Dynamic Attribute-Based Privacy-Preserving Genomic Susceptibility Testing

Mina Namazi
Signal Theory and Communications Dept., University of Vigo, Spain
mnamazi@gts.uvigo.es

Cihan Eryonucu
Computer Engineering Dept., Bilkent University
Ankara, Turkey
cihan.eryonucu@bilkent.edu.tr

Erman Ayday
Computer Engineering Dept., Bilkent University
Ankara, Turkey
Electrical Engineering and Computer Science Dept., Case Western Reserve University
Cleveland, Ohio, USA
exa208@case.edu

Fernando Pérez-González
Signal Theory and Communications Dept., University of Vigo, Spain
fperez@gts.uvigo.es

ABSTRACT

Developments in the field of genomic studies have resulted in the current high availability of genomic data which, in turn, raises significant privacy concerns. As DNA information is unique and correlated among family members, it cannot be regarded *just* as a matter of individual privacy concern. Due to the need for privacy-enhancing methods to protect these sensitive pieces of information, cryptographic solutions are deployed and enabled scientists to work on encrypted genomic data. In this paper, we develop an attribute-based privacy-preserving susceptibility testing method in which genomic data of patients is outsourced to an untrustworthy platform. We determine the challenges for the computations required to process the outsourced data and access control simultaneously within patient-doctor interactions. We obtain a non-interactive scheme regarding the contribution of the patient which improves the safety of the user data. Moreover, we exceed the computation performance of the susceptibility testing over the encrypted genomic data while we manage attributes and embedded access policies. Also, we guarantee to protect the privacy of individuals in our proposed scheme.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols**;

KEYWORDS

Genomic Privacy, Privacy-Preserving Genomic Testing, Lattice-Based Cryptography, Attribute-Based Homomorphic Encryption.

ACM Reference Format:
Mina Namazi, Cihan Eryonucu, Erman Ayday, and Fernando Pérez-González. 2019. Dynamic Attribute-Based Privacy-Preserving Genomic Susceptibility Testing. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297280.3297428

1 INTRODUCTION

Although research on genomic data improves medical diagnoses and predicts disease risk, information leakage through data process and storage may compromise patients' privacy. Emerging new companies, which offer to run genomic tests such as ancestry or paternity test, causes a sharp decrease in the cost of DNA sequencing and raises the availability of the privacy-sensitive genomic information. Because genomic data carries information about individuals' unique identities and their relatives, they are vulnerable to being abused.

Local memories such as mobile devices or computers may not have sufficient space to store all these massive amounts of information. Furthermore, if people are not cautious about their security and their devices are hacked or stolen, their abandoned privacy cannot be regained. Also, in case of emergency, the patient's medical reports should be accessible, hence storing genomic data in various medical centers is not desired. All in all, both due to practical and safety reasons, it is advantageous to store the privacy-sensitive genomic information of individuals in a centralized server.

While the same server, which has sufficient memory and power, is responsible for carrying out the genomic tests, the patient wishes to control the way of utilizing her personal information by employees of various health centers. Therefore, adequate architectures should be designed to store and examine the genomic data of the patients and monitor the ways of accessing the patients' medical records. However, the challenging problem arises when the protection of the patients' privacy in the execution of analysis on their genomic data simultaneously deals with the management of accesses over these data.

Example : In an exemplary setting, there are family doctors and pharmacist at Saint Mary Hospital, a cardiovascular specialist and

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297428

lab researchers at Cleveland Clinic, a brain specialist and nurses at American Hospital, insurance company staff at John Hopkins Institute. These people have their own attribute sets describing their role, specialty, and region. The family doctor is suspicious about some damage in the brain of the patient and wants to run some analyses on her genomic data. The patient sends her biological sample to a trusted authority and defines an access structure (denoted by predicates) over some set of attributes. The trusted authority sequences the patient's biological sample and encrypts their locations within this access structure. The trusted party stores this encrypted DNA with the embedded policy of the patient's choice which suitably describes who can decrypt and obtain the test result regarding job description, specifically, and location. For example, the patient defines her policy to accept "**all**" attributes of {Job U, Specialty Y, Medical Unit I} set. The trusted party issues a secret key for the attributes and distributes them among the other users of the protocol. Upon the patient's request, the server runs the test, and the ultimate result associated with this patient's policy can be decrypted by the parties whose attributes satisfy this access structure. If the genomic data of this patient is encrypted and stored with associated policy {U := Doctor AND Y := Brain AND I := American}, then a brain specialist doctor in American Hospital (Dr. Jill Parsons) whose attributes satisfy this patient's policy, (i.e., Dr. Jill Parsons has this access policy's corresponding decryption key), can decrypt and recover the final result of the test. Cardiovascular specialist, nurses, lab researchers, and insurance company staff cannot decrypt and access the reported result. Fig. 1 illustrates the mapping between the parties and their attributes.

