TU Delft

Delft University of Technology

PREDICT

Efficient Private Disease Susceptibility Testing in Direct-to-Consumer Model

Ugwuoke, Chibuike; Erkin, Zekeriya; Reinders, Marcel; Lagendijk, Reginald

DOI
10.1145/3374664.3375729

Publication date
2020

Document Version
Accepted author manuscript

Published in
CODASPY 2020 - Proceedings of the 10th ACM Conference on Data and Application Security and Privacy

Citation (APA)
Ugwuoke, C., Erkin, Z., Reinders, M., & Lagendijk, R. (2020). PREDICT: Efficient Private Disease Susceptibility Testing in Direct-to-Consumer Model. In B. Carminati, & M. Kantarcioglu (Eds.), *CODASPY 2020 - Proceedings of the 10th ACM Conference on Data and Application Security and Privacy* (pp. 329-340). ACM. https://doi.org/10.1145/3374664.3375729

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

PREDICT: Efficient Private Disease Susceptibility Testing in Direct-to-Consumer Model

Chibuike Ugwuoke
Delft University of Technology
The Netherlands
c.i.ugwuoke@tudelft.nl

Zekeriya Erkin
Delft University of Technology
The Netherlands
z.erkin@tudelft.nl

Marcel Reinders
Delft University of Technology
The Netherlands
m.j.t.reinders@tudelft.nl

Reginald L. Lagendijk
Delft University of Technology
The Netherlands
R.L.Lagendijk@tudelft.nl

ABSTRACT

Genome sequencing has rapidly advanced in the last decade, making it easier for anyone to obtain digital genomes at low costs from companies such as Helix, MyHeritage, and 23andMe. Companies now offer their services in a direct-to-consumer (DTC) model without the intervention of a medical institution. Thereby, providing people with direct services for paternity testing, ancestry testing and disease susceptibility testing (DST) to infer diseases' predisposition. Genome analyses are partly motivated by curiosity and people often want to partake without fear of privacy invasion. Existing privacy protection solutions for DST adopt cryptographic techniques to protect the genome of a patient from the party responsible for computing the analysis. Said techniques include homomorphic encryption, which can be computationally expensive and could take minutes for only a few single-nucleotide polymorphisms (SNPs). A predominant approach is a solution that computes DST over encrypted data, but the design depends on a medical unit and exposes test results of patients to the medical unit, making the design uncomfortable for privacy-aware individuals. Hence it is pertinent to have an efficient privacy-preserving DST solution with a DTC service. We propose a novel DTC model that protects the privacy of SNPs and prevents leakage of test results to any other party save for the genome owner. Conversely, we protect the privacy of the algorithms or trade secrets used by the genome analyzing companies. Our work utilizes a secure obfuscation technique in computing DST, eliminating expensive computations over encrypted data. Our approach significantly outperforms existing state-of-the-art solutions in runtime and scales linearly for equivalent levels of security. As an example, computing DST for 10,000 SNPs requires approximately 96 *milliseconds* on commodity hardware. With this efficient and privacy-preserving solution which is also simulation-based secure, we open possibilities for performing genome analyses on collectively shared data resources.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CODASPY '20, March 16–18, 2020, New Orleans, LA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7107-0/20/03…$15.00
https://doi.org/10.1145/3374664.3375729

CCS CONCEPTS

• **Security and privacy** → *Cryptography*; *Public key encryption*; *Information-theoretic techniques*; *Privacy-preserving protocols.*

KEYWORDS

SNP, Genome, Privacy-Preserving, Obfuscation, Direct-to-customer, Disease Susceptibility Testing

ACM Reference Format:
Chibuike Ugwuoke, Zekeriya Erkin, Marcel Reinders, and Reginald L. Lagendijk. 2020. PREDICT: Efficient Private Disease Susceptibility Testing in Direct-to-Consumer Model. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy (CODASPY '20), March 16–18, 2020, New Orleans, LA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3374664.3375729

1 INTRODUCTION

The technology for sequencing the human genome continues to improve [42, 44], just as the quest to substantially reduce the cost of sequencing has drawn enormous attention to the medical field, research, and commercial platforms in the last two decades [3, 21, 28, 43, 52]. Consequently, research publications corroborate the plummeting in time required for obtaining a digital version of the human genome[10, 24, 39, 44, 51]. For as low as $100, commercial companies such as Helix, MyHeritage and 23andMe state that they are able to unlock your genome and deliver health and ancestry services directly to customers [1, 20, 30], without intervention of your medical doctor. The downward trajectory of the cost and time for sequencing over the last two decades suggests an inevitable upsurge in availability of digital genome in the near future. Furthermore, digital genome is required by the research community for various studies that help in enhancing our understanding of the basic building blocks of life, the relationship between a gene and a phenotype, better understanding of diseases and their causes, patient responses to treatment, and preventive medicare. Only recently, the drug giant GlaxoSmithKline announced that DNA results from the 5 million customer base of 23andMe will be used to design new drugs, and GlaxoSmithKline have also invested $300 million in 23andMe who are already valued at about $1.75 billion [15, 38]. Commercial platforms have also taken advantage by commodification of the genome, and thus, services such as sequencing, ancestry testing and disease susceptibility testing are now offered to willing customers.

Nucleotides are the building blocks for deoxyribonucleic acid (DNA), which can take any of 4 possible bases (A, T, C, G). Often, there is a genetic variation between individuals of the same species, and this variation could happen as a result of a single substitution of a nucleotide base. A single variation in the nucleotide is known as a single-nucleotide polymorphisms (SNP) if it is not observed in more than 1% of the population [32]. Genome Sequencing is usually the first services a customer procures from a commercial platform, allowing an *in vitro* sample of the customer to be used in obtaining an *in silico* dataset. The *in silico* data is commonly presented in the form of SNPs. The National Library of Medicine records that there are roughly 10 million SNPs in the human genome, and more SNPs are still being identified [32]. SNPs have been identified to contribute to an individual's susceptibility to certain diseases [23, 29, 47, 49, 50]. After obtaining a customer's digital SNPs, the commercial platform can further analyze the dataset for various reasons or services, common of which is the customer's predisposition to diseases.

Our work is interested in how commercial platforms utilize SNPs in providing DST services to its customers within a direct-to-consumer market model. For instance, the conventional process of conducting a DST using the SNPs of an individual is comprised of four basic phases. In the first phase, a sequencer who is a commercial platform sequences the biological genome and obtains a digital version for storage and further analysis. The Second phase consists of a genome owner, to whom a digital genome has been provided, requesting a disease susceptibility test on a particular disease of interest. In the third phase, upon receiving such request, a commercial platform will request SNPs it deems relevant for such a test, and will perform the DST with the SNPs provided by the customer. Finally, the fourth phase is when the DST results are handed over to the genome owner or his doctor. This setting whereby commercial platforms render services such as analyzing the genome, directly to the customers qualifies as a direct-to-consumer (DTC) model [37, 46]. Some customers will use these services to settle paternity or maternity disputes, others will use it to trace their ancestry, or to inspect their genome for disease associated variants [26, 36, 46]. The paradigm shift towards a direct-to-consumer model will impact our medical knowledge as an overwhelming amount of digital genomes will become available as these services become popular [22]. In fact, a survey conducted by Lewis [40] and reported by Pascal [46], documents that as much as 94% of people choose genetic testing out of curiosity.

This paradigm shift towards a DTC service delivery, however, requires that one protects the privacy and security of customers' data when being shared with an untrusted third party. Su and Mclaren et al. [27, 46] argue that the DTC requires a robust combination of regulatory and legal solutions, in order to preserve the confidence that consumers have in using the solution. The implication is that we need efficient privacy-preserving methods for computing on genome data in other to satisfy the customer. In the current model as practiced by the commercial platform, the SNPs of the customer are transmitted *in clear*, which means that at least the commercial platform presents an immediate privacy-risk to the customer. It is difficult to prove that a commercial platform will always adhere to the rules and follow ethical guidelines in protecting the privacy of the genome data, as coercion and disgruntled employees could circumvent such trust models. In fact, only a few 100s SNPs is already enough to threaten the privacy of the customer, by re-identifying an individual even in a large dataset of genetic data [45]. This is further complicated because of familiar relationships between individuals, i.e. information on the genome of a relative also releases information about a customer's own genome. This holds even long after an individual is deceased, thereby posing direct privacy-risks to the relatives. Additionally, whenever genome data is leaked, it is irrevocable and the individual cannot replace the leaked genome with a new set [19]. Together, this places strong privacy requirements for genomic data all through its digital lifespan. Although these customers are only interested in analyzing their genome data for the sheer sake of curiosity [33, 46], it still remains necessary that privacy be guaranteed while satisfying their desire.

The need for providing privacy for all sorts of activities regarding the genome is one that is multifaceted [3], and does require conscious efforts and dedication from legal, ethical, information security and other related research fields. The objective is for customers to have full control over their privacy which requires that genome information about an individual is not shared in a disclosed form with any third party. Secondly, since companies invest a lot of money to understand the genome [38], the SNPs that are relevant to disease and the algorithm for computing DST is regarded as trade secret and should be protected as such. Customers and companies therefore require efficient, provable security and privacy measures that will protect the genome data of a customer and the trade secret of the commercial platform while the customer continues to enjoy services offered by the service provider in the popular DTC model.

