Invited Paper

Towards Privacy-preserving Authenticated Disease Risk Queries

Nusrat Jahan Mozumder[1,a)] Maitraye Das[2,b)] Tanzima Hashem[1,c)] Sharmin Afrose[3,d)]
Khandakar Ashrafi Akbar[1,e)]

Received: January 15, 2019, Accepted: June 26, 2019

Abstract: Recent improvement in genomic research is paving the way towards significant progress in diagnosis and treatment of diseases. A disease risk query returns the probability of a patient to develop a particular disease based on her genomic and clinical data. Despite various innovative prospects, frequent and ubiquitous usage of genomic data in medical tests and personalized medicine may cause various privacy threats like genetic discrimination, exposure of susceptibility to diseases, and revelation of genomic data of relatives. Another major concern is on ensuring the reliability of the genome data and the correctness of the computed disease risk, which is known as authentication. We develop a novel secret sharing approach to protect privacy of sensitive genomic and clinical data, disease markers, disease name, and the query answer while ensuring authenticated result of the disease risk query. In addition, we discuss the applicability of our approach in the field of personalized medicine. We perform a comprehensive security analysis for our system. Experiments with real datasets show that our approach for authenticated disease risk queries achieves a high level of privacy with reduced processing and storage overhead.

Keywords: genomic privacy, secret sharing, authenticated disease risk queries, personalized medicine

1. Introduction

Rapid advancement of efficient and cost-effective genome sequencing has opened the door for various novel research directions in genomics. In recent years, researchers have focused on revealing the correlation between genetic variants and an individual's predisposition to diseases or response to the treatment. Thus, genomic data has become popular for early diagnosis and proper treatment of diseases [56]. For example, people having family history of HIV, cancer, leukemia, heart disease, or diabetes may want to measure the risk of inheriting these diseases in advance so that proper diet and preventive treatment can be adopted [6], [7]. Besides, accurate dosage of medicine can also be suggested according to patients' genetic makeup [38]. In this way, standard medical tests are taking turns towards a more personalized route [27].

With this pervasive usage of genomic data in personalized medicine, privacy of an individual is going through potential risks, as genomic data may reveal sensitive information regarding an individual's ethnicity, ancestry, phenotypic traits, health conditions and susceptibility to specific diseases [24]. Thus, the leakage of genetic information may result in genetic discrimination in the sector of health insurance, employment and overall social dynamics [8]. In addition, a person's genomic data can reveal sensitive information of the person's close relatives (possibly without their consents) due to hereditary nature of genome [37]. Therefore, to continue the growth of revolutionary applications on genomes, privacy protection is essential. We focus on protecting privacy of genome data while processing a *disease risk query*, i.e., the probability of an individual to develop a specific disease.

Currently, a heavy layer of access control and legislation is applied for processing genomes in medical centers. However, rules and legislation are based on trust and cannot control malicious attacks on genomes as they may not anticipate the technological advancement. Identity anonymization is also ineffective for storing genomic data, because genome sequence is the unique and irrevocable identifier of its owner [41]. In addition, inaccuracy or absence of valuable genomic data ensuing from obfuscation techniques might cause misleading results for disease risk queries.

Besides protecting privacy of genome data, another challenge is to authenticate a disease risk query. Processing a person's disease risk query involves outside entities, and thus raises concerns on the reliability of the genome data used for a disease risk query and the correctness of the computed disease risk. Authentication ensures that the disease risk is correctly computed using a person's actual genome data. We develop a novel secret sharing approach for privacy-preserving authenticated disease risk queries.

Gene sequencing is done by a *certified* Sequencing Institute (SI) [11], [18], [19], [20], [34], which may be directed by the government or any trusted party. In our approach, the SI distributes SNPs [1] (Single Nucleotide Polymorphism) of genome data among several authorized Distributed Databases (DDBs),

1 Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
2 Technology and Social Behavior, Northwestern University, Evanston, IL, USA
3 Department of Computer Science, Virginia Tech, Blacksburg, VA, USA
a) deeptee.cse12@gmail.com
b) maitraye@u.northwestern.edu
c) tanzimahashem@cse.buet.ac.bd
d) sharminafrose@vt.edu
e) aninditaashrafi@gmail.com

where one DDB is located at the patient's device. The key idea of our approach is that SNPs remain hidden in an aggregate form, and the probability to develop a specific disease is computed by combining partial genetic scores for the specific disease from all the DDBs. If a dishonest DDB alters a patient's SNP data and provides a wrong partial genetic score, then our authentication technique can detect the alteration using an authentication key generated based on the stored SNP data at the DDB and thus verify the correctness of the computed disease risk. Additionally, we show that not a single SNP of a patient can be identified without involving the patient even if all the DDBs become compromised. The portion of data that our approach stores on a patient's device does not cause any significant overhead in terms of the storage size. On the other hand, our approach does not store the full data on a patient's device to ensure that it is also not possible to identify a patient's SNP from the patient's DDB without compromising the other DDBs.

Over the last years, though researchers have developed a few cryptographic approaches for privacy-preserving disease risk queries [11], [14], [18], [19], [20], [34], these approaches cannot authenticate the query answer. Another major limitation of these techniques is that they cannot answer a disease risk query accurately when different alleles of the same SNP in genomes are responsible for two or more different diseases. For example, allele C of SNP rs6313 holds higher risk for rheumatoid arthritis, whereas allele T of the same SNP contributes to depression, panic and stress response [2]. Specifically, existing approaches [11], [18], [19], [20], [34] store the frequency of one allele for an SNP (considering that this allele is always responsible for diseases) in encrypted form in a single Data Center (DC). The DC can only *partially decrypt* the frequency information when it receives a disease risk query from a Medical Unit (MU). Though all common SNPs have two possible allele variations, it is not possible for the DC to infer the frequency of the other allele in the SNP from the partially decrypted frequency of one allele. The DC sends *encrypted* frequency information (not the partially decrypted ones) to the MU. The MU can also *partially decrypt* the frequency information and thus, cannot infer the frequency of the other allele. Only possible way to infer the frequencies of both alleles is the collusion of the MU and the DC, which is not allowed since the collusion will eventually reveal the genome data to both parties and violate user privacy. These two limitations are not possible to overcome by any trivial computation.

Our approach ensures privacy of genomes even if the dishonest MU and the DDBs collude and can evaluate disease risk queries when two alleles of the same SNP are responsible for two or more different diseases. Though a recent cryptographic approach [22], Turkmen et al. have authenticated computed disease risks, this approach has not considered storing both alleles or a dishonest medical unit. More importantly, the authors have not performed any experiment to validate the performance of their approach.

Besides, existing approaches incur high storage overhead due to encryption and the overhead would be doubled if the encrypted frequencies of both alleles of an SNP are stored. Nowadays, the provision of low cost genome sequencing is attracting an increasing number of people to use disease risk queries. The number of SNPs responsible for different diseases is also increasing. Thus, reducing the storage overhead has become an important challenge for any privacy preserving approach for disease risk queries. Our approach offers a substantial improvement in reducing storage cost with a large margin.

A disease risk query for a patient is processed using an MU's disease marker that consists of the SNPs associated with a particular disease, their risk alleles (i.e., which one of the two possible alleles of each SNP is responsible for this particular disease), and contribution factors of risk alleles. Though the SNPs associated with a particular disease and risk alleles are publicly known, it may happen that an MU wants to keep contribution factors of risk alleles confidential from others. On the other hand, it is possible to infer the name of a disease from the publicly available contents of the disease marker, i.e., the SNPs associated with a particular disease and risk alleles. However, a patient may not feel comfortable to disclose the disease name such as Alzheimer's to any party except the MU for treatment purposes. Thus, it is also essential to hide the number and IDs of SNPs and their risk alleles used in the disease risk query to protect privacy of the disease name. Our approach ensures the privacy of a patient's genomic and clinical data, an MU's disease marker, disease name, and the query answer, i.e., the probability of a patient to develop a particular disease.

To the best of our knowledge, we develop the first secret sharing approach for privacy preserving authenticated disease risk queries that eliminates the cryptographic overheads for processing encrypted genomes. In summary, the contributions of our paper are as follows:

- We propose a novel secret sharing approach to privately compute the probability of a patient to develop a specific disease without revealing genomic data, clinical data, the disease name and the query answer to others.
- We ensure that our proposed technique can evaluate disease risk queries when different alleles of the same SNP are responsible for different diseases.
- We authenticate the query results sent by dishonest DDBs to ensure the correctness of the disease risk.
- We protect privacy of genome data against the dishonest MU and its collusion with the DDBs.
- We provide a solution to hide the MU's disease markers from others.
- We present a comprehensive security analysis for our system. We show the effectiveness of our proposed approach via extensive experiments using real human genome datasets. Our approach reduces the storage overhead significantly.
- We investigate the possibility of our approach for personalized medicine.

This paper extends our previous work [36], where we proposed a novel secret sharing approach for privacy preserving authenticated disease risk queries. In this paper, we extend our work in the following ways: (i) we extend the related work section and classify privacy preserving techniques for disease risk queries, personalized medicine and other genetic tests based on different features, (ii) we theoretically show that our system is secured from dishonest entities and malicious attacks and (iii) we discuss the

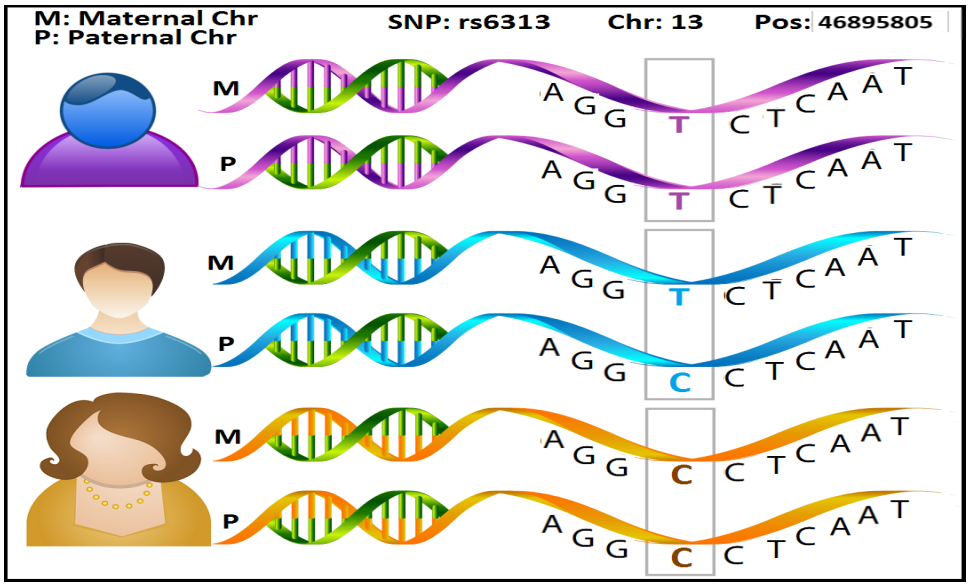


Fig. 1 DNA fragments of three different persons showing SNP rs6313.

possible modifications for applying our approach in the field of personalized medicine.

The rest of the paper is organized as follows: Section 2 describes basic concepts and Section 3 discuses existing work related to our approach. Section 4 gives a brief overview of our system. We show the steps of storing genomic data, preserving privacy and computing authenticated disease risk queries in Section 5. We prove the correctness of the result computed by our system in Section 6 and perform the security analysis in Section 7. We present the results of our experiments using real datasets in Section 8. Section 9 discuses the applicability of our approach in the field of personalized medicine. Finally, in Section 10, we conclude our work with the future research direction.

2. Preliminaries

2.1 Genomic Background

The genetic material of an organism is encoded in DNA (Deoxyribonucleic Acid) or in RNA (Ribonucleic Acid), for many viruses. DNA is a double stranded molecule consisting of two long and complimentary polymer chains of four nucleotides, such as Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). The two separate polynucleotide chains are bound by hydrogen bonds according to the base pairing rule (A with T and C with G) to make the double stranded DNA. Within cell nucleus, DNA is arranged in long structures called chromosomes. Some organisms have single copy of chromosomes (haploid), while some have multiple copies (diploid, triploid, tetraploid etc.). Human diploid cells have 23 pairs of chromosomes, of which, one set of 23 chromosomes comes from each parent.

Between any two given individuals, around 99.9% of the entire genome is same [30]. The remaining 0.1% part is responsible for many of our distinguishable characteristics. Single Nucleotide Polymorphism (SNP) is the most common form of human genetic variations occurring within at least 1% of the individuals in a population. If a single nucleotide in the DNA differs between members of the same species or paired polymer chains of an individual, the variation is called an SNP [1]. For example, DNA fragments $CTGG$ and $CCGG$ differ in a single nucleotide at the underlined position.

Each individual carries two alleles (i.e., two nucleotides) at each position in the chromosome; one inherited from the mother and one from the father. Besides, almost all common SNPs have only two variations (i.e., nucleotides) among A, C, G, and T. **Figure 1** shows DNA fragments from three individuals highlighting SNP rs6313 located at position 46895805 in chromosome 13. SNP rs6313 has two variations C and T [2]. Other two variations A and G are not seen in this SNP.

SNPs are often associated with various phenotypic traits (curly hair, attached earlobes etc.), complex social behavior (reckless driving, marital infidelity etc.) as well as proneness and receptivity to diseases and response to drugs. Each SNP has a different impact on a particular disease risk; some of them are responsible for the development of the disease whereas some are defensive. Generally, for an SNP associated with a particular disease, one of the alleles carries the risk and other does not. Furthermore, it is possible that both the alleles of a particular SNP carries risk for two different diseases. For example, allele C of SNP rs6313, holds higher risk for Rheumatoid Arthritis, whereas allele T of this SNP contributes to depression, panic and stress response [2].

Let f_i denote the number of risk allele r_i in the SNP rs6313, where $f_i \in \{0, 1, 2\}$. That means, if C is the risk allele for any particular disease, f_i of three patients P1, P2 and P3 having genotype CC, CT and TT respectively, are 2, 1 and 0. However, f_i of patients P1, P2 and P3 are 0, 1 and 2, respectively, if T is the risk allele for another disease.