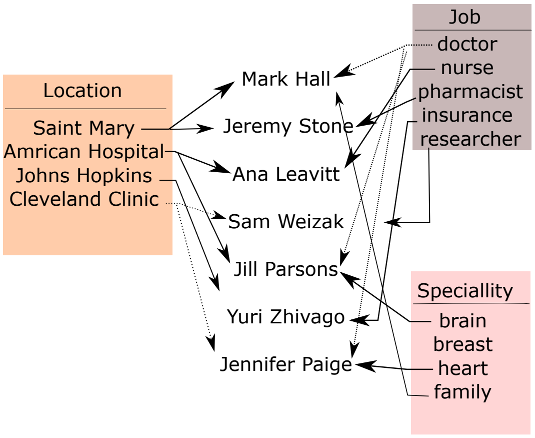

Figure 1: Relation between participants and attributes.

Unfortunately, the existing genomic privacy-preserving methods in the literature are mainly able to perform only one of the following tasks at a time: 1) process the information for storage or running analyses over the encrypted data with a unique medical unit such as [3, 10, 11]; or 2) protect medical records and regulate the access controls by deploying cryptographic methods [1, 13]. Since the defined policies are not dynamic, subsequent refined calculations cannot be supervised automatically by the system.

We develop a privacy-preserving genomic susceptibility testing method by leveraging an attribute-based homomorphic cryptosystem based on mathematical hardness problems over lattices. Our work relies on a genomic privacy-preserving scheme for susceptibility testing developed by Namazi *et al.* [10] concerning *only one* medical unit for medical tests. We enhance this scheme to manage accesses of *more than one* medical unit through attributes and predicates embedded in the cryptosystem while working on genomic data of a patient. Our goal is to calculate the genetic susceptibility test function for a given disease by outsourcing the computation to a processing unit, which should not have access to the patient's sensitive data or the confidential parameters of the analysis. In comparison with the existing methods which also confidentially execute susceptibility test, our proposed scheme has the following **contributions**:

- The proposed scheme can homomorphically run the whole function with both patient data and susceptibility parameters encrypted over the same set of predicates and simultaneously control the accesses of *various* parties to the genomic information of the patients.
- The proposed scheme is non-interactive by keeping the patient out of the protocol after defining the access policies to her data which increases the safety of the user data. The patient does not require to be online after releasing her biological sample for sequencing.
- The proposed scheme leads to releasing the data only to a set of authorized medical units whose attributes satisfy the defined predicate.
- Dynamic access control is obtained to eliminate re-initializing the protocol from the scratch while new members include to the system.
- Although adding the attributes and predicates increases the cost of the interactions, the proposed scheme is practical and highly efficient in comparison to the non-automatic protocols.

1.1 Organization

The rest of the paper is organized as follows: the related works are surveyed in Section 2; the building blocks and the core cryptosystem are presented in Section 3. We present our proposed scheme in Section 4 and discuss its implementation in Section 5. After a brief discussion about different aspects of our proposal in Section 6, final remarks are given in Section 7.

2 RELATED WORK

Pirretti *et al.* [13] investigated the use of cryptographic primitives to control the accesses to genomic data. Their proposal applies to distributed systems and social networks; it is built over bilinear assumptions, and securely manages information access in distributed systems. The scheme solely provides access control where it does not allow to perform operations over encrypted data.

In a similar work, Akinyele *et al.* [1] implement a self-protecting technique for medical records on mobile devices using an attribute-based encryption scheme, in which, as opposed to our method, the server operates on unencrypted genomic data.

Table 1: Used notations.

General Notation	
Calligraphic $\mathcal{P}$	Set of participants in the protocols
Upper vector $\vec{a}$	Attribute and predicate vectors
Boldface capital $\mathbf{C}$	Matrices
$a \cdot b$	Elementwise multiplication
$X_{E_{\vec{h},k}}$	Encrypted X under predicate $\vec{h}$ and key k

Susceptibility Testing Notation	
$\Gamma^{\mathcal{P}}$	Set of positions of real SNPs of patient $\mathcal{P}$
$\gamma^{\mathcal{P}}$	Set of positions of potential SNPs of patient $\mathcal{P}$
$\mathrm{SNP}^{\mathcal{P},i}$	i-th SNP for patient $\mathcal{P}$. $\mathrm{SNP}^{\mathcal{P},i}$ equals 0 when it belongs to $\gamma^{\mathcal{P}}$, and 1 when the patient presents a variant (it belongs to $\Gamma^{\mathcal{P}}$)
Ω_x	Set of positions of SNPs which are related to disease x.
$pr_b^{x,i}$	Probability of developing disease x conditioned on the value of the i-th SNP, with $b \in 0, 1$
$c^{x,i}$	Contribution (likelihood) of the i-th SNP to the susceptibility to disease x
$S^{\mathcal{P},x}$	Predicted susceptibility of patient $\mathcal{P}$ to disease x

In a protocol proposed by Naveed *et al.* [11], a patient outsources her genomic data in an encrypted format with attached policy parameters. The medical units request a new key to calculate the required test function from a central authority. Based on the patient's policy, the latter decides on granting a secret key for the needed test function. In contrast, in our scheme, the user once defines an authorized set of policies. After obtaining the decryption key corresponding to the attributes, the policy is automatically applied, and there is no need for a central authority or any online interaction with the patient, hence our method is more practical.

