BMC
Medical Informatics & Decision Making

PROCEEDINGS Open Access

Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption

Wen-Jie Lu[1*], Yoshiji Yamada[3], Jun Sakuma[1,2]

From 4th iDASH Privacy Workshop
San Diego, CA, USA. 16 March 2015

Abstract

Objective: Developed sequencing techniques are yielding large-scale genomic data at low cost. A genome-wide association study (GWAS) targeting genetic variations that are significantly associated with a particular disease offers great potential for medical improvement. However, subjects who volunteer their genomic data expose themselves to the risk of privacy invasion; these privacy concerns prevent efficient genomic data sharing. Our goal is to presents a cryptographic solution to this problem.

Methods: To maintain the privacy of subjects, we propose encryption of all genotype and phenotype data. To allow the cloud to perform meaningful computation in relation to the encrypted data, we use a fully homomorphic encryption scheme. Noting that we can evaluate typical statistics for GWAS from a frequency table, our solution evaluates frequency tables with encrypted genomic and clinical data as input. We propose to use a packing technique for efficient evaluation of these frequency tables.

Results: Our solution supports evaluation of the D' measure of linkage disequilibrium, the Hardy-Weinberg Equilibrium, the χ^2 test, etc. In this paper, we take χ^2 test and linkage disequilibrium as examples and demonstrate how we can conduct these algorithms securely and efficiently in an outsourcing setting. We demonstrate with experimentation that secure outsourcing computation of one χ^2 test with 10, 000 subjects requires about 35 ms and evaluation of one linkage disequilibrium with 10, 000 subjects requires about 80 ms.

Conclusions: With appropriate encoding and packing technique, cryptographic solutions based on fully homomorphic encryption for secure computations of GWAS can be practical.

Introduction

Because of recent advances in DNA sequencing technologies, the cost of DNA sequencers is dropping rapidly. As a result, the scale of genomic data used by researchers is becoming larger and larger. To conduct computations on a large-scale genomic dataset, a cloud server that provides computational resources at low cost is regarded as a promising option.

It is difficult to argue that genomic and clinical data are highly sensitive. Outsourcing these data to an external server raises concerns about the privacy of sensitive data. Consequently, for outsourcing of computation with genomic data, privacy should be rigorously preserved.

The fully homomorphic encryption (FHE) scheme is attracting attention as a tool for secure outsourcing of data analysis. FHE enables encryption of data and then carrying out arbitrary computation using the encrypted data without decrypting the data. The first FHE scheme was proposed by Gentry [1]: subsequent improvements [2,3] provided more practical FHE schemes.

Actually, FHE has been applied to secure outsourcing of computation that involves genomic and clinical data. Bos et al. [4] proposed a working implementation of cloud service for private computation of encrypted health

* Correspondence: riku@mdl.cs.tsukuba.ac.jp
[1]Graduate School of Systems and Information Engineering, University of Tsukuba, Ten'nodai 1-1-1, Tsukuba, Japan
Full list of author information is available at the end of the article

data using FHE. Lauter et al. [5] demonstrated an approach to conducting private computation using encrypted genomic data with FHE. Unfortunately, these cryptographic solutions are not sufficiently time and space efficient to conduct a GWAS-scale computation, which can involve 300k SNPs for thousands or more subjects.

In this manuscript, we present a protocol for secure outsourced analysis of large-scale genomic data using FHE. Precisely, our proposed protocol evaluates a frequency table with encrypted genomic/clinical data as input. This enables us to outsource computation of typical statistics related to GWAS securely, such as the Hardy-Weinberg Equilibrium (HWE), χ^2 test for independence and Linkage Disequilibrium (LD). Our method works by virtue of the fact that we can pack integer vectors into a single ciphertext of a certain type of FHE. This packing technique enables us to evaluate a scalar product of integer vectors through a single homomorphic multiplication using the packing technique; such a batch style computation helps to conduct computation of GWAS-scale data in an efficient manner.

Our basic strategy is to compute allelic frequency tables and genotype frequency tables privately from encrypted genetic data. With these tables, GWAS-related statistics including D' measure of LD, the Pearson Goodness-of-Fit, HWE, and the χ^2 test are conducted. In this work particularly, we apply our method to the χ^2 test and LD to demonstrate the effectiveness of our protocol.

We review an allelic frequency table and a genotype frequency table with two markers. Table 1 gives a view of a genomic dataset D^g. Each record contains an explicit identifier ID and SNPs. Similarly, Table 2 gives a view of a phenotype dataset D^p. Each record contains an explicit identifier ID′ to identify each subject and an attribute to indicate the disease status of the subject. Presuming that M subjects and N SNPs are involved, then the dataset D^g contains N rows, with each row containing M data points; the dataset D^p includes M rows.

Presuming that A, a are possible alleles. An allelic frequency table (Table 3) consists of 2 × 2 counts

$$o_1 = 2N_{\mathrm{AA}}^{\mathrm{case}} + N_{\mathrm{Aa}}^{\mathrm{case}} \qquad o_2 = 2N_{\mathrm{aa}}^{\mathrm{case}} + N_{\mathrm{Aa}}^{\mathrm{case}}$$
$$o_3 = 2N_{\mathrm{AA}}^{\mathrm{control}} + N_{\mathrm{Aa}}^{\mathrm{control}} \quad o_4 = 2N_{\mathrm{aa}}^{\mathrm{control}} + N_{\mathrm{Aa}}^{\mathrm{control}}\,',$$

where $N_{\mathrm{AA}}^{\mathrm{case}}$ and $N_{\mathrm{Aa}}^{\mathrm{case}}$ are the observed population counts for genotype AA and Aa in the case group: $N_{\mathrm{Aa}}^{\mathrm{control}}$ and $N_{\mathrm{Aa}}^{\mathrm{control}}$ are the observed counts for the control group.

