Robust Optimization-Based Watermarking Scheme for Sequential Data

Erman Ayday
Case Western Reserve University
Cleveland, OH, USA
and
Bilkent University, Turkey

Emre Yilmaz
Case Western Reserve University
Cleveland, OH, USA

Arif Yilmaz
Bilkent University, Turkey

Abstract

In this work, we address the liability issues that may arise due to unauthorized sharing of personal data. We consider a scenario in which an individual shares their sequential data (such as genomic data or location patterns) with several service providers (SPs). In such a scenario, if their data is shared with other third parties without their consent, the individual wants to determine the service provider that is responsible for this unauthorized sharing. To provide this functionality, we propose a novel optimization-based watermarking scheme for sharing of sequential data. The proposed scheme guarantees with a high probability that (i) the malicious SP that receives the data cannot understand the watermarked data points, (ii) when more than one malicious SPs aggregate their data, they still cannot determine the watermarked data points, (iii) even if the unauthorized sharing involves only a portion of the original data or modified data (to damage the watermark), the corresponding malicious SP can be kept responsible for the leakage, and (iv) the added watermark is compliant with the nature of the corresponding data. That is, if there are inherent correlations in the data, the added watermark still preserves such correlations. The proposed scheme also minimizes the utility loss due to changing certain parts of the data while it provides the aforementioned security guarantees. Furthermore, we conduct a case study of the proposed scheme on genomic data and show the security and utility guarantees of the proposed scheme.

1 Introduction

Sequential data includes time-series data such as location patterns, stock market data, speech, or ordered data such as genomic data. Individuals share different types of sequential data for several purposes, typically to receive personalized services from online service providers (SPs). Data collected and processed by these SPs may reveal privacy sensitive information about individuals. Thus, the way these SPs handle the collected data poses a threat to individuals' privacy and it is crucial for individuals to have control on how their data is collected and handled by the SPs.

When an individual shares their personal data with an SP for a particular purpose, they want to make sure that their data will not be observed by other third parties. Privacy leakage occurs when personal data of individuals is further shared by an SP with other third parties (e.g., for financial benefit). To deter the SPs from such unauthorized sharing, it is required to develop technical solutions that would keep them liable for such unauthorized sharing (e.g., by connecting the unauthorized sharing to its source). One well-known tool for such scenarios is watermarking. An individual may add a unique watermark into their data before sharing it with each SP, and if their data is further shared without their authorization, they can associate the unauthorized sharing to the corresponding SP.

Watermarking is a well-known technique to address the liability issues especially for multimedia data [18]. Using the high amount of redundancy in the data and the fact that human eye cannot differentiate slight differences between the pixel values, watermark is inserted into multimedia data by changing some pixel values. However, watermarking is not a straightforward technique for sequential data such as location patterns or genomic data. To insert watermark into sequential data, original data should be modified according to the watermark which reduces the quality of service provided by the SPs. Thus, watermarking sequential data while preserving data utility has unique challenges.

Another challenge for watermarking sequential data is the identifiability (or robustness) of the watermark. An individual cannot identify the SP that is responsible for the data leakage if the SP finds the watermark inserted data points and removes (or tampers with) the watermark before the unauthorized sharing. Furthermore, an SP may partially share the data (rather than sharing the whole data of the individual) or modify the data (to damage the watermark). These make it harder for the individual to identify the source of the leakage.

An SP may utilize different types of auxiliary information in order to determine (and hence tamper with) the watermark in the data. The most common type of such auxiliary information may be the inherent correlations in the data. Location

patterns are correlated in both time and space [9]. Similarly, genomic data carries inherent correlations (referred as linkage disequilibrium) inside. A malicious SP may also use external auxiliary information that is correlated with the data (e.g., phenotype or kinship information for genomic data or interdependent check-in information for location data). Thus, an SP can identify the watermarked data points by identifying the points that violate the expected correlations in the data. One other type of auxiliary information is the data shared by the individual with other SPs. Multiple SPs may collect the same sequential data from the same individual (with different watermarks patterns) and they may compare their collected data in order to identify the watermarked points with higher probability. Thus, a watermarking algorithm for sequential data should be robust against these kinds of threats.

To address these robustness and utility challenges, we propose a novel watermarking scheme to share sequential data. Initially, we assume the data has no correlations and propose an algorithm to determine the data points to be watermarked by solving a non-linear optimization problem. This algorithm is developed to be robust against collusion of malicious SPs. Then, we explain how to deal with correlated data in the proposed algorithm. Hence, we minimize the risk of correlation attacks by malicious SPs who know the pairwise correlations between data points. We evaluate the security (robustness) and the utility of the proposed algorithm on a genomic dataset. The main motivations to choose genomic data sharing as the use case are as follows: (i) genomic data includes privacy-sensitive information such as predisposition to diseases [19], (ii) it is not revokable, and hence it is crucial to make sure that it is not leaked, and (iii) it has inherent correlations that makes watermarking even more challenging.

