Chapter 18
Threats and Solutions for Genomic Data Privacy

Erman Ayday and Jean-Pierre Hubaux

Abstract With the help of rapidly developing technology, DNA sequencing is becoming less expensive. As a consequence, the research in genomics has gained speed in paving the way to personalized (genomic) medicine, and geneticists need large collections of human genomes to further increase this speed. Furthermore, individuals are using their genomes to learn about their (genetic) predispositions to diseases, their ancestries, and even their (genetic) compatibilities with potential partners. This trend has also caused the launch of health-related websites and online social networks (OSNs), in which individuals share their genomic data (e.g., OpenSNP or 23andMe). On the other hand, genomic data carries much sensitive information about its owner. By analyzing the DNA of an individual, it is now possible to learn about his disease predispositions (e.g., for Alzheimer's or Parkinson's), ancestries, and physical attributes. The threat to genomic privacy is magnified by the fact that a person's genome is correlated to his family members' genomes, thus leading to interdependent privacy risks. Thus, in this chapter, focusing on our existing and ongoing work on genomic privacy carried out at EPFL/LCA1, we will first highlight the threats for genomic privacy. Then, we will present the high level descriptions of our solutions to protect the privacy of genomic data and we will discuss future research directions. For a description of the research contributions of other research groups, the reader is referred to Chaps. 16 and 17 of the present volume.

18.1 Threats for Genomic Privacy

Removal of quasi-identifying attributes (e.g., date of birth or zip code) legally protects the privacy of health data. However, it has been shown that anonymization

E. Ayday (✉)
Department of Computer Engineering, Bilkent University, Ankara, Turkey
e-mail: erman@cs.bilkent.edu.tr

J.-P. Hubaux
Institute of Communication Systems, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
e-mail: jean-pierre.hubaux@epfl.ch

A. Gkoulalas-Divanis, G. Loukides (eds.), *Medical Data Privacy Handbook*,
DOI 10.1007/978-3-319-23633-9_18

is an ineffective technique for genomic data [16, 18, 20]. For example, an adversary can infer the phenotype of the donor of an anonymized genome and use this information to identify the anonymous donor.

For instance, genomic variants on the Y chromosome are correlated with the last name (for males). This last name can be inferred using public genealogy databases. With further effort (e.g., using voter registration forms) the complete identity of the individual can also be revealed [18]. Also, unique features in patient-location visit patterns in a distributed healthcare environment can be used to link the genomic data to the identity of the individuals in publicly available records [33]. Furthermore, it has been shown that Personal Genome Project (PGP) participants can be identified based on their demographics without using any genomic information [42].

The identity of a participant of a genomic study can also be revealed by using a second sample, that is, part of the DNA information from the individual and the results of the corresponding clinical study [9, 16, 21, 25, 43]. For this reason even a small set of variants (e.g., single nucleotide variants - SNPs) of the individual might be sufficient as the second sample. For example, it is shown that as few as 100 SNPs are enough to uniquely distinguish one individual from others [31]. Homer et al. [21] prove that the presence of an individual in a case group can be determined by using aggregate allele frequencies and his DNA profile. Homer's attack demonstrates that it is possible to identify a participant of a Gename-wide association study (GWAS) by analyzing the allele frequencies of a large number of SNPs. Wang et al. [43] showed a higher risk that individuals can actually be identified from a relatively small set of statistics such as those routinely published in GWAS papers. In particular, they show that the presence of an individual in the case group can be determined based upon the pairwise correlation (i.e., linkage disequilibrium) among as few as a couple of hundred SNPs. While the methodology introduced in [21] requires on the order of 10,000 SNPs (of the target individual), this new attack requires only on the order of hundreds. Another similar attack involves the association of DNA sequences to personal names, through diagnosis codes [32].

In another recent study [16], Gitschier shows that a combination of information from genealogical registries and a haplotype analysis of the Y chromosome collected for The HapMap Project, allows for the prediction of the last names of a number of individuals held in the HapMap database. Thus, releasing (aggregate) genomic data is currently banned by many institutions due to this privacy risk. Zhou et al. [45], study the privacy risks of releasing aggregate genomic data. They propose a risk-scale system to classify aggregate data and a guide for their release.

Some believe that they have nothing to hide about their genetic structure, hence they might decide to give full consent for the publication of their genomes on the Internet to help genomic research. However, our DNA sequences are highly correlated to our relatives' sequences. The DNA sequences between two random human beings are more than 99.5 % similar, and this value is even higher for closely related people. Consequently, somebody revealing his genome does not only damage his own genomic privacy, but also puts his relatives' privacy at risk [41].

Moreover, currently, a person does not need consent from his relatives to share his genome online. This is precisely where the interesting part of the story begins: *kin genomic privacy*.

18.1.1 Kin Genomic Privacy

A recent New York Times' article[1] reports the controversy about sequencing and publishing, without the permission of her family, the genome of Henrietta Lacks (who died in 1951). On the one hand, the family members think that her genome is private family information and it should not be published without the consent of the family. On the other hand, some scientists argued that the genomes of current family members have changed so much over time (due to gene mixing during reproduction), that nothing accurate could be told about the genomes of current family members by using Henrietta Lacks' genome. As we have shown in [23] (that we briefly describe hereafter), they were wrong. Minutes after Henrietta Lacks' genome was uploaded to a public website called SNPedia, researchers produced a report full of personal information about Henrietta Lacks. Later, the genome was taken offline, but it had already been downloaded by several people, hence both her and (partially) the Lacks family's genomic privacy was already lost.

Unfortunately, the Lacks, even though possibly the most publicized family facing this problem, are not the only family facing this threat. Genomes of thousands of individuals are available online. Once the identity of a genome donor is known, an attacker can learn about his relatives (or his family tree) by using an auxiliary side channel, such as an online social network (OSN) , and infer significant information about the DNA sequences of the donor's relatives. We will show the feasibility of such an attack and evaluate the privacy risks by using publicly available data on the Web.

Although the researchers took Henrietta Lacks' genome offline from SNPedia, other databases continue to publish portions of her genomic data. Publishing only portions of a genome does not, however, completely hide the unpublished portions; even if a person reveals only a part of his genome, other parts can be inferred using the statistical relationships between the nucleotides in his DNA. For example, James Watson, co-discoverer of DNA, made his whole DNA sequence publicly available, with the exception of one gene known as Apolipoprotein E (ApoE), one of the strongest predictors for the development of Alzheimer's disease. However, later it was shown that the correlation (called *linkage disequilibrium* by geneticists) between one or multiple polymorphisms and ApoE can be used to predict the ApoE status [35]. Thus, an attacker can also use these statistical relationships (which are publicly available) to infer the DNA sequences of a donor's family members, even

[1] http://www.nytimes.com/2013/03/24/opinion/sunday/the-immortal-life-of-henrietta-lacks-the-sequel.html?pagewanted=all

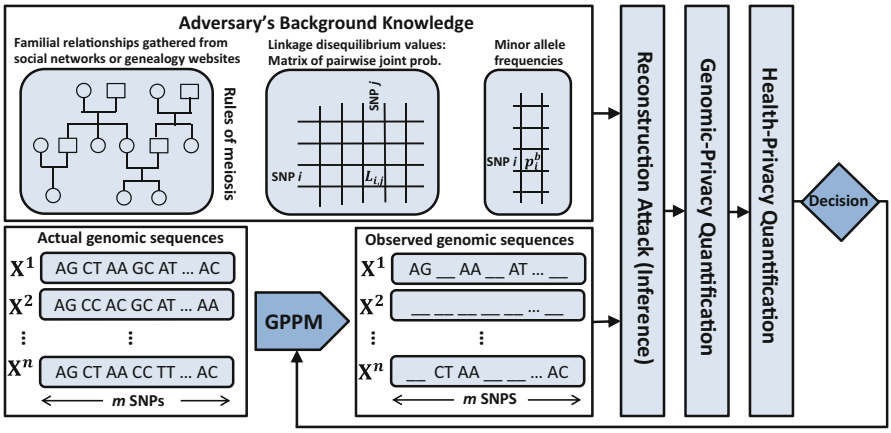


Fig. 18.1 Overview of the proposed framework to quantify kin genomic privacy [23]. Each vector $\mathbf{X}^i$ ($i \in \{1, \ldots, n\}$) includes the set of SNPs for an individual in the targeted family. Furthermore, each letter pair in $\mathbf{X}^i$ represents a SNP x_j^i; and for simplicity, each SNP x_j^i can be represented using $\{BB, Bb, bb\}$ (or $\{0, 1, 2\}$). Once the health privacy is quantified, the family should ideally decide whether to reveal less or more of their genomic information through the genomic-privacy preserving mechanism (GPPM)

if the donor shares only part of his genome. It is important to note that these privacy threats not only jeopardize kin genomic privacy, but, if not properly addressed, these issues could also hamper genomic research due to untimely fear of potential misuse of genomic information.

In [23], we evaluated the genomic privacy of an individual threatened by his relatives revealing their genomes. Focusing on the most common genetic variant in human population, single nucleotide polymorphism (SNP),[2] and considering the statistical relationships between the SNPs on the DNA sequence, we quantify the loss in genomic privacy of individuals when one or more of their family members' genomes are (either partially or fully) revealed. To achieve this goal, first, we design a reconstruction attack based on a well-known statistical inference technique. The computational complexity of the traditional ways of realizing such inference grows exponentially with the number of SNPs (which is on the order of tens of millions) and relatives. Therefore, in order to infer the values of the unknown SNPs in linear complexity, we represent the SNPs, family relationships and the statistical relationships between SNPs on a factor graph and use the belief propagation algorithm [30, 36] for inference. Then, using various metrics, we quantify the genomic privacy of individuals and show the decrease in their privacy level caused

[2] A SNP occurs when a nucleotide (at a specific position on the DNA) varies between individuals of a given population. SNPs carry privacy-sensitive information about individuals' health. Recent discoveries show that the susceptibility of an individual to several diseases can be computed from his or her SNPs.

