PROTECTING PARTICIPANT PRIVACY IN GENOTYPE-PHENOTYPE ASSOCIATION META-ANALYSIS

By

Wei Xie

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Computer Science

December, 2014

Nashville, TN

Approved:

Professor Bradley A. Malin

Professor Yuan Xue

ACKNOWLEDGMENTS

I am very fortunate to have received help and inspiration from numerous people during my Master's study, for which I feel always grateful.

First of all, I would like to thank my advisor, Dr. Bradley A. Malin, who is extremely intelligent and supportive. This thesis would not have been possible without his guiding me in the right direction and contributing insightful discussions. His emphasis on perfectionism has constantly stimulated me to improve both in terms of sciences and my writing. Dr. Malin has also been very patient and supportive even when the project experienced technical drawbacks.

I would also like to thank our collaborators, in particular, Dr. Murat Kantarcioglu of the University of Texas, Dallas, who is very knowledgeable in the field and has been essential to the technical merits of the project by proposing insightful suggestions and creative solutions. Dr. William S. Bush (of the Department of Biomedical Informatics then) also contributed a lot to the project and educated me with his ample domain knowledge and computational skills. Dr. Raymond Heatherly also provided helpful feedback and assistance with the experiments and my writing. Also, I would like to thank Drs. Joshua C. Denny and Dana Crawford for their continuous guidance on the problem motivation and data processing.

I also owe gratitude to my committee member, Dr. Yuan Xue, who is very supportive and has been involved since the early discussions of the project and offered helpful feedback. I feel extremely honored to have her on my committee, and am always grateful for her time and efforts in helping improve my manuscript and the project in general.

Lastly, I thank all members of the Health Information Privacy lab for their support. I also thank the Department of Electrical Engineering and Computer Science, as well as the National Institutes of Health (grant numbers R01LM009989, U01HG006378) and the National Science Foundation (grant number CCF0424422) for the generous financial support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

Decreasing costs in sequencing technologies, in combination with large repositories of clinical information, has enabled many novel discoveries in biomedicine by means of examining the associations between genetic variants and disease. These achievements are facilitated by increased collection and reuse of genomic data (Green et al., 2011), as well as broad efforts to obtain larger sample sizes (by sharing and combing data) for increased statistical power (Panagiotou et al., 2013). Meta-analysis is a common solution for aggregating study results across large consortia to achieve this goal. In fact, meta-analysis is responsible for approximately 37% of the 15,845 genotype-phenotype associations listed in the National Human Genome Research Institute (NHGRI) Catalog of Published Genome-Wide Association Studies (GWAS) Catalog (Welter et al., 2014).

At the same time, the sensitive nature of genomic data has led to numerous discussions around the governance of genomic records (Fullerton et al., 2010; Kaye et al., 2009). Currently, policy and advisory groups recommend removing identifying information (such as personal names) to uphold the privacy of study participants (Lowrance and Collins, 2007; Presidential Commission for the Study of Bioethical Issues, 2012).

I.1 Privacy Concerns on Genomic Data

Yet, the efficacy of such existing protections is increasingly being questioned (Rodriguez et al., 2013), due to various demonstrations the identity of study participants, as well as other sensitive information (such as disease status) can still be inferred from the shared (presumably "safe") genomic data (Gymrek et al., 2013; Lin et al., 2004; Homer et al., 2008; Jacobs et al., 2009; Sankararaman et al., 2009; Im et al., 2012; Humbert et al., 2013; Erlich and Narayanan, 2014).

Most recently, it was shown that a person's identity could be ascertained by profiling a

person's Y-chromosome short tandem repeats (Y-STRs) and searching against various public genealogy databases on the internet – even if the individual's identity was not initially tied to a DNA sequence (Gymrek et al., 2013). Yet, this is only the latest indication of the challenges related with protecting identities and sensitive information from being inferred. In 2004, for instance, it was illustrated that only a handful of single nucleotide polymorphisms (SNPs) were necessary to uniquely distinguish an individual's sequence (Lin et al., 2004). Next, in 2008 and later, it was shown that the rates of SNP alleles in a pool (e.g., a group of diabetics) could reveal disease status (Homer et al., 2008; Jacobs et al., 2009; Sankararaman et al., 2009). Later in 2012, it was indicated that sufficient summary statistics (e.g., regression coefficients and allele dosage) of genetic associations could lead to identifiability concerns (Im et al., 2012). And in 2013, it was also suggested the disclosure of one individual's genome sequence could even jeopardize his relatives' privacy due to the high correlation between familial genomes (Humbert et al., 2013). More extensive reviews on the privacy issues related to managing genomic data can be found at (Erlich and Narayanan, 2014; Naveed et al., 2014).

While certain privacy attacks may seem nontrivial in the knowledge necessary to be executed (Erlich and Narayanan, 2014), they have already raised serious concerns from scientists, policy makers, and the general public. They have also led to reduced sharing of genome sequences and even site-level summary statistics of studies. For instance, based on (Homer et al., 2008), the National Institutes of Health (NIH) and Wellcome Trust stopped sharing aggregate genomic data directly to the public (Zerhouni and Nabel, 2008). These demonstrations have also influenced proposed regulations such as (US Department of Health and Human Services and the Food and Drug Administration, 2011; European Commission, 2012, 2014), some of which would designate all biospecimens and their derived data as identifiable (US Department of Health and Human Services and the Food and Drug Administration, 2011).

I.2 Our Proposal

To address the privacy concerns on person-level genomic information as well as site-level summary statistics, this thesis introduces a practical protocol to securely perform genotype-phenotype association studies via meta-analysis across multiple sites in large consortia (Fig. III.2). The protocol leverages cryptographic technologies to provide provable security guarantees. Unlike alternative proposals such as (Kamm et al., 2013), in the protocol of this thesis, constituent study sites retain full control of their respective individual participants' data and local site analyses. This allows each site to independently make appropriate adjustments to their own data or analyses, and account for site-specific differences in study design, which is pervasive in multi-site genetic association studies but not supported in (Kamm et al., 2013). Our protocol also allows each sites to contribute to the joint meta-analysis *without* exposing site-level summary statistics of studies. This could enable a wide range of collaborative studies across institutions that would otherwise be impossible due to concerns over privacy breaches on site-level summaries, or institutional confidentiality. The comprehensive protections aforementioned make our protocol impervious to most popular privacy attacks over genomic data at both the person- and site-level (Section IV.7).

In this thesis, we demonstrate the design and implementation of our secure protocol (which we call *SecureMA*) for supporting genotype-phenotype association studies via meta-analysis. To show the efficacy of our proposal, we also provide extensive empirical evaluations with three multi-site meta-analyses of genetic association studies from several large consortia, including the Electronic Medical Records and Genomics (eMERGE) network (McCarty et al., 2011) and the Population Architecture using Genomics and Epidemiology (PAGE) group (Fesinmeyer et al., 2013).

CHAPTER II

Background

Before delving into details of the proposed protocol, we first introduce the relevant background information such as genotype-phenotype association studies, meta-analysis, and cryptographic systems.

II.1 Genotype-Phenotype Association Studies

This work focuses on (securely) supporting genotype-phenotype association studies, a widely-used technique for detecting traits that are likely to have caused the disease under investigation. This is achieved by testing for the statistical correlation between genetic variants (i.e., genes or genome regions) and disease status (i.e., phenotypes) on the population (Lewis and Knight, 2012). Association studies have enjoyed continuing research investigation in the past decade, and have led to numerous important discoveries, some of which are listed in the NHGRI GWAS Catalog (Welter et al., 2014).

In a typical association study, the population under study is often categorized into groups of cases (e.g., people exhibiting the disease) and controls (e.g., reference population not infected with disease). And SNPs are the most commonly used genetic markers to test upon in association studies. A high SNP allele frequency among the case group implies that the tested SNP is likely to be a contributing factor to the disease. The level of association correlation is determined via statistical tests and reported in terms of correlation significance p-values and power sizes (e.g., β-coefficients). P-values under 0.05 are deemed as significant.

To ensure the reliability of association findings and eliminate false positives in correlation, large sample sizes are often desirable for increased statistical power. A growing number of collaboration consortia are founded to enable large-scale genotype-phenotype association studies which can span many data-contributing sites from different geographic

locations. For instance, our work uses data from the eMERGE network (McCarty et al., 2011) and the PAGE group (Fesinmeyer et al., 2013).

II.2 Cryptosystems and Relevant Secure Computations

In this section, we present a general description of the cryptography and secure computation techniques involved in the SecureMA protocol.

II.2.1 (Threshold) Paillier Cryptosystem

In this work, we leverage a "semantically secure" homomorphic public-key encryption (HPE) framework to protect certain genomic information. Generally speaking, in a public-key encryption system, a person, say Alice, generates two keys: 1) a *public key*, which is made available to another entity, say Bob, who wishes to communicate messages to Alice in an encrypted manner (i.e., the ciphertext) and 2) a *private key*, which is known only to Alice and is applied to decrypt the ciphertext sent by Bob.