Existing information security based privacy-preserving approaches for computing DST commonly adopt cryptographic techniques in plugging the privacy challenges that arises from interactions between the commercial platforms and their customers. For the purpose of simplicity, information security researchers typically concentrate on the last three phases of the described process. This is important because the sequencing phase is dependent on biological samples which cannot easily be protected with information security techniques. Let us assume that in the first phase, the sequencer deletes all data, both biological and digital relating to the customer, or perhaps the customer now has a secure stand alone device for sequencing the genome. This assumption is consistent with existing solutions [4, 12]. The cryptographic techniques commonly deployed in the last three phases are homomorphic encryption and secret sharing. Homomorphic encryption is a technique that allows anyone to encrypt values in a special way such that basic operations like addition and multiplication can be performed on the encrypted data without using the decryption keys [5, 7, 8, 16, 34, 41]. However, computing the DST algorithm over encrypted data is a non-trivial task. Data expansion and computational complexity of homomorphic operations make it expensive, inefficient and hence undesirable for deploying in the wild, as the whole processes is highly time consuming [13, 48]. While secret sharing recommendations are relatively more efficient than their homomorphic encryption counterparts, secret sharing requires that the data be shared amongst various parties with non-collusion restrictions. This does not exactly reflect the ideal scenario as obtainable by current structure of commercial platforms. This is the case because companies do

not usually collaborate to compute disease predisposition for a customer.

In this paper, we recommend an information security solution that protects the privacy of customers' data, companies' trade secret and equally preserve the direct-to-consumer market model most profitable for the commercial platforms. Our protocol enhances the efficiency of the runtime by replacing the homomorphic encryption construction with a lightweight obfuscation technique which is provably secure. We provide customers the ultimate power to decide how, when and with whom they choose to share their genome data.

Our contributions are as follows: 1) We propose PREDICT, a novel protocol which executes the existing susceptibility testing requirements with the use of SNPs, and preserves the increasingly popular DTC model adopted by commercial platforms. 2) PREDICT prioritizes the privacy of the customer and that of the commercial platform. Genome data are resident with the customer in a secure format and not stored in a centralized cloud nor shared with any third-party in an unprotected manner. The result of a test can only be deduced by the customer and he is left with the prerogative to either share the result or not. 3) Our protocol is efficient and can be deployed for practical use. Our design and implementation of this privacy-preserving DST protocol significantly outperforms existing privacy-protection solutions in memory and computational efficiency. As an example, it takes about 96 *milliseconds* to compute DST using 10,000 SNPs on a commodity hardware. And finally, we provide privacy proof based on simulation paradigm, as well as the complexity analysis of our protocol.

The outline for the rest of the paper is as follows: in Section 2 we discuss relevant literature to our work, and how they differ from our proposal. In Section 3 we introduce important building blocks requisite for the construction of our proposed protocol. In Section 4 we introduce and discuss PREDICT. We present the complexity analysis of PREDICT in Section 5, followed by optimisation in Section 6. We provide further discussion on PREDICT in Section 7 and implementation in Section 8. We present the privacy and security analyses in Section 9. Finally, in Section 10, we conclude this paper.

2 RELATED WORK

In this section, we focus on privacy solutions for disease susceptibility testing from an information security perspective. We continue the rest of the discussion with the assumption that every customer or patient already has a digital genome, and the sequencer will play no further role in the interactions. Such digital genome can be securely stored in the cloud or privately kept by the owner in a secure device, this is consistent with proposals in [4, 12]. Previously, a number of other works [4, 12, 27, 31, 46] have proposed solutions for privacy-preserving protocols which perform DST using SNPs or other sensitive data.

A first drawback across these proposals is the inefficiency of the solutions, because adoption of homomorphic encryption introduces significant computational and storage overhead when compared to the non-privacy-preserving solution. One renowned homomorphic encryption based solution is the method proposed by [4]. In the proposal, the bulk of the steps are computations carried out on encrypted data, as seen in Figure 1. Following the same protocol as

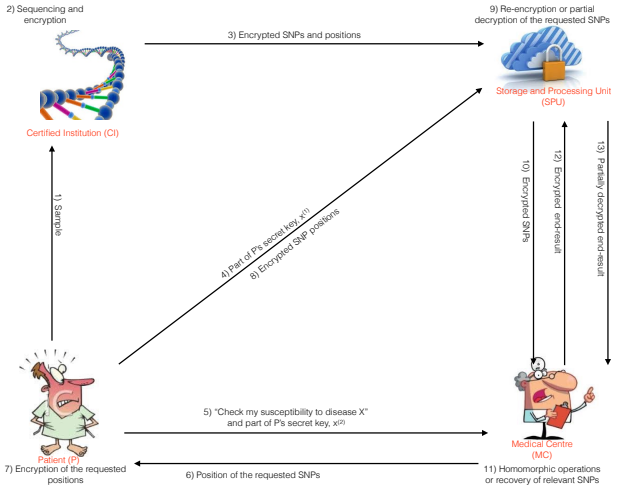

Figure 1: Protocol proposed by Ayday et al.[4]

[4], Namazi et al.[31] proposed a similar solution but replaced the homomorphic scheme used by Ayday et al.[4] with an even more computationally expensive homomorphic scheme. This was done to improve privacy and eliminate communication cost peculiar to additive homomorphic schemes. Lastly, Danezis and Cristofaro [12] recommend an improved proposal that is more efficient than the original work by Ayday et al. Although they offer significant improvements in efficiency, their solution equally involves computation over encrypted data. Danezis and Cristofaro also provides a secret-sharing based variant for privacy-preserving DST, but it still uses encrypted data for its computation. Homomorphic encryption techniques are still evolving and techniques to reduce its computational overhead is still an open problem [2, 6]. It implies that homomorphic encryption based approaches are not yet efficient for scalable practical deployment, especially when large dataset are used.

A second drawback to the homomorphic encryption based protocol by Ayday et al.[4] is that of strict privacy. The disease for which a customer is testing can be easily learned by the processing unit, see Figure 1. In testing for a disease predisposition, the processing unit is allowed to learn the SNPs that are relevant for the computation. Even though the exact values for the SNPs are not known by the processing unit, this is still a privacy concern. Also, in [4, 31], the final result of the test is transmitted to the medical unit rather than the customer, which is not consistent with our objective of granting the customer absolute control to privacy.

A third concern inherent in the existing works [4, 31] is the fact that the protocols proposed do not seamlessly fit into the model as currently observed between the commercial platforms and their customers. The direct-to-consumer relationship preferred by the commercial platforms is not well reflected in the mentioned proposal. The medical units continue to play an inalienable role in those proposals, which makes these protocol not suitable for a DTC service delivery.

Other drawbacks of the state-of-the-art proposal by Ayday et al.[4] include the following:

- Their approach proposes to store the protected genome data with a storage and processing unit (SPU). This requires that individuals must store their genome data encrypted on a central cloud infrastructure, making it enticing for attackers.
- The protocol assumes that the homomorphic operations are computed at the Medical Centres (MC). This is not practical in the wild as such operations are computationally expensive for average medical centres to carry out. Furthermore, the MC is not often equipped with the technical and security requirements for handling such operations.

We propose a protocol that aids customers and commercial platforms to interact and compute disease susceptibility test in a privacy-preserving manner without being bedeviled with the above listed drawbacks.

3 PRELIMINARIES

In this section, we provide the necessary building blocks required to understand our proposed protocol. These include: the cryptographic protocols such as multi-party computation, obfuscation techniques as well as the functions required to compute disease susceptibility testing. In this work, we adopt the *semi-honest* a.k.a *honest-but-curious* security model, which implies that every stakeholder is expected to judiciously follow the rules of the protocol, but can be passively curious to learn extra information from data they can observe.

3.1 Secure Multi-party Computation

A secure multi-party computation (MPC) is an interactive cryptographic protocol that allows for two or more mistrusting parties to jointly compute a function using their private data as input [11, 18]. It allows for the output of the desired function to be public but the contributed inputs remains private upon the assumption that each party does not digress from the rules of the protocol. We deploy the concept of MPC by allowing three mis-trusting parties (customers, commercial platforms and processing unit) to jointly compute a DST function using their private inputs. MPCs are commonly designed in a semi-honest security model, because the adversary is considered to be able to control some parties in the protocol.

3.2 Homomorphic Encryption

Homomorphic encryption allows for arbitrary algebraic operations to be performed on ciphertexts. Let $Enc_{pk}(\cdot)$ and $Dec_{sk}(\cdot)$ represent encryption and decryption functions respectively. (m_1, m_2) are two messages and k is a scalar value, while $\boxplus$, $\boxdot$ and $\boxtimes$ are arbitrary operations on the ciphertexts. Then, homomorphism is defined as:

$$Dec_{sk}(Enc_{pk}(m_1) \boxplus Enc_{pk}(m_2)) = m_1 + m_2 , \quad (1)$$

$$Dec_{sk}(Enc_{pk}(m_1) \boxdot Enc_{pk}(m_2)) = m_1 \cdot m_2 , \quad (2)$$

$$Dec_{sk}(Enc_{pk}(m_1) \boxtimes k) = m_1 \cdot k . \quad (3)$$

3.2.1 Paillier Scheme [34] : Paillier cryptosystem is an additively homomorphic scheme. Given a public key, private key pair (pk, sk), and $n := p \cdot q$, s.t p and q are distinct large primes, $pk := (g, n)$ and $sk := \lambda(n)$, where g is generator of order n, $\lambda(n)$ is the Carmichael's function on n, expressed as $\lambda(n) := lcm(p-1, q-1)$.

Enc: $c := Enc_{pk}(m, r) := g^m \cdot r^n \ mod\ n^2$, where $c \in \mathbb{Z}^*_{n^2}; r \leftarrow \mathbb{Z}^*_n$.

Dec: Given c, $m := \frac{\mathbb{L}_n(c^\lambda \ mod\ n^2)}{\mathbb{L}_n(g^\lambda \ mod\ n^2)} \ mod\ n$ and $\mathbb{L}_n(a) := \frac{a-1}{n}$.