2.2 Contribution of Clinical Data in Disease Risk

Along with genomic data, clinical factors of an individual can contribute significantly to her disease risks, especially for chronic diseases like Coronary Artery Disease, Diabetes etc. The clinical factors can include demographic information (e.g., age, sex etc.), his family history of diseases, laboratory test results (e.g., choles-

terol level, blood sugar level etc.). For this reason, clinical data should also be considered along with genomic data in the computation of the disease risks of an individual [48].

2.3 Computation of Disease Risk

The probability to develop a disease for an individual is computed based on her genomic and clinical data. In practice, multiple SNPs are responsible for a disease and they contribute to the disease to different extents. The contribution factor of an SNP S_i is defined as $\beta_i = ln(OR_i)$, where odds ratio (OR_i) represents the extent to which an SNP S_i is associated with a certain disease. In a particular group of individuals, *odds* is the ratio of instance of the disease to that of its non-instance. Hence, OR is the ratio of *odds* in that group of individuals having a genetic variation to that of those who do not possess it. If we consider a group of 10 people affected by a disease X, among whom 6 are carrying a particular variation Y and the rest are not, then *odds* of disease X for the individuals having variation Y is 6/4, i.e., 1.5. Similarly, in a group of 10 people without disease X (2 with variation Y and 8 without variation Y), *odds* of disease X for the individuals not having variation Y is 2/8 i.e., 0.25. Thus, OR of variation Y for disease X is 1.5/.25 i.e., 6. Odds ratio of different SNPs corresponding to a particular disease can be found from the results of Genome-Wide-Association Studies (GWAS).

Following recent approaches [11], [20], [34], we use a multivariable logistic regression model to calculate the disease risk by weighted averaging contribution factors of all the associated SNPs and clinical data. Let λ be the number of SNPs associated with disease X, β_i be the contribution factor of S_i, and f_i be the number of risk allele for S_i. Let ϕ be the number of clinical factors associated with disease X, $\bar{\beta}_i$ be the contribution factor of the clinical data C_i and v_i be the value of C_i. We let $v_i(X) \in \{0, 1\}$ for simplicity of representation. For example, let v_i denotes smoking behavior. Thus, if a patient is a smoker, $v_i = 1$ and $v_i = 0$, if he is not. Similarly, if the clinical factors are results of various tests (glucose level) or demographic data (age etc.), those attributes can also be easily converted to binary form (e.g., whether age > 50 or not). Let Pr be the probability of a patient P to develop disease X and Z be the total disease score. We have,

$$Z = ln\left(\frac{Pr}{1-Pr}\right) = \sum_{i=1}^{\lambda} \beta_i \times f_i + \sum_{i=1}^{\phi} \bar{\beta}_i \times v_i$$
$$\Rightarrow Pr = \frac{e^Z}{1+e^Z} \quad (1)$$

3. Related Work

In this section, we discuss various technological solutions for preserving privacy of genomic data used in disease risk queries, personalized medicine and other genetic tests. Section 3.1 discuses existing methods that compute the probability of having a disease considering genetic variations. Before considering genetic variation, string searching in small DNA fragments was regarded as a medium for calculating the probability of a disease. These methods are briefly described in Section 3.2. DNA string sequences are also used in other medical fields such as finding common things between two individuals. Section 3.3 describes the privacy preserving architectures that use string sequence comparison in genomic testing. Section 3.4 mentions cryptographic approaches to ensure genomic data privacy in genomic computations. Section 3.5 discuses existing secret sharing methods for genomic data privacy. Finally, Section 3.6 shows privacy preserving techniques for genomes based on other privacy models.

3.1 Privacy Preserving Techniques for Disease Risk Queries with Multiple Genetic Variants

In recent years, researchers focused on protecting privacy of genomic data while computing the probability of developing a particular disease. In Ref. [19], Ayday et al. proposed privacy-preserving disease risk queries using modified paillier cryptosystem and proxy re-encryption. In Ref. [20], Ayday et al. considered clinical data in addition to genomic data for evaluating a disease risk query. In Ref. [11], Danezis et al. identified that it is possible to infer disease name from the IDs and number of SNPs used in a disease risk query and developed solutions to overcome this attack. In Ref. [34], Barman et al. proposed countermeasures to genomic data retrieval attack by dishonest-but-covert medical unit based on the architecture of Refs. [19], [20]. Using techniques similar to Ref. [19], Turkmen et al. used message authentication code and verifiable computing to check correctness of disease susceptibility tests in Ref. [22]. All of these approaches store data in encrypted form in a semi-honest data center and require high computing power and storage facility [8]. On the contrary, we develop a secret sharing approach that does not need to store encrypted genomic data. We also provide necessary authentication measures considering dishonest databases and medical unit. Furthermore, these approaches assume that storage and medical units never collude and also fail to give the correct answer when two alleles of the same SNP are responsible for two or more different diseases. These limitations have been addressed in our approach.

These privacy-preserving techniques [11], [19] along with ours, DA are summarized in **Table 1** on five features: a) privacy protection, b) addressing the scenario when different alleles of the same SNP are responsible for two or more different diseases, c) authentication d) storage overhead, and e) communication overhead. In terms of privacy protection, five types of information are considered as sensitive: i) patient's genomic data (considering individual attacks by storage and medical centers as well as collusion between them), ii) patient's clinical data iii) the name of the disease being tested, iv) disease risk query answer, and v) contribution factors of risk alleles of the associated SNPs. Only our proposed algorithm, DA shows satisfactory outcome for all these features. We note that other cryptographic approaches [18], [34] maintain the generic architecture of Ref. [19] and thus show similar performance for these features.

3.2 Privacy Preserving String Searching Techniques

All of the approaches of Section 3.1 including ours consider the impact of multiple genetic variants, i.e., SNPs to compute the probability to develop a particular disease. However, earlier approaches used private string searching and comparison techniques that used small DNA fragments for disease susceptibility testing. In this approach, a medical unit (e.g., physician, hospital or phar-

Table 1 Comparison of different techniques.

Features		A1 [19]	A2 [11]	A3 [20]	A4 [22]	DA
Privacy	Genomic data	✗	✗	✗	✗	✓
	Clinical data	✗	✗	✓	✗	✓
	Disease name	✗	✓	✗	✗	✓
	Query answer	✓	✓	✓	✓	✓
	Contribution factors	✓	✓	✓	✓	✓
Different alleles' association		✗	✗	✗	✗	✓
Authentication		✗	✗	✗	✓	✓
Storage cost		high	low	high	N/A	low
Communication cost		medium	high	medium	N/A	low

maceutical company) has DNA markers (i.e., substrings that describe mutation leading to a disease) and a patient has a digitized genome (DNA snippet). The patient can verify whether the DNA markers are present in her genome, while not revealing personal genomic data to the medical unit and not learning anything about the DNA markers as well.

Several privacy preserving string matching schemes have been proposed based on Finite State Machine in an oblivious manner [9], [25], [31]. These approaches ensure error resilient searching by representing the DNA marker as a finite automaton and evaluating it on genome sequence of the patient which is treated as input. De Cristofaro et al. [14] proposed a pattern matching approach that not only preserves privacy of the DNA sequence and DNA markers, but also hides the size and position of the markers in the genome so that an adversary cannot infer test specifics.

3.3 Privacy Preserving Sequence Comparison Techniques

Researchers also studied sequence comparison techniques for genomic testing, where two entities want to detect whether two genomes are closely related or not, but does not want to reveal the genomes to each other. Wang et al. [49] proposed a distributed framework for privacy preserving sequence comparison in which they issued sensitive data to the data provider and public data to the data consumer. Genomic computation is partitioned through program specialization that enables data consumer to compute over the genomic sequences sanitized by the data provider, i.e., sensitive data replaced with symbols.

In Refs. [15], [16], Eppstein et al. proposed the use of a Privacy-enhanced Invertible Bloom Filter (PIBF) for comparing two compressed DNA sequences under various querying scenarios. In particular, they considered scenarios where a querier, Bob, wants to test if her DNA string, Q, is close to a DNA string, Y, owned by a data owner, Alice. However, Bob does not want to reveal Q to Alice and Alice is willing to reveal Y to Bob only if it is close to Q. In another scenario, results of the query is published only to a trusted third party, Charles. Solution of Eppstein et al. achieved absolute privacy for Bob (in information theoretic sense) and a quantifiable degree of privacy protection for Alice.

In Ref. [21], raw genomic data is stored, retrieved and processed in a way that allows a medical entity to privately recover a subset of the genome (short reads) while not revealing the test to the genome data centre or biobank. In Ref. [57], Chen et al. presented a secure and effective algorithm to align short DNA sequences to a reference (human) DNA sequence (i.e., read mapping) utilizing a hybrid cloud which includes both the public commercial cloud and the private cloud within an organization.

3.4 Cryptographic Techniques

In the context of genomic data privacy, researchers have applied several cryptographic solutions over the ages. In Ref. [53], Jha et al. presented a privacy-preserving implementation of fundamental genomic computations such as calculating the edit distance and Smith-Waterman similarity scores between two strings (DNA sequences) based on Yao's 'garbled circuits' method. In Ref. [44], Bohannon et al. suggested searchable genetic databases for forensic purposes using a fuzzy encryption scheme that allows searching an identity based on genome data but not the genome data based on identity.

Baldi et al. [13], [46] constructed solutions for paternity tests, personalized medicine and genetic compatibility tests based on well-known cryptographic tools, such as Private Set Intersection, Private Set Intersection Cardinality, and Authorized Private Set Intersection. Although these techniques performed well on small DNA fragments, it needed days of computation and gigabytes of bandwidth to examine the whole genome. Bruekers et al. [23] used homomorphic encryption for secure matching of Short Tandem Repeat (STR) profiles between two parties without exposing the actual genomic data in case of DNA-based identity, paternity and ancestry tests. The authors of Refs. [29], [42], [43], [50] also used homomorphic encryption to perform scientific investigations.

One issue associated with private string matching, sequence comparison and cryptographic techniques is that they do not take into account multiple genetic variants, i.e., SNPs and thus, cannot produce accurate answer for a disease risk query when multiple SNPs and clinical data are responsible for the disease [48].

3.5 Secret-sharing Techniques

It is shown in Ref. [55] that secret sharing techniques are more efficient than encryption-based techniques for privacy-preserving data mining with respect to communication, computation and storage cost. Secure multi-party computation-based secret sharing techniques have been used to protect privacy in evaluating count and ranked queries [39] and in GWAS (Genome-Wide Association Studies) [32], [51], [58]. In Ref. [32], Kamm et al. deployed a secret sharing scheme where each data center divides its collection into small shares, each of which reveals nothing about the original values. The shares are then sent to the other centers,

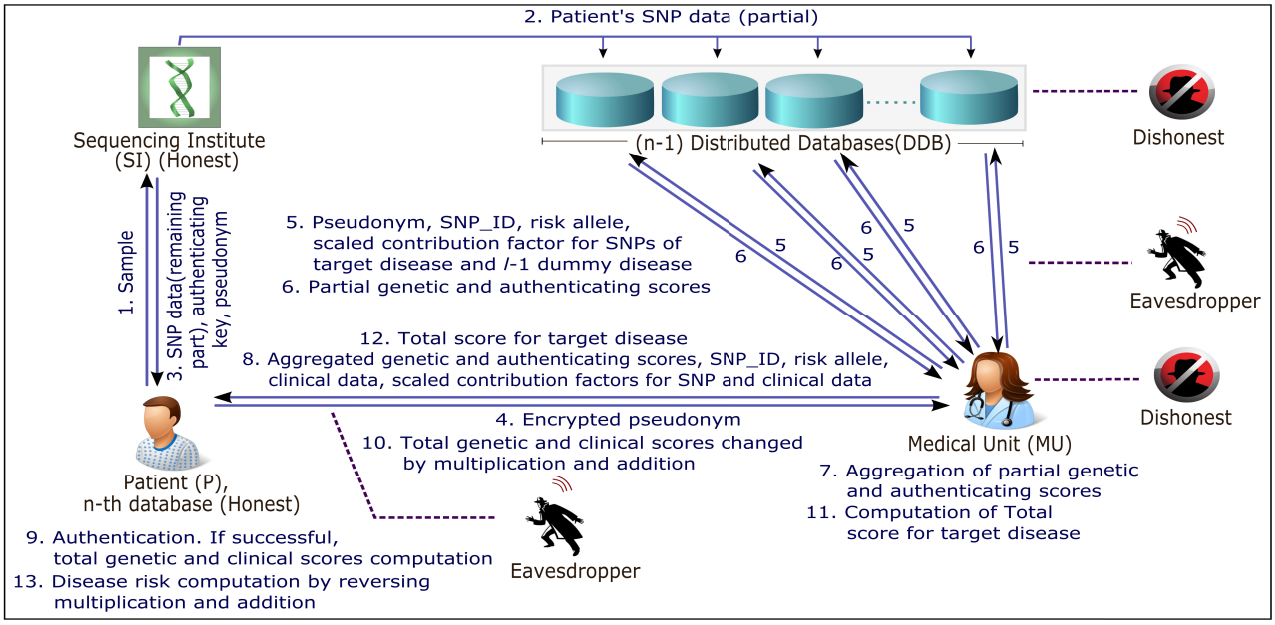


Fig. 2 System architecture of our secret sharing approach for privacy-preserving authenticated disease risk query.

which store them in dedicated servers. The servers have an interface that allows outsiders to initiate a GWAS study on genomic data of interest. Upon request, the servers coordinate to perform association without reconstructing the original genomic data. To the best of our knowledge, no previous work adopts secret sharing techniques or naïve bit encoding [52] (the encoding that we used) to protect genomic privacy for disease risk queries.

3.6 Other Privacy Preserving Techniques

In addition to cryptography and secret sharing, other privacy concepts like access control, obfuscation and anonymization have been studied for privacy preserved storage of genomic data. However, these methods have limitations. The main drawback of access control is that it requires constant management of the resources and creates administrative burden to both users and data custodians. Obfuscation has also become obsolete as it reduces the accuracy of genomic data [17].