Ayday *et al.* proposed several methods [3] to run a test to quantify the susceptibility of a patient for a particular disease. They applied a secret sharing method and proxy re-encryption to assist the parties which partially encrypt and decrypt the final results using the Paillier [12] encryption scheme. Later on, another privacy-preserving method was proposed in [10] which calculates the susceptibility testing based on a homomorphic encryption scheme over lattices. In this scheme, the trusted party gets the biological sample from a patient, generates the public parameters and keys, and distributes them among the parties. It also sequences the DNA sample of the patient, builds a data structure and sends the encrypted form of the corresponding information to the server. Medical unit marks the required locations for the test, and the patients look up inside the data structure to confirm whether their DNA carries these values. The server homomorphically runs the test by leveraging a key-switching technique to modify the encrypted data under the patient's key to be decryptable by the medical center's key to enable the homomorphic operations. The final result is released encrypted to the corresponding medical center which decrypts the test and informs the patient accordingly. Our proposal extends this architecture to efficiently control multiple medical units' accesses to the patient's genomic data while outsources and operates over this information.

3 BUILDING BLOCKS

In this section, we briefly describe the necessary background and assumptions for constructing our method of attribute-based genomic privacy-preserving testing. A summary of the notations is given in Table 1.

3.1 Genomic Background

Among several types of DNA variants in the human genome, "*single nucleotide polymorphisms*" (SNPs) are the most common ones. In a single DNA block which is denoted by nucleotide, each SNP represents a difference. Normally, through someone's DNA, on average, SNP occurs in every 300 nucleotides, i.e., there are approximately 50 million SNPs in the human genome as of now[1]. SNPs play the role of biological markers to assist the scientists to locate the genes associated with a particular disease. Usually, in each SNP position, there are two nucleotides (alleles), major and minor allele. Inherited alleles/variants from each parent can be identical (homozygous) or different (heterozygous). *Reference human genome* which is a digital sequence of nucleotides represents the human genetic makeup and helps to identify the human genetic variants. A genetic variant can take two different alleles: one from the reference genome and one from the alternative version occurring in the human population. At the content of a given SNP position, an individual can take at least one alternative allele or not have a variant. Following the proposal in [3] to measure susceptibility via "weighted averaging" [8], we refer to the set of these SNPs which take at least one alternative alleles for a patient $\mathcal{P}$ as "real SNPs" and the remaining ones where the approved SNPs do not exist for the considered patient as "potential SNPs". The i-th SNP for patient $\mathcal{P}$ is represented as $\mathrm{SNP}^{\mathcal{P},i}$, where $\mathrm{SNP}^{\mathcal{P},i} = 1$ implies a real SNP (i.e., a variant), and $\mathrm{SNP}^{\mathcal{P},i} = 0$ a potential SNP (i.e., non-variant). $\Gamma^{\mathcal{P}}$ denotes the set of positions for real SNPs of patient $\mathcal{P}$ (at which $\mathrm{SNP}^{\mathcal{P},i} = 1$), and $\gamma^{\mathcal{P}}$ the set of positions of potential SNPs, at which $\mathrm{SNP}^{\mathcal{P},i} = 0$.

[1] https://ghr.nlm.nih.gov/primer/genomicresearch/snp

3.2 Cryptographic Primitives

We describe the core cryptographic schemes and their hardness assumptions to construct an attribute-based privacy-preserving genomic susceptibility testing method as follows:

3.2.1 Learning With Errors (LWE) Problem [14]. For a security parameter λ, let $n = n(\lambda)$ be an integer dimension, $q = q(\lambda) \geqslant 2$ be an integer, and $\chi = \chi(\lambda)$ be an error distribution over $\mathbb{Z}$. For the secret $s \leftarrow \mathbb{Z}_q^n$, the LWE distribution $A_{n,q,\chi}$ over $\mathbb{Z}_q^{n+1}$ is sampled by $a \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(a, b = s \cdot a + e \bmod q)$. For m independent samples $(a_i, b_i) \in \mathbb{Z}_q^{n+1}$ uniformly, the decision problem of LWE implies distinguishing between the $A_{s,q,\chi}$ distribution and the uniform distribution over $\mathbb{Z}_q^{n+1}$.

3.2.2 Attribute-Based Homomorphic Encryption Scheme (aBHE) [5]. Informally, in a ciphertext policy attribute-based encryption scheme, an encryptor Alice describes a policy (predicates) while encrypting her data, and a trusted party issues a decryption key for the attributes and distributes them among parties. A decryptor Bob can decrypt this ciphertext if his attributes satisfy Alice's defined policy.

Clear *et al.* introduced a homomorphic attribute-based encryption scheme which evaluates bounded depth circuits [5]. It is constructed based on the Learning with Errors (LWE) assumption and only restricts the number of inputs to the evaluation circuit. This scheme combines the levelled attribute-based homomorphic encryption (lAB) of Gentry *et al.* [7] with the algorithms of {lABSetUp, lABKeyGen, lABEnc, lABDec, lABEval} and the multi-key homomorphic encryption (mKH) consisting of {mKHSetUp, mKHKey Gen, mKHEnc, mKHDec, mKHEval} scheme of Clear *et al.* [6]. The aBHE scheme contains five algorithms: {aBHSetup, aBHKeyGen, aBHEnc, aBHDec, aBHEval}, with respect to a message space $\mathbb{M} = \{0, 1\}^w$, where w can be arbitrarily large input to the circuit C, but bounded by N, an attribute space $\vec{\mathbb{A}}$, a class of access policies $\vec{\mathbb{H}} \subseteq \vec{\mathbb{A}} \to \{0, 1\}$, and a class of circuits $\mathbb{C} \subseteq \mathbb{M}^* \to \mathbb{M}$. The parameter $K \in [D]$ (where $i \in [x] \doteq \{1, \ldots, x\}$) specifies the maximum number of keys that can be passed to the decryption algorithm. The access structure is collaborative model, i.e, if the evaluation is performed over the ciphertexts which are encrypted by different users, a decryptor whose all his d attributes satisfy the predicate can still decrypt on his own. We describe the definitions of the aBHE as follows :