The main contributions of the proposed work are summarized as follows:

- We propose a novel collusion-secure watermarking scheme for sequential data. The proposed scheme minimizes the probability for the identifiability of the watermark by the SPs. We show that even when multiple SPs join their data together or they use the knowledge of inherent correlations in the data the watermark cannot be identified (with a high probability).
- We show that the SPs that are responsible for the unauthorized sharing can be detected with a high probability even when they share a portion of the data or when they modify the data in order to damage the watermark. We also show relationship between the probabilistic limits of this detection and the shared portion of data.
- While providing these security (or robustness) guarantees, the proposed system also minimizes the utility loss in the sequential data due to watermarking.
- We also implement and evaluate the proposed scheme for genomic data sharing.

The rest of the paper is organized as follows. In the next section, we discuss the related work. In Section 3, we introduce the data model, the system model, and the threat model. In Section 4, we provide the details of the proposed solution. In Section 5, we evaluate the security of the proposed watermarking algorithm. In Section 6, we discuss potential extensions of the proposed scheme and possible future research directions. Finally, in Section 7, we conclude the paper.

2 Related Work

Digital watermarking is the act of hiding a message related to a digital signal (e.g., an image, song, or video) within the signal itself [7]. While digital watermarks are typically used for copyright and copy protection [4, 6, 17, 18], they are also used in different applications such as broadcast monitoring [15], transaction tracking [8], and content authentication [27]. Digital watermarks are prominently used for copy protection and copy deterrence on multimedia content. Multimedia watermarking schemes [14] benefit from the high redundancy in the data and they do not consider sophisticated attacks against the watermarking scheme. Since the amount of redundancy is not typically high in non-media data, it is more challenging to add watermark into such data. The watermarking techniques proposed for non-media such as text [12] and graphs [26] cannot be applied to sequential data because they do not consider robustness of the watermark against various types of attacks (that are discussed in Section 3.3).

Several works proposed watermarking techniques for sequential data such as time-series data and spatiotemporal data. Kozat et al. proposed a technique for hiding sensitive metadata such as SSN or date-of-birth into electrocardiograms (ECG) in order to authenticate the ownership of data without distorting important ECG characteristics [13]. In addition, Panah et al. [24] introduced a low complexity watermarking scheme for tamper-proofing of ECG signals at the sensory nodes. Watermarking spatiotemporal data is also challenging due to low redundancy of data and works in this area are mostly focused on watermarking trajectory datasets that include trajectories of multiple objects [11, 16, 25]. However, we consider the case in which an individual wants to share their individual data with multiple SPs after watermarking. This is a more challenging problem since the redundancy in the shared data is much lower. In general, neither of these schemes consider the correlations in data nor the possibility of colluding SPs.

Boneh and Shaw proposed a general fingerprinting (watermarking) solution that is robust against collusion [5]. Their scheme constructs fingerprints in such a way that no coalition of attackers can find a fingerprint. However, there are still some practical drawbacks of this scheme. First, fingerprint length may be very long to guarantee robustness against collusion, which reduce the utility of the data. Furthermore, the scheme does not consider complex attacks against the watermarking algorithms such as the ones using auxiliary

$x_1, \cdots, x_\ell$	Set of ordered data points
$d_1, \cdots, d_m$	Possible values (states) of a data point
I_i	Index set of the data points that are shared with the SP i
D_{I_i}	Set of data points in I_i
W_{I_i}	Set of data points in I_i after watermarking
Z_{I_i}	Set of watermarked data points in W_{I_i}

Table 1: Frequently used symbols and notations.

information or the ones tampering the watermark and it does not consider the correlations in the data. We address these drawbacks in our proposed scheme.

3 Problem Definition

Here, we describe the data model, system model, and the threat model. Frequently used symbols and notations are presented in Table 1.

3.1 Data Model

Sequential data consists of ordered data points $x_1, ..., x_\ell$, where ℓ is the length of the data. The value of a data point x_i can be in different states from the set $\{d_1, \cdots, d_m\}$ according to the type of the data. For instance, x_i can be coordinate pairs in terms of latitude and longitude for location data, it can be location semantics (e.g., cafe or restaurant) for check-in data, or it can be the value of a nucleotide or point mutation for genomic data.