An encryption scheme is said to be semantically secure when it is infeasible for an adversary (with finite computational capability), say Mallory, to gain knowledge about a message when it observes a ciphertext and the corresponding public encryption key. This property implies that even when the same message is encrypted multiple times, the ciphertexts will be indistinguishable to Mallory. In other words, if Bob and Charlie encrypt the same genotype-phenotype association statistics, say a regression coefficient with a value of 10 using the same public key, then the resulting ciphertexts will appear to be different. This mechanism further enhances the security of the encryption scheme (e.g., by protecting against attacks which leverage a pre-computed lookup table with raw data and their corresponding encryptions).

In addition, we require the encryption framework to possess an "additive homomorphic" property. This enables the computation of the encrypted sum of two messages to be completed using only the corresponding ciphertexts (e.g., without decryption).

Paillier Cryptosystem. The Paillier cryptosystem (Paillier, 1999) is a probabilistic

public key encryption protocol with a high confidentiality guarantee. Its additive homomorphic property enables direct support for several arithmetic operations, including addition and multiplication by a constant value, over encrypted data.

The following provides a basic introduction to Paillier encryption:

- Keys: Let $n = pq$, where p and q are large prime numbers, and $\lambda = lcm(p-1, q-1)$, where $lcm(.)$ denotes the function for least common multiple. We define function $L(x) = (x-1)/n$ and let g be an integer, such that $gcd(L(g^{\lambda} \, mod \, n^2), n) = 1$, where $gcd(.)$ is the function for greatest common divisor. The public and private cryptographic keys then consist of (n, g) and (p, q, λ), respectively. Note that there is only one private key.

- Encryption: The encryption of a message m (e.g., the value of a regression coefficient) into a ciphertext c is accomplished by $E(m, r) = g^m r^n \, mod \, n^2$, where g and n correspond to the public key, and r is a random value. For future reference, we will simply refer to this value as $E(m)$.

- Decryption: The decryption of a ciphertext c is computed as:

$$D(c) = \frac{L(c^{\lambda} \mod n^2)}{L(g^{\lambda} \mod n^2)} \mod n$$

Threshold Paillier Cryptosystem. Public key cryptosystems are vulnerable in that the system can be compromised if a private key is disclosed (either unintentionally or maliciously). To enhance the security of the system and ensure that the participants cannot easily violate the protocol, a private key can be split into l distinct "shares", where each share is provided to a different participant (e.g., a data manager in our protocol). This variation on cryptography is called a "threshold" system because it requires at least w out of the l participants to correctly decrypt information for Alice. When fewer than w participants attempt to decrypt, the system will be unable to reveal the corresponding message.

A threshold version of the Paillier cryptosystem was introduced in (Cramer et al., 2001) and was utilized in our protocol. In practice, we assume that the majority of participants in cryptographic systems are honest and thus, it is unlikely collusion will lead to illegitimate decryption. To achieve good security in practice, we set $w > \frac{2}{3}l$ according to the Byzantine fault tolerance principle (Lamport et al., 1982).

For the purposes of the SecureMA protocol, these participants correspond to the data managers who help maintain the encrypted summary statistics of genotype-phenotype associations (as introduced later in Section III.2). To perform decryption, the participants independently decrypt the result of the meta-analysis to obtain partial decryptions. The scientist who issued the original study inquiry will complete the decryption process by aggregating these partial decryptions.

II.2.2 Yao's Garbled Circuits

Yao's garbled circuits is a cryptographic technique which enables two untrusting parties to jointly evaluate a function without revealing their respective private input data (Yao, 1982). It guarantees that nothing is revealed but the final output of the function. In SecureMA, Yao's garbled circuits is leveraged to support a portion of the secure division operation (see Sections IV.5 and IV.6).

The basic idea of garbled circuits is as follows. First, one party called the *garbler*, prepares a "garbled" version of a binary circuit representing the function to be computed. Then the following are transmitted to a second party called the *evaluator*: the garbled circuit, the garbled inputs from the garbler, and the mapping between garbled-circuit outputs and raw bit values. Later on, the evaluator will initiate an oblivious transfer protocol (Naor and Pinkas, 1999) with the garbler and obliviously computes the circuit output without disclosing any intermediate values.

Our garbled circuit implementation, *CircuitService*, is built on top of FastGC (Huang et al., 2011), a Java-based framework which incorporates several optimizations to achieve

state-of-the-art performance. Our extensions to FastGC include speed-up optimizations through reusing the "offline" preparation of circuits, and the design of customized circuits.

CHAPTER III

SecureMA for Secure Meta-analysis

In this chapter, we introduce our novel protocol, *SecureMA*, for securely performing genotype-phenotype association studies in large consortia via meta-analysis. This chapter is organized as follows: first, we introduce meta-analysis and the computations involved; then we describe our SecureMA proposal in detail, showing how it can be applied to support meta-analysis securely. We postpone the description of specific sub-protocols underlying SecureMA to the next chapter (see Chapter IV).

Our protocol assumes a *semi-honest* threat model (i.e., honest-but-curious), a common assumption in many cryptographic systems. This means that each participants in the system are expected to always follow the protocol specification, but may try to infer additional information from what they see in the process.

III.1 Meta-analysis of Genotype-Phenotype Association Studies

To perform genetic association studies across multiple study sites, a commonly used method is meta-analysis. It is a statistical technique for contrasting and aggregating different studies to reach consistent conclusions, and has seen wide adoption in many scientific disciplines (Olkin, 1985).

Genotype-phenotype association studies can take advantage of meta-analysis to obtain larger sample sizes, leading to more robust inference of associations. In this work, we focus on the *inverse-variance* (or fixed-effect) approach to perform meta-analysis (Willer et al., 2010), which computes the average of the effect size weighted by the inverse of its variance:

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

where β is the aggregate effect size, se is the aggregate standard error, β_i is the effect

size of association for the i^{th} constituent study (i.e., one site contributing to meta-analysis), $w_i = 1/se_i^2$ is the weighting term, and se_i corresponds to the standard error of the effect for the i^{th} constituent study.

In our secure protocol, we square Equation III.1 to simplify the calculations because there is no straightforward and efficient way to compute the square root in a cryptographic setting. Following the transformation, our goal is represented as:

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

We point out that once Z^2 is obtained, the final square root and conversion from Z-score to p-value can be easily performed by the software.

In later sections, we will describe how Equation III.2 can be supported in secure.

III.2 Overview of the SecureMA Protocol

The proposed SecureMA protocol consists of two main steps: 1) a one-time Setup step which helps prepare the system for subsequence computations, and 2) the Secure Computation step which involves the actual computations of meta-analysis. The Setup initializes the system by: i) generating and distributing the cryptographic keys (for encryption and decryption), ii) encrypting summary results of local genetic association studies at each study site, and iii) submitting the site-specific encrypted summaries to their respective 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 without revealing their original content (Fig. III.2).

Protocol Participants. Before going into further details of each step, we first summarize the various participants in SecureMA categorized by their roles:

- A *Scientist* (e.g., genomicist) issues meta-analysis study queries to the system 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, conduct their local association studies at their respective institutions, and contribute to the joint meta-analysis via data sharing.

- (Optional) The *Data Managers* (or simply referred to as Managers when without confusion) manage the (encrypted) genomic information on behalf of *local sites*. This optional optimization makes the protocol more practical by supporting meta-analysis while reducing the number of participants required during the running of the protocol (e.g., one manager can delegate multiple local sites). The data managers only have limited decryption capabilities (as introduced later) and thus are not able to learn the content of the hosted data. In practice, these can be coordinating centers or other organizations entitled to manage encrypted data. The use of external data managers is optional.

- The *Mediator* computes the meta-analysis equations securely, and responds to the scientist's queries with encrypted results. Note that the mediator is not capable of learning the content of the data or computations, since they are all carefully protected throughout the process.

III.3 Setup Step of SecureMA

To setup the SecureMA protocol, a one-time step for generating and disseminating the public/private keys is coordinated by a trusted authority – the Key Manager – who is isolated from the SecureMA system and is not involved in any data management or computations (Fig. III.1).[1] To make the cryptographic system more secure, we leverage a threshold Paillier cryptosystem which protects the private key via secret share (Shamir, 1979) such that no corrupt minority of key holders can compromise the system (details in Section II.2.1).

[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. This role could be played by a semi-trusted (outside) third-party organization with a good reputation, a trustworthy computing module, or even a virtual party representing a distributed and secure mechanism for key generation.

