E(\mathbf{L})$ and $E(\mathbf{g})$ as inputs (Step 11).

Now secure Newton model updating is straightforward following Equation V.19 and secure summation (Step 12).

After that, we use secure comparison over encrypted likelihoods to check model convergence, following Equation V.20. The above procedure will proceed until convergence is reached.

V.3.3.6 Security Guarantees and Information Disclosure

Here we briefly analyze the security guarantees of our proposal. We mainly discuss the information disclosed in our protocol, and assess their privacy implications. Our demonstrate is built upon existing secure schemes and subprotocols with well established security guarantees, so the main components are provably secure. Potential information disclosure mainly occurs in the bridging between different schemes or organizations, including the transition between different schemes, and the different privacy definitions between different parties.

V.3.4 Theoretical Proof

In this section, we first analyze the model convergence properties of our QuickLogit proposal in terms of convergence guarantees and rate. We then prove that our proposal guarantees much faster speed and better model quality in practice.

V.3.4.1 Same Theoretical Convergence as Newton

Our QuickLogit can essentially be regarded as Newton method with a carefully chosen (instead of random) initialization. This means that it should at least share the same theoretical convergence properties as Newton method, whose convergence is well established in literature. We thus briefly summarize the convergence of QuickLogit below (similar to Newton method) (interested readers are encouraged to read related proof from classical numerical

optimization textbooks [19, 104]):

(a) QuickLogit can often converge to the global optimum.

(b) QuickLogit has quadratic convergence rate.

V.3.4.2 Better Practical convergence than Newton.

The aforementioned equivalence of convergence between QuickLogit and Newton is mainly from a sense of Big-O notation (i.e., order of magnitude). In practice, our proposal tends to be much better than the Newton method both in terms of model quality and the amortized rate of convergence. This is because our carefully chosen initialization proves to have bounded error and is often much closer to the optimum than some random initialization (the latter is commonly pursued by standard Newton method), a condition for Newton to reach its promised convergence. Our approach can potentially address two well-known limitations of standard Newton method: 1) when faced with poor initializations (e.g., random guesses that are arbitrarily far away), Newton method may divert severely and never reach the optimum; 2) with poor initializations, Newton do not guarantee the fast quadratic convergence rate (not at least in the first many iterations).

In below, we provide theoretical evidence to support our claim on the superior convergence of QuickLogit. Briefly, we prove in two steps:

(a) We prove that our carefully chosen initialization has bounded error to the optimum, thus being more likely to fall in the vicinity of the optimum than random initialization.

(b) We prove that Newton method with an initialization nearer to optimum leads to accelerated convergence.

In order to present the statistical properties of our QuickLogit algorithm, we first state the standard regularity assumptions on the parameter space and loss functions, which are commonly encountered in the context of asymptotic statistics [146, 169].

Assumption 1. *The parameter space is a compact convex set.*

Assumption 2. *In the neighborhood of the optimal value β^*, the objective function is locally strongly concave and the Hessian matrix at β^* is negative definite.*

Assumption 3. *The first, second and third partial derivatives of the objective function exist and are bounded. To be more specific, moments of the derivatives are bounded by G, L, and M respectively.*

Assumption 4. *The samples are evenly distributed among S local nodes.*

Assumption 1 to 3 are clearly satisfied in our distributed logistic regression model. Assumptions 4 is assumed mainly to simplify the notation without losing generalization. In practice, multiparty datasets are often of equivalent sample size, rendering this assumption valid.

Under these regularity assumptions, some recent work [169, 133] examines the statistical properties of the averaged estimator and derived the bound of its mean square error. It is shown that the mean square error of the averaged estimator $\bar{\beta}$ is bounded above by

$$E[\|\bar{\beta} - \beta^*\|_2^2] \leq \mathcal{O}(\frac{G^2}{\lambda^2 N} + \frac{G^4 M^2 S^2}{\lambda^6 N^2} + \frac{L^2 G^2 \log(d) S^2}{\lambda^4 N^2}), \tag{V.24}$$

where N is the total number of samples and S is the number of nodes [137]. In addition, since the objective function is Lipschitz continuous with Lipschitz constant L, Equation V.24 implies that the suboptimality is bounded by

$$E[l_2(\bar{\beta})] - l_2(\beta^*) \leq \mathcal{O}(\frac{LG^2}{\lambda^2 N} + \frac{LG^4 M^2 S^2}{\lambda^6 N^2} + \frac{L^3 G^2 \log(d) S^2}{\lambda^4 N^2}) \tag{V.25}$$

Therefore, compared to a random initial value, $\bar{\beta}$ has the advantage of being more likely to

be close to the optimum β^*. In particular, by Markov's inequality, for any $\varepsilon > 0$,

$$\begin{aligned} & P(\|\bar{\beta} - \beta^*\|_2^2 \geq \varepsilon) \\ \leq & \frac{E(\|\bar{\beta} - \beta^*\|_2^2)}{\varepsilon} \\ \leq & \mathscr{O}(\frac{G^2}{\varepsilon\lambda^2 N} + \frac{G^4 M^2 S^2}{\varepsilon\lambda^6 N^2} + \frac{L^2 G^2 \log(d) S^2}{\varepsilon\lambda^4 N^2}) \end{aligned}$$

When N goes to ∞, the probability of $\bar{\beta}$ being far away from β^* will converge to zero. Moreover, if the number of nodes S is in the order of $O(\sqrt{N})$, then $\bar{\beta}$ is a $\sqrt{N}$-consistent estimator of β^* since the MSE of $\bar{\beta}$ will be bounded above by $O(N^{-1})$ from Equation V.24.

For Newton method, it is critical to find an initial value close to the optimal solution. The convergence process of Newton iterates usually falls into two stages [19]. The first stage is called a damped Newton phase, which is quantified by the condition $\|\nabla l_2(\beta)\|_2 \geq \eta$ for some constant η. During this phase, there exists a number $\gamma > 0$ such that objective function value increases by at least γ per iteration,

$$l_2(\beta_{k+1}) - l_2(\beta_k) \geq \gamma$$

When the estimate is close enough to the optimal solution, namely $\|\nabla l_2(\beta)\|_2 \leq \eta$, the optimization process enters the second stage – the quadratically convergent phase, where the estimate β_k converges to the optimal solution β^* quadratically:

$$l_2(\beta_{k+1}) - l_2(\beta^*) \leq C[l_2(\beta_k) - l_2(\beta^*)]^2$$

for some constant C. In conclusion, the number of iterations until $l_2(\beta^*) - l_2(\beta_k) \leq \varepsilon$ is bounded above by

$$\frac{l_2(\beta^*) - l_2(\beta_0)}{\gamma} + \log(\log(\varepsilon_0/\varepsilon)), \tag{V.26}$$

where β_0 is the starting value, γ and ε_0 are constants depending on the objective function.

