Zhang et al. BMC Medical Informatics and Decision Making 2015, 15(Suppl 5):S5
http://www.biomedcentral.com/1472-6947/15/S5/S5

BMC
Medical Informatics & Decision Making

PROCEEDINGS Open Access

# FORESEE: Fully Outsourced secuRe gEnome Study basEd on homomorphic Encryption

Yuchen Zhang[1,2†], Wenrui Dai[1,2*†], Xiaoqian Jiang[2], Hongkai Xiong[1], Shuang Wang[2]

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

## Abstract

**Background:** The increasing availability of genome data motivates massive research studies in personalized treatment and precision medicine. Public cloud services provide a flexible way to mitigate the storage and computation burden in conducting genome-wide association studies (GWAS). However, data privacy has been widely concerned when sharing the sensitive information in a cloud environment.

**Methods:** We presented a novel framework (FORESEE: Fully Outsourced secuRe gEnome Study basEd on homomorphic Encryption) to fully outsource GWAS (i.e., chi-square statistic computation) using homomorphic encryption. The proposed framework enables secure divisions over encrypted data. We introduced two division protocols (i.e., secure errorless division and secure approximation division) with a trade-off between complexity and accuracy in computing chi-square statistics.

**Results:** The proposed framework was evaluated for the task of chi-square statistic computation with two case-control datasets from the 2015 iDASH genome privacy protection challenge. Experimental results show that the performance of FORESEE can be significantly improved through algorithmic optimization and parallel computation. Remarkably, the secure approximation division provides significant performance gain, but without missing any significance SNPs in the chi-square association test using the aforementioned datasets.

**Conclusions:** Unlike many existing HME based studies, in which final results need to be computed by the data owner due to the lack of the secure division operation, the proposed FORESEE framework support complete outsourcing to the cloud and output the final encrypted chi-square statistics.

## Introduction

Owing to the community effort on big data, biomedical science moves focus towards data-driven methodologies [1], which rely on collecting, integrating and analyzing large scale data. For biomedical studies, especially the genome analysis, the required storage and computational capacities may easily exceed the available resources in a single institution. Recently, cloud computing [2] emerges as a flexible alternative to support cost-effective biomedical research with big data. Researchers can rely on a cloud environment to easily scale up their studies with large scale data. However, the adopt of cloud computing in biomedical studies also yields more and more concerns about the potential data privacy risk in comparison with the local computing environment. As genome data are extremely sensitive, the storage of raw genome in a cloud may increase the disclosure risk.

The recently announced NIH policy [3] allows NIH funded studies to utilize public clouds to facilitate data analysis. However, the researchers instead of the cloud providers are responsible for the data security and privacy. Many existing attacks [4-6] also demonstrate the vulnerability of de-identified genome data. Thus, it is important to protect the privacy of genome data [7-9]. The rapid improvements of the data protection techniques make it possible to perform certain computations over encrypted data [10,11] based on homomorphic encryption.

* Correspondence: wed004@ucsd.edu
† Contributed equally
[1]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Full list of author information is available at the end of the article

In [12], Gentry proposed the first fully homomorphic encryption scheme to enable both addition and multiplication operations over encrypted data. Brakerski *et al.* [13,14] improved homomorphic encryption scheme based on learning with errors (LWE). Lauter *et al.* [15] presented several secure statistical algorithms for genetic association studies based on homomorphic encryption. Besides, Togan *et al.* [16] studied the integer comparison problem over homomorphic encrypted data. Recently, Graepel *et al.* [17] and Naehrig *et al.* [18] also showed that certain machine learning algorithms can be implemented using HME. Wang *et al.* [24] proposed a novel homomorphic encryption based framework to securely computing on exact logistic regression. Cheon *et al.* [19] developed a protocol for HME-based edit distance calculation that employed the greedy algorithm to obtain the upper bound of exact edit distance. Zhang *et al.* [25] improved homomorphic edit distance computation by combining path-finding algorithm and integer comparison.

In this paper, we propose the FORESEE framework to achieve secured and fully outsourced chi-square statistics computation in a public cloud. We assume that the cloud faithfully follows the protocol but may be curious of information from the received data, which is the so-called semi-honest adversary model [20]. The proposed FORESEE framework enables secure division operation over the homomorphic encrypted data and allows the cloud to directly release the study results. To be concrete, the contribution of this paper is two-fold.

- We develop a secure errorless division protocol, where a one-to-one mapping function is constructed for the floating numbers in computation and the study results can be accurately decrypted with a lookup table.
- We present a secure approximation division protocol to balance the complexity and accuracy with well-designed secure integer division in secure computation. In implementation, binary tree product and group-based computation are adopted to reduce circuit depth and the number of homomorphic multiplications.

For validation, experimental results show that the proposed FORESEE framework can identify all the significant SNPs based on the chi-square statistics with a moderate complexity using multiple slots for parallel computation.

## Method

For clarity, in the rest of this paper, we use bold symbols to represent vector and matrix variables and normal symbols for scalar variables. Without specification, $\widehat{\Delta}$ is reserved for the encrypted version of variable or function $\Delta$ and $\log(\cdot)$ stands for the logarithm with base 2.

### Secure outsourcing GWAS

In this paper, we focus on the task of secure outsourcing GWAS in the 2015 iDASH challenge [21]. Given the genotypes from two groups over a number of single nucleotide polymorphisms (SNPs), we aim to securely calculate the chi-square statistics for the SNPs between the given case-control groups. The chi-square statistic) $\chi^2$ is used by chi-square test to statistically assess whether there is significant association between the genetic variants and disease status. Typically, $\chi^2$ is obtained by cumulating the normalized squared deviations between the observed and expected frequency distribution of alleles.

$$\chi^2 = \sum_i \sum_j \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \quad (1)$$

Here, $O_{i,j}$ and $E_{i,j}$ are the observed and expected allele counts for allele $j$, e.g. $j = 1$ for allele 'A' and $j = 2$ for allele 'a' in (see Table 1) from the case ($i = 1$) or control ($i = 2$) group, respectively.

