Privacy-Safe Linkage Analysis with Homomorphic Encryption

Chibuike Ugwuoke, Zekeriya Erkin and Reginald L. Lagendijk
Cyber Security Group
Department of Intelligent Systems
Delft University of Technology
Mekelweg 4, 2628 CD, Delft, The Netherlands
Email: {c.i.ugwuoke, z.erkin and r.l.lagendijk}@tudelft.nl

Abstract—**Genetic data are important dataset utilised in genetic epidemiology to investigate biologically coded information within the human genome. Enormous research has been delved into in recent years in order to fully sequence and understand the genome. Personalised medicine, patient response to treatments and relationships between specific genes and certain characteristics such as phenotypes and diseases, are positive impacts of studying the genome, just to mention a few. The sensitivity, longevity and non-modifiable nature of genetic data make it even more interesting, consequently, the security and privacy for the storage and processing of genomic data beg for attention. A common activity carried out by geneticists is the association analysis between allele-allele, or even a genetic locus and a disease. We demonstrate the use of cryptographic techniques such as homomorphic encryption schemes and multiparty computations, how such analysis can be carried out in a privacy friendly manner. We compute a 3×3 contingency table, and then, genome analyses algorithms such as linkage disequilibrium (LD) measures, all on the encrypted domain. Our computation guarantees privacy of the genome data under our security settings, and provides up to 98.4% improvement, compared to an existing solution.**

I. Introduction

In recent years, it has become possible to perform whole human genome sequencing, which was not an easy feat only a few decades ago [1], [2]. Geneticist and researchers are now depending on the availability of the human genome in digital form to conduct ground breaking research. The genome contains rich information about the direct owner, relatives and even species, and most of these information are yet to be properly understood by scientist [1], [3]. Therefore, it is often the case that the genome is investigated to obtain various types of relationships. These relationships may include paternity relationships, possible human emigrations hundreds of years ago, gene-phenotype relationships, gene-disease relationships, patients response to a particular medication. Investigating relationships as listed above have numerous advantages which would include better understanding of human genome and possibly provide for better preventive and personalised healthcare [1], [4], [2]. However, the genome is a very sensitive data, which contains lots of other information about the owners, thereby posing a privacy threat to those who provide their genomes for various scientific or medical activities [4], [2].

When analysing gene-disease relationship, two conditions are feasible. First, a genetic marker could have a direct effect on a disease, thereby said to have a causal relationship type of association with the disease, and the marker doubles as the disease locus. Alternatively, a disease locus could be in linkage disequilibrium with a genetic marker, hence an indirect gene-disease association. In the latter type of association (which is our mode of interest in this paper), the genetic marker is not the same as the disease locus, and scientist often perform computation of statistical measures to ascertain the degree of LD, being, the threshold to confirm the suspicion of an association. It is not often the case that a single gene is responsible for a specific trait (disease, phenotype) and a known way for investigating gene-disease association could be to compute LD measures between a known genetic marker and a suspected allele. However, when individuals donate their genome as sample for analyses, it is usually common to re-identify participants, creating a huge privacy-risk [5], [4]. The challenge then become, whether we can perform computational analyses on genome data, without compromising the privacy of the genome owners.

We consider a scenario where a processing entity with sufficient resources store and process genome data, and an authenticated researcher seeks to compute an operation over the data stored by the processing entity. The data resident with the processing entity may have been voluntarily contributed by individuals, or provided by some verified medical institution. The researcher is in need of computing an LD statistic measure over the rich dataset resident with the processing entity. Because of the privacy-sensitive nature of the genome, it has become necessary to adapt privacy-preserving measures while performing computations that require genome data. Proposed solutions have suggested obfuscating the genetic data using different approaches like statistical data anonymization techniques and secret sharing [5]. Other studies have also suggested access control and security of databanks as the only measure, but that does not protect the privacy of the participants. The studies [6], [7] recommend cryptographic solutions like homomorphic encryption (HE), to encrypt the data and homomorphically compute the LD statistic measures. Deploying encryption and related cryptographic techniques are preferred because it is easy to mathematically prove the security of cryptosystems and equally extend proofs to the constructed solutions, therefore one is able to estimate possible information leakage. In our work we propose a cryptographic solution, by encrypting the data and computing the relevant statistical analysis over the encrypted data. We treat the secure databank of genomic data as a secure signal sequence, which needs to be outsourced to an untrusted processing entity for computation analysis. The encrypted signals (genetic data) are sensitive and need to be processed in a manner that pays keen attention to privacy of the data. With the encrypted data being transmitted and processed by an entity who only has the computational resource but not able to learn the content of the data, it equally demonstrates that encrypted signals can be generated and processed without threat to privacy.