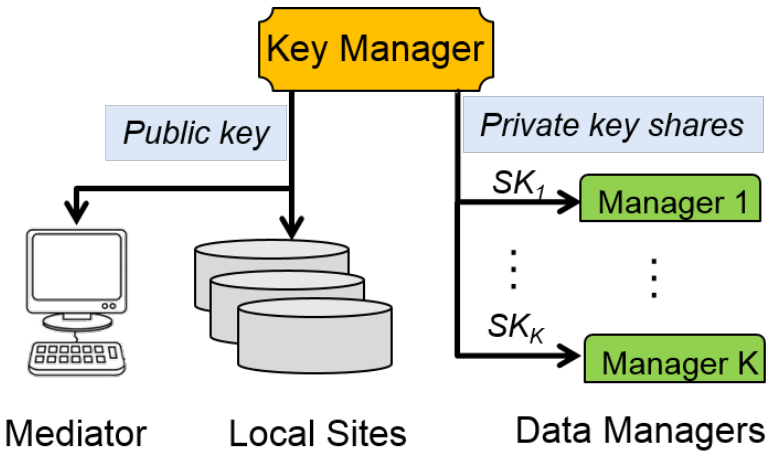


Figure III.1: The Setup step of the SecureMA protocol: cryptographic keys are generated and disseminated. The public key (for encryption) is broadcast to the mediator and local sites, while the private key (for decryption) is split into secret shares (SK_1, ..., SK_K) which are securely transmitted to the respective data managers.

Specifically, the private key is split into multiple secret shares and distributed across the participants in the system (i.e., the i^{th} participant receives the i^{th} share of the private key SK. In SecureMA, these participants correspond to the various data managers). By doing so, to successfully decrypt data, collaboration is required between the majority of key holders. As detailed in Section II.2.1, the splitting of the key enforces an "honest-majority" guarantee to mitigate collusion for illicit decryption. The public key PK is direclty broadcast to relevant participants (e.g., local sites, and the mediator).

Once the preparation on cryptographic keys is complete, the SecureMA protocol can proceed as individual study sites encrypt their local study summaries (i.e., properly scaled w_i and $w_i\beta_i$) with the public key PK, and participate in the joint meta-analysis by contributing (encrypted) data.

Optionally, to make the protocol more practical and eliminate administrative complexity, several intermediate parties – Data Managers – can be setup to host the (encrypted aggregate) data on behalf of the local sites. 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. Following this scheme, the local sites submit encryptions of their

study summaries (e.g., effect size and the inverse of the variance) to their entrusted data managers and can then go offline.[2] We emphasize that, unless massive collusions occur, the data managers are *not* capable of decrypting or inferring the content of the hosted data encryptions, due to the "honest majority" rule enforced by the threshold Paillier cryptosystem.

III.4 Secure Computation Step of SecureMA

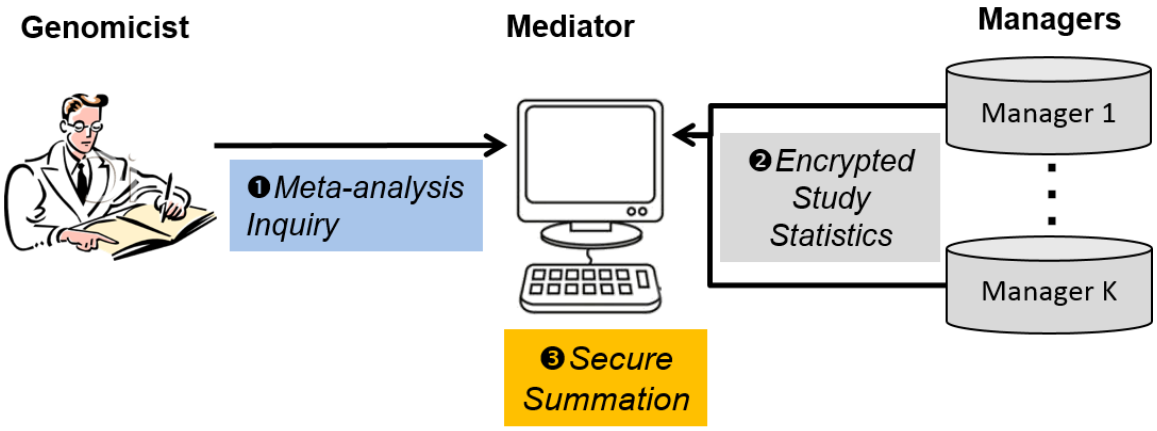
The actual computation of meta-analysis is initiated when a scientist submits a study query to SecureMA (Fig. III.2a). Upon receiving the query, the mediator requests the relevant data encryptions of site-specific association summaries (i.e., $E(w_i)$, $E(\beta_i w_i)$) from data managers. Upon receiving the data encryptions, the mediator securely aggregates them using the *ADD* sub-protocol (Section IV.2). This process yields encryptions $E(\sum \beta_i w_i)$ and $E(\sum w_i)$ for Equation III.2.

Next, the mediator coordinates with one randomly selected data manager to securely perform the division calculation in order to derive the weighted average of the effect size (Fig. III.2b), which is the final operation of meta-analysis (as shown in Equation III.2). This is achieved by following a two-party secure division sub-protocol *DIV* (Section IV.5).

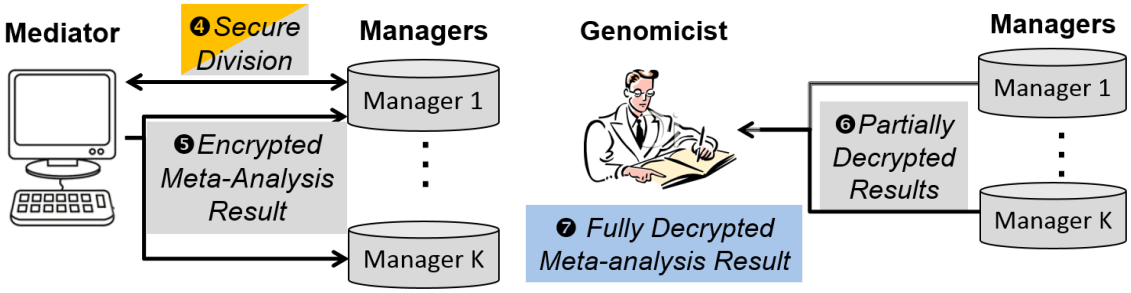
Up to this point, the meta-analysis result is still in an encrypted state. The mediator is then responsible for initiating a final round of collaborative decryption by distributing the result encryption to a majority of the data managers for partial decryption (Fig. III.2c). By collecting a sufficient number of the partially decrypted shares from the data managers, the scientist combines them to reveal the final complete decryption (i.e., Z^2). 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.

Once the squared Z-score, Z^2, is obtained, the final result to the scientist's study inquiry could be derived automatically according to the instructions in Section III.1.

[2] In rare occasions when necessary, the local sites can come back online to provide additional data.



(a) Secure summation



(b) Secure division

(c) Result decryption

Figure III.2: 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.

We provide a complete activity diagram in Fig. III.3 to better illustrate the series of computation procedures and interactions in SecureMA.