$\mathsf{aBHSetUp}(1^\lambda, 1^N)$: The set up algorithm receives security parameter λ and maximum number of decryption keys N as input, then :

- Chooses an integer w.
- Uses a polynomial $g(.,.)$ to give the number of inputs to the decryption circuit for N keys and security parameter λ. Let $L = g(\lambda, N)$:
- Calls $\mathsf{lABSetup}(1^\lambda, 1^L) \to (PP_{lAB}, mpk, msk)$.
- Outputs $PP := (PP_{lAB}, \lambda, N, w)$ and (mpk, msk). The public parameters PP are inputs of the entire algorithms of this scheme.

$\mathsf{aBHEnc}(mpk, \vec{h}, \mu)$: Inputs of the encryption algorithm are master public key of the lAB scheme mpk, a binary message space $\mu = (\mu_1, ..., \mu_w) \in \{0, 1\}^w$, and a policy $\vec{h} \in \vec{\mathbb{H}}$. Ciphertexts are associated with a set of policies $\vec{h}_i \in \vec{\mathbb{H}}$. Each encryptor performs the following operations :

- Calls key generation algorithm of multi-key homomoprhic encryption $\mathsf{mKHKeyGen}(1^\lambda, 1^L) \to (sk, pk, vk)$.
- Calls $\mathsf{lABEnc}(mpk, \vec{h}, sk) \to \psi$.
- Calls $\mathsf{mKHEnc}(pk, mu_i) \to c_i$, for $i \in [w]$.
- The ciphertext $CT = (\text{type} := 0, \text{enc} := (\psi, vk, (c_1, ..., c_w)))$. Type can be 0 or 1 which refers to a fresh ciphertext or result of an evaluated ciphertext, respectively.

$\mathsf{aBHKeyGen}(msk, \vec{a})$: On the inputs of the master secret key msk, and attributes $\vec{a} \in \vec{\mathbb{A}}$, the algorithm generates a secret key for vector $\vec{a}$:

- Calls $\mathsf{lABKeyGen}(msk, \vec{a}) \to sk_{\vec{a}}$ and issues it to the user.

$\mathsf{aBHEval}(mpk, C, CT_1, ..., CT_l)$: On the inputs of a circuit $C \in \mathbb{C}$ and $l \in [N]$, for fresh (Type= 0) ciphertexts, the evaluator performs the following operations if $\vec{\mathbb{H}}(\vec{a}_i) = 1$ for $i \in [d]$:

- Parses fresh ciphertex CT_i as $(\text{type} := 0, \text{enc} := (\psi_i, vk_i,$ $(c_1^{(i)}, ..., c_w^{(i)}))$ for every $i \in [l]$. The predicate h_i is associated with ψ_i.
- The evaluator calls $\mathsf{mKHEval}(C, (c_1^{(1)}, vk_1), \ldots, (c_w^{(1)}, vk_1)$ $, \ldots, (c_1^{(l)}, vk_l), \ldots, (c_w^{(l)}, vk_l)) \to c'$.
- Encrypts c' under predicate $\vec{h}_i$ by calling $\mathsf{lABEnc}(\vec{h}_i, c') \to \psi_{c'}$.
- Using decryption circuit of mKH, $D_{<\lambda,N>}$, calls $\mathsf{lABEval}(D_{<\lambda,N>}, \psi_{c'}, \psi_1, \ldots, \psi_l) \to \psi$.
- Returns the evaluated ciphertext $CT' = (\text{type} := 1, \text{enc} := \psi)$.

$\mathsf{aBHDec}(sk_{\vec{a}_i}, CT)$: Decryption is possible when the attribute set $\vec{a}_i$ is authorized in the access structure $\vec{\mathbb{H}}$, i.e., $\vec{a}_i \in \vec{\mathbb{A}}$ for $i \in [d]$. To decrypt a ciphertext CT = (type, enc) with the secret key of the attributes $sk_{\vec{a}_i}$, and for the associated predicates $\vec{h} \in \vec{\mathbb{H}}$, a decryptor performs :

- For a fresh ciphertext (type= 0), enc is $(\psi, vk, (c_1, \ldots, c_w))$. The decryptor calls $\mathsf{lABDec}(sk_{\vec{a}_i}, \psi) \to sk$. if $sk = \perp$ aborts. Otherwise :
- Calls $\mathsf{mKHDec}(sk, c_j)$ for every $j \in [w]$, and outputs $\mu := (\mu_1, \ldots, \mu_w) \in \{0, 1\}^w$.
- For an evaluated ciphertext (type= 1), enc is parsed as ψ. The decryptor calls $\mathsf{lABDec}(sk_{\vec{a}_i}, \psi) \to x'$. If $x' = \perp$ aborts. Otherwise the plaintext is $\mu := x' = \{0, 1\}^w$.