Additive Homomorphism: Given two ciphertexts of messages m_0 *and* m_1, we can compute the sum as follows:

$$\begin{aligned} Enc_{pk}(m_0, r_0) \times Enc_{pk}(m_1, r_1) &:= (g^{m_0} \cdot r_0^n \times g^{m_1} \cdot r_1^n) \\ &:= (g^{m_0+m_1} \cdot (r_0 \cdot r_1)^n \ mod\ n^2) \\ &:= Enc_{pk}(m_0 + m_1) . \end{aligned}$$

With the above mentioned homomorphic scheme, it is possible to compute simple linear functions such as aggregation of encrypted values. However, more complex function that involves division, multiplication and other complex operations on encrypted data are not feasible. Complex functions are usually computed by the introduction of a third party who decrypts ciphertexts, computes the complex operation *in clear* and re-encrypts the results.

It is evident by inspection that ciphertexts are generated modulo n^2 and this results to data expansion for every encrypted value. Computing on the ciphertext introduces computational overhead since n should not be less than 2048 bits in order to obtain 112-bit security, being that 112-bit security is considered sufficient between the in the year 2016 up until 2030 [17]. The reader is referred to [35] for further details on the Paillier scheme.

3.3 Privacy-Preserving DST

Every individual inherits a pair of allele to make a gene at a locus, each allele comes from each contributing parents. An allele can either be major or minor, depending on what percentage of the population have them. Since a gene is made up of two alleles, it can occur in any of the three possible classes: two major alleles, two minor alleles, or a major and minor allele. Subsequently, we use the term 'SNP value' to represent the class in which an individual's SNP falls within. We denote the 'SNP values' as {0, 1, 2} to denote No SNP (two major alleles), heterozygous SNP (a major and minor allele) and homozygous SNP (two minor alleles) respectively. For a DNA sequence belonging to a customer, we denote the SNP at locus i as SNP_i , where $SNP_i \in \{0, 1, 2\}$.

A *weighting averaging* function is used to compute the susceptibility of a patient to a disease X. A DST can be computed by weighing the contribution of each SNP to the disease X , as follows:

a) Let $L(x)$ represent a set of all known SNPs that contribute to the disease X and C_i represents the weight that a SNP at locus i contributes to the disease X.
b) $Pr(X|SNP_i)$ denotes the probability that an individual has a disease X, conditioned on the SNP value at the locus i.
c) A SNP at locus i contributes $C_i \cdot Pr(X|SNP_i = j)$, with $j \in \{0, 1, 2\}$.
d) The aggregation of all loci is then $\sum_{i \in L(x)} C_i \cdot Pr(X|SNP_i = j)$.
e) The aggregation is then normalized over all weights to obtain a disease susceptibility test score:

$$S^X = \frac{1}{\sum_{t \in L(x)} C_t} \cdot \sum_{i \in L(x)} C_i \cdot Pr(X|SNP_i = j) . \quad (4)$$

It can be seen from Eq. 4 that three inputs are required to compute S^X , all of which are considered privacy-sensitive and should not be shared unprotected.

- Only the customer knows SNP_i values, as this is his private data.
- The commercial platform knows C_i and $Pr(X|SNP_i = j)$ values, these being his trade secrets.

By adopting homomorphic encryption to compute Eq. 4 in a privacy-protected setting, Ayday et al. has re-written the equation to:

$$S^X = \frac{1}{\sum_{t\in L(x)} C_t} \times \sum_{i\in L(x)} C_i \Bigg[\frac{p_0^i(X)}{(0-1)(0-2)}[SNP_i - 1]\times [SNP_i - 2] + \frac{p_1^i(X)}{(1-0)(1-2)}[SNP_i - 0]\times[SNP_i - 2] + \frac{p_2^i(X)}{(2-0)(2-1)}[SNP_i - 0]\times[SNP_i - 1]\Bigg], \quad (5)$$

where $p_0^i(X) = Pr(X|SNP_i = 0)$, $p_1^i(X) = Pr(X|SNP_i = 1)$ and $p_2^i(X) = Pr(X|SNP_i = 2)$.

In the proposal by Ayday et al. [4], refer to Figure 1, Eq. 5 is computed homomorphically with the use of an additive homomorphic encryption scheme. Due to the inability of the adopted homomorphic encryption scheme to perform a homomorphic multiplication operation, their protocol is designed to store the encrypted values of $(SNP_i)^2$ which requires additional storage space. The final result as obtained in the protocol described by Ayday et al.[4] is decrypted by the medical unit, who is able to view the result and communicate professional opinion to the patient. However, the storage and processing unit does not have a view of the SNP values *in clear*, despite having to store and process the data. Nevertheless, the storage and processing unit knows which loci have no SNP from those that have at least a single SNP. Therefore the patient's SNPs are not completely private against the storage and processing unit.

Due to the privacy concerns mentioned above and the computational inefficiency introduced by homomorphically computing Eq. 4, we introduce a novel protocol for computing DST. Our protocol will optimally compute DST by replacing homomorphic encryption with a secure obfuscation technique using MPC, where each sensitive data input is masked using a one-time-only secure random number. Our proposed protocol removes the medical unit from the setting, replacing it with a commercial platform and ensures that the processing unit does not learn the SNP loci of a customer as was the case in the protocol by Ayday et al. Lastly, in order to achieve these, we again re-write Eq. 4, by drawing insight from Eq. 5:

a.) Redefine C_i to be the normalized term $\frac{C_i}{\sum_{t\in L(x)} C_t}$.

b.) Re-write Eq. 4 to a simpler form using Eq. 5,

$$S^X = \sum_{i\in L(x)} C_i \Bigg[\frac{1}{2}p_0^i(SNP_i - 1)(SNP_i - 2) - p_1^i(SNP_i - 0)(SNP_i - 2) + \frac{1}{2}p_2^i(SNP_i - 0)(SNP_i - 1)\Bigg] .$$

c.) Collect like terms,

$$S^X = \sum_{i\in L(x)} C_i \left[SNP_i^2\left(\frac{p_0^i}{2} - p_1^i + \frac{p_2^i}{2}\right) - SNP_i\left(\frac{3p_0^i}{2} - 2p_1^i + \frac{p_2^i}{2}\right) + p_0^i\right] . \quad (6)$$

From Eq. 6, customers own variables SNP_i and SNP_i^2, while the commercial platform owns

$$a_i = C_i\left(\frac{p_0^i}{2} - p_1^i + \frac{p_2^i}{2}\right) ,$$
$$b_i = C_i\left(\frac{3p_0^i}{2} - 2p_1^i + \frac{p_2^i}{2}\right) ,$$
$$v_i = p_0^i .$$

Hence, we can now compute DST as:

$$S^X = \sum_{i\in L(x)} a_i \cdot SNP_i^2 - b_i \cdot SNP_i + v_i . \quad (7)$$

3.4 Secure Inner Product with Obfuscation

Henceforth, we represent vectors in bold characters, example $\mathbf{X}$, and $|\mathbf{X}|$ denotes the number of elements in $\mathbf{X}$. We will occasionally use the notation $\mathbf{X}[i]$ to represent the i-th term of the vector $\mathbf{X}$. Consider two parties *Alice* and *Bob*, each having a vector of values and they wish to compute the inner product of the vectors without revealing the individual values of each vector. *Alice* holds the vector $\mathbf{X} = \{x_0, x_1, \ldots, x_{n-1}\}$ and *Bob* holds $\mathbf{Y} = \{y_0, y_1, \ldots, y_{n-1}\}$ both of size $n \in \mathbb{Z}$. They wish to compute the inner product of their vectors $\mathbf{X}\cdot\mathbf{Y} = \sum_{k=0}^{n-1} x_k \cdot y_k$, and the result known only to *Alice* .

In order to solve the secure inner product problem, Du and Atallah[14] propose a three party protocol which uses additive masking to obfuscate the values. Their protocol introduces an untrusted third party *Charlie* that only helps in data computation. *Charlie* can be viewed as a cloud infrastructure that helps with the computation of the inner product function. Their protocol is described as follows:

1. *Alice* and *Bob* jointly generate two random numbers r and r' .
2. *Alice* and *Bob* jointly generate two random vectors $\mathbf{R}, \mathbf{R'}$ of size n .
3. *Alice* sends $\mathbf{w}_1 = \mathbf{X} + \mathbf{R}$ and $s_1 = \mathbf{X}\cdot\mathbf{R'} + r$ to *Charlie* .
4. *Bob* sends $\mathbf{w}_2 = \mathbf{Y} + \mathbf{R'}$ and $s_2 = \mathbf{R}\cdot(\mathbf{Y} + \mathbf{R}) + r'$ to *Charlie* .
5. *Charlie* computes $v = \mathbf{w}_1\cdot\mathbf{w}_2 - s_1 - s_2$, and sends the result to *Alice* .
6. *Alice* computes $\mathbf{X}\cdot\mathbf{Y} = v + (r + r')$.

This secure inner product uses additive masking to obfuscate each sensitive value before sharing with other parties.

We modify the above inner product protocol by first replacing the r and r' integers with vectors. Also rather than *Alice* and *Bob* jointly generating $\mathbf{r}$, we propose that they do this independently. That way, we can use the values in $\mathbf{r}$ as a one-time only random number only known to the party who generates it.

The modified algorithm is presented in Algorithm 1. All operations in Algorithm 1. are computed modulo a large prime q , and random numbers are chosen to be cryptographically secure. Due to simplicity of expression, the algorithm and other operations are not presented to show reduction modulo q , but it should be noted that it is implied.