We approach the problem for two general sequential data types: (i) sequential data with no correlations in which data points are independent and identically distributed. In this type, value of a data point cannot be predicted using the values of other data points. Sparse check-in data might be a good example for this type. And, (ii) sequential data with correlations between the data points. Correlation between data points may vary based on the type of data. For example, consecutive data points that are collected with small differences in time may be correlated in location patterns. That is, an individual's location at time t can be estimated if their locations at time $(t-1)$ and/or $(t+1)$ are known. In genomic data, point mutations (e.g., single nucleotide polymorphisms or SNPs[1]) may have pairwise correlations between each other. Such pairwise correlations are referred as linkage disequilibrium [23] and they are not necessarily between consecutive data points. The correlation value may differ based on the state of each data point and correlation between the data points is typically asymmetric. Furthermore, it has been shown that correlations in human genome can also be of higher order [22]. For the clarity of the presentation, we first build our solution for uncorrelated sequential data and then extend it for correlated data.

[1] We provide a brief background on genomics in Section 5.

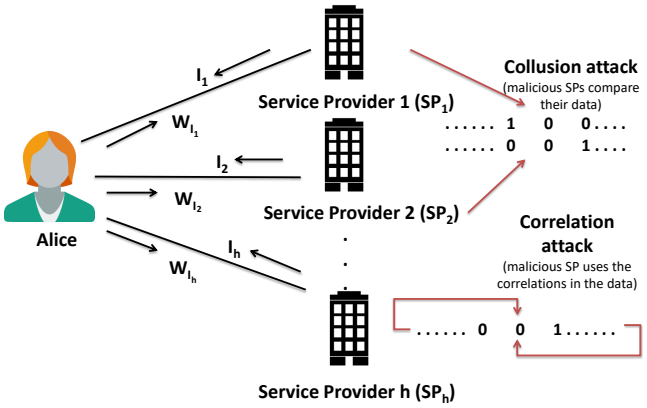


Figure 1: Overview of the system and threat models.

3.2 System Model

We consider a system between a data owner (Alice) and multiple service providers (SPs) as shown in Figure 1. For genomic data, the SP can be a medical institution, a genetic researcher, or direct-to-customer service provider. For location data, the SP can be any location-based service provider. In the description of the scheme, for clarity, we explain some parts of the algorithm on binary data but the proposed scheme can be extended for non-binary data. In fact, for the evaluation of the proposed scheme, we focus on the point mutations in genomic data that may have values from $\{0,1,2\}$. Alice shares parts of her data with the SPs to receive different types of services. Note that the part Alice shares with each SP may be different and we do not need same data to be shared with each SP. When we talk about the collusion attack (as will be detailed in the next section), we consider the intersection of the data parts owned by all malicious SPs.

On one hand, when Alice shares her data with an SP, she wants to make sure that her data will not be shared with other third parties by the corresponding SP. In the case of further unauthorized sharing, she wants to know the SP that is responsible from this leak. Therefore, whenever Alice shares her data with a different SP, she inserts a unique watermark into it. On the other hand, an SP may share Alice's data with third parties without the consent of Alice. While doing so, to avoid being detected, the SP wants to detect and remove the watermark from the data. Instead of sharing the whole data with a third party, an SP may also share a certain portion of Alice's data to reduce the risk of detection (but compromising from the shared data amount). Similarly, malicious SP (or SPs) may try to damage the watermark by modifying the data. Furthermore, two or more SPs may join their data to detect the watermarked points. Security of the watermarking scheme increases (against the attacks discussed in the next section) as the length of the watermark increases. However, a long watermark causes significant modification on the original data, and hence decreases the utility of the shared data. In our proposed scheme, utility loss in the data is minimized while

the watermarking scheme is still robust against the potential attacks with high probability.

3.3 Threat Model

Different types of attacks are defined for image or text watermarking such as elimination attack, collusion attack, masking attack, insertion/deletion attack, and reordering attack [12,21]. We consider the following attacks on the proposed watermarking scheme by adapting previously defined attacks to sequential data and defining new attacks for sequential data such as correlation attack.

Single SP attack on uncorrelated data. Assume that Alice shares her (uncorrelated) sequential data of length ℓ with an SP and she includes a watermark of length w into this data. Since data is uncorrelated, each data point is independent from other, and hence for each data point, the SP infers the probability of being watermarked as w/ℓ. Instead of trying to detect the watermark, the malicious SP may also tamper with the data in order to damage the watermark.

