Protecting Personal Genome Privacy: Solutions from Information Security

Erman Ayday[1], Mathias Humbert[1], Jacques Fellay[2,3], Paul J. McLaren[2,3], Jacques Rougemont[2,4], Jean Louis Raisaro[1], Amalio Telenti[3,†,‡] and Jean-Pierre Hubaux[1,†,‡]

[†]To whom correspondence should be addressed (jean-pierre.hubaux@epfl.ch, amalio.telenti@chuv.ch)

[‡]Co-senior authors

[1]School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
[2]School of Life Sciences, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland
[3]Institute of Microbiology, University Hospital and University of Lausanne, Lausanne, Switzerland
[4]Swiss Institute of Bioinformatics, Lausanne, Switzerland

Genomic medicine holds great promise for personalized care; however, it comes with a risk to privacy. In order to maximize the benefit, detailed information about individual genomes will need to be accessed by various healthcare providers, thus increasing the potential for such sensitive data to be accessed by a malicious third party. Ensuring that only those experts that need access to such data can obtain it, and that they themselves can only access the subset of data they require, is an open area of research. In this perspective, we present techniques stemming from information security, to illustrate how such data can be protected while still allowing accurate and easy access for authorized individuals.

There has been intense attention to the ethical, legal and organizational issues related to human genomic research. The emphasis has been on issues of subject re-identification in anonymous genetic databases, and aggregate data analysis in studies involving large samples. Here, we focus on the protection of genomic information - privacy control - at the level of the individual customer (or patient), in a post-research setting, in the field of genomic medicine.

Privacy control is defined as the ability of individuals to determine when, how, and to what extent information about themselves is revealed to others. Genomics appears as the next significant challenge for privacy control. As a result of the fast evolution in genomic research, substantial progress is expected in terms of better predictive medicine and improved diagnosis. Medically indicated genome sequencing is developing at a rapid pace, notably in oncology, to the point that it could be routinely used in the near future. On the other hand, direct-to-consumer (DTC) genomic companies offer genome profiling services that eliminate the classic physician/patient conduit concerning medical data: genome-wide genotyping now costs less than $200, and the price of a complete genome sequence is quickly decreasing.

The widespread availability of genomic information is an opportunity for interested individuals and for medicine, but the impact on privacy is unprecedented. The companies and hospitals that perform DNA sequencing have to store the genomic data of their customers and patients. Of course, tight legislation regulates their activities, but it is difficult for them to protect themselves against the misdeeds of a hacker or a disgruntled employee. Indeed, in 2007, Vint Cerf (one of the fathers of the Internet and VP at Google) stated that 25% of all online computers are compromised. This means that among 600 million computers that were connected to the Internet at that time, around 150 million were part of botnets, and in most cases, the owners of these computers were unaware of this fact. Even today, this situation has not been improved much. On the other hand, in a non-adversarial scenario, making use of genomic data requires facilitated access to data by legitimate professionals (e.g., physicians and pharmacists). Therefore, new architectures, databases (e.g., Hippocratic databases [1]) and Privacy-Enhancing Technologies (PETs) are needed to store, manage and process genomic data while still enabling its utilization by the healthcare providers.

PETs generally protect users' privacy by either breaking the link between individuals' identities and the data they provide (e.g., removing user's identities from published genomic data), or by decreasing the information provided (e.g., by using cryptographic tools or obfuscation techniques). However, both techniques may reduce the reliability and the accuracy of the interpretation of the genomic data. Thus, it would be of great benefit to develop technologies that protect the privacy of a users' genomic data while preserving its reliability (i.e., enabling accurate test results without disclosing more information than is required) and minimizing the complexity of the privacy-preserving algorithms.

Developing PETs for genomic data presents challenges that arise from the architecture of the human genome (existing privacy-preserving methods do not scale to the size necessary to accommodate the large genomic data sets) and from the evolving knowledge in the field of genomics, which produces many new discoveries every year. This accelerated pace of innovation results in the need for significant flexibility in data access, management and interpretation. Encrypting genomic data in such a way as to allow the necessary flexibility is challenging. Furthermore, integration of genomic data with other privacy-sensitive data (e.g., location, ancestry and other Online Social Network - OSN - data) increases privacy risk through the potential for cross-layer attacks.

We can categorize previous research on genomic privacy into three main categories: i) private string searching and comparison (e.g., for paternity test) [2–5], ii) private release of aggregate data [6–10], and iii) private read mapping [11]. Different from these, in this work, we focus on "privacy of conducting medical tests in the framework of personalized medicine using genomic data". In the next section, focusing on leakage of the genomic data, we address the main threats in genomic privacy and their potential consequences. Then, we present our proposed privacy-preserving algorithm in detail.