Our Contribution: We are proposing an efficient method for computing LD statistic measures over encrypted genome data. We adopt HE as a technique to securely store and privately compute LD measures from a genotype databank, owing to the provable security and privacy guarantees it provides. Furthermore, we introduce an honest-but-curious Key Manager for our solution, in order to improve efficiency of computing parameters and reduce storage costs by 83.3%. Also, we adapt the genotypic LD approach of computing the LD measures as against the allelic LD approach, because the allelic LD approach requires haplotype estimation techniques, which is computationally expensive and often bias [8], [9]. We adopt data packing technique to help manage the data expansion challenge that comes with encrypting the genes. Our encoded data storage and retrieval design makes it easier to dynamically compute the contingency table parameters necessary for computing the statistical measures, which shows a significant improvement from existing works [6], [7] that adopted homomorphic techniques.

Outline: In the rest of this paper, Section II discusses some related literature, while Section III contains preliminaries relevant to our work. In Section IV we propose our solution for privacy-friendly computation of LD measures. Section V contains security and performance analyses and finally, in Section VI we have conclusion.

II. Related Works

Prior to our work, different authors have proposed various techniques for addressing privacy concerns in genome data processing, ranging from differential privacy, secret sharing and homomorphic encryption [10]. Specifically, Wu and Haven [11] demonstrated a secure computing of statistical analysis algorithms over encrypted data. Their work demonstrates the use of leveled homomorphic encryption to compute the mean and covariance of a dataset, other than genome data.

More recently, Lauter et al. [6] conducted a study which demonstrates the application of homomorphic encryption in the analysis of genomic data. The study shows that statistical algorithms (Pearson Goodness-of-Fit Test, r^2-measures of LD, Estimation Maximization (EM) algorithm for haplotyping, etc) peculiar to genetic studies can be replicated over encrypted data. Their solution aims to protect privacy of participants whose genome data are used in the analysis. The study however provides a solution that incurs three times the storage cost, due to their choice of design that maps a single gene value to three homomorphic ciphertexts. Lauter et al. also implement the construction of the 3×3 contingency table in a computationally expensive method.

Lu et al. [7] propose a solution which allows for genomic data to be securely outsourced to a third party who should perform the analysis over the encrypted dataset. They utilised leveled homomorphic encryption [12] and present a result that outperformed Lauter's implementation [6]. They deployed data packing techniques thereby reducing the three ciphertext to one gene mapping that was suggested in [6]. Lu et al. describe their work for chi-square test using allele frequencies, which is obtained from genotype/phenotype values contributed to the processing party.

Other works include that of Shahbazi et al. [13], whose work presents secure computation of LD measures and Cochran Armitage Test for Trennd (CATT) using secret sharing. The adoption of secret sharing requires the use of multiple servers which is bounded by non-collusion assumption. A secret sharing solution allows for a faster computation but incurs more communication rounds and storage requirements.

III. Preliminaries

There are 4 major entities in our description, which are 1) Storage and Processing Entity (SPE), 2) Researcher (R), 3) Key Manager (KM) and 4) Encoder. We loosely refer to whoever is responsible for the encryption of the genotype as encoder, this could be a participant or a verified medical institution. The SPE is responsible for storing all encrypted genome and subsequently performs computation on the genome on behalf of an authenticated R who is interested in computing an LD statistic measure over the dataset. The KM is an honest-but-curious entity who is only responsible for key generation, distribution and secure decryption of final computation results. Also, individuals whose genomes are available with the SPE are also called participants. Therefore, a participant's record is interchangeable with a sample.