Table 18.1 Frequently used notations

$\mathbf{F}$	Set of family members in the targeted family
$\mathbf{S}$	Set of SNP IDs
x_j^i	Value of SNP j for individual i, $x_j^i \in \{0, 1, 2\}$
$\mathbf{X}^i$	Set of SNPs for individual i
$\mathbb{X}$	$n \times m$ matrix that stores the values of the SNPs of all family members
$\mathbb{X}_{\mathrm{U}}$	Set of SNPs from $\mathbb{X}$ whose values are unknown
$\mathbb{X}_{\mathrm{K}}$	Set of SNPs from $\mathbb{X}$ whose values are known by the adversary
$\mathcal{F}_R()$	Function representing the Mendelian inheritance probabilities
$\mathbb{L}$	$m \times m$ matrix representing the pairwise linkage disequilibrium between the SNPs in $\mathbf{S}$
$\mathbb{L}_{i,j}$	Entry of $\mathbb{L}$ at row i and column j
$\mathbf{P}$	Set of minor allele probabilities (or MAF) of the SNPs in $\mathbf{S}$

by the published genomes of their family members. We also quantify the health privacy of the individuals by considering their (genetic) predisposition to certain serious diseases. We evaluate the proposed inference attack and show its efficiency and accuracy by using real genomic data of a pedigree.

In the following, we formalize our approach and present the different components that will allow us to quantify kin genomic privacy. Figure 18.1 gives an overview of the framework. In order to facilitate future references, frequently used notations are listed in Table 18.1.

In a nutshell, the goal of the adversary is to infer some *targeted SNPs* of a member (or multiple members) of a *targeted family*. We define $\mathbf{F}$ to be the set of family members in the targeted family (whose family tree, showing the familial connections between the members, is denoted as $\mathcal{G}_{\mathbf{F}}$) and $\mathbf{S}$ to be the set of SNP IDs (i.e., positions on the DNA sequence), where $|\mathbf{F}| = n$ and $|\mathbf{S}| = m$. Note that the SNP IDs are the same for all the members of the family. We also let x_j^i be the value of SNP j ($j \in \mathbf{S}$) for individual i ($i \in \mathbf{F}$), where $x_j^i \in \{0, 1, 2\}$ (a SNP can only be in one of these three states). Furthermore, $\mathbf{X}^i = \{x_j^i : j \in \mathbf{S}, i \in \mathbf{F}\}$ represents the set of SNPs for individual i. We let $\mathbb{X}$ be the $n \times m$ matrix that stores the values of the SNPs of all family members. Some entries of $\mathbb{X}$ might be known by the adversary (the observed genomic data of one or more family members) and others might be unknown. We denote the set of SNPs from $\mathbb{X}$ whose values are unknown as $\mathbb{X}_{\mathrm{U}}$, and the set of SNPs from $\mathbb{X}$ whose values are known (by the adversary) as $\mathbb{X}_{\mathrm{K}}$.

$\mathcal{F}_R(x_j^M, x_j^F, x_j^C)$ is the function representing the Mendelian inheritance probabilities, where (M, F, C) represent mother, father, and child, respectively. The $m \times m$ matrix $\mathbb{L}$ represents the pairwise linkage disequilibrium (LD)[3] between the SNPs in $\mathbf{S}$, $\mathbb{L}_{i,j}$ refers to the matrix entry at row i and column j. $\mathbb{L}_{i,j} > 0$ if i and j are in LD, and $\mathbb{L}_{i,j} = 0$ if these two SNPs are independent (i.e., there is no LD between them).

[3]LD can be thought as a correlation between two variables.

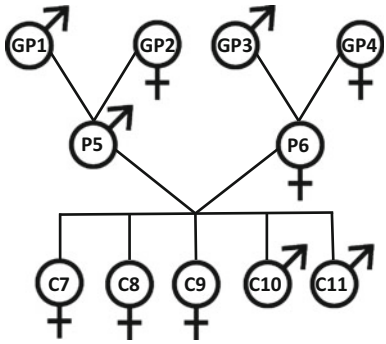


Fig. 18.2 Family tree of *CEPH/Utah Pedigree 1463* consisting of the 11 family members that were considered. The notations *GP*, *P*, and *C* stand for "grandparent", "parent", and "child", respectively. Also, the symbols ♂ and ♀ represent the male and female family members, respectively

$\mathbf{P} = \{p_i^b : i \in \mathbf{S}\}$ represents the set of minor allele probabilities (or MAF) of the SNPs in $\mathbf{S}$. Finally, note that a joint probability $p(x_i, x_j)$ can be derived from $\mathbb{L}_{i,j}$, p_i^b, and p_j^b.

The adversary carries out a reconstruction attack to infer $\mathbb{X}_\mathrm{U}$ by relying on his background knowledge, $\mathcal{F}_R(x_j^M, x_j^F, x_j^C)$, $\mathbb{L}$, $\mathbf{P}$, and on his observation $\mathbb{X}_\mathrm{K}$. We formulate the reconstruction attack (on determining the values of the targeted SNPs) as finding the marginal probability distributions of unknown variables $\mathbb{X}_\mathrm{U}$, given the known values in $\mathbb{X}_\mathrm{K}$, familial relationships, and the publicly available statistical information. To run this attack in an efficient way, we formulate the problem on a graphical model (factor graph) and use the belief propagation algorithm for inference. Once the targeted SNPs are inferred by the adversary, we evaluate genomic and health privacy of the family members based on the adversary's success and his certainty about the targeted SNPs and the diseases they reveal. Finally, we discuss some ideas to preserve the individuals' genomic and health privacy.

For the evaluation, we used the *CEPH/Utah Pedigree 1463* that contains the partial DNA sequences of 17 family members (4 grandparents, 2 parents, and 11 children) [10]. As shown in Fig. 18.2, we only used the first 5 (out of 11) children (without any particular selection criteria) for our evaluation because (i) 11 is much above the average number of children per family, (ii) we observe that the strength of adversary's inference does not increase further (due to the children's revealed genomes) when more that 5 children's genomes are revealed, and (iii) the belief propagation algorithm might have convergence issues due to the number of loops in the factor graph, and this number increases with the number of children.

We construct $\mathbf{S}$ from 100 SNPs on chromosome 1. Among these 100 SNPs, each SNP is in LD with 5 other SNPs on average. Furthermore, the strength of the LD varies between 0.5 and 1. We note that we only use 100 SNPs for this study as the LD values are not yet completely defined over all SNPs, and the definition of such values is still an ongoing research. We define a target individual from the CEPH family, construct the set $\mathbb{X}_\mathrm{U}$ from his/her SNPs, and sequentially reveal other family members' SNPs (excluding the target individual) to observe the decrease in the genomic privacy of the target individual. We start revealing from the most distant family members to the target individual (in terms of number of hops in Fig. 18.2)

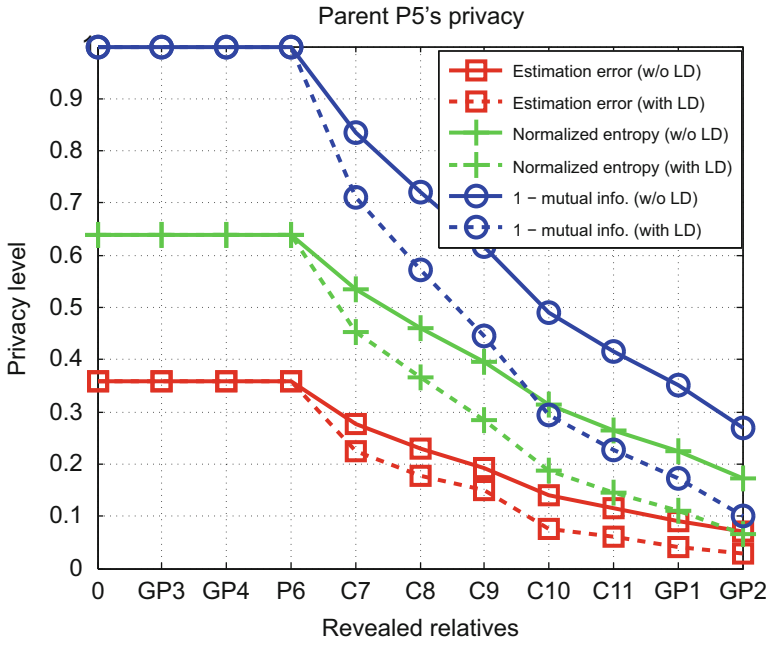


Fig. 18.3 Evolution of the genomic privacy of the parent (P5), with and without considering LD. For each family member, we reveal 50 randomly picked SNPs (among 100 SNPs in **S**), starting from the most distant family members, and the x-axis represents the exact sequence of this disclosure. Note that $x = 0$ represents the prior distribution, when no genomic data is revealed

and we keep revealing relatives until we reach his/her closest family members.[4] We observe that individuals sometimes reveal different parts of their genomes (e.g., different sets of SNPs) on the Internet. Thus, we assume that for each family member (except for the target individual), the adversary observes 50 random SNPs from **S** only (instead of all the SNPs in **S**), and these sets of observed SNPs are different for each family member. In Fig. 18.3, we show the evolution of genomic privacy of one target individual (P5). We quantify the genomic privacy based on (i) attackers incorrectness (bottom plot), (ii) attacker's uncertainty (middle plot), and (iii) an entropy-based metrics that quantities the mutual dependence between the hidden genomic data that the adversary is trying to reconstruct (top plot). We observe that LD decreases genomic privacy, especially when few individuals' genomes are revealed. As more family member's genomes are observed, LD has less impact on the genomic privacy.

As we already mentioned, the Lacks family is just one (albeit famous) example. In the future (and already today), people of the same family might have very differ-

[4]The exact sequence of the family members (whose SNPs are revealed) is indicated for each evaluation.

ent opinions on whether to reveal genomic data, and this can lead to disagreement: relatives might have divergent perceptions of possible consequences. It is high time for the security research community to prepare itself for this formidable challenge. The genetics community is highly concerned about the fact that the proliferation of negative stories could potentially lead to a negative perception by the population and to tighter laws, thus hampering scientific progress in this field.

