Personal Use of the Genomic Data: Privacy vs. Storage Cost

Erman Ayday
School of Comp. and Comm. Sciences
EPFL, Lausanne, Switzerland
Email: erman.ayday@epfl.ch

Jean Louis Raisaro
School of Comp. and Comm. Sciences
EPFL, Lausanne, Switzerland
Email: jean.raisaro@epfl.ch

Jean-Pierre Hubaux
School of Comp. and Comm. Sciences
EPFL, Lausanne, Switzerland
Email: jean-pierre.hubaux@epfl.ch

***Abstract*—In this paper, we propose privacy-enhancing technologies for personal use of the genomic data and analyze the tradeoff between genomic privacy and storage cost of the genomes. First, we highlight the potential privacy threats on the genomic data. Then, focusing specifically on a disease-susceptibility test, we develop a new architecture (between the patient and the medical unit) and propose a privacy-preserving algorithm by utilizing homomorphic encryption. Assuming the whole genome sequencing is done by a certified institution, we propose to store patients' genomic data encrypted by their public keys at a Storage and Processing Unit (SPU). The proposed algorithm lets the SPU process the encrypted genomic data for medical tests while preserving the privacy of patients' genomic data. We extensively analyze the relationship between the storage cost (of the genomic data), the level of genomic privacy (of the patient), and the characteristics of the genomic data. Furthermore, we show via a complexity analysis the practicality of the proposed scheme.**

Privacy control can be defined as the ability of individuals to determine when, how, and to what extent information about themselves is revealed to others. In this way, the usage of private data will remain in context and it will be used exclusively for the purpose the data owner has in mind. Privacy is usually protected by both legal and technological means. By using legal actions, such as data protection directives and fair information practices, privacy regulations can enforce privacy protection on a large scale. Yet, this approach is mostly reactive, as it defines regulations after technologies are put in place. To avoid this issue, Privacy-Enhancing Technologies (PETs) [1] can be incorporated into the design of new systems in order to protect individuals' data. PETs protect privacy by eliminating or obfuscating personal data, thereby preventing misuse or involuntary loss of data, without affecting the functionality of the information system. Their objective is to make it difficult for a malicious entity to link information to specific users.

Genomics is becoming the next significant challenge for privacy [2]. The price of a complete genome profile has dropped below $100 for genome-wide genotyping (i.e., the characterization of about one million common genetic variants), which is offered by a number of companies. Whole genome sequencing is also offered through the same direct-to-consumer model (but at a higher price). This low cost of DNA sequencing will break the physician/patient connection, because private citizens (from anywhere in the world) can have their genome sequenced without involving their family doctor. This can open the door to all kinds of abuse, not yet fully understood. 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. Even though the 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.

In this work, our goal is to protect the privacy of patients' genomic data while (i) enabling medical units to access the genomic data in order to conduct medical tests, and (ii) providing efficient storage of the genomic data. In a medical test, a medical center checks for different health risks (e.g., disease susceptibilities) of a patient by using specific parts of his genome. In order to preserve his privacy, the patient does not want to reveal his complete genome to the medical center. To achieve this goal, we propose to store the genomic data at a *Storage and Processing Unit* (SPU) and conduct the computations on the genomic data utilizing homomorphic encryption.

Medical tests (which use genomic data) are usually conducted by analyzing the variants (i.e., nucleotides which reside at particular positions in the genome and vary between individuals) of the patients. Current discoveries show that there are around 40 million variants in human population, however, this number keeps increasing with new discoveries. Thus, the variants of the patients should be stored in an efficient way to minimize the storage cost at the SPU. At the same time, the variants of a patient should be securely stored in such a way that the SPU would not be able to infer their contents (to preserve the genomic privacy of the patient). Therefore, there is a tradeoff between the privacy and the storage, hence we extensively analyze this tradeoff for our proposed system by also considering the characteristics of the genomic data.

The rest of the paper is organized as follows. In Section I, we summarize the related work on genomic privacy. In Section II, we describe our proposed scheme for privacy-preserving medical tests. Next, in Section III, we analyze the tradeoff between the privacy and the storage cost of the genomic data for different design and genomic criterion. In Section IV, we present the complexity evaluation of the proposed scheme. Finally, in Section V, we conclude the paper.

I. Related Work

We can put the research on genomic privacy in three main categories: (i) private string searching and comparison, (ii) private release of aggregate data, and (iii) private clinical genomics. Our proposed work is closest to the efforts on private string searching and comparison.

In [3], Troncoso-Pastoriza *et al.* propose a protocol for string searching, which is then re-visited by Blanton and Aliasgari [4]. In this approach, one party with his own DNA snippet can verify the existence of a short template within his snippet by using a Finite State Machine in an oblivious manner. To compute the similarity of DNA sequences, in [5], Jha *et al.* propose techniques for privately computing the edit distance of two strings by using garbled circuits. In [6], Bruekers *et al.* propose privacy-enhanced comparison of DNA profiles for identity, paternity and ancestry tests using homomorphic encryption. Similar to our work, in [7], Kantarcioglu *et al.* propose using homomorphic encryption to perform scientific investigations on integrated genomic data. As opposed to [7], we focus on personal use of the genomic data (e.g., in medical tests). In one of the recent works [8], Baldi *et al.* make use of medical tools and private string comparison for privacy-preserving paternity tests, personalized medicine,

and genetic compatibility tests. Instead of utilizing public key encryption protocols, in [9], Canim *et al.* propose securing the biomedical data using cryptographic hardware. Furthermore, we propose privacy-preserving schemes for medical tests and personalized medicine methods that use patients' genomic data [10], [11]. We also propose privacy-preserving techniques for the management of raw genomic data [12] and techniques to quantify kin genomic privacy [13].[1]

As a result of our extensive collaboration with geneticists, clinicians, and biologists, we conclude that DNA string comparison (in which the medical unit can only check if the patient carries a specific combination of variants or not) is insufficient in many medical tests (that use genomic data). As it will become clearer in the next sections, specific variants must be considered individually for each genetic test. Thus, as opposed to the above private string search and comparison techniques, we use the individual variants of the patients to conduct genetic disease susceptibility tests. Further, in our proposed scheme, we consider the statistical relationship between the variants to provide efficient storage of the genomic data while still protecting the genomic privacy of the patients.