The second term is usually small due to the quadratic convergence speed. The dominating term is the first term and the closer the starting value is to the optimal solution, the smaller the iteration number during this phase will be. Equation V.25 shows that the suboptimality of averaged estimator is bounded, which implies that the first term in Equation V.26 is also bounded compared to a random starting values β_0. Moreover, this bound will converge to zero as more samples are collected at each institution.

V.3.4.3 Computational complexity

Regarding computational complexity, we observe that QuickLogit has similar Big-O complexity as the standard (distributed) Newton. However, our proposal has drastically lower amortized cost (as determined by a large constant, which directly translates into our speedup). Since secure computing involving cryptography incurs orders of magnitudes more expensive computation than non-secure counterparts, this literally means that our computational complexity is dominated by center-based cryptographic operations which thus becomes our focus of analysis.

Table V.5: Computational complexity of secure subprotocols.

Operation	Complexity	Note
Cholesky decomposition	$O(p^3)$	Hessian inversion
Back substitution	$O(p^2)$	Hessian inversion
Newton method	$O(p^3)$	Per-iteration
QuickLogit	$O(p^3)$	Per-iteration

Specifically, the second stage of our proposal still relies on traditional Newton updating iterations, which has per-iteration complexity of $O(p^3)$ (note that distributed nodes do not have privacy issues for both methods, thus the other term np^2 of standard Newton is reduced here). For QuickLogit, when taking into account the one-time preprocessing for centrally aggregating local models which has complexity $O(p)$ (which is negligible), our total complexity is roughly: $O(p^3 \times \texttt{QuickLogit iterations})$. For the traditional Newton, the complexity is: $O(p^3 \times \texttt{Newton iterations})$. As is obvious, the complexity difference

is mainly due to the number of iterations to model convergence. As has been proven both theoretically and empirically (later), the iterations required by QuickLogit is substantially fewer than that of Newton. This directly translates into significant time saving.

V.3.5 Experiments

Our secure implementations are primarily in Java and Julia [14]. For demonstration, we choose a hybrid of Yao's garbled circuit and Paillier encryption for cryptographic protections, which also underlie various protocols for privacy-preserving logistic regression and machine learning [118, 121]. Roughly, protection of local summary statistics and central aggregation are using Paillier encryption. Center-based complex computations such as model fitting primarily use garbled circuit. We also implement state-of-the-art secure distributed Newton method as baseline. All our secure implementations leverage state-of-the-art software packages (such as ObliVM-GC [98]). We use default security parameters recommended by NIST or ObliVM-GC.

We also point out that our proposed method is agnostic of, and compatible with, different cryptographic schemes from now and future. This is because the only necessary condition for our protocol is the computational complexity asymmetry between center and distributed nodes, which holds true for any cryptographic distributed settings.

The primary goal of our proposal is on efficiency improvement without compromising model accuracy. Thus, our empirical evaluations will concentrate on performance benchmarks, such as the iterations required for model convergence and the total runtime. We also validate the accuracy of our estimated model.

V.3.5.1 Datasets

Our empirical evaluation covers a wide range of simulated as well as real-world studies. We have simulated datasets of varying scales. We also take several common real-world studies from different domains as case studies, and perform in-depth analysis. We briefly describe these datasets below and later in Table V.6.

- Simulated datasets: DataX5, DataX10, DataX20, DataX50, DataX100, DataX200, DataX400, DataX500, DataX1000.

- Real-world studies: 1) *Adult* data for predicting income level (whether it is greater than $50,000 or not) from several societal factors such as demographics; 2) *Loans* data from a popular online lending platform for predicting the loan application status based on anonymized personal profiles.

V.3.5.2 Runtime Benchmarks

First, we demonstrate the main results in terms of performance gain of our proposal. State-of-the-art secure distributed Newton method is used as the baseline. Certain problems are too large in size to solve securely, thus their iteration performance is based on non-secure simulations and highlighted with brackets in Table V.6.

V.3.5.2.1 Significantly reduced number of iterations to convergence.

In Table V.6, we report on the number of iterations required to reach model convergence for QuickLogit and the baseline Newton method. Note that the number of iterations are directly comparable between QuickLogit and Newton due to their similarities (as will be introduced later). It can been seen that QuickLogit requires substantially fewer iterations to convergence across all datasets. For instance, for the *Loans* data, QuickLogit only requires 2 iterations, a significant improvement over 6 iterations of standard Newton. For *DataX100*, despite the large dimensions, QuickLogit still only requires 2 iterations, compared to 11 iterations for Newton (a 5.5x improvement). When the data dimensionality increases, the iterations numbers also gradually increase and the relative speedup slightly drops (as the denominator, i.e., iterations of Newton, becomes larger) but the overall absolute improvement is still growing (due to the increasing complexity of per-iteration cost).

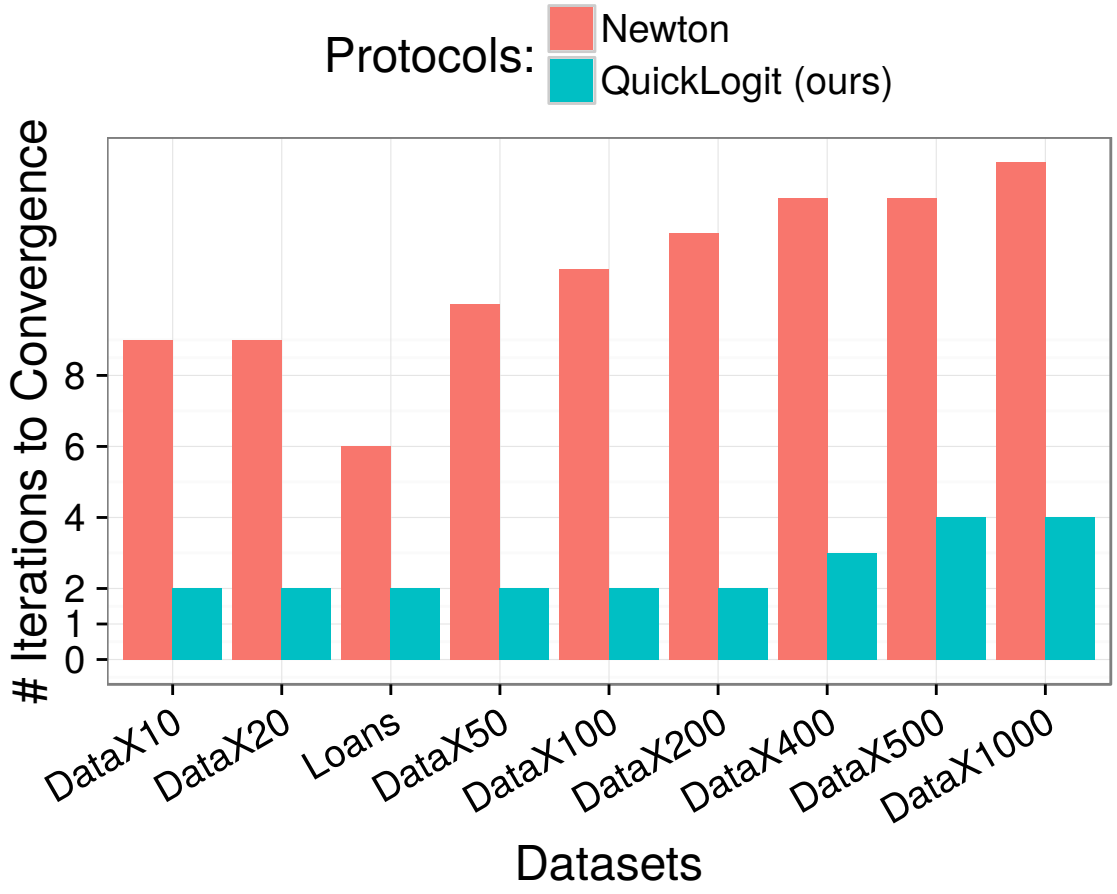


Figure V.11: Comparison of iterations to convergence between QuickLogit (ours) and Newton baseline. Fewer iterations imply faster convergence.