Correlation attack. If an SP has correlated data points and it also knows the corresponding correlation values, it may identify the watermarked points with higher probability. To be general, we assume pairwise, asymmetric correlations between different states of data points. The proposed scheme can be extended to other scenarios (e.g., higher order correlations or symmetric correlations) similarly. For instance, if d_α state of x_i (i.e., $x_i = d_\alpha$) is correlated with d_β state of x_j (i.e., $x_j = d_\beta$), then $Pr(x_i = d_\alpha | x_j = d_\beta)$ is high, but the opposite does not need to hold. Note that d_α state of x_i may be in pairwise correlation with other data points as well. We consider all possible pairwise correlations between different states of all data points in our analysis. Following this example, assume the SP has one of the correlated data points as $x_j = d_\beta$, but $x_i = d_\gamma$ (where $d_\gamma \neq d_\alpha$). Then, the SP can conclude that x_i is watermarked with probability $p(x_i^w) = Pr(x_i = d_\alpha | x_j = d_\beta)$. If d_α state of x_i is also correlated with other data points (that the SP can observe), then the SP computes the watermark probability on x_i as the maximum of these probabilities. Similarly, d_γ state of x_i may also be correlated with other data points. Since $x_i = d_\gamma$, such correlations imply that data point x_i is not watermarked. Using such correlations, the SP also computes the probability that x_i is not watermarked, $p(x_i^n)$. Eventually, the SP computes the probability of data point x_i being watermarked as $(p(x_i^w) - p(x_i^n))$. Once the SP determines the probability of being watermarked for each data point, it sorts them based on the computed probabilities, and identifies the w watermarked data points as the ones with the highest probabilities. We assume that the SP knows the watermarking algorithm, and hence the length of the watermark (w). Thus, the SP may choose w data points corresponding to the w highest probabilities to infer the watermarked data points in the shared data.

Collusion attack. Multiple SPs that receive the same data portion (from the same data owner) with different watermark patterns may join their data to identify the watermarked points with higher probability. In such a scenario, when the SPs vertically align their data points, they will observe some data points with different states. Such data points will definitely be marked as watermarked data points by the SPs. Collusion attack may also benefit from the correlation attack. Malicious SPs may first run the collusion attack to identify some watermarked points and then they may individually run the correlation attack to infer the watermark pattern. We also evaluate the robustness of the proposed scheme against such an attack. Malicious SPs may also try to modify the data in order to damage the watermark.

3.4 Watermark Robustness

"Robustness" and "security" terms have been used interchangeably for watermarking schemes in different works. For text watermarking, robustness of a watermarking scheme is defined as strength of the technique to resist attacks that aim to retrieve or modify hidden data [12]. For image watermarking, Nyeem et al. define robustness as the ability to withstand any distortions and they define security as the ability to resist any hostile attacks that try to circumvent the system or to destroy the watermark's purpose(s) [20]. Same authors formalize the robustness for image watermarking by defining three levels of robustness such as robust, fragile, and semi-fragile by considering detection ability after distortion [21]. Adelsbach et al. provide formal definitions for watermark robustness [3]. Different from our work, in [3], authors consider watermarking mechanisms that use a secret embedding key (that is used when adding watermark to the data). They define watermark robustness as the information of the watermark that is revealed to the adversary and watermark security as the information revealed about the secret embedding key. Inspired from [3], we come up with the following robustness definitions for watermarking sequential data.

Robustness against watermark inference. This property states that watermark should not be inferred by the malicious SP (or SPs) via the aforementioned attack models. In the proposed scheme, inferring the watermark does not rely on a computationally hard problem; malicious SP (or SPs) probabilistically infer the watermark. Thus, we evaluate the proposed scheme for this property in terms of malicious SPs' (or SP's) inference probability for the added watermark. We provide the following definition to evaluate the robustness of a watermarking scheme against watermark inference.

Definition 1 ***p-robustness against f-watermark inference.*** *A watermarking scheme is p-robust against f-watermark inference if probability of inferring at least f fraction of the watermark pattern ($0 \leq f \leq 1$) is smaller than p.*

Robustness against watermark modification. This property states that the malicious SP (or SPs) should not be able to modify the watermark in such a way that the watermark detection algorithm of the data owner misclassifies the source

of the unauthorized data leakage. We evaluate the proposed scheme for this attribute in terms of precision and recall of the data owner to detect the malicious SP (or SPs) that leak their data. For this, we define "false positive" as watermark detection algorithm classifying a non-malicious SP as a malicious one and "false negative" as watermark detection algorithm classifying a malicious SP as a non-malicious one. We provide the following definition to evaluate the robustness of a watermarking scheme against watermark modification.

Definition 2 ***ρ/ζ-robustness against watermark modification.*** *A watermarking scheme is ρ/ζ-robust against watermark modification if malicious SP (or SPs), by modifying the watermark, cannot decrease the precision and recall of the watermark detection algorithm below ρ and ζ, respectively.*

For all the aforementioned attack models, we evaluate the proposed watermarking scheme based on its robustness. In Section 5, we show the limits of the proposed scheme for these definitions considering different variables.

4 Proposed Solution

Here, first we present an overview of the proposed protocol and then describe the details of the proposed watermarking algorithm. When Alice wants to share her data with an SP i, they engage in the following protocol. The SP i sends the indices of Alice's data it requests, denoted by I_i. Alice generates $D_{I_i} = \bigcup_{i \in I_i} x_i$. Alice finds the data points to be watermarked considering her previous sharings of her data. This part is done using our proposed watermarking algorithm as described in detail in this section. Alice inserts watermark into the data points in D_{I_i} and generates the watermarked data W_{I_i}. Alice stores the ID of the SP and Z_{I_i} (watermark pattern for the SP i). Alice sends W_{I_i} to SP i.