In order to prevent some of the aforementioned threats on the privacy of genomic data, we proposed several solutions to protect the privacy of such data in various domains. In the next section, we describe some of these solutions.

18.2 Solutions for Genomic Privacy

In this section, we summarize some of our efforts to protect the privacy of genomic data by focusing on privacy-preserving management of raw genomic data, privacy compliant use of genomic data in personalized medicine and research settings, resistance to brute-force attacks for storage of genomic data, and protecting kin genomic privacy.

18.2.1 Privacy-Preserving Management of Raw Genomic Data

Sequence alignment/map (SAM and its binary version BAM) files are the *de facto* standards used to store the aligned,[5] raw genomic data generated by next-generation DNA sequencers and bioinformatic algorithms. There are hundreds of millions of short reads (each including between 100 and 400 nucleotides) in the SAM file of an individual. Typically, each nucleotide is present in several short reads in order to have sufficiently high coverage of each individual DNA.

In general, geneticists prefer storing aligned, raw genomic data of the patients (i.e., their SAM files), in addition to their variant calls (which include each nucleotide on the DNA sequence once, hence is much more compact) due to the following reasons: (i) Bioinformatic algorithms and sequencing platforms for variant calling are currently not yet mature, and hence geneticists prefer to observe each nucleotide in several short reads; (ii) If a patient carries a disease, which causes specific variations in the diseased cells (e.g., cancer), his or her DNA sequence in his/her healthy cells will be different from those diseased. Such variations can be misclassified as sequencing errors by only looking at the patient's variant calls (rather than his/her short reads). Furthermore, (iii) due to the rapid evolution of genomic research, geneticists do not know enough to decide which information

[5] Alignment is with respect to the reference genome, which is assembled by the scientists.

should really be kept and what is superfluous, hence they prefer to store all outcomes of the sequencing process as SAM files.

In Ayday et al. [4], we proposed a privacy-preserving system for the storage, retrieval, and processing of the SAM files. In a nutshell, the proposed scheme stores the encrypted SAM files of the patients at a *biobank* and it provides the requested range of nucleotides (on the DNA sequence) to a medical unit (for a genetic test) while protecting the patients' genomic privacy. It is important to note that the proposed scheme enables the privacy-preserving processing of the SAM files both for individual treatment (when the medical unit is embodied in a hospital) and for genetic research (when the medical unit is embodied in a pharmaceutical company).

We assume that the sequencing and encryption of the genomes are done at a *certified institution* (CI), which is a trusted entity. We note that having such a trusted entity cannot be avoided as the sequencing has to be done at some institution to obtain the SAM files of the patients. Each part (position, cigar string, and content)[6] of each short read (in the SAM file) is encrypted (via a different encryption scheme) after the sequencing, and encrypted SAM files of the patients are stored at a biobank. We assume that SAM files are stored at the biobank by using pseudonyms; this way, the biobank cannot associate the conducted genetic tests and the *medical unit* (MU), which conduct these tests, with the real identities of the patients. We note that a private company (e.g., cloud storage service) or the government could play the role of the biobank. There are potentially multiple MUs in the system, and each MU is an approved institution (by the medical authorities). Furthermore, we assume that an MU is a broad unit consisting of many sub-units (e.g., physicians or specialized clinics) that can potentially request nucleotides from any parts of a patient's genome.

The cryptographic keys of the patients are stored on a key manager by using the patient's pseudonym (which does not require the participation of the patient in the protocol). From here on, we assume the existence of a *masking and key manager* (MK) in the system to store the cryptographic keys of the patients. The MK can also be embodied in the government or a private company. The connection between these parties in the proposed protocol (along with the assumed threat model) is illustrated in detail in Fig. 18.4.

When the MU requests a specific range of nucleotides (on the DNA sequence of one or multiple patients), the biobank provides all the short reads that include at least one nucleotide from the requested range through the MK. During this process, the patient does not want to reveal his complete genome to the MU, to the biobank, or to the MK. Furthermore, it is not desirable for the biobank to learn the requested range of nucleotides (as the biobank can infer the nature of the genetic test from this requested range). Thus, we developed a privacy-preserving system for the retrieval of the short reads by the MU [4]. The proposed scheme provides the short reads that

[6]Position of a short read tells the position of the first nucleotide on the DNA sequence. Cigar string of a short read denotes the deletions and insertions on the short read. Content of a short read includes the nucleotides.

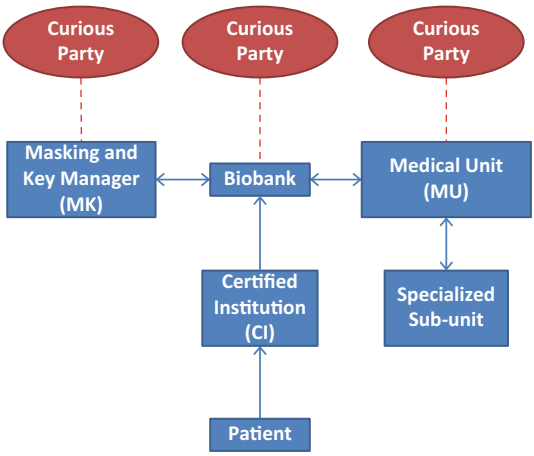


Fig. 18.4 Connections between the parties in the proposed protocol for privacy-preserving management of raw genomic data [4]

include the requested range of nucleotides to the MU without revealing the positions of these short reads to the biobank.

To achieve this goal, we first modify the structure of the genome by permuting the positions of the short reads, and then we use order preserving encryption (OPE) on the positions of the short reads (in the SAM file). OPE is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts [1, 37]. Thus, OPE enables the encryption of the positions of the short reads and preserves the numerical ordering of the plaintext positions.

We prevent the leakage of extra information in the short reads to the MU by masking the encrypted short reads at the biobank (before sending them to the MU). As each short read includes between 100 and 400 nucleotides, some provided short reads might include information out of the MU's requested range of genomic data, as in Fig. 18.5. Similarly, some provided short reads might contain privacy-sensitive SNPs of the patient (which would reveal the patient's susceptibilities to privacy-sensitive diseases such as Alzheimer's), hence the patient might not give consent to reveal such parts, as in Fig. 18.6. To achieve this goal, we encrypt the content of the short reads by using stream cipher and mask certain parts of the encrypted short reads at the biobank, without decrypting them using an efficient algorithm. It is important to note that after the short reads are decrypted at the MU, the MU is not able to determine the nucleotides at the masked positions. This proposed system is very efficient and it has been adopted in real-life by bioinformatics companies.

18.2.2 Private Use of Genomic Data in Personalized Medicine

In Ayday et al. [6], we proposed a scheme to protect the privacy of users' genomic data while enabling medical units to access the genomic data in order to conduct medical tests or develop personalized medicine methods. In a medical test, a medical

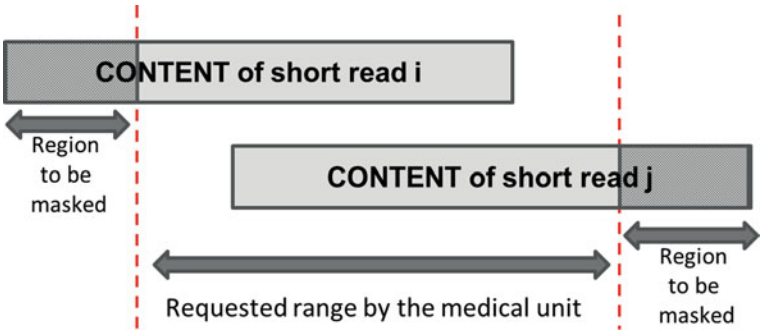
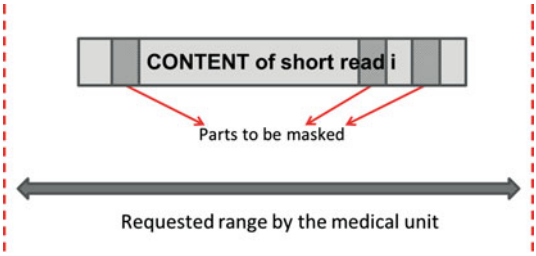


Fig. 18.5 Parts to be masked in the short reads for out-of-range content

Fig. 18.6 Parts to be masked in a short read based on patient's consent. The patient does not give consent to reveal the dark parts of the short read



unit checks for different health risks (e.g., disease susceptibilities) of a user by using specific parts of his genome. Similarly, to provide personalized medicine, a pharmaceutical company tests the compatibility of a user with a particular medicine. It is important to note that these genetic tests are currently done by different types of medical units, and the tools we propose in this work aim to protect the genomic privacy of the patients in such tests. In both medical tests and personalized medicine methods, in order to preserve his privacy, the user does not want to reveal his complete genome to the medical unit or to the pharmaceutical company. In addition, in some scenarios, it is the pharmaceutical companies who do not want to reveal the genetic properties of their drugs. To achieve these goals, we introduced the *privacy-preserving disease susceptibility test* (PDS) [6].

Most medical tests and personalized medicine methods (that use genomic data) involve a patient and a medical unit. In general, the medical unit can be a physician in a medical center (e.g., hospital), a pharmacist, a pharmaceutical company, or a medical council. In this study, we consider the existence of a curious entity in the medical unit as the potential attacker. That is, a medical unit might contain a disgruntled employee or it can be hacked by an intruder that is trying to obtain private genomic information about a patient (for which it is not authorized).

In addition, extreme precaution is needed for the storage of genomic data due to its sensitivity. Thus, we claim that a storage and processing unit (SPU) should be used to store the genomic data. We assume that the SPU is more "security-aware" than a medical unit, hence it can protect the stored genomic data against a hacker

better than a medical unit (yet, attacks against the SPU cannot be ruled out, as we discuss next). Recent medical data breaches from various medical units also support this assumption. Furthermore, instead of every medical unit individually storing the genomic data of the patients (in which case patients need to be sequenced by several medical units and their genomic data will be stored at several locations), a medical unit can retrieve the required genomic data belonging to a patient directly from the SPU. We note that a private company (e.g., cloud storage service), the government, or a non-profit organization could play the role of the SPU.