V.3.5.2.2 Dramatic runtime improvement.

Due to the similarity of QuickLogit (second phase) and baseline Newton in terms of the iterative optimization process, the above improvement in convergence iterations would directly translate into significant runtime reduction overall. This is indeed evidenced by Table V.6, where overall runtime is almost proportional to the iteration numbers. Note in addition to the standard iterative Newton updating, QuickLogit has one extra preprocessing step of securely averaging local models (as the Newton initialization). This step only involves trivial secure primitives, secure aggregation, which is very efficient and can be implemented in many SMC schemes. Since the runtime of this step is negligible compared

with the complicated Newton updating, the comparison of runtime between QuickLogit and Newton essentially reduces to that of Newton iteration numbers.

Table V.6: Runtime benchmark (iteration counts and in seconds).

Datasets	Dimension	Iterations (Newton)	Runtime (Newton)	Iterations (QuickLogit)	Runtime (QuickLogit)
Adult	32,561 X 30	7	443.6	2	126.7
Loans	122,578 X 33	6	491.6	2	163.9
DataX10	100k X 10	9	38.3	2	8.5
DataX20	250k X 20	9	197.7	2	43.9
DataX50	250k X 50	10	2582.2	2	516.4
DataX100	250k X 100	11	20645.2	2	3753.7
DataX200	100k X 200	12	171782.6	2	28630.4
DataX400	250k X 400	(13)	(NA)	3	347993.2
DataX500	100k X 500	(13)	(NA)	(4)	(NA)
DataX1000	250k X 1000	(14)	(NA)	(4)	(NA)

To better demonstrate the runtime speedup, we plot the relative speedup in runtime for QuickLogit over Newton baseline in Figure V.12. It can be seen that QuickLogit achieves consistently drastical speedup between 3x to 6x. Overall, QuickLogit is consistently more efficient in iterations, and with speedup by a large factor.

V.3.5.3 Guaranteed Model Accuracy

Many existing privacy-preserving machine learning protocols achieve efficiency improvement by compromising on model accuracy. In our work, however, model accuracy is guaranteed and no approximations are involved. To see this, we directly compare the final model accuracy from QuickLogit and that from the Newton baseline, as illustrated in Figure V.13. Across all evaluations, we observe that QuickLogit provides exact model estimation, with perfect alignment with Newton baseline ($R^2 = 1.00$ and slope of fitted line is 1.00). This provides empirical evidence for our theoretical proof in Section V.3.4.1 regarding model convergence guarantees.

In addition, we also provide empirical results to demonstrate the properties of aggregated models of local-institution models.

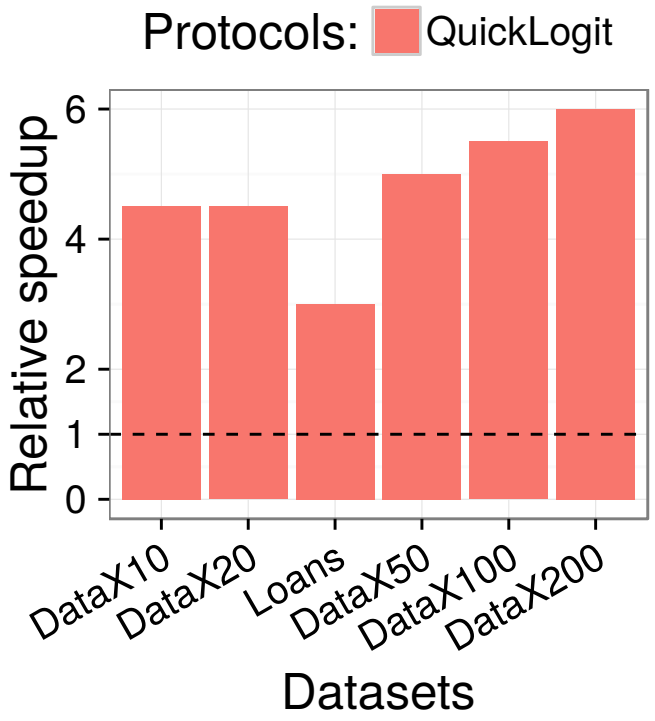


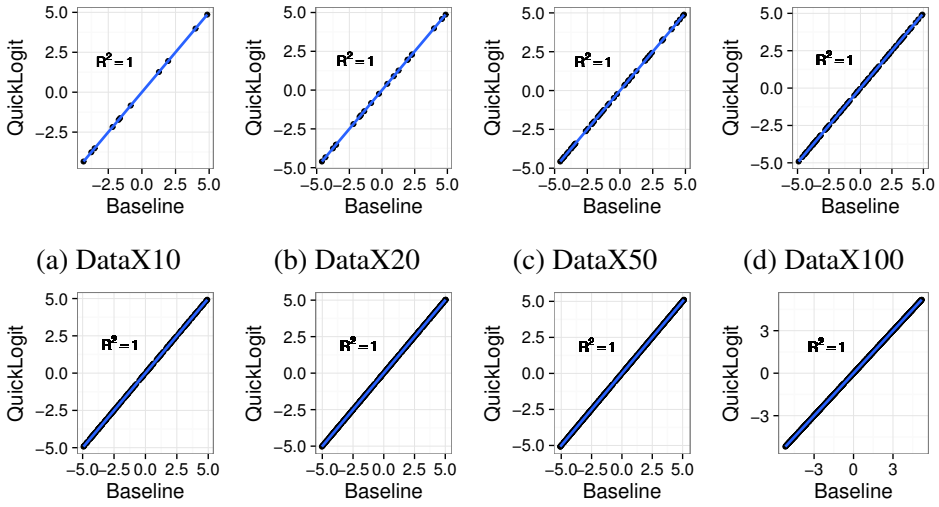
Figure V.12: Relative speedup of QuickLogit over Newton baseline ($y = 1$ dashed line) across all datasets, based on runtime. Larger speedup indicates faster computation.

V.3.5.3.1 Simple model averaging has good approximation power.

