Robust and Imperceptible Watermarking Scheme for GWAS Data Traceability

Reda Bellafqira[1](✉), Musab Al-Ghadi[1], Emmanuelle Genin[2], and Gouenou Coatrieux[1]

[1] IMT Atlantique, Inserm, UMR 1101 LaTIM, Brest, France
{reda.bellafqira,musab.al-ghadi,
gouenou.coatrieux}@imt-atlantique.fr
[2] Inserm UMR 1078, Brest, France
emmanuelle.genin@inserm.fr

Abstract. This paper proposes the first robust watermarking method of outsourced or shared genomic data in the context of genome-wide association studies (GWAS) with the primary purpose of identifying the individual or entity at the origin of an illegal information redistribution or disclosure. Our scheme's first unique feature is that it employs a database watermarking strategy to take advantage of the fact that GWAS data are stored in variant call format (VCF) files, which have a database-like structure. Second, it proposes a quantization index modulation based on watermarking modulation for GWAS data under the constraint of not interfering with identifying candidate variants or genes involved in the pathology. We evaluate the theoretical performance of our method in terms of watermarking insertion capacity, distortion, and robustness against different attacks. Experimental results conducted on real data and the weighted-sum statistic (WSS) GWAS study demonstrate the efficiency of the proposed scheme and that it can be used for identifying the cloud service providers (geneticists) at the origin of an information disclosure even if the genotype data has been modified.

Keywords: Information security · Genome-wide association studies (GWAS) · Traceability · Watermarking · Genomic data

1 Introduction

Nowadays, genomic data are widely collected, stored, processed, and shared for various genomic applications. They can be used in legal and forensics, where a DNA (DeoxyriboNucleic Acid) sample found on a victim or at a crime scene may be exploited as a shred of evidence by law enforcement to track down suspected criminals. In healthcare, genomic data are guiding medical decisions. For instance, it has been demonstrated that women with specific genetic variants in the BRCA (BReast CAncer) genes have about

R. Bellafqira and M. Al-Ghadi—Equal contributions.

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 through the project TADOP.

X. Zhao et al. (Eds.): IWDW 2022, LNCS 13825, pp. 147–161, 2023.
https://doi.org/10.1007/978-3-031-25115-3_10

an 80% chance of developing breast cancer [1]. Therefore, identification of some individuals who carry these variants can help them to opt for preventive mastectomies [2]. In research, genomic data are being used to discover new associations between traits and some diseases. In this case, association tests are conducted through GWAS, the objective of which is to detect genetic variants that are associated with some complex disorders or diseases [3–6].

In general, a GWAS corresponds to an observational study of a set of genetic variants in the genomes of different individuals in order to see if any variant or a set of variants located in a specific region of the genome (e.g., a set of genes) is associated with a disease [7]. The usual design to conduct a GWAS test is a cases-controls, where genotype distributions at different genetic positions are compared between samples of individuals affected by the disease of interest (cases) and unaffected individuals from the same population (controls). This is the case of the WSS algorithm [8], the objective of which is to compare the number of genotypes in a set of genetic variants from both cases and controls for a studied gene. These association tests are externalized in cloud environments to access high-capacity storage and computation capabilities. However, outsourcing genomic data induces several security issues in confidentiality, traceability, integrity, or traitor tracing, ranging from unintentional disclosure of data due to human errors to planned attacks. Furthermore, genomic data are vulnerable because they allow the owner's unique identification [9]. In this work, we are interested in securing genomic data used in case-control studies such as WSS by watermarking to ensure traitor tracing, i.e., the identification of individuals who are the origin of illegal information disclosure. Different tools have been proposed in order to ensure the security of outsourced data. They are based on various security mechanisms such as encryption [10], digital signatures [11,12], data structure [13] or watermarking [14,15]. Even though digital signature and data structure-based solutions are more commonly used in database management systems to verify integrity, they introduce additional information in the data. Encryption-based methods allow the protection of data confidentiality, but data are no longer protected once they are decrypted. Contrary to all these categories, watermarking relies on the invisible embedding of a message, i.e., a watermark into host data, by imperceptibly modifying them with the constraint that the introduced distortion is controlled. This mechanism leaves access to watermarked data while maintaining them protected. Depending on the relationship between the host data and the embedded watermark, watermarking solutions can be used to achieve different security goals, such as ensuring data integrity, protecting copy rights, or finding spies. These methods were proposed either for using genomic data as a storage mediums [16,17], for protecting messages in genomic data [18,19] or for protecting genomic data themselves [20–22]. They can be used for watermarking the DNA of living organisms or not. All these methods allow genomic data watermarking for various purposes. They were proposed for cellular DNA, and they can not be used for genomic data outsourced for GWAS.

This paper presents the first robust watermarking method for ensuring traitor tracing for genomic data externalized for GWAS studies. Our method is unique in that it employs a database watermarking strategy to capitalize on the fact that GWAS data are stored in Variant Call Format (VCF) files, which have a database-like structure. And, it proposes a quantization index modulation (QIM) [23] based on watermarking modulation for GWAS data under the constraint of not interfering with identifying candidate variants or genes involved in the pathology. The contributions of this paper can be summarized as follows: (i) To the best of our knowledge, the proposed approach is the first attempt to demonstrate the application of watermarking on genomic data, specifically securing the variant genetic sequences stored in the VCF file and used in GWAS. (ii) The watermark is secretly embedded within genomic data without violating genomic processing, such as identifying candidate variants or genes involved in the pathology. (iii) The results show that the proposed approach inserts the watermark in the genomic data with very low data distortion and high robustness to common watermark attacks such as tuples suppression and addition.