Threats in Genomic Privacy

The main threats to human genomic data are the identification of the individual the DNA comes from and leakage of genomic data. These types of attacks may reveal privacy-sensitive data about the patient (e.g., predisposition to disease, ethnicity, paternity or filiation, etc.), which can have serious consequences such as denial of access to health insurance, mortgage, education, or employment.

With the evolving technology, a service provider may sequence a patient's genome and share it with other individuals or institutions without the patient's consent. For example, employers may (indirectly) test their employees, insurance companies may obtain the genomes of their clients, or college officials may access the genomes of their students. Such tests may lead to genetic discrimination (e.g., ancestry discrimination or discrimination due to geographic mapping of people). Even though Genetic Information Non-discrimination Act (GINA), which prohibits the use of genomic information in health insurance and employment, attempted to solve some of these problems in the US, these types of laws are very difficult to enforce.

Even more severe, and currently not widely considered, a malicious party may initiate a cross-layer attack by utilizing privacy-sensitive information belonging to a victim retrieved from different sources (e.g., genomic data, location, OSN, etc.), creating the opportunity for a large variety of fraudulent uses of such data. For example, as stated in the Personal Genome Project (PGP) consent form [12], a malicious party may make synthetic DNA of a victim and plant it at a crime scene to falsely accuse him. In this hypothetical situation, the attacker can make his accusation stronger if he has the location patterns of the victim to be blamed, and hence, knows that the victim was close to the crime scene at the time of the crime. Similarly, an attacker can easily learn information on close relatives of a target from OSN data, thus effectively increasing the potential access to privacy sensitive information if the relatives have also been sequenced. That is, even if the victim has perfect privacy on his own genome, if the attacker has access to the DNA sequence of the parents, he can obtain significant information about the victim's DNA sequence.

BOX 1 PAILLIER CRYPTOSYSTEM AND HOMOMORPHIC CRYPTOGRAPHY

We describe a modification of the Paillier cryptosystem (described in detail in [13,14]) for the proposed privacy-preserving scheme.
The public key of patient P is represented as $(n, g, h = g^x)$ with the strong secret key is the factorization of $n = pq$ (where p, q are safe primes), the weak secret key is $x \in [1, n^2/2]$, and g of order $(p-1)(q-1)/2$. Such a g can be easily found by selecting a random $a \in \mathbb{Z}^*_{n^2}$ and computing $g = -a^{2n}$.

Encryption of a message: To encrypt a message $m \in \mathbb{Z}_n$, we first select a random $r \in [1, n/4]$ and generate the ciphertext $E(m, r, g^x) = (T_1, T_2)$ as below:

$$T_1 = g^r \bmod n^2 \quad \text{and} \quad T_2 = h^r(1 + mn) \bmod n^2. \tag{1}$$

Decryption of a message: The message m can be recovered as follows:

$$m = D((T_1, T_2) \mod n^2) = L(T_2/T_1^x), \tag{2}$$

where $L(u) = \frac{(u-1) \bmod n^2}{n}$, for all $u \in \{u < n^2 \mid u = 1 \bmod n\}$.

Homomorphic properties: Assume two messages m_1 and m_2 are encrypted using two different random numbers r_1 and r_2, under the same public key, $(n, g, h = g^x)$, such that $E(m_1, r_1, g^x) = (T_1^1, T_2^1)$ and $E(m_2, r_2, g^x) = (T_1^2, T_2^2)$. Further, assume that k is a constant number. Then, below homomorphic properties are supported by Paillier cryptosystem:
• The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts.

$$D(E(m_1, r_1, g^x) \cdot E(m_2, r_2, g^x)) =$$
$$D(T_1^1 \cdot T_1^2, T_2^1 \cdot T_2^2 \mod n^2) = m_1 + m_2 \mod n. \tag{3}$$

• An encrypted plaintext raised to a constant k will decrypt to the product of the plaintext and the constant.

$$D(E(m_1, r_1, g^x)^k) =$$
$$D((T_1^1)^k, (T_2^1)^k \mod n^2) = km_1 \mod n. \tag{4}$$

These homomorphic operations are conducted at the Trusted Third Party (TTP) to compute the predicted susceptibility of patient P for disease X as will be discussed next.