Our experimental results indicate that simple averaging of local-institution models tend to be rather accurate (though not perfect). This is evidenced from two aspects. First, as discussed earlier in Table V.6, QuickLogit has dramatic reduction in the iterations required for convergence. This means that our initialization based on simple averaging is indeed a very good approximation for (thus very close to) the globally optimal model. Additional results are also illustrated in Figure V.14, where simple averaging-derived models are quite close to, though not exact with, the Newton baseline.

V.3.5.3.2 Simple averaging alone is not perfect.

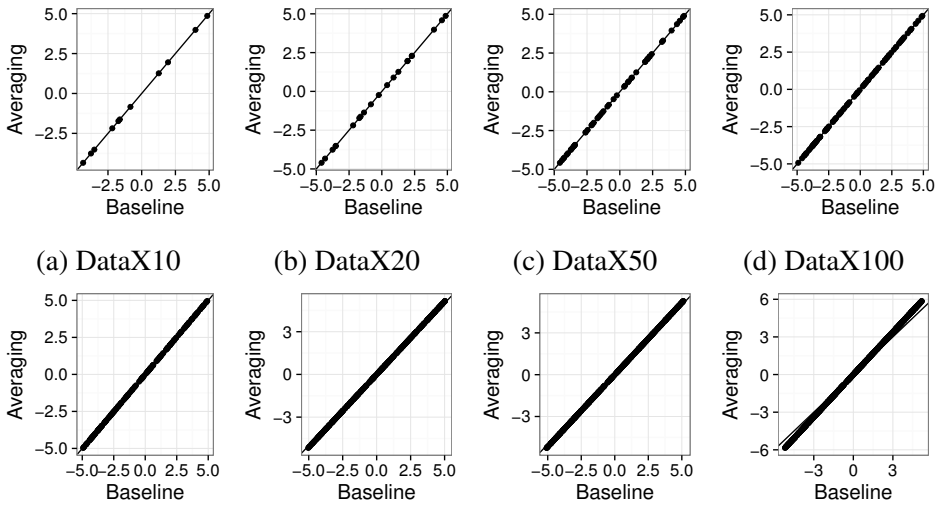
One the other hand, we point out that many times one-step of averaging alone is not perfect, and may not be sufficient in terms of accuracy. For instance, when the data dimensionality



(a) DataX10 (b) DataX20 (c) DataX50 (d) DataX100

(e) DataX200 (f) DataX400 (g) DataX500 (h) DataX1000

Figure V.13: Accuracy of model coefficient estimates from our QuickLogit, with Newton as baseline (x-axis). All correlation $R^2 = 1.00$ and slope of fitted line is 1.00



(a) DataX10 (b) DataX20 (c) DataX50 (d) DataX100

(e) DataX200 (f) DataX400 (g) DataX500 (h) DataX1000

Figure V.14: Accuracy of model coefficient estimates from one-step simple averaging, with Newton as baseline (x-axis).

grows larger or the data is imbalanced between different institutions, the discrepancy between the simple averaging-based model and the optimal baseline may become apparent. This is evident in cases such as *DataX1000* (Figure V.14). Moreover, Table V.6 indicates that some datasets (e.g., *DataX500, DataX1000*) require increased iterations for Quick-Logit. This is a sign that our initialization is not perfect and thus it needs more iterations to refine the model estimation from that initialization.

V.3.6 Related Works

Here we discuss a few streams of research that are closed related to ours.

Privacy protection on logistic regression has received extensive investigation, from the centralized solutions [41] to more recent solutions leveraging distributed machine learning [154, 118, 156, 23, 159, 93]. Many existing proposals [154] directly expose organization-level summary data of model fitting, leading to serious privacy concerns over inference attacks on intermediate data [139, 124, 159, 41]. Nearly all of the aforementioned proposals directly follow distributed machine learning formulations, thus missing the unique performance improvement introduced in our work.

Cryptography-based protocols have also found application in other machine learning tasks such as linear (ridge) regression and association rule mining [96, 84, 121, 4]. None of these solutions provide formulations that are different than off-the-shelf distributed machine learning.

To the best of our knowledge, our proposal is the first to introduce the concept of leveraging local *complete* models to better initialize numerical optimizers and provide significant performance improvement.

V.3.7 Discussion

Our current implementation does not consider potential privacy concerns on the final outcome of the collaborative study. This is in line with the assumption and guarantees of cryptography and SMC (that nothing but the final output is revealed). Another considera-

tion is due to computational efficiency. If we consider the coefficient output to be private, then local computations (across iterations) will have to involve cryptographic protection too because computing summary statistics relies on the coefficients. This will make the secure computation significantly more complicated and expensive, given that many functions involved (e.g., the logistic regression objective) is highly non-linear and not directly computable in secure. Thus it often has to resort to approximations, which is still an open problem because it is difficult to maintain trade-off between computational efficiency and proper approximation accuracy [118]. For such reasons, we do not consider alternatives based on approximation. But we point out that our approach is also directly applicable to the approximation case.

While this work uses logistic regression as our focus application, the proposed new paradigm is widely applicable to nearly all statistical and machine learning models in the secure multi-party setting. In future, we hope to extend to and significantly improve other related tasks by providing a generic secure learning framework.

V.3.7.1 Conclusion

This dissertation questions the common practice of building secure protocols directly from distributed machine learning algorithms, and propose a drastically different paradigm to privacy-preserving distributed logistic regression leveraging local models. Extensive theoretical and empirical evidence demonstrate significant performance advantage of our proposal.

BIBLIOGRAPHY

[1] Home depot's 56 million card breach bigger than target's. http://www.wsj.com/articles/home-depot-breach-bigger-than-targets-1411073571, 2015. Accessed: 2015-02-25.

[2] Massive breach at health care company anthem inc. http://www.usatoday.com/story/tech/2015/02/04/health-care-anthem-hacked/22900925/, 2015. Accessed: 2015-02-25.

[3] Emmanuel A Abbe, Amir E Khandani, and Andrew W Lo. Privacy-preserving methods for sharing financial risk exposures. *American Economic Review*, 102(3):65–70, 2012.

[4] Charu C Aggarwal and S Yu Philip. *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[5] Zack W Almquist and Carter T Butts. Logistic network regression for scalable analysis of networks with joint edge/vertex dynamics. *Sociological Methodology*, 44(1):273–321, 2014.

[6] Russ B Altman, Ellen Wright Clayton, Isaac S Kohane, Bradley A Malin, and Dan M Roden. Data re-identification: societal safeguards. *Science*, 339(6123):1032, 2013.

[7] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang. Scalable and secure logistic regression via homomorphic encryption. In *Sixth ACM Conference on Data and Application Security and Privacy*, pages 142–144, 2016.

[8] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 535–548. ACM, 2013.

[9] Erman Ayday, Jean Louis Raisaro, Urs Hengartner, Adam Molyneaux, and Jean-Pierre Hubaux. Privacy-preserving processing of raw genomic data. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 133–147. Springer, 2014.

[10] Kristin L Ayers and Heather J Cordell. Snp selection in genome-wide and candidate gene studies via penalized logistic regression. *Genetic epidemiology*, 34(8):879–891, 2010.