A. Linkage Disequilibrium measures

We shall consider two hypothetical genetic markers X and Y, with each marker having two alleles of the same gene. Marker X has the alleles A and a while marker Y has the alleles B and b. Linkage Disequilibrium is said to exist when two or more alleles at different loci are observed to often be inherited together in a non random manner [9]. Statistical LD measures such as Pearson's correlation, Lewontin's D, linear regression are computable given counts of genotypes.

There are two ways of measuring LD statistics, *allelic* or *genotype-based*. *Allelic* LD measures require haplotype estimation techniques which is not trivial, but *genotype-based* approach allows computation without haplotype estimation. In our work, we have only the genotype data, hence the adoption of *genotype-based* approach for computing LD statistics measures. We model computations for: 1) The digenic LD between two markers X and Y, represented as $\mathcal{D}_{XY}$. 2) The Pearson's correlation coefficient, represented as Δ_{XY} . Table I is a 3×3 contingency table, which shows an example of genotype counts gen_{ij}, and their marginal sums. n_i , m_j represent the sum of all values in row i , and column j respectively, $i, j \in \{0, 1, 2\}$. For instance, $n_0 = a + b + c$, $gen_{00} = a$, also, $Q = n_0 + n_1 + n_2 = m_0 + m_1 + m_2$, which is the total number of participants. The contingency table shall form the basis of the rest of our computation.

Given that $P(\cdot)$ represents frequency and p_A is the frequency of the allele A, $\mathcal{D}_{XY}$ is computed as [9]:

$$\mathcal{D}_{XY} = 2P\left(\begin{array}{c|c} A & A \\ B & B \end{array}\right) + P\left(\begin{array}{c|c} A & A \\ B & b \end{array}\right) + P\left(\begin{array}{c|c} A & a \\ B & B \end{array}\right) + \frac{1}{2}\left(P\left(\begin{array}{c|c} A & a \\ B & b \end{array}\right) + P\left(\begin{array}{c|c} A & a \\ b & B \end{array}\right)\right) - 2p_A p_B \ , \quad (1)$$

and Eq. (1) can be estimated by,

$$\hat{\mathcal{D}}_{XY} = \frac{1}{Q}(a + b + d + \frac{1}{2}e) - 2\hat{p}_A\hat{p}_B \ , \quad (2)$$

$$\hat{p}_A = \frac{2n_0 + n_1}{2Q} \ , \ \hat{p}_B = \frac{2m_0 + m_1}{2Q} \ . \quad (3)$$

Let $\hat{gen}$ be estimated genotype count. Then,

$$s_{xy} = \left(\sum_{i=0}^{2}\sum_{j=0}^{2} ij \cdot \frac{\hat{gen}_{ij}}{Q}\right) - \bar{x}\bar{y} \ , \quad (4)$$

where,

$$\bar{x} = \sum_{i=0}^{2} i \cdot \frac{n_i}{Q} \ , \text{and } \bar{y} = \sum_{j=0}^{2} j \cdot \frac{m_j}{Q} \ , \quad (5)$$

and,

$$s_x^2 = \left(\sum_{i=0}^{2} i^2 \cdot \frac{n_i}{Q}\right) - \bar{x}^2 \ , \quad \text{and} \quad s_y^2 = \left(\sum_{j=0}^{2} j^2 \cdot \frac{m_j}{Q}\right) - \bar{y}^2 \ . \quad (6)$$

Given the above equations, the Pearson's correlation coefficient can be estimated as

$$\hat{\Delta}_{XY} = \frac{s_{xy}}{s_x \cdot s_y} \ . \quad (7)$$

TABLE I
GENOTYPE COUNTS AT TWO BI-ALLELIC MARKERS X AND Y

	BB	Bb	bb	$\sum$
AA	a	b	c	n_0
Aa	d	e	f	n_1
aa	g	h	i	n_2
$\sum$	m_0	m_1	m_2	Q

B. Homomorphic Encryption

Homomorhpic Encryption (HE) allows for a simple operation to be performed on ciphertexts, such that the resulting ciphertext would decrypt to the same value as would be obtained if the algebraic operation were to be performed on the plaintext values. Let $E_{pk}(\cdot)$ and $D_{sk}(\cdot)$ represent encryption and decryption functions respectively. m_1 and m_2 are two messages and k is a scalar value, while $\oplus$ and $\otimes$ are arbitrary operations on the ciphertexts. Then, homomorphism is defined as follows,