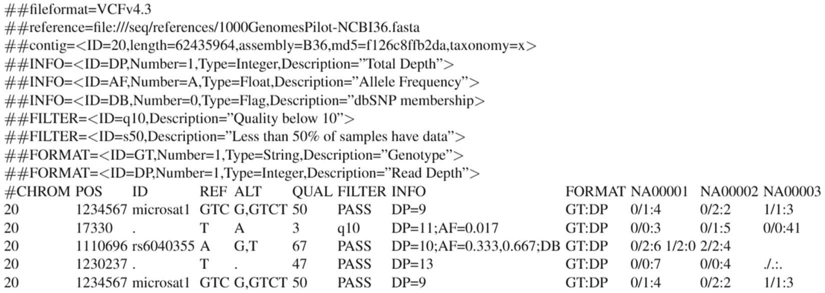
The rest of this paper is organized as follows: In Sect. 2, we come back to the introduction to genomic data and database models. Section 3 provides the details of the watermarking solution we propose, while Sect. 4 details the theoretical performance of our solution. Experimental results and discussion are presented in Sect. 5 and conclusions are given in Sect. 6.

2 Genomic Data and Database Model

This section briefly introduces genomic data used in GWAS, particularly VCF files and weighted sum statistic (WSS) files, before detailing the database model.

2.1 Genomic Data

The human body is made up of billions of cells where each has one nucleus, and this nucleus contains 23 pairs of chromosomes. The complete set of all the DNA contained in one cell is called the genome, and the basic unit of heredity is a particular part of the genome called a gene. Human beings have all the same number of genes, each controlling a particular behavior. However, some behaviors do not express themselves in the same way. These differences between individuals' genomes are called genetic variations. Genetic variants account for about 1% of the difference between two people. One can distinguish three main types of genetic variants [24]: SNP (Single Nucleotide Polymorphisms) that corresponds to a substitution of a single nucleotide at a specific position in the genome; indels (insertions/deletions) that correspond to the insertion or deletion of several nucleotides in the genome; and structural variants that correspond to the deletions, duplications, or rearrangements of large sections of a chromosome or even whole chromosomes. Genetic variants are common genomic data that are used for performing GWAS, and these data are kept in VCF files [25]. The variant call format was developed in order to standardize large-scale genetic variant sharing and storage. A VCF file corresponds to a text file consisting of three parties: meta-data lines, a header line, and data lines. Meta-data lines that begin the file and are included after ## provide data line descriptions. The header line started by # names the columns for data lines. Finally, data lines follow the header line, and each data line or record represents one variant of a given position in the genome. Among several columns per data line present in the VCF file, eight of them are fixed. These are: **CHROM**: a unique identifier from the reference genome that corresponds to chromosome number. **POS**: refers to the position of first base on the reference genome. **ID**: a unique identifier for each record if it exists. **REF**: reference base(s). **ALT**: alternate base(s). **QUAL**: a measure of the quality of the identification of ALT. **FILTER**: filter status. **INFO**: gives additional information such as the number of individuals, frequency alleles, etc. If genotype data is present, fixed columns are followed by a FORMAT column which specifies the data types and order, then an arbitrary number of genotyped individuals. Notice that the dot '.' symbol represents missing value. As shown in Fig. 1, each column that represents genotyped individuals contains genotype information with the same data type indicated in the FORMAT column. One of the significant components of the genotype information is genotype values (GT) which encodes alleles as numbers separated by '|' or '/'; 0 indicates the reference allele, 1 indicates the first allele listed in ALT, 2 indicates the second allele listed in ALT and so on. Therefore, GT could be $0/0, 0/1, 1/2, ./1$ or $1/1$, etc.



```
##fileformat=VCFv4.3
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126c8ffb2da,taxonomy=x>
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership>
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO                        FORMAT NA00001 NA00002 NA00003
20     1234567 microsat1 GTC G,GTCT 50   PASS   DP=9                        GT:DP  0/1:4   0/2:2   1/1:3
20     17330   .         T   A      3    q10    DP=11;AF=0.017              GT:DP  0/0:3   0/1:5   0/0:41
20     1110696 rs6040355 A   G,T    67   PASS   DP=10;AF=0.333,0.667;DB GT:DP  0/2:6 1/2:0 2/2:4
20     1230237 .         T   .      47   PASS   DP=13                       GT:DP  0/0:7   0/0:4   ./.:.
20     1234567 microsat1 GTC G,GTCT 50   PASS   DP=9                        GT:DP  0/1:4   0/2:2   1/1:3
```

Fig. 1. An example of VCF file. It stores genomic data, in particular, genetic variants.

2.2 Weighted Sum Statistic (WSS) Method

To conduct GWAS, individuals who are affected (cases) and unaffected (controls) are genotyped to produce thousands or up to millions of genetic variants stored into VCF files. After that, an intermediary step is conducted to generate other files specific to each GWAS. In this paper, we are interested in watermarking WSS files. As illustrated in Table 1, a WSS file is composed of the following columns: CHROM, POS, ID, REF, ALT, and an arbitrary number of individuals. The WSS is GWAS method that was proposed in [8] as a tool for the identification of the association of rare variants with diseases. Some studies have pointed out that groups of multiple rare variants together can explain a large proportion of the genetic basis for some diseases. In WSS, variants are grouped according to their biological functionality (e.g., gene), and each individual is scored by a weighted sum of the variant counts. To test for an excess of variants in affected individuals, we use a permutation of disease status among affected and unaffected individuals. Using permutations, the WSS method adjusts the variant weights and the requirement that a mutation is observed to be included in the study. In WSS, rare variant counts within the same gene for each individual are accumulated rather than collapsing. Then, it introduces a weighting term to emphasize alleles with a low frequency in unaffected individuals. Finally, the scores for all individuals are ordered, and then, WSS is computed as the sum of ranks for cases. A permutation procedure determines the significance based on the p-value.