Anonymization techniques remove explicit identifiers such as name or social security number from the genomic data. However, they fail to protect privacy, as genomic data itself can be the identifier of an individual [41]. In Ref. [54], Fienberg et al. used differential privacy to release test statistics of GWAS results. In contrast to k-anonymity, differential privacy guarantees privacy against an adversary with arbitrary prior knowledge by adding noise to the results before their release.

4. System Overview

Like existing systems [11], [18], [19], [20], [34], a trusted sequencing institute (SI) performs the sequencing of genomic data of a patient (P). P provides her sample (e.g., hair, saliva etc.) to the SI for genome sequencing. The SI distributes the SNPs of P and relevant information for authentication of genomic data among n independent databases (DDBs). We assume that the DDBs are run by separate authorities such as private companies, cloud storage services or non-profit organizations under the supervision of the government. The n^{th} DDB is stored in the patient's personal computer or mobile device. The SI sends data to all the DDBs except the n^{th} DDB in plain format. On the other hand, the SI encrypts genomic data and authentication key using TDES [40] before sending them to the patient and the patient decrypts the data before storing them to the n^{th} DDB. At this point, it may be argued that the SNP contents could be stored as a whole in the patient's device instead of n separate DDBs. However, patient's device can easily be hacked or stolen leaving the genomic data in risk. In our system, we ensure that even if the patient's device is hacked, genomic data is secure, unless other $n-2$ DDBs are also compromised (Section 5.2). The system architecture is shown in **Fig. 2**.

A Medical Unit (MU) normally located at a health center, has the IDs of SNPs and clinical data responsible for various diseases, risk allele and contribution factor corresponding to each SNP or clinical factor. A pseudonym is assigned to each patient at the time of gene sequencing and used to store genomic data in the DDBs to hide the identity of a patient from adversaries. When a patient P wants to know her probability of developing a particular disease X, P sends her encrypted pseudonym to the MU. The MU decrypts the received data using TDES. We use TDES only for securing communication of the SI and the MU with P. One may argue that revealing the patient's pseudonym to the MU may raise a concern, because using the pseudonym the MU can send multiple queries to know the probabilities to develop multiple diseases for P. We explain in Section 7.1 that even if the MU generates multiple queries to the DDBs without the consent of a patient, the probabilities to develop multiple diseases are not revealed to the MU if the patient's DDB does not contribute to the queries. As such, the shared part of SNP data from the patient's DDB serves as the validation of the query generated by the MU.

The MU sends P's pseudonym, the IDs of relevant SNPs, their

risk alleles, and scaled contribution factors responsible for developing disease X and other randomly selected $l-1$ dummy diseases to all $n-1$ DDBs except the patient's device and makes X indistinguishable from l diseases (details in Section 5.3.1). The MU scales the contribution factors of the SNPs (β) by random constants to ensure that original β values cannot be inferred by the DDBs. Each DDB computes partial genetic and authenticating scores using P's genomic data stored in the database and scaled β values received from the MU, and sends back the partial scores to the MU. The MU separately sums up partial genetic and authenticating scores sent from $n-1$ DDBs. Along with these aggregated genetic and authenticating scores, SNPs of all the l diseases, their risk alleles and scaled contribution factors, the MU sends clinical data related to l diseases and their contribution factors scaled by random constants to the n^{th} DDB at the patient's device. Patient P verifies the correctness of the aggregated genetic scores using the authenticating scores sent by the MU and the authentication key stored in its database. A patient can detect if other $n-1$ DDBs or the MU alter the genomic data (see Theorem 6.2). After authenticating the aggregated genetic scores, P calculates the total genetic and clinical scores, modifies these scores using multiplication and addition operations, and sends to the MU. The MU first scales back the genetic and clinical scores of the target disease X, and then sends the combined score to P. Finally, P accurately computes the disease risk probability by reversing the effect of previous multiplication and addition operations (see Theorem 6.1).

We involve the patient to make sure that not a single SNP is disclosed to anyone without the consent of the patient even if the other DDBs along with the MU are compromised. One may argue that a patient may not agree to take the burden of authentication and storage. We note that our approach is also applicable if a patient does not store the n^{th} DDB, i.e., the n^{th} DDB is run by a separate authority like other $n-1$ DDBs. However, in this case, the patient's privacy is slightly reduced; SNPs of a patient can be identified and authentication process can fail if an adversary compromises $n-1$ DDBs (including the n^{th} DDB).

5. Our Approach

The key idea of our approach is to distribute SNPs and disease risk computations among multiple databases such that SNPs and other sensitive information are not revealed to any involved party or eavesdroppers. We discuss the steps of our approach in the following subsections.

5.1 Gene Sequencing

A patient (P) provides her sample, e.g., saliva, hair etc. to the SI. The SI sequences the sample and extracts SNPs from the raw genomic data. A pseudonym and an authentication key μ for P are generated and given to P, where μ is a constant. The pseudonym is used instead of P's actual name and identity to store her genomic data in the DDBs.

5.2 Storing Data in the Distributed Databases

SNPs are stored in n independent databases and all databases (DDBs) collectively give the actual SNP contents. Each SNP has a unique position and a unique ID and almost all common SNPs have only two probable nucleotides among A, C, G, and T for each of the two alleles. For example, SNP rs6313 has two variations C and T [2]. Other two variations A and G are not possible in SNP rs6313.

Table 2 Sample entries for SNP S_1, $k = DDB$ No.

k		$a_{1,k}$	$w_{1,k}$	$\alpha_{1,k}$	$a_{2,k}$	$w_{2,k}$	$\alpha_{2,k}$
1		01	−12	−1	11	62	2
2		11	50	7	11	−2	2
3		01	0	−8	01	46	8
4		11	−2	−8	01	8	0
5	$t=1$	11	−48	9	11	−59	6
	$t=2$	01	13	7	01	−54	1

Each DDB stores nucleotides of two alleles of an SNP separately using naïve bit encoding as a bit string of length 2 (00, 01, 10, and 11 representing A, C, G, and T, respectively). The actual nucleotide of each allele of an SNP is stored on a randomly selected m DDBs, where $m \leq n-2$ and the other possible nucleotide on the remaining $(n-m-1)$ DDBs. In the n^{th} DDB located at the patient's device, we store both possible nucleotides (e.g., C and T). For each allele of an SNP, we also store a random weight such that the summation of weights from all the DDBs for the true nucleotide of the allele becomes 1 and the false one becomes 0. We make the sum of weight of the false allele 0 to nullify its impact on the disease risk computation. Neither of the total weights (i.e., 1 or 0) can be inferred unless the patient's DDB is stolen and other $n-2$ DDBs are compromised. On the other hand, though all $n-1$ DDBs store the same pseudonym for a single patient as the primary key, it is not possible to predict the total weight of an allele without knowing the weights stored in the n^{th} DDB at the patient's device even if all $n-1$ DDBs collude.

For authentication purpose, each DDB stores another value α for each allele of an SNP such that the summation of weights of that allele from all the $n-1$ DDBs equals the summation of α for that allele from all n DDBs including the patient's device scaled by the authentication key, μ.

Let $a_{j,k}$, $w_{j,k}$ and $\alpha_{j,k}$ denote the j^{th} allele of an SNP in the k^{th} DDB, its weight, and corresponding α value assigned to it, respectively. We note that $j \in \{1,2\}$ as each SNP has two alleles, $k \in \{1,2,\ldots,n-1\}$, $a_{j,k} \in \{00,01,10,11\}$ and $-100 < w_{j,k}, \alpha_{j,k} < 100$. The n^{th} DDB does not have the pseudonym but stores two possible nucleotides of each allele $a_{j,n,t}$ for $t \in \{1,2\}$, corresponding weights $w_{j,k,t}$, and authenticating values $\alpha_{j,k,t}$ for an SNP.

Consider SNP S_1 has two variations C and T and patient P has CT in her genome for S_1. **Table 2** shows a possible distribution of weights in 5 DDBs. For the 1st allele, we have 01 in $a_{1,1}$, $a_{1,3}$ and $a_{1,5,2}$, and 11 in $a_{1,2}$, $a_{1,4}$ and $a_{1,5,1}$. The weight of 01 is

$$w_{1,1} + w_{1,3} + w_{1,5,2} = 1$$

On the other hand, the weight of 11 is

$$w_{1,2} + w_{1,4} + w_{1,5,1} = 0$$

Thus, 01 (i.e., C) is true content of the first allele. Similarly, we can see that 11 (i.e., T) is true content of the other allele. If C is the risk allele of S_1 for any disease X, then in this example, the total number of risk allele in S_1 is, $f_1 = 1$.

Let authentication key, $\mu = 6$. We can see that for allele 11,

$$w_{1,2} + w_{1,4} = (\alpha_{1,2} + \alpha_{1,4} + \alpha_{1,5,1}) \times 6 = 48$$

5.3 Computation of Disease Risk

We describe the detailed process of computing the disease risk in six subsequent steps.

5.3.1 Query Processing at the MU

After receiving pseudonym from P, the MU generates a query message with necessary information for calculating the genetic score of P and sends this message to all DDBs except the n^{th} DDB. To hide the identity of the target disease, X from a curious party at the DDBs or eavesdroppers, the MU chooses $l-1$ distinct dummy diseases $(Y_1, Y_2, \ldots, Y_{l-1})$ from different types of disease groups other than disease X, so that the protection provided to the patient is not mitigated. For example, if breast cancer is the target disease, the dummy diseases will be chosen such that they are not different types of cancers. Otherwise, the DDBs might conclude that the patient has some kind of cancer.

Next, IDs of SNPs associated with all the l diseases are retrieved with their corresponding risk alleles. Let $\mathbb{P}(X), \mathbb{P}(Y_1), \mathbb{P}(Y_2), \ldots, \mathbb{P}(Y_{l-1})$ be the sets of SNPs related to target disease X and dummy diseases $Y_1, Y_2, \ldots, Y_{l-1}$, respectively. The MU also retrieves the contribution factors of the SNPs related to the target disease, X from its database. For the SNPs of the dummy diseases, random values are generated as contribution factor, β. To hide the contribution factors from the adversaries, the MU scales the β_i value of each SNP, S_i belonging to the j^{th} disease set in the query message using a randomly generated constant c_j, where $j \in \{1, \ldots, l\}$. The MU does not disclose the value of c_j to others. Let the scaled β_i value of each SNP S_i be ε_i, such that $\varepsilon_i = \beta_i \times c_j$. Note that the scaling constants c_js are distinct for different diseases.

Consider an example, where the number of DDBs, $n = 5$ and $l = 2$. SNPs related to only one disease Y_1 are used as dummies along with SNPs of the target disease X. Let $\mathbb{P}(X) = \{S_1, S_4\}$ and $\mathbb{P}(Y_1) = \{S_2, S_3, S_5\}$.

All SNP sets related to different diseases with their relevant risk alleles (r_i) and scaled β_i values, i.e., ε_i are accumulated *randomly* to generate the final query message, M. The random organization restricts the DDBs to recognize which SNP set is related to the target disease and which ones to dummies. To scale back the query result derived from the DDBs, the MU saves index value, j of the target disease and constant c_j. Let this index value be γ and the constant be δ. The final query message, M is generated as follows:

$$S_2, 00, \varepsilon_2 : S_3, 01, \varepsilon_3 : S_5, 10, \varepsilon_5 \,|\, S_1, 11, \varepsilon_1 : S_4, 00, \varepsilon_4 \,|$$

As the 2nd SNP set is associated with the target disease X, the MU saves $\gamma = 2$ and $\delta = c_2$ to scale back the results sent by the DDBs. Finally, the MU sends M to each DDB except the n^{th} DDB at the patient's device.

5.3.2 Partial Genetic Score Calculation at the DDBs

Each DDB except the n^{th} DDB at the patient's device uses the query message, M, and patient P's pseudonym, N, to calculate partial genetic and authenticating scores for disease X. Algorithm 1 shows the pseudocode used by the k^{th} DDB to generate the partial scores. It produces return message, R_k as output that contains partial genetic and authenticating scores calculated by the k^{th} DDB.

Table 3 Sample entries for SNP S_4, $k = DDB$ No.

k		$a_{1,k}$	$w_{1,k}$	$\alpha_{1,k}$	$a_{2,k}$	$w_{2,k}$	$\alpha_{2,k}$
1		00	−5	−9	00	−5	−3
2		10	−56	−10	00	−25	−9
3		00	−13	1	10	18	−6
4		10	−10	−8	10	0	4
5	$t = 1$	00	19	5	00	30	7
	$t = 2$	10	66	7	10	−17	5

After necessary parsing, Line 4 finds the ID of the SNP S_i, its risk allele r_i and scaled contribution factor ε_i related to each of the diseases in M. Using the pseudonym, N, Function *RetrieveValues* in Algorithm 1 retrieves the total weight ($\omega_{i,k}$) and the sum of α values ($\alpha_{i,k}$) for the risk allele, r_i of SNP S_i from the k^{th} DDB (Line 5). The function matches r_i with the stored alleles, $a_{1,k}$ and $a_{2,k}$ of S_i. If both the alleles match r_i, *RetrieveValues* returns the summation of corresponding weights $w_{1,k}$ and $w_{2,k}$ as $\omega_{i,k}$ and the summation of values $\alpha_{1,k}$ and $\alpha_{2,k}$ as $\alpha_{i,k}$. If one of these alleles matches r_i, *RetrieveValues* returns the corresponding weight as $\omega_{i,k}$ and the corresponding α as $\alpha_{i,k}$. If none of the alleles matches r_i, 0 is returned as $\omega_{i,k}$ and $\alpha_{i,k}$.