II. Privacy-Preserving Personal Use of the Genomic Data in Medical Tests

Most medical tests (that use genomic data) involve a patient and a medical unit. In general, the medical unit is the family doctor, a physician, a pharmacist, a medical council, or an online service. In this study, we consider a malicious medical unit as the potential attacker. That is, a medical unit can be a malicious institution trying to obtain private genomic information about a patient (for which it is not authorized). Even if the medical unit is non-malicious, it is extremely difficult for medical units to protect themselves against the misdeeds of a hacker or a disgruntled employee, hence the attacker can also be considered as a hacker or a careless employee in the medical unit. Similarly, the genomic data is too sensitive to be stored on patients' personal devices (mostly due to security, availability, and storage issues), hence it is risky to leave the patients' genomic data in their own hands. In addition, extreme precaution is needed between the patient and the medical unit due to the sensitivity of the genomic data. Thus, we believe that a Storage and Processing Unit (SPU) should be used to store and process the genomic data.[2] We assume that the SPU is an honest organization, but it might be curious (e.g., existence of a curious party at the SPU), hence the genomic data should be stored at the SPU in encrypted form (i.e., the SPU should not be able to access the content of patients' genomic data). We also assume the SPU does not have access to the real identities of the patients and data is stored at the SPU by using pseudonyms; this way, the SPU cannot associate the conducted genomic tests to the real identities of the patients. This general architecture is illustrated in Fig. 1.

For the simplicity of presentation, in the rest of this work, we will focus on a particular medical test (namely, computing genetic disease susceptibility). We note that similar techniques would apply for other medical tests and personalized medicine methods. In a typical disease-susceptibility test, a medical center (MC) wants to check the susceptibility of a patient (P) to a particular disease X (i.e., probability that the patient P will develop disease X). It is shown that a genetic disease-susceptibility test can be realized by analyzing particular Single Nucleotide Polymorphisms (SNPs) of the patient via some operations, such as weighted averaging [14] or Likelihood Ratio (LR) test [15].

[1]More information about our activities in this field can be found at: http://lca.epfl.ch/projects/genomic-privacy/.

[2]A private company (e.g., cloud storage service), the government, or a non-profit organization could play the role of the SPU.

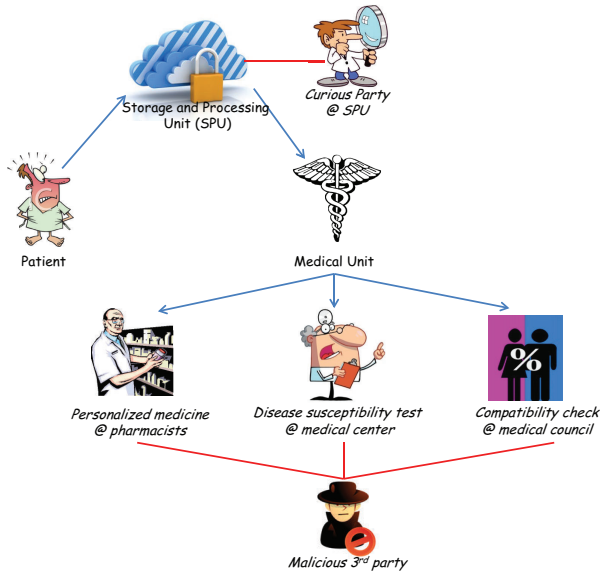

Fig. 1. General architecture between the patient, SPU, and the medical unit.

A SNP is a position in the genome holding a nucleotide (A, T, C or G), which varies between individuals. Each SNP contributes to the susceptibility in a different amount and the contribution amount of each SNP is determined by previous studies on case and control groups (these studies are published in several papers).

In general, there are two alleles (nucleotides which reside at a SNP position) observed at a given SNP position: (i) The major allele is the most frequently observed nucleotide, and (ii) the minor allele is the rare nucleotide. Everyone inherits one allele of every SNP position from each of his parents. If an individual receives the same allele from both parents, he is said to have a *homozygous* variant for that SNP position. If, however, he inherits a different allele from each parent (one minor and one major), he has a *heterozygous* variant. There are approximately 40 million approved SNPs in the human population as of now (according to the NCBI dbSNP [16]) and each patient carries on average 4 million SNPs (e.g., variants) out of this 40 million. Moreover, this set of 4 million SNPs is different for each patient. From now on, to avoid confusion, for each patient, we refer to these 4 million variants as the *real SNPs* and the remaining non-variants (approved SNPs that do not exist for the considered patient) as the *potential SNPs* of the patient; when we only say "SNPs", we mean both the real and potential SNPs.

A potential attacker can learn about the susceptibilities of the patient to privacy-sensitive diseases if he obtains some specific real SNPs of the patient. Moreover, the knowledge of 75 real SNPs (out of approximately 4 million), if not fewer, will enable the attacker to identify a person [17]. Thus, our goal is to build a mechanism in which the patient can preserve the privacy of his genomic sequence (his real SNPs) while enabling the MC to access his genomic data and conduct genetic tests. In the rest of this work, for simplicity of the presentation, we do not consider the type of the variant at a real SNP position (i.e., whether the variation is homozygous or heterozygous for that real SNP); we only consider whether the patient has a real SNP or not at a particular position. However, the proposed approaches and the analysis (in Section III) can easily be extended to cover the types of the variants.