Table 1. An example of VCF file; it stores genomic data, particularly genetic variants.

#CHROM	POS	ID	REF	ALT	NA00001	NA00002	NA00003
20	1234567	microsat1	GTC	G	0	0	1
20	1234567	rs6040355	GTC	GTCT	1	2	0

3 Proposed Database Watermarking Scheme for GWAS Data

In this section, we first present the standard chain of database watermarking [26], the QIM and, by next, the watermarking scheme we propose for WSS data. Before entering in detail, we illustrate in the Table 2 the acronyms used in our scheme.

Table 2. Acronyms that are used in the watermarking method we propose.

Δ	Quantization step (distortion factor)	w	A watermark bit $w \in 0, 1$
N_g	Number of groups	S_g	Number of tuples for each group
G	Group of tuples, i.e. $\{G_i\}_{i=1,\cdots,N_g}$	G^A	Sub-group A of tuples in G
G^B	Sub-group B of tuples in G	$\|C_0^A\|$	Cardinality of zero values in sub-group G^A
$\|C_0^B\|$	Cardinality of zero values in sub-group G^B	d	$\|C_0^A\|$ - $\|C_0^B\|$
N_c	Number of columns in the WSS file	D_Δ	Percentage of modulation for a given Δ
N_r	Number of tuples in the WSS file (i.e. $S_g \times N_g$)	db_{size}	Size of WSS file (i.e. $N_r \times N_c$)

3.1 Database Watermarking

By definition, a database is an organized collection of data that are generally stored and accessed from a computer system. Formally, a database *DB* refers to a finite set of tables or relations $\{R_i\}_{i=1,\cdots,N_r}$. From hereon and for sake of simplicity, we will consider a database that contains one single relation constituted of N tuples $\{\mathrm{t}_u\}_{u=1,\cdots,N}$, each of M attributes $\{A_1, A_2, \cdots, A_M\}$. The attribute A_n takes its values within an attribute domain, and $t_u.A_n$ refers to the value of the n^{th} attribute of the u^{th} tuple of the database. The value $t_u.PK$ is an attribute value or a set of attribute values, represents the unique identifier of each tuple of the database. In the literature, most schemes that have been proposed for database watermarking follow the process illustrated in Fig. 2.

This process is based on two basic procedures: watermark embedding and watermark detection/extraction. The watermark embedding procedure includes a pretreatment, the purpose of which is to make the watermark insertion/extraction independent of the database structure or the way the database's data is stored. To do so, database tuples are grouped into N_g non-overlapping groups $\{G^i\}_{i=1,\cdots,N_g}$. This grouping is usually conducted by calculating the index number $n_u \in [0, N_g - 1]$ of each group for the tuple t$_u$ such that

$$n_u = H(K_w|H(K_w|t_u.PK)) \mod N_g \tag{1}$$

where H, K_w, and | represent the cryptographic hash function, the secret watermarking key, and the concatenation operator, respectively. We use a cryptographic hash function, such as the Secure Hash Algorithm (SHA), to ensure certain grouping and equal distribution of tuples into different groups. After database partitioning, one bit of the watermark is inserted into each group of tuples by modifying or modulating attribute values accordingly to the rules of the retained watermarking modulation, such as the order of database tuples [27]. Therefore, within a database of N_g groups, a watermark $W = \{w_i\}_{i=1,\ldots,N_g}$ of N_g bits is embedded. The watermark detection works similarly. First, the database is partitioned into N_g groups based on the secret watermarking key K_w. Then, one watermark bit is extracted or detected from each group based on the used modulation. In the sequel, we explain the proposed method, which follows these procedures and is based on QIM and majority vote.

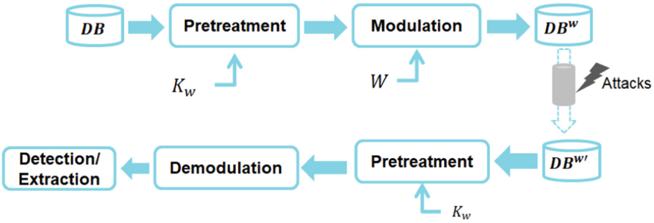


Fig. 2. A common database watermarking chain.

3.2 Quantization Index Modulation (QIM)

QIM [28] relies on quantifying the host data (e.g., image, database) components by rounding each component to the nearest odd/even quantized value according to the value of the watermark bit w and a quantization step size Δ. More specifically, let $w \in \{0,1\}$ be a watermark bit, Δ be a quantization step size that controls the level of distortion. In the QIM method, according to the value of the watermark bit to be embedded, the host data components are shifted by $\pm\Delta$. In this work, we apply this QIM method in order to embed one watermark bit w_i into each group of tuples, i.e. $\{G^i\}_{i=1,\cdots,N_g}$. More clearly, let $w_i \in \{0,1\}$ be a watermark bit, Δ be a quantization step size that control the level of distortion and d be the difference between the cardinality of zero values in sub-group G_A ($|C_0^A|$) and sub-group G_B ($|C_0^B|$) for each individual (P_i: $i = 1, \cdots, |\text{patients}|$), where

$$d = |C_0^A|_{P_i} - |C_0^B|_{P_i} \tag{2}$$

According to the value of w_i, d is rounded to the nearest even/odd quantized value using the quantization step size Δ. Therefore, the embedding modulation is performed as follows:

$$d^* = (\lfloor \frac{d}{\Delta} \rfloor + (\lfloor \frac{d}{\Delta} \rfloor_2 ! = w)) \times \Delta \tag{3}$$