Algorithm 1 Secure 3-Party Inner Product Protocol

procedure INITIALIZATION
  Set variable n, and publish to *Alice* and *Bob*
  Alice and *Bob* jointly generate random vectors $\mathbf{R}_A$ and $\mathbf{R}_B$, each of size n
  Alice and *Bob* independently generate random vectors $\mathbf{r}_A$ and $\mathbf{r}_B$ respectively, each of size n
end procedure
procedure Alice
  for $i = 0 \rightarrow (n-1)$ **do**
    $\mathbf{W}_A[i] := \mathbf{X}[i] + \mathbf{R}_A[i]$
    $\mathbf{S}_A[i] := \mathbf{X}[i] \cdot \mathbf{R}_B[i] + \mathbf{r}_A[i]$
  end for
  Alice sends $\mathbf{W}_A, \sum_{i=0}^{n-1} \mathbf{S}_A[i]$ to *Charlie*
end procedure
procedure Bob
  for $i = 0 \rightarrow (n-1)$ **do**
    $\mathbf{W}_B[i] := \mathbf{Y}[i] + \mathbf{R}_B[i]$
    $\mathbf{S}_B[i] := \mathbf{R}_A[i] \cdot (\mathbf{Y}[i] + \mathbf{R}_B[i]) + \mathbf{r}_B[i]$
  end for
  Bob sends $\mathbf{W}_B, \sum_{i=0}^{n-1} \mathbf{S}_B[i]$ to *Charlie*
  Bob sends $\sum_{i=0}^{n-1} \mathbf{r}_B[i]$ to *Alice*
end procedure
procedure Charlie
  $temp := \sum_{i=0}^{n-1} \mathbf{W}_A[i] \cdot \mathbf{W}_B[i]$
  $V := temp - \mathbf{S}_A[i] - \mathbf{S}_B[i]$
  Send V to *Alice*
end procedure
Alice computes $\mathbf{X} \cdot \mathbf{Y} := V + \sum_{i=0}^{n-1} \mathbf{r}_A[i] + \sum_{i=0}^{n-1} \mathbf{r}_B[i]$

4 PREDICT

It is now clear that our goal is to compute Eq. 7 using secure multi-party computation. We adopt the modified version of the secure inner product protocol proposed by Du and Atallah [14], as presented in Algorithm 1. We replace three parties with the three stakeholders required for computing DST. The secure inner product protocol is used to compute $\sum_{i \in L(x)} a_i \cdot SNP_i^2$ and $\sum_{i \in L(x)} b_i \cdot SNP_i$ components and finally the aggregation of the $\sum_{i \in L(x)} v_i$ component.

4.1 Private DST in DTC model

In our protocol as shown in Figure 2, there are 4 parties involved but only 3 parties required in computing the protocol. There is a non-collusion assumption on the parties, implying that no two parties are allowed to collaborate with others to learn more information than the protocol permits them.

The protocol by Ayday et al.[4] assumes that the genome data belongs to a patient, thereby presuming that the medical centre (genome analysing unit) is trusted to see the result of the DST. We adopt a contrary assumption. An individual ***P*** is any customer who seeks to learn information from their genome due to sheer curiosity. Our assumption makes it easier to appreciate why P might not necessarily want to share the end result of the susceptibility test with any party including the medical centre.

The protocol parties are as follows:

(i) The individual ***(P)***, is the customer whose genome is considered for analysis. P owns the SNPs required as input for the execution of the DST protocol, and will contribute them for computation in a privacy-preserving manner.
(ii) The Genome Analysing Unit ***(GA-Unit)***, represents a commercial platform that offers genome analyses as a service using a DTC model. This entity is considered to have a reputation that must be protected, therefore, should conform to ethical requirements within their field. To be simply put, ***GA-Unit*** is not assumed to be malicious. From Eq. 7, ***GA-Unit*** holds the values for $\{\mathbf{a}, \mathbf{b}, \mathbf{v}\}$ and would want to keep them private as well.
(iii) The Certified Sequencing Institution ***(CSI)***, handles sequencing of the genomes and transforming the biological sample of genomes to a digital format. The ***CSI*** is equally bounded to conform to ethical values. Although the ***CSI*** performs sequencing, it is not involved in the DST protocol and will not be further discussed during the course of computing DST.
(iv) The Processing Unit ***(PU)***, has a lot of processing resources that are required to handle huge computations. He is not to be trusted with unprotected genome data, but is assumed to follow the protocol and execute the expected computations. Since the ***PU*** does not contribute any input data to the evaluation of the function, it has no concern for privacy.

We further denote SNP_i as k_i, hence Eq. 7 is now re-written to:

$$S^X := \sum_{i \in L(x)} a_i k_i^2 - b_i k_i + v_i \tag{8}$$

It has been shown that Eq.8 can be computed using the secure inner product protocol. Let $\pi = |L(x)|$ be the number of SNPs needed for computing the susceptibility of a disease. Each SNP (represented by k_i) requires two multiplications and two additions. This means that if the number of SNPs required for computing a DST for a disease X is π . Therefore, the size of the vector contributed by both ***GA-Unit*** and ***P*** shall be of size 2π . The addition of v_i is taken care of by the aggregation property intrinsic in the secure inner product protocol.

4.2 Privacy and Security Assumptions

Figure 2 shows the interactions and work-flow between parties within our proposed protocol. Our protocol is designed with the following security assumptions in mind:

a) The individual (***P***) is aware of the sensitivity of his genome data and needs to utilise the computational ability of the processing unit or cloud infrastructure and the knowledge of the genome analysing unit, without leaking any sensitive information to the ***PU*** and ***GA-Unit***. On the contrary, ***P*** does not pose any privacy threat to ***PU*** during the course of execution of the protocol.
b) ***GA-Unit*** is in possession of the individual weights of SNPs for diseases. The weights are considered trade secrets and

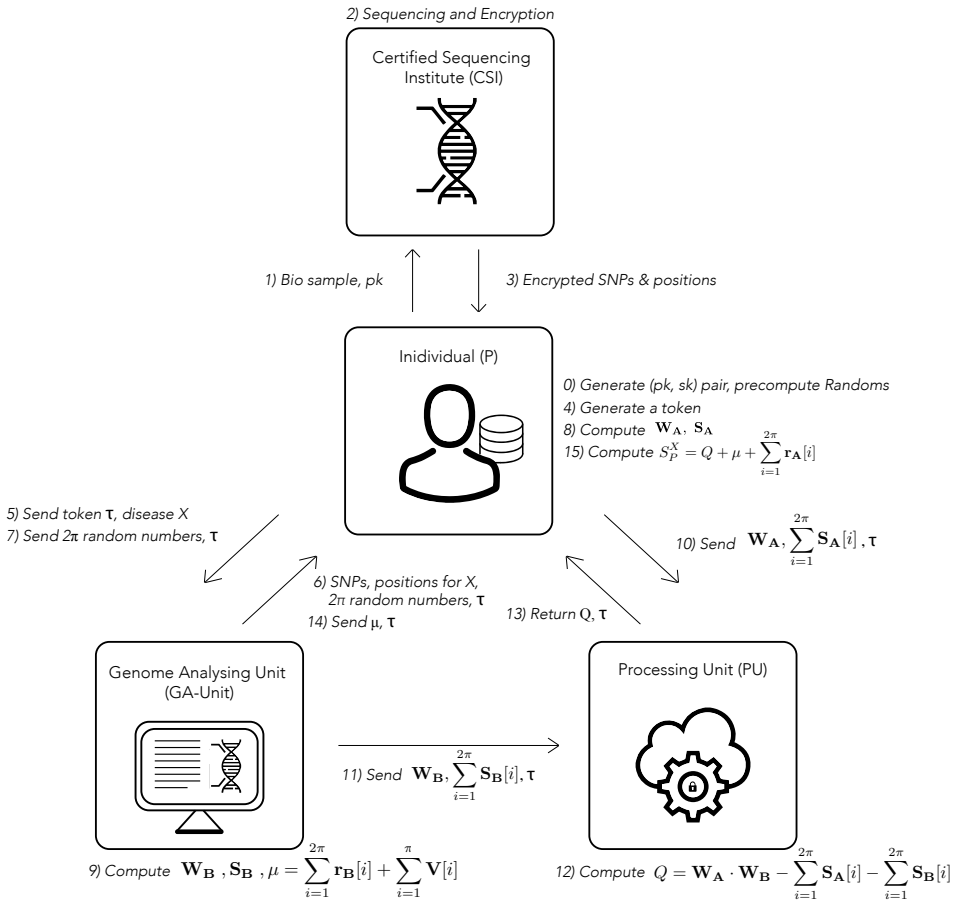

Figure 2: Private Disease Susceptibility Test on a DTC model

should equally be protected from ***P*** and ***PU***, while being used to privately compute the susceptibility of ***P*** to a disease.

c) The device on which ***P***'s data are stored, is considered secure and only accessible with ***P***'s permission.

4.3 Protocol Description

A detailed description of steps in the proposed protocol is as follows:

- **Step 0:** ***P*** generates a pair of cryptographic keys, consisting of a private key and a public key. The public key is made public to ***CSI*** only, for encrypting the digital sequence of ***P***'s genome. This case is slightly different if we adopt the symmetric key option, where ***P*** generates a single private key with the hope of sharing it with ***CSI***.
- **Step 1:** ***P*** sends a biological sample of his genome to the ***CSI*** for sequencing, alongside his public key (or private key in the case of a symmetric crypto scheme).
- **Step 2:** ***CSI*** receives the sample and sequences the genome, produces it *in silico* and encrypts the sequence with the public key of ***P***.
- **Step 3:** ***CSI*** returns the encrypted SNPs and the corresponding locations to ***P***. The ***CSI*** is also obligated to securely delete all copies and traces of the genome data. This is necessary because the ***CSI*** is not expected to further participate actively in the remaining part of the protocol.
- **Step 4:** At this point, ***P*** is in full possession and control of his encrypted SNPs, and our assumption allows that only the individual has any possible copy of his SNPs. Now, ***P*** performs a one-time only decryption of the encrypted data, and saves the SNP related data in his secure device. We assume that such a device is protected and is private to only ***P***. For a susceptibility test to be initiated by ***P***, he is first required to generate a random token τ (a 160-bit value). This token is unique and used to reference every instance of a disease susceptibility test.
- **Step 5:** It is important to note that our assumption presumes that ***P*** is online and is therefore able to carry out his part of the protocol. It is expected that ***P*** is active and is able to locate a publicly available unique disease identifier (ID) published by ***GA-Unit***. The public identifiers (IDs) may be published on a website, and ***P*** can then send a token τ and the disease ID to ***GA-Unit*** after necessary authentication.