We assume that the SPU is an honest organization, but it might be curious. In other words, the SPU honestly follows the protocols and provides correct information to the other parties, however, a curious party at the SPU could access or infer the stored genomic data. Further, it is possible to identify a person only from his genomic data via phenotyping, which determines the observable physical or biochemical characteristics of an organism from its genetic makeup and environmental influences. Therefore, genomic data should be stored at the SPU in encrypted form. Similarly, apart from the possibility of containing a curious entity, the medical unit honestly follows the protocols. Thus, we assume that the medical unit does not make malicious requests from the SPU. We consider the following models for the attacker:

- A curious party at the SPU (or a hacker who breaks into the SPU), who tries to infer the genomic sequence of a patient from his stored genomic data. Such an attacker can infer the variants (i.e., nucleotides that vary between individuals) of the patient from his stored data.
- A semi-honest entity in the medical unit, who can be considered either as an attacker that hacks into the medical unit's system or a disgruntled employee who has access the medical unit's database. The goal of such an attacker is to obtain private genomic data of a patient for whom he or she is not authorized. The main resource of such an attacker is the results of the genetic tests that the patient undergoes.

For the simplicity of presentation, in the rest of this section, we will focus on a particular medical test (namely, computing genetic disease susceptibility). Similar techniques would apply for other medical tests and personalized medicine methods. In a typical genetic disease-susceptibility test, a *medical center* (MC) wants to check the susceptibility of a patient (P) for a particular disease *X* (i.e., the probability that patient P will develop disease *X*) by analyzing particular SNPs of the patient.[7]

For each patient, we propose to store only the *real SNPs* (around four million SNP positions on the DNA at which the patient has a mutation) at the SPU. At this point, it can be argued that these four million real SNPs (nucleotides) could be easily stored on the patient's computer or mobile device, instead of at the

[7]In this study, we only focused on the diseases which can be analyzed using the SNPs. We admit that there are also other diseases which depend on other forms of mutations or environmental factors.

SPU. However, we assert that this should be avoided due to the following issues. On one hand, types of variations in human population are not limited to SNPs, and there are other types of variations such as *copy-number variations* (CNVs), rearrangements, or translocations, consequently the required storage per patient is likely to be considerably more than only four million nucleotides. This high storage cost might still be affordable (via desktop computers or USB drives), however, genomic data of the patient should be available any time (e.g., for emergencies), thus it should be stored at a reliable source such as the SPU. On the other hand, leaving the patient's genomic data in his own hands and letting him store it on his computer or mobile device is risky, because his mobile device can be stolen or his computer can be hacked. It is true that the patient's cryptographic keys (or his authentication material) to access his genomic data at the SPU can also be stolen, however, in the case of a stolen cryptographic key, his genomic data (which is stored at the SPU) will still be safe. This can be considered like a stolen credit card issue. If the patient does not report that his keys are compromised, his genomic data can be accessed by the attacker.

It is important to note that protecting only the states (contents) of the patient's real SNPs is not sufficient in terms of his genomic privacy. As the real SNPs are stored at the SPU, a curious party at the SPU can infer the nucleotides corresponding to the real SNPs from their positions and from the correlation between the patient's potential SNPs and the real ones. That is, by knowing the positions of the patient's real SNPs, the curious party at the SPU will at least know that the patient has one or two minor alleles at these SNP positions (i.e., it will know that the corresponding SNP position includes either a real homozygous or heterozygous SNP), and it can make its inference stronger using the correlation between the SNPs.[8] Therefore, in [6] we proposed to encrypt both the positions of the real SNPs and their states. We assume that the patient stores his cryptographic keys (public-secret key pair for asymmetric encryption, and symmetric keys between the patient and other parties) on his smart card (e.g., digital ID card). Alternatively, these keys can be stored at a cloud-based password manager and retrieved by the patient when required.

In short, the whole genome sequencing is done by a *certified institution* (CI) with the consent of the patient. Moreover, the real SNPs of the patient and their positions on the DNA sequence (or their unique IDs) are encrypted by the same CI (using the patient's public and symmetric key, respectively) and uploaded to the SPU, so that the SPU cannot access the real SNPs of the patient (or their positions). 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 symmetric 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. We also assume the SPU not to have access to the real identities of the

[8]It is public knowledge that a real SNP includes at least one minor allele, and the curious party uses this background information in the attack.

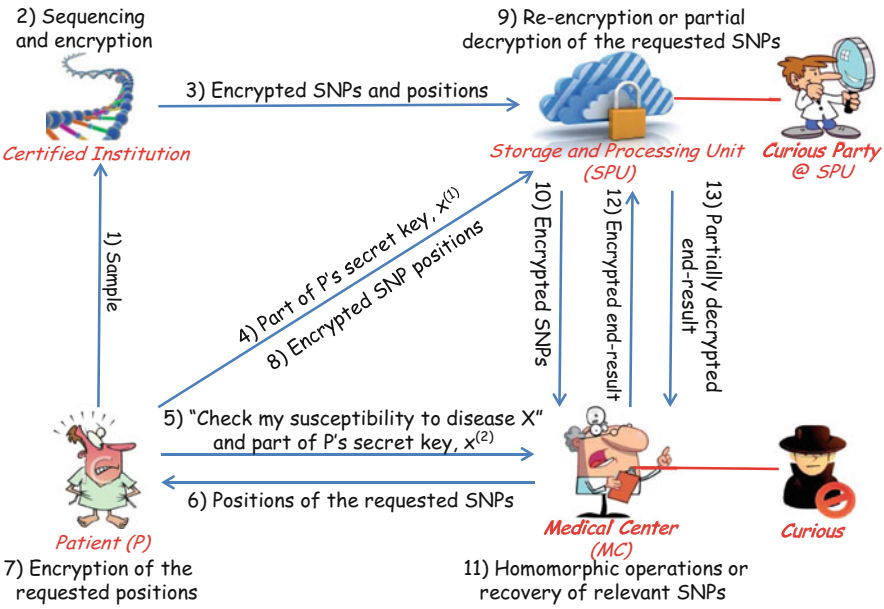


Fig. 18.7 Proposed privacy-preserving disease susceptibility test (PDS) [6]

patients and data to be stored at the SPU by using pseudonyms; this way, the SPU cannot associate the conducted genetic tests to the real identities of the patients.

Depending on the access rights of the MC, either (i) the MC computes Pr(X), the probability that the patient will develop disease X by checking a subset of the patient's encrypted SNPs via homomorphic encryption techniques [7], or (ii) the SPU provides 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 directly by the medical authorities. We note that homomorphic encryption lets the MC compute Pr(X) using encrypted SNPs of patient P. In other words, the MC does not access P's SNPs to compute his disease susceptibility. We use a modification of the Paillier cryptosystem [2, 7] to support the homomorphic operations at the MC. We show our proposed protocol in Fig. 18.7.

Following the steps in the figure, initially, the patient (P) provides his sample (e.g., his blood or saliva) to the certified institution (CI) for sequencing. After sequencing, the CI first determines the positions of P's real SNPs and the set positions at which P has real SNPs. Then, CI encrypts the SNPs (with Paillier cryptosystem using the public key of the patients) and their positions (using the symmetric key shared between the patient and the CI). Next, the CI sends the encrypted SNPs and positions to the SPU and the patient provides a part of his secret key ($x^{(1)}$) to the SPU. This finalizes the initialization phase of the protocol. Then, the MC wants to conduct a susceptibility test on P for a particular disease X, and P provides the other part of his secret key ($x^{(2)}$) to the MC. The MC tells the patient the

positions of the SNPs that are required for the susceptibility test or requested directly as the relevant SNPs (but not the individual contributions of these SNPs to the test). The patient encrypts each requested position with the symmetric key and sends the SPU the encrypted positions of the requested SNPs. Next, the SPU re-encrypts the requested SNPs and sends them to the MC. MC computes P's total susceptibility for disease X by using the homomorphic properties (i.e., homomorphic addition and multiplication with a constant) of the modified Paillier cryptosystem. The MC sends the encrypted end-result to the SPU, which partially decrypts it using $x^{(1)}$ by following a proxy re-encryption protocol and sends it back to the MC. Finally, the MC decrypts the message received from the SPU by using $x^{(2)}$ and recovers the end-result.

Even though this proposed approach provides a secure algorithm, there is still a privacy risk in case the MC tries to infer the patient's SNPs from the end-result of a test. In [6], we also showed that such an attack is indeed possible and one way to prevent such an attack is to obfuscate the end-result before providing it to the MC. Obviously, this causes a conflict between privacy and utility and this conflict is still a hot research topic for genomic privacy.

In a follow up work [5], we also proposed a system for protecting the privacy of individuals' sensitive genomic, clinical, and environmental information, while enabling medical units to process it in a privacy-preserving fashion in order to perform disease risk tests. We introduced a framework in which individuals' medical data (genomic, clinical, and environmental) is stored at a storage and processing unit (SPU) and a medical unit conducts the disease risk test on the encrypted medical data by using homomorphic encryption and privacy-preserving integer comparison. The proposed system preserves the privacy of the individuals' genomic, clinical, and environmental data from a curious party at the SPU and from a curious party (e.g., a hacker) at the medical unit when computing the disease risk. We also implemented the proposed system and showed its practicality via a complexity evaluation.

The general architecture of the proposed system is illustrated in Fig. 18.8. In summary, the patient provides his sample for sequencing to the CI. Meanwhile, he also provides his clinical and environmental data to the SPU and the MU.[9] The CI is responsible for sequencing and encryption of the patient's genomic data. Then, the CI sends the encrypted genomic data to the SPU. Finally, the privacy-preserving computation of the disease risk takes place between the MU and the SPU.

18.2.3 Private Use of Genomic Data in Research

The past years have witnessed substantial advances in understanding the genetic bases of many common phenotypes of biomedical importance. Such an evolution in

[9]Depending on the privacy-sensitivity of the clinical and environmental data, the patient can choose which clinical and environmental attributes to reveal to the MU, and which ones to encrypt and keep at the SPU.