Let us denote $N_1 = O_{1,1} + O_{1,2}$ and $N_2 = O_{2,1} + O_{2,2}$ the total number of alleles in the case and control groups, respectively. In general, $E_{i,j}$ is computed by $((O_{1,j} + O_{2,j}) \cdot N_i)/(N_1 + N_2)$ for $i = 1, 2$ and $j = 1, 2$. If we assume that the case-control groups have the same number of $n$ patients, we can obtain $N_1 = N_2 = 2n$. Thus, Equation (1) can be simplified by

$$\chi^2 = \frac{4n \cdot (O_{1,1} - O_{2,1})^2}{(O_{1,1} + O_{2,1})[4n - (O_{1,1} + O_{2,1})]} \quad (2)$$

Equation (2) indicates that, in addition to homomorphic additions and multiplications, the $\chi^2$ statistic computation over encrypted dataset requires one secure division for fully outsourced GWAS, which is not supported in many existing HME-based schemes [15,17,22]. For example, if the numerator and denominator in Equation (2) are released directly due to the lack of secure division operation, one can easily infer the underlying allele counts (i.e., $O_{1,1}$and $O_{2,1}$) by solving a system of equations. To address the problem, we propose the

**Table 1. Observed allele counts for SNP, where $O_{1,1}$ and $O_{1,2}$ are the number of alleles A and a in the case group, $O_{2,1}$ and $O_{2,1}$ are the corresponding counts in the control group, $N_1$ and $N_2$ are the total allele counts for the case and control group, respectively**

| SNP | A | a | Total |
|---|---|---|---|
| Case | $O_{1,1}$ | $O_{1,2}$ | $N_1 = O_{1,1} + O_{1,2}$ |
| Control | $O_{2,1}$ | $O_{2,2}$ | $N_2 = O_{2,1} + O_{2,2}$ |
| Total | $O_{1,1} + O_{2,1}$ | $O_{1,2} + O_{2,2}$ | $N_1 + N_{12}$ |

FORESEE framework to enable secure division operation for the $\chi^2$ statistic computation on an untrusted cloud.

### The proposed framework

Figure 1 illustrates the proposed FORESEE framework, which allows secured and fully outsourced chi-square statistics computation in a public cloud and enable flexible release of study results. Using homomorphic encryption, the data owner can encrypt observed allele counts and directly upload to the public cloud. Consequently, the chi-square statistics can be securely computed according to Equation (2) based on homomorphic computation. Contrary to many existing HME-based schemes [15,17,22], the proposed framework develops two protocols for secure division operations over encrypted data, so that the final results are not necessarily computed by the data owner. As a result, authorized users are able to access the encrypted study results when granted the private key for decryption. Remarkably, the secrecy of uploaded sensitive information and released study results can be guaranteed under the proposed framework, as the trusted party would not interact with the untrusted public cloud. Thus, the proposed scheme enables secure outsourcing of the chi-square statistic computation to public cloud services, by which individuals or single institutions could contribute to the chi-square statistic computation in GWAS in a secure manner.

In the FORESEE framework, we develop two protocols for secure division operations, namely, secure errorless division and secure approximation division. The secure errorless protocol makes a secure one-to-one mapping from floating numbers to a set of encrypted positive integers. Consequently, authorized users can decrypt the study results with a lookup table. To achieve errorless division, the proposed protocol requires a deep circuit.

To balance the accuracy and complexity in chi-square statistic computation, the secure approximation division protocol is proposed as an alternative solution. Using secure integer division, the protocol approximates the study results with a tunable error rate. To improve its efficiency, binary tree product and group-based computation are designed to reduce circuit depth and the number of homomorphic multiplications.

In the following subsections, we will elaborate both protocols developed for the FORESEE framework.

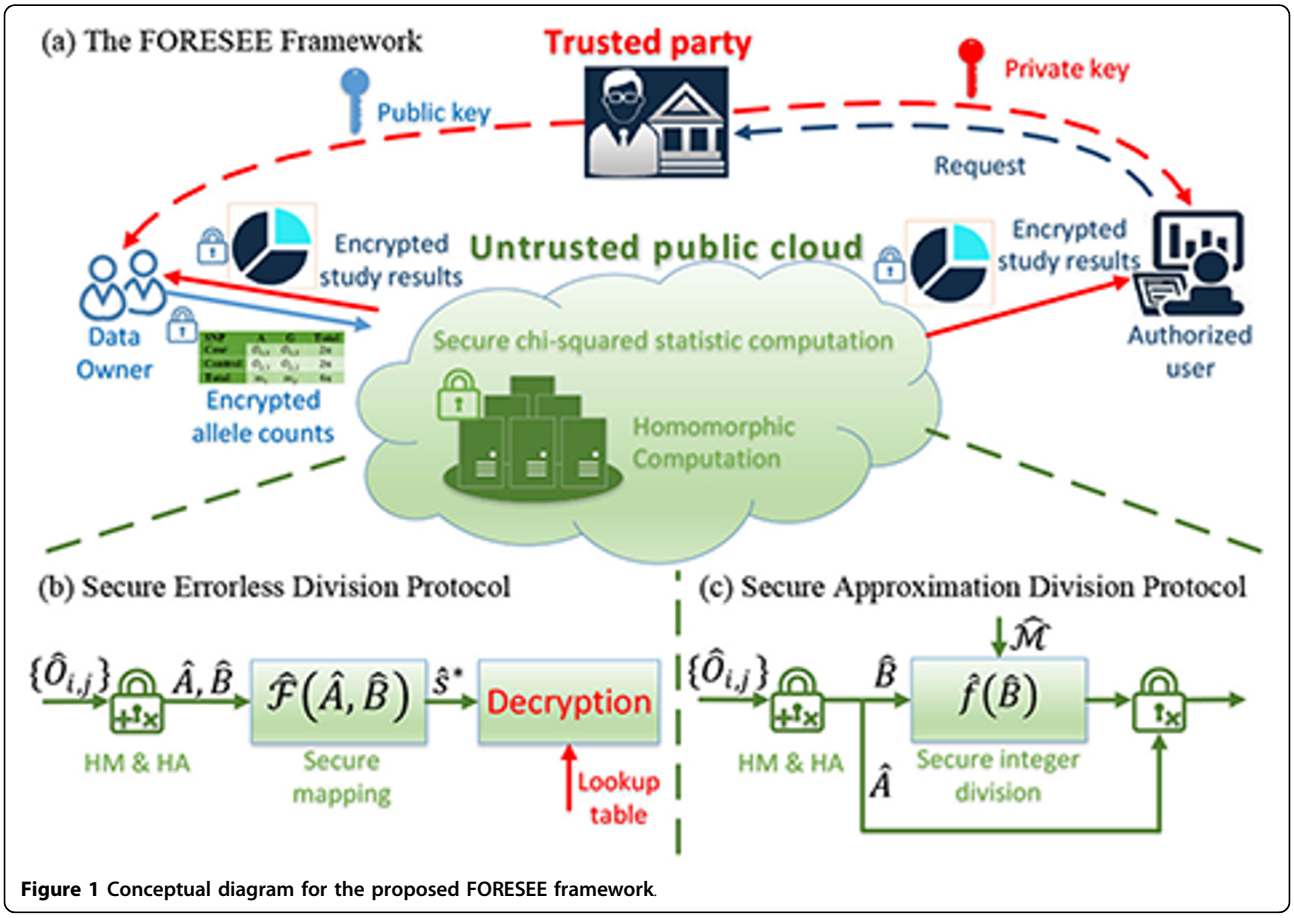


**Figure 1** Conceptual diagram for the proposed FORESEE framework.

### Secure errorless division protocol

In this section, we propose the secure errorless division protocol when both dividend and divisor are small (e.g., less than 100). Considering that secure division operation is not available in existing HME-based schemes [15,17,22], we construct a one-to-one mapping function from floating numbers to a set of encrypted positive integers. Thus, the study results can be accurately decrypted with a lookup table corresponding to the one-on-one mapping function.