$$D_{sk}(E_{pk}(m_1) \oplus E_{pk}(m_2)) \ = \ m_1 + m_2 \ ,$$

$$D_{sk}(E_{pk}(m_1) \otimes k) \ = \ m_1 \times k \ .$$

We leverage on the additive and scalar multiplicative properties of the HE scheme described by Paillier [14] to compute the required statistical algorithm over an encrypted genome dataset. We refer readers to [14] for more detail about the cryptosystem.

Secure Multiplication Protocol (SMP): The homomorphic cryptosystem [14] of choice only offers additive homomorphism and no multiplicative homomorphism, we therefore initiate a secure two-party protocol in order to obtain the multiplication of two encrypted values [15]. Given two parties $Alice$ and Bob, $Alice$ holds two ciphertexts $E_{pk}(m_1)$ and $E_{pk}(m_2)$ and requires to compute the product $E_{pk}(m_1 \cdot m_2)$. Bob holds the secret key sk, $Alice$ picks randoms $r_1, \ r_2$, and encrypts, then computes $E_{pk}(m_1{+}r_1)$ and $E_{pk}(m_2{+}r_2)$, then sends the masked values to Bob. Bob decrypts and multiplies the results, then encrypts $E_{pk}(m_1{\cdot}m_2{+}m_1{\cdot}r_2{+}m_2{\cdot}r_1{+}r_1{\cdot}r_2)$ and sends to $Alice$. Finally, $Alice$ can unmask the value to obtain $E_{pk}(m_1 \cdot m_2)$.

IV. Privacy-Friendly Linkage Analysis

Given the $Encoder$, SPE, R and KM setting, our aim is to preserve the privacy of the participants whose encrypted data are with SPE, and from which an LD-measure is to be computed. R is also guaranteed to obtain a correct computation result from the analysis, but should not learn any identifying information from the result. Our solution is twofold, first we homomorphically construct the 3×3 contingency table, as presented in Table I, with each parameter computed as an independent ciphertext. Secondly, we use the parameters from the constructed table, as input to computing any LD statistic measure of choice. In the entirety of our protocol, there is a non-collusion assumption between any two entities.

The storage overhead of applying encryption is due to data expansion, therefore, we choose to efficiently constrain data expansion in our solution. For this reason, we introduce data packing technique for the encrypted values.

A. Genotype Encodings

To cushion the effect of data growth inherent in encrypting the genotype, the $Encoder$ performs the following one time operation to encode genotypes. Let N be plaintext size of the cryptosystem, recall that Q is the record size of the genotype counts from Table I.

Setup: The KM generates cryptographic keys $(pk, \ sk)$ and makes pk public, and keeps sk secret. Let κ be the security parameter, $\ell = \lfloor \frac{\log_2 N}{\log_2 Q + \kappa + 1} \rfloor$, where ℓ is the number of $slot$ that are contained in the plaintext size of N. $slot_\jmath$ represents the $\jmath^{th}$ slot in N and $\jmath \in \{0, \ldots, \ell - 1\}$. The $Encoder$ reserves the last 4 slots $\{slot_{\ell-4}, slot_{\ell-3}, slot_{\ell-2}, slot_{\ell-1}\}$ for indicating genotype intersections, this means there are only $\ell - 4$ slots reserved for genotypes. Let number of genes to be encoded in a single ciphertext be $\delta = \frac{\ell - 4}{3}$, which means that a single gene needs 3 consecutive slots, each of size $(\log_2 Q + \kappa + 1)$-bits. Let $\{X, Y \ , \cdots \ , Z\}$ be a set of markers.