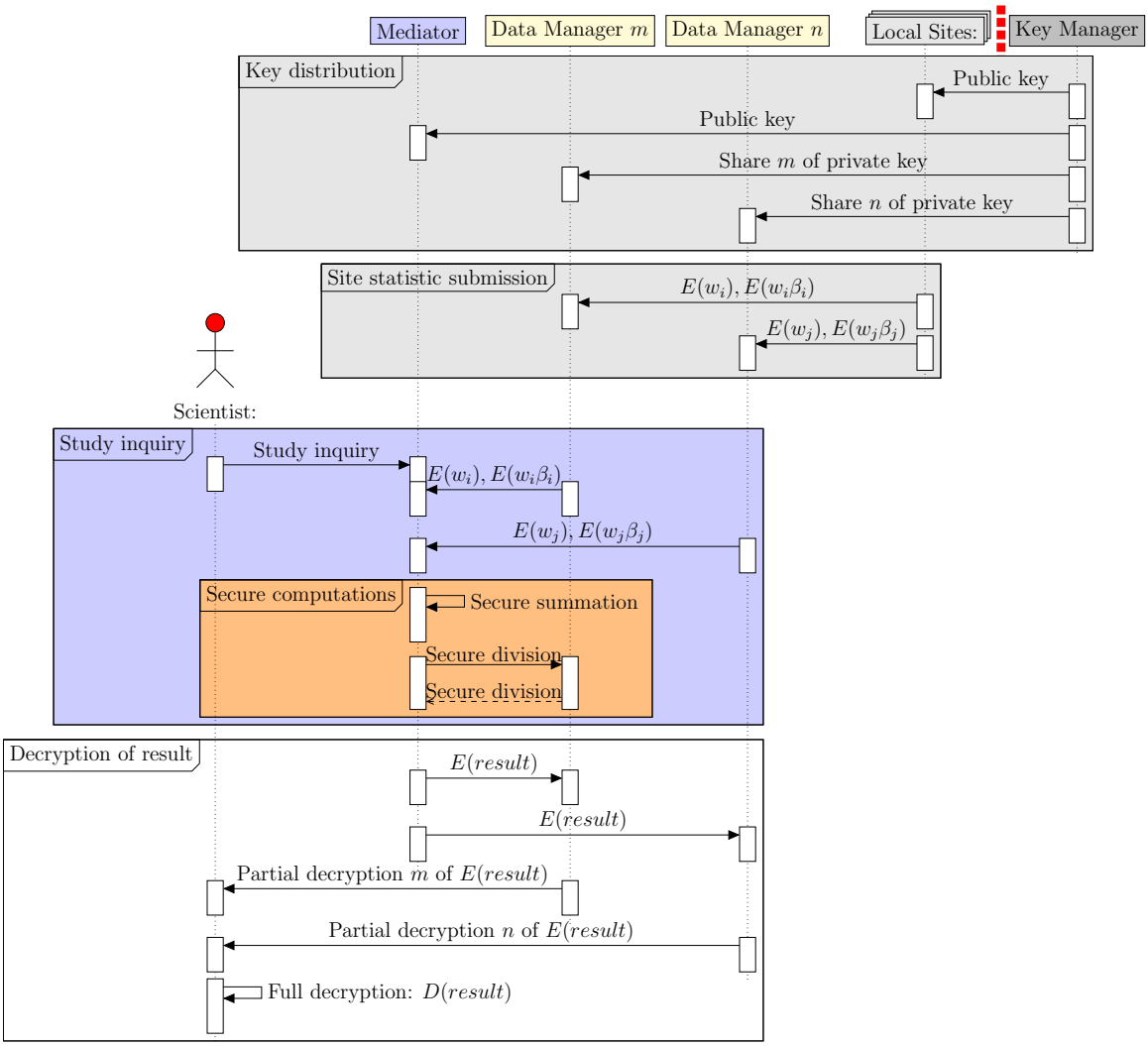


Figure III.3: A complete activity diagram of the SecureMA protocol. Denoted in gray boxes is the one-time Setup step covering key distribution and submission of encrypted site-specific statistics. In a typical running, a scientist issues a study query to start the protocol, and obtains the study result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

CHAPTER IV

Secure Computation Sub-protocols

In this chapter, we delve into the technical details of specific cryptographic sub-protocols underlying SecureMA, by introducing a series of secure transformations and arithmetic primitives for supporting the meta-analysis computation. Specifically, we show how to convert between data representations for our targeted cryptographic schemes (Section IV.1), how to securely aggregate encrypted data (Section IV.2), how to multiply an encryption by a constant value (Section IV.3), how to securely subtract (Section IV.4), and how to derive the division between two encrypted values (Section IV.5). At the end of the section, we briefly analyze the privacy guarantees of the proposed protocols.

IV.1 *SHARE*: Converting Paillier Encryptions to Secret Shares

The *SHARE* sub-protocol is designed to transform data encrypted via the Paillier cryptosystem to randomized values based on a two-party secret-share scheme (Shamir, 1979). It is necessary because the secure logarithm sub-protocol *LOG* (introduced later in Section IV.6) requires input data to be in the form of secret shares, while all data in our protocol are Paillier encryptions which are not compatible with the requirement.

First, we formally define the conversion sub-protocol: given the Paillier encryption $E(x)$ of a secret value x, the goal is to find two values x_1 and x_2, such that $x_1 + x_2 = x$. The two values are randomized to ensure that it is *not* possible to predict the value of one from the other (note that x is not revealed). This is accomplished as follows: first, one participant (i.e., a data manager in our case) generates a random value *rand* to obfuscate the given value $E(x)$ by computing $E(x + rand)$ via the secure addition sub-protocol *ADD* (introduced later in Section IV.2). The resulting encryption $E(x + rand)$ is then securely transmitted to the other participant (i.e., the mediator in our case). Later a decryption process helps obtain the mediator's data share $x_2 = x + rand$, while the data manager holds his share $x_1 = -rand$.

This means that the two participants are collaboratively holding a shared secret (i.e., the x value, which is equal to $x_1 + x_2$).

IV.2 *ADD*: Secure Addition

The Paillier cryptosystem supports secure summation directly through an additive homomorphic property (Section II.2). Given two encrypted values $E(m_1), E(m_2)$, the encryption of the sum $(m_1 + m_2)$ can be computed as:

$$\begin{aligned} E(m_1 + m_2) &= g^{m_1+m_2} \cdot r^n \mod n^2 \\ &= (g^{m_1} \cdot r_1^n) \cdot (g^{m_2} \cdot r_2^n) \mod n^2 \\ &= E(m_1) \cdot E(m_2) \mod n^2, \end{aligned} \tag{IV.1}$$

where r_1, r_2 are random values, $r = r_1 r_2$, and n is a parameter in the Paillier cryptosystem (as introduced in Section II.2.1).

IV.3 *MULC*: Secure Multiplication-by-Constant

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

$$E(k \cdot m) = g^{k \cdot m} \cdot r^n \mod n^2 = (E(m))^k \mod n^2 \tag{IV.2}$$

IV.4 *SUB*: Secure Subtraction

Since the Paillier cryptosystem is only "additive" homormorphic, subtraction is not natively supported. We observe that secure subtraction (*SUB* sub-protocol) can be achieved by first negating the subtrahend and converting it into an addition problem, which can then take advantage of the existing multiplication-by-constant (i.e., *MULC*) and addition (i.e., *ADD*)

sub-protocols described above. In brief, given two encryptions $E(m_1)$ and $E(m_2)$, the subtraction $(m_1 - m_2)$ can proceed in secure as:

$$E(m_1 - m_2) = ADD(E(m_1),\ MULC(E(m_2), -1)) \tag{IV.3}$$

IV.5 *DIV*: Secure Division

As shown in Equation III.2, meta-analysis requires a 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 (i.e., the *DIV* sub-protocol) into a subtraction problem using a secure logarithmic transformation (the *LOG* sub-protocol introduced later in Section IV.6).

For simplicity, we denote: $a = \sum_i \beta_i w_i$ and $b = \sum_i w_i$ (see Equation III.2). Via the logarithmic transformation, the goal of meta-analysis in Equation III.2 becomes:

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

We leverage the secure logarithm sub-protocol *LOG* introduced later to compute $\ln a$ and $\ln b$ for the transformed division operation (where both a and b are secret values). The final Z^2 value can be easily derived by taking the exponential, $\exp(.)$, on the final subtraction result.

IV.6 *LOG*: Secure Logarithmic Transformation

As described earlier, a secure logarithmic transformation (i.e., *LOG* sub-protocol) is utilized in SecureMA to perform the secure division operation of meta-analysis. Our *LOG* sub-protocol builds upon the secure $\ln x$ protocol in (Lindell and Pinkas, 2000). More formally, given a private 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.6.1 Logarithm Phase 1: Rough Estimate via Garbled Circuit

In the first phase, the logarithm $\ln x$ is (roughly) approximated by 2^y leveraging Yao's garbled circuits (see Section II.2.2) 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 Yao's garbled circuits involve two participants and no intermediate information other than the function output 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.6.2 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 (Naor and Pinkas, 1999), such that a secure polynomial from one participant is computed on the data contributed by the other participant without disclosing private information (such as the polynomial coefficients or private inputs of the participants). 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 in an encrypted state.

IV.6.3 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 further 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 III.1.

IV.7 Security Analysis

Here we provide a brief analysis on the privacy guarantees of the SecureMA protocol. In this thesis, we regard the privacy of genomic data to be breached if the original identities associated with the shared data, or other sensitive information such as medical conditions, are directly revealed or could be inferred by the protocol participants.

We prove the security of the protocol by using Goldreich's Composition Theorem (Goldreich, 2001). Briefly, it aims at showing that the view of the messages received (typically measured by its distribution) from each participant during the execution of the protocol can be effectively simulated given the input of that participant and the global output. In other words, we want to prove that all participants learn nothing except for the final output. In below, we show how each of the SecureMA sub-protocols preserves privacy.

Theorem IV.7.1. *ADD and MULC sub-protocols are privacy-preserving.*

Proof. Since both secure addition and multiplication-by-constant are supported by default as part of the additive homomorphic property of the Paillier cryptosystem, the security proof in the original publication applies (Paillier, 1999). □

Theorem IV.7.2. *SUB sub-protocol is privacy-preserving.*

Proof. Since *SUB* is a direct composition of *ADD* and *MULC* (both of which prove to be privacy-preserving), *SUB* preserves privacy. □

Theorem IV.7.3. *SHARE sub-protocol is privacy-preserving.*

Proof. We observe that the only message exchanged during the process is $E(x + rand)$. Since *rand* is uniformly distributed, its decryption $(x + rand)$ is computationally indistinguishable from a uniform distribution, according to the variation distance analysis in (Hall et al., 2011). As a result, *SHARE* is also privacy-preserving. □

Theorem IV.7.4. *LOG sub-protocol is privacy-preserving.*

Proof. Since our enhancement to the original $\ln x$ protocol in (Lin et al., 2004) does not affect its cryptographic properties, the security analysis on $\ln x$ in (Lindell and Pinkas, 2000) applies. □

Theorem IV.7.5. *DIV sub-protocol is privacy-preserving.*

Proof. Since *DIV* is a direct composition of the *LOG* and *SUB* sub-protocols (both of which prove to be privacy-preserving), we conclude that *DIV* preserves privacy. □

Since all underlying sub-protocols preserve privacy, we conclude that the SecureMA protocol is privacy-preserving.