- **Step 6:** The ***GA-Unit*** will verify the request from $\boldsymbol{P}$, and then responds with the SNP positions for the disease X, where X is the corresponding disease for the supplied ID. The ***GA-Unit*** will always demand a constant size π of SNPs. This is to make it more difficult for an observer to infer what disease $\boldsymbol{P}$ is interested in. The extra (dummy) SNPs are pertinent to obfuscate the actual SNPs relevant to the disease. The set of dummy SNPs are to be deterministic for any disease. As to prevent $\boldsymbol{P}$ from distinguishing real SNPs from dummy SNPs. Also, the ***GA-Unit*** sends 2π random numbers to $\boldsymbol{P}$, called vector $\mathbf{R}_B$. These randoms will be used for the secure inner product protocol. Finally, ***GA-Unit*** generates vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{v}$.
- **Step 7:** When $\boldsymbol{P}$ receives the SNP positions from ***GA-Unit***, he generates 2π random numbers and sends the vector to ***GA-Unit***. We denote this random number vector as $\mathbf{R}_A$.
- **Step 8:** In order for both $\boldsymbol{P}$ and ***GA-Unit*** to compute Eq. 8, they agree to split the equation into two parts $a_i k_i^2$ and $(-b_i k_i + v_i)$. $\boldsymbol{P}$ generates another set of 2π random numbers, denoted as $\mathbf{r}_A$. Generates the vector $\mathbf{W}_A$ by computing $\mathbf{W}_A[i] := k_i^2 + \mathbf{R}_A[i]$ and $\mathbf{W}_A[i+\pi] := -k_i + \mathbf{R}_A[i+\pi]$. The vector $\mathbf{S}_A$ is also generated by computing $\mathbf{S}_A[i] := k_i^2 \cdot \mathbf{R}_B[i] + \mathbf{r}_A[i]$ and $\mathbf{S}_A[i+\pi] := -k_i \cdot \mathbf{R}_B[i+\pi] + \mathbf{r}_A[i+\pi]$.
- **Step 9:** The ***GA-Unit*** first has to generate a vector of 2π random numbers, which we refer to as $\mathbf{r}_B$. Then, just like $\boldsymbol{P}$ generated $\mathbf{W}_A, \mathbf{S}_A$, ***GA-Unit*** generates $\mathbf{W}_B, \mathbf{S}_B$ as follows: $\mathbf{W}_B[i] := a_i + \mathbf{R}_B[i]$ and $\mathbf{W}_B[i+\pi] := b_i + \mathbf{R}_B[i]$. Then the vector $\mathbf{S}_B$ is generated as $\mathbf{S}_B[i] := \mathbf{R}_A[i](a_i + \mathbf{R}_B[i]) + \mathbf{r}_B[i]$ and $\mathbf{S}_B[i+\pi] := \mathbf{R}_A[i+\pi](b + \mathbf{R}_B[i+\pi]) + \mathbf{r}_B[i+\pi]$. Lastly, ***GA-Unit*** computes the variable $\mu := \sum_{i=1}^{2\pi} \mathbf{r}_B[i] + \sum_{i=1}^{\pi} \mathbf{v}[i]$.
- **Step 10:** $\boldsymbol{P}$ transmits the values $\mathbf{W}_A$, $\sum_{i=1}^{2\pi} \mathbf{S}_A[i]$ to ***PU***.
- **Step 11:** The ***GA-Unit*** transmits the values $\mathbf{W}_B$, $\sum_{i=1}^{2\pi} \mathbf{S}_B[i]$ to ***PU***.
- **Step 12:** ***PU*** computes $Q := \mathbf{W}_A \cdot \mathbf{W}_B - \sum_{i=1}^{2\pi} \mathbf{S}_A[i] - \sum_{i=1}^{2\pi} \mathbf{S}_B[i]$.
- **Step 13:** ***PU*** sends Q to $\boldsymbol{P}$.
- **Step 14:** The ***GA-Unit*** sends μ to $\boldsymbol{P}$.
- **Step 15:** Finally, $\boldsymbol{P}$ computes $S_P^X := Q + \mu + \sum_{i=1}^{2\pi} \mathbf{r}_A[i]$.

4.4 Correctness

The proof of correctness for the computation of S^X in PREDICT inherits the proof of correctness of the secure inner product protocol. The original protocol by Du and Atallah [12] computes an inner product of vectors between two mistrusting parties. However, our equation is of the form $\sum_{i=1}^{\pi} a_i \cdot k_i^2 - b_i \cdot k_i + v_i$, hence we require a slight modification to the original protocol. Variables $(-\mathbf{k}, \mathbf{k^2})$ are contributed by $\boldsymbol{P}$ while variables $(\mathbf{a}, \mathbf{b}, \mathbf{v})$ belong to the ***GA-Unit***. Each party then holds a vector for their variables, and each vector is of size π . First, we split the equation into two parts that can each be executed using an instance of the secure inner product protocol. One half of the equation is $\sum_{i=1}^{\pi} a_i \cdot k_i^2$ and the other $\sum_{i=1}^{\pi} (-k_i \cdot b_i + v_i)$. Nevertheless, since we are computing a modified version of the form $\sum_{i=1}^{\pi} (b_i \cdot k_i + v_i)$, it suffices to show that a circuit that correctly computes $\sum_{i=1}^{\pi} a_i \cdot k_i$, can be modified to compute $\sum_{i=1}^{\pi} (a_i \cdot k_i + v_i)$ without loss of security.

Since $\mathbf{v}$ is considered to be part of ***GA-Unit***'s trade secret [12], we have to transfer $\mathbf{v}$ to $\boldsymbol{P}$ without revealing the value *in clear*. We do this by additively masking $\mathbf{v}$ values with random numbers. Specifically, we use the random numbers $\mathbf{r}_B$ which is known to ***GA-Unit*** but oblivious to $\boldsymbol{P}$ in **Step 9** to mask the values of $\mathbf{v}$. This operation still preserves the correctness of the computation. Having established that $\sum_{i=1}^{\pi} (-k_i \cdot b_i + v_i)$ can be computed using the secure inner product protocol, we perform one more step. We merge the vectors of $\sum_{i=1}^{\pi} a_i \cdot k_i^2$ and $\sum_{i=1}^{\pi} (-k_i \cdot b_i + v_i)$ by appending the latter to the former. We produce a new vector of size 2π, with which we compute the secure inner product. Thus, an equation of the form $\sum_{i=1}^{\pi} a_i \cdot k_i^2 - b_i \cdot k_i + v_i$, can be correctly computed using PREDICT.

5 COMPLEXITY ANALYSIS

In Table 1, we present an overview of the communication complexity of our protocol. All units of data transfer are in bits, except for Step 3 of Figure 2, where the data is represented in megabyte (MB). Let $\mathcal{N}$ denote the plaintext size (in bits) for the crypto scheme adopted, which provides an appropriate security level (default value: κ = 112-bits). An example of a SNP reference is *rs138055828*, we only consider the number numeric part of the reference code. All random numbers generated are of size 132-bits, while each of the variables $(\mathbf{a}, \mathbf{b}, \mathbf{v})$ contributed by ***GA-Unit*** is of size 20 bits. Let M represent the number of SNPs that can be packed into a single ciphertext (without any SNP overflowing into a new block of ciphertext). Consequently,

$$M_{RSA} := \left\lfloor \frac{\mathcal{N} - 128}{36} \right\rfloor = \left\lfloor \frac{2048 - 128}{36} \right\rfloor = 53 \text{ , and}$$
$$M_{AES} := \left\lfloor \frac{\mathcal{N}}{36} \right\rfloor = \left\lfloor \frac{128}{36} \right\rfloor = 3 \text{ ,} \quad (9)$$

where 128-bits are required for RSA padding.

If an individual has 10 million SNPs as reported by The National Library of Medicine[32], then let t denote the number of packed ciphertext blocks produced on encrypting 10 million records (SNPs).

$$t_{RSA} := \left\lceil \frac{10,000,000}{M_{RSA}} \right\rceil = \left\lceil \frac{10,000,000}{53} \right\rceil = 188,680 \text{ , and}$$
$$t_{AES} := \left\lceil \frac{10,000,000}{M_{AES}} \right\rceil = \left\lceil \frac{10,000,000}{3} \right\rceil = 3,333,334 \text{ .} \quad (10)$$

For any individual who has 10 million SNPs, we require 188680 units of RSA ciphertexts, where a single ciphertext is of size 2048 bits. From Table 1, it is clear that the communication complexity is linear in the number of SNPs required for a DST. In fact, even for 10 million SNPs, the protocol will require less than 1GB of data transfer during the DST computation.

The computational overhead of our protocol is shown in Table 2. Although the number of operations are provided for simplicity, we note that not all operations are of the same complexity. For instance, addition counts in Table 2 include adding a 2-bit and a 133-bit numbers, as well as adding a 132-bit and a 265-bit number. From Table 2, it can be deduced that the computational complexity

Table 1: DATA COMMUNICATION COMPLEXITY FOR THE PROPOSED PROTOCOL

	Received (bits)				
Sent (bits)		***CSI***	***P***	***PU***	***GA-Unit***
	CSI	–	3) AES: 51MB, RSA: 47MB	–	–
	P	1) AES: 128, RSA: 2065	–	10) $266\pi + 297 + \log_2(2\pi)$	5) 288
	PU		13) $426 + \log_2(2\pi)$	–	7) $264\pi + 160$
	GA-Unit	–	6) $298\pi + 160$ 14) $293 + \log_2(2\pi)$	11) $266\pi + 426 + \log_2(2\pi)$	–

Table 2: COMPUTATION COMPLEXITY

	P	***GA-Unit***	***PU***
Addition	$6\pi + 2$	12π	2
Multiplication	2π	2π	2π

for computing a disease susceptibility test is linear in the size (π) of the SNPs relevant for computing such a test.

