IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 23, 2020, accepted June 1, 2020, date of publication June 16, 2020, date of current version June 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002966

Privacy-Preserving Genome-Wide Association Study for Rare Mutations - A Secure FrameWork for Externalized Statistical Analysis

REDA BELLAFQIRA[1], (Member, IEEE), THOMAS E. LUDWIG[2], DAVID NIYITEGEKA[1], (Member, IEEE), EMMANUELLE GÉNIN[2], AND GOUENOU COATRIEUX[1], (Senior Member, IEEE)

[1]Institut Mines-Telecom, IMT Atlantique, Inserm, UMR 1101, 29238 Brest, France
[2]Univ Brest, Inserm, EFS, CHU Brest, UMR 1078, 29238 Brest, France

Corresponding author: Reda Bellafqira (reda.bellafqira@imt-atlantique.fr)

This work was supported in part by the French Government support granted to the Labex CominLabs and managed by the ANR through the "Investing for the Future" Program under Grant ANR-10-LABX-07-01, and in part by the Labex GenMed through the Project PrivGen under Grant ANR-10-LABX-0013.

ABSTRACT This paper proposes a new privacy-preserving framework to perform rare variant case-control association tests with information provided by two parties: a Genomic Research Unit (GRU) with sequencing data from individuals affected by a disease D (cases); a Genomic Research Center (GRC) with sequencing data from healthy individuals (controls). To identify genes with rare variants involved in D, GRU needs to compare cases against controls using association tests (genome-wide association study). The main originality of our proposal is twofold. First, it positions GRC as a proxy between GRU and the server. Doing so makes it possible to use classical cryptographic tools to securely conduct association tests with no computation complexity increase, contrarily to actual state of the art proposals which are of very high complexity being based on homomorphic encryption, for instance. In particular, we show how sensitive data confidentiality can be ensured with secret key based cryptographic hashing with no need to modify statistical algorithms. In our protocol the server simply conducts statistical analyses on partially hashed data. Secondly, we introduce a novel privacy constraint: GRU's identity should remain unknown to the server as this knowledge can give it clues about GRU's data (e.g., diseases and genes of interest). We exhibit how Pretty Good Privacy (PGP) can be used to solve this problem. We illustrate our protocol in the case of one rare variant association test, the Weighted-Sum Statistic (WSS) algorithm, carried out on real genetic data. This secure WSS achieves the same accuracy as its nonsecure version with no increase of complexity. Furthermore, we establish that our protocol can be extended to the different rare variant association tests available in the literature.

INDEX TERMS Data confidentiality, data outsourcing, genome-wide association study (GWAS), privacy, secure GWAS platform, weighted-sum statistic (WSS).

I. INTRODUCTION

Nowadays, genomic data are getting widely collected, stored, processed and shared for various genomic applications. Among them, case-control association studies play an important role to understand disease etiology [1]. In these studies, genetic data are compared between cases affected by the disease of interest and unaffected controls. The genetic data compared consist on common variants that are tested individually or rare variants within a gene that are considered together. These data can be obtained by genotyping SNP-chip (for the common variants) or by genome sequencing (common and rare variants are assessed). Because of the volume of genetic data to process and the number of genes to test, association tests require the use of high computation and storage capacities (e.g., cloud computing). That is particularly the case of the Weighted-Sum Statistic (WSS) algorithm [2] which objective is to decide if rare variants located within a gene are involved in disease susceptibility. Roughly, for a given gene, the test compares the burden in rare variants in case and control samples through the computation of a score for the different individuals. The power to detect an association if it exists depends on the sample size and increases with increasing sample sizes.

The associate editor coordinating the review of this manuscript and approving it for publication was Tossapon Boongoen.

Sequencing cost has considerably decreased over the last few years but still remain prohibitive on large samples of individuals as required in association studies. In order to limit the cost, control data can be shared between different research groups who work on different diseases.This was for example the solution used by the Wellcome Trust Case-Control Consortium in 2007 where they compared genome-wide data from 14,000 patients affected by 7 different diseases against 3,000 shared controls (see [3]). In this work, we are interested by a such scenario where a Genome Research Unit (GRU) that has collected sequence data on cases wants to compare them against sequence data from controls collected by a Genomic Research Center (GRC). This will require data sharing between the GRU and the GRC and this data sharing is usually done in open environments; data being exchanged through internet and often processed by a third-party (e.g., server). This obviously induces several security problems, especially in terms of privacy and data confidentiality. These aspects are reinforced based on the facts that genetic data are very unique to an individual [4] and that health-care records are very valuable for hackers. IBM recently reported that their value on the black market is as much as 60 times higher than that of stolen credit cards [5]. There is thus an interest to develop secure methods to allow collaborative association studies.

A. RELATED WORK

Securing shared or externalized genetic association studies does not simply mean securing the storage and transmission of genomic data [25], [26]. Indeed, parties involved in such studies may not want that the other parties access their data, the objective and the conclusions of the study, these ones being highly valuable assets. At the same time, the trust one can have in a cloud service provider is quite relative. Thus, it is the data analysis algorithm itself and the way it is shared between parties that have to be secured. Different methods have been proposed in order to perform privacy-preserving association studies, especially for common variants (these studies are usually referred to as Genome-Wide Association Studies, GWAS). We propose to distinguish the methods depending on the cryptographic techniques they rely on: Differential Privacy (DP), Homomorphic Encryption (HE), Secure Multiparty Computation (SMC) and Secure Hardware (SH).

Many privacy-preserving GWAS are based on differential privacy [27] due to the ineffectiveness of data anonymization techniques like k-anonymity [28], [29] or l-diversity [30] as demonstrated in [31]. Basically, DP adds a random noise to real data in order to ensure individuals' privacy. In [6], a solution allows researchers to perform exploratory analysis in a differentially private way, including the computation of: i) the number and location of the most significant SNPs to a disease, ii) the *p-values* of a statistical test between a SNP and a disease, iii) any correlation between two SNPs, and iv) the block structure of correlated SNPs. Uhler *et al.* [7] propose a differentially private release of aggregate GWAS data. They provide DP versions of the χ^2-statistic test and of the minor allele frequencies (MAFs) test. Tramèr *et al.* [8] build on the notion of Positive Membership Privacy along with a weaker adversarial model also known as relaxed DP. In the common adversarial model: the semi honest adversarial model, where entities follow a given protocol but may attempt to derive additional information about data of other entities (e.g., some individuals who participle in the study). In the weaker adversarial model, the most appropriate adversarial setting is searched for by bounding the adversary's knowledge in order to better preserve the utility of data. Simmons *et al.* [32] introduce a computational GWAS framework that adapts DP principles to protect private phenotype information (e.g., disease status), while correcting for population stratification at the same time. The authors of [9] developed a new statistic tests for private hypothesis testing. These statistics are designed specifically so that their asymptotic distributions, after accounting for the noise added for privacy concerns, match the distributions of the classical (nonprivate) χ^2 statistic test. Similar methods: RandChi and RandChiDist, have been proposed in [10]. In a more general way and as pointed out in [10], it is inherently challenging to use DP techniques for GWAS. The noise added to the original data reduces the utility of data and makes accurate statistical analysis much harder. The level of noise depends on the dataset and on the study's objective and also has to be refined when more data are added.

Homomorphic encryption (HE) is another mechanism used to protect genomic data. HE allows performing linear operations, such as additions and multiplications over encrypted data while ensuring that the decrypted results are equal to the ones carried out on clear data [33], [34]. Recently, many methods to conduct privacy-preserving computation of GWAS using homomorphic encryption have been proposed [17]–[19]. In [17], authors developed a method that allows secure computation of basic statistics which are commonly used in genetic association studies such as χ^2-statistic and Cochran-Armitage Test for Trend (CATT). However, this method is no practical due do its storage and computation complexities. Wang *et al.* [16] adopted homomorphic encryption on rare variants to perform exact logistic regression. Kim and Lauter [18] proposed a scheme that allows secure computation of MAFs, and the χ^2-statistic using homomorphic encryption. Even though they use a specific encoding technique to improve the work presented in [17], they only homomorphically compute the allele counts, and execute other operations on decrypted data. Another work was proposed by Zhang *et al.* [19]. This method allows the computation of χ^2-statistic in the homomorphic domain. To compute the division, a nonlinear operator, authors construct a lookup table linking the division result to the nominator and denominator of the corresponding simplified fraction. This table is encrypted and only known by an authorized party.This one receives the encrypted versions of the fraction numbers and decrypts the results of the division based on the table without the knowledge of the decryption secret key. Even though the proposed strategy performs well, it does not scale enough to treat large-scale data. In [20], Lu *et al.* perform GWAS on homomorphically encrypted genotype and phenotype data. In this method, they use a packing technique for the frequency table to improve the efficiency of their method in terms of communication complexity compared to previous ones. Nevertheless, this method is still limited to a small number of variants.

Recently, Bonte *et al.* [21] proposed two solutions to perform secure GWAS: (1) a somewhat homomorphic encryption (HE) approach, and (2) a secure multiparty computation (SMC) approach. These approaches aim at preventing data breaches when calculating the χ^2-statistic with the idea of not revealing any information other than whether the statistics is significant or not (binary response). Their approach perform better than previous ones taking advantage of a data masking technique so as to perform secure comparison of data between two parties. Unfortunately, while being secure, these methods are most suited for GWAS based on frequencies. Indeed, HE is limited when it comes to statistical analysis processes that are already of great complexity when applied over unencrypted data. Indeed, some algorithms take several days to yield results [2] comparing for instance users' datasets element by element. To better understand the extremely high computation and storage complexity of HE cryptosystems, let us consider the optimized FV cryptosystem [35]. As shown in [36], 707.07 MB of clear data is encrypted into 5.82 GB that is to say a storage overhead 8 times greater. A multiplication of two integers in the clear leads to a multiplication in the cipher domain with a cost of 116 ms on a computer with a processor of 4.2 Ghz. HE also only secures linear operations. Nonlinear functions can be approximated with a reduction of the analysis accuracy as a consequence [37], or shared between parties or with a trusted party at the price of a high increase of communications. To sum up, today, homomorphic encryption based privacy-preserving GWAS are limited in terms of practical use.