Here we provide some intuition with respect to the biomedical implications of the privacy guarantees of SecureMA. Throughout the 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 person-level genome sequences (e.g., no risk of unique identifiability based on the uniqueness of SNPs as posed by (Lin et al., 2004)).

Moreover, site-level summaries (e.g., genetic association study statistics of each local site) are protected via strong encryption throughout the protocol. And the final meta-analysis results (limited to aggregate p-values only) are only made known to the original issuer of the study inquiry. Such protections make it impossible to perform inference attacks based on study statistics or allele frequencies or regression coefficients, which are features relied upon in various attacks, such as in (Homer et al., 2008; Jacobs et al., 2009; Sankararaman et al., 2009; Im et al., 2012).

CHAPTER V

Experimental Results

To demonstrate the efficacy of our proposal, we implemented the SecureMA protocol, and carried out a series of empirical evaluations on its computation accuracy, running time efficiency, and sensitivity to certain protocol parameters. In this chapter, we describe our software implementations, study data, our evaluation on three real-world meta-analysis studies, and additional proof of scalability of the protocol.

V.1 Software Implementations & Experiment Settings

We implemented the SecureMA protocol in the Scala and Java programming languages and provide programmable interfaces (API) for Java. The entire software package can be deployed as a single Java Archive (JAR) package and run on any computing platform where the Java Virtual Machine (JVM) is available.

Our SecureMA software is released open-source:
http://github.com/XieConnect/SecureMA

Our customized Yao's garbled circuits framework, *CircuitService*, is also released open-source:
http://github.com/XieConnect/CircuitService

All the experiments were performed on a quad-core Xeon computer (2.4 GHz, 4 GB memory), running 64-bit Ubuntu system and Java 1.7. We simulated the different participants of the protocol using separate system processes communicating via local network connections. All experiments were performed without parallelization to avoid potential interference of running time between different system processes.

V.2 Study Data

In this work, we used three sets of study data for evaluation, which are summarized below.

The eMERGE hypothyroidism study. The first collection of datasets is from a GWAS on hypothyroidism (Denny et al., 2011) provided by the eMERGE network (McCarty et al., 2011). It consists of 6,370 study participants across five study sites who contributed data: i) the Group Health Cooperative, ii) the Marshfield Clinic, iii) the Mayo Clinic, iv) Northwestern University Medical Center, and v) Vanderbilt University Medical Center. 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. Local-site studies were adjusted for birth decade and sex following the approach described in (Denny et al., 2011).

The PAGE obesity study. The second collection of datasets is from a genetic association study on obesity and body mass index (Fesinmeyer et al., 2013) provided by the PAGE consortia (Matise et al., 2011). It 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). 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. Local-site studies were completed following the processing procedures described in (Fesinmeyer et al., 2013).

The EAGLE diabetes study. The third collection of datasets is from a genetic association study on Type II Diabetes provided by the EAGLE group (Haiman et al., 2012), 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. It contains 14,998 participants and spans several ethnicities (e.g., non-Hispanic white, non-Hispanic black, Mexican-American, and others). We analyzed

216 SNPs. The published study (Haiman et al., 2012) did not report p-values for all SNPs and, thus, for comparison, we only focus on a controlled benchmark of the result accuracy using the standard non-secure meta-analysis as the baseline.

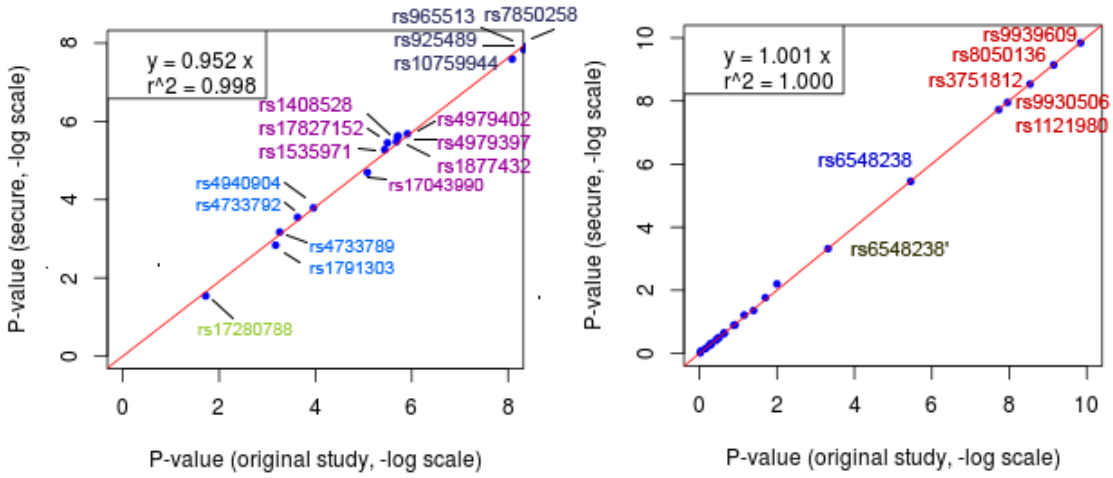
V.3 Accuracy of Genetic Association Results

We compared the accuracy of our secure computations with those reported by the original studies associated with these datasets (Denny et al., 2011; Fesinmeyer et al., 2013) (EAGLE is excluded from comparison due to lack of published p-values as baseline). These comparisons are summarized as QQ-plots of the SNP association p-values on a negative logarithmic scale (Fig. V.1). The plots for the eMERGE and PAGE studies 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 are 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 approaches produce highly consistent results.

V.4 Result Accuracy in a Controlled Setting

In the main text, we pointed out that 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 (Willer et al., 2010) instead of using the reported results from their original studies).

The comparisons are reported as QQ-plots on a negative logarithmic scale (Fig. V.2).



(a) eMERGE accuracy

(b) PAGE accuracy

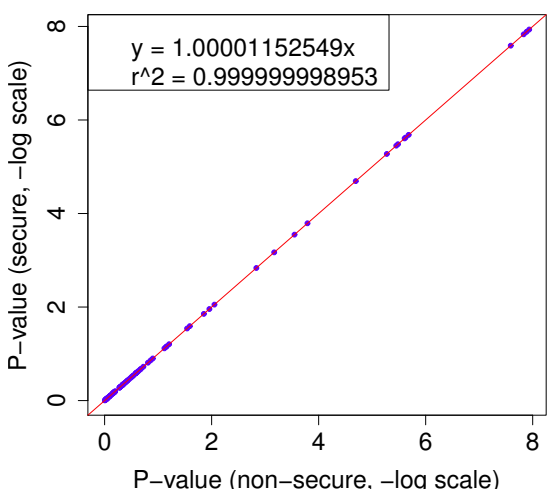
Figure V.1: Meta-analysis result accuracy from SecureMA. The correlation plots correspond to: (a) the p-values (secure protocol vs. original publication) based on the 16 SNPs from eMERGE; (b) the p-values (secure protocol vs. original publication) based on the 25 SNP-ethnicity pairs from PAGE (all SNPs annotated correspond to one ethnicity subpopulation, except for rs6548238', which corresponds to another)

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. 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 our extensive sensitivity analysis (Section V.6).

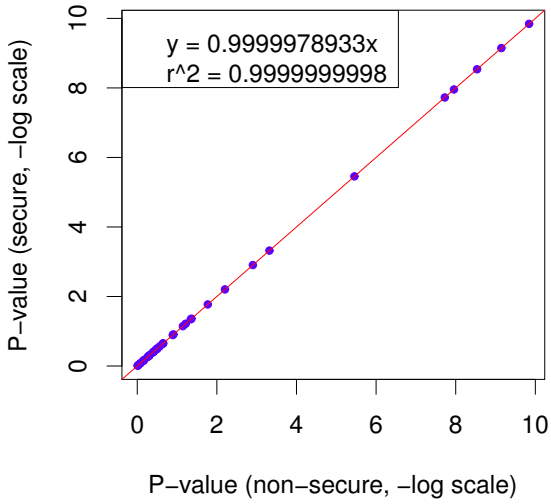
V.5 Running Time Efficiency

To evaluate the running time of the protocol, we performed a series of experiments using the aforementioned system settings. All times are reported based on the actual user time (instead of the CPU time).