6 OPTIMIZATION OF PREDICT.

Firstly, a variant setting of this protocol could be achieved by altering the flow of operations halfway into the protocol. For instance, rather than have the ***GA-Unit*** send $\mu := \sum_{i=1}^{2\pi} \mathbf{r}_B[i] + \sum_{i=1}^{\pi} \mathbf{v}[i]$ to ***P***, ***GA-Unit*** can send $\mu := \sum_{i=1}^{2\pi} \mathbf{r}_B[i]$ to ***P*** and send the other value to ***PU*** indirectly by modifying **Step 11:** to $\sum_{i=1}^{2\pi} \mathbf{S}_B[i] - \sum_{i=1}^{2\pi} \mathbf{r}_B[i] - \sum_{i=1}^{\pi} \mathbf{v}[i]$. This will reduce the computation and communication overhead on ***P*** and place it on ***PU*** who is assumed to have sufficient resources. This will save both computation and communication costs for ***P***. Moreover, the correctness of the protocol will still hold. Adopting this optimization will offer significant improvement where large number of SNPs are required and the security level is equally very high. However, for the default setting of this protocol, such an optimization will not offer a significant improvement.

Secondly, due to the time it takes to generate random numbers, ***P*** and ***GA-Unit*** might have to pre-generate random numbers as part of the preprocessing phase. This saves time and allow for the rest of the protocol to be executed seamlessly, with only basic operations.

Thirdly, another optimization step is to adopt a symmetric key crypto scheme for encryption in **Step 1**. By this, we achieve a reduction in bits of the ciphertext being transferred from the ***CSI*** to ***P*** in **Step 3**. Since a symmetric crypto scheme will offer faster encryption and decryption operations, this offers a computational reduction in the time it takes ***P*** to decrypt his sequence. Recall that the decryption operation has to be performed only once. Thereafter, the data will be stored *in clear* within the secure device of ***P***. However, this approach requires that the ***CSI*** and ***P*** will share P's secret key.

Finally, we recommend a data packing technique in encrypting the SNPs. Every SNP can be represented with 36 bits, given that 2 bits represent the SNP value (0, 1, 2) refer to Section 3.3. The other 34 bits are for referencing the SNP, otherwise known as the SNP position. Data packing will group more than one SNP into a single block of ciphertext, thereby optimizing the time required to decrypt and access the entire SNPs of an individual.

7 DISCUSSION

PREDICT differs from the proposal by Ayday et al.[4] as follows:

- Our primary aim is to protect the privacy of an individual's genome data from all other entities in the protocol, while being able to harness their abilities to test for disease susceptibility. Only the individual ***P*** is allowed to view the result of every susceptibility test. However, Ayday et al's protocol does not seek to protect the privacy of the individual's genome data from the genome analyzing unit, which results from their assumption that the individual is a patient and the genome analyzing unit is a doctor in a medical institution.
- We propose that genome data should be stored in a dedicated piece of hardware, that should only be accessible by the individual ***P***. This allows ***P*** to have full control of his digital genome data and also provides him the freedom to change the cryptographic keys and other security measure when necessary. These can be done without incurring much cost or informing a third party about the intentions to make changes to the cryptographic keys. Our choice to decentralize the genome data storage helps to reduce the risk of targeting a central cloud storage infrastructure.
- Our protocol guarantees ***P***'s independence from a medical unit. Thereby, realizing our aim of providing privacy for curiosity driven individuals, and at the same time offering a DTC service for disease susceptibility testing using genetic data. The protocol by Ayday et al. is not designed to target a DTC scenario.
- In our setting, the obfuscation of the SNP positions and values are meant to be computed by ***P*** and sent to the ***PU***. Replacing encryption with randomization eliminates the expensive homomorphic operations for all parties. The ***GA-Unit*** is not expected to possess the processing power required to compute over encrypted data. However, introducing randomisation as opposed to encryption requires that we have a secure means for generating fresh and cryptographically secure random numbers. The hardware on which ***P*** stores

his genome data is assumed to provide such requirements. Random number can be pre-generated and securely stored on such devices.
- Our protocol does not leak SNPs of ***P*** to the processing unit. Since the processing unit cannot distinguish the real SNPs from the dummy SNPs.
- Our protocol offers reduced storage cost to the individual. This is a result of storing encrypted data using data packing techniques.

8 IMPLEMENTATION

Here, we present the implementation of PREDICT as a prototype using basic tools. Our implementation uses simulated data rather than real dataset, since a real dataset can always be substituted whenever such data is available. We simulate ten thousand SNPs values as random numbers uniformly distributed between 0 and 2, to represent input data for the customer. The weights are equally simulated and scaled to integer values, which represent the input data of the commercial platform. The prototype of PREDICT was implemented in C++, using NTL and GMP as dependency libraries. All codes are written and executed as sequentially. Our implementation was tested on a computer with Intel Core i7-4770, 3.40 GHz, 16 GB of RAM, and 64-bit version of Ubuntu 18.04 LTS. The prototype implementation shows that PREDICT scales linearly in the size of SNP values required for a DST. As an example, computing DST using 10,000 SNP only takes about 96 *milliseconds*. In Table 3 we show comparison of our protocol with that by Ayday et al.[4].

Table 3: COMPUTATION COMPLEXITY

	Ayday et al.	**PREDICT**
Technique	Additive HE	Masking
SNP Storage	Centralized	Decentralized
Privacy Leaks	Yes	No
Performance	2 mins/10 SNPs	96 ms/10,000 SNPs

9 PRIVACY ANALYSES

Our Direct-to-Consumer DST protocol is described in the semi-honest (honest but curious) security model. Also, there is a non-collusion assumption on the entities (***P***, ***GA-Unit***, ***PU***, ***CSI***) apart from those explicitly specified within the protocol. Actually, the value π is chosen as follows: Let $\mathbb{D} = \{X_1, \ldots, X_m\}$ be the set of all diseases, and $|X_i|$ represents the number of SNPs that are associated with a disease X_i . If $\top := max\{|X_i| \ , \ \forall \ X_i \in \mathbb{D}\}$, then $\pi := \top + \kappa$.

The privacy of data is argued using simulation-based security reduction [9, 18, 25].

Definition 9.1. **Negligible function**: A function $\mu(\cdot)$ is negligible if for every positive polynomial $p(\cdot)$ and all sufficiently large $\kappa \in \mathbb{N}$, it holds that $\mu(\kappa) < 1/p(\kappa)$.

Definition 9.2. **Computational indistinguishability**: Given that $a \in \{0,1\}^*$ and κ is security parameter, let $X = X(a, \kappa)$ and $Y = Y(a, \kappa)$ be two probability ensembles. X and Y are said to be computationally indistinguishable, denoted by $X \stackrel{c}{\equiv} Y$, if for every non-uniform probabilistic polynomial-time (PPT) algorithm D , there exists a negligible function $\mu(\cdot)$ such that

$$|\Pr[D(X(a,\kappa)) = 1] - \Pr[D(Y(a,\kappa)) = 1]| \leq \mu(\kappa) \ . \qquad (11)$$

$\stackrel{s}{\equiv}$ denotes statistical indistinguishability.

Definition 9.3. **Security**: Let $f = (f_1, f_2)$ be an ideal functionality and let Π be a real-world two-party protocol for computing f. Where f_1, f_2 denote the results corresponding to parties 1 and 2 respectively on running f. The view of the party $i \in \{1,2\}$ during the execution of Π on input (a,b) and security parameter κ is denoted by $\mathbf{view}_i^{\Pi}(a,b,\kappa) := (w, r^i; m_1^i, \ldots, m_t^i)$, where $w \in (a,b)$, and r^i is the content of party i's internal random tape, and m_j^i represents the j-th message received.

The output of party i during the execution of Π on the inputs (a,b) with security parameter κ is denoted by, $\mathbf{output}_i^{\Pi}(a,b,\kappa)$ and can be computed from its own view of the execution. The joint output of both parties is denoted by

$$\mathbf{output}^{\Pi}(a,b,\kappa) = (\mathbf{output}_1^{\Pi}(a,b,\kappa), \mathbf{output}_2^{\Pi}(a,b,\kappa)) \ .$$

We say that Π securely computes f in the presence of semi-honest adversaries if there exists PPT algorithms $\mathcal{S}_1$ and $\mathcal{S}_2$ such that:

$$\{\mathcal{S}_1(1^\kappa, a, f_1(a,b)), f(a,b)\} \stackrel{c}{\equiv} \{(\mathbf{view}_1^{\Pi}(a,b,\kappa), \mathbf{output}^{\Pi}(a,b,\kappa))\} \ .$$

$$\{\mathcal{S}_2(1^\kappa, b, f_2(a,b)), f(a,b)\} \stackrel{c}{\equiv} \{(\mathbf{view}_2^{\Pi}(a,b,\kappa), \mathbf{output}^{\Pi}(a,b,\kappa))\} \ . \qquad (12)$$

Although we have provided definitions for a two-party computation, the remainder of the security proof is extended for a three-party computation without loss of generality. The ideal functionality f takes ordered inputs from ***P***, ***GA-Unit*** and ***PU*** respectively. The aim of the proof is to show that the view of a PPT adversary $\mathcal{A}$ in the real-world execution of the protocol Π, is computationally indistinguishable from the view of a simulator $\mathcal{S}_i$ for $i \in \{1,2,3\} \equiv \{\boldsymbol{P}, \boldsymbol{GA\text{-}Unit}, \boldsymbol{PU}\}$ in the ideal world execution of the protocol f. Specifically, we consider three distinct scenarios where an adversary compromises each of the parties in order to gain information about the private data of other parties.