In each disease susceptibility test, depending on the access rights of the MC, the SPU can either (i) compute $\Pr(X)$, the probability that the patient will develop the disease X, by checking the patient's encrypted SNPs via homomorphic encryption techniques [18], or (ii) provide the relevant SNPs to the MC (e.g., for complex diseases that cannot be interpreted

using homomorphic operations). These access rights are defined either jointly by the MC and the patient or by the medical authorities. We note that homomorphic encryption lets the SPU compute $\Pr(X)$ using encrypted SNPs of the patient P. In other words, the SPU does not access P's SNPs to compute his predicted disease susceptibility. We use a modification of the Paillier cryptosystem (described in Section II-A) to support the homomorphic operations at the SPU.

A. Paillier Cryptosystem

In this section, we briefly review the modified Paillier cryptosystem (described in detail in [18], [19]), which we use in this work, and its homomorphic properties.

The public key of the patient P is represented as $(n, g, h = g^x)$, where the strong secret key is the factorization of $n = pq$ (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 pair (T_1, T_2) as below:

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

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

$$m = \Lambda(T_2/T_1^x), \tag{2}$$

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

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 $\mathrm{E}(m_1, r_1, g^x) = (T_1^1, T_2^1)$ and $\mathrm{E}(m_2, r_2, g^x) = (T_1^2, T_2^2)$. Assume also that c is a constant number. Then the below-mentioned homomorphic properties are supported by Paillier cryptosystem:

• The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts.

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

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

$$\begin{aligned}\mathrm{D}(\mathrm{E}(m_1, r_1, g^x)^c) &= \mathrm{D}((T_1^1)^c, (T_2^1)^c \mod n^2) \\ &= cm_1 \mod n.\end{aligned} \tag{4}$$

Proxy re-encryption: The patient's weak secret key x is randomly divided into two shares: $x^{(1)}$ and $x^{(2)}$ (such that $x = x^{(1)} + x^{(2)}$). $x^{(1)}$ is given to the SPU and $x^{(2)}$ is given to the MC. Using the above Paillier cryptosystem, an encrypted message (T_1, T_2) (under the patient's public key) can be partially decrypted by the SPU (using $x^{(1)}$) to generate the ciphertext pair $(\tilde{T_1}, \tilde{T_2})$ as below:

$$\tilde{T_1} = T_1 \quad \text{and} \quad \tilde{T_2} = T_2/T_1^{x^{(1)}} \mod n^2. \tag{5}$$

Now, $(\tilde{T_1}, \tilde{T_2})$ can be decrypted at the MC using $x^{(2)}$ to recover the original message.

B. Proposed Solution

Even though the contents of the SNPs are stored encrypted (via the patient's public key), we assume that the positions (or IDs) of the corresponding SNPs (on the DNA sequence) are stored in plaintext at the SPU. This is because, when a particular SNP (or set of SNPs) are queried by the MC, the SPU should know which SNPs to process (or send to the MC) without the involvement of the patient in the protocol.[3] More importantly, the SPU needs to see the positions (or IDs) of the requested SNPs by the MC in order to check whether the MC holds the required access rights for the corresponding SNPs of the patient.

We assume that the type of SNP_i (i.e., SNP whose ID is i) at the patient P is represented as SNP_i^P and $\mathrm{SNP}_i^P = 1$, if P has a real SNP (i.e., variant) at this position, and $\mathrm{SNP}_i^P = 0$, if P does not have a variant at this position. We let Υ_P be the set of real SNPs of the patient P (at which $\mathrm{SNP}_i^P = 1$). We also let Ω_P represent the set of potential SNPs (at which $\mathrm{SNP}_i^P = 0$).

As the positions of the SNPs are stored in plaintext, if the SPU only stores the real SNPs in Υ_P, a curious party at the SPU can learn all real SNP positions of the patient, hence much about his genomic sequence.[4] To avoid this, a trivial solution is to let the SPU store the contents of both real and potential SNP positions (in $\{\Upsilon_P \cup \Omega_P\}$) in order to preserve the privacy of the patient. However, this trivial solution causes a significant storage cost (which is projected to increase as the number of discovered SNPs keeps increasing with new discoveries in the field of genomics). Thus, we propose another technique that reduces the storage cost at the SPU at the expense of decrease in privacy. In a nutshell, instead of storing the contents of all potential and real SNP positions, we store the real SNPs of the patient along with a certain level of redundancy (i.e., contents of some potential SNP positions). In other words, to mislead a curious party at the SPU, among the 40 million discovered SNPs, we store the approximately 4 million real SNPs (for which $\mathrm{SNP}_i^P = 1, i \in \Upsilon_P$) along with some redundant content from Ω_P (with $\mathrm{SNP}_j^P = 0$), for each patient.

An important privacy issue to consider at this point is the *Linkage Disequilibrium* (LD) between SNPs [20]. LD occurs when SNPs at the two loci (SNP positions) are not independent of each other. Using the LD relationships between the stored and un-stored SNPs, a curious party at the SPU might infer the contents of the stored SNPs from the un-stored ones (by using the fact that an un-stored SNP j is a potential SNP of the patient with $\mathrm{SNP}_j^P = 0$). This causes a tradeoff between the privacy and the storage cost of the genomic data. We discuss this tradeoff in detail in Section III. Below, we summarize the proposed approach for the privacy-preserving disease-susceptibility test. This approach is illustrated in Fig. 2.

• **Step 0:** The cryptographic keys (public and secret keys) of each patient are generated and distributed to the patients during the initialization period. Then, symmetric keys are established between the parties, using which the communication between the parties is protected from an eavesdropper. We note that the 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 (CI) for sequencing.

• **Step 2:** The CI sequences P with the consent of the patient. Let Ω_P^s and Ω_P^u denote the set of P's potential SNPs that will be

[3] Generally, the involvement of the patient is not desirable during the interaction between the MC and the SPU.