#### *Secure mapping for division outcomes*

To map the study result (in floating numbers), we construct a function with an integer output that uniquely corresponds to the division outcomes given a dividend and divisor. Let us denote $m \in [0, \bar{m}]$ and $w \in [1, \bar{w}]$ the dividend and divisor, respectively. Here, the upper bounds $\bar{m}$ and $\bar{w}$ of $m$ and $w$ should be predefined, so that the lookup table for decryption can be synchronized for all the authorized users. Consequently, we construct a two-dimensional function $\mathcal{F}(m, w)$ that returns the positive integer $u_{m,w}$ corresponding to an index of the division result of $m/w$ in floating number.

$$\mathcal{F}(m, w) = u_{m,w} \tag{3}$$

In the ciphertext domain, $u_{m,w}$ can be determined by the polynomials of $m$ and $w$ related to the ciphertext modulus $p$. According to the Fermat Theory, we can construct a simplified function with less number of homomorphic multiplications. Given the prime $p > mw$, the secure mapping function is

$$\hat{\mathcal{F}}(\hat{m}, \hat{w}) \equiv \hat{m}\hat{w}^{p-2} (\mathrm{mod} p). \tag{4}$$

In Proposition 1, we demonstrated that the secure mapping proposed in Equation (4) is a one-to-one mapping from floating outcomes of $m/w$ to a set of encrypted positive integers.

**Proposition 1** *Given arbitrary positive integers* $m_1$, $m_2$, $w_1$, *and* $w_2$ *taking their values* $[1, \lfloor\sqrt{p}\rfloor]$, *they satisfy*

$$\frac{m_1}{w_1} = \frac{m_2}{w_2}, \tag{5}$$

*if and only if* $\mathcal{F}(m_1, w_1) \equiv \mathcal{F}(m_2, w_2) (\mathrm{mod} p)$, *where* $\lfloor\sqrt{p}\rfloor$ *is the round function that returns the maximum integer not greater than* $\sqrt{p}$.

*Proof.* Please refer to Appendix I.

Proposition 1 implies that Equation (4) can map any pairs of $(\hat{m}, \hat{w})$ with the same irreducible fraction to the same outcome $\hat{\mathcal{F}}(\hat{m}*, \hat{w}*)$, where $m^*$ and $w^*$ are the integer numerator and denominator, respectively that have no other common divisors. For example, given the ciphertext modulus $p = 101$, $\hat{\mathcal{F}}(\hat{2}, \hat{1})$ would be $\hat{2}$ for the pairs $(\hat{2}, \hat{1}), (\hat{4}, \hat{2})$ and $(\hat{8}, \hat{4})$. This fact means that the encrypted outcome can be securely released, as the authorized users can only obtain the accurate irreducible fraction $m^*/w^*$, but cannot infer the exact value of $(\hat{m}, \hat{w})$.

**Algorithm 1: Secure errorless division**

0: **Inputs:** encrypted variable $\hat{m}, \hat{w}$, upper bound $\bar{m}, \bar{w}$, the ciphertext modulus $p$.
1: Let $\hat{s}_0^* = \hat{w}, \hat{s}^* = \hat{m}$.
2: Let $u* = \lfloor \log(p-2) \rfloor$
3: Decompose $p - 2$ as $p - 2 = \sum_{i=0}^{k} 2^{v_i}$, where $k$ is the number of nonzero bits in the binary representation of $p - 2$ and $v_i$ is the position of i-th nonzero bit.
4: **For** each $i$ = 1,2,..., $u^*$
5: $\hat{s}_i^* = \hat{s}_{i-1}^* * \hat{s}_{i-1}^*$
6: **end for**
7: **For** each $i$ = 0,1,...,$k$
8: $\hat{s}^* = \hat{s}^* * \hat{s}_{v_i}^*$
9: **end for**
10: **Outputs**: $\hat{s}^*$

During decryption, users can find the accurate study result $r$ with a lookup table, which consists of all possible irreducible fractions within ranges $m \in [0, \bar{m}]$ and $w \in [1, \bar{w}]$. Here, we provide two examples, where $\bar{m} = \bar{w} = 10$ and $p$ is set to 101 as the smallest prime greater than $\bar{m}\bar{w} = 100$. It is worth mentioning that we can obtain the study result $r$ in floating number in Example 2. This fact verifies the accuracy of the proposed secure errorless division.

**Example 1** The authorized users would obtain $s^* = 2$ by decrypting $\hat{s}^* = \hat{2}$. The pair of co-prime integers ($m$, $w$) corresponding to $\hat{\mathcal{F}}(\hat{m}, \hat{w}) \equiv \hat{2} \pmod{101}$ is (2,1). Thus, $r = m/w = 2$.

**Example 2** When $\hat{s}^* = \widehat{35}, (m, w) = (4, 3)$ as $\hat{\mathcal{F}}(\hat{m}, \hat{w}) = \hat{4} \cdot \hat{3}^{99} \equiv \widehat{35} (\mathrm{mod} 101)$. As a result, $r = 4/3$.

### Secure approximation division protocol

In this subsection, we aim to develop the secure approximation division protocol. Since $n$ (i.e., the number of patients in case or control group) is assumed to be a known integer, we denote $A$ and $B$ the dividend and divider of $\frac{(O_{1,1} - O_{2,1})^2}{(O_{1,1} + O_{2,1})[4n - (O_{1,1} + O_{2,1})]}$ in Equation (2), respectively. Thus, the chi-square statistic can be rewritten as

$$\chi^2 = 4n * A * (\frac{1}{B}) \tag{6}$$

where $A = (O_{1,1} - O_{2,1})^2$ is a nonnegative integer, and $B = (O_{1,1} + O_{2,1})[4n - (O_{1,1} + O_{2,1})]$ is a positive integer. Thus, given encrypted counts $\hat{O}_{1,1}$ and $\hat{O}_{2,1}$, $\hat{A}$ and $\hat{B}$ can be obtained with homomorphic multiplications

and additions. Since the fraction team $1/B$, with the value less than one, cannot be evaluated in the ciphertext domain, we scale it up by multiplying a positive integer $\mathcal{M}$. Therefore, the $\chi^2$ statistic can be approximated by

$$\chi^2 = \frac{4n \cdot decrypt\left(\hat{A}\left\lfloor \frac{\widehat{M}}{\hat{B}} \right\rfloor\right)}{M}, \quad (7)$$

where $\lfloor M/B_i \rfloor$ is the round function that returns the maximum integer not greater than $M/B_i$, e.g., $\lfloor 7/3 \rfloor = 2$ and $\lfloor 10/15 \rfloor = 0$. Here, $\mathcal{M}$ is a public information and should be large enough, as the upper bound of relative error is determined by $1/\min\left(\frac{\mathcal{M}}{B}\right) \times 100\% = \left(\frac{400n^2}{\mathcal{M}}\right)\%$. Usually, we set $\mathcal{M} = \min\left(p-1, \left\lfloor \frac{p-1}{\max\left(\frac{A}{B}\right)} \right\rfloor\right)$, where $p$ is the ciphertext modulus. According to Equation (7), we develop the secure approximation division protocol based on secure integer division.