Several other SMC secure GWAS methods have been proposed [11]–[15]. Kamm *et al.* [11] present a data collection and computation system where genomic data are distributed among several parties based on additive secret sharing. SS allows several parties to jointly compute the value of a target function f without compromising the privacy of its input data, its output being known to all parties. Constable *et al.* [12] present a privacy-preserving GWAS framework on federated genomic datasets. They secure the χ^2-statistic test on top of SMC systems based on garbled circuit. This latter allows any function to be computed between multiple parties, hiding both their inputs from each other and the outside world. However, this scheme cannot be generalized to more than two participants. Zhang *et al.* [13] propose a secret sharing based SMC approach to secure the χ^2-statistic test, MAF and Hamming distance (HD) computations. Contrarily to [12], this one can be scaled to more than two parties. Cho *et al.* [14] describe a protocol for large-scale genome-wide analysis using multiparty computation techniques. The GWAS method they focus on is a method that enables the identification and the correction for population stratification biases before computing CATT statistics. Bloom [15] proposed a distributed algorithm based on secure multiparty computation in order to secure a linear regression. The works in [11]–[15] show better performance than those based on Homomorphic Encryption. However, they still have an important overhead in terms of communication complexity compared to the same computation in a centralized nonencrypted environment. Thus, this higher complexity hinders practical adoption of SMC solutions over the large-scale genomic data.

To overcome these problems, a few numbers of solutions based on the combination of encryption and hardware-based technologies have been suggested. The basic idea is to isolate sensitive data in a protected enclave that allows secure computation. For instance, Chen *et al.* [22] present a method based on AES encryption and Intel's Software Guard Extensions (SGX). Data are encrypted with AES before being sent to SGX, where data are decrypted before being securely processed. In [24], authors propose a hybrid framework where several algorithms used in GWAS such as Linkage Disequilibrium (LD) computation, Hardy-Weinberg Equilibrium (HWE) test, CATT and Fisher's Exact Test (FET) can be securely performed on federated genomic datasets. They exploit homomorphic encryption and SGX due to the fact that HE allows to compute linear operation over encrypted data in a secure way, especially, the sum of all entities frequencies tables in secure way. Moreover, HE allows to achieve randomness in encrypted data thanks to its probabilistic properties. It is important to know that [38]–[40] recently demonstrate that SGX is sensitive to side-channel attacks. The consequence of these attacks and their possible remedies is an open research problem.

Table 1 sums-up all the above methods accordingly the GWAS algorithm they have been applied to. Most HE cryptosystems that have been used are fully homomorphic (they allow the computation of both addition and multiplication), like BGV, YASHE and FV. Due to their complexity, some other work have been proposed to exploit the Paillier cryptosystem. This one is additive only. Other encryption algorithms that have been used are AES and Lightweight computational footprints (cryptosystems with low computation complexity). As we will see in the following, our solution simply uses *SHA256* (Secure Hash Algorithm) and AES (Advanced Encryption Standard), two well-known and fast cryptographic mechanisms. It is also important to notice that all these proposals do not consider mutualizing genotypes. At the least, parties share frequency tables, after having computed them locally on their respective data, that is to say without sharing these data into a unique server for instance. Moreover, all the methods developed so far considered single marker tests where each marker (SNP) is tested individually. These tests are not useful with rare variants as they will lack power. Only in [16] is the case of rare variants considered

TABLE 1. Previous research works in secure and privacy-preserving GWAS.

Existing techniques	GWAS algorithms	Security mechanisms	Type of variants considered in association test
[6]	χ^2-statistic, FET, Logistic regression	DP	Common
[7]	χ^2-statistic, MAFs	DP	Common
[8]	χ^2-statistic	DP	Common
[9]	χ^2-statistic, GOF	DP	Common
[10]	χ^2-statistic	DP	Common
[11]	χ^2-statistic	Secret sharing	Common
[12]	χ^2-statistic, MAFs	Garbled circuit	Common
[13]	χ^2-statistic, MAFs and HD	Secret sharing, Lightweight computational footprints	Common (Start from VCF file (same entry format as for sequencing data) Association study is only performed with common variants but consider also rare variants in sequence comparison)
[14]	CATT	Secret sharing	Common
[15]	Linear regression	Secret sharing	Common
[16]	Exact logistic regression	BGV	Rare and common (But each SNP is tested individually (one at a time))
[17]	GOF, χ^2-statistic, CATT	FHE (Not developed yet)	Common
[18]	MAFs, χ^2-statistic	BGV and YASHE cryptosystem	Common (Similar to [13] : start from a VCF file only for sequence comparison and do not consider rare variant association tests)
[19]	χ^2-statistic	BGV cryptosystem	Common
[20]	χ^2-statistic	BGV cryptosystem	Common
[21]	χ^2-statistic	Secret sharing, Blinding, FV cryptosystem	Common
[22]	Queries on VCF, many possible computations	AES-GCM cryptosystem, SGX	None (They do not propose association test but solutions to query data on VCF file)
[23]	Transmission Disequilibrium Test	AES-GCM cryptosystem, SGX	Common
[24]	LD, HWE, CATT, FET	Paillier cryptosystem, SGX	Common
Our solution	WSS	Hash, AES	Rare

but the solution proposed is to still test for association at the single locus but use exact logistic regression to deal with parse data. None of the methods proposed solution to perform rare variant burden test at the gene level.

B. CONTRIBUTIONS

In this paper, we present a new secure GWAS protocol adapted to various GWAS statistical analysis, especially iterative ones based on large sets of genotypes provided and shared by different parties in open and nonsecure environments. We were particularly interested by the analysis of sequence data and testing association with rare variants since sequencing data are more informative than genotyping data used to test for association with common variants and considered in all the previous studies. Rare variants that can even be private to a single individual more easily allow individual identification than common variants. To test for association with rare variants, they need to be considered in group within a gene and a score is computed to measure the rare variant burden in each individual and scores are then compared between cases and controls. The Weighted Sum Statistics (WSS) is an example of method commonly used to test for association between rare variants and disease. Like in common GWAS studies, this protocol considers three distinct entities: a Genomic Research Unit (GRU) with genomes of individuals presenting a phenotype (case) who wants to conduct association studies in collaboration with a Genomic Research Center (GRC) who possesses genomes of healthy people (used as control), using the large computation and storage capacities of a cloud service provider (Server).

In our framework, in addition to the common security constraints (all entities are considered as honest but curious (HBC); none of the parties want to disclose their confidential data), we introduce a new constraint: GRU does not want to be identified by the Server. This constraint takes into account the fact that most genomic research units are known for the diseases they are studying. Under the HBC model, this information can for example give clues to an attacker about the name of a gene and its expression for the individuals considered in a study.

The protocol we propose responds to these constraints and more. One originality stands on the fact that GRC serves as an intermediary, similarly to a proxy, in communications between all entities. By doing so, and as we will see, it becomes possible to come back to classical cryptographic tools in order to secure the WSS algorithm, or any algorithm working in a similar way. In particular, our solution takes advantage of the combination of Pretty Good Privacy (PGP) encryption with secure cryptographic hash Functions. Our main idea is that GRC and GRU ensure data confidentiality with the help of secure hash functions salted with a secret key. By using the same hashed data values, GRC and GRU allow the cloud server to conduct WSS counting operations on their data without accessing to their clear text values. More clearly, Server will run WSS on partially hashed data. On its side, PGP is used to secure communications while considering GRC as proxy. As we will see, GRC will never access GRU data while Server will never know GRU's identity nor his confidential data. Compared to actual solutions, our protocol preserves data and WSS result confidentiality with no WSS computation complexity increase. It can be extended to any statistical analysis equivalent to WSS, being iterative or not.

To go further, we extend our proposal under the malicious security model. It is important to notice that all papers listed in Table 1 as well as the vast majority of privacy-preserving GWAS solutions, only consider the semi-honest security model where it is assumed that parties will not alter data. This model is less constraint-full than the malicious model, and leads to computation and communication complexities of lower orders of magnitude. We suggest considering the case where Server is a malicious adversary, that is to say, it can deviate from the protocol and fails the correctness of the output or the input. To overcome this issue, we propose a practical countermeasure based on the zero-knowledge protocol, capable for instance to detect if a malicious server modifies the result of a GWAS study.

The rest of this paper is organized as follows: Section II gives background information about secure cryptographic hash functions, Pretty Good Privacy encryption and Weighted-Sum Statistic algorithm. The details of the proposed protocol are presented in Section III. Experimental results and discussion are given in Section IV. Section VI concludes this paper.

II. PRELIMINARIES

A. SECURE CRYPTOGRAPHIC HASH FUNCTION

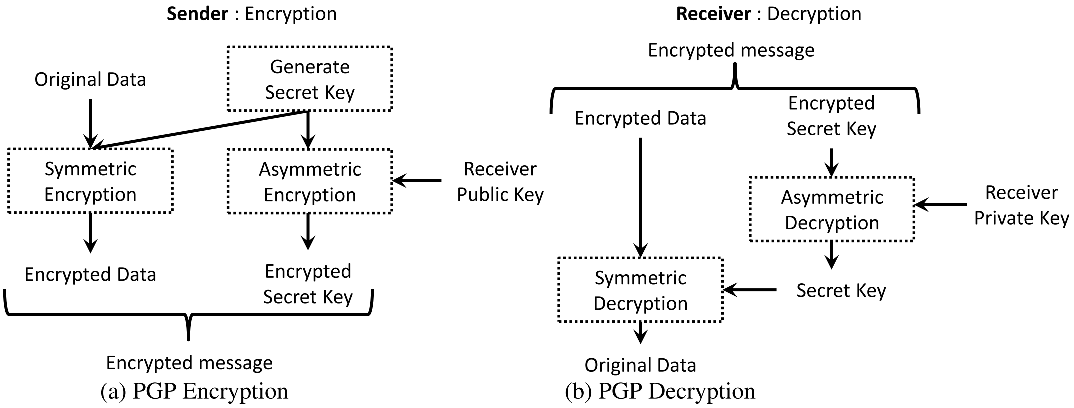
A secure cryptographic hash function takes a set of characters and maps it to a fixed length value (called a hash value). An important property of such functions is that they are one way: it is not possible to get an idea of the input of the function from its output. The hash calculation can also be made secret key dependent. For instance, to do so, a user just has to compute the hash value of a piece of data concatenated with a secret hash key K_{hash}. In the following, the secret hash value a^H of a message a is given by

$$a^H = hash(a||K_{hash}) = SHA256(a||K_{hash}) \tag{1}$$

where || is the concatenation operator, and *SHA256* is the well-known secure hash algorithm standardized by National Institute of Standards and Technology (NIST) [41]. For any data of maximum 2^{64} bits, *SHA256* provides hash value encoded on 256 bits. It has three properties: preimage resistance, second preimage resistance, and collision resistance. More clearly, from a given hash, it is extremely difficult to retrieve the input message. In addition, for a given message, it is extremely difficult to find another message with the same hash as well as to find two different messages with same hashes. Notice that the probability two messages lead to the same hash value is $\frac{1}{2^{128}} \approx 2.9 \times 10^{-39}$.