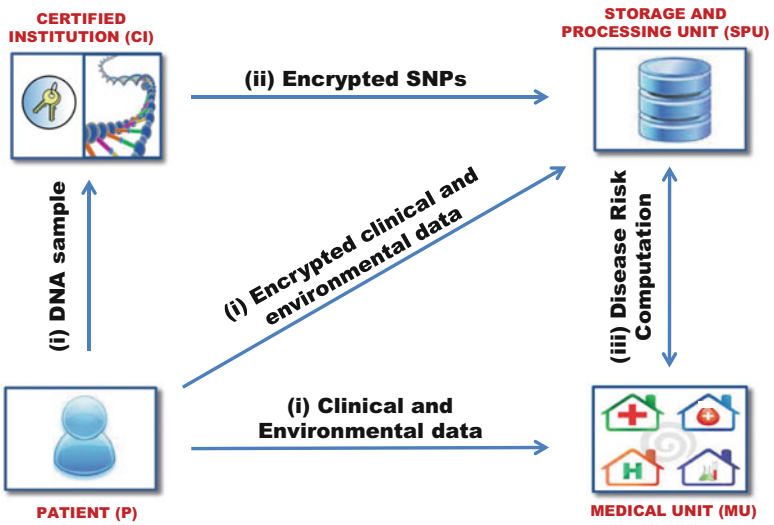


Fig. 18.8 Proposed system model for the privacy-preserving computation of the disease risk [5]

the medical field has pushed companies like Google to set up new infrastructures (e.g., Google Genomics [17]) to store, process and share genetic data at a large scale. Genome-wide association studies (GWAS) have become a popular method to investigate the relationship between the genomic variation and several diseases. They represent a starting point on the journey for translating this knowledge into clinics and they pave the way for personalized medicine, which is expected to have an unprecedented impact for clinical care by enabling treatment of diseases based on the genomic makeups of the individuals.

Even though much emphasis is given to GWAS, replication studies and fine-mapping of associated regions (which are both based on the *a priori* knowledge generated with GWAS) are crucial to identify true positive associations and variants that are causal for a phenotype. Replication studies are investigations performed in independent cohorts to validate variants identified by GWAS. Fine-mapping studies are useful in the post-GWAS phase when a few associations have been convincingly demonstrated and exhaustive work has to be performed to identify the actual causative variants. Additionally, it is becoming much more frequent to investigate multiple phenotypes across the same set of patient data, in so called phenome-wide association studies (PheWAS), which allow researchers to better understand the genetic architecture of complex traits and gain insights into disease mechanisms.

As genetic association studies depend on a large amount of genomic-phenomic data, strong privacy guarantees are required in order to protect the sensitive health information of individuals and, thus, facilitate the pace of genomic research by encouraging people to participate in such studies knowing that their privacy is protected. As discussed, genomic data includes privacy-sensitive information about

an individual, such as his ethnicity, kinship, and predisposition to specific diseases. Leakage of such information may cause genetic discrimination or blackmail. Similarly, phenotype data of individuals is also sensitive as it includes an individual's disease status and identifiers. Even though standard anonymization techniques can be used to publish phenotype data (with decreased accuracy), they have proved to be ineffective for genomic data [18, 21]. Hence, more sophisticated privacy-enhancing technologies have to be developed.

In Raisaro et al. [38], we proposed a privacy-preserving technique to conduct replication and fine-mapping genetic association studies.[10] We note that our solution is flexible enough to be generalized and to ensure privacy protection in different applications of the medical research field. Increasingly, large-scale data sets are being generated and applied in the medical setting, including proteomic, transcriptomic and metabolomic data. By recombining the building blocks of our privacy-preserving algorithm, the proposed architecture can easily support also secure analyses of multiple 'omics datasets for personalized medicine methods as proposed in Ayday et al. [6].

Existing techniques to conduct association studies in a privacy-preserving way include (i) adding noise to the result of the study to satisfy differential privacy [26, 44] (e.g., when the study is done at a trusted database and only the results of the study is shared with the researchers), and (ii) cryptographic techniques, such as using homomorphic encryption [28, 29] (e.g., when genomic data is shared with the researchers and the study is done by the researchers). Techniques in the former category reduce the utility of genomic data, and hence are criticized by genomic researchers, while cryptographic solutions enable computing exact answers with some computational and storage overhead [11, 14]. Our proposed technique falls into the latter category. However, as opposed to the existing crypto-based works, our proposed method in [38] (i) stores each participant's genotype and phenotype data encrypted by his own cryptographic key, (ii) addresses, for the first time in a privacy-preserving way, the problem of population stratification, and (iii) is highly parallelizable. We emphasize that our method, by storing each participant's data encrypted by his own key, avoids a single point of failure in the system. If a key is leaked or hacked, only the data of a single participant is compromised and other participants' data is still protected. Conversely, previous solutions assume that all participants' data is stored encrypted under the same key, therefore, they are less secure as the leakage of such a key could jeopardize the entire system.

In a nutshell, we developed an efficient privacy-preserving algorithm for genetic association studies on encrypted genotypes and phenotypes stored in a centralized dataset. Our solution addresses the pervasive challenge of dataset stratification by inferring, in a privacy-preserving way, the ancestry of each subject in the dataset. Identification of dataset stratification represents a crucial preprocessing step of genetic association studies to avoid spurious associations due to systematic ancestry

[10] Our solution may also be used for GWAS, but it better scales for replication/fine-mapping association studies which are based on the *a priori* knowledge generated with GWAS.

differences within and between sample populations. Furthermore, our algorithm automatically generates case and control groups (i.e., two sets of individuals differing in one or more phenotypic traits) and outputs only the final result of the association study without leaking any information of the intermediate steps of the computation. We prove the security of the proposed technique and assess its performance with an implementation on real data. We also propose a MapReduce implementation as a proof-of-concept of parallelization.

One real-life application of the proposed technique is clinical studies conducted by pharmaceutical companies in collaboration with national biobanks. The goal of these studies is to assess the effectiveness of a treatment (or effect of a drug) for a certain group of people. In such a scenario, we can assume the biobank stores the encrypted genotypes and phenotypes of a set of individuals. Then, a pharmaceutical company can run a privacy-preserving genetic association study to identify *in a few hours* the set of genetic variants that influence the efficacy of the treatment. Today, these types of pharmacogenetic studies are performed through methods that are not privacy-preserving. Since biobanks cannot release data without the explicit consent from the participants or a special approval from an ethics committee, a pharmacogenetic study can require months to be completed. Therefore, the proposed technique not only preserves the privacy of the individuals' sensitive health-related data, but it also accelerates the pace of genomic research.

In general, genetic association studies involve a cohort of participants (P), who provide upon consent their genotype and phenotype information for research purposes, and a medical unit (MU) that performs the association study on this cohort. As discussed, the MU can be either a pharmaceutical company willing to conduct a clinical trial for a particular drug, or a research institution willing to test the association between some single nucleotide variations (SNVs) of significant interest and complex phenotypic traits. As shown in Fig. 18.9, the proposed system in [38] includes a certified institution (CI) and a centralized storage and processing unit (SPU), along with the P and the MU. The CI is responsible for (i) recruiting the participants for association studies, (ii) genotyping their genome (i.e., identifying and extracting their genetic variations), (iii) collecting their phenotype information, (iv) encrypting the data, and (v) generating and distributing the cryptographic keys between the parties.

We assume, for efficiency and security, the storage of encrypted genotypes and phenotypes to be at the SPU. That is, instead of several MUs storing the same large amount of genomic and phenomic data, the information of each participant is stored at a centralized SPU and, upon request, made accessible (for association studies) to different MUs. Storing genotype and phenotype information at the SPU also enables (i) data from multiple hosts to be pooled into a single and centralized repository, and (ii) genomic association studies to be conducted on an amount of data often beyond the capability of a sole researcher or institution. The purpose of such an architecture is to overcome the main limiting factor of association studies, i.e., insufficient sample size, as the individual effect of genomic differences is usually small, and large sample sizes are required in order to increase the sensitivity of statistical tests and data-mining techniques. As before, a private company (e.g.,

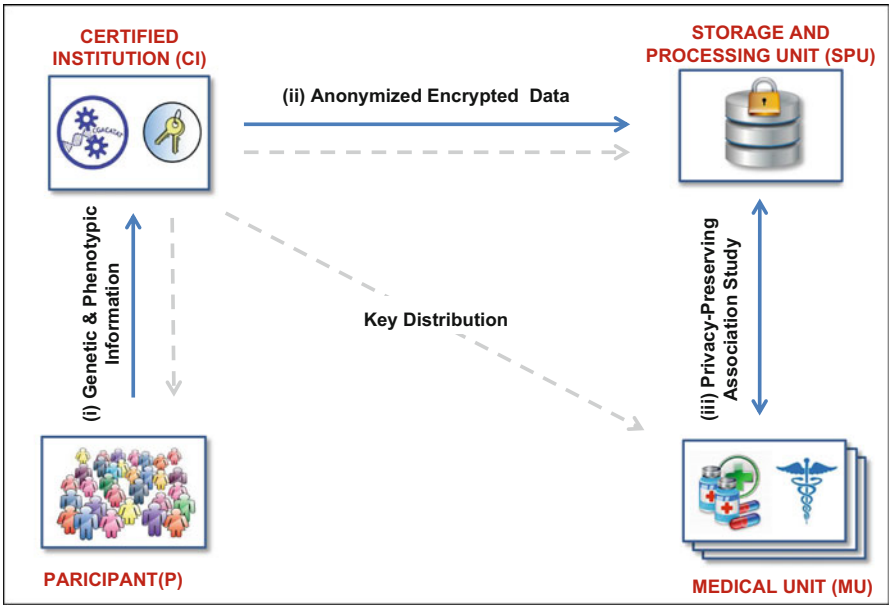


Fig. 18.9 System model for private use of genomic data in research setting [38]: participants (P), certified institution (CI), storage and processing unit (SPU), and medial units (MU)

cloud storage service), the government, or a non-profit organization can play the role of the SPU. The proposed algorithm for privacy-preserving genetic association studies takes place between the MU and the SPU.

