Privacy-Enhancing Technologies for Medical Tests Using Genomic Data

Erman Ayday, Jean Louis Raisaro and Jean-Pierre Hubaux
School of Computer and Communication Sciences
Ecole Polytechnique Federale de Lausanne (EPFL)
firstname.lastname@epfl.ch

***Abstract*—We propose privacy-enhancing technologies for medical tests and personalized medicine methods, which utilize patients' genomic data. Focusing specifically on a typical disease-susceptibility test, we develop a new architecture (between the patient and the medical unit) and propose a privacy-preserving algorithm by utilizing homomorphic encryption and proxy re-encryption. Assuming the whole genome sequencing is done by a certified institution, we propose to store patients' genomic data encrypted by their public keys at a Storage and Processing Unit (SPU). The proposed algorithm lets the SPU process the encrypted genomic data for medical tests and personalized medicine methods while preserving the privacy of patients' genomic data. Furthermore, we implement and show via a complexity analysis the practicality of the proposed scheme.**

I. Introduction

As a result of the rapid evolution in genomic research, substantial progress is expected in terms of improved diagnosis and better preventive medicine. However, the impact on privacy is unprecedented, because (i) genetic diseases can be unveiled, (ii) the propensity to develop specific diseases (such as Alzheimer's) can be revealed, (iii) a volunteer accepting to have his genomic code made public can leak substantial information about his ethnic heritage and genomic data of his relatives (possibly against their will), and (iv) complex privacy issues can arise if DNA analysis is used for criminal investigations and insurance purposes. Such issues could lead to genetic discrimination. Even though the Genetic Information Non-discrimination Act (GINA), which prohibits the use of genomic information in health insurance and employment, attempted to solve some of these problems in the US, these types of laws are very difficult to enforce.

Due to the sensitivity of genomic data, research on the privacy of genomic data has considerably accelerated over the past few years. In [1], Troncoso-Pastoriza *et al.* propose a protocol for string searching, which is then extended by Blanton and Aliasgari [2]. To compute the similarity of DNA sequences, in [3], Jha *et al.* propose techniques for privately computing the edit distance of two strings by using garbled circuits. In [4], Bruekers *et al.* propose privacy-enhanced comparison of DNA profiles for identity, paternity and ancestry tests using homomorphic encryption. In [5], Kantarcioglu *et al.* propose using homomorphic encryption to perform scientific investigations on integrated genomic data. In one of the recent works [6], Baldi *et al.* make use of both medical and cryptographic tools for privacy-preserving paternity tests, personalized medicine, and genetic compatibility tests. Finally, instead of utilizing public key encryption protocols, in [7], Canim *et al.* propose securing the biomedical data using cryptographic hardware.

As a consequence of our extensive collaboration with geneticists, clinicians, and biologists, we conclude that DNA string comparison is insufficient in many medical tests (that use genomic data) and would not be enough to pave the way to personalized medicine. As it will become clearer in the next sections, specific variants (i.e., nucleotides which reside at particular positions in the genome and vary between individuals) must be considered individually for each genetic test. Thus, as opposed to the aforementioned private string search and comparison techniques, which focus on privately comparing the distance between the genomic sequences, we use the individual variants of the users to conduct genetic disease susceptibility tests and develop personalized medicine methods. Therefore, in this work, 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.[1] In a medical test, a medical center 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 on a particular medicine, or a pharmacist checks the compatibility of a given medicine (e.g., over-the-counter drug) to a given user. In both scenarios, in order to preserve his privacy, the user does not want to reveal his complete genome to the medical center or to the pharmaceutical company. To achieve this goal, we propose to store the genomic data at a *Storage and Processing Unit* (SPU) and conduct the computations on genomic data utilizing homomorphic encryption and proxy re-encryption to preserve the privacy of the genomic data.

II. Privacy- Preserving Medical Tests and Personalized Medicine Methods

Most medical tests and personalized medicine methods (that use genomic data) involve a patient and a medical unit. In general, the medical unit is the family doctor, a physician, a pharmacist, or a medical council. In this study, we consider a malicious medical unit as the potential attacker. That is, a medical unit can be a malicious institution trying to obtain private information about a patient. Even if the medical unit is non-malicious, it is extremely difficult for medical units to protect themselves against the misdeeds of a hacker or a disgruntled employee. Similarly, the genomic data is too sensitive to be stored on users' personal devices (mostly due to security, availability, and storage issues), hence it is risky to leave the users' genomic data in their own hands. Thus, we believe that a Storage and Processing Unit (SPU) should be used to store and process the genomic data. 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 (e.g., existence of a curious party at the SPU), hence genomic data should be stored at the SPU in encrypted form.