[4] The nucleotides corresponding to variants at particular positions of the DNA sequence are public knowledge. Thus, even though the contents of patient's real SNPs are encrypted, a curious party at the SPU can infer the nucleotides corresponding to these SNPs from their plaintext positions (or IDs).

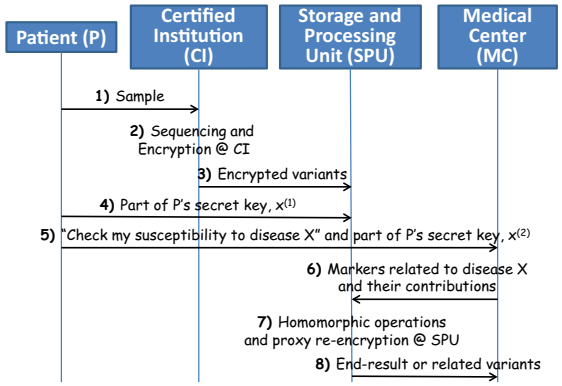

Fig. 2. Privacy-preserving protocol for disease-susceptibility test.

stored and not stored at the SPU, respectively ($\Omega_P^s \cup \Omega_P^u = \Omega_P$). Then, the CI encrypts the contents of P's real and potential SNPs (in $\{\Upsilon_P \cup \Omega_P^s\}$) by using P's public key. We are aware that the number of discovered SNPs increases with time. Thus, the patient's complete DNA sequence is also encrypted as a single vector file (via symmetric encryption using the patient's key) and stored at the SPU, thus when new SNPs are discovered, these can be included in the pool of the previously stored SNPs of the patient.

• **Step 3:** The CI sends the encrypted SNPs of P to the SPU along with their plaintext IDs (so that the SPU cannot access the contents of P's SNPs).

• **Step 4:** The patient provides a part of his secret key ($x^{(1)}$) to the SPU.

• **Step 5:** The MC wants to conduct a susceptibility test on P to a particular disease X, and P provides the other part of his secret key ($x^{(2)}$) to the MC.

• **Step 6:** The MC provides genetic variant markers, along with their individual contributions (to the disease susceptibility), to the SPU.

• **Step 7:** If the disease susceptibility can be interpreted by homomorphic operations, the SPU computes P's total susceptibility to disease X from the individual effects of SNPs by using the homomorphic properties of the Paillier cryptosystem as described next. Otherwise, the SPU provides the relevant SNPs to the MC based on MC's access rights. We note that if a particular potential SNP_j (requested by the MC or needed in the susceptibility test) is not stored at the SPU (i.e., $\mathrm{SNP}_j \in \Omega_P^u$), one of the following two scenarios occurs: (i) If the SPU provides the relevant SNPs to the MC, MC infers the missing potential SNPs from the reference genome (since it is known that the missing potential SNPs are not variants for P), or (ii) if the SPU provides the end-result of the susceptibility test, the SPU uses the fact that $\mathrm{SNP}_j^P = 0$ for each missing potential SNP j.

In the following, we discuss how to compute the predicted disease susceptibility at the SPU by using the function proposed in [15] (i.e., multiplication of LR values) and show how the predicted susceptibility is computed using encrypted SNPs.[5] The predicted disease susceptibility is computed by multiplying the initial risk of the patient (e.g., for disease X) by the LR value of each SNP related to that disease (LR value of a SNP i depends on the value of SNP_i^P at patient P). The initial risk of patient P for the disease X is represented as I_X^P. We note that I_X^P is determined by considering several factors (other than patient's genomic data) such as the patient's age, gender, height, weight, and environment. Thus, this initial risk can be computed directly by the MC.

We assume that the susceptibility for disease X is determined by the set of SNPs in set φ_X. We denote the LR values due to $\mathrm{SNP}_i^P = 0$ and $\mathrm{SNP}_i^P = 1$ for disease X as $L_X^i(0)$ and $L_X^i(1)$, respectively. The SPU receives the following from the MC (in Step 6): (i) $L_X^i(j)$ values ($i \in \varphi_X$ and $j \in \{0,1\}$) in plaintext, and (ii) the IDs of the SNPs (in φ_X) that are related to disease X. The MC also encrypts the log of initial risk value, $\ln(I_X^P)$, by P's public key and sends $\mathrm{E}(\ln(I_X^P), g^x)$ to the SPU.[6] Next, the SPU encrypts j ($j \in \{0,1\}$) using P's public key to obtain $\mathrm{E}(0, g^x)$ and $\mathrm{E}(1, g^x)$ for the homomorphic computations.

The predicted susceptibility of P for disease X ($\mathbb{S}_P^X$) can be computed (using the likelihood ratio test) in plaintext as below:

$$\mathbb{S}_P^X = I_X^P \times \prod_{i \in \varphi_X} \left\{ \left[\mathrm{SNP}_i^P - 1\right] \times -L_X^i(0) + \left[\mathrm{SNP}_i^P - 0\right] \times L_X^i(1) \right\}. \quad (6)$$

The Paillier cryptosystem does not support multiplicative homomorphism in ciphertext; it only supports the multiplication of a ciphertext with a constant, as discussed in Section II-A. Thus, instead of multiplying the LR values, we use addition in log-domain at the SPU. Therefore, the SPU computes the predicted susceptibility of P for disease X, $\mathrm{E}(\ln(\mathbb{S}_P^X), g^x)$, in log-domain by using $\ln(L_X^i(j))$ values ($i \in \varphi_X$ and $j \in \{0,1\}$) and the homomorphic properties of the Paillier cryptosystem (Section II-A).

In some genetic tests, the types of the real SNPs (e.g., homozygous or heterozygous) become also important. In this case, SNP_i^P can take three different values from the set $\{0,1,2\}$ to represent a potential SNP (i.e., non-variant), a real homozygous SNP, and a real heterozygous SNP, respectively. In such a scenario, to conduct the disease-susceptibility test via homomorphic operations, the SPU should store the squared values of the SNPs. That is, for each SNP_i^P of the patient P, the SPU should store $\mathrm{E}((\mathrm{SNP}_i^P)^2, g^x)$. Depending on the types of genomic tests that would be supported by the SPU (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.