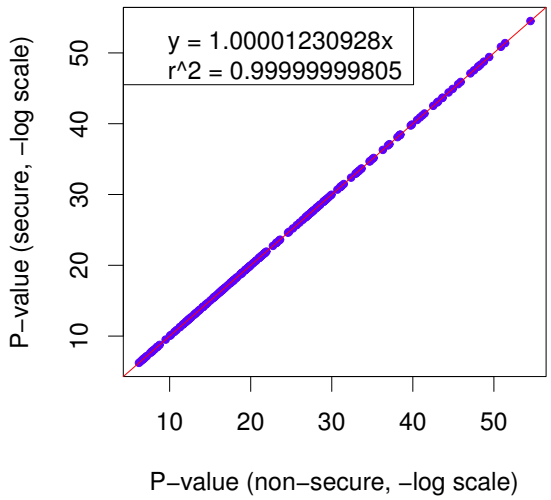
On average, the secure meta-analysis for most SNPs completed in 1.20 to 1.34 seconds (with a standard deviation $\leq$ 0.024 seconds) and no SNP required more than 1.38 seconds (Table V.1). In comparison to the eMERGE and PAGE datasets, the EAGLE study consumed slightly more time, due to the fact that EAGLE consists of much larger numeric



(a) eMERGE



(b) PAGE



(c) EAGLE

Figure V.2: 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.

values which leads to longer processing time.

Table V.1: Per-SNP running time for SecureMA and the proportion of the time dedicated to the 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)

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[1]. This implies that our protocol can be efficient even in studies with very large sample sizes, which is common for GWAS in large consortia.

Number of sites. We also point out that the majority of the running 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 V.1). 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.

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. V.3). 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.

[1] Individual research 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.

V.6 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 fractional values, nor division over encryptions, are directly supported in cryptographic protocols. Here we demonstrate the impact of these parameters both theoretically and empirically.

V.6.1 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 Equation III.2.

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.4). Briefly, $\ln x$ can be approximated as follows:

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

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 V.1 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.

V.6.2 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. V.4 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. V.4a), 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.4 diverge in opposite directions, which results in a magnification of the total error.

Nonetheless, in Fig. V.4b) 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.

V.6.3 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

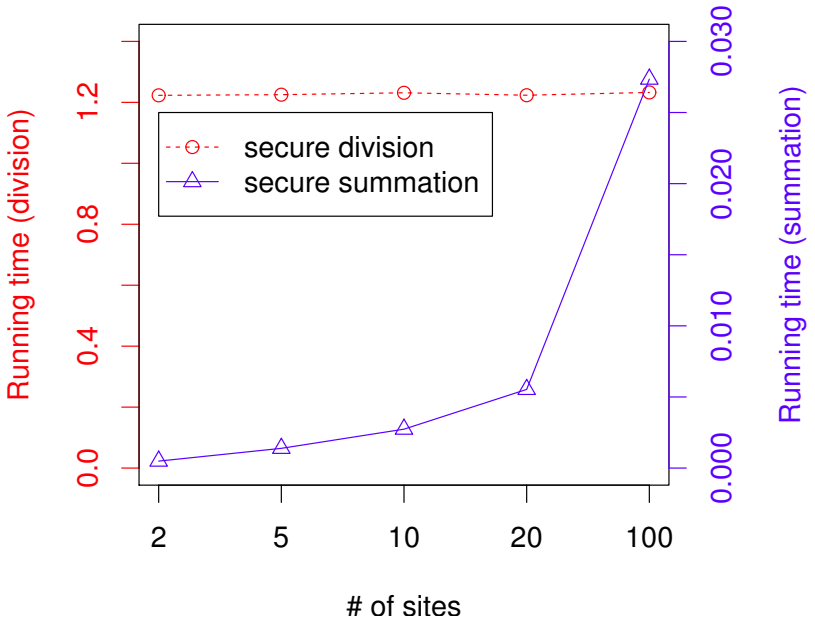
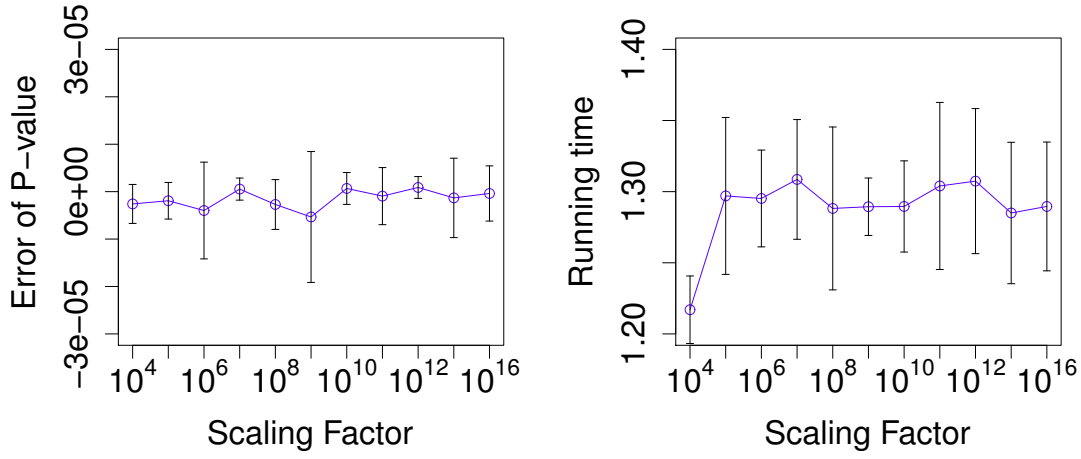


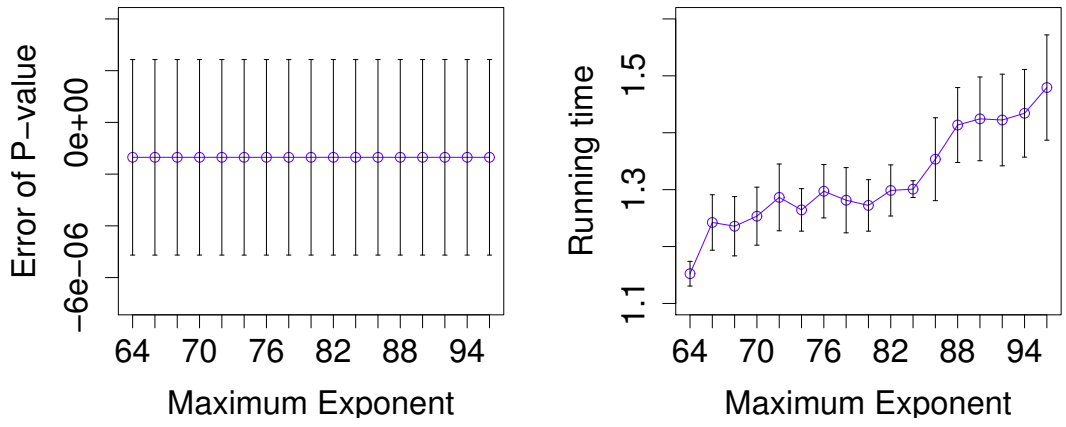
Figure V.3: Average running time of SecureMA, per SNP, as a function of the number of sites providing data (all times reported in seconds).



(a) Accuracy

(b) Running time (in seconds)

Figure V.4: 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).



(a) Accuracy

(b) Running time (in seconds)

Figure V.5: 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).

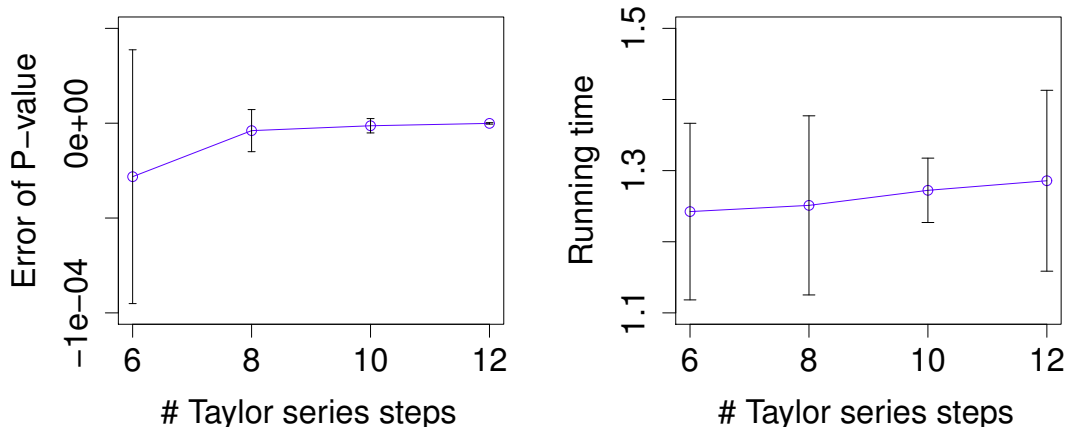
the upper bound for the exponent estimate. The second step corresponds to the application of a Taylor series, which we discuss in further depth below.

Fig. V.5 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. V.5b). However, it can be seen that the computational accuracy is almost identical across all test cases (Fig. V.5a). This is because, in this particular scenario, the other two protocol parameters are the dominating factors regarding computational accuracy.

V.6.4 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. V.6 shows how the computational error and the overall running time, respectively, of the secure meta-analysis is affected as the number of steps in



(a) Accuracy

(b) Running time (in seconds)

Figure V.6: The impact of the number of steps in the Taylor series (i.e., k in Equation V.1) on (a) computational accuracy and (b) running time efficiency. The results are based on 10 SNPs from the eMERGE dataset (mean +/- one standard deviation).

the series varies from 6 to 12. For context, SecureMA uses a default value of 10.