[1] An extended version of this work is available in [8].

We also assume the SPU does not have access to the real identities of the patients and data is stored at the SPU by using pseudonyms; this way, the SPU cannot associate the conducted genomic tests to the real identities of the patients.

For the simplicity of presentation, in the rest of this work, we will focus on a particular medical test (namely, computing genetic disease susceptibility). We note that similar techniques would apply for other medical tests and personalized medicine methods. In a typical disease-susceptibility test, a medical center (MC) wants to check the susceptibility of a patient (P) to a particular disease X (i.e., probability that the patient P will develop disease X). It is shown that a genetic disease-susceptibility test can be realized by analyzing particular Single Nucleotide Polymorphisms (SNPs) of the patient via some operations [9], [10]. A SNP is a position in the genome holding a nucleotide (A, T, C or G), which varies between individuals. Each SNP contributes to the susceptibility in a different amount and the contribution amount of each SNP is determined by previous studies on case and control groups.

In general, there are two alleles (nucleotides) observed at a given SNP position: (i) The major allele is the most frequently observed nucleotide, and (ii) the minor allele is the rare nucleotide. Everyone inherits one allele of every SNP position from each of his parents. If an individual receives the same minor allele from both parents, he is said to have a *homozygous* variant for that SNP position. If, however, he inherits a different allele from each parent (one minor and one major), he has a *heterozygous* variant. There are approximately 40 million approved SNPs in the human population as of now (according to the NCBI dbSNP [11]) and each patient carries on average 4 million SNPs (i.e., homozygous or heterozygous variants) out of this 40 million. Moreover, this set of 4 million SNPs is different for each patient. From now on, to avoid confusion, for each patient, we refer to these 4 million variants as the *real SNPs* and the remaining non-variants (approved SNPs that do not exist for the considered patient) as the *potential SNPs* of the patient; when we only say "SNPs", we mean both the real and potential SNPs. In the rest of this work, for simplicity of the presentation, we do not consider the type of the variant at a real SNP position (i.e., whether the variation is homozygous or heterozygous for that real SNP); we only consider whether the patient has a real SNP or not at a particular position.

A. *Proposed Solution*

We assume that the state of SNP_i at the patient P is represented as SNP_i^P and $\text{SNP}_i^P = 1$, if P has a real SNP (i.e., variant) at this position, and $\text{SNP}_i^P = 0$, if P does not have a variant at this position. We let Υ_P be the set of real SNPs of the patient P (at which $\text{SNP}_i^P = 1$). We also let Ω_P represent the set of potential SNPs (at which $\text{SNP}_i^P = 0$). As the positions of the SNPs are stored in plaintext, if the SPU only stores the real SNPs in Υ_P, a curious party at the SPU can learn all real SNP positions of the patient, and hence, much about his genomic sequence. Therefore, the SPU stores the states of both real and potential SNP positions (in $\{\Upsilon_P \cup \Omega_P\}$) in order to preserve the privacy of the patient. Below, we summarize the proposed approach for the privacy protecting disease-susceptibility test. This approach is illustrated in Fig. 1.

- **Step 0:** The Cryptographic keys (public and secret keys) of each patient are generated and distributed to the patients during the initialization period. Then, symmetric keys are established between the parties, using which the communication between the parties is protected from an eavesdropper. We note that the distribution, update and revocation of cryptographic keys are handled by a trusted entity (similar to e-banking platforms).