B. PRETTY GOOD PRIVACY ENCRYPTION

Pretty Good Privacy (PGP) is a well-known secure protocol adapted to the exchange of a large volume of data between two parties. It relies on the combination of a public key encryption (PKE) with a symmetric encryption cryptosystem (see Fig. 1). As given in Fig. 1a, to send a message with PGP, the emitter first symmetrically encrypts it with a secret key. The same key will be used during the decryption process (see Fig. 1b). Then, it asymmetrically encrypts this secret key by the recipient public key and sends both pieces of information (i.e., the symmetrically encrypted message and the asymmetrically encrypted secret key). On its side, the recipient first accesses the secret key by asymmetrically decrypting it



(a) PGP Encryption (b) PGP Decryption

FIGURE 1. PGP protocol from the sender to the receiver.

using his private key. It just has to use this key to finally get access to the message. In this work, PGP is implemented with RSA [42] and AES [43] algorithms, two well-known PKE and symmetric encryption cryptosystems, respectively. RSA is parameterized by a pair of keys (K_p, K_s) where K_p is the public key and K_s the private key while the secret key of AES is noted by K_{AES}. For a given message m and a recipient A, the PGP encryption is such as

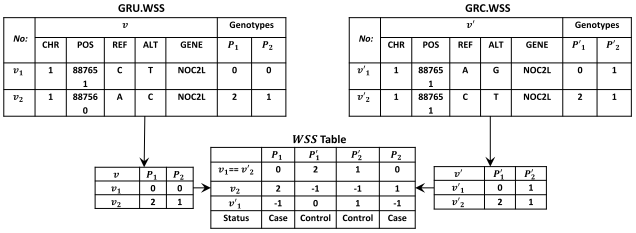
$$(m^e, K^e) = PGP(m, K_p^A, K_{AES}) \tag{2}$$

where m^e is the AES encryption version of m and K^e is the RSA encryption of K_{AES}. m is retrieved from m^e as follows:

$$m = PGP^{-1}(m^e, K^e, K_s^A) \tag{3}$$

C. WEIGHTED-SUM STATISTIC ALGORITHM (WSS)

WSS is one commonly used rare variant association test that was designed to identify the association between a phenotype and rare variants located in a region of the genome (e.g., gene) using sequence data on cases and controls [2]. WSS tests whether there exists an enrichment in rare variant in a gene of interest in cases compared to controls. The input data are two WSS tables. One contains case data, extracted from the database of the Genomic Research Unit (case table: $GRU.WSS$), and the second table contains control data provided by the Genomic Research Center ($GRC.WSS$). As shown in Fig. 2, both tables hold the information about genetic variants for one or more individuals. One line corresponds to one variant uniquely indexed or identified by: the chromosome (CHR) where it is located; its position in this chromosome (POS); the reference allele (REF); the alternate alleles (ALT); and, the name of the gene ($GENE$). Following these five columns is the list of genotypes for the sample of individuals (see P_i and P'_j in Fig. 2). The genotype of a patient at a given position is given by a positive integer indicating the number of alternate alleles the patient has. "0" indicates that both chromosomes of this patient contain the reference allele at this position, "1" indicates that the individual is heterozygous with one REF and one ALT and "2" indicates that the individual is homozygous with 2 ALT alleles. If data is missing then the value is "−1".



GRU.WSS

No:	v					Genotypes	
	CHR	POS	REF	ALT	GENE	P_1	P_2
v_1	1	887651	C	T	NOC2L	0	0
v_2	1	887560	A	C	NOC2L	2	1

GRC.WSS

No:	v'					Genotypes	
	CHR	POS	REF	ALT	GENE	P'_1	P'_2
v'_1	1	887651	A	G	NOC2L	0	1
v'_2	1	887651	C	T	NOC2L	2	1

v	P_1	P_2
v_1	0	0
v_2	2	1

WSS Table

	P_1	P'_1	P'_2	P_2
$v_1 == v'_2$	0	2	1	0
v_2	2	-1	-1	1
v'_1	-1	0	1	-1
Status	Case	Control	Control	Case

v'	P'_1	P'_2
v'_1	0	1
v'_2	2	1

FIGURE 2. Aggregation of cases and controls tables, i.e., of *GRU.WSS* and *GRC.WSS* respectively, in order to produce the WSS table that servers will use as input of the WSS algorithm.

The WSS algorithm requires first to select genomic positions within the gene where there are variants of interest (based on their predicted effect on the gene protein product and on their frequencies) and then to construct a genetic score for each individual based on their genotypes at these different genomic positions and to contrast these genetic scores between cases and controls. To better explain how the WSS algorithm works and its complexity, let us consider one gene that contains v genomic positions where there are variants of interest. The first step consists in merging $GRU.WSS$ and $GRC.WSS$ tables in a single WSS table. To do so, and as illustrated in Fig. 2, individual information on the same variants are grouped together. Genetic scores are then computed as a linear combination of the number of rare alleles carried by the individual at each of the v variants weighted by the minor allele frequency at this position in the control group. The idea is to give more weight to the least frequent variants since these variants are expected to be more often deleterious and thus more likely involved in disease. All individuals affected and unaffected are ranked according to this genetic score and the sum of ranks S_{obs} for affected individuals is calculated. To test the null hypothesis H_0 that the gene is not associated to the disease, a permutation procedure is then used where the case/control status are permuted between individuals N times and the sum of ranks S_{rep} is recomputed each time to obtain the distribution of S under H_0. A *p-value* which is the probability to reject H_0 given H_0 is true is estimated by determining how many time the S_{rep} value obtained on the permuted data exceeds S_{obs}. The null hypothesis is rejected if this *p-value* is less than a fixed threshold value α. Since many different tests are performed, it is necessary to account for multiple testing and fix a very small α value, typically in the range $[10^{-5}, 10^{-8}]$. The WSS algorithm works in four iterative steps. We herein describe them in details in order to give an idea about WSS complexity.

1) For each variant $i \in \{1, 2, \cdots, v\}$, we calculate a weight w_i that depends on the allele frequencies

$$w_i = \sqrt{n_i q_i (1 - q_i)} \tag{4}$$

where: n_i is the number of individuals genotyped for the i^{th} variant (cases and controls), $q_i = \frac{m_i+1}{2d_i+2}$ where d_i is the number of control individuals genotyped for the i^{th} variant, and m_i is the number of minor alleles observed at the i^{th} variant in the control individuals.

2) A genetic score is computed for each individual j:

$$s_j = \sum_{j=1}^{v} \frac{g_{ij}}{w_i} \tag{5}$$

where g_{ij} is the genotype of individual j for the variant i (it takes values 0, 1 or 2 depending on the number of minor alleles).

3) Individuals are ranked accordingly to their genetic scores (s_j) and the rank sum x for affected individuals (cases) is calculated

$$x = \sum_{j \in Cases} rank(s_j) \tag{6}$$

4) A standard permutation test [44] is used to compute an empirical *p-value*. The statuses (case/control) are permuted for all individuals and steps 1 to 3 are repeated k times to obtain k rank sums $x_1, x_2, \cdots, x_k$. These values are compared to the observe rank sum x and the number of permutations k_0 where it exceeds x are determined to obtain the *p-value*:

$$p - value = \frac{k_0 + 1}{k + 1} \quad (7)$$

where k_0 is the number of permutations that give a rank sum x_l at least as extreme as x, and k is total number of permutations (this is a number that will determine the maximum level of significance that can be reached).

III. PROPOSED PRIVACY-PRESERVING WSS ALGORITHM

A. GENERAL GWAS FRAMEWORK AND THREAT MODEL

The scenario considered in order to conduct an outsourced GWAS study is described in Fig. 3 where both GRUs and GRC send their data to a server. Once Server has performed the computation and obtained the *p-value* results, it sends them to the GRUs. In such a framework, and as seen in Section I, different threats have to be considered. Beyond common security needs such as data confidentiality, integrity and availability [45], data privacy is of major concern.

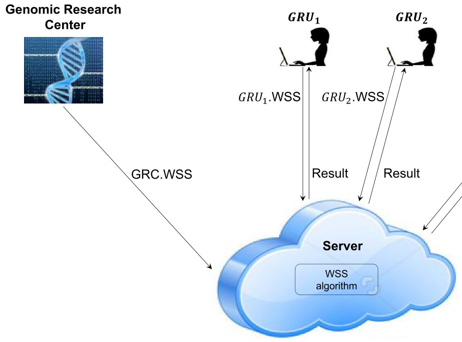


FIGURE 3. General framework of outsourced GWAS-WSS.

The $GRU.WSS$, $GRC.WSS$ and WSS tables contain pieces of information that can be used to identify individuals [46]. Indeed, they provide the genotypes of several individuals for a set of variants, identified by their position (*POS*) on a specific chromosome (*CHR*) (see Section II-B and Fig. 2). Moreover, information is provided on the gene that contains the variants. As a consequence, *CHR*, *POS* as well as *GENE* are very sensitive pieces of information from a privacy point of view. They constitute a potential leak of information with important consequences for an individual and his/her relatives and penalties for institutions [4]. Nevertheless, it is important to notice that knowing genotypes with no information about the gene, the chromosome or the variants they belong to, it is not possible to infer information about individuals. The result of a WSS test along with the knowledge of the gene GRU is interested in, also leak important information [45]. Unfortunately, in the classic framework depicted in Fig. 3, Server knows the identity of GRU, by definition. As a consequence, it has clues about the disease the GRU study focuses on, and so knows the *p-values* that measure the degree of association between all genes and this disease. This can both lead to patient re-identification (if data were taken from a database related to this disease) and to an intellectual property breach about the association of the gene X with the disease Y. As we will see in the next Section III-B, we propose a novel architecture to overcome this problem. It is important to notice that, in a WSS study, even if the server has some knowledge about the study results (i.e., *p-values*) and about unlocalized WSS genotypes, it can not infer significant information without knowing details about the variant and the gene name.