Proxy encryption: Patient's weak secret key x is randomly divided into two shares: x_1 and x_2 (such that $x = x_1 \oplus x_2$). x_1 is given to the TTP and x_2 is given to the MU. x_2 can be provided to the MU once the patient is registered to the medical unit or through patient's digital ID card using attribute-based cryptography [15]. Further details about the distribution of shares are out of the scope of this paper.
TTP does the homomorphic operations on P's encrypted genetic variants (via P's public key) and obtains the end-result (i.e., predicted disease susceptibility of P for the disease X). Then, TTP partially decrypts the end-result by using its share x_1 and sends the partially decrypted end-result to the MU. Finally, MU completes the decryption process using its share x_2 and recovers the end-result (this process will be discussed in detail in Box 2).

Even though, at this stage, the field of genomics looks generally well-intentioned (i.e., free from serious attacks), it is likely that the above threats will become more serious as the number of sequenced individuals become larger. Such was the case of the Internet that was initially run and used by well-intentioned researchers. However, once it became more

widely used, it became plagued by uncountable attacks such as spyware, viruses, spam, botnets, Denial-of-Service attacks, etc. Therefore, the need to adapt PETs to personal genomic data will only grow with time.

Solutions

Our views in genomic privacy stem from our background in information security [16–20]. Strong analogies can be made between protecting the privacy of users' OSN (or smartphone) data and protecting the privacy of users' genomic data.

Most medical tests and personalized medicine techniques suffer from common privacy threats. For the simplicity of the presentation, in the rest of this section, we will focus on a particular medical test (i.e., computing genetic disease susceptibility).

In a typical disease susceptibility test, a medical unit (MU, i.e, the family doctor, a specialized physician, a pharmacist, or a medical center) wants to check the susceptibility of a patient (P) against a particular disease X. This would be realized by measuring an individual's genotype at one or a combination of disease associated variants [21, 22]. For example, it is reported that 3 genes bearing a total of 10 variants can be queried to analyze a patient's susceptibility for Alzheimer's disease [23]. Our goal is to build a mechanism such that the patient preserves the privacy of his genomic sequence (his variants) while the MU accurately conducts a disease susceptibility test using the genomic data of the patient.

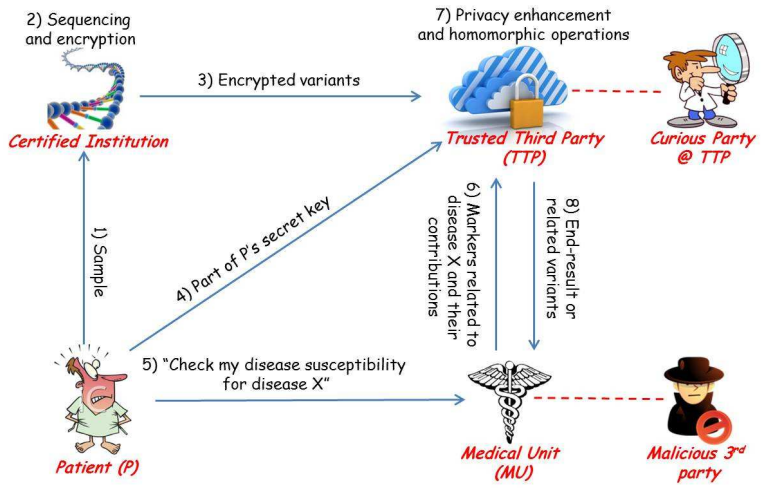
In this study, we consider a malicious MU as the potential attacker trying to obtain private information about the patients. Even if the MU is non-malicious, it is extremely difficult for MUs to protect themselves against the misdeeds of a hacker or a disgruntled employee. Thus, we propose to use a Trusted Third Party (TTP) between the patient and the MU to store and process the genomic data. The TTP can be embodied as a private company (e.g., cloud service provider), the government, or a non-profit organization. Further, we assume that TTP is an honest organization, but it might be curious (e.g., existence of a curious party at TTP), and hence, genomic data should be stored at TTP in encrypted form (i.e., TTP should not be able to access the content of patients' genomic data). We illustrate and summarize our proposed approach for this application in Fig. 1.