Table 1. Raw genome data D^g

ID	Genomic Data
1	CC CG CT GG AA
2	AG CT CT AG CT
3	CT GG CC AG AA
4	AA GG GG AG CC

Table 2. Raw phenotype data D^p

ID′	Disease Status
1	Case
2	Control
3	Control
4	Case

A χ^2 test for the additive model is equivalent to the χ^2 test based on Table 3. The one degree of freedom (d.f.) test statistic is written as

$$\chi_a^2 = \frac{2M(o_2(o_3 + o_4) - o_4(o_1 + o_2))^2}{N_1 N_2 N'_1 N'_2}.$$

In addition to a χ^2 test, we can evaluate the Hardy-Weinberg Equilibrium directly from an allelic frequency table similarly.

Given alleles (A/a and B/b) at two markers, a genotype frequency table (Table 4) with two markers is obtained that consists of 3 × 3 counts

$$\begin{aligned} o_{11} &= N_{\mathrm{AABB}}\ o_{12} = N_{\mathrm{AaBB}}\ o_{13} = N_{\mathrm{aaBB}} \\ o_{21} &= N_{\mathrm{AABb}}\ o_{22} = N_{\mathrm{AaBb}}\ o_{23} = N_{\mathrm{aaBb}} \\ o_{31} &= N_{\mathrm{AAbb}}\ o_{32} = N_{\mathrm{Aabb}}\ o_{33} = N_{\mathrm{aabb}}. \end{aligned}$$

The value $N_{ii'jj'}$ denotes the observed population counts for genotype i$i^{'}$ and $jj^{'}$ where $i, i' \in \{A, a\}$, and $j, j' \in \{B, b\}$.

We evaluate LD from Table 4. The linkage disequilibrium is calculated as $D = \mathrm{p_{AB}} - \mathrm{p_A p_B}$, where probabilities $\mathrm{p_{AB}}$, $\mathrm{p_A}$ and $\mathrm{p_B}$ are computed, respectively, as $(2o_{11} + o_{12} + o_{21})/2M$, $(2N'_1 + N'_2 - o_{22})/2M$ and $(2N_1 + N_2 - o_{22})/2M$. We omit the frequency o_{22} to avoid the problem of haplotype ambiguity, especially when only genotypes are measured. See [6] for more details.

We remark that several measures for measuring linkage disequilibrium were proposed, including Pearson's correlation, Lewontin's D', frequency difference and Yule's Q. Our proposal works for all these measures. However, we applied our method to Lewontin's D' measure in the experimentation because of space limitations. Additional details related to these measurements are explained in an earlier report of the literature [6].

Table 3. Observed allele frequency in a case-control study of *M* subjects

	Allele Type		total
	A	a	
case	o_1	o_2	N_1
control	o_3	o_4	N_2
total	N'_1	N'_2	2M

Table 4. Genotype frequencies at markers M_1 and M_2 of M subjects

		Marker M_1			Total
		AA	Aa	aa	
Marker M_2	BB	o_{11}	o_{12}	o_{13}	N_1
	Bb	o_{21}	o_{22}	o_{23}	N_2
	bb	o_{31}	o_{32}	o_{33}	N_3
	Total	N'_1	N'_2	N'_3	$2M$

Problem settings and threat model

Problem settings

For our secure outsourcing of GWAS, we consider three stakeholders, *data contributors*, *researchers*, and *the cloud*. The data contributors (e.g. hospitals, research institutes or subjects) contribute private genomic or clinical data to the cloud. A researcher is an entity that wishes to conduct a GWAS. The cloud is an untrusted entity that includes researchers and data contributors with computational resources.

We assume that genotype/phenotype data of one subject can be contributed from different contributors. In other words, datasets D^g and D^p can be horizontally or vertically partitioned and can receive contributions from different contributors. Additionally, we assume that all subjects are identified with obfuscated IDs so that the cloud can correctly merge contributed data from two or more sources.

Given the contributed datasets D^g and D^p, the protocol proceeds as follows. 1) The cloud computes sufficient statistics with D^g and D^p, although it knows nothing about the contributed data and sends the resulting sufficient statistics to the researcher. 2) The researcher first reconstructs a frequency table from the sufficient statistics and then conducts GWAS.

Threat model

The goal of our system is to ensure that 1) the cloud server cannot learn anything about the private data contributed by data contributors beyond the public information, such as the total number of subjects; 2) the researcher cannot learn beyond what is revealed by the frequency table. Even in the case in which the cloud server colludes with some contributors, they still have no means to learn anything about the data contributed by other contributors except the final results.

In our setting, we assume that the cloud servers do not behave maliciously. However, the cloud server has motivation to learn some information related to the private data contributed by data contributors. This assumption naturally holds when the cloud server wishes to maintain a good reputation of their services. To avoid a man-in-the-middle attack, we assume that the key setup works correctly and that all data contributors obtain the correct encryption key from the analyst which can be enforced with appropriate use of Certificate Authorities. The Figure 1 to be described in the following section is thus designed to be secure against an honest-but-curious cloud server. Additional assumptions that must be made are the following.

1) The cloud server is not in collusion with the researcher to disclose private data contributed by data contributors. 2) Existence of a secure channel between data contributors and the cloud, e.g. SSH.

Methods

Before description of our protocol, we first introduce a homomorphic encryption and packing technique used as building blocks of our protocol.