3.3 Watermark Embedding in WSS Data

In this work, we consider a framework which is composed by three entities: a Genomic Research Unity (GRU), a Genomic Research Center (GRC) and a Cloud Services Provider (CSP). GRU and GRC decide to outsource their genetic data on the cloud for storage and/or processing purposes. Before being outsourced, these data are watermarked so as to ensure their copyright protection and traitor tracing. To do so, we describe in this section a robust database watermarking scheme that allows message embedding for WSS data. Let us consider a WSS database *DB*, which consists of many genes, our solution is implemented as follow:

- First the table *DB* is secretly reorganized into the database DB^r. To do so, data owner assigns a primary key $v_u.PK$ for each variant $v_u \in \{u = 1, \cdots, |variants|$ $\}$, where $v_u.PK = CHROM\|POS\|GENE$. Then, this primary key is used for partitioning the database into N_g groups using a secret watermarking key K_w. The group index number for each variant n_{v_u} is computed based on secure hash algorithm using (4) and N_g groups $\{G_i\}_{i=1,2,\cdots,N_g}$, are constituted.

$$n_{v_u} = H(K_w(H(v_u.PK|K_w)) \mod N_g \tag{4}$$

  Once all groups are obtained, one bit of the watermark is embedded into each group.
- The data owner (in our case GRU/GRC) generates a binary watermark
  $W = \{w_1, w_2, \cdots, w_{N_g}\}$ uniformly distributed.
- Each group G_i of the database is divided into two tuple sub-groups G_i^A and G_i^B, based on the secret watermarking key K_w. To do so, the sub-group index number n_{gv_u} for each variant v_u in G_i, is computed using secure hash algorithm such that:

$$n_{gv_u} = H(K_w||(H(v_u.PK||K_w)) \mod 2 \tag{5}$$

  If the value $n_{gv_u} = 1$, then the variant v_u belongs to G_i^A, otherwise ($n_{gv_u} = 0$), then it belongs to G_i^B.
- QIM is used for embedding one watermark bit in these sub-groups so as to produce the watermarked sub-groups $G_{A,i}^W$ and $G_{B,i}^W$. The watermark embedding process is illustrated in Algorithm 1 according to three cases. After sub-group watermarking, the watermarked database DB^{rw} is constituted.

3.4 Watermark Extraction

It is worth noting that during watermarking process, one watermark bit is embedded in each database column. Thus, during extraction stage a majority vote is performed in order to decide which watermark bit will be extracted. Indeed, majority vote is one

Algorithm 1. Watermark embedding modulation in one group

```
INPUT: Subgroups $G_i^A$ and $G_i^B$ of G$_i$, A watermark bit $w_i$, a quantization step size $\Delta$
procedure GroupWatermarking($G_i^A$,$G_i^B$,$w_i$,$\Delta$, $d$ = $|C_0^A| - |C_0^B|$)
    if $\lfloor \frac{d}{\Delta} \rfloor \%2 == w_i$ then
        $d^* = \lfloor \frac{d}{\Delta} \rfloor \times \Delta$
    else
        $d^*$ = $\lfloor \frac{d}{\Delta} \rfloor \times \Delta + \Delta$
    end if
    modulationValue = abs($d^*$ - $\lfloor \frac{d}{\Delta} \rfloor$)
Case 1
    if $d^* \geq$ d and $|C_0^B| \geq$ modulationValue then
        $|C_0^{BW}|$ = $|C_0^B|$ - modulationValue
Case 2
    else if d$^* <$ d and $|C_0^A| \geq$ modulationValue then
        $|C_0^{AW}|$ = $|C_0^A|$ - modulationValue
Case 3
    else
        Not embeddable group
    end if
    return $G_{A,i}^W$,$G_{B,i}^W$
end procedure
```

of the popular optimal algorithms which is used to find the majority element among the given elements that have more than $\frac{N}{2}$ occurrences. However, watermark reading works similarly. The watermarked database DB^w is first reorganized into N_g groups, and each group $\{G_i^W\}_{i=1,\cdots,N_g}$ is partitioned into two sub-groups $G_{A,i}^W$,$G_{B,i}^W$. From each group, one message bit w_{P_i} is detected and extracted in each column according to the Eq. (6). After that, a majority vote is conducted in order to decide which watermark bit is extracted. While tuple primary keys are not modified, the knowledge of the watermarking key ensures synchronization between watermark embedding and watermark detection/extraction. The watermark extraction process is illustrated in Algorithm 2. We discuss the theoretical performances of our solution in the next section before presenting experimental results.

$$w_{P_i} = \lfloor \frac{d^{w*} + \frac{\Delta}{2}}{\Delta} \rfloor \mod 2 \tag{6}$$

where

$$d^{w*} = |C_0^{AW}|_{P_i} - |C_0^{BW}|_{P_i}$$

4 Theoretical Performance

In this section, we start by presenting the constraints of some parameters in the proposed algorithm and then present the theoretical performance of our scheme in terms of distortion introduced to data during the watermarking, the insertion capacity, and the robustness against different database watermarking attacks.

Algorithm 2. Watermark extraction in one group

```
INPUT: Subgroups $G^W_{A,i}$,$G^W_{B,i}$ of $G^W_i$, a quantization step size $\Delta$
procedure WATERMARK DETECTION($G^W_{A,i}$,$G^W_{B,i}$,$\Delta$)
    for each individual ($P_{i:i=1,\cdots,|patients|}$) in $G^W_i$ do
        $d^{w*} = |C^{AW}_0|_{Pi} - |C^{BW}_0|_{Pi}$
        $w_{P_i} = \lfloor \frac{d^{w*} + \frac{\Delta}{2}}{\Delta} \rfloor \mod 2$
    end for
    $w^{'}_i$ = majority-vote($w_{P_{i:i=1,\cdots,|patients|}}$)
    return extracted watermark ($w^{'}_i$)
end procedure
```

4.1 Parameter Constraints

In our solution, in order to work properly and intuitively, some constraints such as distortion factor (Δ), number of groups in the database (N_g), number of tuples in the database (N_r) and the probability to have 0 in one group (Pr_0) must to be defined and respected. These constraints are such that

$$\frac{S_g}{2} > \Delta \Leftrightarrow \frac{N_r}{N_g} > 2 \times \Delta \Leftrightarrow N_r > 2 \times \Delta \times N_g \Leftrightarrow Pr_0 \times N_r > 2 \times \Delta \times N_g \quad (7)$$

this constrain is important, because the number of zeros in a sub-group should be greater than the distortion factor Δ otherwise we can't embed the watermark into the group. As we will see later, this constraint will help us in analyzing the performance of our watermarking method.