Let t be the plaintext encoding, so that,

$$t = |X_0|X_1|X_2|Y_0|Y_1|Y_2|...|Z_0|Z_1|Z_2|a|b|d|e| \ . \tag{8}$$

For a given genotype database with maximum record threshold of Q, and each record $E_{pk}(t)$ being a ciphertext of δ-genes, SPE reconstructs a similar table as the model in Table I. For a genetic marker X, with dual alleles A/a and possible genotype values of $\{AA, Aa, aa\}$ with corresponding index $\{0,1,2\}$ respectively. The $Encoder$ allocates a triple-slot to the marker X, with the genotypes mapped to the corresponding slot index, i.e X_i for $i \in \{0,1,2\}$ as indicated in Eq. 8. For every gene in a sample encoding, and given the participant's genotypes, the value 1 is entered in the slot for every corresponding genotype expressed by the participant, and every other genotype slot is completed with value 0.

Step 1: For two genetic markers X, Y in the encoding that are of interest for computation, the $Encoder$ indicates corresponding intersection slots with value 1 for where an intersection for one of $\{a = AA/BB, b = AA/Bb, d = Aa/BB, e = Aa/Bb\}$ exists, and value 0 otherwise. $Encoder$ then sends $E_{pk}(t)$ to SPE.

Step 2: To reconstruct Table I from a database of T records, with each record $E_{pk}(t_j)$ for $0 \leq j < T$ representing encrypted genotypes modelled after Eq. 8. SPE computes, $E_{pk}(genSum) = \prod_{j=0}^{T-1} E_{pk}(t_j) = E_{pk}(\sum_{j=0}^{T-1} t_j)$, as a single ciphertext courtesy of the Paillier cryptosystem [14]. It can be observed that the summation allows for slots to be summed component-wise, without overflowing into a neighbouring slot, therefore providing a ciphertext that should decrypt to a plaintext which preserves the encoding in Eq. 8.

Step 3: In reference to Table I, SPE can obtain the following parameters $\{a, b, d, e, n_0, n_1, n_2, m_0, m_1, m_2\}$ as a packed ciphertext from the homomorphic addition result.

Secure Unpacking Protocol (SUP): An SUP requires that SPE initiates a secure two-party protocol with KM in order to unpack a single ciphertext of encoded sums of t's, into 10 independent ciphertexts of the parameters $\{a, b, d, e, n_0, n_1, n_2, m_0, m_1, m_2\}$, from which ciphertexts of the remaining variables can be obtained homomorphically. SPE chooses a cryptographic secure random number r of size N-bits, encrypts r and performs an additive masking of $genSum$ to obtain $E_{pk}(genSum + r)$, which is sent to KM for unpacking.

Step 4: KM decrypts the masked ciphertext to obtain a masked plaintext P, and splits P into ℓ parts, each of size $(\kappa + \log_2 Q + 1)$-bits, then encrypts each of $P_0, \cdots, P_{\ell-1}$ and returns the ordered values to SPE.

Step 5: SPE receives $\{E_{pk}(P_0), \cdots, E_{pk}(P_{\ell-1})\}$ and splits r into $\{r_0, \cdots, r_{\ell-1}\}$, each of size $(\kappa + \log_2 Q + 1)$-bits, encrypts each r_i for unmasking the corresponding P_i.

Step 6: To construct Table I, SPE has the encryption of each of $\{a, b, d, e, n_0, n_1, n_2, m_0, m_1, m_2\}$, from which SPE deduces the rest of the variables as follows: $c = n_0 - a - b$; $f = n_1 - d - e$; $g = m_0 - d - a$; $h = m_1 - b - e$; $i = m_2 - c - f$; and $Q = n_0 + n_1 + n_2$. And the table structure in Table I is correctly constructed.