Beyond the sensitivity of WSS data, in our framework, we further assume that first GRC and Server are honest but curious and that they do not collude. More clearly, both of them may try to infer information about confidential data but they will not exchange information they have to keep secret.

To sum up the above discussion, to outsource a WSS computation in such an open environment, the following security constraints have to be considered:

1) Confidential data of GRU (resp. GRC) that can help to identify individuals should not be disclosed to GRC (resp. GRU) and Server.
2) Server should have no idea about the gene GRU is working on, nor on the GRU identity.
3) GRC should not know the results of the WSS (*p-values* of a set of genes) due to the fact it knows the GRU identity and thus the disease the GRU might be interested in.

In the next section, we propose a new framework that satisfies these constraints while securing the WSS algorithm. As we will discuss in Section III-B, the following framework can be extended to any other statistical analysis processes close to WSS, these ones being also concerned by the above constraints and using the same type of inputs.

B. PROPOSED SECURED WSS ALGORITHM

The implementation of our framework will guarantee that all point-to-point communications in-between parties are secured with common security mechanisms (e.g., user authentication, access control policy, firewalls, SSH protocol and so on). Furthermore, in order to escape a man-in-the-middle attack, we assume that the key setup works correctly and that all entities obtain the correct encryption key which can be enforced with appropriate use of Certificate Authorities and/or a Public Key Infrastructure.

As stated above, the framework we propose takes into account a new constraint: Server should not be able to identify GRU, as this knowledge can give clues about the possible disease of the genotyped individuals. To achieve this goal, and as depicted in Fig. 4, we suggest that GRC plays the role of a "proxy" between GRU and Server. More clearly, all communications from GRU to Server and from Server to GRU go through GRC. Server thus has no idea about the GRU. In this situation, we take advantage of PGP in order to ensure the confidentiality of GRC's data. To do so and as explained in Section II, GRU first AES encrypts his data

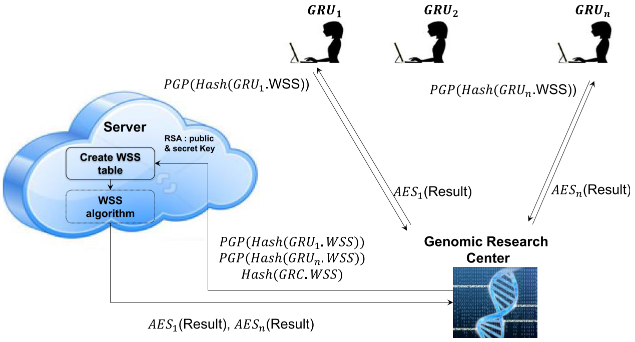


FIGURE 4. Our secure GWAS-WSS framework.

based on an AES secret key it generates and, then sends these data along with the AES secret key asymmetrically encrypted with the Server RSA public key. Only Server will be able to access the AES key and consequently decrypt the data. Server can conduct this task without knowing the identity of GRU. As GRC has no knowledge of the AES key nor Server's Private Key, it is unable to decrypt GRU data while transmitting them to Server. The second important point to manage is to make it possible for Server to compute the WSS algorithm without being able to identify the variants of GRU and GRC. To ensure the confidentiality of GRC and GRU variants, the confidential attributes CHR, POS, REF, ALT and $GENE$ values in $GRC.WSS$ and $GRU.WSS$ tables are substituted by secret hashed values, computed with a cryptographic hash function based on a secret hash key K_{hash} GRU and GRC previously agreed on through the use of a secure channel of communication. This step allows the creation of secured WSS tables without compromising GRU and GRC data security. Notice that genotype data in $GRC.WSS$ and $GRU.WSS$ are not modified. As seen in Section I, this does not endanger individual privacy as Server does not know the real variant's genomic location and alleles.

In the following, we give more details about this protocol when only one GRU collaborates with GRC to conduct a WSS study, but it can easily be extended to support several GRUs. If GRC provides several data sets, it is of course essential that GRU selects the one most suited to its analysis and especially that cases and controls are matched on ethnicity to limit population stratification bias. Thus, let us consider that one GRU wants to perform a WSS study with GRC for a specific gene so as to see if this latter is associated to a phenotype. Prior to any security consideration, we assume that GRU and GRC have followed common guidelines to produce their data, and that similar quality controls have been applied on the data. Let us also assume that Server has a RSA pair of key (K_p^S, K_s^S). The main steps of our protocol which are depicted in Fig. 5 works as follows:

1) **Secret hash key management**: GRC and GRU first have to agree on a unique secret hash key K_{hash} using a secure key exchange protocol like the SFTP protocol [47].
2) **Data confidentiality**: GRU and GRC substitute the confidential attribute values in their WSS tables (i.e., $GRU.WSS$ and $GRC.WSS$, respectively), by secure

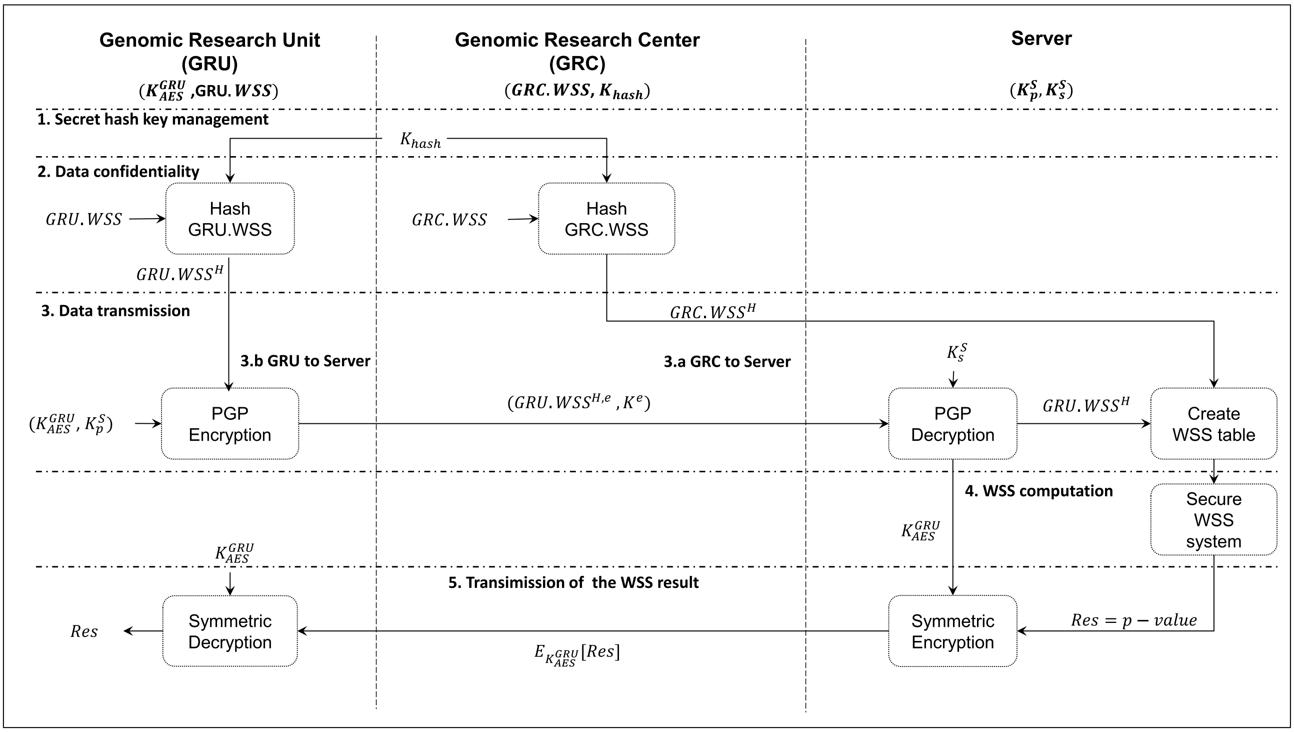


FIGURE 5. Different steps of our secured WSS protocol in the case of one gene.

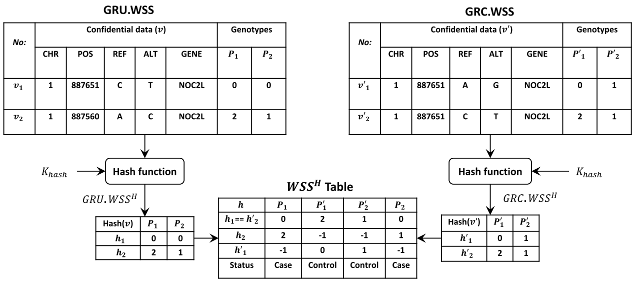
hash values using the secret hash key K_{hash}. More clearly, taking $GRU.WSS$ as example, GRU computes:

$$\begin{aligned} &hash(CHR_i||POS_i||REF_i||ALT_i||K_{hash}) \\ &\quad ||hash(GENE||K_{hash}) \\ &\quad = v_i^H||hash(GENE||K_{hash}) = h_i \end{aligned} \tag{8}$$

where the confidential attributes CHR_i, POS_i, REF_i, ALT_i and $GENE$ constitute what we name in the following the variant v_i. It can be noticed that in (8), we concatenate the secret hashes of the variant confidential attributes with the one of the gene (i.e., "$GENE''$). This is due to the fact WSS computes one *p-value* per gene and not per variant (see Section II-B). Server has thus to be able to discriminate the variants located on the same gene. In the case GRU just wants to study one gene, then h_i can be refined in

$$\begin{aligned} hash(CHR_i||POS_i||REF_i||ALT_i||K_{hash}) &= v_i^H \\ &= h_i \end{aligned}$$

The resulting hash tables are referred to as $GRU.WSS^H$ and $GRC.WSS^H$. An example of this process is given in Fig. 6. Finally, GRC sends its hashed table $GRC.WSS$ to Server



GRU.WSS

No:	Confidential data (v)					Genotypes	
	CHR	POS	REF	ALT	GENE	P_1	P_2
v_1	1	887651	C	T	NOC2L	0	0
v_2	1	887560	A	C	NOC2L	2	1

GRC.WSS

No:	Confidential data (v')					Genotypes	
	CHR	POS	REF	ALT	GENE	P'_1	P'_2
v'_1	1	887651	A	G	NOC2L	0	1
v'_2	1	887651	C	T	NOC2L	2	1

$GRU.WSS^H$

Hash(v)	P_1	P_2
h_1	0	0
h_2	2	1

WSS^H Table

h	P_1	P'_1	P'_2	P_2
$h_1 == h'_2$	0	2	1	0
h_2	2	-1	-1	1
h'_1	-1	0	1	-1
Status	Case	Control	Control	Case

$GRC.WSS^H$

Hash(v')	P'_1	P'_2
h'_1	0	1
h'_2	2	1

FIGURE 6. Creation of the secure WSS table from the hashed versions of $GRU.WSS$ and $GRC.WSS$. h_i and h'_i represent the hash values of v_i and v'_i, respectively.