Algorithm 1 *CalculatePartialGeneticScore*

Input: M, N
Output: R_k, where k is the number of the DDB

```
1: for each disease set T_j ∈ M do
2:     s_{j,k} ← 0, m_{j,k} ← 0
3:     for each SNP S_i ∈ T_j do
4:         S_i, r_i, ε_i ← Parse(T_j)
5:         ω_{i,k}, α_{i,k} ← RetrieveValues(S_i, r_i, N)
6:         s_{j,k} ← s_{j,k} + ω_{i,k} × ε_i
7:         m_{j,k} ← m_{j,k} + α_{i,k} × ε_i
8:     end for
9:     R_k.append("s_{j,k}, m_{j,k} :")
10: end for
11: return R_k
```

Consider the second SNP set $\langle S_1, 11, \varepsilon_1 : S_4, 00, \varepsilon_4 \rangle$ of the example in Section 5.3.1. **Table 3** shows sample distributions of weight values in 5 DDBs for SNP S_4. From Table 2 and Table 3, we see that at the 1st DDB, the retrieved weight of risk allele of S_1 (11) and S_4 (00) are respectively,

$$\omega_1 = 62 + 0 = 62, \text{and}$$

$$\omega_4 = (-5) + (-5) = -10$$

Therefore, the partial genetic score is,

$$s_{1,1} = 62 \times \varepsilon_1 - 10 \times \varepsilon_4$$

Similarly, partial authenticating score is,

$$m_{1,1} = 2 \times \varepsilon_1 - 12 \times \varepsilon_4$$

In this way, 1st DDB calculates partial genetic and authenticating scores for $l = 2$ combinations and sends back reply message, R_1 to the MU. A sample R_1 looks like

$$R_1 = s_{1,1}, m_{1,1} : s_{2,1}, m_{2,1}$$

5.3.3 Query Processing at the MU for the n^{th} DDB

The MU extracts partial genetic and authenticating scores from the return messages R_k sent by each of the $n-1$ DDBs. Let $s_{j,k}$ and $m_{j,k}$ respectively be a partial genetic and an authenticating score sent by the k^{th} DDB for the j^{th} SNP set related to any particular disease, where $j \in \{1, \ldots, l\}$. The partial scores in the return messages are maintained in sequence with the SNP sets in the query message, M. The authentication process can detect if a dishonest DDB changes the order or value of the partial scores (see Theorem 6.2). The MU separately adds up all the partial genetic and authenticating scores sent by $n-1$ DDBs to generate the sum $\eta_{j,s}$ and $\eta_{j,m}$, respectively. The SNP set for each disease in the query message, M sent to the $n-1$ DDBs are concatenated with these summation values to generate the new query message, $\bar{M}$ that will be sent to the n^{th} DDB.

The MU retrieves the set of clinical data, $\mathbb{N}(D)$ and contribution factors of these clinical data, $\bar{\beta}$, where D can be any of the l diseases in the query - target and dummy ones. Each set of clinical data related to a disease is randomly partitioned into two separate subsets. To hide the contribution factors (secret of the MU) from malicious parties, $\bar{\beta}_i$ of each clinical data C_i associated with the r^{th} subset of j^{th} disease is multiplied by a randomly generated constant $\bar{c}_{r,j}$ to generate the scaled contribution factor, $\bar{\varepsilon}_i$, such that $\bar{\varepsilon}_i = \bar{\beta}_i \times \bar{c}_{r,j}$, where $r \in \{1, 2\}$, $j \in \{1, \ldots, l\}$. Note that $\bar{c}_{r,j}$ values are distinct for different diseases. The MU saves the scaling constants $\bar{\delta}_r = \bar{c}_{r,j}$, where j^{th} disease is the target disease. Note that $\bar{\delta}_r \neq \delta$, where δ is the constant used to scale the contribution factors of the SNPs related to the target disease. Clinical data are partitioned into two subsets so that the n^{th} DDB cannot infer the contribution factors from the aggregated disease risk.

All the clinical data and their contribution factors are appended at the end of the SNP set for the related disease in the query message, $\bar{M}$. Finally, MU sends $\bar{M}$ to the n^{th} DDB at the patient's device. Continuing our previous example, we assume that $\mathbb{N}(X) = \{C_1, C_2, C_4\}$ and $\mathbb{N}(X)$ is partitioned into two subsets, $\mathbb{N}_1(X) = \{C_1, C_2\}$ and $\mathbb{N}_2(X) = \{C_4\}$. For the dummy disease, $\mathbb{N}_1(Y_1) = \{C_3\}$, and $\mathbb{N}_2(Y_1) = \{C_5\}$. Similar to the previous query message, M, a sample for the new query message, $\bar{M}$ can be as follows:

$$\eta_{1,s}\ , \eta_{1,m}; S_2, 00, \varepsilon_2 : S_3, 01, \varepsilon_3 : S_5, 10, \varepsilon_5\ ; C_3, \bar{\varepsilon}_3 :: C_5, \bar{\varepsilon}_5|$$
$$\eta_{2,s}\ , \eta_{2,m}; S_1, 11, \varepsilon_1 : S_4, 00, \varepsilon_4\ ; C_1, \bar{\varepsilon}_1 : C_2, \bar{\varepsilon}_2 :: C_4, \bar{\varepsilon}_4|$$

5.3.4 Authentication at the n^{th} DDB

After receiving the query message, $\bar{M}$, the n^{th} DDB at the patient's device authenticates the aggregated genetic score sent from the other $n-1$ DDBs and calculates the total genetic and clinical scores for all the l diseases. Algorithm 2 shows the pseudocode for this process. The input to this algorithm is the query messages, $\bar{M}$, the SNP set related to the target disease, $\mathbb{P}(X)$, the authentication key μ stored at the patient's device, and two randomly generated constants ρ and τ used to change the total scores by multiplication and addition. The output is the reply message $\bar{R}$ containing the total scores of l diseases. The SNPs associated with a particular disease and their risk alleles are normally available in public. Since patient P naturally knows the name of the target disease, X, we assume that $\mathbb{P}(X)$ is also known to her.

Algorithm 2 *CalculateAuthenticatedScore*

Input: $\bar{M}, \mathbb{P}(X), \mu, \rho, \tau$
Output: $\bar{R}$

```
 1: γ ← GetIndex(M̄, ℙ(X))
 2: for each disease set T_j ∈ M̄ do
 3:     η_{j,s}, η_{j,m} ← Parse(T_j)
 4:     for each SNP S_i ∈ T_j do
 5:         S_i, r_i, ε_i ← Parse(T_j)
 6:         ω_{i,n}, α_{i,n} ← RetrieveValues(S_i, r_i)
 7:         η_{j,m} ← η_{j,m} + α_{i,n} × ε_i
 8:     end for
 9:     if η_{j,s} = η_{j,m} × μ then
10:         if j = γ then
11:             for each SNP S_i ∈ T_j do
12:                 η_{j,s} ← η_{j,s} + ω_{i,n} × ε_i
13:             end for
14:             η_j ← (η_{j,s} × ρ) + τ
15:             for r = 1 to 2 do
16:                 η̄_{r,j,c} ← 0
17:                 for each clinical data C_i ∈ subset N_{r,j} do
18:                     C_i, ε̄_i ← Parse(N_{r,j})
19:                     v_i ← ReceiveValue(C_i)
20:                     η̄_{r,j,c} ← η̄_{r,j,c} + v_i × ε̄_i
21:                 end for
22:                 η̄_{r,j} ← (η̄_{r,j,c} × ρ) + τ
23:             end for
24:         else
25:             η_j, η̄_{1,j}, η̄_{2,j} ← Random()
26:         end if
27:         R̄.append("η_j, η̄_{1,j}, η̄_{2,j} :")
28:     else
29:         R̄.append("authentication error :")
30:     end if
31: end for
32: return R̄
```

Function *GetIndex* in Algorithm 2 matches $\mathbb{P}(X)$ with the SNP sets in $\bar{M}$ to find the index, γ, of the target disease, X in the query message, $\bar{M}$. After necessary parsing, the algorithm finds the aggregated genetic score $\eta_{j,s}$, aggregated authenticating score $\eta_{j,m}$ and ID of the SNP S_i, its risk allele r_i and scaled contribution factor ε_i related to each of the diseases in $\bar{M}$. Similar to Algorithm 1, Function *RetrieveValues* in Algorithm 2 retrieves the total weight ($\omega_{i,n}$) and the sum of α values ($\alpha_{i,n}$) for the risk allele, r_i of SNP S_i from the n^{th} DDB at the patient's device (Line 6). The function matches r_i with the stored alleles, $a_{1,n,1}$, $a_{1,n,2}$, $a_{2,n,1}$ and $a_{2,n,2}$ of S_i. The weight $\omega_{i,n}$ is calculated by summing those weight ($w_{i,n,t}$) values, whose corresponding allele encoding matches r_i, where $t \in \{1, 2\}$. Similarly, $\alpha_{i,n}$ is calculated by summing the $\alpha_{i,n,t}$ values of the matched alleles. We note that the total number of risk allele r_i in the SNP S_i is,

$$f_i = \sum_{1 \leq k \leq n} \omega_{i,k}$$

Line 7 multiplies $\alpha_{i,n}$ values with the scaled contribution factor, ε_i of each SNP S_i and adds up with the aggregated authenticating score $\eta_{j,m}$. The parameter $\eta_{j,m}$ is multiplied by the authentication key μ and checked whether the multiplied value is equal to the aggregated genetic score $\eta_{j,s}$ (Line 9). If the result does not

match, then the n^{th} DDB decides that the genetic scores are altered or disease sequence is changed by dishonest $n-1$ DDBs or a dishonest MU. Otherwise, if the aggregated genetic score $\eta_{j,s}$ is authenticated as correct, Line 10 checks if the j^{th} disease is the target disease, i.e., $j = \gamma$ or not. If $j = \gamma$, Line 12 multiplies $\omega_{i,n}$ values with the scaled contribution factor, ε_i of each SNP S_i and adds up with $\eta_{j,s}$ to generate the total genetic score. Line 14 multiplies the total genetic score with the constant ρ and adds to the constant τ to generate scaled genetic score, η_j for the j^{th} disease. The n^{th} DDB at the patient's device saves ρ and τ for final computation of the disease risk. This scaling is done so that the MU cannot infer the genetic score even if the patient decides to share the final disease risk with the MU for the purpose of treatment.

Similar to the SNP sets, each clinical data C_i and its scaled contribution factor $\bar{\varepsilon}_i$ are parsed from the query message. The value, v_i of C_i is received from patient, P. Recall that $v_i \in \{0, 1\}$. In Line 20, each v_i is multiplied with the scaled contribution factor, $\bar{\varepsilon}_i$ and summed up to generate the total clinical score for the r^{th} subset of the j^{th} disease, $\bar{\eta}_{r,j,c}$. Similar to Line 14, Line 22 generates scaled clinical score $\bar{\eta}_{r,j}$ using the same constants ρ and τ. The n^{th} DDB saves the values of γ^{th} genetic score, $\eta_{\gamma,s}$ and clinical scores $\bar{\eta}_{r,\gamma,c}$ to check whether a dishonest MU has forged contribution factors to infer genomic or clinical data.

In Line 25, random values are generated as η_j and $\bar{\eta}_{r,j}$ for a dummy disease. This is done so that a dishonest MU cannot generate score for any disease except the target disease without patient's consent. Scaled genetic and clinical scores for all the l diseases are sent in the return message, $\bar{R}$ to the MU.

5.3.5 Aggregation at the MU

The MU finds the total genetic score η_j, and the clinical scores $\bar{\eta}_{r,j}$ corresponding to the r^{th} clinical data subset of the j^{th} disease from the return message, $\bar{R}$ sent by the n^{th} DDB, where $j \in \{1, \ldots, l\}, r \in \{1, 2\}$. Recall that the index value, γ and the scaling constants, δ for genetic score and $\bar{\delta}_r$ for clinical scores related to the target disease, X are saved at the MU during query processing. Thus, γ^{th} scores, $\eta_\gamma, \bar{\eta}_{r,\gamma}$ correspond to the target disease. The MU scales back η_γ and $\bar{\eta}_{r,\gamma}$ using the constants, δ and $\bar{\delta}_r$ respectively and generates $\bar{Z}$ by adding the results as follows,

$$\bar{Z} = \eta_\gamma \times \delta^{-1} + \sum_{r=1,2} \bar{\eta}_{r,\gamma} \times \bar{\delta}_r^{-1}$$

Next, the MU adds inverse of the scaling constants to generate a value Δ such that,

$$\Delta = \delta^{-1} + \sum_{r=1,2} \bar{\delta}_r^{-1}$$

For final computation of the total disease risk, the MU sends $\bar{Z}$ and Δ to the n^{th} DDB at the patient's device.

5.3.6 Final Computation at the n^{th} DDB

After receiving $\bar{Z}$ and Δ from the MU, the n^{th} DDB generates the final score, Z using the previously saved constants ρ and τ such that,

$$Z = (\bar{Z} - \Delta \times \tau) \times \rho^{-1}$$

Final score, Z is used to compute the probability of the patient to develop target disease, X using Eq. (1).

In the n^{th} DDB, the value of Z is checked whether

$$Z = \frac{\eta_{\gamma,s}}{\Delta} \text{ or}$$

$$Z = \frac{\bar{\eta}_{r,\gamma,c}}{\Delta}, r \in \{1, 2\},$$

where $\eta_{\gamma,s}$ and $\bar{\eta}_{r,\gamma,c}$ are the genetic and clinical scores respectively, corresponding to the target disease, X and are saved in the n^{th} DDB (Section 5.3.4). If any of these values are equal to Z, the patient concludes that a dishonest MU has altered contribution factors to infer her SNP contents or clinical data and will not share the final score, Z with the MU.