Scenario 1: Let us assume that ***P*** has been compromised by an adversary $\mathcal{A}$. Then, $\mathcal{S}_1$ is provided with the inputs and outputs of ***P***, and is required to simulate the view:
Note that the privacy assets are the weights of SNPs to diseases, which are trade secrets and denoted as the vectors $(\mathbf{a}, \mathbf{b}, \mathbf{v})$. Refer to Figure 2 and Eq.8 for details on variables. Since ***PU*** does not contribute any input, we do not have to worry about ***PU***'s privacy. Let $\perp$ denote an empty string.

Theorem 9.4. *The DST protocol Π securely and privately computes the DST functionality $f((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}), \perp) = (S_P^X, \perp, \perp)$ in the presence of any honest-but-curious PPT adversary.*

Proof.

$$\mathbf{view}_1^{\Pi}(((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v})), \kappa) := (1^\kappa, r^P, \mathbf{k}, \mathbf{k}^2, \mathbf{R}_B, Q, \mu) \ , \qquad (13)$$

where r^P is a uniformly distributed random tape.
In order to simulate the view of ***P***, $\mathcal{S}_1$ does the following:

(1) $\mathcal{S}_1$ starts the protocol with his inputs $(\mathbf{k}, \mathbf{k}^2)$, and generates the vectors of random numbers $\mathbf{r}'_A, \mathbf{R}'_A$. Observe that the randoms are different from those generated by an honest ***P***.
(2) For Step 6: $\mathcal{S}_1$ generates the vector of randoms $\mathbf{R}'_B$, to simulate incoming input from ***GA-Unit***.
(3) For Step 13: In order to simulate Q as received from ***PU***, $\mathcal{S}_1$ first generates vectors $\mathbf{W}'_B, \mathbf{S}'_B$ such that the elements of the vector $\mathbf{W}'_B$ and $\mathbf{S}'_B$ come from the same space as elements of $\mathbf{W}_B$ and $\mathbf{S}_B$ respectively. Then, compute $\mathbf{W}'_A$ and $\mathbf{S}'_A$ as prescribed in Step 8. Finally, compute $Q' = \mathbf{W}'_A \cdot \mathbf{W}'_B - \sum_{i=1}^{2\pi} \mathbf{S}'_A - \sum_{i=1}^{2\pi} \mathbf{S}'_B$.
(4) For Step 14: $\mathcal{S}_1$ computes $\mu' = \sum_{i=1}^{2\pi} \mathbf{r}'_B - \sum_{i=1}^{\pi} \mathbf{v}'$. The vector $\mathbf{v}'$ is generate from the same space as $\mathbf{v}$.

From the above, the simulated view of $\mathcal{S}_1$ can be expressed as:

$$\mathcal{S}_1(1^\kappa, \mathbf{k}, \mathbf{k}^2, f_1((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}))) := (1^\kappa, r^P, \mathbf{k}, \mathbf{k}^2, \mathbf{R}'_B, Q', \mu') \tag{14}$$

From Equations 13 & 14, we conclude that

$$\mathcal{S}_1(1^\kappa, \mathbf{k}, \mathbf{k}^2, f_1((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}))) \stackrel{s}{\equiv} \mathbf{view}_1^{\Pi}(((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v})), \kappa)$$

For any PPT distinguisher D ,

$$Pr[D(1^\kappa, r^{S_1}, \mathbf{k}, \mathbf{k}^2, \mathbf{R}'_B, Q', \mu') = 1] -$$
$$Pr[D(1^\kappa, r^P, \mathbf{k}, \mathbf{k}^2, \mathbf{R}_B, Q, \mu) = 1] \leq \frac{1}{\mu(\kappa)} \tag{15}$$

□

Scenario 2: We assume that ***GA-Unit*** is compromised and the aim is to learn the values of the SNPs which are the vectors $\mathbf{k}, \mathbf{k}^2$.

Proof.

$$\mathbf{view}_2^{\Pi}(((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v})), \kappa) := (1^\kappa, r^G, \mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{R}_A) \tag{16}$$

where r^G is a uniformly distributed random tape. In order for $\mathcal{S}_1$ to simulate the operations of ***GA-Unit***, the following steps occur:

(1) $\mathcal{S}_2$ is provided with the inputs of ***GA-Unit*** and the disease X. These are the vectors (**a**, **b**, **v**).
(2) For Step 7: $\mathcal{S}_2$ generates a vector of random numbers $\mathbf{R}'_A$, which should be sampled from the same space as $\mathbf{R}_A$.

No other input is received by ***GA-Unit***, and this makes the proof trivial. The simulated view of $\mathcal{S}_2$ is then expressed as:

$$\mathcal{S}_2(1^\kappa, \mathbf{a}, \mathbf{b}, \mathbf{v}, f_2((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}))) := (1^\kappa, r^{S_2}, \mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{R}'_A) \tag{17}$$

From Equations 16 & 17, we have that

$$\mathcal{S}_2(1^\kappa, \mathbf{a}, \mathbf{b}, \mathbf{v}, f_2((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}))) \stackrel{s}{\equiv} \mathbf{view}_2^{\Pi}(((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v})), \kappa)$$

For any PPT distinguisher D ,

$$Pr[D(1^\kappa, r^{S_2}, \mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{R}'_A) = 1]$$
$$- Pr[D(1^\kappa, r^G, \mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{R}_A) = 1] \leq \frac{1}{\mu(\kappa)}$$

□

Scenario 3: We assume that ***PU*** is compromised by an adversary $\mathcal{A}$. The aim is to learn the private values of ***P*** and ***GA-Unit*** which include $(\mathbf{k}, \mathbf{k}^2)$, (**a, b, c**) .

Proof.

$$\mathbf{view}_3^{\Pi}(((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v})), \kappa) := (1^\kappa, r^{PU}, \perp, \mathbf{W}_A, \mathbf{W}_B, \sum_{i=1}^{2\pi} \mathbf{S}_A, \sum_{i=1}^{2\pi} \mathbf{S}_B) \tag{18}$$

where r^{PU} is a uniformly distributed random tape. For simulator $\mathcal{S}_3$ to simulate the view of ***PU***, the following steps are followed:

(1) $\mathcal{S}_3$ is provided with the security parameter κ.
(2) For Step 10: $\mathcal{S}_3$ generates the vector $\mathbf{W}'_A$ from the same space as $\mathbf{W}_A$. Then, he generates a vector $\mathbf{S}'_A$ and computes the value $\sum_{i=1}^{2\pi} \mathbf{S}'_A$.
(3) For Step 11: $\mathcal{S}_3$ generates the vector $\mathbf{W}'_B$ from the same space as $\mathbf{W}_B$. Then, he generates a vector $\mathbf{S}'_B$ and computes the value $\sum_{i=1}^{2\pi} \mathbf{S}'_B$.

The simulated view of $\mathcal{S}_3$ is then expressed as:

$$\mathcal{S}_3(1^\kappa, \perp, f_3((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}))) := (1^\kappa, r^{S_3}, \perp, \mathbf{W}'_A, \mathbf{W}'_B, \sum_{i=1}^{2\pi} \mathbf{S}'_A, \sum_{i=1}^{2\pi} \mathbf{S}'_B) \tag{19}$$

From Equations 18 & 19, we have that

$$\mathcal{S}_3(1^\kappa, \perp, f_3((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v}))) \stackrel{s}{\equiv} \mathbf{view}_3^{\Pi}(((\mathbf{k}, \mathbf{k}^2), (\mathbf{a}, \mathbf{b}, \mathbf{v})), \kappa)$$

For any PPT distinguisher D ,

$$Pr[D(1^\kappa, r^{S_3}, \perp, \mathbf{W}'_A, \mathbf{W}'_B, \sum_{i=1}^{2\pi} \mathbf{S}'_A, \sum_{i=1}^{2\pi} \mathbf{S}'_B) = 1] -$$
$$Pr[D(1^\kappa, r^{PU}, \perp, \mathbf{W}_A, \mathbf{W}_B, \sum_{i=1}^{2\pi} \mathbf{S}_A, \sum_{i=1}^{2\pi} \mathbf{S}_B) = 1] \leq \frac{1}{\mu(\kappa)} \tag{20}$$

□

10 CONCLUSION

This paper presents a protocol that blends the direct-to-consumer genetic testing model and the need to protect consumers' privacy. We have shown that a cryptographic solution to the problem is possible and implementable for practical use. Under our proposed protocol, the use of one-time-only masking is deployed to obfuscate sensitive data. We show that our proposed protocol provides security and privacy for both the genome data owners and the commercial platform while they collaborate to perform a disease susceptibility test. The design we propose introduces less work for all parties, as they are required to compute over randomized data instead of encrypted data. Our approach eliminates the storage of encrypted data on a third-party cloud infrastructure as was suggested by some earlier works. Rather, we recommend decentralizing the storage of the genome data and only allowing for storage on a device owned and controlled by genome owners. Distributing the data also eliminates a single point of failure. Our proposal allows any customer to easily update newly discovered SNPs. A prototype implementation shows that with as much as 10,000 SNPs, the DST can be computed in about 96 *milliseconds* on a commodity hardware, ignoring the network transfer time. This outperforms other existing homomorphic encryption based approaches where computational complexity is dominated by homomorphic operations. Our

proposal scales linearly in the size of the SNPs, and has shown to be practicable in the wild. Finally, we mention that our solution does not plug the analog hole. For instance, it does not protect a scenario where an attacker is able to physically force an individual to reveal his SNP values. Such an attack can be compared to an adversary retrieving a biological sample from the individual, to sequence the genome.

REFERENCES

[1] 23andMe. 2018. 23andme. https://www.23andme.com/en-eu/ Online; accessed January, 2018.