The proposed solution in [38] can be summarized as follows. First, the participants provide to the CI their biological sample for genotyping, along with their phenotype information. Then, the CI encrypts each participant's information and sends it to the SPU. Finally, after a preprocessing phase for ancestry inference, the privacy-preserving genetic association study takes place between the MU and the SPU through a secure two-party protocol (using the homomorphic properties of the Paillier cryptosystem and some SMC protocols between the MU and the SPU). In such a protocol, the MU specifies the input parameters to the SPU and obtains only the allele frequencies for the two study groups.

18.2.4 Coping with Weak Passwords for the Protection of Genomic Data

Appropriately designed cryptographic schemes can preserve the data utility, but they provide security based on assumptions about the computational limitations of adversaries. Hence, they are vulnerable to brute-force attacks when these

assumptions are incorrect or erode over time. Given the longevity of genomic data, serious consequences can result. Compared with other types of data, genomic data has especially long-term sensitivity. A genome is (almost) stable over time and thus needs protection over the lifetime of an individual and even beyond, as genomic data is correlated between the members of a single family. It has been shown that the genome of an individual can be probabilistically inferred from the genomes of his or her family members [23].

In many situations, though, particularly those involving direct use of data by consumers, keys are weak and vulnerable to brute-force cracking *even today*. Users' tendency to choose weak passwords is widespread and well documented [12]. This problem arises in systems that employ password-based encryption (PBE), a common approach to protection of user-owned data.

Recently, Juels and Ristenpart introduced a new theoretical framework for encryption called *honey encryption* (HE) [27]. Honey encryption has the property that when a ciphertext is decrypted with an incorrect key (as guessed by an adversary), the result is a plausible-looking yet incorrect plaintext. Therefore, HE gives encrypted data an additional layer of protection by serving up fake data in response to every incorrect guess of a cryptographic key or password. Notably, HE provides a hedge against brute-force decryption in the long term, giving it a special value in the genomic setting.

However, HE relies on a highly accurate distribution-transforming encoder (DTE) over the message space. Unfortunately, this requirement jeopardizes the practicality of HE. To use HE in any scenario, we have to understand the corresponding message space quantitatively, that is, the precise probability of every possible message. When messages are not uniformly distributed, characterizing and quantifying the distribution is a highly non-trivial task. Building an efficient and precise DTE is the main challenge when extending HE to a real use case.

In Huang et al. [22], we proposed to address the problem of protecting genomic data by combining the idea of honey encryption with the special characteristics of genomic data in order to develop a secure genomic data storage (and retrieval) technique that is (i) robust against potential data breaches, (ii) robust against a computationally unbounded adversary, and (iii) efficient.

In the original HE paper [27], Juels and Ristenpart propose specific HE constructions that rely on existing generation algorithms (e.g., for RSA private keys), or operate over very simple message distributions (e.g., credit card numbers). These constructions, however, are inapplicable to plaintexts with considerably more complicated structure, such as genomic data. Thus, substantially new techniques are needed in order to apply HE to genomic data. Additional complications arise when the correlation between the genetic variants (on the genome) and phenotypic side information are taken into account. Our work in [38] is devoted mainly to addressing these challenges.

We proposed a scheme called GenoGuard. In GenoGuard [38], genomic data is encoded to generate a *seed* valve, the seed is encrypted under a patient's password,[11] and stored at a centralized biobank. We propose a novel tree-based technique to efficiently encode (and decode) the genomic sequence in order to meet the special requirements of honey encryption. Legitimate users of the system can retrieve the stored genomic data by typing their passwords.

A computationally unbounded adversary can break into the biobank protected by GenoGuard, or remotely try to retrieve the genome of a victim. The adversary could exhaustively try all the potential passwords in the password space for any genome in the biobank. However, for each password he tries (thanks to our encoding phase), the adversary will obtain a plausible-looking genome without knowing whether it is the correct one. We also consider the case when the adversary has side information about a victim (or victims) in terms of his physical traits. In this case, the adversary could use genotype-phenotype associations to determine the real genome of the victim. GenoGuard is designed to prevent such attacks, hence it provides protections beyond the normal guarantees of HE.

We show the main steps of the GenoGuard protocol in Fig. 18.10. We represent the patient and the user as two separate entities, but they can be the same individual, depending on the application.

GenoGuard is highly efficient and can be used by the service providers that offer DTC services (e.g., 23andMe) to securely store the genomes of their customers. It can also be used by medical units (e.g., hospitals) to securely store the genomes of

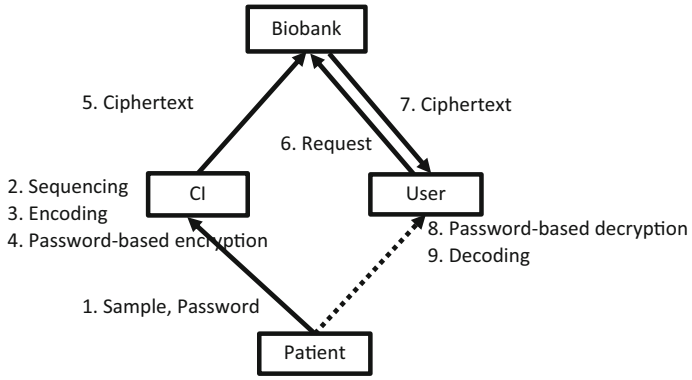


Fig. 18.10 GenoGuard protocol [38]. A patient provides his biological sample to the CI, and chooses a password for honey encryption. The CI does the sequencing, encoding and password-based encryption, and then sends the ciphertext to the biobank. During a retrieval, a user (e.g., the patient or his doctor) requests for the ciphertext, decrypts it and finally decodes it to get the original sequence

[11] A patient can choose a low-entropy password that is easier for him/her to remember, which is a common case in the real world [12].

patients and to retrieve them later for clinical use. The general protocol in Fig. 18.10 can work in a healthcare scenario without any major changes. In this scenario, a patient wants a medical unit (e.g., his doctor) to access his genome and perform medical tests. The medical unit can request for the encrypted seed on behalf of (and with consent from) the patient. Hence, there is a negotiation phase that provides the password to the medical unit. Such a phase can be completed automatically via the patient's smart card (or smart phone), or the patient can type his password himself. In this setup, the biobank can be a public centralized database that is semi-trusted. Such a centralized database would be convenient for the storage and retrieval of the genomes by several medical units.

For direct-to-customer (DTC) services, the protocol needs some adjustments. For instance, Counsyl[12] and 23andme[13] provide their customers various DTC genetic tests. In such scenarios, the biobank is the private database of these service providers. Thus, such service providers have the obligation to protect customers' genomic data in case of a data breach. In order to perform various genetic tests, the service providers should be granted permission to decrypt the sequences on their side, which is a reasonable relaxation of the threat model because customers share their sequences with the service providers. Therefore, steps 8 and 9 in Fig. 18.10 should be moved to the biobank. A user who requests a genetic test result logs into the biobank system, provides the password for password-based decryption and asks for a genetic test on his sequence. The plaintext sequence is deleted after the test.

18.2.5 Protecting Kin Genomic Privacy

In Humbert et al. [24], we presented a genomic-privacy preserving mechanism (GPPM) for reconciling people's willingness to share their genomes (e.g., to help research[14]) with privacy. Our GPPM acts at the individual data level, not at the aggregate data (or statistical) level like in [26]. Focusing on the most relevant type of variants (the SNPs), we study the trade-off between the usefulness of disclosed SNPs (utility) and genomic privacy. We consider an individual who wants to share his genome, yet who is concerned about the subsequent privacy risks for himself and his family. Thus, we design a system that maximizes the disclosure utility but does not exceed a certain level of privacy loss within a family, considering (i) kin genomic privacy, (ii) personal privacy preferences (of the family members), (iii) privacy sensitivities of the SNPs, (iv) correlations between SNPs, and (v) the research utility of the SNPs. The proposed GPPM in [24] can automatically evaluate the privacy risks of all the family members and decide which SNPs to disclose. To achieve

[12] https://www.counsyl.com/.

[13] https://www.23andme.com/.

[14] http://opensnp.wordpress.com/2011/11/17/first-results-of-the-survey-on-sharing-genetic-information/.

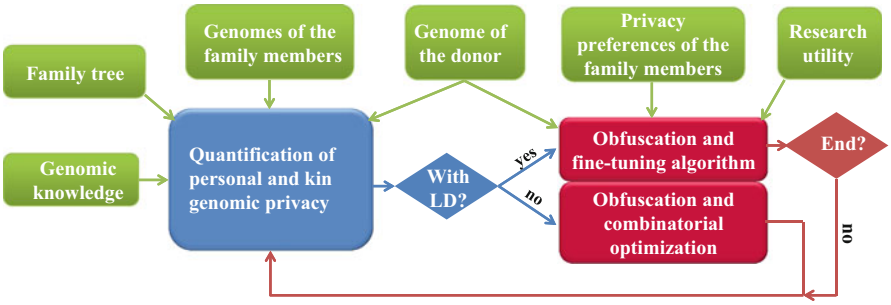


Fig. 18.11 General protection framework. The GPPM [24] takes as inputs (i) the privacy levels of all family members, (ii) the genome of the donor, (iii) the privacy preferences of the family members, and (iv) the research utility. First, correlations between the SNPs (LD) is not considered in order to use combinatorial optimization. Note that we go only once through this box. Then, LD is used and a fine-tuning algorithm is used to cope with non-linear constraints. The algorithm outputs the set of SNPs that the donor can disclose

this goal, it relies on probabilistic graphical models and combinatorial optimization. Our results indicate that, given the current data model, genomic privacy of an entire family can be protected while an appropriate subset of genomic data can be made available.

In order to mitigate attribute-inference attacks and protect genomic and health privacy, the GPPM relies upon an *obfuscation mechanism*. In practice, obfuscation can be implemented by adding noise to the SNP values, by injecting fake SNP values, by reducing precision, or by simply hiding the SNP values. In this work, we choose SNP hiding, essentially because the genomic research community would not receive other options positively. Indeed, genetic researchers are very reluctant about adding noise or fake data, notably because of the huge investment they make to increase (sequencing) accuracy. We assume one family member, at a given time, who wants to disclose his SNPs and to guarantee a minimum privacy level for him and his family. Figure 18.11 provides an overview of the proposed GPPM in [24].