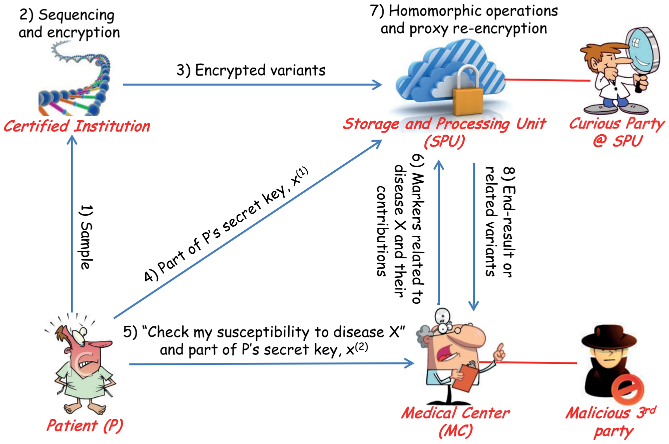
2) Sequencing and encryption
3) Encrypted variants
7) Homomorphic operations and proxy re-encryption
Certified Institution
Storage and Processing Unit (SPU)
Curious Party @ SPU
1) Sample
4) Part of P's secret key, $x^{(1)}$
6) Markers related to disease X and their contributions
8) End-result or related variants
5) "Check my susceptibility to disease X" and part of P's secret key, $x^{(2)}$
Patient (P)
Medical Center (MC)
Malicious 3rd party

Fig. 1. Privacy-preserving protocol for disease-susceptibility test.

- **Step 1:** The patient (P) provides his sample (e.g., his saliva) to the Certified Institution (CI) for sequencing.
- **Step 2:** The whole genome sequencing is done by the CI with the consent of the patient. Moreover, the CI encrypts the states of the patient's real and potential SNP positions (in $\{\Upsilon_P \cup \Omega_P\}$) by using P's public key. We use a modification of the Paillier cryptosystem [12], [13] (for encryption) to support the homomorphic operations (i.e., addition of two encrypted messages and multiplication of an encrypted message with a constant) at the SPU. The public key of the patient P is represented as $(n, g, h = g^x)$, where the strong secret key is the factorization of $n = pq$ (p, q are safe primes), the weak secret key is $x \in [1, n^2/2]$, and g of order $(p-1)(q-1)/2$. We are aware that the number of discovered SNPs increases with time. Thus, the patient's complete DNA sequence is also encrypted as a single vector file (via symmetric encryption using the patient's key) and stored at the SPU, thus when new SNPs are discovered, these can be included in the pool of the previously stored SNPs of the patient.
- **Step 3:** The CI sends the encrypted SNPs of P to the SPU (so that the SPU cannot access to P's SNPs).
- **Step 4:** The patient's weak secret key x is randomly divided into two shares: $x^{(1)}$ and $x^{(2)}$ (such that $x = x^{(1)} + x^{(2)}$). $x^{(1)}$ is given to the SPU (at this step) and $x^{(2)}$ is given to the MC (at the next step). Using the Paillier cryptosystem, an encrypted message (under the patient's public key) can be partially decrypted by the SPU using $x^{(1)}$ (i.e., proxy re-encryption), and then decrypted at the MC using $x^{(2)}$ to recover the original message.
- **Step 5:** The MC wants to conduct a susceptibility test on P to a particular disease X, and P provides the other part of his secret key ($x^{(2)}$) to the MC.
- **Step 6:** The MC provides genetic variant markers, along with their contributions (to the disease susceptibility), to the SPU.
- **Step 7:** Depending on the access rights of the MC and the virtue of the test, the SPU either (i) computes $\Pr(X)$, the probability that the patient will develop the disease X by checking the patient's encrypted SNPs via homomorphic operations (as discussed in Section II-B), or (ii) 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 by the medical authorities.

- **Step 7:** The SPU partially decrypts the end-result (or the relevant SNPs) using a part of P's secret key ($x^{(1)}$) following the proxy re-encryption protocol.
- **Step 8:** The SPU sends the partially decrypted end-result (or the relevant SNPs) to the MC.
- **Step 9:** The MC decrypts the message received from the SPU using the other part of P's secret key ($x^{(2)}$) and recovers the end-result (or the relevant SNPs).

B. Computing Disease Susceptibility at the SPU

In the following, we discuss how to compute the predicted disease susceptibility at the SPU by using homomorphic operations. There are different functions for computing the predicted susceptibility (e.g., counting the number of unfavorable alleles [9] or multiplying likelihood ratios of the most important SNPs for a particular disease [10]). We use the *weighted averaging* function (which is an advanced version of [9]) which computes the predicted susceptibility by weighting the contributions of SNPs by their contributions. We note that the function proposed in [10] can also be utilized similarly.