Semantic security of this aBHE [5] is the same as lAB [7] semantic security with extra access of the adversary to $\mathsf{aBHEval}$ algorithm.

3.2.3 Homomorphic Operations. In this section, we briefly describe how the homomorphic operations perform over the encrypted data. The evaluation function of Section 3.2.2 contains two parts, first it calls the evaluation function of the multi-key homomorphic encryption scheme and later on it evaluates using the decryption circuit of the levelled homomorphic attribute-based encryption scheme. A ciphertext $\mathbf{C}$ is a $M \times M$ matrix over $\mathbb{Z}_q$ that encrypts μ under the M-dimensional vector v as a secret key if $\mathbf{C} \cdot v = \mu \cdot v + e \in \mathbb{Z}_q^N$, where e is small noise vector. Let $\mathbf{C}_1$ and $\mathbf{C}_2$

be the encryptions of μ_1 and μ_2 respectively, then homomorphic addition is defined as follows [7] :

$$\begin{aligned} \mathbf{C}^{+} &:= \mathbf{C}_1 + \mathbf{C}_2 \\ \mathbf{C}^{+} \cdot v &= (\mu_1 + \mu_2) \cdot v + (e_1 + e_2) \end{aligned}$$

And homomorphis multiplication according to [7] is described as as follows :

$$\begin{aligned} \mathbf{C}^{\times} &:= \mathbf{C}_1 \cdot \mathbf{C}_2 \\ \mathbf{C}^{\times} \cdot v &= \mathbf{C}_1 \cdot (\mu_2 \cdot v + e_2) \\ &= \mu_2 \cdot (\mu_1 \cdot v + e_1) \cdot \mathbf{C}_1 \cdot e_2 \\ &= \mu_1 \cdot \mu_2 \cdot v + \mu_2 \cdot e_1 + \mathbf{C}_1 \cdot e_2 \\ &= \mu_1 \cdot \mu_2 \cdot v + \text{"small"}. \end{aligned}$$

Clear *et al.* [6] extended these operations to perform over the ciphertexts which are encrypted by multiple parties under different keys. Therefore, suppose $\mathbf{C}_1$ and $\mathbf{C}_2$ be the encryptions of μ_1 and μ_2 under the secret keys of v_1 and v_2 respectively. Clear *et al.* proposed a transformation for $\mathbf{C}_1$ and $\mathbf{C}_2$, which both input to the same circuit and produce $\mathbf{C}'$, where $C \in \mathbb{C}$ be the circuit. This $2M \times 2M$ ciphertext matrix $\mathbf{C}'$ encrypts $\mu' = C(\mu_1 + \mu_2)$ under the concatenations of v_1 and v_2 as the secret key. The details of this transormation is out of the scope of this paper and we refer the readers to the original paper [6] for further information.

4 PROPOSED SCHEME

We develop an efficient solution to operate on outsourced genomic data of individuals while the data owners can control the accesses to different parts of their sequenced genome. Below, we explain the interactions of the involved parties and the threat model of our protocol.

4.1 Protocol Setting

Throughout the paper, we use the same notation for the involved parties as the prior work of Ayday *et al.* [3] : a patient $\mathcal{P}$ who owns the genomic data; a trusted certified institute $\mathcal{CI}$; a storage and processing unit $\mathcal{SPU}$, and different individuals with their specialization from different regions inside medical units which for simplicity we denote them by $\mathcal{MU}_1, \ldots, \mathcal{MU}_n$, where n is the maximum number of involved medical unit in the protocol. We allow each medical units to have attributes describing its job, specialty, and location. The patient is the one who (1) defines the policies restricting the accesses of the medical units to the result of the information according to their attributes, and (2) enforces releasing the data to only the parties whose attributes meet these policies.

4.2 Threat Model

All the parties are assumed to be semi-honest, i.e., they follow the protocols and they are not allowed to modify their inputs to obtain unauthorized information. However, there might be curious parties inside the medical units or $\mathcal{SPU}$ who are willing to obtain more information from the transactions they can observe. The $\mathcal{CI}$ is a trusted party that sequences, encrypts, generates, and distributes keys between the parties. The security of our proposed scheme is based on one-wayness and semantic security of the underlying attribute-based homomorphic encryption schemes [5]. We assume that the parties do not collude or share their secret key of various attributes with each other.

4.3 Protocol Overview

Patient $\mathcal{P}$ sends her biological sample to the certified institute $\mathcal{CI}$ for sequencing. She describes the medical units that she intends to consult after the test is accomplished (for further research, remedies, or specialized treatment). For this purpose, she decides and embeds the access structure, AS, which is a boolean formula referring to the attributes of the users who can access different parts of her genomic data. The $\mathcal{CI}$ sequences the patient's DNA runs the setup and key generation algorithms for the aBHE to generate public parameters *PP*, the master secret and public key (*msk*, *mpk*) and distributes *PP* and *mpk* among the participants. The $\mathcal{CI}$ uses *msk* to generate decryption keys for the attributes $sk_{\vec{a}}$ according to the defined access structure AS. Furthermore, the $\mathcal{CI}$ encrypts each SNP position with the master public key *mpk* with an embedded predicate and sends them to the $\mathcal{SPU}$. The medical unit sends the parameters of the test encrypted under the master public key *mpk* within the same access structure to the $\mathcal{SPU}$. The $\mathcal{SPU}$ chooses the particular locations of the encrypted SNPs which are relevant to the required test and performs the test illustrated in (1) by attribute-based homomorphic evaluation of the test function for this access structure. Those medical units whose attributes satisfy the defined policy and owns the decryption key of the attributes can decrypt and obtain the test result.