We assume that the whole genome sequencing is done by a certified institution with the consent of the patient. Further, the genomic data of the patient is encrypted by the same certified institute (using the patient's public key) and uploaded to the TTP (see Fig. 1 and Box 1) so that only the patient can decrypt the stored variants, and the TTP cannot access the variants of the patient. We emphasize that TTP stores the contents of all the potential variant locations (on the DNA sequence) of the patient instead of only his actual variants. Since the locations of the patient's variants are stored in plaintext, if TTP only stored the actual variants of the patient (e.g., around 4 million SNPs), it could easily determine the genomic sequence of the patient. Therefore, TTP stores the contents of all potential variant locations (around 40 million according to the NCBI dbSNP) of the patient in order to preserve the privacy of the patient. Because the number of known variant sites increases with time, the complete DNA sequence is also encrypted (as a single vector file) and stored at the TTP, and hence, when new variants are discovered, these can be included to the pool of previously stored variants of the patient. Obviously, this increases the storage cost at the TTP, however, this is a requirement for the privacy of the patient. Further, this storage cost can be optimized via using advanced storage techniques (which will not hurt the privacy of the patient). However, for simplicity, we do not dive into the details of such techniques. From now on, we refer to these encrypted potential variants as the patient's encrypted variants (even though not all of them are actual variants, as discussed). We also assume the TTP does not have access to the real identities of the patients and data is stored at the TTP using pseudonyms.

TTP may i) check patient's encrypted variants via homomorphic encryption techniques [13] (Box 1) to compute $\mathbf{Pr}(X)$, the probability that the patient will develop disease X, given his genotype across one or several variants, or ii) provide the relevant variants to the MU, depending on the access rights of the MU. These access rights are defined either jointly by the MU and the patient or by the medical authorities.

Homomorphic encryption allows a potentially curious third-party to perform computations on private information without having direct access to the information itself. In our case, it lets TTP to compute $\mathbf{Pr}(X)$ using encrypted variants of patient P (i.e., without accessing P's variants) via a pre-determined function. In [21], focusing on one example of many diseases which require a susceptibility test involving multiple variants, the authors proposed to count the number of risk alleles across implicated loci (to compute the predicted disease susceptibility). Similarly, in [22], the authors proposed to multiply the Likelihood Ratios (LRs) of the most important variants for a particular disease in order to compute a patient's predicted susceptibility. Further, a *weighted averaging* function can also be used (as a generalization of [21]), which computes the predicted susceptibility by weighting the contributions of variants by their contributions (e.g., LR of the variants). In Box 2, we discuss how to compute the predicted disease susceptibility at the TTP using a toy example to show how the homomorphic encryption is used at the TTP. It is important to note that our proposed privacy preserving mechanism is not limited by the types functions (used to test the disease susceptibility). It is expected that these functions will evolve over time, and hence, the proposed algorithm is developed to keep up with this evolution.

To evaluate the practicality of the proposed privacy-preserving algorithm, we assessed its storage requirements and computational complexity on AMD Opteron 8354 with 2.2 GHz processor under Linux. We assumed that i) the size of the security parameter is 4096 bits, ii) there are 1 million patients in the database, iii) contents of 40 million potential variant (e.g., SNP) locations are stored per patient, and iv) weighted averaging is used at the TTP (as in Box 2). We computed that at the patient side, the individual encryption of all his variants requires 200 minutes, which is a one-time operation and is significantly faster than sequencing (which takes more than a day). Further, the time required to check the disease susceptibility of a single patient at the TTP (using 10 variants) is 1.6 ms., and the decryption of the end result at the MU takes 0.04 ms. Furthermore, conducting a statistical test for one disease (requiring 10 genetic variants) on 1



- (Step 0) Cryptographic keys (public and secret keys) of each patient are generated and distributed to the patients by a trusted entity during the initialization period. We note that distribution, update and revocation of cryptographic keys are handled by a trusted entity (similar to e-banking platforms).
- (Step 1) The patient (P) provides his sample (e.g., his saliva) to the certified institution for sequencing.
- (Step 2) The certified institution sequences P, and encrypts the contents of potential variant locations using P's public key.
- (Step 3) The certified institution sends the encrypted (potential) variants of P to the TTP (so that TTP cannot access to P's variants).
- (Step 4) Patient provides a part of his secret key to the TTP.
- (Step 5) MU wants to conduct a susceptibility test on P for a particular disease X. P provides the other part of his secret key to the MU.
- (Step 6) MU provides genetic variant markers along with their individual contributions (to the disease susceptibility) to the TTP (encrypted with the public key of the TTP).
- (Step 7) If the disease susceptibility can be interpreted by homomorphic operations, TTP computes P's total susceptibility of disease X from the individual effects of (potential) variants using the homomorphic properties of the Paillier cryptosystem as described in Box 1. Otherwise, TTP provides the relevant (potential) variants to the MU based on MU's access rights.
- (Step 7) TTP partially decrypts the end-result (or the relevant variants) using a part of P's secret key following a proxy encryption protocol (discussed in Box 1 and Box 2).
- (Step 8) TTP sends the partially decrypted end-result (or the relevant variants) to the MU.
- (Step 9) MU decrypts the message received from TTP using the other part of P's secret key and recovers the end-result (or the relevant variants).