4.2 Distortion Performance

Let us consider a database DB which contains N_c columns, N_r rows and db_{size} attribute values. During the watermarking process, this database is divided into N_g groups, and each group is partitioned into two sub-groups. If S_g is the number of tuples in one group. Then, the distortion value D_Δ for the database DB corresponds to the number of modified attribute values in the database for a given Δ, and can be computed as follows:

$$D_\Delta = N_g \times \frac{\Delta}{2} \times N_c \Longrightarrow D_\Delta = \frac{N_r}{S_g} \times \frac{\Delta}{2} \times N_c \Longrightarrow D_\Delta = db_{size} \times \frac{\Delta}{2 \times S_g} \quad (8)$$

As example, if we take Δ = 2 and S_g = 100, then we can say that the distortion is $\frac{1}{100}$ of the db_{size}. This is due to symmetric distribution for the difference value of zero frequency between sub-group G^A and sub-group G^B. This distortion does not disturb the results of WSS as we will see in Sect. 5.

4.3 Robustness Performance

In this section, we analyze the robustness of our watermarking scheme under two well-known database attacks that are deletion attacks and insertion attacks. We evaluate the robustness of our solution by means of the bit error rate (BER), which corresponds to the ratio of the number of incorrectly extracted watermark bits to the number of the original watermark bits. BER is such that:

$$BER = \frac{\sum_{i=1}^{N_g} w_i \oplus w_i^{'}}{N_g} \tag{9}$$

where w_i and $w_i^{'}$ are the embedded and the extracted watermark bit respectively. Lower value of BER means that we have a higher watermarking robustness. In the following, we discuss the attacks considered in this paper.

Deletion Attacks : Let us consider an attack that consists of a random deletion of attribute values or tuples in the database. We distinguish two cases for this attack: **(i) Column deletion:** in this case, an attacker tries to delete N_{c_1} columns in the database. No matter how many columns are deleted, one column is enough to detect the watermark if all columns are watermarked. **(ii) Tuple deletion:** if the attacker randomly eliminates N_d tuples in the database. The watermark may not be detected depending on the percentage of deleted data and the group to which deleted elements belongs. We will come back to this case in Sect. 5, where we demonstrated that the robustness of our solution against this attack using BER.

Insertion Attacks: An attacker may try to insert a certain number of columns or tuples in the database. Two cases differ. **(i) Column insertion:** An attacker tries to insert a certain number of columns in the database. By doing so, it requires an attacker to duplicate at least one time the number of columns (or individuals) so as to change the watermark bit. Assume that the original group verifies the probability to have 1 values is greater than the probability to have 0 values ($Pr_1 > Pr_0$). Then, the watermarked group verifies $Pr_0^w > Pr_1^w$. Hence, we can define *X* = $Pr_1 - Pr_0$ and $X^w = Pr_0^w - Pr_1^w$. There are three cases in which the data can be added by an attacker.

- **Case 1:** If $Pr_1 < Pr_0$, there is no problem as the attack will be always detected.
- **Case 2:** If $Pr_1 = Pr_0$, as in the previous case, the attack will always be detected.
- **Case 3:** If $Pr_1 > Pr_0$, the attacker requires to add *M* elements in the database as:

$$M = \frac{N_c \times X^w}{X} \tag{10}$$

(ii) Tuple Insertion: This attack corresponds to the insertion a certain number of tuples in the database. If *N* is the number of tuples that the attacker want to insert in the database. Let *k* be the number of success out of the total number of trials and *p* the probability to succeed, while *q* is the probability of failure. Thus, we have

$$p = \frac{1}{2 \times N_g} \quad , \quad q = 1 - p \tag{11}$$

The probability of *k* successes out of *N* trials when the probability of one success is *p* is computed according to the Eq. (12)

$$P(N, k, p) = \binom{N}{k} p^k q^{N-k} \tag{12}$$

In the previous Eq. (12), the binomial coefficient express the number of combinations of N takes k. It is calculated according to Eq. (13).

$$\binom{N}{k} = \frac{N!}{(N-k)!k!} \tag{13}$$

We give in next section, obtained results after simulating the discussed attacks.

5 Experimental Results and Discussion

We evaluate our watermarking method in terms of distortion, robustness, and insertion capacity in the case of a real genetic database.

5.1 Test Database

We used a genetic relational database composed of one table of 80 tuples issued from a real genetic database, from the FrEx project [29], that contains pieces of information related to genetic variants of 733 individuals. Such genetic variants are used by researchers or/and geneticists in GWAS [3] in order to determine if there is a relationship between these genetic variants and certain diseases. For each individual and each variant, the genotype corresponds to an integer value that takes the value 0 if the alternative allele is equal to the reference allele, 1 to the second alternative allele, and $k \in \{1, \cdots, g\}$ in case of g possible alternative alleles. In the sequel, a set of attributes composed by the chromosome, the position, and the gene is considered as the primary key. We chose these attributes because their combination uniquely identifies each database tuple of variants.