Finally, the SPU partially decrypts the end-result (or the relevant SNPs) using $x^{(1)}$ (its share of P's secret key) following a proxy re-encryption protocol (Section II-A).

• **Step 8:** The SPU sends the partially decrypted end-result (or the relevant SNPs) to the MC.

• **Step 9:** The MC decrypts the message received from the SPU using $x^{(2)}$ (its share of P's secret key) and recovers the end-result (or the relevant SNPs).

The above technique provides a high practicality for the patient, because the patient is not involved in the protocol after the sequencing (except for the consent between the patient and the MC for a particular test). We note that the proposed scheme preserves the privacy of patients' genomic data relying on the security strength of modified Paillier cryptosystem (the extensive security evaluation of the modified Paillier cryptosystem can be found in [18]).

[5]The function in [14] can be also utilized similarly.

[6]From now on, we drop the r values in the 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 Section II-A).

III. Privacy Analysis

As expected, the amount of storage redundancy (due to the storage of the SNPs in Ω_P^s), along with the LD between the SNPs and their characteristics, determine the level of a patient's genomic privacy. Therefore, in the rest of this section, we analyze the relationship between the amount of redundancy (i.e., storage cost), LD values, characteristics of the SNPs, and the level of privacy. To do so, first, we observe the average probability of correctly inferring the positions of P's real SNPs (in Υ_P) considering varying amounts of redundancy and the LD values between the SNPs. That is, how much information would a patient's un-stored potential SNPs reveal about the positions of his real SNPs to the curious party at the SPU? This problem can also be formulated similarly if the goal of the attacker is to determine the type of the variant at a real SNP position (e.g., homozygous or heterozygous). It is worth noting that for this study, we create realistic models for the LD values and the characteristics of the SNPs. Further, for the created models, we try a wide range of parameters and observe a wide range of results to address most potential scenarios. However, as the field of genomics becomes more mature, our models can be replaced by the values obtained from the medical research.

The LD relationship between two SNPs i and j can be represented as $\Pr(\text{SNP}_i|\text{SNP}_j)$, where SNP_i (or SNP_j) takes values from the set $\{0,1\}$.[7] We note that LD relationships are defined among all 40 million discovered SNPs, regardless of their type (i.e., real or potential) at a particular patient.

As before, we let Ω_P^s and Ω_P^u denote the set of P's potential SNPs that are stored (for redundancy) and not stored at the SPU, respectively. Further, K_i is the set of SNPs with which a particular SNP i has LD, and $|K_i| = k$ (for each SNP, these k SNPs are chosen among approximately 40 million SNPs). We assume that $k \geq 0$ and it is a truncated Gaussian random variable with only discrete values and obtained from a distribution with mean $\mu(k)$ and standard deviation $\sigma(k)$.

Initially, we compute $\Pr(\text{SNP}_i^P)$ for all (real and potential) SNPs in $\{\Upsilon_P \cup \Omega_P^s\}$ by using the LD relationships between these SNPs and those in Ω_P^u. As all SNPs in $\{\Upsilon_P \cup \Omega_P^s\}$ are encrypted and stored at the SPU, only the LD relationships between these SNPs and the un-stored SNPs in Ω_P^u are helpful for the curious party. Therefore, for each real SNP $i \in \Upsilon_P$, we observe $\Pr(\text{SNP}_i^P = 1|\text{SNP}_m^P = 0)$ for all $m \in \{K_i \cap \Omega_P^u\}$, get the average of these values, and compute $\Pr(\text{SNP}_i^P = 1)$. Similarly, for each potential SNP $j \in \Omega_P^s$, we observe $\Pr(\text{SNP}_j^P = 0|\text{SNP}_m^P = 0)$ for all $m \in \{K_j \cap \Omega_P^u\}$, average these values, and compute $\Pr(\text{SNP}_j^P = 0)$. We let l be the indicator of the LD strength between two SNPs. Thus, we represent $\Pr(\text{SNP}_i^P = 1|\text{SNP}_m^P = 0) = l$ ($i \in \Upsilon_P$, $m \in \{K_i \cap \Omega_P^u\}$) and $\Pr(\text{SNP}_j^P = 0|\text{SNP}_m^P = 0) = l$ ($j \in \Omega_P^s$, $m \in \{K_j \cap \Omega_P^u\}$) as truncated Gaussian random variables with range $[0.5, 1]$, obtained from a distribution with mean $\mu(l)$ and standard deviation $\sigma(l)$. Finally, if $|K_i| = k = 0$ or $|K_i \cap \Omega_P^u| = 0$ for a SNP i in $\{\Upsilon_P \cup \Omega_P^s\}$, we update $\Pr(\text{SNP}_i^P = 1)$ considering the fact that the expected value of all stored SNPs is known by the curious party as below:

$$\frac{1}{|\Upsilon_P \cup \Omega_P^s|} \sum_{j \in \Upsilon_P \cup \Omega_P^s} (\text{SNP}_j^P) \times \Pr(\text{SNP}_j^P) = \frac{|\Upsilon_P|}{|\Upsilon_P \cup \Omega_P^s|}. \quad (7)$$

In the following, we illustrate our numerical results that represent the relationship between the storage cost, the inference power of the curious party at the SPU, and the LD values. We assume $|\Upsilon_P| = 4$ million and $|\Upsilon_P \cup \Omega_P| = 40$ million. We define the percentage of storage redundancy at the SPU as $\frac{|\Omega_P^s|}{|\Upsilon_P|} \times 100$ and compute the average value of $\Pr(\text{SNP}_i^P = 1)$ for a SNP in Υ_P for varying values of $\mu(k)$, $\sigma(k)$, $\mu(l)$, and $\sigma(l)$.[8] We repeat each simulation 100 times to obtain an average.