Figure 1: Privacy-preserving protocol for disease susceptibility test.

million patients takes around 27 minutes at the TTP, and scales linearly with patients and variants. Finally, the storage of all the (encrypted) genetic variants of 1 million patients at the TTP requires 20 GB disk space per patient. All these numbers show that our proposed algorithm is very realistic and could be implemented with current computing technology.

Conclusions and Future Work

While much attention has been paid to the generation, storage, sharing, and protection of collective genome data, the personal use of genome data, by the individual and by accredited personnel, represents a profoundly different scenario. In this paper, we have identified the problem of privacy protection of genomic data once genomics has moved into clinical practice.

BOX 2 COMPUTING DISEASE SUSCEPTIBILITY VIA HOMOMORPHIC OPERATIONS

Here, we discuss how to compute the predicted disease susceptibility at the TTP i) via weighted averaging (which is a generalization of the function proposed in [21]), and ii) via the function proposed in [22] (i.e., multiplication of LR values) using Single Nucleotide Polymorphisms (SNPs) as genetic variants (although the proposed techniques are directly applicable to all types of genetic variation, we refer to SNPs for simplicity).

Weighted Averaging

Assume that (for simplicity) the susceptibility for disease X is determined by only two SNPs in the set $\Omega = \{\text{SNP}_m, \text{SNP}_n\}$, which occur at locations m and n of the DNA sequence. Also assume that the base corresponding to SNP_i at patient P is represented as SNP_i^P and the set of alleles corresponding to SNP_i are $\Lambda_i = \{\lambda_i^1, \lambda_i^2\}$, respectively.

The contributions of the two alleles (λ_i^1 and λ_i^2) for every SNP_i to the susceptibility for disease X are computed via previous studies (on case and control populations) and they are already known by the MU. That is, $\text{p}^i_{\lambda_i^1}(X) \triangleq \Pr(X|\text{SNP}_i^P = \lambda_i^1)$ and $\text{p}^i_{\lambda_i^2}(X) \triangleq \Pr(X|\text{SNP}_i^P = \lambda_i^2)$ ($i \in \{m, n\}$) are determined and known by the MU. Further, the contribution (e.g., OR or LR of the SNPs) of SNP_i to the susceptibility for disease X is denoted by C_i^X.

TTP stores the set of SNPs of patient P, encrypted by P's public key $(n, g, h = g^x)$ using the modified Paillier cryptosystem as discussed in Box 1. Thus, TTP uses $E(\text{SNP}_m^P, g^x)$ and $E(\text{SNP}_n^P, g^x)$ to compute the predicted susceptibility of P for disease X. From now on, we drop the r values in the above encrypted messages for the clarity of the presentation (r values are chosen randomly from the set $[1, n/4]$ for every encrypted message, as discussed in Box 1).

MU provides the following to the TTP in plaintext: i) the markers for disease X (SNP_m and SNP_n) along with their corresponding alleles, ii) corresponding probabilities ($\text{p}^i_{\lambda_i^j}(X)$, $i \in \{m, n\}$ and $j \in \{1, 2\}$), and iii) the contributions of each SNP (C_i^X, $i \in \{m, n\}$). For simplicity, we assume that C_i^X values are normalized (i.e., $C_m^X + C_n^X = 1$). Next, TTP encrypts λ_i^j ($i \in \{m, n\}$ and $j \in \{1, 2\}$) using P's public key in order to conduct the homomorphic computations. Thus, TTP obtains $E(\lambda_m^1, g^x)$, $E(\lambda_m^2, g^x)$, $E(\lambda_n^1, g^x)$, and $E(\lambda_n^2, g^x)$.

TTP computes the encrypted disease susceptibility, $E(\mathbb{S}_P^X, g^x)$, using the homomorphic properties of the Paillier cryptosystem (as discussed in Box 1) using the encrypted SNPs of the patient via weighted averaging as below:

$$E(\mathbb{S}_P^X, g^x) = \Bigg\{ \prod_{i \in \{m,n\}} \Big\{ \Big[E(\text{SNP}_i^P, g^x) \cdot E(\lambda_i^2, g^x)^{-1} \Big]^{\Delta_i^1} \times \Big[E(\text{SNP}_i^P, g^x) \cdot E(\lambda_i^1, g^x)^{-1} \Big]^{\Delta_i^2} \Big\}^{C_i^X} \Bigg\}, \quad (5)$$