6. Correctness Analysis

6.1 Proof of Correctness

Theorem 6.1. *Let $\mathbb{P}(X)$ and $\mathbb{N}(X)$ be the sets of SNPs and clinical data related to a disease X, where $|\mathbb{P}(X)| = \lambda$ and $|\mathbb{N}(X)| = \phi$. For each SNP $S_i \in \mathbb{P}(X)$, β_i be the contribution factor of risk allele r_i and f_i be the total number of r_i in S_i. For each clinical data $C_i \in \mathbb{N}(X)$, $\bar{\beta}_i$ be the contribution factor and v_i be the value of C_i. Then the total score of a patient P for developing disease X is*

$$Z = \sum_{1 \le i \le \lambda} f_i \times \beta_i + \sum_{1 \le i \le \phi} v_i \times \bar{\beta}_i$$

Proof. Without loss of generality, we assume that each SNP set related to l different diseases sent by the MU to n DDBs has equal size λ. Recall that $\varepsilon_i = \beta_i \times c_j$ for the j^{th} disease, where $j \in \{1, \ldots, l\}$. Parameter $\omega_{i,k}$ represents the total weight of the risk allele r_i of SNP S_i retrieved from the k^{th} DDB. Thus, the partial score, $s_{j,k}$ generated at the k^{th} DDB, is expressed as

$$s_{j,k} = \sum_{1 \le i \le \lambda} \omega_{i,k} \times \varepsilon_i$$

If authentication is successful at the n^{th} DDB, total genetic score $\eta_{j,s}$ is calculated as follows:

$$\begin{aligned} \eta_{j,s} &= \sum_{1 \le k \le n-1} s_{j,k} + \sum_{1 \le i \le \lambda} \omega_{i,n} \times \varepsilon_i \\ &= c_j \times \sum_{\substack{1 \le k \le n \\ 1 \le i \le \lambda}} \omega_{i,k} \times \beta_i \end{aligned}$$

Without loss of generality, we assume that each set of clinical data related to l different diseases has equal size ϕ and is partitioned into two subsets of equal size θ, i.e, $\phi = 2\theta$. For each clinical data C_i in the r^{th} subset of the j^{th} disease, $\bar{\varepsilon}_i = \bar{\beta}_i \times \bar{c}_{r,j}$, where $r \in \{1, 2\}$. The n^{th} DDB computes clinical score, $\bar{\eta}_{r,j,c}$ using the following equation:

$$\begin{aligned} \bar{\eta}_{r,j,c} &= \sum_{1 \le i \le \theta} v_i \times \bar{\varepsilon}_i \\ &= \bar{c}_{r,j} \times \sum_{1 \le i \le \theta} v_i \times \bar{\beta}_i \end{aligned}$$

The n^{th} DDB changes the genetic and clinical scores of the target disease X ($j = \gamma$) using constants ρ and τ such that

$$\eta_\gamma = (\eta_{\gamma,s} \times \rho) + \tau \text{ and}$$

$$\bar{\eta}_{r,\gamma} = (\bar{\eta}_{r,\gamma,c} \times \rho) + \tau$$

We note that for $c_\gamma = \delta$, $\bar{c}_{r,\gamma} = \bar{\delta}_r, r \in \{1, 2\}$, and $\Delta = \delta^{-1} +$

$\sum_{r=1,2} \bar{\delta}_r^{-1}$.

The MU calculates $\bar{Z}$ such that

$$\begin{aligned}
\bar{Z} &= \eta_\gamma \times \delta^{-1} + \sum_{r=1,2} \bar{\eta}_{r,\gamma} \times \bar{\delta}_r^{-1} \\
&= \frac{\eta_{\gamma,s} \times \rho}{\delta} + \frac{\tau}{\delta} + \sum_{r=1,2} \frac{\bar{\eta}_{r,\gamma,c} \times \rho}{\bar{\delta}_r} + \frac{\tau}{\bar{\delta}_r} \\
&= \rho \left(\frac{\eta_{\gamma,s}}{\delta} + \sum_{r=1,2} \frac{\bar{\eta}_{r,\gamma,c}}{\bar{\delta}_r} \right) + \tau \times \left(\frac{1}{\delta} + \sum_{r=1,2} \frac{1}{\bar{\delta}_r} \right) \\
&= \rho \left(\sum_{\substack{1 \le k \le n \\ 1 \le i \le \lambda}} \omega_{i,k} \times \beta_i \delta \times \frac{1}{\delta} + \sum_{\substack{r=1,2 \\ 1 \le i \le \theta}} v_i \times \bar{\beta}_i \bar{\delta}_r \times \frac{1}{\bar{\delta}_r} \right) + \tau\Delta \\
&= \rho \left(\sum_{1 \le i \le \lambda} f_i \times \beta_i + \sum_{1 \le i \le \phi} v_i \times \bar{\beta}_i \right) + \tau\Delta,
\end{aligned}$$

since we have $f_i = \sum_{1 \le k \le n} \omega_{i,k}$ from Section 5.3.4. Finally, the n^{th} DDB calculates the total score, Z as follows:

$$\begin{aligned}
Z &= (\bar{Z} - \tau\Delta) \times \rho^{-1} \\
&= \sum_{1 \le i \le \lambda} f_i \times \beta_i + \sum_{1 \le i \le \phi} v_i \times \bar{\beta}_i
\end{aligned}$$

□

6.2 Proof of Authentication

Theorem 6.2. *Let the number of DDBs be n. The n^{th} DDB can detect if the other $n-1$ DDBs or the MU alter SNP data used in a disease risk query.*

Proof. Each DDB stores weight w and a value α for each allele of an SNP. Let $\omega_{i,k}$ and $\alpha_{i,k}$ respectively be the total weight and the total value of α for risk allele r_i of SNP S_i in the k^{th} DDB. According to Section 5.2, we have

$$\sum_{1 \le k \le n-1} \omega_{i,k} = \mu \times \sum_{1 \le k \le n} \alpha_{i,k}$$

If the DDBs change weight or α value for an SNP or the MU changes the sum of partial genetic and authenticating scores generated by $n-1$ DDBs arbitrarily, patient P can detect the changes, since authentication key μ is only known to P. □

Note that it is not possible to ensure the accuracy of disease risk queries if the MU uses inaccurate values of contribution factors for SNPs and clinical data. Hence, we focus on the integrity of SNP data for authentication purpose.

7. Security Analysis

In our architecture, the SI is an honest entity. This assumption, also considered in Refs. [11], [18], [19], [20], [34] is inevitable in the sense that if sequencing institution is not trusted then the security of genomic data cannot be guaranteed. We assume that all involved entities except the SI and the patient are dishonest. Dishonest entities can arbitrarily tamper with the stored data or the messages to infer sensitive information and to change the query result. We also take into account the collusion of the DDBs and the MU, which is not considered in the previous works [11], [18], [19], [20], [34]. Additionally, we consider the attack from eavesdroppers at the time of transferring data among the SI, the MU, the DDBs, and the patient (i.e., the n^{th} DDB).

A patient has four types of sensitive information: genomic and clinical data, the name of the target disease, and the query answer. Additionally, the contribution factors of SNPs and clinical data can be trade secrets of the MU. The DDBs, and the eavesdroppers are considered adversaries for all of these five types of sensitive information. A patient (i.e., the n^{th} DDB) is considered adversary for an MU's contribution factors. Similarly, an MU is considered adversary for a patient's genomic and clinical data. However, the MU needs to know the name of the target disease for query processing and the patient can share the query answer with the MU for treatment purposes. Thus, an MU is not considered adversary for these two types of sensitive information of the patient.

In the following subsections, we prove that our approach does not allow an adversary to infer sensitive data, and guarantees the required privacy levels.

7.1 Security of the Genomic and Clinical Data

We will first show how our approach protects the privacy of genomic and clinical data of a patient when the DDBs and the MU individually try to infer these information, and then consider the collusion among the DDBs and the MU.

Theorem 7.1. *Let the number of DDBs be n, where the n^{th} DDB is the patient's device. Every SNP value of the patient stored in n DDBs is secure, unless $n-1$ DDBs including the n^{th} DDB are compromised.*

Proof. We know that each SNP has two alleles and there are only two probable nucleotides from $\{A, C, G, T\}$ for each of the two alleles of an SNP. For each allele, one of these nucleotides and its weight are stored in a randomly selected m DDBs, where $m \le n-2$. On the remaining $n-m-1$ DDBs, other nucleotide with corresponding weight are stored. Let $a_{j,k}$ and $w_{j,k}$ denote the nucleotide and the corresponding weight of the j^{th} allele of an SNP stored in the k^{th} DDB, where $j \in \{1,2\}, k \in \{1, \ldots, n-1\}$, and $-100 \le w_{j,k} \le 100$.

In the n$^{\text{th}}$ DDB at the patient's device, both probable nucleotides are stored for each allele. Let $a_{j,n,t}$ and $w_{j,n,t}$ be the nucleotide and the corresponding weight stored for the j^{th} allele of an SNP in the n^{th} DDB, where $t \in \{1,2\}$.

According to the technique to store SNP information in DDBs (Section 5.2), the weights of two probable nucleotides are distributed in a way such that the summation of weights is 1 for the actual nucleotide which the patient contains in the j^{th} allele of a particular SNP, and for the false nucleotide, the summation of weights is 0. Assume that for $j=1$, $a_{1,n,1}$ is the actual nucleotide and $a_{1,n,2}$ is the false one. Formally, for $j=1$, we have

$$\sum_{1 \le k \le m} w_{1,k} + w_{1,n,1} = 1, \text{and}$$
$$\sum_{1 \le k \le n-m-1} w_{1,k} + w_{1,n,2} = 0$$

Thus, if an adversary compromises $n-1$ DDBs except the n^{th} DDB, it can only have partial total weights ($\sum w_{j,k}$), which do not reveal any information as both weights can get modified after adding the weights stored in the n^{th} DDB ($w_{j,n,t}$). On the other hand, hacking only the patient's device without compromising other $n-2$ DDBs also does not reveal anything about the total

weights of a nucleotide for an allele of an SNP, i.e., the true content of an SNP. □

To ensure that an eavesdropper cannot infer actual SNP contents, the SI encrypts data before sending them to the n^{th} DDB at the patient's device and the patient decrypts them before storing. The clinical data of the patient is never shared with the other $n-1$ DDBs. Moreover, the patient sends aggregated clinical score to the MU at the time of computation. Thus, a DDB and an eavesdropper cannot infer the values of the patient's clinical data.

Theorem 7.2. *An MU cannot infer the SNP contents and value of clinical data of a patient P, if P does not share the final disease risk, Pr with the MU. If P shares Pr with the MU, the probability to infer SNP contents and value of clinical data of P related to the target disease, X by the MU is* $\frac{1}{3^{\lambda}\times 2^{\phi}}$*, where* λ *and* ϕ *respectively be the number of SNPs and clinical data related to X.*

Proof. In our approach, the MU receives η_{γ} and $\bar{\eta}_{r,\gamma}$ from the n^{th} DDB at the patient's device such that

$$\eta_{\gamma} = \eta_{\gamma,s} \times \rho + \tau \text{ and}$$
$$\bar{\eta}_{r,\gamma} = \bar{\eta}_{r,\gamma,c} \times \rho + \tau, r \in \{1,2\}$$

where $\eta_{\gamma,s}$ and $\bar{\eta}_{r,\gamma,c}$ are the total genetic score and total clinical scores for the r^{th} clinical data subset of the target disease X. As constants ρ and τ are only known to the n^{th} DDB, the MU cannot learn the values of $\eta_{\gamma,s}$ and $\bar{\eta}_{r,\gamma,c}$ and thus, cannot infer SNP contents and clinical data from the information at hand.

If the patient P shares final disease risk Pr with the MU for treatment purpose, the MU can find total disease score, Z using Eq. (1). Recall that

$$Z = \sum_{1\le i\le\lambda} f_i \times \beta_i + \sum_{1\le i\le\phi} v_i \times \bar{\beta}_i$$

where f_i is the number of risk alleles in SNP S_i and v_i is the value of clinical data C_i related to X, $f_i \in \{0,1,2\}$ and $v_i \in \{0,1\}$. Since contribution factors β_i and $\bar{\beta}_i$ are known to the MU, there are $3^{\lambda}\times 2^{\phi}$ possible values for Z. Thus, the probability of the MU to infer SNP contents and value of clinical data is $\frac{1}{3^{\lambda}\times 2^{\phi}}$. □

To directly infer SNP contents from the disease risk, a dishonest MU can tamper values of Δ and $\bar{Z}$ before sending those to the n^{th} DDB. Assume that the MU uses $\bar{\delta}_r^{-1} = 0$ for $r \in \{1,2\}$ in the equations

$$\Delta = \delta^{-1} + \sum_{r=1,2} \bar{\delta}_r^{-1} \text{ and}$$
$$\bar{Z} = \eta_{\gamma} \times \delta^{-1} + \sum_{r=1,2} \bar{\eta}_{r,\gamma} \times \bar{\delta}_r^{-1},$$

where γ is the index of the target disease, X in the query message. Thus, total disease score calculated in the patient's device at the n^{th} DDB will be,

$$Z = (\bar{Z} - \tau\Delta) \times \rho^{-1} = \sum_{1\le i\le\lambda} f_i \times \beta_i$$

The MU can directly infer SNP contents (i.e., f_i) from the previous equation, if it alters β_i values to be consecutive powers of a value greater than the highest value of SNP (i.e., 2) as described in Ref. [34].

To handle this attack, the n^{th} DDB checks whether

$$Z = \frac{\eta_{\gamma,s}}{\Delta} \text{ or}$$
$$Z = \frac{\bar{\eta}_{r,\gamma,c}}{\Delta}, r \in \{1,2\},$$

where $\eta_{\gamma,s}$ and $\bar{\eta}_{r,\gamma,c}$ are the genetic and clinical scores of the target disease X calculated at the n^{th} DDB (Section 5.3.4). If any of these values are equal to Z, the patient (i.e., the n^{th} DDB) can detect that the MU has altered contribution factors to infer the SNP contents or clinical data. One may argue that for this checking, a patient needs to know the value of Δ and the MU can change Δ arbitrarily. However, in that case, Z will not be accurate and the MU will not be able to infer real values of SNP.

We note that patient P reveals her pseudonym to the MU for generating a single disease risk query. However, even if a dishonest MU generates multiple queries using the pseudonym without the consent of P and the queries are different only in terms of SNP IDs or contribution factors, SNP contents of P are not revealed to the MU, because P does not share part of SNP data stored on her device. In such a crucial case, the reply from the patient's device is considered as the patient's authentication for the validity of the queries. Moreover, the MU cannot generate risk score for a dummy disease instead of the target one, because P only retrieves weights of those SNPs that relate to the target disease from her database. For dummy diseases, random numbers are used as SNP weights in the patient's device.