Building block I: homomorphic encryption

Homomorphic encryption is a cryptosystem that allows performance of arithmetic operations of ciphertexts without decryption.

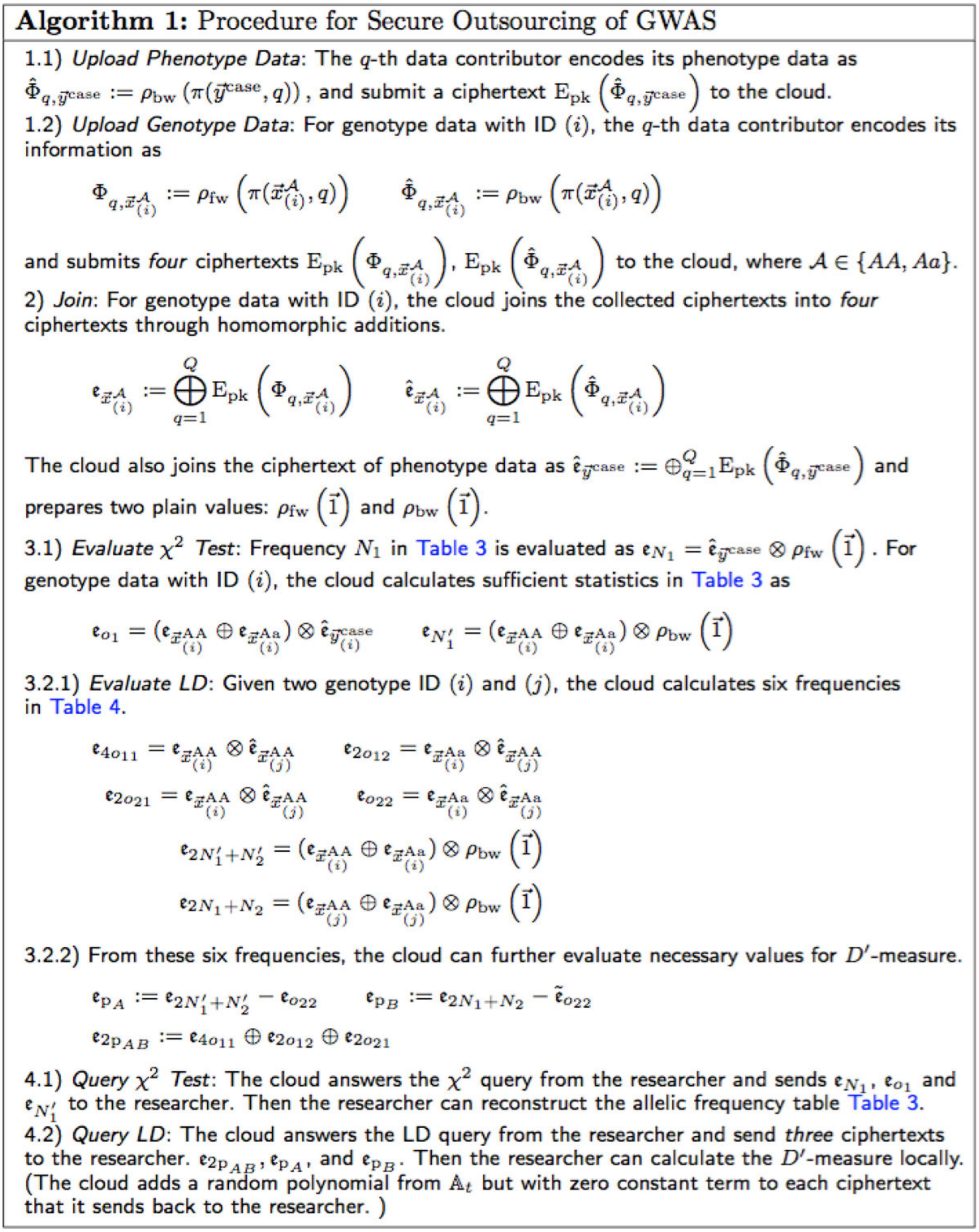
We detail a homomorphic encryption scheme based on ring-Learning with Errors (RLWE) assumption [7]. Let n be the lattice dimension of the scheme, where n is given as an integer of 2-power. Then, the message space of the scheme is given as a polynomial ring $\mathbb{A}_t : \mathbb{Z}_t[x]/(x^n + 1)$, where t is a prime number. Simply, we identify $\mathbb{A}_t$ with the set of integer polynomials of degree up to $n - 1$ reduced modulo t. Moreover, we identify modulo t in the interval $(-t/2, t/2]$.

For our implementation, we used HElib [8], which is an implementation of the Brakerski-Gentry-Vaikuntanathan (BGV) scheme proposed in [2]. The BGV's scheme is a public-key cryptosystem that supports homomorphic operations. Pre-suming that m_1, $m_2 \in \mathbb{A}_t$ are two plain polynomials and $\mathrm{E}_{\mathrm{pk}}\ (m_1)$, *then* $\mathrm{E}_{\mathrm{pk}}\ (m_2)$ are the corresponding ciphertexts encrypted by BGV's scheme under an encryption key pk. The BGV's scheme supports both homomorphic addition and multiplication:

$$\begin{aligned}
\mathrm{D}_{\mathrm{sk}}(\mathrm{E}_{\mathrm{pk}}(m_1) \oplus \mathrm{E}_{\mathrm{pk}}(m_2)) &\equiv m_1 + m_2 \mod (x^n + 1, t)\\
\mathrm{D}_{\mathrm{sk}}(\mathrm{E}_{\mathrm{pk}}(m_1) \otimes \mathrm{E}_{\mathrm{pk}}(m_2)) &\equiv m_1 \times m_2 \mod (x^n + 1, t)\\
\mathrm{D}_{\mathrm{sk}}(\mathrm{E}_{\mathrm{pk}}(m_1) \oplus c) &\equiv m_1 + c \mod (x^n + 1, t)\\
\mathrm{D}_{\mathrm{sk}}(\mathrm{E}_{\mathrm{pk}}(m_1) \otimes c) &\equiv m_1 \times c \mod (x^n + 1, t),
\end{aligned}$$

where $c \in \mathbb{A}_t$ and $\mathrm{D}_{\mathrm{sk}}(\cdot)$ is the decryption function using the corresponding decryption key sk. It is noteworthy that homomorphic multiplication costs much more time than a homomorphic addition does in terms of magnitude.