Assume that the susceptibility to disease X is determined by the set of SNPs $\Omega = \{\mathrm{SNP}_m,\mathrm{SNP}_n\}$, which occur at particular positions of the DNA sequence. The contributions of different states of SNP_i^P for $i \in \{m,n\}$ to the susceptibility to disease X are computed via previous studies and they are already known by the MC. That is, $\mathrm{p}_0^i(X) \triangleq \Pr(X|\mathrm{SNP}_i^P = 0)$ and $\mathrm{p}_1^i(X) \triangleq \Pr(X|\mathrm{SNP}_i^P = 1)$ ($i \in \{m,n\}$) are determined and known by the MC. Further, the contribution of SNP_i to the susceptibility to disease X is denoted by C_i^X.

The SPU uses P's encrypted SNPs ($\mathrm{E}(\mathrm{SNP}_m^P, g^x)$ and $\mathrm{E}(\mathrm{SNP}_n^P, g^x)$) for the computation of predicted susceptibility of P to disease X. Similarly, the MC provides the following to the SPU in plaintext: (i) the markers for disease X (SNP_m and SNP_n), (ii) corresponding probabilities ($\mathrm{p}_j^i(\mathrm{X})$, $i \in \{m,n\}$ and $j \in \{0,1\}$), and (iii) the contributions of each SNP (C_i^X). Next, the SPU encrypts j ($j \in \{0,1\}$) using P's public key to obtain $\mathrm{E}(0, g^x)$ and $\mathrm{E}(1, g^x)$ for the homomorphic computations. The SPU computes the predicted susceptibility of the patient P to disease X by using weighted averaging. This can be computed in plaintext as below:

$$\mathbb{S}_P^X = \frac{1}{C_m^X + C_n^X} \times \sum_{i \in m,n} C_i^X \left\{ \frac{\mathrm{p}_0^i(X)}{(0-1)} \left[\mathrm{SNP}_i^P - 1\right] + \frac{\mathrm{p}_1^i(X)}{(1-0)} \left[\mathrm{SNP}_i^P - 0\right] \right\}. \tag{1}$$

The computation in (1) can easily be realized using the encrypted SNPs of the patient (and utilizing the homomorphic properties of the Paillier cryptosystem) to compute the encrypted disease susceptibility, $\mathrm{E}(\mathbb{S}_P^X, g^x)$. Then, the SPU partially decrypts the end-result $\mathrm{E}(\mathbb{S}_P^X, g^x)$ using its share ($x^{(1)}$) of P's secret key (x) to obtain $\mathrm{E}(\mathbb{S}_P^X, g^{x^{(2)}})$ and sends it to the MC. Finally, the MC decrypts $\mathrm{E}(\mathbb{S}_P^X, g^{x^{(2)}})$ using its share ($x^{(2)}$) of P's secret key to recover the end-result $\mathbb{S}_P^X$.

III. Implementation and Complexity Evaluation

We implemented the proposed solution, and assessed its storage requirement and computational complexity on Intel Core i7-2620M CPU with 2.70 GHz processor. We set the size of the security parameter (n in Paillier cryptosystem) to 1024 bits. We computed the disease susceptibility using weighted averaging and real SNP profiles from [14]. We used the Java programming language along with the open-source Integrated Development Environment, NetBeans IDE 7.1.1., for the implementation of the Java code.

We observed that (i) Paillier encryption takes 30 ms. per variant at the CI, (ii) proxy re-encryption takes 2 ms. at the SPU, (iii) homomorphic operations takes 10 sec. at the SPU (using 10 variants), and (iv) decryption of the end-result (or relevant SNPs) takes 26 ms. at the MC. Moreover, storage of the SNPs at the SPU needs 5 GB of storage per patient. In summary, all these numbers show the practicality of our privacy-preserving algorithm.

IV. Conclusion

In this paper, we have introduced a privacy-preserving scheme for the utilization of genomic data in medical tests and personalized medicine methods. We have proposed a new model based on the existence of a Storage and Processing Unit (SPU) between the patient and the medical unit. We have shown that encrypted genomic data of the patients can be stored at the SPU and processed (for medical tests and personalized medicine methods) using homomorphic encryption and proxy re-encryption. We also implemented the proposed scheme and showed its efficiency and practicality. We are confident that our proposed privacy-preserving scheme will encourage the use of genomic data, by the individual and by the medical unit, and accelerate the move of genomics into clinical practice.

References

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

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

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

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

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

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

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

[8] http://lca.epfl.ch/projects/genomic-privacy/.

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

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

[11] http://www.ncbi.nlm.nih.gov/projects/SNP/, Visited on 29/Oct/2012.

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

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

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