[11] Chloé-Agathe Azencott. Machine learning and genomics: precision medicine vs. patient privacy. *arXiv preprint arXiv:1802.10568*, 2018.

[12] Al Aziz, Md Momin, Mohammad Z Hasan, Noman Mohammed, and Dima Alhadidi. Secure and efficient multiparty computation on genomic data. In *Proceedings*

of the 20th International Database Engineering & Applications Symposium, pages 278–283. ACM, 2016.

[13] Amos Beimel. Secret-sharing schemes: a survey. In *Coding and cryptology*, pages 11–46. Springer, 2011.

[14] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, 2012.

[15] Dan Bogdanov, Sven Laur, and Jan Willemson. Sharemind: A framework for fast privacy-preserving computations. In *Computer Security-ESORICS 2008*, pages 192–206. Springer, 2008.

[16] Dankmar Böhning and Bruce G Lindsay. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663, 1988.

[17] Steven M Boker, Timothy R Brick, Joshua N Pritikin, Yang Wang, Timo von Oertzen, Donald Brown, John Lach, Ryne Estabrook, Michael D Hunter, Hermine H Maes, et al. Maintained individual data distributed likelihood estimation (middle). *Multivariate behavioral research*, 50(6):706–720, 2015.

[18] Danah Boyd. Facebook's privacy trainwreck. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):13–20, 2008.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[20] William S Bush and Jason H Moore. Genome-wide association studies. *PLoS computational biology*, 8(12):e1002822, 2012.

[21] Nilanjan Chatterjee, Bill Wheeler, Joshua Sampson, Patricia Hartge, Stephen J Chanock, and Ju-Hyun Park. Projecting the performance of risk prediction based on polygenic analyses of genome-wide association studies. *Nature genetics*, 45(4):400–405, 2013.

[22] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, pages 289–296, 2009.

[23] Colin Chen. Distributed iteratively reweighted least squares and applications. *STATISTICS AND ITS INTERFACE*, 6(4):585–593, 2013.

[24] Feng Chen, Shuang Wang, Xiaoqian Jiang, Sijie Ding, Yao Lu, Jihoon Kim, S Cenk Sahinalp, Chisato Shimizu, Jane C Burns, Victoria J Wright, et al. Princess: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions. *Bioinformatics*, 33(6):871–878, 2016.

[25] You Chen, Wei Xie, Carl A Gunter, David Liebovitz, Sanjay Mehrotra, He Zhang, and Bradley Malin. Inferring clinical workflow efficiency via electronic medical record utilization. In *AMIA annual symposium proceedings*, volume 2015, page 416. American Medical Informatics Association, 2015.

[26] Long Cheng, Ying Wang, Yulong Pei, and Dick Epema. A coflow-based co-optimization framework for high-performance data analytics. In *Parallel Processing (ICPP), 2017 46th International Conference on*, pages 392–401. IEEE, 2017.

[27] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *bioRxiv*, page 142760, 2018.

[28] Andrew R Conn, Nicholas IM Gould, and Ph L Toint. Convergence of quasi-newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50(1-3):177–195, 1991.

[29] 1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

[30] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[31] Chaoyue Dai, Feng Qian, Wei Jiang, Zhoutian Wang, and Zenghong Wu. A personalized recommendation system for netease dating site. *Proceedings of the VLDB Endowment*, 7(13), 2014.

[32] Jon P Daries, Justin Reich, Jim Waldo, Elise M Young, Jonathan Whittinghill, Andrew Dean Ho, Daniel Thomas Seaton, and Isaac Chuang. Privacy, anonymity, and big data in the social sciences. *Communications of the ACM*, 57(9):56–63, 2014.

[33] Abhijit Dasgupta, Yan V Sun, Inke R König, Joan E Bailey-Wilson, and James D Malley. Brief review of regression-based and machine learning methods in genetic epidemiology: the genetic analysis workshop 17 experience. *Genetic epidemiology*, 35(S1):S5–S11, 2011.

[34] Martine De Cock, Rafael Dowsley, Caleb Horst, Raj Katti, Anderson Nascimento, Wing-Sea Poon, and Stacey Truex. Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation. *IEEE Transactions on Dependable and Secure Computing*, 2017.

[35] Joshua C Denny, Lisa Bastarache, Marylyn D Ritchie, Robert J Carroll, Raquel Zink, Jonathan D Mosley, Julie R Field, Jill M Pulley, Andrea H Ramirez, Erica Bowton, et al. Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. *Nature biotechnology*, 31(12):1102, 2013.

[36] Joshua C Denny, Dana C Crawford, Marylyn D Ritchie, Suzette J Bielinski, Melissa A Basford, Yuki Bradford, High Seng Chai, Lisa Bastarache, Rebecca Zuvich, Peggy Peissig, et al. Variants near foxe1 are associated with hypothyroidism and other thyroid conditions: using electronic medical records for genome-and

phenome-wide studies. *The American Journal of Human Genetics*, 89(4):529–542, 2011.

[37] Cynthia Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[38] Cynthia Dwork. Differential privacy. In *Encyclopedia of Cryptography and Security*, pages 338–340. Springer, 2011.

[39] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[40] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. 2017.

[41] Khaled El Emam, Saeed Samet, Luk Arbuckle, Robyn Tamblyn, Craig Earle, and Murat Kantarcioglu. A secure distributed logistic regression protocol for the detection of rare adverse drug events. *Journal of the American Medical Informatics Association*, 20(3):453–461, 2013.

[42] Martin Enserink and Gilbert Chin. The end of privacy. *Science*, 347(6221):490–491, 2015.

[43] Yaniv Erlich and Arvind Narayanan. Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15(6):409–421, 2014.

[44] European Commission. Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation). http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf, 2012. (29 June 2014, date last accessed).

[45] European Commission. Opinion 05/2014 on anonymisation techniques, adopted 10 april, wp216. http://www.cnpd.public.lu/fr/publications/groupe-art29/wp216_en.pdf, 2014. (29 June 2014, date last accessed).

[46] Evangelos Evangelou and John PA Ioannidis. Meta-analysis methods for genome-wide association studies and beyond. *Nature Reviews Genetics*, 14(6):379–389, 2013.

[47] 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. http://www.ftc.gov/sites/default/files/documents/reports/federal-trade-commission-bureau-consumer-protection-preliminary-ftc-staff-report-protecting-cor 101201privacyreport.pdf, 2010. Accessed: 2015-01-10.

[48] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Progress in Artificial Intelligence*, pages 535–546. Springer, 2015.

[49] Megan D Fesinmeyer, Kari E North, Marylyn D Ritchie, Unhee Lim, Nora Franceschini, Lynne R Wilkens, Myron D Gross, Petra Bůžková, Kimberly Glenn, P Miguel Quibrera, et al. Genetic risk factors for bmi and obesity in an ethnically diverse population: results from the population architecture using genomics and epidemiology (page) study. *Obesity*, 21(4):835–846, 2013.

