Cryptographic Solutions for Genomic Privacy

Erman Ayday(✉)

Computer Engineering Department, Bilkent University,
06800 Bilkent, Ankara, Turkey
erman@cs.bilkent.edu.tr

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. In this work, focusing on our existing and ongoing work on genomic privacy, we will first highlight one serious threat for genomic privacy. Then, we will present the high level descriptions of our cryptographic solutions to protect the privacy of genomic data.

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 On the one hand, the family members think that her genome is private family information and it should not be published without the consent of the family.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 shown in [10] (that we briefly describe in the latter), they are 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.

J. Clark et al. (Eds.): FC 2016 Workshops, LNCS 9604, pp. 328–341, 2016.
DOI: 10.1007/978-3-662-53357-4_22

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 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 [13]. 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 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 this work, we evaluate 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), 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.[1] 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 [12,14] for inference. Then, using various metrics, we quantify the genomic privacy of individuals and show the decrease in their level of genomic privacy caused 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. Figure 1 gives an overview of the framework.

[1] 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 SNPs.

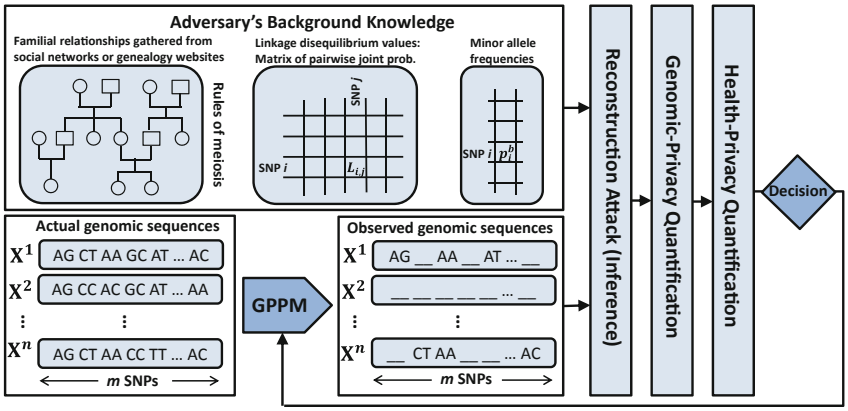

Fig. 1. Overview of the proposed framework to quantify kin genomic privacy. Each vector $\mathbf{X}^i$ ($i \in \{1, \ldots, n\}$) includes the set of SNPs for an individual in the targeted family. Furthermore, each letter pair in $\mathbf{X}^i$ represents a SNP x_j^i; and for simplicity, each SNP x_j^i can be represented using $\{BB, Bb, bb\}$ (or $\{0, 1, 2\}$). Linkage disequilibrium (LD) can be thought as a correlation between two variables (SNPs) and minor allele frequency can be considered as the probability of observing a SNP in the population. 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).

In a nutshell, the goal of the adversary is to infer the unknown (unobserved) SNPs of a member (or multiple members) of a *targeted family*. For the evaluation, we use the *CEPH/Utah Pedigree 1463* that contains the partial DNA sequences of 17 family members (4 grandparents, 2 parents, and 11 children) [7]. As shown in Fig. 2 that we only use 5 (out of 11) children for our evaluation.

We consider 100 SNPs on chromosome 1. We define a target individual from the CEPH family 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. 2) and we keep revealing relatives until we reach his/her closest family members.[2] 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 out of 100 only, and these sets of observed SNPs are different for each family member. In Fig. 3, we show the evolution of genomic privacy of one target individual (P5). We quantify the genomic privacy based on (i) attackers in correctness (red plot), (ii) attacker's uncertainty (green plot), and (iii) an entropy-based metrics that quantifies the mutual dependence between the hidden genomic data that

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

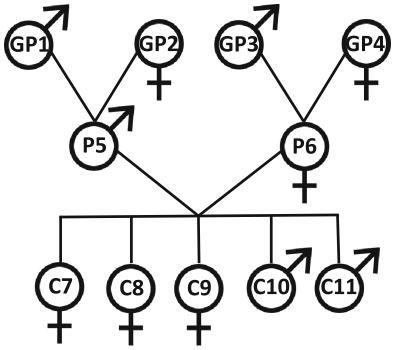

Fig. 2. Family tree of *CEPH/Utah Pedigree 1463* consisting of the 11 family members that were considered. The symbols ♂ and ♀ represent the male and female family members, respectively.