***Secure integer division***

In this subsection, we describe the secure integer division protocol to achieve secure To compute $\left\lfloor \frac{\widehat{\mathcal{M}}}{B} \right\rfloor$ o in Equation (7), we first introduce a vector $\boldsymbol{T}$ with its 6-th element defined by

$$t_i = \left\lfloor \frac{\mathcal{M}}{B_i} \right\rfloor i \in [1, 2n] \quad (8)$$

where $B_i = i * (4n - i)$, $i = 1,2,...,2n$ are the possible values of $B$ in chi-square statistic computation (see equation (6)). Consequently, given $\mathcal{M}$, we define a function $f(x)$ which satisfies

$$f(B_i) = \left\lfloor \frac{\mathcal{M}}{B_i} \right\rfloor i \in [1, 2n] \quad (9)$$

In our implementation, a one-dimensional function is formulated using Lagrange interpolating polynomial with $x \in \{B_1,..., B_{2n}\}$

$$f(x) = \sum_{i=1}^{2n} t_i \frac{\prod_{1 \le l \le 2n, l \ne i}(x - B_l)}{\prod_{1 \le l \le 2n, l \ne i}(B_i - B_l)} \quad (10)$$

Since division is intractable for homomorphic encrypted data, we need to derive a surrogate function for Equation (10) that can be implemented based on homomorphic multiplications and additions. For simplicity, we denote $u_i$. the divisor for $x = B_i$. in Equation (10).

$$u_i = \prod_{1 \le l \le 2n, l \ne i} (B_i - B_l) \quad (11)$$

Consequently, we can construct a surrogate function for Equation (10) by numerically finding a set of integers $v_i$. with $1 \le v_i. \le p - 1$ for $1 \le i \le 2n$, that satisfy

$$u_i v_i \equiv 1 \pmod{p} \quad (12)$$

Here, $p$ is the cipheretext modulus (i.e., a prime under double-CRT representation in the BGV scheme). Thus, we demonstrate the existence of $\{v_i\}$ in Proposition 2 to guarantee the computational tractability of $f(x)$ in the ciphertext domain.

**Proposition 2** *For each $u_i = 1,2,...,2n$, given $p > \mathcal{M}$, at least one $v_i$ can be found to satisfy (12).*

*Proof.* Please refer to Appendix II.

Substituting $1/\Pi_{1 \le l \le 2n, l \ne i}(B_i - B_l)$ with $v_i$ in Equation (10), we can reformulate $f(x)$ with multiplications instead.

$$f(x) = \sum_{i=1}^{2n} [t_i v_i \prod_{1 \le i \le 2n, \atop l \ne i} (x - B_l)] \quad (13)$$

We transform Equation (13) into the combination of polynomials of $x$ by expanding the products and combining the coefficients.

$$f(x) = \sum_{i=0}^{2n-1} h'_i x^i \quad (14)$$

Here, $h'_i$ is the coefficient for the $i$-th order of $x$ (i.e., $x^i$.) after polynomial expansion, which includes $v_i B_l$ and $t_i$. In the ciphertext domain, we can construct the function $\hat{f}(\hat{x})$ for secure integer division $\left\lfloor \widehat{\mathcal{M}}/\hat{x} \right\rfloor$.

$$\hat{f}(\hat{x}) \equiv \sum_{i=0}^{2n-1} \hat{h}_i \hat{x}^i \pmod{p} \quad (15)$$

where $\hat{x} \in \{\hat{B}_1, \hat{B}_2, \ldots, \hat{B}_{2n}\}$ are finite positive encrypted integers, and $\hat{h}_i \in [\hat{0}, \hat{p} - \hat{1}]$ is obtained by encrypting $h_i \equiv h'_i \pmod{p}$. We set $h_i = 0$ with $i > 2n - 1$.

***Implementation optimization***

The secure integer division can be optimized to further reduce the cumulative circuit depths (CCD) and number of homomorphic multiplications (HMs). To achieve this goal, we adopt group-based computation and binary tree product to generate $\hat{f}(\hat{x})$ in implementation.

To reduce the number of HMs, a group-based computation is adopted to calculate $\hat{f}(\hat{x})$. The key idea of the proposed group-based optimization is to first compute a set of $\hat{h}_{c \cdot d+i}\hat{x}^i$ with $i \in [0, d]$, where $d$ is number of

elements in each group, and $c = 0,..., C$ is the group index with the total number of groups $C = \lfloor (2n-1)/d \rfloor + 1$. After grouping, we get the following equation with a reduced number of HMs.

$$\hat{f}(\hat{x}) \equiv \sum_{c=0}^{C-1} \left[ \hat{x}^{c \cdot d} \left( \sum_{i=0}^{d-1} \hat{h}_{c \cdot d+i} \hat{x}^i \right) \right] \pmod{p} \quad (16)$$

Algorithm 2 describes the generation of $\widehat{X} = \left(\hat{1}, \hat{x}, \ldots, \hat{x}^d\right)$ using binary tree product. The number of HMs and CCD required to calculate $\widehat{X}$ can be reduced to $d - 1$ and $\lfloor \log(d-1) \rfloor + 1$, respectively.

**Algorithm 2: Binary tree product for generating $\widehat{X}$**

0: **Inputs:** encrypted variable $\hat{x}$, the maximum power $d$
1: **For** $i = 2,3,..., d$
2: Let $l_1 = 2^{\lfloor \log(i-1) \rfloor}$.
3: Let $l_2 = i - l_1$.
4: $\hat{x}^i = \hat{x}^{l_1} \cdot \hat{x}^{l_2}$.
5: **end for**
6: **Outputs**: $\hat{X} = \left(\hat{1}, \hat{x}, \ldots, \hat{x}^d\right)$

An additional optimization can be applied in equation (16) by replacing the multiplication $\hat{h}_{c \cdot d+i} \hat{x}^i$ as the summation over a total number of $\hat{h}_{c \cdot d+i}$ additions of $\hat{x}^i$ to reduce the number of HMs.