3) **Data transmission-**

a) **GRC to Server**: GRC sends $GRC.WSS^H$ to Server. Due to the fact that the communication between GRC and Server is point-to-point, and by definition secured (see above), there is no need to use PGP.

b) **GRU to Server**: GRU securely sends its secured table $GRU.WSS^H$ to Server using PGP. To do so, it generates the PGP symmetric key K_{AES}^{GRU}. Then it entirely PGP encrypts them, that is to say (see Section II-B).

$$\begin{aligned} &(GRU.WSS^{H,e}, K^e) \\ &\qquad = PGP(GRU.WSS^H, K_p^S, K_{AES}^{GRU}) \end{aligned}$$

where K_p^S is the Server RSA public key. Next, GRU sends $(GRU.WSS^{H,e}, K^e)$ to Server through GRC so as to preserve its privacy.

4) **WSS computation**- When Server receives $(GRU.WSS^{H,e}, K^e)$, it first decrypts the AES key K_{AES}^{GRU} from K^e using its RSA secret key K_s^S. Then, it AES deciphers $GRU.WSS^{H,e}$ to get access to $GRU.WSS^H$. Server also gets the data from GRC. As shown in Fig. 6, Server creates the WSS hashed table (WSS^H) from $GRU.WSS^H$ and $GRC.WSS^H$ (see Section II-C). Due to the fact that genotype data are not encrypted, Server can directly apply WSS on WSS^H. Indeed, the WSS algorithm is not modified. It will simply work with hashed values instead of real values, by comparing hashed values of genes to group variants and hashed values of variants to group genotypes.

5) **Transmission of WSS result**- Once Server obtains the WSS results, that is to say the Gene's WSS *p-value* (see Section II-C), it AES encrypts it using the GRU AES key (K_{AES}^{GRU}) and sends it to GRU through GRC. Finally, GRU just has to decrypt this piece of data using the same AES Key to get access to the results of its WSS study. By doing so, its identity is never revealed to Server.

Notice that, in the case GRU wants to analyze several genes, it will receive as many *p-values* from Server. In order to generalize this approach to more than one GRU willing to pool their data for more powerful statistical studies, all GRUs will follow the same steps as above:

i) They hash their sensitive data (variants) by using GRC secret key K_{hash}. Since all of them have access to the public key of Server, they encrypt their WSS table with PGP parameterized with their respective AES key and the public key of Server.

ii) The encrypted data are sent to Server through GRC.

iii) As shown in Fig. 4, Server decrypts the PGP encrypted $WSS.GRU$ tables and merges them with the $WSS.GRC$ table.

iv) Finally, Server runs the WSS algorithm, encrypts the results using the AES Key of each GRUs before sending it through GRC. The results received by each GRU contains only the *p-values* associated to the genes that particular GRU provided.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed secure GWAS framework was tested on real genetic data: exome data were compared between (1) 100 healthy individuals from the FrEx project [48] that served as GRC control data and (2) 59 individuals affected by a rare disease sequenced independently to the FrEx data (GRU cases). Cases and controls were sequenced on the same platform (CNRGH, Evry, France) at different times and using the Agilent SureSelect Human all exon V5 capture kit for the cases and the Agilent SureSelect Human all exon V5+UTR capture kit for the controls. Sequence data were processed using the exome analysis platform developed at CNG, which follows GATK best practices. Coverage/depth statistics were as follow: for each sample a minimum of 20X coverage for 80% of the targets was obtained and the average sequencing depth was of at least 70 to 80X. Polymorphism detection for each sample was performed using read mapping procedure onto the reference genome (hg19) followed by "SNP calling" algorithm implemented in GATK/samtools software. Stringent quality controls were performed after variant and genotype calling. Only genotypes with min GQ $\geq$ 20 and min DP $\geq$ 10 were kept and the other genotypes were set to missing. Variants failing any of the following thresholds in any of the two datasets were discarded from both GRC and GRU datasets: min callrate $\geq$ 0.9, HQ variants (as define in ExAC: 80% of genotype with DP > 10 & GQ > 20, at least one variant genotype with DP > 10 & GQ > 20), min QD $\geq$ 2, min inbreeding coef ≥ -0.8, ABhet in the range [0.25; 0.75], min MQRanSum ≥ -12.5, max FS $\leq$ 60 for SNV or $\leq$200 for INDEL, max SOR $\leq$ 3 for SNV or $\leq$10 for INDEL, min MQ $\geq$ 40 for SNV or $\geq$10 for INDEL, min ReadPosRankSum ≥ -8 for SNV or ≥ -20 for INDEL. Note that each party is expected to perform this same QC on its own dataset and send to the other party the list of variant sites excluded (only chromosome, position, reference and alternative alleles and no individual data).

In our example, a total of 11196 genes contained at least two qualifying variants and were tested for association. Qualifying variants kept in the analysis were those with an expected effect on the encoded protein (i.e., variants that were annotated as transcript ablation, splice acceptor or donor, stop gained or lost, start lost, frameshift, inframe insertion or deletion and missense) and variants with a Minor Allele Frequency below 0.05.

To compute the genetic score, missing genotypes were replaced by the most frequent genotype in the sample at the variant position. The WSS algorithm was run on each gene with a maximum of 10^9 permutations, and the overall runtime was 10 hours and 18 minutes on a server with 56 processors at 2.40 GHz and 512 GB RAM running on Ubuntu 16.04 LTS. Since in our implementation no encrypted data are used in the actual computation, runtime is the same as in the classical implementation of the algorithm. The only difference is an overhead of a few seconds to hash, encrypt and decrypt the input tables. Furthermore, the WSS *p-values* obtained for each gene are similar to the ones obtained from doing the same test on non-distributed data.

To determine if batch effects could be a concern linked to the fact that cases and controls were not sequenced together, we produced the corresponding QQ-plot as suggested in different works [49]–[51] and we computed an inflation factor [52]. This inflation factor was obtained by transforming the observed *p-values* into one degree-of-freedom χ^2-statistic and computing the median of these values divided by the expected median of the corresponding one degree-of-freedom χ^2 distribution.

Visual inspection of the QQ-plot (see Fig. 7) suggests that the stringent QC performed was efficient at correcting for batch effects and it even leads to conservative results with an inflation factor below 1 ($\lambda = 0.75$). This was however a favorable situation as cases and controls were sequenced on the same platform with capture kits that were only slightly different.

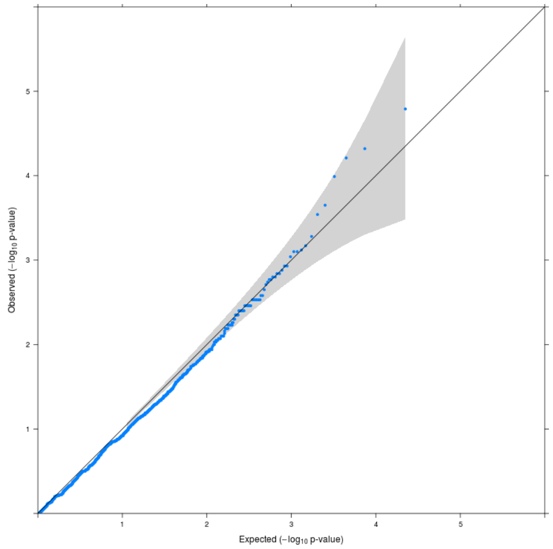


FIGURE 7. Quantile-Quantile plot of the WSS test *p-values* obtained when comparing exomes from 59 cases coming from one project against 100 controls coming from another project. Cases and controls were sequenced on the same sequencing platform but at different times and using different capture kits. The same variant calling pipeline was used and stringent QC were performed. Results are presented for each of the 11196 genes that contain at least two qualifying variant for the association test. The genomic inflation factor is $\lambda = 0.75$..

A. COMPUTATION AND COMMUNICATION COMPLEXITY

On the GRU and GRC sides, the computation complexity corresponds to the WSS table hashing and encryption processes. Notice that *SHA256* and AES computation complexities are low and increase linearly with the size of the WSS table. To give an idea, it takes about 0.53s to both hash and to AES encrypt the WSS table of 406 gene and 733 patients. Regarding Server, this one has to: 1) decrypt the $GRU.WSS$ table, 2) merge $GRU.WSS$ and $GRC.WSS$ into the complete WSS table and 3) perform the WSS algorithm before AES encrypting the WSS results. Here, the complexity of step 2) and 3) are the same as working with data in their clear form. The complexity overhead stands on the AES decryption of WSS tables; complexity which is quite close to the AES encryption process.

One can also notice in this Table 2 that our WSS implementation was parallelized in order to increase its speed. As seen in Section II-C after computing the rank sum x at the step 2, the status (case/control) is permuted k times so as to compute the *p-value*.

To take advantage of a server with multiple processing units (e.g., $PU_1, \ldots, PU_n$), this permutation test can be separated into k/n parts of n permutations, namely $\{x_{1,j}, \ldots, x_{n,j}\}_{j=1..k/n}$ where $x_{i,j}$ is the j^{th} rank-sum permutation computed at processing unit PU_i (see Section II-C). As the processing units $PU_1, \ldots, PU_n$ can run in parallel, the *p-value* computation at step 4 (see Section II-C) becomes as follows:

$$p - value = \frac{\sum_{j=1}^{k/n}\sum_{i=1}^{n}(x > x_{i,j}) + 1}{k+1} \tag{9}$$

where k is the number of permutations. Therefore, the use of parallel computation significantly increases the WSS algorithm speed, as experimentally shown in Table 2.

TABLE 2. Computational costs of the WSS algorithm with and without parallelism.