The proposed watermarking algorithm describes the selection of data points to be watermarked in the sequential data so that the watermark will be secure against the attacks discussed in Section 3.3. We insert watermark into a data point by changing this data point's state. For instance, if data is binary, this change is from 0 to 1, or vice versa. If each data point can have states from the set $\{d_1, \cdots, d_m\}$, the change is from the current state to some other state d_j^*. In Section 4.1, since we assume there is no correlation in data, a data point x_i is changed to a state d_j^* uniformly at random. However, due to the correlation in data, the new state d_j^* of a data point x_i is determined to minimize the probability of correlation attack in Section 4.2. In the following, we first detail our solution for sequential data that has no correlations (data points are independent from each other) and then, we describe how to extend our solution for correlated sequential data.

4.1 Watermarking Sequential Data without Correlations

Before giving the details of the proposed algorithm, we first provide the following notations to facilitate the discussion.

- n_i^h: number of data points that are watermarked i times when the whole data is shared with h SPs.
- $\hat{y}_i^h$: number of data points that are watermarked i times when the whole data is shared h times and will not be watermarked in the $(h+1)$-th sharing.
- y_i^h: number of data points that are watermarked i times when the whole data is shared h times and will be watermarked in the $(h+1)$-th sharing.

When Alice shares her data with a new SP, first, watermark locations in the data are determined for the new request according to the watermark patterns in previously shared data and with the goal of minimizing the success of the collusion attack. From these definitions, it is obvious that $n_i^h = \hat{y}_i^h + y_i^h$, which means that among the n_i^h data points that are watermarked i times after h sharings, y_i^h of them will be shared in the $(h+1)$-th sharing and the remaining $\hat{y}_i^h$ of them will not be shared in the $(h+1)$-th sharing. Therefore, the proposed algorithm computes y_i^h and $\hat{y}_i^h$ values to minimize the probability of collusion attack when Alice shares her data with $(h+1)$-th SP. After computing these values any y_i^h of n_i^h data points that are watermarked i times can be selected to insert the watermark since data points are not correlated.

As discussed, a malicious SP may increase its probability to find the watermark inserted data points by colluding with other malicious SPs that received the same data from Alice with different watermark patterns. For simplicity, assume that each data point's state can be either 0 or 1 and h malicious SPs have the same data portion (belonging to Alice) with different watermark patterns. They vertically align their data portions, compare their data, and find the differences. For instance, for a data point x_i, they observe k 0s and $(h-k)$ 1s (where $0 \leq k \leq h$) and they conclude that the corresponding data point has been watermarked either k or $(h-k)$ times. We assume that the proposed watermarking algorithm is also known by the malicious SPs. Therefore, these h SPs may run our proposed algorithm (as discussed next) and find n_k^h and n_{h-k}^h values. Once they have these values, they may conclude that (i) the corresponding data point has been watermarked k times with probability $n_k^h/(n_k^h + n_{h-k}^h)$, and (ii) $(h-k)$ times with probability $n_{h-k}^h/(n_k^h + n_{h-k}^h)$. In our algorithm, watermarks are inserted into the watermark locations that minimizes the probability of identifying the whole watermark patterns of all malicious SPs when they collude. To do so, we propose solving an optimization problem to determine the data points to be watermarked at each data sharing instance of Alice. The objective function of this problem for the $(h+1)$-th sharing can be formulated as follows:

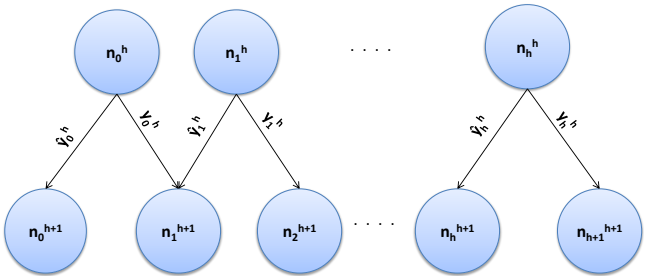


Figure 2: Relationship between n_i^h, n_i^{h+1}, y_i^h, and $\hat{y}_i^h$ values in the watermark insertion scheme.