In Fig. 3, we illustrate the variance in the average value of $\Pr(\text{SNP}_i^P = 1)$ for different values of $\mu(k)$, when $\mu(l) = 0.8$, $\sigma(l) = 0.15$, and $\sigma(k) = 0.75$. We note that "no LD" curve in the figure represents the case in which the LD values between the SNPs are ignored. We observe that as the available side information (i.e., number of un-stored potential SNPs in Ω_P^u having LD with the stored ones) increases, the inference power of the curious party increases, especially for low values of storage redundancy. For example, to have the same inference power for the curious party, 200% storage redundancy is required when $\mu(k) = 0$, whereas it is 700% when $\mu(k) = 4$. Furthermore, even at the maximum (i.e, 900%) storage redundancy, the curious party still has a slight probability of inferring the variants of the patient, because it knows that 4 out of 40 million of the stored content are variants.

Next, in Fig. 4, we illustrate the variance in the same probability, this time for different values of $\mu(l)$, when $\mu(k) = 2$, $\sigma(k) = 0.75$, and $\sigma(l) = 0.25$.[9] As expected, the inference power of the curious party increases when the strength of LD between the SNPs increases (i.e., when $\mu(l)$ increases). We observe that the strength of LD, however, does not affect the inference power as strong as k. Then, in Figs. 5 and 6, we show the $\text{Average}\{\Pr(\text{SNP}_i^P = 1)\}$ for varying standard deviations of k and l, and with 500% storage redundancy as follows: (i) in Fig. 5, we vary $\sigma(k)$ and $\mu(k)$, when $\mu(l) = 0.8$ and $\sigma(l) = 0.15$, and (ii) in Fig. 6, we vary $\sigma(l)$ and $\mu(l)$, when $\mu(k) = 2$ and $\sigma(k) = 0.75$. We observe that the inference power of the curious party varies (either increases or decreases) with an increasing value of $\sigma(k)$ ($\sigma(l)$) depending on $\mu(k)$ ($\mu(l)$), and, as expected, all curves converge to a single value for higher values of $\sigma(k)$ ($\sigma(l)$).

Next, considering the individual characteristics of the real SNPs (i.e., their severity levels), we study the level of genomic privacy of a patient against a curious party at the SPU. The severity of a SNP i can be defined as the privacy-sensitivity of the SNP when $\text{SNP}_i^P = 1$ (i.e., when it exists as a variant at the patient P). For example, a real SNP revealing the predisposition of a patient for Alzheimer's disease can be considered more severe than another real SNP revealing his predisposition to a more benign disease. Severity values of the SNPs are determined as a result of medical studies (depending on their contributions to various diseases) and tables of disease severities provided by insurance companies (e.g., percentage of invalidity). We denote the severity of a real SNP i as V_i, and $0 \leq V_i \leq 1$ (1 denotes the highest severity). Thus, we define the genomic privacy of the patient P as below:

$$\Phi_P = -\sum_{i \in \Upsilon_P} \log_2 \Big(\Pr(\text{SNP}_i^P = 1)\Big) \times V_i. \quad (8)$$

We do not use the traditional entropy metric [21], [22] to quantify privacy, as only one state of SNP_i^P poses privacy risks (i.e., $\text{SNP}_i^P = 1$), as discussed before.

First, we study the relationship between the storage redundancy and the severity of the real SNPs by focusing on three types of patients: (i) patient A, carrying mostly low severity real SNPs (in Υ_A), (ii) patient B, carrying mostly high severity real

[7] In compliance with genetic observations, we assume that the LD between two SNPs are not symmetric, i.e., $\Pr(\text{SNP}_i|\text{SNP}_j) \neq \Pr(\text{SNP}_j|\text{SNP}_i)$.

[8] Higher values of $\Pr(\text{SNP}_i^P = 1)$ indicate a higher inference power for the curious party at the SPU.

[9] For higher values of $\sigma(l)$, the gap between the different $\mu(l)$ curves becomes negligible, because the distribution becomes almost uniform, rather than truncated Gaussian.

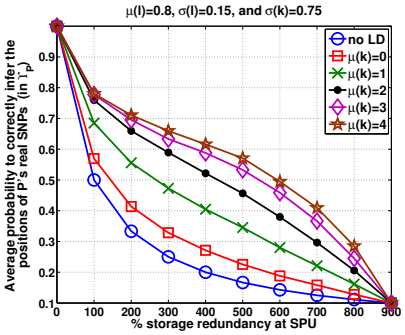

Fig. 3. Average probability to correctly infer the positions of patient's real SNPs (for the curious party at the SPU) with varying mean values of the number of LD pairs per SNP (i.e., $\mu(k)$) and storage redundancy.

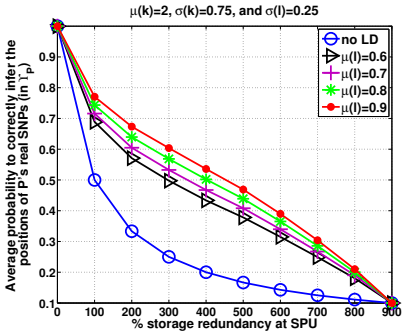

Fig. 4. Average probability to correctly infer the positions of patient's real SNPs (for the curious party at the SPU) with varying mean values of the LD strength between two SNPs (i.e., $\mu(l)$) and storage redundancy.

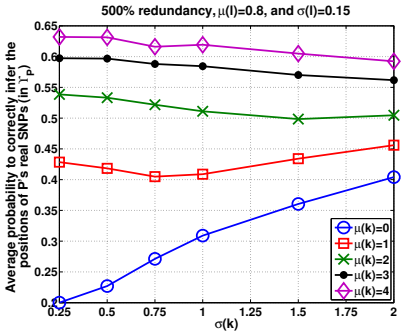

Fig. 5. Average probability to correctly infer the positions of patient's real SNPs (for the curious party at the SPU) with varying standard deviation and mean values of the number of LD pairs per SNP (i.e., $\sigma(k)$ and $\mu(k)$).