where $\Delta_i^1 = \frac{\text{p}^i_{\lambda_i^1}(X)}{\lambda_i^1 - \lambda_i^2}$ and $\Delta_i^2 = \frac{\text{p}^i_{\lambda_i^2}(X)}{\lambda_i^2 - \lambda_i^1}$. We note that the end-result in (5) is encrypted by P's public key and (5) corresponds to the below computation in plaintext:

$$\mathbb{S}_P^X = \sum_{i \in m,n} C_i^X \left\{ \frac{\text{p}^i_{\lambda_i^1}}{(\lambda_i^1 - \lambda_i^2)} \left[\text{SNP}_i^P - \lambda_i^2 \right] + \frac{\text{p}^i_{\lambda_i^2}}{(\lambda_i^2 - \lambda_i^1)} \left[\text{SNP}_i^P - \lambda_i^1 \right] \right\}. \quad (6)$$

Then, TTP partially decrypts the end-result $E(\mathbb{S}_P^X, g^x)$ using its share (x_1) of P's secret key (x) as discussed in Box 1 to obtain $E(\mathbb{S}_P^X, g^{x_2})$ and sends it to MU. Finally, MU decrypts $E(\mathbb{S}_P^X, g^{x_2})$ using its share (x_2) of P's secret key to recover the end-result $\mathbb{S}_P^X$.

In some genetic tests, genotypes of the patients are used instead of single SNPs. In this particular scenario, SNP pairs are considered from both strands of patient's DNA, and hence, there are 3 potential variations (i.e., major, minor and hetero genotype) for each SNP of a patient. In this case, in order to apply the weighted averaging function on encrypted SNPs via homomorphic operations, TTP should store the squared values of the SNPs. That is, for each SNP_i of patient P, TTP should store $E((\text{SNP}_i^P)^2, g^x)$. Depending on the types of genomic tests that would be supported by the TTP (and the functions required for these tests), the format of storage (of patient's SNPs) can be determined beforehand, and SNPs can be stored accordingly just after the sequencing process.

Likelihood Ratio Test

In this approach, the predicted disease susceptibility is computed by multiplying the initial risk of patient (e.g., for disease X) by the LR value of each SNP related to that disease (LR value depends on the value of patient's SNPs) as in [22]. Initial risk of patient P for the disease X is represented as I_X^P, which is determined considering several factors (other than patient's genomic data) such as patient's age, gender, height, weight, environment, etc. Thus, this initial risk can be computed directly by the MU.

We assume that the susceptibility for disease X is determined by the set of SNPs $\Omega = \{\text{SNP}_m, \text{SNP}_n\}$, each with two possible alleles, for simplicity. We denote the LR value due to j^{th} allele ($j \in \{1, 2\}$) of SNP_i for disease X as $L_X^i(\lambda_i^j)$.

TTP stores the SNPs of patient P, encrypted by P's public key. Further, MU sends the following to TTP: i) $L_X^i(\lambda_i^j)$ values ($i \in \{m, n\}$ and $j \in \{1, 2\}$) in plaintext, and ii) the markers for disease X along with their corresponding alleles. MU also encrypts the log of initial risk value, $\ln(I_X^P)$, by P's public key and sends $E(\ln(I_X^P), g^x)$ to TTP. Alternatively, the contribution of initial risk to the disease susceptibility can be included to the end-result at the end, at MU. Next, TTP encrypts λ_i^j ($i \in \{m, n\}$ and $j \in \{1, 2\}$) using P's public key to conduct the homomorphic computations as before.

Thus, TTP computes the predicted susceptibility of patient P for disease X as below:

$$E(\ln(\mathbb{S}_P^X), g^x) = E(\ln(I_X^P), g^x) \times \prod_{i \in m,n} \Bigg\{ \Big[E(\text{SNP}_i^P, g^x) \cdot E(\lambda_i^2, g^x)^{-1} \Big]^{\Xi_i^1} \times \Big[E(\text{SNP}_i^P, g^x) \cdot E(\lambda_i^1, g^x)^{-1} \Big]^{\Xi_i^2} \Bigg\}, \quad (7)$$

where $\Xi_i^1 = \frac{\ln(L_X^i(\lambda_i^1))}{(\lambda_i^1 - \lambda_i^2)}$ and $\Xi_i^2 = \frac{\ln(L_X^i(\lambda_i^2))}{(\lambda_i^2 - \lambda_i^1)}$. Further, (7) corresponds to the below computation in plaintext:

$$\mathbb{S}_P^X = I_X^P \times \prod_{i \in m,n} \left\{ \left[\text{SNP}_i^P - \lambda_i^2 \right] \times \frac{L_X^i(\lambda_i^1)}{(\lambda_i^1 - \lambda_i^2)} + \left[\text{SNP}_i^P - \lambda_i^1 \right] \times \frac{L_X^i(\lambda_i^2)}{(\lambda_i^2 - \lambda_i^1)} \right\}. \quad (8)$$