Fig. V.6a illustrates that the more steps in the Taylor series, the better the computational accuracy is on average. Fig. V.6b further demonstrates that there is a slight linear 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.

CHAPTER VI

Discussion

VI.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 (e.g., (Kantarcioglu et al., 2008; Kamm et al., 2013)) whose running time is positively correlated, in a linear and sometimes exponential manner, with the number of study participants and/or sites.

At the same time, the SecureMA protocol can be made more computationally efficient to support analysis on a genome-wide scale. First, the SecureMA protocol can easily be run in parallel on large computer clusters or cloud computing servers because each SNP can be analyzed independently. Thus the total running time for a large-scale GWAS via SecureMA 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 III.1). In such a scenario, the time-consuming secure division calculation could be avoided entirely, reducing the overall running time per SNP to milliseconds (10^{-3} seconds).

Third, recent advances in the optimization of cryptographic protocols (e.g., (Asharov et al., 2013; Henecka and Schneider, 2013)) may be ready to transition into practice in the near future. This could allow for certain sub-protocols in SecureMA, such as secure division, to have significant gains in efficiency.

VI.2 Limitations & Future Work

We recognize that 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) (real-world scenarios include deposited data in dbGaP (Mailman et al., 2007)). To support more "dirty" data in the wild, it will be necessary to embed QC processes for meta-analysis in the protocol (Winkler et al., 2014). Certain QC procedures may be vulnerable to attacks on privacy, but those which are based on standard algebraic computations should be translatable into secure computations by leveraging existing sub-protocols. At the same time, it should be noted that many procedures can be directly applied in the clear in a distributed fashion at each site because they do not violate privacy (e.g., file-level QC and SE-N plots in (Winkler et al., 2014)). Since QC is a relatively independent and large pipeline, we leave it for future work.

Second, the current SecureMA implementation relies on a trusted authority to generate cryptographic keys, which sometimes may not be desirable (alternative solutions are discussed in Section III.3).

Third, in situations when person-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 (Altman et al., 2013)).

CHAPTER VII

Related Work

To provide context for the contributions of our SecureMA protocol, we briefly review other recent developments with respect to privacy protection. There are generally two categories of data protection mechanisms that have been proposed to maintain participant privacy while supporting scientific investigations on genomic data: i) societal and regulatory protections, and ii) technological protections. In addition to examining the recent developments in these categories, we also briefly describe the latest trends in cryptographic solutions in general.

VII.1 Societal & Regulatory Protections

From a societal and regulatory perspective, it has been suggested that research participants consent to the risk of being re-identified (Lunshof et al., 2008) (which could bias participant recruitment), while users of such data (such as scientists) contractually agree not to attempt to re-identify the participants (Taylor, 2008). We believe such mechanisms can lower risk. However, while data use agreements assign liability, they do not provide any technological deterrent and can only be enforced when violations *could* be detected.

VII.2 Technological Protections

At the same time, various technological solutions have been proposed to promise privacy on genomic data. Methods (Lin et al., 2002; Malin, 2005) based on the classical k-anonymity model (Sweeney, 2002) have not seen wide adoption in the field because genomic data themselves contain both identifying information and scientific utility which makes it very challenging, if not impossible, to balance the two conflicting goals. Alternative proposals (Fienberg et al., 2011; Johnson and Shmatikov, 2013) based on differential privacy (Dwork, 2006) or noise addition in general (Lin et al., 2004) also turn out to be prob-

lematic, because their underlying concept of adding noisy data to manipulate and protect GWAS seems against the strong emphasis on computational accuracy by genomicists (Naveed et al., 2014). Thus in this work, we primarily review cryptographic solutions for protecting genomic data.

There are various cryptography-based proposals for securely managing genomic data. Some of these methods focus on protecting genome sequences: for instance, encrypting genome sequences and supporting simple queries of statistics (Kantarcioglu et al., 2008), encrypting identities of genes and variants to prevent unique identification of their carriers from rare variants (Singh et al., 2013), and obfuscating raw (short) genome sequences and allowing for retrieval (Ayday et al., 2014). Other proposals aim at supporting common genetic tests, such as enabling popular genetic tests (e.g., paternity tests, ancestry and genealogical tests, and tests for personalized medicine) without disclosing personal sequences (De Cristofaro et al., 2012), and protecting the test of genetic relatives via cryptographic solutions without revealing raw genotypes (He et al., 2014; Hormozdiari et al., 2014). More recently, several solutions have been proposed to securely conduct GWAS. These include splitting the regression analysis into local-site computations and center-level aggregation to shield person-level records from attacks (Wolfson et al., 2010), hosting person-level genomic data securely using secret share and facilitating GWAS (Kamm et al., 2013), and protecting genomic data with an efficient homomorphic encryption and customized implementation for various analytics on the genome (Lauter et al., 2014).

We point out that the two alternatives (Wolfson et al., 2010; Kamm et al., 2013) most relevant to our proposal, as discussed briefly in Section I.2, are hampered by practical limitations. First, it has been suggested that (Wolfson et al., 2010) may leak sensitive information because local sites inappropriately disclose intermediate summary statistics during the computation (El Emam et al., 2013; Sparks et al., 2008); The other recent proposal based on secret share (Kamm et al., 2013) fails to account for site-specific control variables and other data preprocessing steps within sites, which is a common practice for multi-site

genetic association studies. Their solution may also suffer from computational scalability and network bottleneck issues in studies with large sample sizes and strict security requirement, because higher-level security requires more servers to secret-share the data and all individual genomic data have to pass through, and be analyzed by, every server.

VII.3 Cryptographic Solutions in General

Cryptographic protocols, or secure multi-party computation (SMC), have seen increasing adoption in many areas where data privacy is gaining awareness. For instance, they are used for secure auction (Bogetoft et al., 2009), for safe-guarding machine learning tasks such as decision tree (Lindell and Pinkas, 2000), matrix factorization (Nikolaenko et al., 2013), Hidden Markov Models (Aliasgari and Blanton, 2013) and other algorithms (Graepel et al., 2013), for novel biomedical applications such as (El Emam et al., 2013).

At the same time, there is also encouraging progress in making cryptographic protocols more practical. Yao's garbled circuits have been significantly accelerated due to (Huang et al., 2011; Asharov et al., 2013; Henecka and Schneider, 2013); fully homomorphic encryption is also gaining computational efficiency due to recent progress (Brakerski and Vaikuntanathan, 2014); tools are being designed to make it more accessible for general users to adopt cryptographic protocols (Bogdanov et al., 2008; Zhang et al., 2013).

CHAPTER VIII

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 participant privacy and/or institutional confidentiality over genomic data is of concern. If appropriately implemented, our protocol can prevent privacy intrusions on genomic data posed by the attacks published to date. While there are opportunities to make this protocol computationally more efficient and to incorporate quality control procedures, 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.

BIBLIOGRAPHY

Aliasgari, M. and Blanton, M. (2013). Secure computation of hidden markov models. In *SECRYPT*, pages 242–253.

Altman, R. B., Clayton, E. W., Kohane, I. S., Malin, B. A., and Roden, D. M. (2013). Data re-identification: societal safeguards. *Science*, 339(6123):1032.

Asharov, G., Lindell, Y., Schneider, T., and Zohner, M. (2013). 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.

Ayday, E., Raisaro, J. L., Hengartner, U., Molyneaux, A., and Hubaux, J.-P. (2014). Privacy-preserving processing of raw genomic data. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 133–147. Springer.

Bogdanov, D., Laur, S., and Willemson, J. (2008). Sharemind: A framework for fast privacy-preserving computations. In *Computer Security-ESORICS 2008*, pages 192–206. Springer.

Bogetoft, P., Christensen, D. L., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., Nielsen, J. D., Nielsen, J. B., Nielsen, K., Pagter, J., et al. (2009). Secure multiparty computation goes live. In *Financial Cryptography and Data Security*, pages 325–343. Springer.

Brakerski, Z. and Vaikuntanathan, V. (2014). Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871.

Cramer, R., Damgård, I., and Nielsen, J. B. (2001). *Multiparty computation from threshold homomorphic encryption*. Springer.

De Cristofaro, E., Faber, S., Gasti, P., and Tsudik, G. (2012). Genodroid: are privacy-preserving genomic tests ready for prime time? In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, pages 97–108. ACM.

Denny, J. C., Crawford, D. C., Ritchie, M. D., Bielinski, S. J., Basford, M. A., Bradford, Y., Chai, H. S., Bastarache, L., Zuvich, R., Peissig, P., et al. (2011). 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.

Dwork, C. (2006). Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer.

El Emam, K., Samet, S., Arbuckle, L., Tamblyn, R., Earle, C., and Kantarcioglu, M. (2013). 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.