[50] Tobias Freilinger, Verneri Anttila, Boukje de Vries, Rainer Malik, Mikko Kallela, Gisela M Terwindt, Patricia Pozo-Rosich, Bendik Winsvold, Dale R Nyholt, Willebrordus PJ van Oosterhout, et al. Genome-wide association analysis identifies susceptibility loci for migraine without aura. *Nature genetics*, 44(7):777–782, 2012.

[51] Christian Fuchsberger, Daniel Taliun, Peter P Pramstaller, and Cristian Pattaro. Gwatoolbox: an r package for fast quality control and handling of genome-wide association studies meta-analysis data. *Bioinformatics*, 28(3):444–445, 2012.

[52] Stephanie M Fullerton, Nicholas R Anderson, Greg Guzauskas, Dena Freeman, and Kelly Fryer-Edwards. Meeting the governance challenges of next-generation biorepository research. *Science translational medicine*, 2(15):15cm3–15cm3, 2010.

[53] GIANT. Giant consortium. https://www.broadinstitute.org/collaboration/giant/index.php/GIANT_consortium_data_files, 2016. Accessed: 2015-02-25.

[54] Richard A Gibbs, John W Belmont, Paul Hardenbol, Thomas D Willis, Fuli Yu, Huanming Yang, Lan-Yang Ch'ang, Wei Huang, Bin Liu, Yan Shen, et al. The international hapmap project. *Nature*, 426(6968):789–796, 2003.

[55] Oded Goldreich. Foundation of cryptography (in two volumes: Basic tools and basic applications), 2001.

[56] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[57] Eric D Green, Mark S Guyer, National Human Genome Research Institute, et al. Charting a course for genomic medicine from base pairs to bedside. *Nature*, 470(7333):204–213, 2011.

[58] Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 149–192, 1984.

[59] Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.

[60] Christopher A Haiman, Megan D Fesinmeyer, Kylee L Spencer, Petra Bůžková, V Saroja Voruganti, Peggy Wan, Jeff Haessler, Nora Franceschini, Kristine R Monroe, Barbara V Howard, et al. Consistent directions of effect for established type 2 diabetes risk variants across populations the population architecture using genomics and epidemiology (page) consortium. *Diabetes*, 61(6):1642–1647, 2012.

[61] Rob Hall, Stephen E Fienberg, and Yuval Nardi. Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27(4):669, 2011.

[62] Jihun Hamm. Minimax filter: Learning to preserve privacy from inference attacks. *Journal of Machine Learning Research*, 18(129):1–31, 2017.

[63] Jihun Hamm, Yingjun Cao, and Mikhail Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, pages 555–563, 2016.

[64] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.

[65] Frank Harrell. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer, 2015.

[66] Larry V Hedges and Ingram Olkin. *Statistical methods for meta-analysis*. Academic press, 2014.

[67] Wilko Henecka and Thomas Schneider. Faster secure two-party computation with less memory. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 437–446. ACM, 2013.

[68] Julian Higgins and Simon G Thompson. Quantifying heterogeneity in a meta-analysis. *Statistics in medicine*, 21(11):1539–1558, 2002.

[69] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[70] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, volume 201, 2011.

[71] Zhicong Huang, Huang Lin, Jacques Fellay, Zoltán Kutalik, and Jean-Pierre Hubaux. Sqc: secure quality control for meta-analysis of genome-wide association studies. *Bioinformatics*, page btx193, 2017.

[72] Kathy L Hudson, MK Holohan, and Francis S Collins. Keeping pace with the times—the genetic information nondiscrimination act of 2008. *New England Journal of Medicine*, 358(25):2661–2663, 2008.

[73] Thomas J Hudson, Warwick Anderson, Axel Aretz, Anna D Barker, Cindy Bell, Rosa R Bernabé, MK Bhan, Fabien Calvo, Iiro Eerola, Daniela S Gerhard, et al. International network of cancer genome projects. *Nature*, 464(7291):993–998, 2010.

[74] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. Addressing the concerns of the lacks family: Quantification of kin genomic privacy. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1141–1152. ACM, 2013.

[75] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. Quantifying interdependent risks in genomic privacy. *ACM Transactions on Privacy and Security (TOPS)*, 20(1):3, 2017.

[76] iDash Privacy & Security Workshop. Secure genome analysis competition. http://www.humangenomeprivacy.org/2015, 2015. Accessed: 2015-05-03.

[77] Hae Kyung Im, Eric R Gamazon, Dan L Nicolae, and Nancy J Cox. On sharing quantitative trait gwas results in an era of multiple-omics data and the limits of genomic privacy. *The American Journal of Human Genetics*, 90(4):591–598, 2012.

[78] Kevin B Jacobs, Meredith Yeager, Sholom Wacholder, David Craig, Peter Kraft, David J Hunter, Justin Paschal, Teri A Manolio, Margaret Tucker, Robert N Hoover, et al. A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies. *Nature genetics*, 41(11):1253–1257, 2009.

[79] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[80] Daniel E Jonas, Halle R Amick, Cynthia Feltner, Georgiy Bobashev, Kathleen Thomas, Roberta Wines, Mimi M Kim, Ellen Shanahan, C Elizabeth Gass, Cassandra J Rowe, et al. Pharmacotherapy for adults with alcohol use disorders in outpatient settings: A systematic review and meta-analysis. *JAMA*, 311(18):1889–1900, 2014.

[81] Gulce Kale, Erman Ayday, and Oznur Tastan. A utility maximizing and privacy preserving approach for protecting kinship in genomic databases. *Bioinformatics*, 34(2):181–189, 2017.

[82] Liina Kamm, Dan Bogdanov, Sven Laur, and Jaak Vilo. A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, 29(7):886–893, 2013.

[83] Murat Kantarcioglu, Wei Jiang, Ying Liu, and Bradley Malin. A cryptographic approach to securely share and query genomic sequences. *Information Technology in Biomedicine, IEEE Transactions on*, 12(5):606–617, 2008.

[84] Alan F Karr, William J Fulp, Francisco Vera, S Stanley Young, Xiaodong Lin, and Jerome P Reiter. Secure, privacy-preserving analysis of distributed databases. *Technometrics*, 49(3):335–345, 2007.

[85] Aniket Kate and Ian Goldberg. Distributed key generation for the internet. In *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on*, pages 119–128. IEEE, 2009.

[86] Jane Kaye, Catherine Heeney, Naomi Hawkins, Jantina de Vries, and Paula Boddington. Data sharing in genomics—re-shaping scientific practice. *Nature Reviews Genetics*, 10(5):331–335, 2009.

[87] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[88] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[89] LAPACK authors. LAPACK – linear algebra package. http://www.netlib.org/lapack, 2010. Accessed: 2010-09-30.

[90] Saskia Le Cessie and Johannes C Van Houwelingen. Ridge estimators in logistic regression. *Applied statistics*, pages 191–201, 1992.

[91] LengingClub. Loans data. http://www.lendingclub.com, 2016. Last accessed: 02-02-2016.