Since the time cost of HMs is larger than HAs, we determine $d$ by minimizing the number of HMs. As shown in Table 9 the total number of HMs required for secure integer division is $2C + d - 3$. The number of groups $C$ and the number of elements in each group $d$ are selected to minimize the number of HMs $F(d) = d + 2\lfloor (2n-1)/d \rfloor - 3$. Given an integer $n, F(d) \approx d + \frac{2(2n-1)}{d} - 3$ can obtain its minimum $2\sqrt{4n-2} - 3$, only when $d = \sqrt{2(2n-1)}$. Since $d$ is an integer, it is estimated by $\left\lfloor \sqrt{2(2n-1)} \right\rfloor$ to minimize $F(d)$. Thus, $C$ can be estimated by $\lfloor (2n-1)/d \rfloor + 1$ accordingly. Using the optimal $d$ and $C$, secure integer division can be achieved based on the encrypted function $\hat{f}(\hat{x})$ in Equation (15). Algorithm 3 elaborates the secure integer division. In line 2, in order to obtain $\widehat{X'}$, the inputs of Algorithm 2 are set to $\hat{x}^d$ and $C - 1$, respectively.

**Algorithm 3: Secure integer division**

0: **Inputs:** encrypted variable $\hat{x}$, group size $d$, the number of groups $C$, the ciphertext modulus $p$, the polynomial parameters $h_i$, $i = 0,1,...,2n-1$
1: Compute $\hat{X} = (\hat{1}, \hat{x}, \ldots, \hat{x}^d)$ according to Algorithm 2
2: Compute $\hat{X} = \left(\hat{1}, \hat{x}^d, \ldots, \hat{x}^{(C-1)d}\right)$ according to Algorithm 2
3: **For** each $c = 0,1,..., C-1$
4: **For** each $i = 0,1,..., d-1$
5: Calculate $\hat{h}_{cd+i} \hat{x}^i$
6: **end for**
7: **end for**
8: Let $\hat{a} = \hat{0}$
9: **For** each $c = 0,1,..., C-1$
10: $\hat{a}' = \hat{0}$.
11: **For** each $i = 0,1,2,..., d-1$
12: Update $\hat{a}' = \hat{a}' + \hat{h}_{cd+i} \hat{x}^i$
13: **end for**
14: Update $\hat{a} = \hat{a} + \hat{a}' \hat{x}^{cd}$
15: **end for**
16: **Outputs**: $\hat{a} = \hat{f}(\hat{x})$..

#### Parallel computation using multiple slots

Since HME schemes with ciphertext space $\mathbb{Z}_q^{L_s}$ support single instruction multiple data (SIMD) with $L_s$ slots, we can use parallel computation to reduce the number of homomorphic multiplications (HMs) and homomorphic additions (HAs). Denote $\hat{a} = (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{L_s})$ and $\hat{b} = (\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_{L_s})$ the two encrypted ciphertexts with $L_s$ slots. SIMD is applicable to simultaneous computation of the addition $\hat{a} + \hat{b} = \left(\hat{a}_1 + \hat{b}_1, \hat{a}_2 + \hat{b}_2, \ldots, \hat{a}_{L_s} + \hat{b}_{L_s}\right)$ and $\hat{a} \cdot \hat{b} = \left(\hat{a}_1 \cdot \hat{b}_1, \hat{a}_2 \cdot \hat{b}_2, \ldots, \hat{a}_{L_s} \cdot \hat{b}_{L_s}\right)$ multiplication. In two ciphertexts, only two slots in the same position can operate with each other.

In Algorithm 1, multiple encrypted outputs can be calculated at the same time with parallel computation. When the result is returned back to the user, the user extracts the integer in each slot and search it in the lookup table. Noticeably, in the parallel computation, $m^u$ and $n^u$ should be selected as the upper bounds of all the dividends and divisors in the slots. Similarly, multiple slots can also be used in the secure approximation division protocol. The secure integer division developed in Algorithm 3 can be simultaneously conducted for $L_s$ pairs of inputs $(\hat{a}_i, \hat{b}_i), i = 1, 2, \ldots, L_s$ using multiple slots.

## Results

In this section, we evaluate the proposed FORESEE framework, which was implemented with HElib [23], one of the most efficient open-source HME libraries based on the LWE theory [13,14]. The evaluations were made on an Ubuntu 14.04 server with Intel Xeon CPU E5-2687W @ 3.10GHz and 256 GB memory. We present the performance in the terms of time and memory cost. First, we provide the results of secure errorless division on simulated data. Moreover, we provide the performance of chi-square statistics based on the secure approximation division.

### Simulation study

Table 2 elaborates the experimental setups for the secure errorless division protocol. Given the ciphertext modulus $p$, the upper bound $\bar{m}$ (i.e., dividend) and $\bar{w}$ (i.e., divisor) is set to $\lfloor\sqrt{p}\rfloor$. A number of $L_s$ slots are used for parallel computation. The lifting parameter for plaintext base is set to 1. The security level is 80. The number of columns in key switching is 2. Hamming distance is 64.

Using HElib, one ciphertext can contain multiple slots to have many integers encrypted into the ciphertext with the public key. Thus, the size of the public key is related to the number of multiple slots $L_s$ in addition to the ciphertext modulus $p$ and the number of levels in modulus chain $L$. Taking Table 2 for example, $L_s$ for $(\bar{m}, \bar{w}) = (70, 70)$ is 3144, which is greater than most ones. Thus, its ciphertext sizes are much larger than the other configurations with close values of $\bar{m}$ and $\bar{w}$.

Using HElib, we are able to evaluate all the slots in the ciphertext in parallel. Table 3 shows the average execution time for the secure errorless division protocol. Based on the lookup table generated for various parameters $(\bar{m}, \bar{w})$, the proposed protocol is efficient for secure division operation over $m \le 100$ and $w \le 100$. However, its circuit depths increase rapidly with the growth of $m$ and $w$, which limits its application for larger dividends and divisors.

### Chi-square statistic computation

We employ the secure approximation division protocol in secure chi-square statistic computation. Two datasets from iDASH genome privacy protection challenge are used for evaluation, which contain 311 SNPs and 610 SNPs, respectively in the case-control groups, each consisted of with 200 individuals,

In homomorphic encryption, the ciphertext modulus $p$ and the number of levels in modulus chain $L$ are set to 25600000039 and 51, respectively. The public and private key sizes are both around 2.6 GB. The lifting parameter for plaintext base is set to 1. The security level is 80. The number of columns in key switching is 2 and the Hamming distance is 64. To reduce computational complexity, we use $L_S$ = 864 slots in parallel computation. For secure integer division, $\mathcal{M}$ is 25600000000. f and a are accordingly set to 28 and 15 for 200 individuals in each group.