$$min \prod_{i=0}^{h+1} \left(\frac{n_i^{h+1}}{n_i^{h+1} + n_{h-i+1}^{h+1}}\right)^{n_i^{h+1}} \quad (1)$$

with the following constraints: (i) $\sum_{i=0}^{h+1} y_i^h = w$, (ii) $n_0^{h+1} = \hat{y}_0^h$, (iii) $n_{h+1}^{h+1} = y_h^h$, (iv) $n_i^{h+1} = y_{i-1}^h + \hat{y}_i^h$ for $i = 1, ..., h$, (v) $\hat{y}_i^h + y_i^h = n_i^h$, (vi) $y_i^h, \hat{y}_i^h \geqslant 0$, and (vii) $y_0^h > 0$. Here, constraint (i) determines the number of data points that we watermark. That is, the algorithm does not modify more data points than the limit defined in this constraint. Constraints (ii), (iii), (iv), and (v) denote the relationship between n_i^h, n_i^{h+1}, y_i^h, and $\hat{y}_i^h$. In Figure 2, we show this relationship. Constraint (vi) is used to prevent negative y_i^h and $\hat{y}_i^h$ values. Finally, constraint (vii) is to make sure that each SP has a unique watermark pattern. As the solution of this optimization problem, we obtain the y_i^h and $\hat{y}_i^h$ values. As mentioned before, for the $(h+1)$-th sharing, the algorithm selects any y_i^h of the data points that are watermarked i times after h sharings.

4.2 Addressing Correlations in the Data

By solving the optimization problem in Section 4.1, we obtain the y_i^h and $\hat{y}_i^h$ values. Since this time data is correlated, watermarks should be inserted in such a way that no malicious SP can understand the watermark inserted data points by checking the validity of the correlations. To guarantee this, if a data point x_i's state is changed from d_α to d_β (due to added watermark), the states of other data points that are correlated with d_β state of x_i should be also changed (since we assume asymmetric correlations). Assume data has been shared for h times before. Watermark insertion algorithm for the $(h+1)$-th sharing of the data with SP ψ is summarized as follows.

From the solution of the optimization problem, we know the number of data points which are watermarked i times and will be watermarked in the current sharing (y_i^h). Since a data point could be watermarked between 0 and h times, we have the solution set of the optimization problem as $Y = \{y_0^h, y_1^h, \cdots, y_h^h\}$. Data points to be shared with SP ψ are $D_{I_\psi} = \{x_1, \cdots, x_\ell\}$ and the states of a data point are from the set $\{d_1, d_2, \cdots, d_m\}$. To add watermarks into data points that are watermarked for t times ($t = 0, 1, \cdots, h$) in the previous h sharings, we find the set of t times watermarked data points (T_t) and sort them in ascending order according to their presence probabilities. Presence probability can be found as follows. Assume d_j state of data point x_j is correlated with the set of data points in $C = \{x_{i_0} = d_{i_0}, \cdots, x_{i_n} = d_{i_n}\}$. Then, the presence probability for $(x_j = d_j)$ is computed as $\prod_{t=0}^{n} Pr(x_j = d_j | x_{i_t} = d_{i_t})$.

Then, for each t value from 0 to h, we get the data point with minimum presence probability (x_j) in T_t, determine the state (d_j^*) that maximizes its presence probability, and change the state of x_j. This way, we choose the most likely state value for x_j according to the whole data. If the state of x_j is already d_j^*, we skip this data point and continue with the next data point with minimum presence probability. Since we change a data point that is watermarked for t times, we also decrement the value of $Y[t]$ (y_t^h) by 1. After the state of x_j is changed to d_j^*, we find the data points that are correlated with d_j^* state of x_j. That is, we construct a set C with data points that satisfy $Pr(x_i | x_j = d_j^*) > \tau$ and change the states of the data points in C. For each data point in C, we find its "desired state" (i.e., correlated state with d_j^* state of x_j) and change it. During this process, if we change a data point that is watermarked for t^* times, we also decrement the value of $Y[t^*]$ ($y_{t^*}^h$) by 1. We continue this process until we add w watermarks to the data.

In this algorithm, we consider pairwise correlations between the data points. When correlations between the data points are more complex (e.g., higher order), we can still use a similar algorithm to handle them. We assume that malicious SPs also have the same resources we use in this algorithm to use the correlations (in order to detect the watermarked points) and evaluate the scheme accordingly in Section 5.

5 Evaluation

We implemented the proposed watermarking scheme on genomic data and evaluated its security (robustness) and utility guarantees. To solve the proposed non-linear optimization problem, we used the APMonitor Optimization Suite [10]. In this section, we provide the details of the data model we used in our evaluation and our results.