Number of gene	Parallel WSS algorithm (56 cores)	Standard WSS algorithm	*p-value*	k_0	number of permutation (k)
1	20.22 sec	14 min	5.504e-5	5	109000

The communication complexity of our secure WSS algorithm for GRU or GRC is bounded by $O(n)$ bits where n is the size in bits of the WSS table. Compared to the nonsecured WSS algorithm, the communication overhead corresponds to the size of the hash key K_{hash} and to the RSA encryption of the AES key. This overhead does not depend on the size of the WSS table and is very small. Therefore, it is negligible compared to the rest of the WSS data to transmit.

B. DISCUSSION AND SECURITY ANALYSIS

The following analysis considers the semi-honest adversary model where it is assumed that parties involved in the protocol do not collude but try to infer information about sensitive data; that is to say GRU and GRC data. In our scheme, the confidentiality of WSS tables during their communication is ensured by the AES cryptosystem, the security of which has been demonstrated in [43]. GRC will never have any clues about the GRU data, these being PGP exchanged with Server. Once decrypted on the Server side, the confidentiality of the sensitive attributes of these tables (e.g., *CHR*, *POS*, *GENE* and so on, see Section III-B) stands on the secure hash function *SHA256*, the security level of which has been investigated in [53]. It is not possible for Server to retrieve the original sensitive attribute values from their hash values without the knowledge of the hash Key. This key is only known from GRC and GRU. Notice that, the fact GRC sends several times its data to Server for different studies is not a problem at the condition a new secret hash key is used. Doing so makes the computation of *SHA256* values semantically secure (i.e., the same variant has different hashes values for distinct studies). Notice that, as GRC has no knowledge about GRU AES key (K_{AES}^{GRU}), it can not access to the hashed GRU table nor to the results provided by Server.

Beyond data confidentiality, one must also consider statistical inference techniques that can be used for the re-identification of genomic data donors. These attacks have been extensively investigated [31]. They depend on the *a priori* knowledge one can have of the frequencies of genotypes for given variants or a gene. Homer *et al.* [46] showed that inference techniques could be used to identify the presence/absence of an individual in a genomic dataset from aggregate statistics (e.g., allele frequencies). In [4], authors presented an attack for genomic data sharing beacons (publicly available genomic databases). This attack aims at seeing if an individual is in a beacon or not. To do so, they assume that the attacker has the genomic profile of an individual and a VCF file [54] listing all the variants for this individual. From the variants, and more specifically from the heterozygous alternate alleles of the victim, the attacker generates some queries he next addresses to the beacons. Based on the responses, he conducts a statistical hypothesis test so as to decide if the victim is present in a particular beacon.

In our framework because GRU and GRC hash their confidential variants' values, Server is not able to conduct such an attack. In fact, Server has no idea about the variants and the genes being evaluated. This statement is valid at the condition GRU or GRC do not collude with Server. For instance, if Server and GRC collude, they have access to the AES and hash keys and can consequently breach GRU data confidentiality. Nevertheless, it is hard to believe that GRC or Server would collude, as their reputations are invaluable assets.

Although Genomic Research Units (GRUs) are known for the diseases they are working on, that is to say the genes that they more frequently focus on, Server cannot deduce any clues from GRU identity due to the fact Server only communicates with GRC; GRC which acts as a proxy.

To go further, one can notice that all papers listed in Table 3, as well as the vast majority of genome privacy solutions, only consider the semi-honest security model. This one assumes that all entities involved follow the protocol and will not try to alter data or the result of a process. At the same time, under this model, solutions are significantly easier to instantiate with computation and communication of smaller complexities than under the malicious model. Under this latter model, there is no guaranty that the association test or patient information are not going to be altered. For instance, Server could modify the WSS algorithm or change the correct value of the *p-value*. To overcome this issue and to extend our framework under such a malicious model, we propose a zero-knowledge protocol. In this one, GRC sometimes plays the role of GRU and GRC at the same time. By doing so, GRU sends to Server both the *GRU.WSS* and *GRC.WSS* tables for which GRC has already the knowledge of the result (i.e., the *p-value*). If GRC finds that the *p-values* computed by Server did not match the pre-computed *p-values*, it can then deduce that Server is malicious.

It is important to notice that our framework is not limited to secure WSS association tests, it can easily be extended to any other GWAS statistic algorithms that rely on the same kind of data. CAST, SKAT [55] and SKAT-O [56] are association

TABLE 3. Comparison of the most representative genomics privacy methodologies. Columns correspond performance criteria. Meaning of the acronyms: (Security) - Sh - semi-honest model - NC noncollude model; (overhead) L.S.O: low storage overhead, H.S.O: high storage overhead, L.T.O: low time overhead, H.T.O: high time overhead, L.T.O: low communication overhead, H.T.O: high communication overhead.

References	Security Model	Security Mechanisms	Overhead	Utility Loss
[6]–[10]	–	DP	L.S.O, L.T.O, L.C.O	High
[11]	SH, NC	Secret sharing	L.S.O, H.T.O, H.C.O	High
[12]	SH	Garbled circuit	L.S.O, H.T.O, H.C.O	Low
[13]	SH	Secret sharing, Lightweight computational footprints	L.S.O, H.T.O, H.C.O	High
[14]	SH, NC	Secret sharing	L.S.O, H.T.O, H.C.O	Low
[15]	–	Secret sharing	L.S.O, L.T.O, H.C.O	Low
[16]	SH	BGV	H.S.O, H.T.O, H.C.O	Low
[17]	SH, NC	FHE	H.S.O, H.T.O	High
[18]	SH	BGV, YASHE	H.S.O, H.T.O	High
[19]	SH	BGV	H.S.O, H.T.O	High
[20]	SH	BGV	H.S.O, H.T.O, L.C.O	Low
[21]	SH	Secret sharing, Blinding, FV	H.S.O, L.T.O	Low
[22]	Malicious	AES-GCM, SGX	L.S.O, L.T.O, L.C.O	Low
[23]	Malicious	AES-GCM, SGX	L.S.O, L.T.O, L.C.O	Low
[24]	SH	Paillier, SGX	H.S.O, L.T.O, H.C.O	Low
Our work	SH or malicious, NC	Hash, AES	L.S.O, L.T.O, L.C.O	Low

tests that can be implemented in our framework. Another useful method that could be implemented is Principle Component Analysis (PCA). This statistical method, run before the GWAS algorithm itself, can ensure that the merged dataset can be used to perform such an analysis. Indeed, a PCA where GRU and GRC data are separated indicates that any signal obtained through GWAS is unreliable and results from divergent quality of the data or population stratification.

The pieces of data they rely on and which are sensitive from a confidentiality/privacy point of view can also be replaced by secure hash values.

V. COMPARISON TO THE EXISTING SOLUTIONS

Comparing in terms of performance our framework with other proposals from the literature is a nontrivial task because each work in the genome privacy does not necessarily secure the same process. For this reason, we compare whenever is possible the secure versions to the nonsecure versions of the same functionality. Inspired by [57], we choose different criteria aiming at capturing different aspects related to security, efficiency, and data utility. They correspond to:

A. PERFORMANCE CRITERIA

- **Privacy Overhead.** It quantifies the overhead introduced by the security mechanisms used to secure an association test. All solutions given in Table 3 have been analyzed in order to assess their efficiency in terms of communication, time and storage overhead in comparison with their nonsecured counterpart. We quantify these performances by means of three values: Communication - L.C.O: Low Communication Overhead vs. H.C.O: High Communication Overhead; Storage - L.S.O: Low Storage Overhead, H.S.O: High Storage Overhead; Time - L.T.O: Low Time Overhead, H.T.O: High Time Overhead.
- **Utility Loss.** This criterion evaluates the impact of privacy tools on the utility of the association test. This measurement also includes the overall flexibility of the proposed solution with the intended task. We quantify the utility loss on two levels: High or Low.
- **Security model.** It indicates which security model has been considered by the authors: semi-honest model or malicious model.

As shown in table 3, all methods based on differential privacy (DP) induce a utility loss compared to the same process over clear data. This is due to the fact these schemes add a noise to the data. Homomorphic encryption (HE) can help to solve this problem but at the price of significant computational and storage overheads. Most of the time, they are impractical for real life applications [58]. Secure multiparty computation (SMC) constitutes a nice alternative due to its lower computational overhead. However, garbled circuit-based need complex and optimized circuit design limiting its flexibility and usability, greatly. On its side, secret sharing involves huge communication overhead and is not suitable for client server architecture. Secure hardware-based approaches, like SGX based techniques, isolate sensitive data into a protected enclave for secure computation. However, they remain sensitive to side-channel attacks [40]. Notice that the full extent of SGX security has yet to be explored.

Compared to the previous solutions, our framework is based on PGP and *SHA256*, two cryptographic mechanisms of very low complexity, contrarily to HE. Furthermore, we do not intrinsically modify the association test algorithm. Sensitive data in terms of confidentiality are substituted by secret hash values. Thus, and as shown in Section III, our framework preserves the accuracy of the association test. That is not the case of DP [7], [8], [32]. Server can also conduct the WSS algorithm without the need of additional communication as required in approaches based on SMC [11]–[15], [21] or to encrypt homomorphically the genotypes as proposed in [18]–[21], [24] which leads to high computation and storage complexity. Thus, our solution has no loss of accuracy and insignificant overheads (in memory, computation and communication) compared to the original WSS algorithm.

B. STATISTICAL POWER CRITERIA

Implementations of secured association tests proposed in the literature have considered single variant association tests that are mostly performed on genotyping data. In our work, we have considered rare variant association tests where, rather than testing each variant individually, we grouped them within a unit of analysis, here the gene. Rare variant association tests explore alternative genetic architectures for common diseases than the classical ≪ common disease-common variant≫ model that was considered before. Indeed, different real examples and simulation studies have shown that rare variants might contribute more than common variants to common diseases [59]. To study the impact of these rare variants on disease susceptibility, it is necessary to sequence the genome of individuals and the sharing of sequence data is even more problematic than the sharing of genotyping data since sequence data contain information on all the genetic variants present in an individual genome including deleterious variants possibly involved in monogenic diseases that the individual could develop in the future and could transmit to offspring. It is therefore important to specifically address the problem of rare variant association tests as we have done here in a general framework that could also integrate common variant tests. This is the case in our proposed framework that could easily be extended to include other statistical tests and measures considered in previous works such as χ^2-statistic, Fisher's Exact Test, Logistic regression, MAF test, Cochran-Armitage Test for Trend, Goodness of Fit, Hardy-Weinberg Equilibrium. In the same way, we have only implemented one rare variant association test here but our framework is general enough to allow the easy implementation of other rare variant association tests including variance component tests that are widely used in rare variant association studies [60].