As before, TTP partially decrypts $E(\ln(\mathbb{S}_P^X), g^x)$ using x_1 (its share of P's secret key) to obtain $E(\ln(\mathbb{S}_P^X), g^{x_2})$ and sends it to MU. Finally, MU decrypts $E(\ln(\mathbb{S}_P^X), g^{x_2})$ using x_2 (its share of P's secret key) to recover $\ln(\mathbb{S}_P^X)$, and computes $e^{(\ln(\mathbb{S}_P^X))}$ to obtain $\mathbb{S}_P^X$. Similar to weighted averaging, if genotypes of the patients are used for the test, squared values of the SNPs should be stored at the TTP for each patient.

The extension of this work opens the doors for various exciting research opportunities in genomic privacy. Complex diseases may require information on dozens to hundreds of variants for a clinically useful determination to be made [24]. On the other hand, releasing too many variants is a risk to the privacy of a person. Thus, it is important to determine the optimal number of variants and the types of variants to be used for the susceptibility test of various diseases. Further, for the diseases that share some susceptibility factors, the adversary may launch an attack such that once he obtains the end-result for low-risk (in terms of privacy) disease, he infers the susceptibility of the patient for higher-risk disease. A good example for this scenario is *ApoE4 variation* which predicts a lesser medical problem, hyperlipidemia, but also Alzheimer's disease. Similarly, patterns of linkage disequilibrium may reveal privacy-sensitive variants even if they are protected by privacy-enhancing algorithms [25]. Furthermore, a variant currently not associated with disease risk may become privacy-sensitive in the future due to ongoing research. Thus, we need to answer how we can smoothly adapt the proposed privacy-preserving algorithms based on new developments in genomics.

Another important research task is to find an optimal point between the accuracy of the medical test, privacy of the patient, and the complexity of the privacy-preserving algorithm at which the privacy and accuracy is maximized and the complexity is minimized. It is worth noting that the proposed privacy-preserving algorithm does not reduce the accuracy of the medical test by obfuscating the genomic data. The number and types of variants that are used in the test (or provided to the MU), and hence the accuracy of the medical test are determined by the patient, considering his privacy loss due to the revealed genomic data.

Finally, it is also important to quantify the level of privacy associated with various applications using genomic data by defining appropriate metrics. In other words, how much information leaks to a third party or to a service provider during a particular medical test that uses genomic data? A tool to quantify the level of genomic privacy would enable a patient to monitor his/her privacy loss due to the medical tests he underwent and to determine the level of future tests (e.g., number and types of genetic variants that will be used in future tests) he will undergo. Tools developed for the quantification of location privacy [17, 18] could be used to develop a framework to quantify genomic privacy.

In summary, we have established a parallel with privacy protection in computer science and described an operational and realistic solution based on state-of-the-art homomorphic encryption. In this way, genomic data is always stored in an encrypted format and each medical unit can access only the subset of genomic data required for healthcare.

Acknowledgements

We would like to thank Vincent Mooser, Didier Trono and Martin Vetterli for their encouragements in this research endeavor. We also owe a special thanks to Keith Harshman for his useful feedback on an earlier version of this paper.

References

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic databases," *Proceedings of the 28th International Conference on Very Large Databases*, 2002.

[2] J. R. Troncoso-Pastoriza, S. Katzenbeisser, and M. Celik, "Privacy preserving error resilient DNA searching through oblivious automata," *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 519–528, 2007.

[3] M. Blanton and M. Aliasgari, "Secure outsourcing of DNA searching via finite automata," *DBSec'10: Proceedings of the 24th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy*, pp. 49–64, 2010.

[4] S. Jha, L. Kruger, and V. Shmatikov, "Towards practical privacy for genomic computation," *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pp. 216–230, 2008.

[5] P. Baldi, R. Baronio, E. De Cristofaro, P. Gasti, and G. Tsudik, "Countering GATTACA: Efficient and secure testing of fully-sequenced human genomes," *CCS '11: Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 691–702, 2011.

[6] N. Homer, S. Szelinger, M. Redman, D. Duggan, and W. Tembe, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genetics*, vol. 4, August 2008.

[7] J. Gitschier, "Inferential genotyping of y chromosomes in latter-day saints founders and comparison to utah samples in the hapmap project," *Am. J. Hum. Genet.*, vol. 84, pp. 251–258, 2009.