[92] Cathryn M Lewis and Jo Knight. Introduction to genetic association studies. *Cold Spring Harbor Protocols*, 2012(3):pdb–top068163, 2012.

[93] Wenfa Li, Hongzhe Liu, Peng Yang, and Wei Xie. Supporting regularized logistic regression privately and efficiently. *PloS ONE*, 11(6), 2016.

[94] Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015.

[95] Zhen Lin, Art B Owen, and Russ B Altman. Genomic research and human subject privacy. *Science*, 305(5681):183, 2004.

[96] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. In *Advances in Cryptology—CRYPTO 2000*, pages 36–54. Springer, 2000.

[97] Max A Little, Patrick E McSharry, Eric J Hunter, Jennifer Spielman, and Lorraine O Ramig. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *Biomedical Engineering, IEEE Transactions on*, 56(4):1015–1022, 2009.

[98] Chang Liu, Xiao Shaun Wang, Kartik Nayak, Yan Huang, and Elaine Shi. Oblivm: A programming framework for secure computation. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 359–376. IEEE, 2015.

[99] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[100] Mei Liu, Yonghui Wu, Yukun Chen, Jingchun Sun, Zhongming Zhao, Xue-wen Chen, Michael Edwin Matheny, and Hua Xu. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *Journal of the American Medical Informatics Association*, 19(e1):e28–e35, 2012.

[101] Zhe Liu, Yuanyuan Shen, and Jurg Ott. Multilocus association mapping using generalized ridge logistic regression. *BMC bioinformatics*, 12(1):384, 2011.

[102] William W Lowrance and Francis S Collins. Identifiability in genomic research. *Science*, 317:600–602, 2007.

[103] Edmund G Lowrie and Nancy L Lew. Death risk in hemodialysis patients: the predictive value of commonly measured variables and an evaluation of death rate differences between facilities. *American Journal of Kidney Diseases*, 15(5):458–482, 1990.

[104] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[105] Jeantine E Lunshof, Ruth Chadwick, Daniel B Vorhaus, and George M Church. From genetic privacy to open consent. *Nature Reviews Genetics*, 9(5):406–411, 2008.

[106] Matthew D Mailman, Michael Feolo, Yumi Jin, Masato Kimura, Kimberly Tryka, Rinat Bagoutdinov, Luning Hao, Anne Kiang, Justin Paschall, Lon Phan, et al. The ncbi dbgap database of genotypes and phenotypes. *Nature genetics*, 39(10):1181–1186, 2007.

[107] Bradley A Malin, Khaled El Emam, and Christine M O'Keefe. Biomedical data privacy: problems, perspectives, and recent advances. *Journal of the American medical informatics association*, 20(1):2–6, 2013.

[108] Nathalie Malo, Ondrej Libiger, and Nicholas J Schork. Accommodating linkage disequilibrium in genetic-association analyses via ridge regression. *The American Journal of Human Genetics*, 82(2):375–385, 2008.

[109] Naoki Masuda, Issei Kurahashi, and Hiroko Onari. Suicide ideation of individuals in online social networks. *PloS one*, 8(4):e62262, 2013.

[110] Tara C Matise, Jose Luis Ambite, Steven Buyske, Christopher S Carlson, Shelley A Cole, Dana C Crawford, Christopher A Haiman, Gerardo Heiss, Charles Kooperberg, Loic Le Marchand, et al. The next page in understanding complex traits:

design for the analysis of population architecture using genetics and epidemiology (page) study. *American journal of epidemiology*, 174(7):849–859, 2011.

[111] Catherine A McCarty, Rex L Chisholm, Christopher G Chute, Iftikhar J Kullo, Gail P Jarvik, Eric B Larson, Rongling Li, Daniel R Masys, Marylyn D Ritchie, Dan M Roden, et al. The emerge network: a consortium of biorepositories linked to electronic medical records data for conducting genomic studies. *BMC medical genomics*, 4(1):13, 2011.

[112] P McDaniel and S McLaughlin. Security and privacy challenges in the smart grid. *Security & Privacy, IEEE*, 7(3):75–77, 2009.

[113] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

[114] Thomas P Minka. A comparison of numerical optimizers for logistic regression. http://research.microsoft.com/en-us/um/people/minka/papers/logreg/, 2003.

[115] Ramal Moonesinghe, Muin J Khoury, Tiebin Liu, and John PA Ioannidis. Required sample size and nonreplicability thresholds for heterogeneous genetic associations. *Proceedings of the National Academy of Sciences*, 105(2):617–622, 2008.

[116] Adam C Naj, Gyungah Jun, Gary W Beecham, Li-San Wang, Badri Narayan Vardarajan, Jacqueline Buros, Paul J Gallins, Joseph D Buxbaum, Gail P Jarvik, Paul K Crane, et al. Common variants at ms4a4/ms4a6e, cd2ap, cd33 and epha1 are associated with late-onset alzheimer's disease. *Nature genetics*, 43(5):436–441, 2011.

[117] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254. ACM, 1999.

[118] yuval nardi, stephen e fienberg, and robert j hall. achieving both valid and secure logistic regression analysis on aggregated data from different private sources. *journal of privacy and confidentiality*, 4(1):9, 2012.

[119] K Nigam. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information filtering*, 1999.

[120] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 334–348. IEEE, 2013.

[121] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 334–348. IEEE, 2013.

[122] Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Entity resolution and federated learning get a federated resolution. *arXiv preprint arXiv:1803.04035*, 2018.

[123] HHS Office for Civil Rights. Standards for privacy of individually identifiable health information. final rule. *Federal Register*, 67(157):53181, 2002.

[124] Christine M O'Keefe and James O Chipperfield. A summary of attack methods and confidentiality protection measures for fully automated remote analysis systems. *International Statistical Review*, 81(3):426–455, 2013.

[125] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*, pages 223–238. Springer, 1999.

[126] Orestis A Panagiotou, Cristen J Willer, Joel N Hirschhorn, and John PA Ioannidis. The power of meta-analysis in genome wide association studies. *Annual review of genomics and human genetics*, 14:441, 2013.

[127] Ju-Hyun Park, Sholom Wacholder, Mitchell H Gail, Ulrike Peters, Kevin B Jacobs, Stephen J Chanock, and Nilanjan Chatterjee. Estimation of effect size distribution from genome-wide association studies and implications for future discoveries. *Nature genetics*, 42(7):570–575, 2010.

[128] Mee Young Park and Trevor Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1):30–50, 2008.

[129] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, 96(1):3–14, 2002.

[130] Presidential Commission for the Study of Bioethical Issues. Privacy and progress in whole genome sequencing. Washington, DC, 2012.

[131] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.

[132] Laura L Rodriguez, Lisa D Brooks, Judith H Greenberg, and Eric D Green. The complexities of genomic identifiability. *Science*, 339(6117):275–276, 2013.

[133] Jonathan Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *arXiv preprint arXiv:1407.2724*, 2014.

[134] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967, 2009.