Moreover, the collusion of the MU and one or all of the $n-1$ DDBs does not cause any leakage of the SNP contents from our system, unless the n^{th} DDB stored at the patient's database is also stolen. This stealing issue can be handled similarly as a stolen smart card. The patient needs to immediately inform the SI at once so that the SI can freeze the patient's records in the other DDBs.

7.2 Security of the Name of the Target Disease

The DDBs or the eavesdroppers can predict the disease name if they can know which and how many SNPs along with their risk alleles are being used in a disease risk query. For example, if the MU's request includes SNPs relevant to the heart disease, the DDBs might conclude that the patient has chances to develop heart disease. Though the actual identity of the patient is hidden from the DDBs using pseudonym, extensive research has proved that re-identification of individuals is plausible if background demographic data (location, time, age etc.) or additional genomic information about the person or his/her relatives are available to the adversaries [37]. The following theorem shows that our system hides the name of the target disease from the DDBs and eavesdroppers.

Theorem 7.3. *Let an MU send a disease risk query message M to the DDBs for calculating the probability of a patient P to develop a disease X, where M contains SNPs related to other* $l-1$ *dummy diseases along with the SNPs related to X. An eavesdropper can know the content of M during the communication of M between MU and DDBs. The probability for the DDBs or an eavesdropper to infer the SNPs related to the target disease, X, and thus the name of the target disease is* $\frac{1}{l}$.

Note that $l-1$ dummy diseases are chosen carefully from di-

verse areas (not closely related to the target disease). As such, the DDBs will not be able to make any guess about the disease type, not even by keeping track of multiple queries generated for the same patient. For example, if breast cancer is the target disease, the dummy diseases will be chosen such that they are not different types of cancers.

We note that if a patient wants to check her susceptibility to a disease through multiple MUs, the name of the target disease can be inferred from the intersection of the different queries generated by the MUs. To deal with this issue, different MUs need to use the same set of dummy diseases for a specific disease risk query of a patient. Recall from Section 5.3.4 that the patient's device receives a query message from the MU for the purpose of authentication and calculation of genetic and clinical scores. Since the query message includes the SNP set for every dummy disease, the patient's device can store the set of SNPs for every dummy disease. In case the patient wants to conduct a further query about the risk of the same target disease from a different MU, she can provide these dummy SNP sets to the new MU along with her pseudonym. Consequently, the MU needs to generate query message using the dummy SNP sets it received from the patient. In case a disease risk query is initiated for the first time, an MU will not receive any dummy SNP sets from the patient, and thus, it is allowed to choose dummy diseases for the query. One may argue that a malicious MU may not follow this protocol and use different dummy SNP sets in the query sent to the DDBs. However, since the patient's device stores the dummy SNP sets, it is possible to check whether or not the MU follows the protocol by running an authentication procedure for each dummy disease. We acknowledge that this process will introduce a small storage cost to the patient's device depending on the number of SNPs associated with the dummy diseases. Also, it will add to time complexity, since an authentication procedure is performed for every disease in the query instead of only for the target disease, as done originally. However, the storage and time cost is negligible compared to the overall storage cost and time complexity of the disease risk query.

7.3 Security of the Contribution Factors

The β and $\bar{\beta}$ values represent contribution factors of SNPs and clinical values respectively associated with the particular disease. Hence, β and $\bar{\beta}$ values are sensitive data from the MU's point of view and should not be exposed to a patient, a DDB or an eavesdropper. Considering this fact, we develop a solution in which the MU scales β and $\bar{\beta}$ values before sending those to the DDBs and the patient's device (the n^{th} DDB).

Theorem 7.4. *An adversary cannot infer the β and $\bar{\beta}$ values from the messages sent by an MU to the DDBs and the patient's device for a disease risk query.*

Proof. The set of clinical data related to each of the l diseases are partitioned into two subsets. Before sending the query message to the DDBs, the MU scales contribution factors β_i of each SNP S_i and $\bar{\beta}_i$ of each clinical data C_i for the r^{th} subset of the j^{th} disease with randomly generated constants, c_j, and $\bar{c}_{r,j}$ such that

$$\varepsilon_i = \beta_i \times c_j \text{and } \bar{\varepsilon}_i = \bar{\beta}_i \times \bar{c}_{r,j}, r \in \{1,2\}, j \in \{1,\ldots,l\}$$

As the scaling constants, c_j and $\bar{c}_{r,j}$ are known to only the MU, a DDB, or an eavesdropper can only learn the ε_i and $\bar{\varepsilon}_i$ values.

For the target disease X, $c_j = \delta$, $\bar{c}_{r,j} = \bar{\delta}_r$, and $j = \gamma$. Let $\eta_{\gamma,s}$ and $\bar{\eta}_{r,\gamma,c}$ respectively be the total genetic and clinical scores related to the target disease X calculated at the n^{th} DDB, where $r \in \{1,2\}$. During final computation, the n^{th} DDB receives the values of Δ and $\bar{Z}$ from the MU such that

$$\Delta = \delta^{-1} + \sum_{r=1,2} \bar{\delta}_r^{-1}$$

and

$$\begin{aligned}
\bar{Z} &= \eta_\gamma \times \delta^{-1} + \sum_{r=1,2} \bar{\eta}_{r,\gamma} \times \bar{\delta}_r^{-1} \\
&= \frac{\eta_{\gamma,s} \times \rho}{\delta} + \frac{\tau}{\delta} + \sum_{r=1,2} \frac{\bar{\eta}_{r,\gamma,c} \times \rho}{\bar{\delta}_r} + \frac{\tau}{\bar{\delta}_r} \\
&= \rho \left(\frac{\eta_{\gamma,s}}{\delta} + \sum_{r=1,2} \frac{\bar{\eta}_{r,\gamma,c}}{\bar{\delta}_r} \right) + \tau \times \left(\frac{1}{\delta} + \sum_{r=1,2} \frac{1}{\bar{\delta}_r} \right) \\
&= \rho \left(\frac{\eta_{\gamma,s}}{\delta} + \sum_{r=1,2} \frac{\bar{\eta}_{r,\gamma,c}}{\bar{\delta}_r} \right) + \tau \times \Delta
\end{aligned}$$

The values of ρ, τ, Δ, $\eta_{\gamma,s}$ and $\bar{\eta}_{r,\gamma,c}$ are known to the n^{th} DDB. However, it cannot infer the values of three separate scaling constants δ, $\bar{\delta}_1$, and $\bar{\delta}_2$ from the two equations at hand. Thus, contribution factors β_i of SNP S_i and $\bar{\beta}_i$ of clinical data C_i remain hidden from the adversaries. We note that knowing an important part of own genome does not add any advantage for the n^{th} DDB in reverse engineering β and $\bar{\beta}$ values. □

7.4 Security of the Query Answer

The DDBs and the eavesdroppers are not allowed to infer the query answer, i.e., the total score to develop any particular disease by a patient P. The following theorem shows that our system ensures the security of the query answer from the adversaries.

Theorem 7.5. *An adversary cannot infer the total score, Z of a patient P for a disease risk query.*

Proof. Let n be the number of DDBs and $\bar{Z}$ be the score sent from the MU to the n^{th} DDB in the patient's device at the final step. The n^{th} DDB calculates the total score,

$$Z = \frac{\bar{Z} - \tau\Delta}{\rho}$$

Since, ρ and τ are known only to the patient, the DDBs and the eavesdroppers cannot infer the Z, even if all the $n-1$ DDBs collude. □

8. Results

In this section, we evaluate the effect of varying the privacy level for disease risk queries on the performance of our proposed system. The privacy level is expressed using the number of the DDBs (n), the number of diseases used in a query (l), and the total number of SNPs related to l diseases in the query. We use 0.3 million SNPs from a real SNP profile [3] released by the 1000 genome project [4]. The relevant information, i.e., SNPs, their risk alleles, clinical data have been collected from Ref. [5]. The contribution factors corresponding to the SNPs related to the Coronary Artery Disease (CAD) are collected from Ref. [48]. For

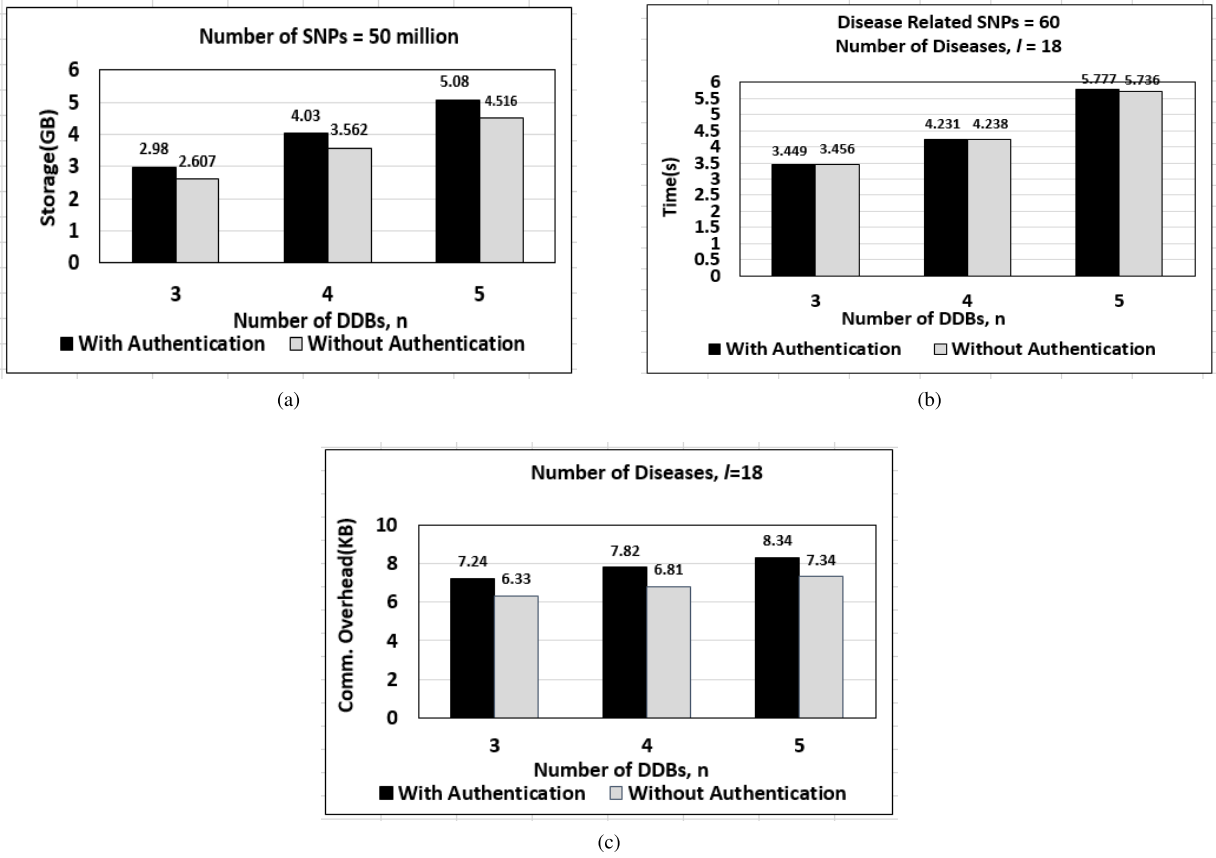


Fig. 3 Effect of n on (a) storage, (b) time and (c) communication overhead.

Table 4 Values of different parameters.

Parameter	Values	Default
n	3–5	4
l	5–25	18
Total SNP count	50–75	–

the rest of the diseases, contribution factors are randomly generated.

Table 4 summarizes the parameter values used in our experiments. We repeat every experiment for 100 disease risk queries and present the average result in terms of storage, computational and communication overhead. To represent the communication overhead independent of the used communication link, we measure the communication cost in terms of transferred data size among involved parties. We can approximate the communication delay from the known latency of the used communication link.

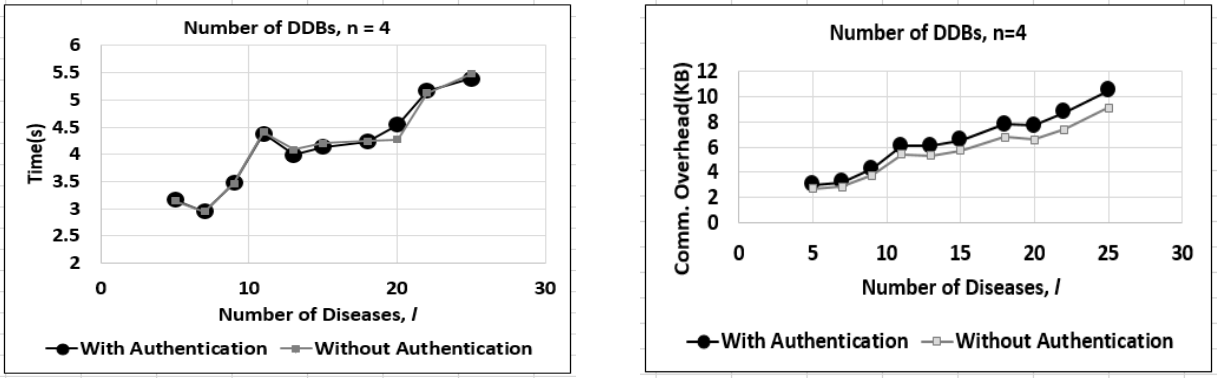
We have performed experiment on our proposed system on Intel Core i5 CPUs with 2.7 GHz processor under macOS using Eclipse 4.6 and MySQL database. In Sections 8.1–8.2, we present our result for varying n, l, and the total number of SNPs in a query. In Section 8.3, we compare our approach with the existing literature.