Contrary to the WSS test we have implemented here, some of the tests can be adjusted on covariates such as age or gender. Information on these covariates for each individual could be transmitted by the GRU to the GRC and to the GRC to the Server together with the WSS tables. Some particular covariates on which adjustment could also be required to avoid false positives due to population stratification are leading principal components (PCs) from the principal component analysis performed on genotypes data of both cases and controls. To obtain these leading PCs, a possibility will be to add ancestry informative SNPs and exchange information on individual genotypes at these SNPs to perform principal component analysis on the Server. This will however involve the sharing of genetic data. Another possibility could be to use spectral graphs in a manner similar to the approach suggested by Bodea *et al.* [61] or the singular value decomposition suggested by Artomov *et al.* [62]. This will however require some further developments that are beyond the scope of this paper. Another concern when comparing sequence data of cases and controls that were not generated together is the possibility of systematic bias due to batch effects. The problem is even more drastic when different platforms are used to sequence cases and controls. Different studies have evaluated these biases and proposed some solutions to reduce them [49]–[51]. Strict quality control is key in this process and it is also important to visualize QQ-plot in order to diagnose any inflation of the statistics. We have illustrated this in the example provided and shown that with the strict QC parameters we used the QQ-plot was not inflated. In this example however, cases and controls were sequenced on the same platform and only the capture kits were slightly different. In less favorable conditions, it might be necessary to test different QC parameters to determine the best combinations. This would require some extra-computations and a lighter version of the test where cases and controls statuses are not permuted should perhaps then be considered to fix the QC parameters. It might also be necessary to pre-select some different sets of parameters with different levels of QC and evaluate the level of inflation by computing a statistics similar to the genomic inflation factor [52].

VI. CONCLUSIONS

In this paper, we have proposed a new privacy-preserving GWAS framework that allows performing in a secure way association tests similar to the WSS algorithm. Its main originality relies (1) on a Genomic Research Center which acts as proxy in order to preserve the privacy of Genomic Research Units, (2) on Pretty Good Privacy to secure communications and (3) on cryptographic hash function to ensure the confidentiality of sensitive data in WSS input tables. The security analysis of our solution demonstrates that it is secure under the honest but curious adversarial model and robust to statistical inference attacks. We also have extended our framework under the malicious security model by means of zero-knowledge protocol. Experimental results conducted on real genomic data demonstrate that the proposed solution achieves the same performances and accuracy as the nonsecured WSS algorithm. Consequently, it can be used in real world environments contrarily to other proposed solutions based on Homomorphic encryption. Furthermore, this solution can be extended to any other GWAS algorithms similar to the WSS algorithm. Future works will focus on adapting our protocol considering that parties can collude.

REFERENCES

[1] M. H. Wang, H. J. Cordell, and K. Van Steen, "Statistical methods for genome-wide association studies," *Seminars Cancer Biol.*, vol. 55, pp. 53–60, Apr. 2019.

[2] B. E. Madsen and S. R. Browning, "A groupwise association test for rare mutations using a weighted sum statistic," *PLoS Genet.*, vol. 5, no. 2, pp. 1–11, Feb. 2009, doi: 10.1371/journal.pgen.1000384.

[3] The Wellcome Trust Case Control Consortium, "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," *Nature*, vol. 447, no. 7145, p. 661, 2007.

[4] S. S. Shringarpure and C. D. Bustamante, "Privacy risks from genomic data-sharing beacons," *Amer. J. Hum. Genet.*, vol. 97, no. 5, pp. 631–646, Nov. 2015.

[5] J. Schlesinger. (2018). *Dark Web is Fertile Ground for Stolen Medical Records*. [Online]. Available: https://www.cnbc.com/2016/03/10/dark-web-is-fertile-ground-for-stolen-medical-records.html

[6] A. Johnson and V. Shmatikov, "Privacy-preserving data exploration in genome-wide association studies," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2013, pp. 1079–1087. [Online]. Available: http://doi.acm.org/10.1145/2487575.2487687

[7] C. Uhler, A. B. Slavković, and S. E. Fienberg, "Privacy-preserving data sharing for genome-wide association studies," *J. Privacy Confidentiality*, vol. 5, no. 1, p. 137, Aug. 2013.

[8] F. Tramèr, Z. Huang, J.-P. Hubaux, and E. Ayday, "Differential privacy with bounded priors: Reconciling utility and privacy in genome-wide association studies," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1286–1297.

[9] D. Kifer and R. Rogers, "A new class of private chi-square tests," 2016, *arXiv:1610.07662*. [Online]. Available: http://arxiv.org/abs/1610.07662

[10] Y. Sei and A. Ohsuga, "Privacy-preserving chi-squared testing for genome SNP databases," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 3884–3889.

[11] L. Kamm, D. Bogdanov, S. Laur, and J. Vilo, "A new way to protect privacy in large-scale genome-wide association studies," *Bioinformatics*, vol. 29, no. 7, pp. 886–893, Apr. 2013.

[12] S. D. Constable, Y. Tang, S. Wang, X. Jiang, and S. Chapin, "Privacy-preserving GWAS analysis on federated genomic datasets," *BMC Med. Informat. Decis. Making*, vol. 15, Dec. 2015, Art. no. S5.

[13] Y. Zhang, M. Blanton, and G. Almashaqbeh, "Secure distributed genome analysis for GWAS and sequence comparison computation," *BMC Med. Informat. Decis. Making*, vol. 15, Dec. 2015, Art. no. S4.

[14] H. Cho, D. J. Wu, and B. Berger, "Secure genome-wide association analysis using multiparty computation," *Nature Biotechnol.*, vol. 36, no. 6, p. 547, 2018.

[15] J. M. Bloom, "Secure multi-party linear regression at plaintext speed," 2019, *arXiv:1901.09531*. [Online]. Available: http://arxiv.org/abs/1901.09531

[16] S. Wang, Y. Zhang, W. Dai, K. Lauter, M. Kim, Y. Tang, H. Xiong, and X. Jiang, "HEALER: Homomorphic computation of ExAct logistic rEgRession for secure rare disease variants analysis in GWAS," *Bioinformatics*, vol. 32, no. 2, pp. 211–218, 2015.

[17] K. Lauter, A. López-Alt, and M. Naehrig, "Private computation on encrypted genomic data," in *Proc. Int. Conf. Cryptol. Inf. Secur. Latin Amer.* Cham, Switzerland: Springer, 2014, pp. 3–27.

[18] M. Kim and K. Lauter, "Private genome analysis through homomorphic encryption," *BMC Med. Informat. Decis. Making*, vol. 15, Dec. 2015, Art. no. S3.

[19] Y. Zhang, W. Dai, X. Jiang, H. Xiong, and S. Wang, "FORESEE: Fully outsourced secuRe gEnome study basEd on homomorphic encryption," *BMC Med. Informat. Decis. Making*, vol. 15, Dec. 2015, Art. no. S5.

[20] W.-J. Lu, Y. Yamada, and J. Sakuma, "Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption," *BMC Med. Informat. Decis. Making*, vol. 15, Dec. 2015, Art. no. S5.

[21] C. Bonte, E. Makri, A. Ardeshirdavani, J. Simm, Y. Moreau, and F. Vercauteren, "Privacy-preserving genome-wide association study is practical," IACR Cryptol. ePrint Arch., Tech. Rep., 2018, p. 955.

[22] F. Chen, C. Wang, W. Dai, X. Jiang, N. Mohammed, M. M. Al Aziz, M. N. Sadat, C. Sahinalp, K. Lauter, and S. Wang, "PRESAGE: PRivacy-preserving gEnetic testing via SoftwAre guard extension," *BMC Med. Genomics*, vol. 10, no. 2, p. 48, Jul. 2017.

[23] F. Chen, S. Wang, X. Jiang, S. Ding, Y. Lu, J. Kim, S. C. Sahinalp, C. Shimizu, J. C. Burns, V. J. Wright, E. Png, M. L. Hibberd, D. D. Lloyd, H. Yang, A. Telenti, C. S. Bloss, D. Fox, K. Lauter, and L. Ohno-Machado, "PRINCESS: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions," *Bioinformatics*, vol. 33, no. 6, pp. 871–878, 2016.

[24] M. N. Sadat, M. M. Al Aziz, N. Mohammed, F. Chen, X. Jiang, and S. Wang, "SAFETY: Secure gwAs in federated environment through a hYbrid solution," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 1, pp. 93–102, Jan. 2019.

[25] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin, "A cryptographic approach to securely share and query genomic sequences," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 5, pp. 606–617, Sep. 2008.

[26] R. Ghasemi, M. M. Al Aziz, N. Mohammed, M. H. Dehkordi, and X. Jiang, "Private and efficient query processing on outsourced genomic databases," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 5, pp. 1466–1472, Sep. 2017.

[27] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.* Berlin, Germany: Springer, 2008, pp. 1–19.

[28] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[29] B. Malin and L. Sweeney, "How (not) to protect genomic data privacy in a distributed network: Using trail re-identification to evaluate and design anonymity protection systems," *J. Biomed. Informat.*, vol. 37, no. 3, pp. 179–192, Jun. 2004.

[30] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "*L*-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, p. 24, 2006.

[31] Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nature Rev. Genet.*, vol. 15, no. 6, p. 409, 2014.

[32] S. Simmons, C. Sahinalp, and B. Berger, "Enabling privacy-preserving GWASs in heterogeneous human populations," *Cell Syst.*, vol. 3, no. 1, pp. 54–61, Jul. 2016.

[33] R. Bellafqira, G. Coatrieux, D. Bouslimi, G. Quellec, and M. Cozic, "Proxy re-encryption based on homomorphic encryption," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Dec. 2017, pp. 154–161.

[34] D. Niyitegeka, G. Coatrieux, R. Bellafqira, E. Genin, and J. Franco-Contreras, "Dynamic watermarking-based integrity protection of homomorphically encrypted databases–application to outsourced genetic data," in *Proc. Int. Workshop Digit. Watermarking*. Cham, Switzerland: Springer, 2018, pp. 151–166.