We remark that the BGV's scheme supports the evaluation of circuits that are not deeper than a pre-defined level L. In other words, L denotes the maximal depth of evaluable circuits. The scheme security was analyzed intensively by Gentry et al. in [9]. We omit details of the security analysis and state their results below. The following equation describes the lattice dimension n that



Algorithm 1: Procedure for Secure Outsourcing of GWAS

1.1) *Upload Phenotype Data*: The q-th data contributor encodes its phenotype data as $\hat{\Phi}_{q,\vec{y}^{\text{case}}} := \rho_{\text{bw}}\left(\pi(\vec{y}^{\text{case}}, q)\right)$, and submit a ciphertext $\mathrm{E}_{\text{pk}}\left(\hat{\Phi}_{q,\vec{y}^{\text{case}}}\right)$ to the cloud.

1.2) *Upload Genotype Data*: For genotype data with ID (i), the q-th data contributor encodes its information as

$$\Phi_{q,\vec{x}^{\mathcal{A}}_{(i)}} := \rho_{\text{fw}}\left(\pi(\vec{x}^{\mathcal{A}}_{(i)}, q)\right) \qquad \hat{\Phi}_{q,\vec{x}^{\mathcal{A}}_{(i)}} := \rho_{\text{bw}}\left(\pi(\vec{x}^{\mathcal{A}}_{(i)}, q)\right)$$

and submits *four* ciphertexts $\mathrm{E}_{\text{pk}}\left(\Phi_{q,\vec{x}^{\mathcal{A}}_{(i)}}\right)$, $\mathrm{E}_{\text{pk}}\left(\hat{\Phi}_{q,\vec{x}^{\mathcal{A}}_{(i)}}\right)$ to the cloud, where $\mathcal{A} \in \{AA, Aa\}$.

2) *Join*: For genotype data with ID (i), the cloud joins the collected ciphertexts into *four* ciphertexts through homomorphic additions.

$$\mathfrak{e}_{\vec{x}^{\mathcal{A}}_{(i)}} := \bigoplus_{q=1}^{Q} \mathrm{E}_{\text{pk}}\left(\Phi_{q,\vec{x}^{\mathcal{A}}_{(i)}}\right) \qquad \hat{\mathfrak{e}}_{\vec{x}^{\mathcal{A}}_{(i)}} := \bigoplus_{q=1}^{Q} \mathrm{E}_{\text{pk}}\left(\hat{\Phi}_{q,\vec{x}^{\mathcal{A}}_{(i)}}\right)$$

The cloud also joins the ciphertext of phenotype data as $\hat{\mathfrak{e}}_{\vec{y}^{\text{case}}} := \oplus_{q=1}^{Q} \mathrm{E}_{\text{pk}}\left(\hat{\Phi}_{q,\vec{y}^{\text{case}}}\right)$ and prepares two plain values: $\rho_{\text{fw}}\left(\vec{1}\right)$ and $\rho_{\text{bw}}\left(\vec{1}\right)$.

3.1) *Evaluate* χ^2 *Test*: Frequency N_1 in Table 3 is evaluated as $\mathfrak{e}_{N_1} = \hat{\mathfrak{e}}_{\vec{y}^{\text{case}}} \otimes \rho_{\text{fw}}\left(\vec{1}\right)$. For genotype data with ID (i), the cloud calculates sufficient statistics in Table 3 as

$$\mathfrak{e}_{o_1} = (\mathfrak{e}_{\vec{x}^{\text{AA}}_{(i)}} \oplus \mathfrak{e}_{\vec{x}^{\text{Aa}}_{(i)}}) \otimes \hat{\mathfrak{e}}_{\vec{y}^{\text{case}}_{(i)}} \qquad \mathfrak{e}_{N'_1} = (\mathfrak{e}_{\vec{x}^{\text{AA}}_{(i)}} \oplus \mathfrak{e}_{\vec{x}^{\text{Aa}}_{(i)}}) \otimes \rho_{\text{bw}}\left(\vec{1}\right)$$

3.2.1) *Evaluate LD*: Given two genotype ID (i) and (j), the cloud calculates six frequencies in Table 4.