We choose the weighted average method to calculate the susceptibility test by generalizing the observations made in [2, 3]. The susceptibility to disease x using weighted averaging is as follows :

$$S^{P,x} = \frac{\sum_{i \in \Omega_x} c^{x,i} \{pr_0^{x,i}[1 - \text{SNP}^{P,i}] + pr_1^{x,i}\text{SNP}^{P,i}\}}{\sum_{i \in \Omega_x} c^{x,i}}. \quad (1)$$

4.4 Protocol

We illustrate our proposed scheme in Fig. 2, and describe the interaction between the parties as follows :

Set up and key generation :

Step s1 : SetUp : The $\mathcal{CI}$ runs aBHSetup to get (*PP*, *mpk*, *msk*). *PP* is part of the input in the following algorithms.

Step s2 : KeyGen$(msk, \vec{a}) \rightarrow k$: The $\mathcal{CI}$ runs aBHKeyGen to obtain $sk_{\vec{a}}$, and outputs $k := sk_{\vec{a}}$.

Sequencing and generation of input encryption :

Step e1 : The patient $\mathcal{P}$ decides on the predicates. She sends her access structure AS which defines the relations over the attributes (we refer to the example 1 in the Section 1). Moreover, $\mathcal{P}$ sends her biological sample for sequencing to the $\mathcal{CI}$.

Step e2 : The $\mathcal{CI}$ sequences the sample and encrypts each bit of SNP positions with the master public key *mpk* and embeds the predicates over the set of attributes for the relevant tests. Also, the $\mathcal{CI}$ sends these encrypted positions and SNPs to the $\mathcal{SPU}$.

Encrypted susceptibility test :

Step t1 : The $\mathcal{MU}$ sends the required test parameters of these SNPs for disease x encrypted under master public key *mpk* and the same predicate as the patient's authorized to the $\mathcal{SPU}$ equals to $\{pr^{x,i}_{E_{\vec{h},mpk}}, c^{x,i}_{E_{\vec{h},mpk}}\}_{i \in \Omega_x}$, along with required locations corresponding to this test.

Step t2 : The $\mathcal{SPU}$ runs the susceptibility test in (1) on patient $\mathcal{P}$'s encrypted SNPs and $\mathcal{MU}$'s encrypted susceptibility parameters

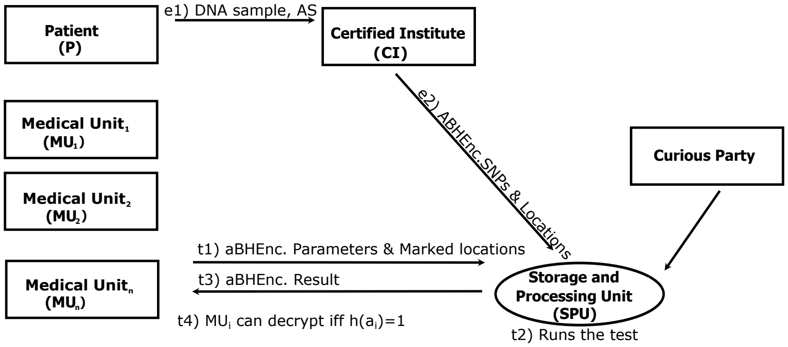

Figure 2: Proposed attribute-based genomic privacy-preserving scheme.

for x for the same set of predicates using the relevant SNPs for this particular test by running the aBHEval algorithm which leverages the operations defined in Section 3.2.3, and obtains the encrypted value of $S^{P,x}_{E_{\vec{h},mpk}}$ for the specified set of predicates.

Step t3 : The $\mathcal{SPU}$ releases the encrypted result to $\mathcal{MU}_i$.

Step t4 : $\mathcal{MU}_j$, where j belongs to the set of authorized attributes in the aBHE scheme, decrypts using its own $sk_{\vec{a}}$ to obtain the clear-text test result $S^{P,x}$ of patient $\mathcal{P}$ for disease x.

5 EVALUATION

The evaluation of the main attribute-based homomorphic encryption scheme [5] which is used by the participants of our proposed scheme in Section 4 to encrypt, evaluate and decrypt is enhanced by implementing the two underlying encryption schemes: 1) a multi-key homomorphic encryption scheme (mKH), and 2) a leveled attribute-based encryption scheme (lAB). To implement the mKH scheme, we substitute the mKH scheme of Clear *et al.* with a simpler multi-key homomorphic encryption scheme of Mukherjee *et al.* [9]. We expand the TFHE library [2] to manage different ciphertexts which are encrypted by multiple keys. We use the same TFHE library to control the access structures and attributes for lAB scheme. We implement and examine our program using C++. The test environment is a Mac OSX operating system with Intel Core i5 processor, and the key size has 1024−bit length.