For clarity of presentation, we focus on one family whose members are defined by the set $\mathbf{F}$ ($|\mathbf{F}| = n$). We assume that there is only one donor D who makes the decision to share his genome at a given time. His relatives might have already publicly shared some of their genomic data on the Internet. D takes this into account when he makes his own disclosure decision. We let $\mathbf{S}$ ($|\mathbf{S}| = m$) be the set of SNP IDs. Its cardinality m can go up to 50 million, as this is currently the approximate number of SNPs in the human population. In practice, however, people put online (e.g., on OpenSNP) up to one million of the most significant SNPs. We let $\mathbf{X}^D = \{x_j^D : j \in \mathbf{S}\}$ represent the set of SNPs of D (x_j^D is the value of SNP j of the donor D), that are all initially undisclosed. Finally, we let $\mathbf{y}^D = \{y_j^D : j \in \mathbf{S}\}$ represent the decision vector of D, where $y_j^D = 1$ means the corresponding SNP will be disclosed, and $y_j^D = 0$ means x_j^D will remain hidden.

We express the privacy constraints of a family member both in terms of genomic and health privacy. Our framework can account for different privacy preferences for different family members, SNPs, and diseases. For all $i \in \mathbf{F}$, $j \in \mathbf{S}$, we define the privacy sensitivity of a SNP j for individual i as s_j^i. We can set the s_j^i's to be equal by default. Then, an individual willing to personalize his privacy preferences may further define his own privacy sensitivities regarding specific SNPs based on his privacy concerns regarding, e.g., certain phenotypes. The most well-known example of such a scenario is the case of James Watson, co-discoverer of DNA, who made his whole DNA sequence publicly available, with the exception of one gene known as Apolipoprotein E (ApoE), one of the strongest predictors for the development of Alzheimer's disease.[15] We let the sets $\mathbf{P_s^i}$ and $\mathbf{P_d^i}$ include the privacy-sensitive SNP IDs and privacy-sensitive diseases of individual i, respectively. We represent the tolerance to the genomic-privacy loss of individual i as $\text{Pri}(i, \mathbf{P_s^i})$, and the tolerance to the health-privacy loss of individual i regarding disease $d \in \mathbf{P_d^i}$ as $\text{Pri}(i, d)$. These tolerance values represent the maximum privacy loss (after the disclosure of D's SNPs) that an individual would bear. By considering the privacy losses instead of the absolute privacy levels, we ensure that the donor will more likely reveal a SNP whose value is already well inferred by the attacker before donor's disclosure (e.g., by using SNPs previously shared by the donor's relatives). Note that these tolerance values can always be updated for any new family member willing to disclose his genome. Finally, the utility function is a non-decreasing function of the norm of $\mathbf{y}^D$, as the knowledge of more SNPs can only help genomic research. We define u_j to be the utility provided by SNP j. Note that, in practice, the utility of the SNPs can be determined by the research authorities and can vary based on the study.

The donor faces an optimization problem: How to maximize research utility while protecting his own and his relatives' genomic and health privacy. First, the objective function is formally defined as $\sum_{j \in \mathbf{S}} u_j y_j^D$. Then, privacy constraints are defined, for each individual, as the sum of privacy losses induced by the donor's disclosure over all SNPs. This sum must be capped by the respective privacy loss tolerances of all family members. Formally, for all individuals $i \in \mathbf{F}$ and SNPs $j \in \mathbf{S}$, the privacy loss induced by the disclosure of x_j^D is defined as $(E_j^i(y_j^D = 0) - E_j^i(y_j^D = 1))$. Note here that the privacy loss at a given SNP j for any relative is only affected by the donor's decision y_j^D regarding SNP j but no other SNP $k \neq j$, meaning that LD correlations are not taken into account. Finally, note that if an individual i has already revealed his SNP j, i.e., $x_j^i \in \mathbf{X_O}$, the privacy loss at this SNP for i is zero, because $E_j^i(y_j^D = 0) = E_j^i(y_j^D = 1) = 0$. For all $i \in \mathbf{F}$, $j \in \mathbf{S}$, the privacy weight p_j^i is defined as

$$p_j^i = s_j^i \times (E_j^i(y_j^D = 0) - E_j^i(y_j^D = 1)). \tag{18.1}$$

[15] Later researchers have used correlations in the genome to unveil Watson's predisposition to Alzheimer's [35]. In this work, we also consider such correlations.

Clearly, p_j^i at a given SNP j can be different for each family member, depending on how close he is from the donor in the family tree, on the actual values x_j^i and x_j^D of his and the donor's SNPs, and on his sensitivity. Note that $s_j^i = 0 \; \forall j \notin \mathbf{P_s^i}$.

We can now define the linear optimization problem as

$$\begin{aligned} \underset{\mathbf{y}^D}{\text{maximize}} \quad & \sum_{j \in \mathbf{S}} u_j y_j^D \\ \text{subject to} \quad & \sum_{j \in \mathbf{P_s^i}} p_j^i y_j^D \leq \text{Pri}(i, \mathbf{P_s^i}), \forall i \in \mathbf{F} \\ & \sum_{k \in \mathbf{S}_d} p_k^i y_k^D \leq \text{Pri}(i, d), \forall d \in \mathbf{P_d^i}, \forall i \in \mathbf{F} \\ & y_j^D \in \{0, 1\}, \forall j \in \mathbf{S}, \end{aligned} \tag{18.2}$$

where $\mathbf{S}_d$ is the set of SNPs that are associated with disease d.

Our optimization problem is very similar to the multidimensional knapsack problem [15]. We decide to follow the branch-and-bound method proposed by Shih [40], because it finds the optimal solution, represents a good trade-off between time and storage space, and allows for the extension of the algorithm to null and negative (privacy) weights. However, the LD correlations between the SNPs are not considered in the above optimization problem in order for the constraints to remain linear. Therefore, after getting the initial results from the linear optimization problem, we use a fine-tuning algorithm in order to decide to reveal less or more SNPs when LD is also considered.

18.3 Future Research Directions

Advances in genomics will soon result in large numbers of individuals having their genomes sequenced and obtaining digitized versions thereof. This poses a wide range of technical problems, which we explore below [3].

Storage and Accessibility: Genome at Rest Due to its sensitivity and size (about 3.2 billion nucleotides), one key challenge is where and how a digitized genome should be stored. It is reasonable to assume that an individual who requests (and likely pays for) genome sequencing should own the result, as is already the case with any other personal medical results and information. This raises numerous issues, including:

- Should the genome be stored on one's personal devices, e.g., a PC or a smartphone? If so, what, if any, special hardware security features (e.g., tamper-resistance) are needed?
- Can it be outsourced to a cloud provider?

- Should the sequencing facility keep an escrowed copy of the genome?
- Should it be entrusted to one's personal physician and/or health insurance provider?
- How is it to be stored: in the clear or encrypted? If the latter, where are encryption keys generated: at the lab? at owner's premises? at the cloud provider? Where are these keys stored?
- How to guarantee integrity and authenticity of the digitized genome?
- Should backups be made? If so, how often and where can copies be kept?
- How can one erase a genome securely?
- Should an individual periodically re-sequence their genome to take advantage of more accurate technology?

Privacy: Genome in Action Given the genome's sensitivity, an individual should, ideally, never disclose any information contained therein. However, this would prevent the access to any genomic application that cannot be entirely and securely performed *in situ*, i.e., within a secure perimeter of one's own personal device. In principle, this might be possible if operations are performed in some standardized and certified form. For example, if testing for a genetic disease requires matching a well-known pattern in some approximate location in the genome, that pattern and its parameters can be certified by some trusted agency (such as the US Food and Drug Administration). Thus, an individual could be assured that a legitimate test for a specific genetic disease is being conducted and the result is clearly communicated to that individual; the latter would then have the option to keep the result private.

At the same time, it is hard to foresee the range and complexity of future genetic operations: some (future) tests might be too computationally complex to be performed within the confines of a personal device. Furthermore, some genetic testing would probably involve multiple genomes, e.g., when tracing origins of some conditions, siblings or parents/children might need to be tested together. Similarly, in assessing risks of genetic conditions for future progeny, both prospective parents have to be tested. Also, some genetic tests constitute intellectual property of a pharmaceutical/biomedical company (which needs to be protected) [8, 19, 34].

As soon as genomic information leaves the (virtual) hands of its owner, purely technical approaches to privacy become insufficient. Legal and professional guidelines are certainly needed to govern how information is transmitted, stored, processed, and eventually disposed of on the receiving end, e.g., by the physician, hospital, pharmacist or medical lab.

Long-term Data Protection Even if genomes are encrypted, encryption schemes considered strong today might gradually weaken in the long term, whereas genome sensitivity does not dissipate over time. It is not too far-fetched to imagine that a third-party in possession of an encrypted genome might be able to decrypt it years or decades later. For instance, the Advanced Encryption Standard (AES) scheme supports key lengths up to 256 bits—a key length estimated by NIST, following Moore's law, to be secure several years after 2030. However, computational breakthroughs or unforeseen weaknesses might allow breaking the encryption earlier than

expected. Also, even leakage of a long-deceased individual's genome could affect genomic privacy of that person's living progeny.

Assuming that it cannot be copied, an encrypted genome could be periodically re-encrypted. Alternatively, one could split the genome (e.g., by using secret-sharing techniques [39]), and partition it among several providers. However, this opens the problem of efficient reassembly of the genome for various operations as well as how to guarantee non-collusion between providers.

Accuracy and Accountability Computational genomic tests should guarantee accuracy at least equivalent to that of their current analog *in vitro* counterparts. For example, a software implementation of the paternity test should offer at least the same confidence as its *in vitro* counterpart currently admissible in a court of law. Also, computational tests should aim at accountability, e.g., by providing lasting guarantees of correctness for both execution and input information.