8.1 Effect of n

For evaluating authenticated disease risk query, each tuple of a DDB entry needs 8×8 (8 character pseudonym) + 8×10 (10 character SNP ID) + 2×2 (two 2 bit naïvely encoded alleles) + 2×8 (two 8 bit tiny integer weight of the two alleles) + 2×8 (two 8 bit tiny integer for authenticating value α of the two alleles) = 180 bits. Only exception is the patient's DDB for which each tuple needs $8 \times 10 + 4 \times (2 + 8 + 8)$ i.e., 152 bits, as it does not have the pseudonym attribute but has both possible nucleotides and corresponding weights and α values for each of the two alleles. The patient's DDB also saves authentication key μ as an 8 bit tiny integer.

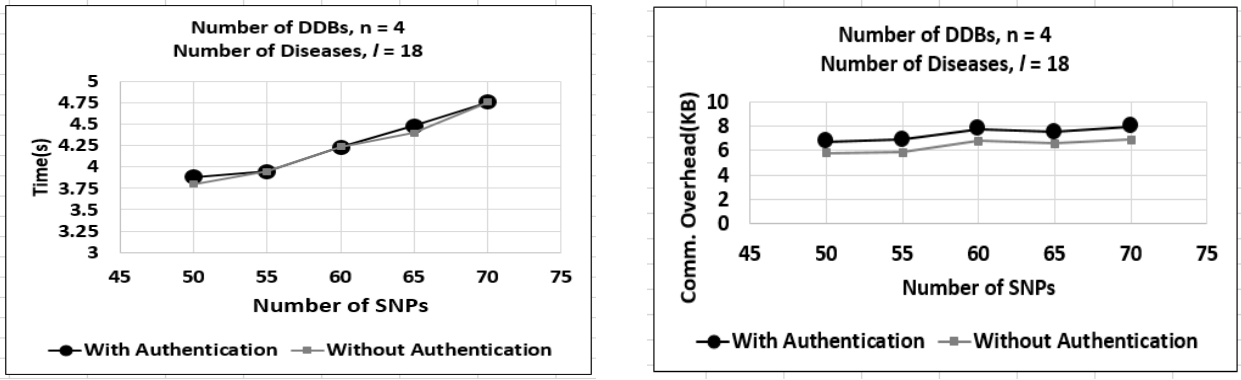
In an unauthenticated system, there will be no authenticating value α in the DDBs. As such, each tuple of a DDB entry will need only 164 bits and each tuple in the patient's DDB will need only 120 bits. Thus to store 50 million SNPs, total storage is $(180(n-1) + 152) \times 50 \times 10^6 + 8$ bits for an authenticated system, and $(164(n-1) + 120) \times 50 \times 10^6$ bits for an unauthenticated system.

Again, in a system with no privacy measure, there are only two message transfers between the MU and a central data center for the computation of the genetic score. However, in a system with n number of DDBs, number of message transfers between the MU and the DDBs is $(2n + 1)$ if the system is authenticated and $2n$ otherwise. As such the storage size and communication overhead increases linearly with the increase of n (**Fig. 3** (a) and Fig. 3 (c)). We emphasize that the DDBs are linked to the MU with a parallel interface connections and all the DDBs compute partial genetic scores simultaneously. Hence, with the increase of n, the computational time is not affected significantly apart from the time needed for the connection setup and packet transfer. Figure 3 (b) shows that the computational time varies in ms range between authenticated and unauthenticated system.



(a) (b)

Fig. 4 Effect of l on (a) time and (b) communication overhead.



(a) (b)

Fig. 5 Effect of number of SNPs on (a) time and (b) communication overhead.

8.2 Effect of l and Number of SNPs in a Query

The time and number of bits needed to generate the query message at the MU and the return messages with partial genetic scores at the DDBs depend on the total number of SNPs that are subject to randomly chosen (l-1) diseases. Thus, with the increase of total number of SNPs, time and communication overhead increases linearly (**Fig. 5** (a) and Fig. 5 (b)). However, **Fig. 4** (a) and Fig. 4 (b) show almost linear patterns with several peaks and valleys. The reason behind this behavior is that the number of SNPs related to a disease can vary in a large range. As such, smaller value of l may result in larger number of SNPs used in a query.

Figure 4 and Fig. 5 show that communication overhead increases slightly in an authenticated system compared to an unauthenticated one and computational time remains almost same in both systems. Furthermore, we note that increasing l or total number of SNPs does not affect the storage size.

8.3 Comparative Analysis

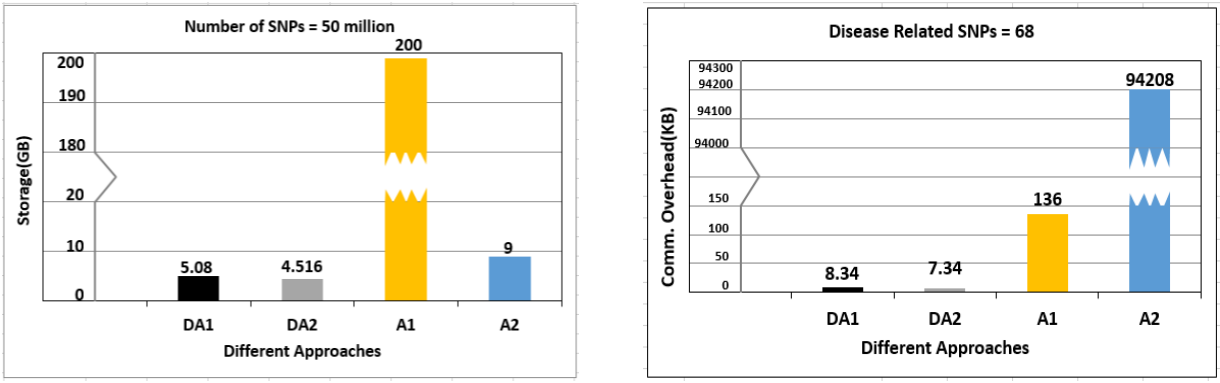
We have compared the performance of our system (authenticated and unauthenticated systems denoted as DA1 and DA2, respectively in graphs) with recent cryptographic approaches [11], [19], [34] (denoted as A2, A1 and A3, respectively in graphs). These approaches consider the effect of multiple SNPs on disease risk queries. However, none of these approaches authenticates the disease risk query.

8.3.1 Storage Overhead

In Ref. [19], two BCP ciphertexts (one for the SNP, other for its square) for approximately 50 million known SNPs are stored in encrypted form. Each BCP ciphertext is a pair of 4096-bit group elements. Thus, the total storage for 50 million SNPs is $2 \times (50 \times 10^6) \times (2 \times 4096)$ bits, i.e., almost 100 GB. In Ref. [11], all the 50 million SNPs are sent at once and the storage needed to encrypt all the SNPs takes $2 \times (50 \times 10^6) \times (2 \times 193)$ bits, i.e., about 4.5 GB. Both of these approaches store only the frequency of one allele in each SNP. If these approaches consider storing both alleles, the storage becomes double (**Fig. 6** (a)). The storage of Ref. [34] is similar to Ref. [19], as it follows the encryption method of Ref. [19]. On the contrary, the storage of our system depends on the number of DDBs, n. For $n = 5$, which ensures a good privacy level, our authenticated and unauthenticated approaches require about 5.08 GB and 4.516 GB, respectively and the cost lies below [11] till $n \leq 8$ and $n \leq 9$, respectively to store 50 million SNPs.

8.3.2 Communication Overhead

For $n = 5$, $l = 18$ and the number of total SNPs related to l diseases = 68, communication overhead of our authenticated and unauthenticated systems are 8.34 KB and 7.34 KB, respectively. On the contrary, in Ref. [19], the data center needs to send two BCP ciphertexts for each SNP (one for the SNP, other for its square). If we consider these 68 SNPs, the communication overhead entails $2 \times 68 \times (2 \times 4{,}096)$ bits, i.e., 136 KB which is significantly higher than the overhead incurred by our system (Fig. 6 (b)). The approach proposed in Ref. [34] also incurs similar communication traffic as Ref. [19], since these two approaches



(a) (b)

Fig. 6 Comparative analysis in terms of (a) storage and (b) communication overhead.

use the same encryption method. Again, the approach in Ref. [11] always uses 1 million SNPs to hide the disease name for which its communication cost amounts to 92 MB, which is extremely high.

9. Applicability of Our Approach in Personalized Medicine

With the emergence of recent technologies, researchers have come up with novel techniques for prescribing patients with accurate dosage of medicines. One way to prescribe medicine for a patient is to use the patient's genomic information in addition to her clinical diagnosis [35], which leads to the field of personalized medicine. According to Ref. [26], the base of personalized medicine comes from the unique characteristics of individuals at the physiological, molecular and environmental exposure. Variation in genomic data is responsible for those unique characteristics. Hence the genomic configuration of a patient plays a key role in personalized medicine. It can be considered as an extension of traditional approaches where patients are treated only on the basis of clinical data.

Over the past few years, researchers have discovered that genomes have a close connection with the effects of medicines on certain diseases [28], [33], [47]. For example, CYP2D6 is one of those SNPs that are responsible for clinical depression and Tricyclic Antidepressants (TCAs) are one of the oldest prescribed medications used for the treatment of clinical depression. Different mutations of CYP2D6 have different impact on the effect of TCAs [47]. Hence different types of TCAs are used according to patient's genomic profile in personalized medicine. Like TCAs, many other medications are also dependent on the mutations of genomes that a patient has in her genomic profile. Traditional methods do not consider this relationship between our genomic profile and our medication which reduces their success rate of treatment. On the other hand, personalized medicine always depends on the genomic profile of a patient along with her clinical diagnosis in order to find the genetic mutations that are responsible for a disease and selects medication according to the findings. Thus the success rate of personalized medicine is always greater than that of traditional medicine. However, although the cost of clinical diagnosis is same for both treatments, the cost of personalized medicine rises because of considering a patient's genomic profile [45]. Recently a number of approaches [10], [12] have been introduced for personalized medicine that can handle a large number of diseases like breast cancer and clinical depression. In Ref. [10], Chan et al. described a novel approach for treating breast cancer using the identification of genetic changes that are associated with the occurrence of cancer symptoms. They used optimized drug doses that differ from patient to patient depending on which mutation a patient has in her genomic profile. However, the authors did not describe techniques to preprocess or store genomic data, since the genomic profile was assumed to be readily available. In contrast, Rodel et al. considered preprocessing of genomic data and developed a biobank to store mapping between genotype and phenotype of a patient using DNA extracted from discarded blood sample [12]. However, this genotype-phenotype relationship is difficult to apply in routine healthcare. Additionally, none of these methods ensure privacy of genomic data from dishonest entities.

Section 5 presents our novel secret sharing approach to generate authenticated result for a disease risk query while preserving privacy of the genomic data. Our approach ensures the usage of genomic data of a patient without violating the privacy of the genomic data from any dishonest entities. Specifically, a patient does not need to disclose genomic data to potentially malicious Medical Units (MU) while processing disease risk queries in our approach. However, for personalized medicine a patient cannot hide her genomic data from the MU, as the MU is responsible for prescribing medicine to the patient. Now we will discuss the modifications required in our approach to apply it in the field of personalized medicine while preserving privacy from all other entities except the MU.

Approval of Patient for Sharing Genomic Data with the MU: Section 7 shows that our system does not allow any malicious entity to learn the SNP values stored in a patient's database without her approval. However, unlike the disease risk query, there is no fixed formula that the MU can use to prescribe personalized medicine without knowing the patient's genomic mutations responsible for the particular disease. As such, our system needs to collect approval from the patient to disclose genetic information to the MU.

Choice of Contribution Factor for SNPs: According to our structure, the sum of all weights stored in all DDBs for an allele of a particular position of an SNP is either 1 (if the SNP

contains that allele in that position) or 0 (otherwise). We also consider both positions of an SNP for an allele. Hence if we send a contribution factor for that allele to all DDBs, after aggregating all values received from all DDBs we can receive either that contribution factor (if the allele is in a single position in the SNP), or double of that contribution factor (if the allele is in both position) or zero (if the allele is absent in that SNP). Moreover, the patient's device generates the aggregated value of all the contribution factors for all SNPs. Thus to find out the genetic information, we need to find out a way where the MU can take decision about all SNPs by looking at the aggregated value only. One possible solution can be a careful choice of the contribution factors so that they produce different values for different combinations. For example, let us assume that the MU needs to know about alleles of three SNPs; S_1, S_2 and S_3. For simplification, we are considering that the MU is searching same allele, C, in all SNPs. Thus $P = \{S_1, S_2, S_3\}$ has total eight subsets $(\{S_1\}; \{S_2\}; \{S_3\}; \{S_1, S_2\}; \{S_2, S_3\}; \{S_1, S_3\}; \{S_1, S_2, S_3\}; \emptyset)$. If the MU considers $\{2, 3, 9\}$ as the set of their contribution factors, then it produces different aggregated values for each subset regardless of whether an SNP has one or two C. On the other hand, if the MU considers $\{2, 3, 5\}$ as the set of contribution factors, both $\{S_1, S_2\}$ and $\{S_3\}$ have 10 as aggregated value when each SNP has C in both position. In this case, it is not possible to learn individual SNP values from the aggregated value. Thus the second combination cannot be used for learning SNP information.

Disclosure of Actual Genetic Score: As we are disclosing genetic information to the MU, we do not need to scale the total genetic score in order to hide from the MU described in Section 5.3.4. After authenticating the partial aggregating value and generating total genetic score from the value, the patient directly sends it to the MU for further procedure.

Disclosure of Clinical Data: Like genetic information, the MU needs to know about the clinical factors of a patient. Thus the patient needs to share her clinical data with the MU.

10. Conclusion

We introduced a novel secret sharing approach to evaluate privacy preserving authenticated disease risk queries that overcomes the limitations of existing approaches. Our approach can compute the probability of an individual to develop a disease when both the alleles of an SNP are responsible for two or more different diseases, and protect privacy of genome and clinical data even if the MU alters important parameters and colludes with the DDBs. Moreover, we ensure the correctness of the disease risk query by authenticating genomic data shared by the DDBs. Our security analysis shows that our approach protects the privacy of contribution factors, disease name, and the query answer from dishonest entities. An important advantage of our approach is that the storage cost for SNPs is reduced significantly. Experiments show that our approach outperforms the existing approaches in terms of storage with a large margin. Furthermore, our approach provides a high level of privacy for a smaller value of n (i.e., 3) and incurs less computational and communication overheads.