We evaluate the operation costs regarding the effort of 1) the $\mathcal{MU}$ in encrypting the two related test parameters, 2) the $\mathcal{CI}$ in the encryption of the patient's variants, and 3) the $\mathcal{SPU}$ to calculate Eq. (1) via the attribute-based homomorphic operations. In Table 2, we summarize the achieved running times of the participants of our protocol in the presence of 1 and 3 medical units.

We emphasize that $\mathcal{CI}$ encrypts the SNPs once, but each medical unit encrypts 2 various parameters of the related test. Therefore, it is logical that the effort of each medical unit dominates the effort of the certified institute. Furthermore, as the number of medical units increases in the protocol, the $\mathcal{SPU}$ requires more effort to evaluate the test function.

We compare the runtime and storage cost of each involved party in our proposed scheme with just one medical unit in the system while the participants leverage aBHE scheme [5] with the existing privacy-preserving protocol of Namazi. *et al.* [10] which allows storage and processing on genomic data via a homomorphic encryption scheme over lattices denoted by BGV [4] for only one medical unit without access policies. We investigate how adding access policies affects the operations in return for gaining more safety for the genomic data of the patients. Finally, we briefly compare the implementation cost of our proposed scheme with that of Ayday *et al.* [3] which applies the Paillier scheme [12] and represent the results in Table 3.

Since our proposed scheme is non-interactive, the patient $\mathcal{P}$ spends no effort after releasing her biological sample. In [10] the patient's effort to encrypt the variants with BGV scheme takes 1.8 ms. In [3], the patient performs the decryption of the Paillier encryption scheme in 26 ms.

Switching to homomorphic encryption over lattices (via BGV encryption scheme) significantly increases the efficiency of the evaluation in running the test function of Eq. (1) at the $\mathcal{SPU}$ side to be 6 ms in [10], and adding attributes slightly increases the running time to 11 ms. The $\mathcal{MU}$ performs the evaluation operation with Paillier encryption in the protocol of [3] with the cost of 1 sec. In our scheme, the $\mathcal{MU}$s require around 319 ms to encrypt the two parameters related to the test with the embedded access structure, while this effort takes 3.5 ms in [10] without the contribution of the access structures. Besides, the $\mathcal{MU}$s are responsible for decrypting the final result which lasts 0.7 ms via BGV scheme in [10], and 1.5 ms in our scheme via aBHE scheme.

The certified institute $\mathcal{CI}$ encrypts each patient's SNPs in a one-time operation in [3] with Paillier in 30 ms while this time falls to 1.8 ms by BGV scheme in [10]. Adding access structures decreases the encryption speed to 210 ms.

Storage cost at the $\mathcal{SPU}$ in [3] which is 500 MB sharply shrinks to be approximately 30 MB in our scheme, while the protocol in [10] needs 7 MB for storage.

Since our proposed scheme manages access policies and evaluates over data that are encrypted by different parties, its evaluation running time and storage cost in the $\mathcal{SPU}$ side is slightly higher than [10] which deals with just one medical unit with no embedded access policies.

[2] https ://github.com/tfhe/tfhe/ library

Table 2: Running time of participants of our proposed scheme (Sec. 4) with 1 and 3 $\mathcal{MU}$s.

	Medical Units	Certified Institute	Storing and Processing Unit
Our scheme (Sec. 4) with 1−MU	319 ms	210 ms	11 ms
Our scheme (Sec. 4) with 3−MU	319 ms	212 ms	25 ms

Table 3: Complexity comparison of our proposed scheme with [10] and [3].

@$\mathcal{P}$	@$\mathcal{MU}$s	@$\mathcal{CI}$	@$\mathcal{SPU}$
Ayday *et al.*'scheme [3]			
Paillier Dec : 26 ms	Hom. Operation : 1000 ms (per 10 variants)	Paillier Enc : 30 ms	Storage :500 MB/patient
Namazi *et al.*'scheme [10]			
BGV Enc : 1.8 ms	BGV Enc : 3.5 ms BGV Dec : 0.7 ms	BGV Enc :1.8 ms	Hom. Operation : 6 ms Storage : 7 MB/patient
Our proposed scheme (Sec. 4)			
	aBHE Enc : 319 ms aBHE Dec : 1.5	aBHE Enc : 210 ms	Hom. Operation : 11 ms Storage : 31 MB/patient

6 DISCUSSION

We guarantee the security of our protocol by running all the interactions in an encrypted format. The $\mathcal{SPU}$ cannot observe the genomic data of the patient and the clear-text of the final result. During the interactions, the $\mathcal{SPU}$ receives encrypted SNPs of the patient from the $\mathcal{CI}$ which are partially encrypted under her public key and the master public key of the protocol. With the same argument, the $\mathcal{SPU}$ does not have access to the test parameters provided by each $\mathcal{MU}$s. The test is a homomorphic evaluation over these confidential data where the $\mathcal{SPU}$ has no decryption key to observe the clear-text of the final result which provides more security in our protocol. This decryption key which is issued by the certified institute to the attributes is the only way of accessing the final results if the attributes of a medical unit satisfy the defined policies. Also, the collision of the medical units with each other or with the $\mathcal{SPU}$ is not allowed. Hence, unauthorized medical units whose attributes do not satisfy the defined policies have no chance to decrypt and observe the final test results.