[135] Joe V Selby, Anne C Beal, and Lori Frank. The patient-centered outcomes research institute (pcori) national priorities for research and initial research agenda. *Jama*, 307(15):1583–1584, 2012.

[136] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[137] Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication efficient distributed optimization using an approximate newton-type method. *arXiv preprint arXiv:1312.7853*, 2013.

[138] Xia Shen, Moudud Alam, Freddy Fikse, and Lars Rönnegård. A novel generalized ridge regression method for quantitative genetics. *Genetics*, 193(4):1255–1268, 2013.

[139] Ross Sparks, Chris Carter, John B Donnelly, Christine M O'Keefe, Jodie Duncan, Tim Keighley, and Damien McAullay. Remote access methods for exploratory data analysis and statistical modelling: Privacy-preserving analytics®. *Computer methods and programs in biomedicine*, 91(3):208–222, 2008.

[140] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[141] Patrick Taylor. Personal genomes: when consent gets in the way. *Nature*, 456(7218):32–33, 2008.

[142] Stephen Turner, Loren L Armstrong, Yuki Bradford, Christopher S Carlson, Dana C Crawford, Andrew T Crenshaw, Mariza Andrade, Kimberly F Doheny, Jonathan L Haines, Geoffrey Hayes, et al. Quality control procedures for genome-wide association studies. *Current protocols in human genetics*, pages 1–19, 2011.

[143] US Department of Health and Human Services and the Food and Drug Administration. Advance notice of proposed rulemaking: Human subjects research protections: Enhancing protections for research subjects and reducing burden, delay, and ambiguity for investigators. *Federal Register*, 76:44512–44531, 2011.

[144] Peter J van der Most, Ahmad Vaez, Bram P Prins, M Loretto Munoz, Harold Snieder, Behrooz Z Alizadeh, and Ilja M Nolte. Qcgwas: A flexible r package for automated quality control of genome-wide association results. *Bioinformatics*, 30(8):1185–1186, 2014.

[145] Peter Van Der Putten and Maarten van Someren. Coil challenge 2000: The insurance company case. *Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report*, 9:1–43, 2000.

[146] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[147] Peter M Visscher, Matthew A Brown, Mark I McCarthy, and Jian Yang. Five years of gwas discovery. *The American Journal of Human Genetics*, 90(1):7–24, 2012.

[148] Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. Learning your identity and disease from research papers: information leaks in genome wide association study. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 534–544. ACM, 2009.

[149] Shuang Wang, Yuchen Zhang, Wenrui Dai, Kristin Lauter, Miran Kim, Yuzhe Tang, Hongkai Xiong, and Xiaoqian Jiang. Healer: Homomorphic computation of exact logistic regression for secure rare disease variants analysis in gwas. *Bioinformatics*, 32(2):211–218, 2015.

[150] Larry Wasserman. *All of statistics*. Springer Science & Business Media, 2011.

[151] Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, et al. The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic acids research*, 42(D1):D1001–D1006, 2014.

[152] Cristen J Willer, Yun Li, and Gonçalo R Abecasis. Metal: fast and efficient meta-analysis of genomewide association scans. *Bioinformatics*, 26(17):2190–2191, 2010.

[153] Thomas W Winkler, Felix R Day, Damien C Croteau-Chonka, Andrew R Wood, Adam E Locke, Reedik Mägi, Teresa Ferreira, Tove Fall, Mariaelisa Graff, Anne E Justice, et al. Quality control and conduct of genome-wide association meta-analyses. *Nature protocols*, 9(5):1192–1212, 2014.

[154] Michael Wolfson, Susan E Wallace, Nicholas Masca, Geoff Rowe, Nuala A Sheehan, Vincent Ferretti, Philippe LaFlamme, Martin D Tobin, John Macleod, Julian Little, 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*, page dyq111, 2010.

[155] Yuan Wu, Xiaoqian Jiang, Jihoon Kim, and Lucila Ohno-Machado. Grid binary logistic regression (glore): building shared models without sharing data. *Journal of the American Medical Informatics Association*, 19(5):758–764, 2012.

[156] Yuan Wu, Xiaoqian Jiang, and Lucila Ohno-Machado. Preserving institutional privacy in distributed binary logistic regression. In *AMIA Annual Symposium Proceedings*, volume 2012, page 1450. American Medical Informatics Association, 2012.

[157] Wei Xie. Circuitservice: run garbled circuit as a backend service, 2014.

[158] Wei Xie. Protecting participant privacy in genotype-phenotype association meta-analysis. Master's thesis, Vanderbilt University, 2014.

[159] Wei Xie, Murat Kantarcioglu, William S Bush, Dana Crawford, Joshua C Denny, Raymond Heatherly, and Bradley A Malin. Securema: protecting participant privacy in genetic association meta-analysis. *Bioinformatics*, 30(23):3334–3341, 2014.

[160] Wei Xie, Murat Kantarcioglu, Joshua C Denny, Todd L Edwards, Nancy J Cox, and Bradley A Malin. Privacy leaks in quality control on gwas meta-analysis and effective countermeasures. *American Society of Human Genetics*, 2015.

[161] Wei Xie, Yang Wang, Steven M Boker, and Donald E Brown. Privlogit: Efficient privacy-preserving logistic regression by tailoring numerical optimizers. *arXiv preprint arXiv:1611.01170*, 2016.

[162] Wei Xie, Yang Wang, Steven M. Boker, and Donald E. Brown. Quicklogit: A new paradigm for efficient privacy-preserving logistic regression. *Manuscript*, 2016.

[163] Jian Yang, S Hong Lee, Michael E Goddard, and Peter M Visscher. Gcta: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.

[164] Andrew C Yao. Protocols for secure computations. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, 1982.

[165] Zhijun Yin, Bradley Malin, Jeremy Warner, Pei-Yun Sabrina Hsueh, and Ching-Hua Chen. The power of the patient voice: Learning indicators of treatment adherence from an online breast cancer forum. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017.*, pages 337–346, 2017.

[166] Zhijun Yin, Wei Xie, and Bradley Malin. Talking about my care: Detecting mentions of hormonal therapy adherence behavior in an online breast cancer community. In *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, 2017.

[167] Elias A Zerhouni and Elizabeth G Nabel. Protecting aggregate genomic data. *Science*, 322(5898):44a, 2008.

[168] Junjun Zhang, Joachim Baran, A Cros, Jonathan M Guberman, Syed Haider, Jack Hsu, Yong Liang, Elena Rivkin, Jianxin Wang, Brett Whitty, et al. International cancer genome consortium data portal—a one-stop shop for cancer genomics data. *Database*, 2011:bar026, 2011.

[169] Yuchen Zhang, Martin J Wainwright, and John C Duchi. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pages 1502–1510, 2012.

[170] Tao Zheng, Wei Xie, Liling Xu, Xiaoying He, Ya Zhang, Mingrong You, Gong Yang, and You Chen. A machine learning-based framework to identify type 2 diabetes through electronic health records. *International journal of medical informatics*, 97:120–127, 2017.