B. Homomorphic Computations

Once the variables of Table I are all computed, any LD statistic measure which require only the available parameters as input, can be computed homomorphically by the SPE without learning the contents of the ciphertexts. For example, in order to estimate the Pearson's correlation coefficient $\hat{\Delta}_{XY}$ from our constructed table of ciphertexts, we rewrite Eq. 7 as follows;

$$\begin{aligned}\hat{\Delta}^2_{XY} &= \\ &\frac{[Q \cdot (e + 2f + 2h + 4i) - (m_1 + 2m_2)(n_1 + 2n_2)]^2}{[Q \cdot (n_1 + 4n_2) - (n_1 + 2n_2)^2][Q \cdot (m_1 + 4m_2) - (m_1 + 2m_2)^2]} \\ &= \frac{[Q \cdot (e + 2(f + h + 2i)) - (m_1 + 2m_2)(n_1 + 2n_2)]^2}{[Q \cdot (n_1 + 4n_2) - (n_1 + 2n_2)^2][Q \cdot (m_1 + 4m_2) - (m_1 + 2m_2)^2]} \ .\end{aligned} \tag{9}$$

The above equation correctly computes the square of the Pearson's correlation coefficient using encrypted inputs. In the same way, we can rewrite Eq. 2 as:

$$\hat{\mathcal{D}}_{XY} = \frac{Q \cdot (2(a + b + d) + e) - (2n_0 + n_1)(2m_0 + m_1)}{2Q^2} \ . \tag{10}$$

According to [16], the goodness-of-fit statistic test for a locus can be re-written as:

$$\chi^2 = \frac{Q \cdot D^2}{p_A \cdot (1 - p_A)} = \frac{Q \cdot (4Q \cdot n_0 - (2n_0 + n_1)^2)^2}{(2Q - (2n_0 + n_1))(2n_0 + n_1)^2} \ , \tag{11}$$

where,

$$D = P_{AA} - p_A^2 \ . \tag{12}$$

Step 7: When the numerators and denominators are computed homomorphically, the encrypted result is forwarded by the SPE to R, who is then required to further run a secure two-party computation with KM for secure decryption. Also note that for the algorithms that require homomorphic multiplication, SPE runs an SMP with KM.

V. Security and Performance Analyses

A. Security and Privacy Analyses

Our aim is to provide privacy of genome data during storage and processing of the data, such that the utility of the data is not lost. For that, we recommend at least 80-bits of security, and for our chosen homomorphic scheme, we require at least 2048-bits of Paillier [14] plaintext size. Our choice of parameters can handle up to $100,000$ samples. On the condition that no two entities within the protocol collude, we have:

Encoder: The $Encoder$ performs a one time operation by encoding and encrypting the inputs. After which, he is not an active member of the protocol. Apart from the values being learnt during encoding, the $Encoder$ learns nothing else about other samples. However, a single individual should not be allowed to encode all data submitted to the SPE, because, such an individual will know the result of computations requested of the SPE. Finally, an $Encoder$ should not know how much samples another $Encoder$ submits to SPE.

TABLE II
COMPUTATIONAL COMPLEXITY FOR CONTINGENCY TABLE.

	Enc.	Add.	Mult.	Dec
Lauter et al.	$6N$	$9N + 14$	$9N$	0
Our proposal	$N + 21$	$N + 12$	0	1

TABLE III
OPERATIONS TIMING RESULTS IN SECONDS.

	Enc.	Add.	Mult.	Dec.
Paillier	0.02287	0.000012	0.14601	0.022738
SEAL	0.08990	0.000323	0.67082	0.087052

SPE: Only ciphertexts are stored on the server, SPE does not learn the content of the data stored on his server, and does not also learn the content of the processed results, since it does not have the secret key. The SPE can however learn the total number of records, and can therefore deduce the value for Q, but this can be mitigated by allowing the $Encoder$ add some dummy samples which are encryptions of zero. The dummy samples will not affect results of computations, but will only add a computation cost to generating Table I, as well as storage cost to the server. Only the SPE is responsible for knowing the changes (insertion and deletion) of samples.

Key Manager: The KM is an honest-but-curious entity, and is only allowed to interact with masked values. Therefore, the KM does not learn the true values he is presented to operate on, so long as he does not collude with another entity.

Researcher: He obtains aggregated results such as numerator and denominator. He does not possess enough information to solve for the individual variables, therefore the privacy of the contributing samples are protected from the Researcher.