We showed that our approach is applicable for personalized medicine with modifications. In the future, we aim to address the required modifications and perform experiments on the application of our modified approach for personalized medicine.

References

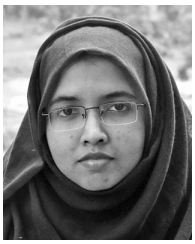
[1] Scitable, available from ⟨https://www.nature.com/scitable/definition/single-nucleotide-polymorphism-snp-295⟩.
[2] rs6313, available from ⟨http://www.snpedia.com/index.php?title=Rs6313⟩.
[3] 1000Genomes, available from ⟨ftp://ftp.ncbi.nih.gov/1000genomes/ftp/technical/reference/⟩.
[4] IGSR, The International Genome Sample Resource. available from ⟨http://www.1000genomes.org/about⟩.
[5] Eupedia, available from ⟨www.eupedia.com/genetics/medical_dna_test.shtml⟩.
[6] 23andMe, available from ⟨https://www.23andme.com/welcome⟩.
[7] Counsyl, available from ⟨https://www.counsyl.com⟩.
[8] Akgün, M., Bayrak, A.O., Ozer, B. and Sağłroğlu, M.Ş.: Privacy preserving processing of genomic data: A survey, *Journal of Biomedical Informatics*, Vol.56, pp.103–111 (2015).
[9] Blanton, M. and Aliasgari, M.: Secure outsourcing of DNA searching via finite automata, *Data and Applications Security and Privacy XXIV*, pp.49–64 (2010).
[10] Chan, C.W.H., Law, B.M.H., So, W.K.W., Chow, K.M. and Waye, M.M.Y.: Novel strategies on personalized medicine for breast cancer treatment: An update, *International Journal of Molecular Sciences*, Vol.18, No.11, p.2423 (2017).
[11] Danezis, G. and Cristofaro, E.D.: Fast and Private Genomic Testing for Disease Susceptibility, *WPES*, pp.31–34 (2014).
[12] Roden, D.M., Pulley, J.M., Basford, M.A., Bernard, G.R., Clayton, E.W., Balser, J.R. and Masys, D.: Development of a large-scale de-identified DNA biobank to enable personalized medicine, *Clinical Pharmacology & Therapeutics*, Vol.84, No.3, pp.362–369 (2008).
[13] De Cristofaro, E., Faber, S., Gasti, P. and Tsudik, G.: Genodroid: Are privacy-preserving genomic tests ready for prime time?, *WPES*, pp.97–108 (2012).
[14] De Cristofaro, E., Faber, S. and Tsudik, G.: Secure genomic testing with size- and position-hiding private substring matching, *WPES*, pp.107–118 (2013).
[15] Eppstein, D. and Goodrich, M.T.: Straggler identification in round-trip data streams via Newton's identities and invertible Bloom filters, *IEEE Trans. Knowledge and Data Engineering*, Vol.23, No.2, pp.297–306 (2011).
[16] Eppstein, D., Goodrich, M.T. and Baldi, P.: Privacy-enhanced methods for comparing compressed DNA sequences, arXiv preprint arXiv:1107.3593 (2011).
[17] Erlich, Y. and Narayanan, A.: Routes for breaching and protecting genetic privacy, *Nature Reviews Genetics*, Vol.15, No.6, pp.409–421 (2014).
[18] Ayday, E., Raisaro, J.L. and Hubaux, J.-P.: Personal use of the genomic data: Privacy vs. storage cost, *GLOBECOM*, pp.2723–2729 (2013).
[19] Ayday, E., Raisaro, J.L., Hubaux, J.-P. and Rougemont, J.: Protecting and evaluating genomic privacy in medical tests and personalized medicine, *WPES*, pp.95–106 (2013).
[20] Ayday, E., Raisaro, J.L., McLaren, P.J., Fellay, J. and Hubaux, J.-P.: Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data, *HealthTech* (2013).
[21] Ayday, E., Raisaro, J.L., Hengartner, U., Molyneaux, A. and Hubaux, J.-P.: Privacy-preserving processing of raw genomic data, *Data Privacy Management and Autonomous Spontaneous Security*, pp.133–147 (2014).
[22] Turkmen, F., Asghar, M.R. and Demchenko, Y.: iGenoPri: Privacy-Preserving Genomic Data Processing with Integrity and Correctness Proofs, *PST*, pp.407–410 (2016).
[23] Bruekers, F., Katzenbeisser, S., Kursawe, K. and Tuyls, P.: Privacy-preserving matching of DNA profiles, Technical Report (2008).
[24] Fowler, J.H., Settle, J.E. and Christakis, N.A.: Correlated genotypes in friendship networks, *National Academy of Sciences*, Vol.108, No.5, pp.1993–1997 (2011).
[25] Frikken, K.B.: Practical private DNA string searching and matching through efficient oblivious automata evaluation, *Data and Applications Security XXIII*, Vol.5645, pp.81–94 (2009).
[26] Goetz, L.H. and Schork, N.J.: Personalized medicine: Motivation, challenges, and progress, *Fertility and Sterility*, Vol.109, No.6, pp.952–963 (2018).
[27] Ginsburg, G.S. and Willard, H.F.: Genomic and Personalized Medicine: Foundations and Applications, *Translational Research*,

Vol.154, No.6, pp.277–287 (2009).
[28] Hamburg, M.A. and Collins, F.S.: The path to personalized medicine, *New England Journal of Medicine*, Vol.363, No.4, pp.301–304 (2010).
[29] Raisaro, J.L., Choi, G., Pradervand, S., Colsenet, R., Jacquemont, N., Rosat, N., Mooser, V. and Hubaux, J.-P.: Protecting Privacy and Security of Genomic Data in i2b2 with Homomorphic Encryption and Differential Privacy, *IEEE/ACM Trans. Computational Biology and Bioinformatics*, Vol.15, No.5, pp.1413–1426 (2018).
[30] Jorde, L.B. and Wooding, S.P.: Genetic variation, classification and 'race', *Nature Genetics*, Vol.36, p.S28 (2004).
[31] Troncoso-Pastoriza, J.R., Katzenbeisser, S. and Celik, M.: Privacy preserving error resilient DNA searching through oblivious automata, *CCS*, pp.519–528 (2007).
[32] Kamm, L., Bogdanov, D., Laur, S. and Vilo, J.: A new way to protect privacy in large-scale genome-wide association studies, *Bioinformatics*, Vol.29, No.7, pp.886–893 (2013).
[33] Lai, E.: Application of SNP technologies in medicine: Lessons learned and future challenges, *Genome Research*, Vol.11, No.6, pp.927–929 (2001).
[34] Barman, L., Elgraini, M.-T., Raisaro, J.L., Hubaux, J.-P. and Ayday, E.: Privacy Threats and Practical Solutions for Genetic Risk Tests, *SPW*, pp.27–31 (2015).
[35] Whirl-Carrillo, M., McDonagh, E.M., Hebert, J.M., Gong, L., Sangkuhl, K., Thorn, C.F., Altman, R.B. and Klein, T.E.: Pharmacogenomics knowledge for personalized medicine, *Clinical Pharmacology & Therapeutics*, Vol.92, No.4, pp.414–417 (2012).
[36] Das, M., Mozumder, N.J., Afrose, S., Akbar, K.A. and Hashem, T.: A Novel Secret Sharing Approach for Privacy-Preserving Authenticated Disease Risk Queries in Genomic Databases, *COMPSAC*, pp.645–654 (2018).
[37] Humbert, M., Ayday, E., Hubaux, J.-P. and Telenti, A.: Addressing the Concerns of the Lacks Family: Quantification of Kin Genomic Privacy, *CCS*, pp.338–347 (2013).
[38] Mayer, A.N. et al.: A timely arrival for genomic medicine, *Genet Med*, Vol.13, No.3, pp.195–196 (2011).
[39] Aziz, M.M.A., Hasan, M.Z., Mohammed, N. and Alhadidi, D.: Secure and Efficient Multiparty Computation on Genomic Data, *IDEAS*, pp.278–283 (2016).
[40] Merkle, R.C. and Hellman, M.E.: On the security of multiple encryption, *Comm. ACM*, Vol.24, No.7, pp.465–467 (1981).
[41] Naveed, M., Ayday, E., Clayton, E.W., Fellay, J., Gunter, C.A., Hubaux, J.-P., Malin, B.A. and Wang, X.: Privacy in the genomic era, *Computing Surveys*, Vol.48, No.1, pp.1–44 (2015).
[42] Kantarcioglu, M., Jiang, W., Liu, Y. and Malin, B.: A cryptographic approach to securely share and query genomic sequences, *IEEE Trans. Information Technology in Biomedicine*, Vol.12, No.5, pp.606–617 (2008).
[43] McLaren, P.J., Raisaro, J.L., Aouri, M., Rotger, M., Ayday, E., Bartha, I., Delgado, M.B., Vallet, Y., Günthard, H.F., Cavassini, M., Furrer, H., Doco-Lecompte, T., Marzolini, C., Schmid, P., Di Benedetto, C., Decosterd, L.A., Fellay, J., Hubaux, J.P. and Telenti, A.: Privacy-preserving genomic testing in the clinic: A model using HIV treatment, *Genetics in medicine*, Vol.18, No.8, p.814 (2016).
[44] Bohannon, P., Jakobsson, M. and Srikwan, S.: Cryptographic Approaches to Privacy in Forensic DNA Databases, *Proc. 2000 International Workshop on Practice and Theory in Public Key Cryptography*, pp.373–390 (2000).
[45] Philipson, T.J.: The Economic Value and Pricing of Personalized Medicine, *Economic Dimensions of Personalized and Precision Medicine* (2018).
[46] Baldi, P., Baronio, R., De Cristofaro, E., Gasti, P. and Tsudik, G.: Countering GATTACA: Efficient and secure testing of fully-sequenced human genomes, *CCS*, pp.691–702 (2011).
[47] Laing, R.E., Hess, P., Shen, Y., Wang, J. and Hu, S.X.: The role and impact of SNPs in pharmacogenomics and personalized medicine, *Current Drug Metabolism*, Vol.12, No.5, pp.460–486 (2011).
[48] Rotger, M. et al.: Contribution of Genetic Background, Traditional Risk Factors, and HIV-Related Factors to Coronary Artery Disease Events in HIV-Positive Persons, *Clinical Infectious Diseases*, Vol.57, No.1, pp.112–121 (2013).
[49] Wang, R., Wang, X.F., Li, Z., Tang, H., Reiter, M.K. and Dong, Z.: Privacy-preserving genomic computation through program specialization, *CCS*, pp.338–347 (2009).
[50] Schneider, T. and Tkachenko, O.: Towards Efficient Privacy-Preserving Similar Sequence Queries on Outsourced Genomic Databases, *Proc. 2018 Workshop on Privacy in the Electronic Society*, pp.71–75 (2018).
[51] Constable, S.D., Tang, Y., Wang, S., Jiang, X. and Chapin, S.: Privacy-preserving GWAS analysis on federated genomic datasets, *BMC Medical Informatics and Decision Making*, Vol.15, No.5, p.S2 (2015).
[52] Wandelt, S., Bux, M. and Leser, U.: Trends in genomic compression, *Current Bioinformatics*, Vol.9, No.3, pp.315–326 (2013).
[53] Jha, S., Kruger, L. and Shmatikov, V.: Towards practical privacy for genomic computation, *Proc. 2008 IEEE Symposium on Security and Privacy*, pp.216–230 (2008).
[54] Stephen, E., Fienberg, A.S. and Uhler, C.: Privacy preserving GWAS data sharing, *ICDMW*, pp.628–635.
[55] Thomas, B., Pedersen, Y.S. and Savas, E.: Secret sharing vs. encryption-based techniques for privacy preserving data mining, *Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality* (2007).
[56] Yang, Y. et al.: Clinical whole-exome sequencing for the diagnosis of mendelian disorders, *New England Journal of Medicine*, Vol.369, No.16, pp.1502–1511 (2013).
[57] Chen, Y., Peng, B., Wang, X.F. and Tang, H.: Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds, *NDSS* (2012).
[58] Zhang, Y., Blanton, M. and Almashaqbeh, G.: Secure distributed genome analysis for GWAS and sequence comparison computation, *BMC Medical Informatics and Decision Making*, Vol.15, No.5, p.S4 (2015).

Nusrat Jahan Mozumder did B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2017. She is currently pursuing M.Sc. in Computer Science and Engineering from BUET. She is also working as a faculty member in Department of Computer Science and Engineering of Notre Dame University Bangladesh. Her research interests include machine learning, data security and bioinformatics.

Maitraye Das is currently pursuing joint PhD in Computer Science and Communication Studies at Northwestern University, USA. Before that, she received her B.Sc. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 2015. Her current research interest falls broadly in the intersection of Human-Computer Interaction, Accessibility, Assistive Technology and Social Computing. She is a member of the ACM, SIGCHI and SIGACCESS.

Tanzima Hashem is a professor at the Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology (BUET). She received her Ph.D. degree in Computer Science and Software Engineering from the University of Melbourne, Australia in 2012. Her research interest falls in the area of ubiquitous computing, spatial databases, GIS and privacy. In 2017, she received the prestigious OWSD-Elsevier Foundation Award for Early-Career Women Scientists in the Developing World in Engineering Sciences.



Sharmin Afrose received her B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 2015. Currently, she is a graduate student of the Department of Computer Science at Virginia Polytechnic Institute and State University. Her research interests include software security, data mining and machine learning.

Khandakar Ashrafi Akbar is a faculty member in the Department of Computer Science and Engineering of Northern University Bangladesh. Prior to that, she was in Reve System as a Junior Software Engineer. She received B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2017. Her current research interest falls broadly in Data Mining, Machine Learning, Cognitive Science and Artificial Intelligence.