5.1 Data Model

For the evaluation, we used single-nucleotide polymorphism (SNP) data on the DNA. The human genome consists of approximately three billion letters (A, T, C, or G). Even though more than 99% of these letters are identical in any two individuals, there are differences between us due to genetic variations. SNP is the most common DNA variation in human population. A SNP is a position in the genome holding a nucleotide, which varies between individuals [2]. For example, in Figure 3, two sequenced DNA fragments from two different individuals contain a single different nucleotide at a particular SNP position. In general, there are two types of alleles (nucleotides) observed at a given SNP position: (i) the major allele is the most frequently observed nucleotide, and (ii) the minor allele is the rare nucleotide. For instance, the two alleles for the SNP position in Figure 3 are C and T (G

and A in the figure are the complementary nucleotides for C and T, respectively). Almost all common SNPs have only two alleles, and everyone inherits one allele of every SNP position from each of their parents. If an individual receives the same allele from both parents, they are said to be homozygous for that SNP position. If however, they inherit a different allele from each parent (one minor and one major), they are called heterozygous. Depending on the alleles the individual inherits from their parents, the state (or value) of a SNP position can be simply represented as the number of minor alleles it possesses, i.e., 0, 1, or 2. SNPs may have pairwise correlations between each other. Such pairwise correlations are referred as linkage disequilibrium [23].

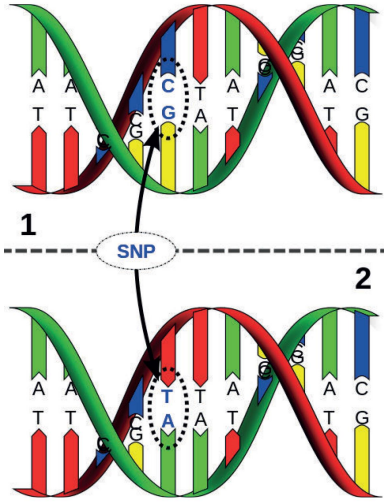


Figure 3: Single nucleotide polymorphism (SNP) with alleles C and T (©David Hall, License: Creative Commons).

We obtained SNP data of 99 individuals from 1000 Genomes Project [1]. In the obtained dataset, each individual has 7690 SNP values meaning that we have a 99 by 7690 matrix and elements of matrix are either 0, 1, or 2. We used this dataset to learn the statistics (e.g., correlations between the SNPs) that are used in the proposed algorithm. Thus, the size of this dataset is not an indicator for the scalability of the proposed algorithm.

5.2 Experimental Results

We evaluated the proposed watermarking scheme in various aspects. In particular, we evaluated its security (robustness) against watermark inference and watermark modification (Section 3.4). Robustness against watermark inference is evaluated by running collision and correlation attacks (as discussed in Section 3.3). In all collusion attack scenarios, we assume that Alice shares the same data portion with the SPs. This assumption provides the maximum amount of information to the malicious SPs. If different set of data points are shared with the SPs, malicious SPs can use the intersection of these data points for the collusion attack. Robustness against watermark modification is evaluated under various attacks in terms of the (watermark) detection performance of the data owner. The results also include evaluation of the loss in data utility due to watermark addition. We ran all experiments for 1000 times and report the average values. We denote the fraction of watermarked data (or watermark ratio) as $r = w/\ell$. Watermark ratio r also represents the utility loss in the shared data due to the added watermark.

5.2.1 Robustness against watermark inference

Here, we evaluate the robustness of the proposed scheme against watermark inference under collision and correlation attacks.

Collusion attack: First, we evaluated the probability of identifying the whole watermarked points in the collusion attack (when correlations in data are not considered). We considered the worst case scenario and assumed that all the SPs that has Alice's data are malicious, and hence they exactly know how many times Alice has shared her data to compute the exact probabilities for the attack (as discussed in Section 4.1). In Figure 4, we show the logarithm of this inference probability when data is shared with h SPs and they are all malicious (where $h = (1, 2, \cdots, 10)$) and when different fractions of data is watermarked. Overall, we observed that the probability to completely identify the watermark via the collusion attack is significantly low when the proposed technique is used for watermarking the data. Following our definition of robustness against watermark inference (in Section 3.4), under this attack model, the proposed scheme is p-robust against f-watermark inference for $f = 1$ and $p \leq 10^{-2}$ when h is as high as 10 and data utility is as high as 97% (i.e., r is as small as 0.025). As expected, we observed that the inference probability of the malicious SPs increases with decreasing r and increasing h values. That is, as data is shared with more malicious SPs, the probability to identify the watermarked data increases due to the collusion attack. Also note that even for significantly low values of r (that corresponds to high data utility), the proposed scheme provides high resiliency against collusion attacks.