$$\mathfrak{e}_{4o_{11}} = \mathfrak{e}_{\vec{x}^{\text{AA}}_{(i)}} \otimes \hat{\mathfrak{e}}_{\vec{x}^{\text{AA}}_{(j)}} \qquad \mathfrak{e}_{2o_{12}} = \mathfrak{e}_{\vec{x}^{\text{Aa}}_{(i)}} \otimes \hat{\mathfrak{e}}_{\vec{x}^{\text{AA}}_{(j)}}$$

$$\mathfrak{e}_{2o_{21}} = \mathfrak{e}_{\vec{x}^{\text{AA}}_{(i)}} \otimes \hat{\mathfrak{e}}_{\vec{x}^{\text{AA}}_{(j)}} \qquad \mathfrak{e}_{o_{22}} = \mathfrak{e}_{\vec{x}^{\text{Aa}}_{(i)}} \otimes \hat{\mathfrak{e}}_{\vec{x}^{\text{Aa}}_{(j)}}$$

$$\mathfrak{e}_{2N'_1+N'_2} = (\mathfrak{e}_{\vec{x}^{\text{AA}}_{(i)}} \oplus \mathfrak{e}_{\vec{x}^{\text{Aa}}_{(i)}}) \otimes \rho_{\text{bw}}\left(\vec{1}\right)$$

$$\mathfrak{e}_{2N_1+N_2} = (\mathfrak{e}_{\vec{x}^{\text{AA}}_{(j)}} \oplus \mathfrak{e}_{\vec{x}^{\text{Aa}}_{(j)}}) \otimes \rho_{\text{bw}}\left(\vec{1}\right)$$

3.2.2) From these six frequencies, the cloud can further evaluate necessary values for D'-measure.

$$\mathfrak{e}_{\mathrm{p}_A} := \mathfrak{e}_{2N'_1+N'_2} - \mathfrak{e}_{o_{22}} \qquad \mathfrak{e}_{\mathrm{p}_B} := \mathfrak{e}_{2N_1+N_2} - \tilde{\mathfrak{e}}_{o_{22}}$$

$$\mathfrak{e}_{2\mathrm{p}_{AB}} := \mathfrak{e}_{4o_{11}} \oplus \mathfrak{e}_{2o_{12}} \oplus \mathfrak{e}_{2o_{21}}$$

4.1) *Query* χ^2 *Test*: The cloud answers the χ^2 query from the researcher and sends $\mathfrak{e}_{N_1}$, $\mathfrak{e}_{o_1}$ and $\mathfrak{e}_{N'_1}$ to the researcher. Then the researcher can reconstruct the allelic frequency table Table 3.

4.2) *Query LD*: The cloud answers the LD query from the researcher and send *three* ciphertexts to the researcher. $\mathfrak{e}_{2\mathrm{p}_{AB}}$, $\mathfrak{e}_{\mathrm{p}_A}$, and $\mathfrak{e}_{\mathrm{p}_B}$. Then the researcher can calculate the D'-measure locally. (The cloud adds a random polynomial from $\mathbb{A}_t$ but with zero constant term to each ciphertext that it sends back to the researcher.)

Figure 1 Protocol of secure outsourcing of χ^2 test & linkage disequilibrium.

is necessary to evaluate deep-L circuits correctly with guarantee of κ-bits security,

$$n > \frac{(L(\log\ n + 23) - 8.5)\ (\kappa + 110)}{7.2}.$$

Building block II: packing technique

The BGV encryption scheme takes *polynomials* as plaintexts. An integer vector is transformed into a polynomial form. Then the encryption function takes as input the polynomial and outputs a ciphertext, which also forms a polynomial [10,11]. These techniques are called packing techniques.

Transformations introduced by Yasuda et al. [10] were designed originally for secure Hamming distance evaluation of binary vectors. We introduce their method and designate the method as *forward* and *backward* packing. Letting $\mathbb{A}_t$ be the given polynomial ring (with parameters n, t), and presuming that $\vec{u}$ and $\vec{v}$ are integer vectors with length ℓ, then forward packing ρfw $(\cdot)$ and backward packing ρbw $(\cdot)$ are defined respectively as

$$\rho_{\text{fw}}(\vec{u}) := \sum_{i=0}^{\ell-1} u_i x^i, \qquad \rho_{\text{bw}}(\vec{v}) := -\sum_{j=0}^{\ell-1} v_j x^{n-j}. \tag{1}$$

In the equations above, u_i is the i-th element of $\vec{u}$; u_j is the j-th element of $\vec{v}$. It is readily apparent that if v_i, $u_i \in (-t/2, t/2]$ for $0 \leq i < \ell$ and $\ell \leq n$, then ρfw and ρbw respectively transform vectors $\vec{u}$ and $\vec{v}$ into elements of the ring $\mathbb{A}_t$.

One benefit of this transformation is that homomorphic multiplication of the ciphertexts with this packing engenders a scalar product $\vec{u} \cdot \vec{v}$.

$$\begin{aligned}
&\mathrm{E}_{\mathrm{pk}}(\rho_{\mathrm{fw}}(\vec{u})) \otimes \mathrm{E}_{\mathrm{pk}}(\rho_{\mathrm{bw}}(\vec{v})) \\
&= \mathrm{E}_{\mathrm{pk}}\left(\sum_{i=0}^{\ell-1} u_i x^i \times \left(-\sum_{j=0}^{\ell-1} v_j x^{n-j}\right)\right) = \mathrm{E}_{\mathrm{pk}}\left(-\sum_{i=0}^{\ell-1}\sum_{j=0}^{\ell-1} u_i v_j x^{n+i-j}\right) \\
&= \mathrm{E}_{\mathrm{pk}}\left(\sum_{i=0}^{\ell-1} u_i v_j x^0 + \sum_{i=0}^{\ell-1}\sum_{j=0}^{j+h<\ell} u_{h+j} v_j x^h - \sum_{k=1}^{\ell-1}\sum_{j=0}^{j+k<\ell} u_i v_{j+k} x^{n-k}\right).
\end{aligned} \tag{2}$$

The scalar product between vectors $\vec{u}$ and $\vec{v}$ is obtained from the constant term of Equation 2. The remaining $2\ell - 2$ terms are unconcerned.