Table 4 provides the time cost for homomorphic evaluation of both datasets in chi-square statistics computation, including key generation, encryption, total and average execution time. Using multiple slots, the secure approximation division protocol can achieve the chi-square statistics computation in less than one second in average. Table 5 evaluates the accuracy of computation in terms of the mean-squared error (MSE) and maximum error between the exact and the approximate chi-square statistics, where the MSE are less than $5 \times 10^{-10}$. The evaluation on maximum error also supports the conclusion.

Remarkably, we also computed the $p$-value for each SNP based on the chi-square statistic and applied different $p$-value cutoffs as 0.05, 0.01, and 0.005. The secure approximation division protocol is demonstrated to find out all the significant SNPs for both datasets under different $p$ -value cutoffs. As a result, the proposed protocol provides a good tradeoff between accuracy and complexity for secure chi-square statistic computation.

**Table 2. Experimental setups for secure errorless divisio**

| $(\bar{m}, \bar{w})$ | $p$ | $L$ | $L_s$ | Public key size | Private key size |
|---|---|---|---|---|---|
| (30,30) | 907 | 23 | 678 | 0.67 GB | 0.68 GB |
| (40,40) | 1,601 | 26 | 1,309 | 1.00 GB | 1.00 GB |
| (50,50) | 2,503 | 29 | 276 | 0.50 GB | 0.51 GB |
| (60,60) | 3,607 | 30 | 270 | 0.67 GB | 0.68 GB |
| (70,70) | 4,903 | 31 | 3,144 | 3.30 GB | 3.30 GB |
| (80,80) | 6,421 | 31 | 342 | 0.81 GB | 0.82 GB |
| (90,90) | 8,101 | 31 | 309 | 0.81 GB | 0.82 GB |
| (100,100) | 10,007 | 33 | 5,952 | 3.40 GB | 3.40 GB |

The inputs are positive integers $m$ (i.e., dividend) and $w$ (i.e., divisor) with corresponding upper bounds $\bar{m}$ and $\bar{w}$. $p$ is plaintext base; $L$ is number of levels in modulus chain; and $L_s$ is number of slots for parallel computation. Moreover, the storage costs of key generation are also provided as reference.

**Table 3. Time cost in seconds for key generation, encryption, and errorless division computation using various parameters**

| $(\bar{m}, \bar{w})$ | Key generation | Encryption | Execution time | |
|---|---|---|---|---|
| | | | Total | Average |
| (30,30) | 44.8395 | 9.13476 | 9.13476 | 0.0135 |
| (40,40) | 64.4364 | 9.61973 | 9.61973 | 0.0074 |
| (50,50) | 76.5504 | 11.3389 | 11.3389 | 0.0411 |
| (60,60) | 73.7685 | 12.0161 | 12.0161 | 0.0445 |
| (70,70) | 117.846 | 13.9717 | 13.9717 | 0.0044 |
| (80,80) | 87.9779 | 4.14286 | 13.3369 | 0.0390 |
| (90,90) | 87.9528 | 4.10576 | 15.5242 | 0.0502 |
| (100,100) | 127.002 | 16.7352 | 16.7352 | 0.0028 |

The average execution time is obtained by averaging over all the slots used.

**Table 4. Time cost in seconds for key generation, encryption, and the computation of chiRsquare statistics using different parameters based on secure approximation division**

| # of SNPs | Key generation | Encryption | Execution time | |
|---|---|---|---|---|
| | | | Total | Average |
| 311 | 212.32 | 355.61 | 286.75 | 0.92 |
| 610 | 222.53 | 428.05 | 315.67 | 0.52 |

The average execution time is obtained by averaging over all the slots used.

**Table 5. Recall and precision with different *p*-value cutoffs in identifying significant SNPS on both datasets, where the mean-squared error (MSE) and maximum error between the exact and the approximate chi-square statistics**

| # of SNPs | *p*-value cutoff | # of significant SNPs | Precision | Recall | MSE ($\times 10^{-10}$) | Maximum error ($\times 10^{-6}$) |
|---|---|---|---|---|---|---|
| 311 | 0.05 | 24 | 1 | 1 | 3.21 | 5.5 |
| | 0.01 | 20 | 1 | 1 | | |
| | 0.005 | 20 | 1 | 1 | | |
| 610 | 0.05 | 56 | 1 | 1 | 4.07 | 6.0 |
| | 0.01 | 27 | 1 | 1 | | |
| | 0.005 | 23 | 1 | 1 | | |

Furthermore, we compare the two proposed protocols in the chi-squared statistics computation. For secure errorless division protocol, we use the same parameters list above, except that $L$ is set to 151 to guarantee the required circuit depth in implementation. Table 6 and 7 compare the computational complexity and storage cost for the two protocols, respectively. In Table 6 the secure errorless division protocol requires about 10, 20 and 5 times in complexity for key generation, encryption, and execution (computation), when compared with the secure approximation division protocol. Table 7 shows that the ciphertext key sizes for secure errorless division are about 8 times larger due to the greater $L$. These results imply that the secure approximation division protocol provides a good trade-off in terms of complexity and accuracy for chi-squared statistic computation.

## Discussions

In this section, we analyze the computational complexity of the proposed FORESEEE protocol and discuss its potential extension and its limitation.

### Complexity analysis

In this subsection, we make an analysis on the computational complexity of the secure errorless division and secure approximation division protocols. Cumulative circuit depth2 (CCD) and the numbers of homomorphic multiplications (HMs) are provided for both protocols in the FORESEE framework.

We begin with the complexity analysis for secure errorless division protocol (i.e., Algorithm 1). As shown in Table 8 the number of HMs to calculate $\hat{s}_i^*$ at each iteration in A1 line 5 is 1. The number of HMs to obtain $s^*$ at each iteration in A1 lines 8 is also 1. Therefore, the CCD in calculating $\hat{s}_i^*$ in A1 lines 4-6 are $\lfloor \log(p-2) \rfloor$. The depths to obtain $s^*$ in A1 lines 7-9 are $\lfloor \log(p-2) \rfloor + h + 1$

Table 9 provides the CCD and number of HMs for secure approximate division (i.e., Algorithm 3). The number of HMs to obtain $\hat{X}$ and $\hat{X}'$ are $d - 1$ and $C - 2$, respectively. To evaluate Equation (18), the total number of HMs are $d + 2C - 3$. By using binary tree product based optimization, the circuit depths required for computing $\hat{X}$ and $\hat{X}'$ are $\lfloor \log(d-1) \rfloor + 1$ and $\lfloor \log(c-2) \rfloor + 1$,respectively. Finally, the total CCD for secure approximate division operation is $\lfloor \log(c-2) \rfloor + \lfloor \log(d-1) \rfloor + 3$.