[2] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 79.

[3] Erman Ayday, Mathias Humbert, Jacques Fellay, Mc Laren, Paul Jack, Jacques Rougemont, Jean Louis Raisaro, Amalio Telenti, and Jean-Pierre Hubaux. 2012. *Protecting personal genome privacy: Solutions from information security.* Technical Report. EPFL.

[4] Erman Ayday, Jean Louis Raisaro, Jean-Pierre Hubaux, and Jacques Rougemont. 2013. Protecting and evaluating genomic privacy in medical tests and personalized medicine. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 95–106.

[5] Joppe W Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. 2013. Improved security for a ring-based fully homomorphic encryption scheme. In *IMA International Conference on Cryptography and Coding*. Springer, 45–64.

[6] Zvika Brakerski. 2018. Fundamentals of Fully Homomorphic Encryption - A Survey. *Electronic Colloquium on Computational Complexity* Rep. No. 125 (2018).

[7] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2012. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 309–325.

[8] Zvika Brakerski and Vinod Vaikuntanathan. 2014. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.* 43, 2 (2014), 831–871.

[9] Ran Canetti. 2001. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*. IEEE, 136–145.

[10] George Church. 2006. The Race for the $1000 Genome. *Science* 311 (2006).

[11] Ronald Cramer, Ivan Bjerre Damgård, et al. 2015. *Secure multiparty computation*. Cambridge University Press.

[12] George Danezis and Emiliano De Cristofaro. 2014. Simpler protocols for privacy-preserving disease susceptibility testing. In *14th Privacy Enhancing Technologies Symposium, Workshop on Genome Privacy (GenoPri?14), Amsterdam*.

[13] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2017. Manual for using homomorphic encryption for bioinformatics. *Proc. IEEE* 105, 3 (2017), 552–567.

[14] Wenliang Du and Mikhail J Atallah. 2001. Protocols for secure remote database access with approximate matching. In *E-Commerce Security and Privacy*. Springer, 87–111.

[15] Maggie Fox. July, 2018. Drug giant Glaxo teams up with DNA testing company 23andMe. https://www.nbcnews.com/health/health-news/drug-giant-glaxo-teams-dna-testing-company-23andme-n894531 Online; accessed November, 2018.

[16] Craig Gentry and Dan Boneh. 2009. *A fully homomorphic encryption scheme*. Vol. 20. Stanford University Stanford.

[17] Damien Giry. December, 2018. Cryptographic Key Length Recommendation. https://www.keylength.com/en/4/ Online; accessed December, 2018.

[18] Oded Goldreich. 1998. Secure multi-party computation. *Manuscript. Preliminary version* (1998), 86–97.

[19] Kay Hamacher. 2014. A Taxonomy of Genomic Privacy and Beyond. *1st PETS Workshop on Genome Privacy (GenoPri)* (2014).

[20] Helix. 2018. Helix. https://www.helix.com Online; accessed January, 2018.

[21] Lucia A Hindorff, Praveen Sethupathy, Heather A Junkins, Erin M Ramos, Jayashri P Mehta, Francis S Collins, and Teri A Manolio. [n. d.]. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proceedings of the National Academy of Sciences* 106, 23 ([n. d.]).

[22] Xiaoqian Jiang, Yongan Zhao, Xiaofeng Wang, Bradley Malin, Shuang Wang, Lucila Ohno-Machado, and Haixu Tang. 2014. A community assessment of privacy preserving techniques for human genomes. *BMC medical informatics and decision making* 14, Suppl 1 (2014), S1.

[23] Sekar Kathiresan, Olle Melander, Dragi Anevski, Candace Guiducci, Noël P Burtt, Charlotta Roos, Joel N Hirschhorn, Göran Berglund, Bo Hedblad, Leif Groop, et al. 2008. Polymorphisms associated with cholesterol and risk of cardiovascular events. *New England Journal of Medicine* 358, 12 (2008), 1240–1249.

[24] Michael KK Leung, Andrew Delong, Babak Alipanahi, and Brendan J Frey. 2016. Machine learning in genomic medicine: a review of computational problems and data sets. *Proc. IEEE* 104, 1 (2016), 176–197.

[25] Yehuda Lindell. 2017. How To Simulate It–A Tutorial on the Simulation Proof Technique. In *Tutorials on the Foundations of Cryptography*. Springer, 277–346.

[26] Olejnik Lukasz, Kutrowska Agnieszka, and Castelluccia Claude. 2014. I?m 2.8% Neanderthal. *1st PETS Workshop on Genome Privacy (GenoPri)* (2014).

[27] Paul J McLaren, Jean Louis Raisaro, Manel Aouri, Margalida Rotger, Erman Ayday, István Bartha, Maria B Delgado, Yannick Vallet, Huldrych F Günthard, Matthias Cavassini, et al. 2016. Privacy-preserving genomic testing in the clinic: a model using HIV treatment. *Genetics in Medicine* (2016).

[28] D McMorrow. 2010. *The $100 genome: Implications for the DoD*. Technical Report. DTIC Document.

[29] Kyriaki Michailidou, Sara Lindström, Joe Dennis, Jonathan Beesley, Shirley Hui, Siddhartha Kar, Audrey Lemaçon, Penny Soucy, Dylan Glubb, Asha Rostamianfar, et al. 2017. Association analysis identifies 65 new breast cancer risk loci. *Nature* 551, 7678 (2017), 92.

[30] MyHeritage. 2018. MyHeritage. https://www.myheritage.nl Online; accessed November, 2018.

[31] Mina Namazi, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. 2016. Dynamic Privacy-Preserving Genomic Susceptibility Testing. In *Proc. of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 45–50.

[32] NLM. 2018. https://ghr.nlm.nih.gov/primer/genomicresearch/snp. https://ghr.nlm.nih.gov/primer/genomicresearch/snp Online; accessed January, 2018.

[33] Alexandra Ossola. 2015. Gene Tests Are Quite Telling - Should You Get One? http://www.popsci.com/i-tested-my-genes Online; accessed November, 2018.

[34] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology?EUROCRYPT?99*. Springer, 223–238.

[35] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*. 223–238.

[36] pgEd. 2014. What is consumer genetics? https://www.pged.org/direct-to-consumer-genetic-testing// Online; accessed November, 2018.

[37] Andelka M Phillips. 2015. Genomic Privacy and Direct-to-Consumer Genetics: Big Consumer Genetic Data–What's in that Contract?. In *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 60–64.

[38] Spencer Prashad and Shan Srikanthan. April, 2018. 23ANDME: BUILDING A GENETICALLY-SOUND COMPANY. https://iveybusinessreview.ca/6346/23andme-building-genetically-sound-company/ Online; accessed Nov., 2018.

[39] Miriam S Reuter, Susan Walker, Bhooma Thiruvahindrapuram, Joe Whitney, Iris Cohn, Neal Sondheimer, Ryan KC Yuen, Brett Trost, Tara A Paton, Sergio L Pereira, et al. 2018. The Personal Genome Project Canada: findings from whole genome sequences of the inaugural 56 participants. *Canadian Medical Association Journal* 190, 5 (2018), E126–E136.

[40] Lewis Ricki. 2012. Direct-to-Consumer Genetic Testing: A New View. http://blogs.plos.org/dnascience/2012/11/08/direct-to-consumer-genetic-testing-a-new-view/.

[41] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. [n. d.]. On data banks and privacy homomorphisms. *Foundations of secure computation* 4, 11 ([n. d.]).

[42] Andrea Sboner, Xinmeng Jasmine Mu, Dov Greenbaum, Raymond K Auerbach, and Mark B Gerstein. 2011. The real cost of sequencing: higher than you think! *Genome biology* 12, 8 (2011), 125.

[43] David L Selwood. 2013. Beyond the Hundred Dollar Genome–Drug Discovery Futures. *Chemical biology & drug design* 81, 1 (2013), 1–4.

[44] Jay Shendure, Shankar Balasubramanian, George M Church, Walter Gilbert, Jane Rogers, Jeffery A Schloss, and Robert H Waterston. 2017. DNA sequencing at 40: past, present and future. *Nature* 550, 7676 (2017), 345.

[45] Suyash S Shringarpure and Carlos D Bustamante. 2015. Privacy risks from genomic data-sharing beacons. *The American Journal of Human Genetics* 97, 5 (2015), 631–646.

[46] Pascal Su. 2013. Direct-to-consumer genetic testing: a comprehensive view. *The Yale journal of biology and medicine* 86, 3 (2013), 359.

[47] Alun Thomas, Nicola J Camp, James M Farnham, Kristina Allen-Brady, and Lisa A Cannon-Albright. 2008. Shared genomic segment analysis. Mapping disease predisposition genes in extended pedigrees using SNP genotype assays. *Annals of human genetics* 72, 2 (2008), 279–287.

[48] Chibuike Ugwuoke, Zekeriya Erkin, and Reginald L Lagendijk. 2017. Privacy-safe linkage analysis with homomorphic encryption. In *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 961–965.

[49] Xiang Wan, Can Yang, Qiang Yang, Hong Xue, Nelson LS Tang, and Weichuan Yu. 2009. MegaSNPHunter: a learning approach to detect disease predisposition SNPs and high level interactions in genome wide association study. *BMC bioinformatics* 10, 1 (2009), 1.

[50] Minro Watanabe. 1998. Polymorphic CYP genes and disease predisposition?what have the studies shown so far? *Toxicology letters* 102 (1998), 167–171.

[51] Kris A Wetterstrand. 2016. DNA sequencing costs: data from the NHGRI Genome sequencing program (GSP). 2013. *URL http://www. genome. gov/sequencingcosts* (2016).

[52] Jean E Wylie and Geraldine P Mineau. 2003. Biomedical databases: protecting privacy and promoting research. *Trends in biotechnology* 21, 3 (2003), 113–116.