B. Performance Analyses

To obtain a fair comparison of our proposal with Lauter et al's. [6], while preserving the same level of security offered by their work, we present here a C++ implementation of both approaches, using GMP library version 6.1.2 on a 64-bits Intel core 2 Quad @ 2.66GHz, running Ubuntu 14.04 LTS and Paillier cryptosystem, with 80-bits security for N samples. We also mention that one SUP requires 21 encryptions, 11 homomorphic additions and one decryption. Furthermore, we present a complexity comparison only for constructing the contingency table in Table II, from which variables are used as inputs to compute LD measures and other genome analyses. We ignore the comparison for computing genome analyses algorithms because those are independent of the sample size.

It can be observed from Table II and III that our approach for computing the contingency table improves Lauter et al's. [6] proposal by 83% storage cost and 98.4% computational cost. Also we present the average runtime for 1000 runs of the cyrptosystems (Paillier, Microsoft Simple Encrypted Arithmetic Library) operations used in Table III.

Finally, we present the complexity for various genome analysis algorithms we implemented in Table IV.

VI. CONCLUSION

We present a secure, privacy-preserving and efficient approach for computing LD measures over genome data. Our approach provides significant improvements in storage and computational complexity from the existing work we compared with. We produce a 98.4% improvement for computing the 3×3 contingency table over that of Lauter et al. We introduce an efficient Key Manager in a semi honest security setting, who we leverage on to implement a secure and privacy-safe packing technique. We implement and show performance results for algorithms that use genome data, such as, Pearson's correlation coefficient, Goodness-of-fit test, LD coefficient. Our approach can also be used to compute other LD measures for which the equations are re-written to be executed using basic additions, subtractions and multiplications. Our construction is robust and can accommodate up to $100,000$ samples for the parameters presented here.

TABLE IV
ALGORITHM COMPLEXITY

	Add.	Sub.	Mult.	Scalar Mult.
Pearson's Corr. Coeff.	7	3	7	4
LD Coeffiecient	5	1	3	4
Goodness-of-fit Test	1	2	6	3

REFERENCES

[1] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[2] D. McMorrow, "The $100 genome: Implications for the dod," tech. rep., DTIC Document, 2010.

[3] S. Wang, X. Jiang, D. Fox, and L. Ohno-Machado, "Preserving genome privacy in research studies," in *Medical Data Privacy Handbook*, pp. 425–441, Springer, 2015.

[4] Z. Lin, A. B. Owen, and R. B. Altman, "Genomic research and human subject privacy," *Science*, vol. 305, no. 5681, pp. 183–183, 2004.

[5] B. A. Malin, "An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future," *Journal of the American Med. Informatics Association*, vol. 12, no. 1, pp. 28–34, 2005.

[6] K. Lauter, A. López-Alt, and M. Naehrig, "Private computation on encrypted genomic data," in *International Conference on Cryptology and Information Security in Latin America*, pp. 3–27, Springer, 2014.

[7] W. Lu, Y. Yamada, and J. Sakuma, "Efficient secure outsourcing of genome-wide association studies," in *Security and Privacy Workshops (SPW),*, pp. 3–6, IEEE, 2015.

[8] B. S. Weir, "Inferences about linkage disequilibrium," *Biometrics*, pp. 235–254, 1979.

[9] A. Ziegler, I. R. König, and F. Pahlke, *A Statistical Approach to Genetic Epidemiology: Concepts and Applications, with an E-learning platform.* John Wiley & Sons, 2010.

[10] L. Kamm, D. Bogdanov, S. Laur, and J. Vilo, "A new way to protect privacy in large-scale genome-wide association studies," *Bioinformatics*, vol. 29, no. 7, pp. 886–893, 2013.

[11] D. Wu and J. Haven, "Using homomorphic encryption for large scale statistical analysis," 2012.

[12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, p. 13, 2014.

[13] A. Shahbazi, F. Bayatbabolghani, and M. Blanton, "Private computation with genomic data for genome-wide association and linkage studies," 2016.

[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, 1999.

[15] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE transactions on information forensics and security*, vol. 7, no. 3, pp. 1053–1066, 2012.

[16] C. Barbacioru, "Biomedical data management." University Lecture, 2005.