Each patient can define as many access policies as it is required and explain how she authorizes each participant to access different parts of her genomic data based on their attributes. However, the number of decryption keys should remain below N, and the carnality of the set of attributes $|\{a_1, \ldots, a_d\}|$ should always remain below D ($d \in [D]$). Otherwise, the evaluation fails and decryption does not return the correct message.

This scheme can be extended to multiple patients. Then different patients define various access structures, and with a universal master secret and public key, the $\mathcal{CI}$ generates keys for each attribute. In this case, a doctor U with specialty Y in medical unit I can access the medical records of all the patients if his attributes satisfy all the defined policies. Assigning a master secret and public key to each patient to define a unique access structure for each patient leads to obtaining a system where the authorized parties can recover the genomic data of this particular patient — a protocol that supports this setting in on progress.

Our protocol enables various medical units to contribute to a genomic test by sending their test parameters in an encrypted format to the $\mathcal{SPU}$. In our protocol's core encryption scheme multi-key homomorphic encryption is deployed which enables the evaluation of data sets encrypted with multiple senders. However, solely possessing the decryption key of the attributes which satisfy the defined policy is sufficient to decrypt and recover final data.

Regarding granting access to a newly joined medical unit, it is sufficient to generate a key corresponding to his attributes. In a case that this medical unit asks for a test query, he should encrypt his data under the master public key and embeds the policy in his encryption. Later on, if this medical unit's attributes satisfy the defined policy, he can decrypt and recover the corresponding result. No further attempt is necessary to encrypt the stored data or to initialize the protocol from scratch. Revoking the accesses of the parties who leave the system is not a straightforward task. Hence, it is obligatory to implement the system with a list of revoked access structures and check the list before issuing a decryption key to a party.

7 CONCLUSION

We achieved an efficient and practical privacy-preserving susceptibility testing method that describes the way of storing and processing patients' genomic data delegated to an untrustworthy server. We defined how several medical units can access the authorized parts of patients' medical records by leveraging an attribute-based homomorphic encryption scheme. Participants of the protocol enforce their policies as access structures, and the server performs the required test on encrypted genomic data without compromising individuals' privacy. Parties with the authorized attributes are the only ones who can obtain the result of such tests. We enhanced automatic and non-interactive access structures while performing the operations over genomic data at the cost of increased run time by the medical units. However, such overhead is negligible since the medical units are usually equipped with powerful computing machines, where the operations required by the tests are carried out. We further characterized the security of our proposed solution for semi-honest participants.

ACKNOWLEDGMENTS

GPSC is funded by the Agencia Estatal de Investigación (Spain) and the European Regional Development Fund (ERDF) under project WINTER (TEC2016-76409-C2-2-R), MYRADA (TEC2016-75103-C2-2-R). Also funded by the Xunta de Galicia and the European Union (European Regional Development Fund - ERDF) under projects Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019, Grupo de Referencia ED431C2017/53 and Red Temática RedTEIC 2017-2018.

REFERENCES

[1] Joseph A Akinyele, Matthew W Pagano, Matthew D Green, Christoph U Lehmann, Zachary NJ Peterson, and Aviel D Rubin. 2011. Securing electronic medical records using attribute-based encryption on mobile devices. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*. ACM, 75–86.

[2] Erman Ayday, Jean Louis Raisaro, Urs Hengartner, Adam Molyneaux, and Jean-Pierre Hubaux. 2014. Privacy-preserving processing of raw genomic data. In *Data Privacy Management and Autonomous Spontaneous Security*. Springer, 133–147.

[3] Erman Ayday, Jean Louis Raisaro, and Jean-Pierre Hubaux. 2012. *Privacy-enhancing technologies for medical tests using genomic data.* Technical Report.

[4] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2014. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)* 6, 3 (2014), 13.

[5] Michael Clear and Ciarán Mc Goldrick. 2017. Attribute-based fully homomorphic encryption with a bounded number of inputs. *International Journal of Applied Cryptography* 3, 4 (2017), 363–376.

[6] Michael Clear and Ciaran McGoldrick. 2015. Multi-identity and multi-key leveled FHE from learning with errors. In *Annual Cryptology Conference*. Springer, 630–656.

[7] Craig Gentry, Amit Sahai, and Brent Waters. 2013. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*. Springer, 75–92.

[8] Sekar Kathiresan, Olle Melander, Dragi Anevski, Candace Guiducci, Noël P Burtt, Charlotta Roos, Joel N Hirschhorn, Göran Berglund, Bo Hedblad, Leif Groop, et al. 2008. Polymorphisms associated with cholesterol and risk of cardiovascular events. *New England Journal of Medicine* 358, 12 (2008), 1240–1249.

[9] Pratyay Mukherjee and Daniel Wichs. 2016. Two round multiparty computation via multi-key FHE. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 735–763.

[10] Mina Namazi, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. 2016. Dynamic Privacy-Preserving Genomic Susceptibility Testing. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 45–50.

[11] Muhammad Naveed, Shashank Agrawal, Manoj Prabhakaran, XiaoFeng Wang, Erman Ayday, Jean-Pierre Hubaux, and Carl Gunter. 2014. Controlled functional encryption. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1280–1291.

[12] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 223–238.

[13] Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. 2010. Secure attribute-based systems. *Journal of Computer Security* 18, 5 (2010), 799–837.

[14] Oded Regev. 2009. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)* 56, 6 (2009), 34.