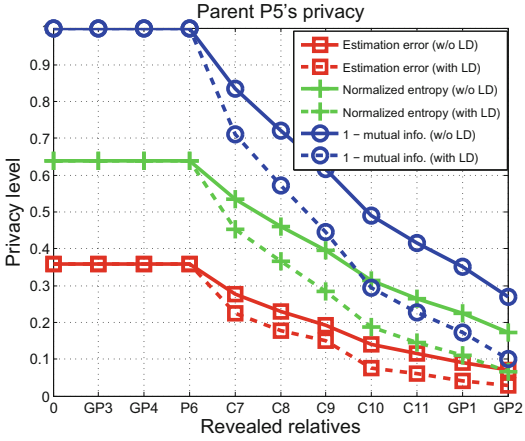

Fig. 3. Evolution of the genomic privacy of the parent (P5), with and without considering LD. For each family member, we reveal 50 randomly picked SNPs (out of 100 SNPs on chromosome 1), starting from the most distant family members, and the x-axis represents the exact sequence of this disclosure. Note that $x = 0$ represents the prior distribution, when no genomic data is revealed.

the adversary is trying to reconstruct (blue 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 different 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 genetic 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.

2 Solutions for Genomic Privacy

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 this part, we summarize some of those efforts by focusing on privacy-preserving use of genomic data in personalized medicine and post-quantum privacy for storage of genomic data, and protecting kin genomic privacy.

2.1 Private Use of Genomic Data in Personalized Medicine

As we have shown in [5], our goal is 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 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 introduce the *privacy-preserving disease susceptibility test* (PDS).

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 malicious 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 malicious 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 which it is not authorized. The main resource of such an attacker is the results of the genetic tests 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.[3]

For each patient, we propose to store only the real SNPs (around 4 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 4 million real SNPs (nucleotides) could be easily stored on the patient's computer or mobile device, instead of at the 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*

[3] In this study, we only focus 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.

(CNVs), rearrangements, or translocations, consequently the required storage per patient is likely to be considerably more than only 4 million nucleotides. This high storage cost might still be affordable (via desktop computers or USB drives), however, the 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 as soon as possible, 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. Therefore, we propose 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 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 [6], 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 [2,6] to support the homomorphic operations at the MC. We show our proposed protocol in Fig. 4.

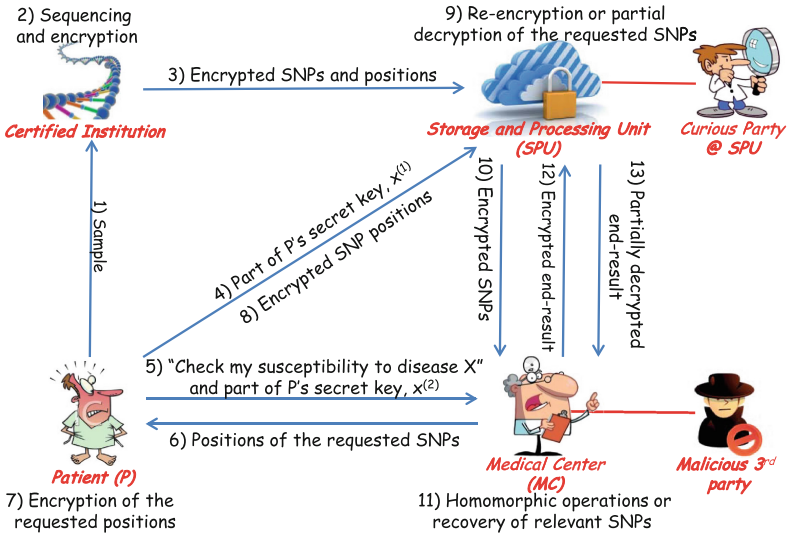

Fig. 4. Proposed privacy-preserving disease susceptibility test (PDS).

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 then 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. The SPU partially decrypts the end-result 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. We also show 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 [4], we also propose 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 introduce 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 malicious party (e.g., a hacker) at the medical unit when computing the disease risk. We also implement the proposed system and show its practicality via a complexity evaluation.

The general architecture of the proposed system is illustrated in Fig. 5. In summary, the patient provides his sample for sequencing to the CI. Meanwhile,

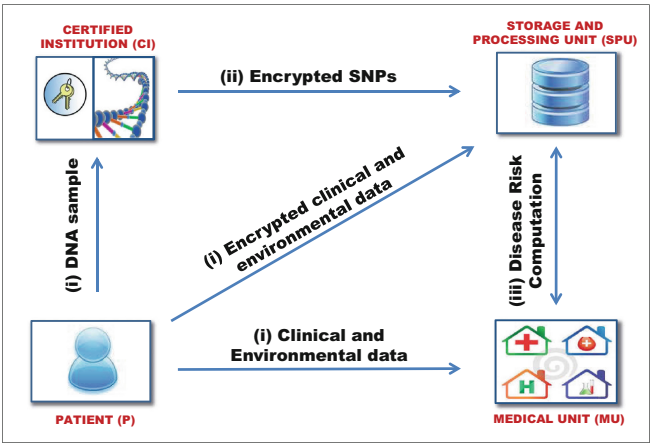

Fig. 5. Proposed system model for the privacy-preserving computation of the disease risk.

he also provides his clinical and environmental data to the SPU and the MU.[4] 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.

2.2 Coping with Weak Passwords for the Protection of Genomic Data

Appropriately designed cryptographic schemes can preserve the utility of data, 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 family members [10].