5.2 Distortion Results

To test the impact of the proposed watermarking scheme for GWAS results, we have conducted a secure WSS method presented in [30]. In this context, the p-value has been used as a descriptive statistic, which is the p-value of the association, and the null hypothesis is that the allele frequencies at SNP are the same in cases and controls (For more details about the computation of the WSS p-value, please refer to [8]). To test our watermarking method, the database is divided into N_g groups, considering several cases. These cases correspond to $N_g \in \{1, \cdots, 20\}$. We have also chosen different values of distortion step Δ such that $\Delta \in \{2, 4, 6, \cdots, 34\}$, each group is also divided into two sub-groups. In order to check if our experiments are going to be different in the results obtained from the experimental and control groups, Fig. 3 presents the p-values with different percentages of modulated data after applying our watermarking method on the given database. The obtained results in Fig. 3 set up such that they conveyed a meaning that there exists no distinction between the different samples and no interference in the association test results. Moreover, the results show us that the differences are real and not just due to chance as the p-value increases as the ratio of data distortion increases. In addition, the Table 3 presents the p-value results for above chosen N_g and Δ. The mentioned p-value are very low even with variable number of groups N_g and quantization step Δ. These results confirm their statistically significant and are less likely to be caused by noise.

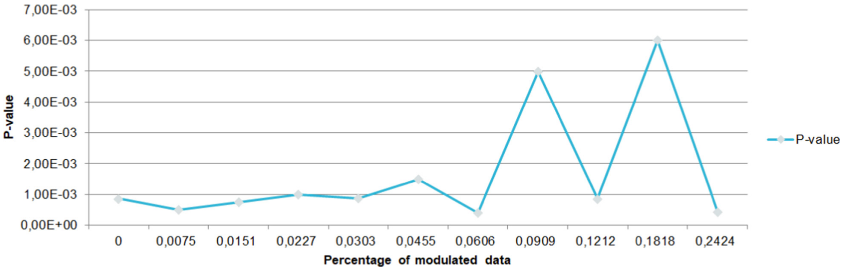


Fig. 3. Distortion percentage of modulated data.

Table 3. P-value results in function of the quantization step Δ and the number of groups N_g.

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	4.9×10^{-4}	3.5×10^{-4}	4.2×10^{-4}	4.4×10^{-3}	5.9×10^{-4}	2.9×10^{-3}	3.0×10^{-4}	5.9×10^{-4}	5.3×10^{-4}	5.4×10^{-4}	3.4×10^{-3}	7.9×10^{-3}	5.9×10^{-4}	3.9×10^{-3}	5.9×10^{-3}	3.9×10^{-3}	3.7×10^{-4}
2	5.9×10^{-4}	4.9×10^{-4}	6.6×10^{-4}	1.4×10^{-3}	4.2×10^{-4}	5.9×10^{-3}	5.9×10^{-4}	3.6×10^{-4}	9.9×10^{-4}	2.9×10^{-3}	3.8×10^{-4}	5.9×10^{-3}	2.9×10^{-3}	3.4×10^{-3}	2.3×10^{-3}	5.4×10^{-3}	9.9×10^{-4}
3	4.2×10^{-4}	1.9×10^{-3}	1.1×10^{-3}	4.9×10^{-4}	6.9×10^{-4}	1.1×10^{-3}	7.4×10^{-4}	1.1×10^{-3}	4.2×10^{-4}	5.8×10^{-4}	4.9×10^{-4}						
4	6.6×10^{-4}	6.6×10^{-4}	3.9×10^{-3}	2.9×10^{-3}	5.2×10^{-4}	5.4×10^{-4}	5.9×10^{-4}	7.7×10^{-4}	2.4×10^{-4}								
5	5.4×10^{-4}	7.4×10^{-4}	2.3×10^{-4}	8.5×10^{-4}	6.6×10^{-4}	2.9×10^{-3}	4.2×10^{-4}	8.7×10^{-4}									
6	5.9×10^{-4}	5.4×10^{-4}	9.9×10^{-4}	2.9×10^{-3}	2.9×10^{-3}	2.3×10^{-3}											
7	5.9×10^{-4}	6.6×10^{-4}	5.4×10^{-4}	1.3×10^{-3}	6.6×10^{-4}												
8	5.4×10^{-4}	3.7×10^{-4}	4.6×10^{-4}	6.6×10^{-4}													
9	1.1×10^{-3}	4.4×10^{-3}	6.6×10^{-4}														
10	3.3×10^{-4}	1.9×10^{-3}	2.3×10^{-4}														
11	2.3×10^{-3}	3.5×10^{-4}															
12	7.7×10^{-4}	2.3×10^{-3}															
13	9.9×10^{-4}	4.2×10^{-4}															
14	2.6×10^{-4}	5.9×10^{-4}															
15	1.4×10^{-3}	8.9×10^{-3}															
16	7.4×10^{-4}																
17	4.2×10^{-4}																
18	1.9×10^{-3}																
19	2.9×10^{-3}																
20	4.2×10^{-4}																

5.3 Capacity Results

The insertion capacity is evaluated by the ratio of database elements that can be used for the watermark embedding to the total number of elements in the database. Higher watermarking capacity means that more watermark information that we can embed in the database. The watermarking capacity of our solution depends on the number of embeddable groups that we have in the database. This capacity can reach 100% depending on genotypes that we have in the database. This means that in some cases, each group in the database can embed a watermark bit. However, if the capacity is the maximum, the robustness is reduced.

5.4 Robustness Results

We have simulated, different attacks on our watermarked database. We have considered an attacker that tries to insert, delete 10%, 20% and 30% of the data in the database. Obtained results are presented in Tables 4, 5, 6, 7, 8, 9. From these results, the watermark can be correctly detected from the database when BER approaches zero.