Efficiency Computational genomic tests should incur minimal communication and computational costs. Minimality in this setting is relative to the context of such tests. For instance, patients may be inclined (and accustomed) to wait several days to obtain results of genetic tests that concern their health. However, in the computational setting, long running times on personal devices might hinder the real-world practicality of these tests (besides negating one of the main motivations for computational tests).

Usability Computational genomic tests that involve end-users should be usable by, and meaningful to, regular non-tech-savvy individuals. This translates into non-trivial questions, such as: how much understanding should be expected from a user running a test? What information (and at what level of granularity) should be presented to the user as part of a test and as its outcome? Do privacy perceptions and concerns experienced by patients match those expected by the scientific community? Some users might be willing to forego their genomic privacy in certain cases. For instance, one may think that patients will be likely to reveal their genomes to their medical doctors (and hence trade off privacy of their genomes) to enable tests that can save them from, e.g., cancer. In contrast, in the case of online services or pharmaceuticals, an individual might not wish to forgo privacy. However, very few efforts (e.g., [13]) have focused on users' concerns, thus prompting the need for ethnographic studies. Also, there remains an open problem of how to effectively communicate to the users potential privacy risks associated with genomic information and its disclosure.

Large-scale Research on Human Genomes Potential privacy, legal, and ethical concerns appear to conflict with large-scale research on human genomes, such as Genome-Wide Association Studies (GWAS). However, large scale studies are needed to discover associations between genetic make-up and medical conditions. One current trend is to store donors' genomes in the cloud and use analytics techniques running on powerful computer clusters. Once again, this prompts many privacy and legal concerns.

18.4 Conclusion

In this chapter, focusing on the work carried out by EPFL/LCA1 and Bilkent University, we first discussed some threats for genomic privacy. Then, we described some of our solutions to protect genomic privacy. Namely, we focused on privacy-preserving management of raw genomic data, privacy compliant use of genomic data in personalized medicine and research settings, resistance to brute-force attacks and protecting kin genomic privacy. Finally, we discussed some future research directions. More information on this topic can be found at: https://genomeprivacy.org

Acknowledgements The authors would like to express their gratitude to Mathias Humbert, Jean Louis Raisaro, Zhicong Huang, Emiliano De Cristofaro, Gene Tsudik, Jacques Fellay, Amalio Telenti and Paul Mc Laren.

References

1. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order preserving encryption for numeric data. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 563–574 (2004)
2. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. **9**, 1–30 (2006)
3. Ayday, E., Cristofaro, E.D., Tsudik, G., Hubaux, J.-P.: Whole genome sequencing: revolutionary medicine or privacy nightmare. IEEE Computet **48**(2), pp. 58–66 (2015)
4. Ayday, E., Raisaro, J.L., Hengartner, U., Molyneaux, A., Hubaux, J.-P.: Privacy-preserving processing of raw genomic data. In: Proceeding of 8th International Workshop on Data Privacy Management (DPM). Egham, UK (2013)
5. Ayday, E., Raisaro, J.L., Mclaren, P.J., Fellay, J., Hubaux, J.-P.: Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data. In: Proceedings of USENIX Security Workshop on Health Information Technologies (HealthTech) (2013)
6. Ayday, E., Raisaro, J.L., Rougemont, J., Hubaux, J.-P.: Protecting and evaluating genomic privacy in medical tests and personalized medicine. In: CM Workshop on Privacy in the Electronic Society (WPES). Berlin, Germany (2013)
7. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Proceedings of Asiacrypt (2003)
8. Caulfield, T., Cook-Deegan, R.M., Kieff, F.S., Walsh, J.P.: Evidence and anecdotes: an analysis of human gene patenting controversies. Nat. Biotechnol. **24**(9), pp. 1091–1094 (2006)
9. Clayton, D.: On inferring presence of an individual in a mixture: a bayesian approach. Biostatistics **11**(4), 661–673 (2010)
10. Drmanac, R., Sparks, A.B., Callow, M.J., Halpern, A.L., Burns, N.L., Kermani, B.G., Carnevali, P., Nazarenko, I., Nilsen, G.B., Yeung, G., et al.: Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. Science **327**(5961), 78–81 (2010)
11. Erlich, Y., Narayanan, A.: Routes for breaching and protecting genetic privacy. Nat. Rev. Genet. **15**(6), 409–421 (2014)
12. Florencio, D., Herley, C.: A large-scale study of web password habits. In: Proceedings of the 16th International Conference on World Wide Web, WWW '07, pp. 657–666. ACM, New York (2007). doi:10.1145/1242572.1242661. url:http://doi.acm.org/10.1145/1242572.1242661

13. Francke, U., Dijamco, C., Kiefer, A.K., Eriksson, N., Moiseff, B., Tung, J.Y., Mountain, J.L.: Dealing with the unexpected: consumer responses to direct-access BRCA mutation testing. PeerJ **1** (2013)
14. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In: Proceedings of the 23rd USENIX Security Symposium (2014)
15. Fréville, A.: The multidimensional 0–1 knapsack problem: an overview. Eur. J. Oper. Res. **155**(1), 1–21 (2004)
16. Gitschier, J.: Inferential genotyping of y chromosomes in latter-day saints founders and comparison to Utah samples in the hapmap project. Am. J. Hum. Genet. **84**(2), 251–258 (2009)
17. Google Genomics: (2015) https://cloud.google.com/genomics/
18. Gymrek, M., McGuire, A.L., Golan, D., Halperin, E., Erlich, Y.: Identifying personal genomes by surname inference. Science **339**(6117), 321–324 (2013)
19. Hawkins, N.: The impact of human gene patents on genetic testing in the UK. J. Gene Med. **13**(4), pp. 320–324 (2011)
20. Hayden, E.C.: Privacy protections: the genome hacker. Nature **497**, 172–174 (2013)
21. Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W.: Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS Genet. **4** (2008)
22. Huang, Z., Ayday, E., Hubaux, J.-P., Fellay, J., Juels, A.: Genoguard: protecting genomic data against brute-force attacks. In: Proceedings of IEEE Symposium on Security and Privacy (2015)
23. Humbert, M., Ayday, E., Hubaux, J.-P., Telenti, A.: Addressing the concerns of the Lacks family: quantification of kin genomic privacy. In: Proceeding of the 20th ACM Conference on Computer and Communications Security (CCS) (2013)
24. Humbert, M., Ayday, E., Hubaux, J.-P., Telenti, A.: Reconciling utility with privacy in genomics. In: Proceedings of ACM Workshop on Privacy in the Electronic Society (WPES) (2014)
25. Im, H.K., Gamazon, E.R., Nicolae, D.L., Cox, N.J.: On sharing quantitative trait GWAS results in an era of multiple-omics data and the limits of genomic privacy. Am. J. Hum. Genet. **90**(4), 591–598 (2012)
26. Johnson, A., Shmatikov, V.: Privacy-preserving data exploration in genome-wide association studies. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), pp. 1079–1087 (2013)
27. Juels, A., Ristenpart, T.: Honey encryption: security beyond the brute-force bound. In: Advances in Cryptology–EUROCRYPT, pp. 293–310 (2014)
28. Kamm, L., Bogdanov, D., Laur, S., Vilo, J.: A new way to protect privacy in large-scale genome-wide association studies. Bioinformatics. 2013 Apr 1;29(7):886-93
29. Kantarcioglu, M., Jiang, W., Liu, Y., Malin, B.: A cryptographic approach to securely share and query genomic sequences. IEEE Trans. Inf. Technol. Biomed. **12**(5), 606–617 (2008). doi: 10.1109/TITB.2007.908465
30. Kschischang, F., Frey, B., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theory **47**, pp. 498–519 (2001)
31. Lin, Z., Owen, A.B., Altman, R.B.: Genomic research and human subject privacy. Science **305**(5681), 183 (2004)
32. Loukides, G., Gkoulalas-Divanis, A., Malin, B.: Anonymization of electronic medical records for validating genome-wide association studies. PNAS **107**(17), 7898–7903 (2010)
33. Malin, B.A., Sweeney, L.: How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. J. Biomed. Inform. **37**(3), 179–192 (2004)
34. National Human Genome Research Institute: Intellectual Property and Genomics. (2015) http://www.genome.gov/19016590
35. Nyholt, D., Yu, C., Visscher, P.: On Jim Watson's APOE status: genetic information is hard to hide. Eur. J. Hum. Genet. **17**, 147–149 (2009)

36. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Mateo (1988)
37. Popa, R.A., Li, F.H., Zeldovich, N.: An ideal-security protocol for order-preserving encoding. In: Proceedings of the 2013 IEEE Symposium on Security and Privacy (2013)
38. Raisaro, J.L., Ayday, E., McLaren, P., Telenti, A., Hubaux, J.P.: On a novel privacy-preserving framework for both personalized medicine and genetic association studies. In: Privacy-Aware Computational Genomics (PRIVAGEN) (2015)
39. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
40. Shih, W.: A branch and bound method for the multiconstraint zero-one knapsack problem. J. Oper. Res. Soc. **30**, 369–378 (1979)
41. Stajano, F., Bianchi, L., Liò, P., Korff, D.: Forensic genomics: kin privacy, driftnets and other open questions. In: Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society (2008)
42. Sweeney, L., Abu, A., Winn, J.: Identifying Participants in the Personal Genome Project by Name. Harvard University, Cambridge (2013)
43. Wang, R., Li, Y.F., Wang, X., Tang, H., Zhou, X.: Learning your identity and disease from research papers: information leaks in genome wide association study. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 534–544 (2009)
44. Yu, F., Fienberg, S.E., Slavkovic, A.B., Uhler, C.: Scalable privacy-preserving data sharing methodology for genome-wide association studies. J. Biomed Inform. 2014 Aug;50:133-41
45. Zhou, X., Peng, B., Li, Y.F., Chen, Y., Tang, H., Wang, X.: To release or not to release: evaluating information leaks in aggregate human-genome data. In: Proceedings of the 16th European Conference on Research in Computer Security (ESORICS'11), pp. 607–627 (2011)