[35] C. Aguilar-Melchor, J. Barrier, S. Guelton, A. Guinet, M.-O. Killijian, and T. Lepoint, "NFLlib: Ntt-based fast lattice library," in *Proc. Cryptograph. Track RSA Conf.* Cham, Switzerland: Springer, 2016, pp. 341–356.

[36] J. L. Raisaro, G. Choi, S. Pradervand, R. Colsenet, N. Jacquemont, N. Rosat, V. Mooser, and J.-P. Hubaux, "Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 5, pp. 1413–1426, Oct. 2018.

[37] M. Blatt, A. Gusev, Y. Polyakov, K. Rohloff, and V. Vaikuntanathan, "Optimized homomorphic encryption solution for secure genome-wide association studies," IACR Cryptol. ePrint Arch., Tech. Rep., 2019, vol. 2019, p. 223.

[38] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A.-R. Sadeghi, "Software grand exposure: SGX cache attacks are practical," in *Proc. 11th USENIX Workshop Offensive Technol. (WOOT)*, 2017, pp. 1–12.

[39] M. Hähnel, W. Cui, and M. Peinado, "High-resolution side channels for untrusted operating systems," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2017, pp. 299–312.

[40] M. Schwarz, S. Weiser, and D. Gruss, "Practical enclave malware with Intel SGX," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*. Cham, Switzerland: Springer, 2019, pp. 177–196.

[41] A. Biryukov, M. Lamberger, F. Mendel, and I. Nikolić, "Second-order differential collisions for reduced SHA-256," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2011, pp. 270–287.

[42] P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: An engineering perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 83:1–83:33, Dec. 2017. [Online]. Available: http://doi.acm.org/10.1145/3124441

[43] J. Daemen and V. Rijmen, *The Advanced Encryption Standard Process*. Berlin, Germany: Springer, 2020.

[44] H. A. David, "The beginnings of randomization tests," *Amer. Statist.*, vol. 62, no. 1, pp. 70–72, Feb. 2008, doi: 10.1198/000313008X269576.

[45] R. Wang, Y. F. Li, X. Wang, H. Tang, and X. Zhou, "Learning your identity and disease from research papers: Information leaks in genome wide association study," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2009, pp. 534–544. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653726

[46] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genet.*, vol. 4, no. 8, Aug. 2008, Art. no. e1000167.

[47] M. Krishna, P. Jamwal, K. S. R. Chaitanya, and B. V. Kumar, "Secure file multi transfer protocol design," *J. Softw. Eng. Appl.*, vol. 4, no. 5, pp. 311–315, Mar. 2011. [Online]. Available: https://www.scirp.org/journal/PaperInformation.aspx?PaperID=5070

[48] E. Genin, R. Redon, J.-F. Deleuze, D. Campion, J.-C. Lambert, and J.-F. Dartigues, "The French exome (FREX) project: A population-based panel of exomes to help filter out common local variants," *Int. Genet. Epidemiol. Soc.*, vol. 41, no. 7, p. 691, 2017.

[49] A. Mahajan and N. Robertson, "Rare variant quality control," in *Assessing Rare Variation in Complex Traits*. New York, NY, USA: Springer, 2015, pp. 33–43.

[50] J. A. Tom, J. Reeder, W. F. Forrest, R. R. Graham, J. Hunkapiller, T. W. Behrens, and T. R. Bhangale, "Identifying and mitigating batch effects in whole genome sequencing data," *BMC Bioinf.*, vol. 18, no. 1, p. 351, Dec. 2017.

[51] K. Panoutsopoulou and K. Walter, "Quality control of common and rare variants," in *Genetic Epidemiology*. New York, NY, USA: Springer, 2018, pp. 25–36.

[52] B. Devlin and K. Roeder, "Genomic control for association studies," *Biometrics*, vol. 55, no. 4, pp. 997–1004, Dec. 1999.

[53] S. Bakhtiari *et al.*, "Cryptographic hash functions: A survey," Dept. Comput. Sci., Citeseer, Univ. Wollongong, Wollongong, NSW, Australia, Tech. Rep. 09, Jul. 1995.

[54] P. Danecek, A. Auton, G. Abecasis, C. A. Albers, E. Banks, M. A. DePristo, R. E. Handsaker, G. Lunter, G. T. Marth, S. T. Sherry, G. McVean, and R. Durbin, "The variant call format and VCFtools," *Bioinformatics*, vol. 27, no. 15, pp. 2156–2158, Aug. 2011.

[55] M. C. Wu, S. Lee, T. Cai, Y. Li, M. Boehnke, and X. Lin, "Rare-variant association testing for sequencing data with the sequence kernel association test," *Amer. J. Hum. Genet.*, vol. 89, no. 1, pp. 82–93, Jul. 2011.

[56] S. Lee, M. J. Emond, M. J. Bamshad, K. C. Barnes, M. J. Rieder, D. A. Nickerson, D. C. Christiani, M. M. Wurfel, and X. Lin, "Optimal unified approach for rare-variant association testing with application to small-sample case-control whole-exome sequencing studies," *Amer. J. Hum. Genet.*, vol. 91, no. 2, pp. 224–237, Aug. 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3415556/

[57] A. Mittos, B. Malin, and E. De Cristofaro, "Systematizing genome privacy research: A privacy-enhancing technologies perspective," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 1, pp. 87–107, Jan. 2019.

[58] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proc. 3rd ACM Workshop Cloud Comput. Secur. Workshop*, 2011, pp. 113–124.

[59] A. S. Pierre and E. Génin, "How important are rare variants in common disease?" *Briefings Funct. Genomics*, vol. 13, no. 5, pp. 353–361, Sep. 2014.

[60] B. M. Neale, M. A. Rivas, B. F. Voight, D. Altshuler, B. Devlin, M. Orho-Melander, S. Kathiresan, S. M. Purcell, K. Roeder, and M. J. Daly, "Testing for an unusual distribution of rare variants," *PLoS Genet.*, vol. 7, no. 3, Mar. 2011, Art. no. e1001322.

[61] C. A. Bodea, B. M. Neale, and S. Ripke, "A method to exploit the structure of genetic ancestry space to enhance case-control studies," *Amer. J. Hum. Genet.*, vol. 98, no. 5, pp. 857–868, 2016.

[62] M. Artomov, A. A. Loboda, M. N. Artyomov, and M. Daly, "A platform for case-control matching enables association studies without genotype sharing," *BioRxiv*, 2018, Art. no. 470450. [Online]. Available: https://www.biorxiv.org/content/early/2018/11/14/470450

REDA BELLAFQIRA (Member, IEEE) received the M.Sc. degree in information security and cryptology from the University of Limoges, Limoges, France, in 2014, and the Ph.D. degree from the IMT Atlantique Bretagne Pays de la Loire, Brest, France, in 2017.

Since 2018, he has been a Postdoctoral Researcher with the Department of Information and Image Processing, IMT Atlantique, and conducting his search activities within the Laboratory of Medical Information Processing, Inserm UMR1101, and the joint laboratory Security and Processing of Externalized Medical Image Data (SePEMeD). His research interest includes data security, with a particular interest for protecting medical images and database from malicious entities and service providers using cryptographic and watermarking tools.

THOMAS E. LUDWIG received the M.S. degree in computer science and the Ph.D. degree in life sciences from Strasbourg University, France, in 2004 and 2008, respectively. He joined the French National Institute of Health and Medical Research (INSERM) unit 1078, Brest, in 2015. He is currently involved in genetics and bioinformatics research focused on pathologies as well as population genetics.

DAVID NIYITEGEKA (Member, IEEE) received the B.Sc. degree in mathematics from Moulay Ismail University, Meknes, Morocco, in 2013, and the M.Sc. degree in information security and cryptology from the University of Limoges, Limoges, France, in 2015. He is currently pursuing the Ph.D. degree with the Department of Information and Image Processing, IMT Atlantique, and the Laboratory of Medical Information Processing, INSERM U1101, Brest, France, focusing on genomic data protection based on different security mechanisms, including homomorphic encryption and watermarking.

His main research interests include genomic information security, encryption, and watermarking.

EMMANUELLE GÉNIN received the Ph.D. degree from the University Pierre and Marie Curie of Paris. Her Ph.D. research work was on the contribution of inbreeding to the study of human disease and was performed under the supervision of Françoise Clerget-Darpoux with the Inserm unit of Josué Feingold.

After her Ph.D., she took a postdoctoral position at the Glenys Thomson Laboratory, University of California at Berkeley. She contributed in many projects aiming to discover genes causing monogenic diseases—in particular rare recessive diseases by homozygosity mapping. She has an expertise in the study of isolated populations and the statistical methods that can be used in this particular context. She is also interested in the study of gene-environment (GXE) interactions and has developed a method to study GXE interactions in genome-wide association studies when no information is available on exposure factors in the controls as it is often the case when reference control panels are used. In September 2009, she was awarded a French Government Fellowship at the Churchill College in Cambridge and spent a year at the Sanger Institute with the Genome Dynamics and Evolution Team where, in collaboration with the Team Leader, Matthew Hurles, and David Clayton from the University of Cambridge, she studied methods to test for association with rare variants. She also worked with Ele Zeggini from the Sanger Institute to study the stratification of rare variants in the U.K. population using the Welcome Trust Case Control Consortium GWAS data. In September 2012, she moved to Brest to start some new projects focusing on the population of Brittany and the characterization of genetic variants in this population. She has authored more than 100 original articles.

GOUENOU COATRIEUX (Senior Member, IEEE) is currently a Full Professor with IMT Atlantique, Brest, France, and conducts his research in the Laboratory of Medical Information Processing, Inserm UMR1101, Brest, where he leads the research axis Multimedia Medical Information Analysis, Protection and Secondary Use. He is also the Head of the joint laboratory Security and Processing of Externalized Medical Image Data (SePEMeD). His primary research interests include watermarking, crypto-watermarking, secure processing of data, and digital forensics. He is a member of the International Federation for Medical and Biological Engineering "Global Citizen Safety and Security Working Group" and the European Federation for Medical Informatics "Security, Safety, and Ethics Working Group," and has contributed to the Technical Committee of the "Information Technology for Health" of the IEEE EMB Society. He is also an Associate Editor of the Innovation and Research in Bio-Medical and a Past Associate Editor of engineering and the IEEE Journal on Biomedical and Health Informatics and *Digital Signal Processing*.