SNPs (in Υ_B), and (iii) patient C, carrying mixed severity real SNPs (in Υ_C). For each patient, the highest level of privacy is achieved when the storage redundancy is maximum (i.e., when all potential SNPs of the patient are stored at the SPU). Thus, we recognize this level as 100% genomic privacy for the patient. For the evaluation, we take the highest privacy level of patient C as the base and normalize everything with respect to this value. We use the following parameters for the simulation. The severities of patient A's and patient B's real SNPs are represented as truncated Gaussian random variables with $(\mu_A, \sigma_A) = (0.25, 0.15)$ and $(\mu_B, \sigma_B) = (0.75, 0.15)$, respectively. Furthermore, the severity of patient C's real SNPs are represented as a uniform distribution between 0 and 1. We also set $\mu(l) = 0.8$, $\sigma(l) = 0.25$, $\mu(k) = 2$, and $\sigma(k) = 0.75$. In Fig. 7, we illustrate the increase in privacy with increments in the storage redundancy for these three types of patients (A, B, and C). We observe that by increasing the storage redundancy, a patient with high severity real SNPs gains more privacy than a patient with lower severity real SNPs, hence the storage redundancy can be customized for each patient differently based on the types of his real SNPs. It can be argued that the amount of storage redundancy for a patient can leak information (to the curious party the SPU) about the severities of his real SNPs. However, the severity of the SNPs is not the only criteria to determine the storage redundancy for a desired level of genomic privacy as we discuss next.

Finally, we study the relationship between the severity of the real SNPs, the number of LD pairs per SNP (number of SNPs with which a particular SNP has LD, i.e., k), and the storage redundancy. We assign the V_i values of the real SNPs (in Υ_P) following a uniform distribution between 0 and 1. We set the LD parameters as $\mu(l) = 0.8$, $\sigma(l) = 0.25$, $\mu(k) = 2$, and $\sigma(k) = 1.5$. Then, we observe and compare the following potential scenarios in different types of patients: (i) The real low severity SNPs of the patient (i.e., his real SNPs with low V_i values) have a higher number of LD pairs (i.e., higher k values) with respect to his high severity real SNPs[10]; (ii) k values are assigned randomly to the SNPs; and (iii) the real high severity SNPs of the patient (i.e., his real SNPs with high V_i values) have a higher number of LD pairs (i.e., higher k values) with respect to his low severity real SNPs. Again, we set a patient's genomic privacy to 100% when the storage redundancy is maximum at the SPU. We illustrate our results in Fig. 8, and show different storage redundancy requirements for different types of patients (to provide the same level of privacy). For example, to achieve 40% genomic privacy, the SPU requires 400% storage redundancy for a patient whose less severe real SNPs have more LD pairs, whereas it requires 600% storage redundancy for another patient whose more severe real SNPs have more LD pairs (which means more storage cost per patient, as discussed in Section IV). This result also supports our belief to customize the storage redundancy for each patient.

We obtained similar patterns for further variations of the variables but we do not present these results due to the space limitation. In summary, depending on the actual $\mu(k)$, $\sigma(k)$, $\mu(l)$, $\sigma(l)$, and V_i values (which will be determined as a result of the medical research), the storage redundancy can be determined (and customized for each patient based on the types of his variations) to keep the genomic privacy of the patient at a desired level. Note that the curious party at the SPU cannot infer the real SNPs of the patient (or the severities of the patient's real SNPs) from the amount of customized storage redundancy, because the storage redundancy (for a desired level of genomic privacy) depends on various factors. For example, a patient with low storage redundancy (for a desired level of genomic privacy) could mean that (i) he carries mostly low severity real SNP (as in Fig. 7), (ii) he carries mixed severity real SNPs, but his less severe real SNPs have more LD pairs (as in Fig. 8), (iii) his real SNPs (regardless of their severities) have low number of LD pairs (as in Fig. 3), or (iv) his real SNPs (regardless of their severities) have low LD strengths (as in Fig. 4).

IV. Complexity Evaluation

We implemented the proposed scheme and assessed its storage requirement and computational complexity on Intel Core i7-2620M CPU with 2.70 GHz processor under Windows 7 64-bit Operating System. We set the size of the security parameter (n in Paillier cryptosystem in Section II-A) to 2048 bits. We computed the disease susceptibility using a real SNP profile from [23]. Our implementation relies on a MySQL 5.5 database and to provide a platform-independent implementation, we used the Java programming language.

Let ϑ represent the percentage of storage redundancy at the SPU. Then, $\left(1 + \frac{\vartheta}{100}\right)$ GB storage is required at the SPU per patient. We observed that encryption of the patient's variants (via the Paillier encryption) takes 90 ms per variant at the Certified Institution (CI). We emphasize that the encryption of the variants at the CI is a one-time operation and is significantly faster than the sequencing and analysis of the sequence (which takes days). Further, this encryption can be conducted much more efficiently by computing some parameters, such as (g^r, h^r) pairs, offline for various r values, for each patient. Indeed, by computing (g^r, h^r) pairs offline, we observe that the encryption takes only 0.04 ms per variant at the CI. We also observed that disease-susceptibility test at the SPU via homomorphic operations (using ten variants) takes around 20 sec. and proxy re-encryption takes

[10]We note that, in all cases, k values are obtained from the same truncated Gaussian distribution with $\mu(k) = 2$, and $\sigma(k) = 1.5$.