Equation 2 allows evaluation of a scalar product between two length-ℓ encrypted vectors only by a single homomorphic multiplication. The correctness of this evaluation is presented in Theorem 1.

Theorem 1 *Let n be lattice dimension and t be prime modulo. Let* $\vec{u}$ *and* $\vec{v}$ *denote length-*ℓ *vectors. Then, the constant term of the decryption* $D_{sk}(\mathfrak{e}_u \otimes \hat{\mathfrak{e}}_v)$, *where* $\hat{\mathfrak{e}}_v := E_{pk}(\rho_{bw}(\vec{v}))$ *and* $\hat{\mathfrak{e}}_v := E_{pk}(\rho_{bw}(\vec{v}))$, *gives the scalar product* $\langle\vec{u}, \vec{v}\rangle$ *if (1)* u_i, $v_i \in (-t/2, t/2]$ *for* $0 \leq i, j < \ell$; *(2)* $\ell \leq n$; *(3)* $\langle\vec{u}, \vec{v}\rangle \in (-t/2, t/2]$.

The proof was obtained immediately from the derivation of Equation 2 and so is omitted here.

Proposed secure outsourcing of GWAS

Recall that our goal is to outsource the evaluation of frequency tables efficiently while maintaining the genotype/phenotype data private to the cloud servers. We present an encoding scheme for genotype/phenotype data. Particularly, with this encoding, we can securely evaluate a frequency table through scalar products by the technique introduced into the previous section. We present a protocol for secure outsourcing GWAS in the last part of this section. The detail of the protocol is described in Figure 1.

Data encoding

Let A and a be the alleles of the biallelic locus. Consequently, the genomic data at the locus is either AA, Aa, or aa. We represent each row of the genomic dataset D^g as two integer vectors $\vec{x}^{AA}$, $\vec{x}^{Aa}$. Here, x_i^{AA}, the i-th element of $\vec{x}^{AA}$, represents the frequency of genotype AA at the marker locus: $x_i^{AA} = 2$ for AA and $x_i^{AA} = 0$ for other genotypes. x_i^{Aa} is similar to x_i^{AA} except that $x_i^{Aa} = 1$ for Aa.

We presume that the disease status of each subject is represented by a binary variable, then "disease" is represented by 1 (case); "non-disease" is represented by 0 (control). The phenotype dataset D^p for all subjects is therefore represented by a binary vector $\vec{y}^{case}$.

Presume in addition to the following that dataset D^g consists of N SNPs with M subjects. Q data contributors are involved in the procedure. Therefore, they separately hold the phenotype vector $\vec{y}^{\text{case}}$ and $2N$ genotype vectors $\vec{x}_{(i)}^{\text{AA}}$ and $\vec{x}_{(i)}^{\text{Aa}}$, where (i) is the ID of the genotype data. Let $\pi : \{0, 1, 2\}^M \times \{1, 2, ..., Q\} \mapsto \{0, 1, 2\}^M$ be an assignment function that represents the partition of genotype/phenotype held by the q-th data contributor. For example, the vertical partition of a vector $\vec{x}$ for the q-th data contributor is represented as shown below.

$$\pi(\vec{x}, q)_j = \begin{cases} x_j & \text{if } q \text{ - th data contributor holds the } j \text{ - th element of } \vec{x} \\ 0 & \text{o.w.} \end{cases}.$$

We assume that each element of vectors is contributed from only one data contributor, i.e. $\sum_q \pi(\vec{x}, q)_j = x_j$ holds for every j. For simplicity, we view $\pi(\vec{x}, q)$ as a polynomial whose j-th coefficient has value $\pi(\vec{x}, q)_j$.

We use this data encoding in Step 1.1 and Step 1.2 in Figure 1.

Evaluate the allelic frequency table

With the encoding described, we evaluate Table 3 through *scalar products* of the representing vectors. More specifically, frequencies o_1, N_2', and N_1 in Table 3 are evaluated respectively through three scalar products as

$$o_1 = \langle\vec{x}^{AA} + \vec{x}^{Aa}, \vec{y}^{case}\rangle, N_1' = \langle\vec{x}^{AA} + \vec{x}^{Aa}, \vec{1}\rangle, N_1 = \langle\vec{y}^{case}, \vec{1}\rangle,$$

Where $\vec{1}$ is a vector of which the elements are 1. Because Table 3 is freedom-1 and the number of objects M is assumed to be known, whole Table 3 can be reconstructed with values o_1, N_1' and N_1. Therefore, three homomorphic multiplications are needed here. Step 3.1 of Figure 1 shows that the three scalar products can be evaluated with homomorphic multiplication.

Evaluate the genotype frequency table

Similarly, we compute the genotype frequency table described by Table 4 with two markers by scalar products of the represented vectors as well. In particular, to calculate a D'-measure for the LD, the following six scalar products are needed.

$$\begin{aligned} 4 \cdot o_{11} &= \langle \vec{x}^{AA}, \vec{x}^{BB} \rangle, & 2 \cdot o_{12} &= \langle \vec{x}^{Aa}, \vec{x}^{BB} \rangle, \\ 2 \cdot o_{21} &= \left\langle \vec{x}^{AA}, \vec{x}^{Bb} \right\rangle, & o_{22} &= \left\langle \vec{x}^{Aa}, \vec{x}^{Bb} \right\rangle, \\ 2N'_1 + N'_2 &= \langle \vec{x}^{AA} + \vec{x}^{Aa}, \vec{1} \rangle, & 2N_1 + N_2 &= \left\langle \vec{x}^{BB} + \vec{x}^{Bb}, \vec{1} \right\rangle. \end{aligned}$$

Step 3.2.1 of Figure 1 shows that the six scalar products can be computed with homomorphic multiplication as well.