**Table 6. Time cost in seconds for key generation, encryption and the computation of chi8squared statistics using different parameters based on secure errorless division (ED) and secure approximation division (AD) protocols**

| # of SNPs | Key generation | | Encryption | | Execution time | |
|---|---|---|---|---|---|---|
| | ED | AD | ED | AD | ED | AD |
| 311 | 2206 | **212.3** | 9575 | **355.6** | 1900 | **287** |
| 610 | 2131 | **222.5** | 9026 | **428.1** | 1660 | **316** |

### Potential extension

In this paper, we proposed the FORESEE framework to address the problem of fully outsourcing chi-square statistic computation to a public cloud. However, the application scenarios for the FORESEE framework, especially the secure approximation division protocol, can be further extended to securely compute other statistics tests that involve division operations. One intuitive example is the Transmission disequilibrium test (TDT) developed to assess the genetic linkage between the genetic variants and disease status in family-based association studies. TDT is based on the binomial test with one degree of freedom, which is asymptotically equivalent to the chi-square hypothesis test.

### Limitation

There are several limitations in the FORESEE framework. First, in secure approximation division protocol,

**Table 7 *L* (i**

| # of SNPs | *L* | | Public key size | | Private key size | |
|---|---|---|---|---|---|---|
| | ED | AD | ED | AD | ED | AD |
| 311 | 151 | **51** | 23.3GB | **2.6GB** | 23.6GB | **2.6GB** |
| 610 | | | | | | |

**Table 8. Complexity analysis in terms of cumulative circuit depth2 (CCD) and the number of homomorphic multiplications (HMs) for secure errorless division protocol (Algorithm 1)**

| Algorithm 1 | | CCD | # of HMs |
|---|---|---|---|
| 1: Let $\hat{s}_0* = \hat{w}, \hat{s}* = \hat{m}$. | | | |
| 2: Let $u^* = \lfloor \log(p-2) \rfloor$ | | | |
| 3: Decompose $p-2 = \sum_{i=0}^{\mathcal{k}} 2^{v_i}$ | – | | – |
| 4:**For** each $i$ = 1,2, ..., $u^*$ | | | |
| 5: $\hat{s}_i^* = \hat{s}_{i-1}^* * \hat{s}_{i-1}^*$ | | $\lfloor \log(p-2) \rfloor$ | 1 |
| 6: **end for** | | | |
| 7: **For** each i = 0,1, ..., $\mathcal{k}$ | | | |
| 8: $\hat{s}^* = \hat{s}^* * \hat{s}_{v_i}^*$ | | $1 + \mathcal{k} + \lfloor \log(p-2) \rfloor$ | 1 |
| 9: **end for** | | | |
| | **Total:** | $1 + \mathcal{k} + \lfloor \log(p-2) \rfloor$ | $1 + \mathcal{k} + \lfloor \log(p-2) \rfloor$ |

the upper bound of approximation error depends on the ciphertext modulus G. Therefore, G should be large enough to guarantee the accuracy in computation, which degrades the efficiency of the FORESEE framework. Second, the computational and storage costs based on homomorphic encryption are still very high. For example, key generation and encryption is much more timeconsuming than computation. The ciphertext sizes are also a heavy burden for communication. Finally, it is still a challenging problem to generalize the secure errorless division protocol. In summary, there is still room to improve the proposed division protocols in the FORESEE framework through better algorithm design, efficient coding in the HElib and parallelization.

## Conclusion

In this paper, we proposed a novel FORESEE framework for the secure outsourcing GWAS in the iDASH genome privacy protect challenge, especially for the chi-square statistic computation. The proposed framework consists of two protocols for secure division operation, namely secure errorless division and secure approximation division. The secure errorless protocol made a bijection between floating numbers and a set of encrypted positive

**Table 9. Complexity analysis in terms of cumulative circuit depth2 (CCD) and the number of homomorphic multiplications (HMs) for secure approximation division (Algorithm 3)**

| Algorithm 3 | | CCD | # of HMs |
|---|---|---|---|
| 1: Compute $\hat{X}$ | | $\lfloor \log(d-1) \rfloor + 1$ | $d-1$ |
| 2: Compute $\hat{X}'$ | | $\lfloor \log(C-2) \rfloor + \lfloor \log(d-1) \rfloor + 2$ | $C-2$ |
| 3:**For** $c$ = 0,1, ..., $C-1$ | | | |
| 4: **For** $i$ = 0,1, ..., $d-1$ | | | |
| 5: Calculate $\hat{h}_{cd+i}\hat{x}^i$ | – | | – |
| 6: **end for** | | | |
| 7: **end for** | | | |
| **8:** $\hat{a} = \hat{0}$ | | | |
| 9: **For** $c$ = 0,1, ..., $C-1$ | | | |
| 10: $\hat{a}' = \hat{0}$ | – | | – |
| 11: **For** $i$ = 0,1, ..., $d-1$ | | | |
| 12: $\hat{a}' = \hat{a}' + \hat{h}_{cd+i}\hat{x}^i$ | – | | – |
| 13: **end for** | | | |
| 14: $\hat{a} = \hat{a} + \hat{a}'\hat{x}^{cd}$ | | $\lfloor \log(C-2) \rfloor + \lfloor \log(d-1) \rfloor + 3$ | 1 |
| 15: **end for** | | | |
| | **Total:** | $\lfloor \log(C-2) \rfloor + \lfloor \log(d-1) \rfloor + 3$ | $2C + d - 3$ |

integers. Thus, it could output the accurate study results based on a lookup table. On the other hand, the secure approximation division protocol adopted secure integer division to obtain approximate study results with a tunable accuracy. The protocol was able to balance the complexity and accuracy by using the group-based computation and binary tree product with improved efficiency. In comparison to existing HME-based schemes [15,17,22], both protocols enabled fully outsourced secure GWAS in an untrusted public cloud and could directly release study results to authorized users for decryption. Experimental results show that the secure approximation division protocol can capture all the significant SNPs in chi-square statistic computation with a moderate computational complexity.

## Appendix I: Proof of Proposition 1

Since $m_1 w_1^{p-2} \equiv m_2 w_2^{p-2} (\mathrm{mod}\ p)$, we can obtain Equation (17) by multiplying $w_1 w_2$ on the both sides.

$$w_2 m_1 w_1^{p-2} \equiv w_1 m_2 w_2^{p-2} (\mathrm{mod}\ p) \quad (17)$$

According to the Fermat's little theorem,

$$w_i^{p-1} \equiv 1 (\mathrm{mod}\ p) \quad i = 1, 2. \quad (18)$$

When $w_1$ and $w_2$ are coprime with $p$, we can find that

$$w_2 m_1 \equiv w_1 m (\mathrm{mod}\ p) \quad (19)$$

Since $w_2 m_1 \leq \lfloor \sqrt{p} \rfloor * \lfloor \sqrt{p} \rfloor < p$ and $w_1 m_2 \leq \lfloor \sqrt{p} \rfloor * \lfloor \sqrt{p} \rfloor < p$, it holds for the prime $p$ that

$$-p < w_2 m_1 - w_1 m_2 < p \quad (20)$$

From Equations (19) and (20), we obtain that $w_2 m_1 = w_1 m_2$, which comes to Proposition 1.