Erlich, Y. and Narayanan, A. (2014). Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15(6):409–421.

European Commission (2012). 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. (29 June 2014, date last accessed).

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

Fesinmeyer, M. D., North, K. E., Ritchie, M. D., Lim, U., Franceschini, N., Wilkens, L. R., Gross, M. D., Bůžková, P., Glenn, K., Quibrera, P. M., et al. (2013). 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.

Fienberg, S. E., Slavkovic, A., and Uhler, C. (2011). Privacy preserving gwas data sharing. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 628–635.

Fullerton, S. M., Anderson, N. R., Guzauskas, G., Freeman, D., and Fryer-Edwards, K. (2010). Meeting the governance challenges of next-generation biorepository research. *Science translational medicine*, 2(15):15cm3–15cm3.

Goldreich, O. (2001). Foundation of cryptography (in two volumes: Basic tools and basic applications).

Graepel, T., Lauter, K., and Naehrig, M. (2013). Ml confidential: Machine learning on encrypted data. In *Information Security and Cryptology–ICISC 2012*, pages 1–21. Springer.

Green, E. D., Guyer, M. S., Institute, N. H. G. R., et al. (2011). Charting a course for genomic medicine from base pairs to bedside. *Nature*, 470(7333):204–213.

Gymrek, M., McGuire, A. L., Golan, D., Halperin, E., and Erlich, Y. (2013). Identifying personal genomes by surname inference. *Science*, 339(6117):321–324.

Haiman, C. A., Fesinmeyer, M. D., Spencer, K. L., Bůžková, P., Voruganti, V. S., Wan, P., Haessler, J., Franceschini, N., Monroe, K. R., Howard, B. V., et al. (2012). 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.

Hall, R., Fienberg, S. E., and Nardi, Y. (2011). Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27(4):669.

He, D., Furlotte, N. A., Hormozdiari, F., Joo, J. W. J., Wadia, A., Ostrovsky, R., Sahai, A., and Eskin, E. (2014). Identifying genetic relatives without compromising privacy. *Genome research*, 24(4):664–672.

Henecka, W. and Schneider, T. (2013). 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.

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J. V., Stephan, D. A., Nelson, S. F., and Craig, D. W. (2008). Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167.

Hormozdiari, F., Joo, J. W. J., Wadia, A., Guan, F., Ostrosky, R., Sahai, A., and Eskin, E. (2014). Privacy preserving protocol for detecting genetic relatives using rare variants. *Bioinformatics*, 30(12):i204–i211.

Huang, Y., Evans, D., Katz, J., and Malka, L. (2011). Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, volume 201.

Humbert, M., Ayday, E., Hubaux, J.-P., and Telenti, A. (2013). 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.

Im, H. K., Gamazon, E. R., Nicolae, D. L., and Cox, N. J. (2012). 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.

Jacobs, K. B., Yeager, M., Wacholder, S., Craig, D., Kraft, P., Hunter, D. J., Paschal, J., Manolio, T. A., Tucker, M., Hoover, R. N., et al. (2009). A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies. *Nature genetics*, 41(11):1253–1257.

Johnson, A. and Shmatikov, V. (2013). Privacy-preserving data exploration in genome-wide association studies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1079–1087. ACM.

Kamm, L., Bogdanov, D., Laur, S., and Vilo, J. (2013). A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, 29(7):886–893.

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

Kaye, J., Heeney, C., Hawkins, N., de Vries, J., and Boddington, P. (2009). Data sharing in genomics—re-shaping scientific practice. *Nature Reviews Genetics*, 10(5):331–335.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401.

Lauter, K., Lopez-Alt, A., and Naehrig, M. (2014). Private computation on encrypted genomic data. 14^{th} Privacy Enhancing Technologies Symposium, Workshop on Genome Privacy. http://seclab.soic.indiana.edu/GenomePrivacy/papers/Genome%20Privacy-paper9.pdf. (29 July 2014, date last accessed).

Lewis, C. M. and Knight, J. (2012). Introduction to genetic association studies. *Cold Spring Harbor Protocols*, 2012(3):pdb–top068163.

Lin, Z., Hewett, M., and Altman, R. B. (2002). Using binning to maintain confidentiality of medical data. In *Proceedings of the AMIA Symposium*, page 454. American Medical Informatics Association.

Lin, Z., Owen, A. B., and Altman, R. B. (2004). Genomic research and human subject privacy. *Science*, pages 183–183.

Lindell, Y. and Pinkas, B. (2000). Privacy preserving data mining. In *Advances in Cryptology—CRYPTO 2000*, pages 36–54. Springer.

Lowrance, W. W. and Collins, F. S. (2007). Identifiability in genomic research. *Science*, 317:600–602.

Lunshof, J. E., Chadwick, R., Vorhaus, D. B., and Church, G. M. (2008). From genetic privacy to open consent. *Nature Reviews Genetics*, 9(5):406–411.

Mailman, M. D., Feolo, M., Jin, Y., Kimura, M., Tryka, K., Bagoutdinov, R., Hao, L., Kiang, A., Paschall, J., Phan, L., et al. (2007). The ncbi dbgap database of genotypes and phenotypes. *Nature genetics*, 39(10):1181–1186.

Malin, B. A. (2005). An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future. *Journal of the American Medical Informatics Association*, 12(1):28–34.

Matise, T. C., Ambite, J. L., Buyske, S., Carlson, C. S., Cole, S. A., Crawford, D. C., Haiman, C. A., Heiss, G., Kooperberg, C., Le Marchand, L., et al. (2011). 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.

McCarty, C. A., Chisholm, R. L., Chute, C. G., Kullo, I. J., Jarvik, G. P., Larson, E. B., Li, R., Masys, D. R., Ritchie, M. D., Roden, D. M., et al. (2011). The emerge network: a consortium of biorepositories linked to electronic medical records data for conducting genomic studies. *BMC medical genomics*, 4(1):13.

Naor, M. and Pinkas, B. (1999). Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254. ACM.

Naveed, M., Ayday, E., Clayton, E. W., Fellay, J., Gunter, C. A., Hubaux, J.-P., Malin, B. A., and Wang, X. (2014). Privacy and security in the genomic era. *arXiv preprint arXiv:1405.1891*.

Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., and Boneh, D. (2013). Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 801–812. ACM.

Olkin, I. (1985). Statistical methods for meta-analysis. *San Diego, CA: Academic*.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*, pages 223–238. Springer.

Panagiotou, O. A., Willer, C. J., Hirschhorn, J. N., and Ioannidis, J. P. (2013). The power of meta-analysis in genome wide association studies. *Annual review of genomics and human genetics*, 14:441.

Presidential Commission for the Study of Bioethical Issues (2012). Privacy and progress in whole genome sequencing. Washington, DC.

Rodriguez, L. L., Brooks, L. D., Greenberg, J. H., and Green, E. D. (2013). The complexities of genomic identifiability. *Science*, 339(6117):275–276.

Sankararaman, S., Obozinski, G., Jordan, M. I., and Halperin, E. (2009). Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

Singh, A. P., Zafer, S., and PE'ER, I. (2013). Metaseq: privacy preserving meta-analysis of sequencing-based association studies. In *Pac Symp Biocomput*, pages 356–367. World Scientific.

Sparks, R., Carter, C., Donnelly, J. B., O'Keefe, C. M., Duncan, J., Keighley, T., and McAullay, D. (2008). Remote access methods for exploratory data analysis and statistical modelling: Privacy-preserving analytics®. *Computer methods and programs in biomedicine*, 91(3):208–222.

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

Taylor, P. (2008). Personal genomes: when consent gets in the way. *Nature*, 456(7218):32–33.

US Department of Health and Human Services and the Food and Drug Administration (2011). 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.

Welter, D., MacArthur, J., Morales, J., Burdett, T., Hall, P., Junkins, H., Klemm, A., Flicek, P., Manolio, T., Hindorff, L., et al. (2014). The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic acids research*, 42(D1):D1001–D1006.

Willer, C. J., Li, Y., and Abecasis, G. R. (2010). Metal: fast and efficient meta-analysis of genomewide association scans. *Bioinformatics*, 26(17):2190–2191.

Winkler, T. W., Day, F. R., Croteau-Chonka, D. C., Wood, A. R., Locke, A. E., Mägi, R., Ferreira, T., Fall, T., Graff, M., Justice, A. E., et al. (2014). Quality control and conduct of genome-wide association meta-analyses. *Nature protocols*, 9(5):1192–1212.

Wolfson, M., Wallace, S. E., Masca, N., Rowe, G., Sheehan, N. A., Ferretti, V., LaFlamme, P., Tobin, M. D., Macleod, J., Little, J., et al. (2010). 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.

Yao, A. C. (1982). Protocols for secure computations. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE.

Zerhouni, E. A. and Nabel, E. G. (2008). Protecting aggregate genomic data. *Science*, 322(5898):44a.

Zhang, Y., Steele, A., and Blanton, M. (2013). Picco: a general-purpose compiler for private distributed computation. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 813–826. ACM.