Table 4. BER results against column deletion 10%

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0.48	0.12	0.63	0.37	0.24
3	0	0	0	0	0	0	0	0	0.47	0.27	0.25	0.43					
4	0	0	0	0	0	0.12	0.24	0.48	0.43								
5	0	0	0	0	0.09	0.15	0.47	0.39									
6	0	0	0	0.09	0.23	0.31											
7	0	0	0.04	0.09	0.53												
8	0	0	0.09	0.24													
9	0	0	0.16														
10	0	0															
11	0	0															
12	0	0.02															
13	0	0.12															
14	0	0.17															
15	0																
16	0																
17	0																
18	0.06																
19	0.06																
20	0.05																

Table 5. BER results against column deletion 20%

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0.48	0.12	0.44	0.37	0.24
3	0	0	0	0	0	0	0	0	0.47	0.27	0.25	0.43					
4	0	0	0	0	0	0.12	0.24	0.48	0.43								
5	0	0	0	0	0.13	0.15	0.47	0.39									
6	0	0	0	0.09	0.23	0.31											
7	0	0	0.04	0.09													
8	0	0	0.09														
9	0	0															
10	0	0															
11	0	0															
12	0	0.05															
13	0	0.12															
14	0	0.15															
15	0.04																
16	0																
17	0																
18	0.05																
19	0.06																
20	0.05																

Table 6. BER results against column deletion 30%

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0.48	0.12	0.56	0.37	0.24
3	0	0	0	0	0	0	0	0.15	0.47	0.27	0.25	0.43					
4	0	0	0	0	0	0.12	0.24	0.48	0.43								
5	0	0	0	0	0.13	0.15	0.47										
6	0	0	0	0.09	0.23	0.31											
7	0	0	0.04	0.09	0.53												
8	0	0	0.09	0.24													
9	0	0	0.1														
10	0	0															
11	0	0															
12	0	0.04															
13	0	0.12															
14	0	0.17															
15	0.04																
16	0																
17	0																
18	0.02																
19	0.06																
20	0.05																

Table 7. BER results against column addition 10%

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0.48	0.12	0.69	0.37	0.24
3	0	0	0	0	0	0	0	0.19	0.47	0.27	0.25	0.43					
4	0	0	0	0	0	0.12	0.36	0.48	0.43								
5	0	0	0	0	0.14	0.15	0.47										
6	0	0	0.04	0.09	0.23	0.31											
7	0	0	0.09	0.24	0.53												
8	0	0	0.19	0.24													
9	0	0	0.20														
10	0	0															
11	0	0															
12	0	0.10															
13	0	0.12															
14	0	0.17															
15	0.04																
16	0																
17	0																
18	0.06																
19	0.06																
20	0.05																

Table 8. BER results against column addition 20%

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0.40	0.48	0.12	0.69	0.37	0.24
3	0	0	0	0	0	0	0	0.51	0.47	0.27	0.45	0.43					
4	0	0	0	0	0	0.12	0.36	0.48	0.43								
5	0	0	0	0.05	0.14	0.24	0.47	0.39									
6	0	0	0	0.09	0.24												
7	0	0	0.04	0.24	0.53												
8	0	0	0.19	0.24													
9	0	0	0.25														
10	0	0															
11	0	0															
12	0	0.12															
13	0	0.14															
14	0	0.17															
15	0																
16	0																
17	0																
18	0.06																
19	0.06																
20	0.05																

Table 9. BER results against column addition 30%

$\frac{\Delta}{N_g}$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0.40	0.48	0.55	0.69	0.37	0.24
3	0	0	0	0	0	0	0.11	0.51	0.47	0.27	0.45	0.43					
4	0	0	0	0	0	0.11	0.37	0.50	0.64								
5	0	0	0	0.05	0.35	0.25	0.74	0.39									
6	0	0	0	0.09	0.24												
7	0	0	0.04														
8	0	0	0.19														
9	0	0	0.22														
10	0	0															
11	0	0															
12	0	0.12															
13	0	0.14															
14	0																
15	0																
16	0																
17	0																
18	0.07																
19	0.07																
20	0.08																

6 Conclusion

In this paper, we have proposed a new robust database watermarking method that allows watermarking of genomic data used in GWAS. It is the first method of this kind, and it can be used for statistical algorithms such as the WSS method. It can be used in protecting traitor tracing and copyright protection, and it is based on QIM and majority vote. We have studied theoretical performance and experimentally verified the performance of our solution in terms of robustness against deletion and addition attacks, insertion capacity, and distortion. In this method, a watermark is embedded in genetic data without altering the results of association tests that can be conducted on these data. This comfort its future use in real-life applications, especially in cloud environments. As the primary form of the proposed modulation technique is to preserving the statistical analysis of GWA studies, we plan to study the proposed technique for other GWA studies in the future.