## Appendix II: Proof of Proposition 2

We recall the Fermat's little theorem that, given a prime $p$,

$$q^{p-1} \equiv 1 (\mathrm{mod}\ p) \quad (21)$$

where $p$ and $q$ are coprime numbers. Since $u_i \in [1, p-1]$, the greatest common divisor for $u_i$ and $p$ is always 1. Given an integer $u_i$ , we can derive $v'_i = u_i^{p-2}$ to satisfy $u_i v'_i \equiv 1 \pmod{p}$ in Equation (12). Thus, by considering $v_i \equiv v'_i \pmod{p}$, $v_i. \in [1, p-1]$ can be found for Equation (12). As a result, we draw the Proposition 2.

### Abbreviations

GWAS: GenomeRwide association study; HME: Homomorphic Encryption; HM: Homomorphic Multiplication; CCD: Cumulative circuit depth.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

YZ and WD drafted the majority of the manuscript, YZ conducted the experiments. HX and XJ provided some helpful comments. SW provided the motivation for this work, detailed edits and critical suggestions.

### Acknowledgements

This work was funded in part by the NHGRI (K99HG008175), NLM (R00LM011392, R21LM012060), NHLBI (U54HL108460), NSFC (61425011, 61271218, 61501294 and U1201255), and "Shu Guang" project (13SG13). This article has been published as part of *BMC Medical Informatics and Decision Making* Volume 15 Supplement 5, 2015: Proceedings in 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.

### Authors' details

[1]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. [2]Department of Biomedical Informatics, University of California, San Diego, La Jolla, CA 92093, USA.

Published: 21 December 2015

### References

1. Howe D, Costanzo M, Fey P, Gojobori T, Hannick L, Hide W, Hill DP, Kania R, Schaeffer M, St Pierre S, Twigger S, White O, Rhee SY: **Big data: The future of biocuration.** *Nature* 2008, **455**:47-50.
2. *The NIST Definition of Cloud Computing* National Institute of Standards and Technology.
3. **NOTLODL15L086: Notice for Use of Cloud Computing Services for Storage and Analysis of ControlledLAccess Data Subject to the NIH Genomic Data Sharing (GDS) Policy.** [http://grants.nih.gov/grants/guide/noticeRfiles/NOTRODR15-086.html].
4. Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, Pearson JV, Stephan DA, Nelson SF, Craig DW: **Resolving individuals contributing trace amounts of DNA to highly complex mixtures using highLdensity SNP genotyping microarrays.** *PLoS Genet* 2008, **4**:e1000167.
5. Gymrek M, McGuire AL, Golan D, Halperin E, Erlich Y: **Identifying personal genomes by surname inference.** *Science (80-)* 2013, **339**:321-324.
6. Wang R, Li YF, Wang X, Tang H, Zhou X: **Learning your identity and disease from research papers.** *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09* New York, New York, USA: ACM Press; 2009, 534-44.
7. Naveed M, Ayday E, Clayton EW, Fellay J, Gunter CA, Hubaux JRP, Malin BA, Wang X: *Privacy and Security in the Genomic Era* 2014.
8. Wang S, Mohammed N, Chen R: **Differentially private genome data dissemination through topLdown specialization.** *BMC Med Inform Decis Mak* 2014, **14(Suppl 1)**:S2.
9. Kamm L, Bogdanov D, Laur S, Vilo J: **A new way to protect privacy in largeLscale genomeLwide association studies.** *Bioinformatics* 2013, **29**:886-93.
10. Zhou M, Zhang R, Xie W, Qian W, Zhou A: **Security and Privacy in Cloud Computing: A Survey.** *2010 Sixth International Conference on Semantics, Knowledge and Grids. IEEE* 2010, 105-112.
11. Wang W, Hu Y, Chen L: **Accelerating fully homomorphic encryption using GPU.** *IEEE Conference on High Performance Extreme Computing (HPEC)* 2012, 1-5.
12. Gentry C: **Fully homomorphic encryption using ideal lattices.** *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09* New York, NY, USA: ACM Press; 2009, 169-178.
13. Brakerski Z, Gentry C, Vaikuntanathan V: **(Leveled) fully homomorphic encryption without bootstrapping.** In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12. Volume 111.* New York, NY, USA: ACM Press; 2012:309-325.
14. Brakerski Z, Vaikuntanathan V: **Efficient fully homomorphic encryption from (standard) LWE.** *SIAM J Comput* 2011, **43**:831-871.
15. Lauter K, LópezRAlt A, Naehrig M: **Private computation on encrypted genomic data.** *14th Privacy Enhancing Technologies Symposium, Workshop on Genome Privacy (GenoPri'14)* Amsterdam, The Netherlands; 2014.

16. Togan M, Plesca C: **ComparisonLbased computations over fully homomorphic encrypted data.** *Communications (COMM), 2014 10th International Conference on* 2014, 1-6.
17. Graepel T, Lauter K, Naehrig M: **ML confidential: Machine learning on encrypted data.** *Information Security and Cryptology ICISC 2012* Springer; 2013, 1-21.
18. Naehrig M, Lauter K, Vaikuntanathan V: **Can homomorphic encryption be practical?** *Proceedings of the 3rd ACM workshop on Cloud computing security workshop - CCSW '11* New York, NY, USA: ACM Press; 2011, 113.
19. Cheon JH, Kim M, Lauter K: **Homomorphic Computation of Edit Distance.** *WAHC'15 - 3rd Workshop on Encrypted Computing and Applied Homomorphic Cryptography* 2015.
20. Hazay C, Lindell Y: **Efficient Secure Two-Party Protocols.** Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, Information Security and Cryptography.
21. **2015 iDASH Privacy and security Workshop.** [http://www.humangenomeprivacy.org/2015/].
22. Bos JW, Lauter K, Naehrig M: **Private predictive analysis on encrypted medical data.** *J Biomed Inform* 2014, **50**:234-243.
23. [https://github.com/shaih/HElib].
24. Wang S, Zhang Y, Dai W, Lauter K, Kim M, Tang Y, Xiong H, Jiang X: **HEALER: Homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS.** *Bioinformatics* 2015, [accepted].
25. Zhang Y, Dai W, Wang S, Kim M, Lauter K, Sakuma J, Xiong H, Jiang X: **SECRET: Secure EditLdistance computation over homomoRphic Encrypted daTa.** *Proceedings of the 5th Annual Translational Bioinformatics Conference Tokyo, Japan* 2015, [accepted].

doi:10.1186/1472-6947-15-S5-S5
**Cite this article as:** Zhang *et al.*: FORESEE: Fully Outsourced secuRe gEnome Study basEd on homomorphic Encryption. *BMC Medical Informatics and Decision Making* 2015 **15**(Suppl 5):S5.