Secure outsourcing GWAS protocol

The procedure of secure outsourcing GWAS is shown in Figure 1. Recall that the evaluation of scalar product in Equation 2 requires a forward-packed vector and a backward-packed vector. Consequently, at Step 1.2, data contributors upload four copies for one genotype data in the form of the forward-packed and backward-packed vectors. The cloud aggregates the collected ciphertexts at Step 2, which only involves homomorphic additions. Then the cloud computes the allelic frequency table and the genotype frequency table respectively at Step 3.1 and 3.2.

Results

We benchmarked the computational costs of our method and compared it with a method proposed by Lauter et al. in [5], in which a genetic data point and a clinical data point are encoded respectively into three bits and two bits. All experiments were conducted on computers with a 2.60 GHz CPU (Xeon; Intel Corp.) and 32 GB RAM. We measured the computation time separately for Step 1.1 and 1.2 as the preparation time and for Steps 3.1 and 3.2 as the evaluation time. Details of the experiment settings are presented following. 1) An artificial dataset includes 1.0×10^4 subjects. 2) $Q = 5$ data contributors are sharing same quantity of data points. 3) We used 8 threads for computation in parallel. 4) Parameters of the encryption scheme were set as $n = 8192$, $t = 640007$, and $L = 6$.

Performance of homomorphic encryption and implementation hints

The implementation of Lauter et al. was done on an algebraic computation system, Magma, whereas our implementation was developed on native codes. To compare our method with their method fairly, we measured the computation time of operations in HElib and re-estimated the computation time method of Lauter et al. Table 5 shows the computation time of the operations of homomorphic encryption scheme. Values are the mean of 1000 runs of each operation with 8-threads. We used parameter $n = 8192$, which is not sufficiently large to conduct more than 8192 subjects. Indeed, we partitioned vectors into smaller parts and encrypted each part as a ciphertext. In doing so, we were able to conduct a large-scale dataset while maintaining smaller n. We remark that as the number of the partition increases, more communication time must be used during the upload phase.

Table 5. Timing of fully homomorphic scheme with parameters $n = 8192$, $t = 640007$, $L = 6$

Operation	Encrypt	Mult	Add	Add with Plaintext
Time (ms)	3.08	7.57	0.032	0.789

Artificial genotype & phenotype dataset

We benchmarked our proposed protocol of evaluating χ^2 test on an artificial dataset that contains 1.0×10^4 subjects. The results are presented in Figure 2. The number of the total SNPs was varied from 1.0×10^3 to 1.0×10^6. At Step 3.1 of the Figure 1, only three homomorphic multiplications are necessary to evaluate a χ^2 test statistics. Recalling that parameter $n = 8192$, one can thereby maximally pack genotype/phenotype data of 8192 subjects into a single ciphertext. Consequently, to conduct the experiment with 1.0×10^4 subjects, we partitioned a vector into two parts having equal length. Figure 2 depicts the performance of our proposed method and the estimated computation time of the method of Lauter et al. [5]. As shown in Figure 2, for evaluation of χ^2 test statistics of 1.0×10^6 SNPs with 1.0×10^4 subjects, our method took about 12 hours (about 43 ms per test).

The benchmark of the evaluation of LD is presented in Figure 3. In this experiment, we considered a smaller synthetic data containing 1.0×10^3 SNPs of 1.0×10^4 subjects. The number of LD to be evaluated with p SNPs is $p(p - 1)/2$. We therefore evaluated about 5.0×10^5 LDs in this experiment. With this settings, our method costs less than 11 hours (about 80 ms per LD).

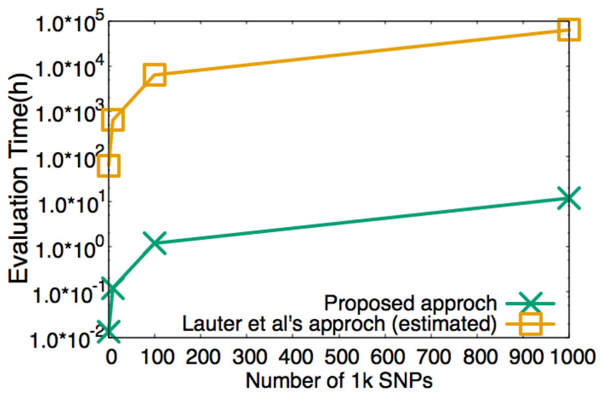


Figure 2 Benchmark for outsourcing χ^2 test with 1.0×10^4 subjects.

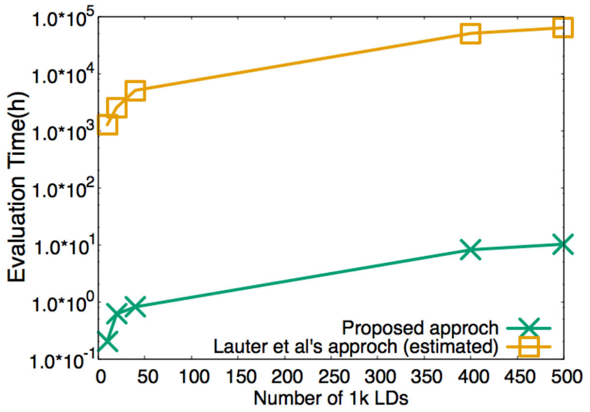


Figure 3 Benchmark for outsourcing LD with 1.0 × 10^4 subjects.