References

1. Mehrgou, A., Akouchekian, M.: The importance of BRCA1 and BRCA2 genes mutations in breast cancer development. Med. J. Islamic Repub. Iran (MJIRI) **30**(369), 1–12 (2016)
2. Ginsburg, G.: Medical genomics: gather and use genetic data in health care. Nat. News **508**(7497), 451–453 (2014)
3. Wang, M.H., Cordell, H.J., Van Steen, K.:Statistical methods for genome-wide association studies. In: Seminars in Cancer Biology, vol. 55, pp. 53–60. Elsevier (2019)
4. Taleb, A., Kirchler, M., Monti, R., Lippert, C.: ContiG: self-supervised multimodal contrastive learning for medical imaging with genetics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20 908–20 921. IEEE (2022)
5. Michael, B.E., Yann, L.G., Sarah, E.J., Napolioni, V., Michael, G.D., Zihuai, H.: A fast and robust strategy to remove variant-level artifacts in alzheimer disease sequencing project data. Neurol. Genet. **8**(5), e200012 (2022)
6. Shin, J., et al.: PhenGenVar: a user-friendly genetic variant detection and visualization tool for precision medicine. J. Personalized Med. **12**(6), 1–11 (2022)
7. Ozaki, K., et al.: Functional SNPs in the lymphotoxin-α gene that are associated with susceptibility to myocardial infarction. Nat. Genet. **32**(4), 650–654 (2002)
8. Madsen, B.E., Browning, S.R.: A groupwise association test for rare mutations using a weighted sum statistic. PLoS Genet. **5**(2), 1–11 (2009)
9. Ding, H., Tian, Y., Peng, C., Zhang, Y., Xiang, S.: Inference attacks on genomic privacy with an improved HMM and an RCNN model for unrelated individuals. Inf. Sci. **512**, 207–218 (2020)
10. Bellafqira, R., Coatrieux, G., Genin, E., Cozic, M.: Secure multilayer perceptron based on homomorphic encryption. In: Yoo, C.D., Shi, Y.-Q., Kim, H.J., Piva, A., Kim, G. (eds.) IWDW 2018. LNCS, vol. 11378, pp. 322–336. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11389-6_24
11. Rady, M., Abdelkader, T., Ismail, R.: Integrity and confidentiality in cloud outsourced data. Ain Shams Eng. J. **10**, 275–285 (2019)
12. Wang, X., Jiang, X., Vaidya, J.: Efficient verification for outsourced genome-wide association studies. J. Biomed. Inform. **117**, 103714 (2021)

13. Wang, J., Du, X., Lu, J., Lu, W.: Bucket-based authentication for outsourced databases. Concurrency Comput. Pract. Experience **22**(9), 1160–1180 (2010)
14. Niyitegeka, David, Coatrieux, Gouenou, Bellafqira, Reda, Genin, Emmanuelle, Franco-Contreras, Javier: Dynamic watermarking-based integrity protection of homomorphically encrypted databases – application to outsourced genetic data. In: Yoo, Chang D.., Shi, Yun-Qing., Kim, Hyoung Joong, Piva, Alessandro, Kim, Gwangsu (eds.) IWDW 2018. LNCS, vol. 11378, pp. 151–166. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11389-6_12
15. Boujdad, F.-Z., Niyitegeka, D., Bellafqira, R., Coatrieux, G., Génin, E., Südholt, M.S.: A hybrid cloud deployment architecture for privacy-preserving collaborative genome-wide association studies. In: Gladyshev, P., Goel, S., James, J., Markowsky, G., Johnson, D. (eds.) ICDFC 2021. LNICST, vol. 441, pp. 342–359. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06365-7_21
16. Chen, W.: An artificial chromosome for data storage. Nat. Sci. Rev. **8**(5), nwab028 (2021)
17. Nguyen, T.T., Cai, K., Song, W., Immink, K.A.S.: Optimal single chromosome-inversion correcting codes for data storage in live DNA. In: IEEE International Symposium on Information Theory (ISIT), pp. 1791–1796. IEEE (2022)
18. Vinodhini, R., Malathi, P.: Hiding information in the DNA sequence using DNA steganographic algorithms with double-layered security. Int. J. Inf. Secur. Priv. (IJISP) **16**(1), 1–20 (2022)
19. Wang, Y., Han, Q., Cui, G., Sun, J.: Hiding messages based on DNA sequence and recombinant DNA technique. IEEE Trans. Nanotechnol. **18**, 299–307 (2019)
20. Lee, S.-H.: Reversible data hiding for DNA sequence using multilevel histogram shifting. Secur. Commun. Netw. **2018**, 1–13 (2018)
21. Hamad, S., Elhadad, A., Khalifa, A.: DNA watermarking using codon postfix technique. IEEE/ACM Trans. Comput. Biol. Bioinf. **15**(5), 1605–1610 (2017)
22. Ayday, E., Yilmaz, E., Yilmaz, A.: Robust optimization-based watermarking scheme for sequential data. In: 22^{nd} International Symposium on Research in Attacks, Intrusions and Defenses, pp. 323–336 (2019)
23. Kuribayashi, M., Fukushima, T., Funabiki, N.: Robust and secure data hiding for PDF text document. IEICE Trans. Inf. Syst. **102**(1), 41–47 (2019)
24. Pabinger, S., et al.: A survey of tools for variant analysis of next-generation genome sequencing data. Brief. Bioinform. **15**(2), 256–278 (2014)
25. Danecek, P.: The variant call format and VCF tools. Bioinformatics **27**(15), 2156–2158 (2011)
26. Rani, S., Halder, R.: Comparative analysis of relational database watermarking techniques: an empirical study. IEEE Access **10**, 27970–27989 (2022)
27. Li, Y., Guo, H., Jajodia, S.: Tamper detection and localization for categorical data using fragile watermarks. In: Proceedings of the 4^{th} ACM Workshop on Digital Rights Management, pp. 73–82 (2004)
28. Chen, B., Wornell, G.W.: Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans. Inf. Theory **47**(4), 1423–1443 (2001)
29. Genin, E., Redon, R., Deleuze, J.-F., Campion, D., Lambert, J.-C., Dartigues, J.-F.: The French exome (FREX) project: a population-based panel of exomes to help filter out common local variants. Int. Genet. Epidemiol. Soc. **41**, 691 (2017)
30. Bellafqira, R., Ludwig, T.E., Niyitegeka, D., Génin, E., Coatrieux, G.: Privacy-preserving genome-wide association study for rare mutations-a secure framework for externalized statistical analysis. IEEE Access **8**, 112515–112529 (2020)