In many situations, though, particularly those involving direct use of data by consumers, keys are weak and vulnerable to brute-force cracking *even today*. This problem arises in systems that employ password-based encryption (PBE), a common approach to protection of user-owned data. Users' tendency to choose weak passwords is widespread and well documented [8].

Recently, Juels and Ristenpart introduced a new theoretical framework for encryption called *honey encryption* (HE) [11]. 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, and it is what we do in this work. Hopefully, the techniques proposed in this work are not limited to genomic data; they are intended to inspire those who want to apply HE to other scenarios, typically when the data shares similar characteristics with genomic data.

[4] 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.

As we have shown [9], we propose 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 [11], 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. This work is devoted mainly to addressing these challenges.

We propose a scheme called GenoGuard. In GenoGuard, genomic data is encoded, encrypted under a patient's password[5], and stored at a centralized biobank. We propose a novel tree-based technique to efficiently encode (and decode) the genomic sequence 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. 6. 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 patients and to retrieve them later for clinical use. The general protocol in Fig. 6 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

[5] 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 [8].

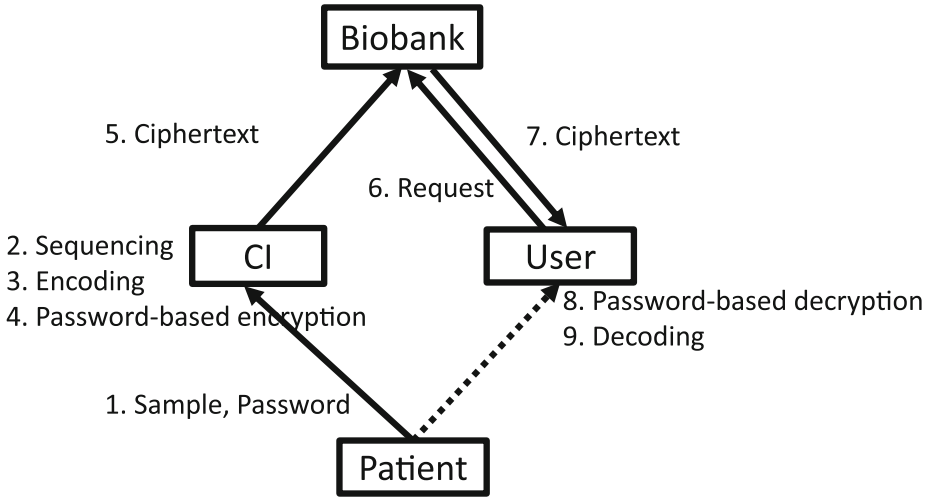

Fig. 6. GenoGuard protocol. A patient provides his biological sample to the CI, and chooses a password for honey encryption. The CI does the sequencing, encoding and password-based encryption, and then sends the ciphertext to the biobank. During a retrieval, a user (e.g., the patient or his doctor) requests for the ciphertext, decrypts it and finally decodes it to get the original sequence.

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[6] and 23andMe[7] 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. 6 should be moved to the biobank. A user (customer) 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.

[6] https://www.counsyl.com/.
[7] https://www.23andme.com/.

3 Conclusions

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 also explore in detail in a recent work [3]. Mitigating privacy issues of genomic data will require long-term collaboration among geneticists, other healthcare providers, ethicists, lawmakers, and computer scientists. In order to foster this collaboration, funding agencies need to target this topic. There are numerous EU, US, and nationally funded projects focusing on e-health, some of which address data protection. However, the genomic privacy challenge has been overlooked, and the number of computer scientists working on the topic is currently low. We hope that the privacy issues highlighted here will encourage collaboration among researchers in the fields outlined above. We believe that consideration of such privacy issues will have a positive benefit to society and individuals in their daily lives.

References

1. http://www.nytimes.com/2013/03/24/opinion/sunday/the-immortal-life-of-henrietta-lacks-the-sequel.html?pagewanted=all
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 Comput. Mag. **48**(2), 58–66 (2015)
4. 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)
5. Ayday, E., Raisaro, J.L., Rougemont, J., Hubaux, J.-P.: Protecting and evaluating genomic privacy in medical tests and personalized medicine. In: WPES 2013 (2013)
6. 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)
7. 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)
8. Florencio, D., Herley, C.: A large-scale study of web password habits. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 657–666. ACM, New York (2007)
9. 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)
10. Humbert, M., Ayday, E., Hubaux, J.-P., Telenti, A.: Addressing the concerns of the Lacks family: quantification of kin genomic privacy. In: CCS 2013 (2013)

11. Juels, A., Ristenpart, T.: honey encryption: security beyond the brute-force bound. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 293–310. Springer, Heidelberg (2014)
12. Kschischang, F., Frey, B., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theor. **47**, 498–519 (2001)
13. 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)
14. Pearl, J., Reasoning, P.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)