Conclusions

From Figure 2 and 2 we can see that Lauter et al's cryptographic solution [5] might take about 2000 days to conduct the evaluation of χ^2 test of one million SNPs and takes about 2600 days to conduct the evaluation of half million of linkage disequilibrium. At the meantime, it respectively took our approach about 12 hours and 11 hours to conduct the same computation. We conclude that with the appropriate encoding and packing technique, secure outsourcing of GWAS using FHE can be practical.

Related work

Studies of privacy-preserving data processing in GWAS involve different techniques. Kamm et al. proposed a secret sharing-based method in [12], by which private information is divided into several parts and is transferred to at least three collusion-free servers. All servers share the workload equally. The final result is aggregated from the output of each server. Computation based on secret-sharing requires multiple rounds of communication between servers; the computation is secret as long as no two servers collude. Because our outsourcing approach executes the whole computation with single cloud servers, computational environments employed for the computation are different.

A cryptographic solution was proposed recently from the work of Lauter et al. [5]. They constructed a method for computation on encrypted genomic data using a cryptosystem that is similar to BGV's scheme. Each genetic datum is encoded into three ciphertexts, which can cause inefficiency in both time and space. Our previous work [13] proposed a specified approach for secure outsourcing χ^2 test. In this manuscript we propose a more general approach for secure outsourcing of χ^2 test, HWE and LD etc.

An orthogonal method to ours is differential privacy [14]. With perturbation noise, differential privacy ensures that distribution of the output is insensitive to any data contributor's record, making it impossible to infer data from the obfuscated output. In our case, we can incorporate the perturbation noise in the query phase. Therefore, differential privacy can enforce the privacy properties of our protocol.

Competing interests

The authors declare that they have no competing interests related to this study.

Authors' contributions

Wen-jie Lu and Jun Sakuma designed the algorithm and drafted the majority of the manuscript. Wen-jie Lu conducted the experiments. Yoshiji Yamada gave useful comments on bio-information and provided genotype and phenotype data.

Acknowledgements

This work is supported by JST CREST program "Advanced Core Technologies for Big Data Integration" and is partly supported by JSPS KAKENHI 24680015. This article has been published as part of *BMC Medical Informatics and Decision Making* Volume 15 Supplement 5, 2015: Proceedings of the 4th iDASH Privacy Workshop: Critical Assessment of Data Privacy and Protection (CADPP) challenge. The full contents of the supplement are available online at http://www.biomedcentral.com/1472-6947/15/S5.

Declarations

Publication funding for this supplement was supported by iDASH U54HL108460, iDASH linked R01HG007078 (Indiana University), NHGRI K99HG008175 and NLM R00LM011392.

Authors' details

[1]Graduate School of Systems and Information Engineering, University of Tsukuba, Ten'nodai 1-1-1, Tsukuba, Japan. [2]JST CREST, Honchou 4-1-8, Kawaguchi, Japan. [3]Life Science Research Center, Mie University, Kurimamachiya-cho 1577, Tsu, Japan.

Published: 21 December 2015

References

1. Gentry C: **A fully homomorphic encryption scheme.** PhD thesis, Stanford University; 2009.
2. Brakerski Z, Gentry C, Vaikuntanathan V: **(Leveled) fully homomorphic encryption without bootstrapping.** *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ACM* 2012, 309-325.
3. Brakerski Z: **Fully homomorphic encryption without modulus switching from classical gapsvp.** *Advances in Cryptology-CRYPTO* 2012, 868-886.
4. Bos JW, Lauter K, Naehrig M: **Private predictive analysis on encrypted medical data.** *Journal of biomedical informatics* 2014, **50**:234-243.
5. Lauter K, López-Alt A, Naehrig M: **Private computation on encrypted genomic data.** *Progress in Cryptology-LATINCRYPT* 2014, 3-27.
6. Ziegler A, König IR: *A Statistical Approach to Genetic Epidemiology: Concepts and Applications.* 2nd edition. John Wiley & Sons, Berlin; 2010, 247-254.
7. Lyubashevsky V, Peikert C, Regev O: **On ideal lattices and learning with errors over rings.** *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques, Springer-Verlag* 2010, 1-23.
8. **HELib.** [http://shaih.github.io/HElib/index.html], Accessed: 2014-12-10.
9. Gentry C, Halevi S, Smart N: **Homomorphic evaluation of the AES circuit.** *Advances in Cryptology-CRYPTO* 2012, 850-867.
10. Yasuda M, Shimoyama T, Kogure J, Yokoyama K, Koshiba T: **Secure pattern matching using somewhat homomorphic encryption.** *Proceedings of the 2013 ACM CCSW ACM* 2013, 65-76.
11. Smart NP, Vercauteren F: **Fully homomorphic SIMD operations.** *Designs, codes and cryptography* 2014, **71(1)**:57-81.

12. Kamm L, Bogdanov D, Laur S, Vilo J: **A new way to protect privacy in large-scale genome-wide association studies.** *Bioinformatics* 2013.
13. Lu W, Yamada Y, Sakuma J: **Efficient secure outsourcing of genome-wide association studies.** *IEEE Symposium on Security and Privacy Workshops, SPW 2015, San Jose, CA, USA, May 21-22, 2015* 2015, 3-6.
14. Johnson A, Shmatikov V: **Privacy-preserving Data Exploration in Genome-wide Association Studies.** *KDD '13, ACM, New York, NY, USA* 2013, 1079-1087.

doi:10.1186/1472-6947-15-S5-S1
Cite this article as: Lu *et al.*: Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption. *BMC Medical Informatics and Decision Making* 2015 **15**(Suppl 5): S1.