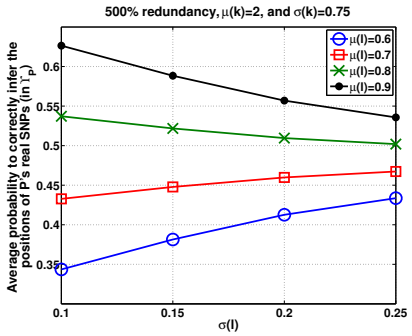

Fig. 6. Average probability to correctly infer the positions of patient's real SNPs (for the curious party at the SPU) with varying standard deviation and mean values of the LD strength between two SNPs (i.e., $\sigma(l)$ and $\mu(l)$).

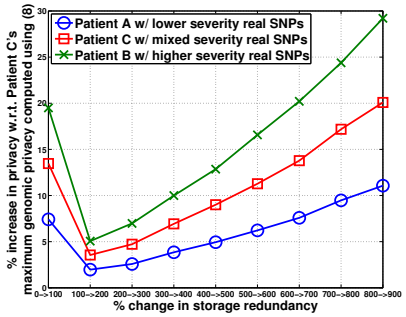

Fig. 7. Increase in genomic privacy of different types of patients with 100% increments in the storage redundancy. For example, increasing the storage redundancy from 400% to 500% would increase the privacy of Patient A (who carries mostly low severity real SNPs) by 5%, whereas the same scenario increases the privacy of Patient B (who carries mostly high severity SNPs) by 13%.

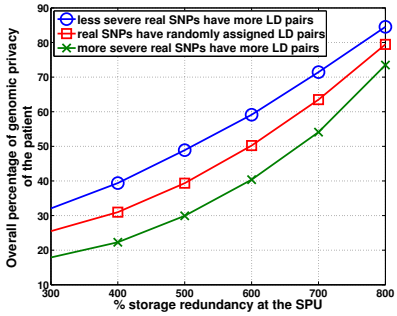

Fig. 8. Level of genomic privacy, as defined by (8), for different types of patients with varying storage redundancy.

30 ms. Finally, decryption of the end-result (or relevant SNPs) takes 200 ms at the MC. In summary, all these numbers show the practicality of our privacy-preserving algorithm.

V. Conclusion

In this paper, we have introduced a privacy-preserving scheme for the utilization of the genomic data in medical tests. We have shown that encrypted genomic data of the patients can be stored at the Storage and Processing Unit (SPU) and processed (for medical tests) using homomorphic encryption. Moreover, we analyzed the relationship between the storage cost, privacy of the patient, strength of relationship between the genetic markers, and the characteristics of the markers. This analysis could play a key role for customizing the storage redundancy of the genomic data for each patient, while keeping the privacy of the patient at a desired level. We also implemented the proposed scheme and showed its efficiency and practicality through a complexity evaluation. We are confident that our proposed privacy-preserving scheme will encourage the use of the genomic data, by the individual and by the medical unit, and accelerate the move of genomics into clinical practice.

VI. Acknowledgements

We would like to thank Dr. Amalio Telenti, Dr. Jacques Fellay, Dr. Philip E. Tarr, Dr. Jacques Rougemont, and Dr. Paul J. McLaren for their useful comments and suggestions. We also would like to thank Dr. Vincent Mooser, Dr. Didier Trono and Dr. Martin Vetterli for their encouragements in this research endeavor.

References

[1] M. Langheinrich, "Principles of privacy-aware ubiquitous systems," *Proceedings of Ubiquitous Computing (UbiComp)*, 2001.

[2] E. Ayday, E. D. Cristofaro, G. Tsudik, and J. P. Hubaux, "The chills and thrills of whole genome sequencing," *arXiv:1306.1264*, 2013. [Online]. Available: http://arxiv.org/abs/1306.1264

[3] 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.

[4] 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.

[5] 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.

[6] F. Bruekers, S. Katzenbeisser, K. Kursawe, and P. Tuyls, "Privacy-preserving matching of DNA profiles," Tech. Rep., 2008.

[7] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin, "A cryptographic approach to securely share and query genomic sequences," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 5, pp. 606–617, 2008.

[8] 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.

[9] M. Canim, M. Kantarcioglu, and B. Malin, "Secure management of biomedical data with cryptographic hardware," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, 2012.

[10] E. Ayday, J. L. Raisaro, and J. P. Hubaux, "Privacy-enhancing technologies for medical tests using genomic data," *(short paper) in 20th Annual Network and Distributed System Security Symposium (NDSS)*, 2013.

[11] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J. P. Hubaux, "Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data," *Proceedings of USENIX Security Workshop on Health Information Technologies (HealthTech)*, 2013.

[12] E. Ayday, J. L. Raisaro, U. Hengartner, A. Molyneaux, and J. P. Hubaux, "Privacy-preserving processing of raw genomic data," *Proceedings of 8th Data Privacy Management (DPM 2013) International Workshop (in conjunction with ESORICS)*, 2013.

[13] M. Humbert, E. Ayday, A. Telenti, and J. P. Hubaux, "Addressing the concerns of the Lacks Family: Quantification of kin genomic privacy," *Proceedings of 20th ACM Conference on Computer and Communications Security (CCS 2013)*, 2013.

[14] 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.

[15] 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.

[16] http://www.ncbi.nlm.nih.gov/projects/SNP/, Visited on 14/Mar/2013.

[17] D. Greenbaum, A. Sboner, X. Mu, and M. Gerstein, "Genomics and privacy: Implications of the new reality of closed data for the field," *PLoS Computational Biology*, vol. 7, no. 12, 2011.

[18] 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.

[19] 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, Feb. 2006.

[20] D. S. Falconer and T. F. Mackay, *Introduction to Quantitative Genetics (4th Edition)*. Harlow, Essex, UK: Addison Wesley Longman, 1996.

[21] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," *Proceedings of Privacy Enhancing Technologies Symposium (PETS)*, 2002.

[22] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," *Prioceedings of Privacy Enhancing Technologies Symposium (PETS)*, 2002.

[23] The 1000 Genomes Project Consortium, "A map of human genome variation from population-scale sequencing," *Nature*, vol. 467, pp. 1061–1073, 2010.