[8] X. Zhou, B. Peng, Y. F. Li, Y. Chen, H. Tang, and X. Wang, "To release or not to release: Evaluating information leaks in aggregate human-genome data," *ESORICS'11: Proceedings of the 16th European Conference on Research in Computer Security*, pp. 607–627, 2011.

[9] S. E. Fienberg, A. Slavkovic, and C. Uhler, "Privacy preserving GWAS data sharing," *Proceedings of the IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, dec. 2011.

[10] B. Malin and L. Sweeney, "How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems," *Journal of Biomedical Informatics*, vol. 37, pp. 179–192, June 2004.

[11] Y. Chen, B. Peng, X. Wang, and H. Tang, "Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds," *NDSS'12: Proceeding of the 19th Network and Distributed System Security Symposium*, 2012.

[12] http://www.personalgenomes.org/consent/PGP_Consent_Approved_02212012.pdf.

[13] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," *Proceedings of Asiacrypt 03, LNCS 2894*, pp. 37–54, 2003.

[14] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, pp. 1–30, February 2006.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[16] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy in mobile computing for location-sharing-based services," *Proceedings of the 11th International Conference on Privacy Enhancing Technologies*, pp. 77–96, 2011.

[17] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux, "Quantifying location privacy," *Proceedings of the IEEE Symposium on Security and Privacy*, 2011.

[18] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. L. Boudec, "Quantifying location privacy: The case of sporadic location exposure," *Proceedings of Privacy Enhancing Technologies Symposium (PETS)*, 2011.

[19] J. Freudiger, R. Neu, and J.-P. Hubaux, "Private sharing of user location over online social networks," *Proceedings of HotPETs*, 2010.

[20] I. Bilogrevic, M. Jadliwala, I. Lam, I. Aad, P. Ginzboorg, V. Niemi, L. Bindschaedler, and J.-P. Hubaux, "Big brother knows your friends: On privacy of social communities in pervasive networks," *Proceedings of the 10th International Conference on Pervasive Computing*, 2012.

[21] S. Kathiresan, O. Melander, D. Anevski, C. Guiducci, and N. Burtt, "Polymorphisms associated with cholesterol and risk of cardiovascular events," *The New England Journal of Medicine*, vol. 358, pp. 1240–1249, 2008.

[22] E. Ashley, A. Butte, M. Wheeler, R.Chen, and T. Klein, "Clinical assessment incorporating a personal genome," *The Lancet*, vol. 375, no. 9725, pp. 1525–1535, 2010.

[23] S. Seshadri, A. Fitzpatrick, M.A. Ikram, A. DeStefano, V. Gudnason, M. Boada, J. Bis, A. Smith, M. Carassquillo, J. Lambert, C. Consortium, G. Consortium, and E. Consortium, "Genome-wide analysis of genetic loci associated with alzheimer disease," *JAMA*, vol. 303, pp. 1832–1840, 2010.

[24] J. Quevedo, A. Bahamonde, M. Perez-Enciso, and O. Luaces, "Disease liability prediction from large scale genotyping data using classifiers with a reject option," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 1, pp. 88–97, jan.-feb. 2012.

[25] D. Nyholt, C. Yu, and P. Visscher, "On Jim Watson's APOE status: genetic information is hard to hide," *European Journal of Human Genetics*, vol. 17, pp. 147–149, 2009.

Glossary

Attacker/adversary:	Malicious entity whose objective is to infer some privacy-sensitive information about someone.
Botnet:	Collection of computers compromised by an attacker that uses them to send out spam, spread viruses and attack computers and servers.
Ciphertext:	Information encoded by a cryptographic system.
Cryptographic tools/algorithms:	Sequences of processes, or rules that protect data by making sure that unwanted people cannot access it. Cryptographic algorithms are designed around computational hardness assumptions, making them hard to break in practice by any adversary.
Denial-of-Service:	Attack that makes a computer or network resource unavailable to its intended users (generally launched by a botnet).
Hacker:	User who accesses a computer system by circumventing its security system.
Hippocratic databases:	Database systems that comply with privacy legislations and guidelines, and fulfill the following key principles: purpose specification, consent, limited collection, limited use, limited disclosure, limited retention, accuracy, safety, openness, and compliance.
Plaintext:	Input of an encryption algorithm (usually a cleartext).
Privacy-Enhancing Technologies (PETs):	General term for a set of computer tools, applications and mechanisms which allow users to protect the privacy of their personal information provided to and handled by third party services or applications.
Spyware:	Type of malicious software that collects information about users without their knowledge.