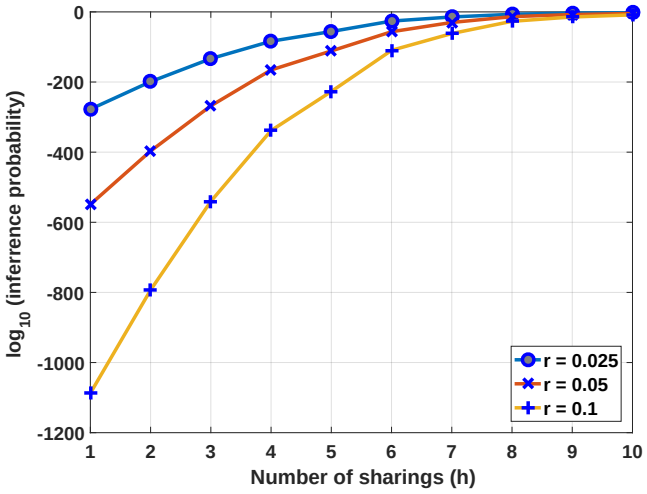


Figure 4: Probability of identifying the whole watermarked points in the collusion attack when h malicious SPs collude. r represents the fraction of watermarked data.

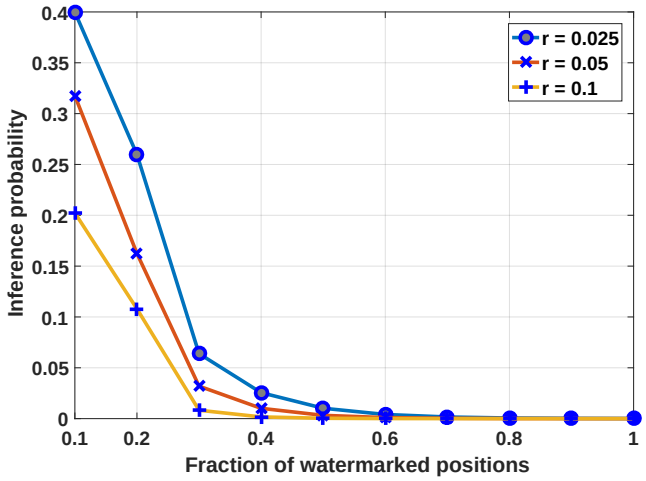
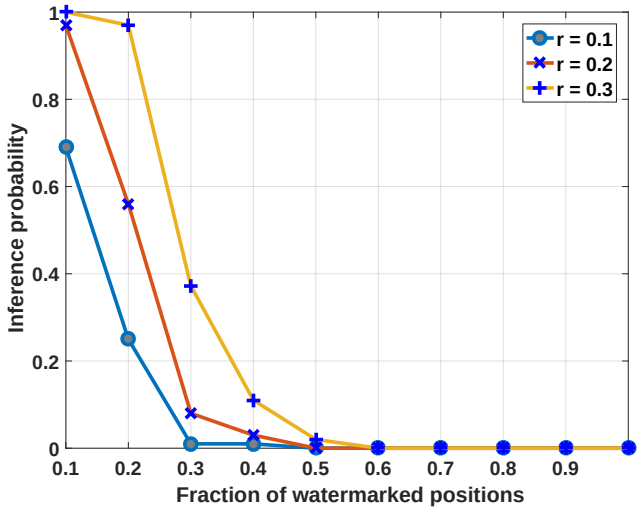


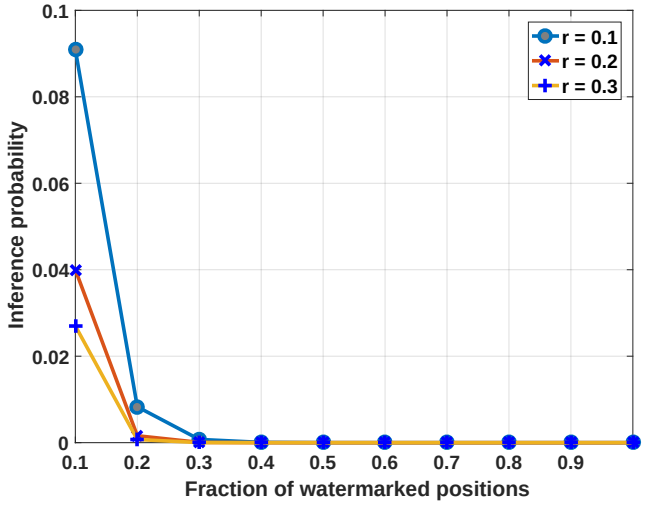
Figure 5: Inference probability to identify different fractions of the watermarked positions in the collusion attack when the number of colluding malicious SPs $h = 6$. r represents the fraction of watermarked data.

We also ran the same experiment to observe the probability of malicious SPs to identify different fractions of the watermarked positions. In Figure 5, we show this inference probability. For this experiment, we assume that the malicious SPs initially try to identify the watermark positions that has higher probability to be watermarked. Since we assume that the watermarking algorithm is publicly known by the malicious SPs, once they observe vertically aligned data points, they can compute the probability of being watermarked for each data position (as discussed in Section 4.1) and initially try to identify high probability watermark positions. We also set the number of colluding malicious SPs $h = 6$ and watermarked different fractions of the whole data (i.e., varied the r value). We observed that colluding SPs can identify small portion of watermark locations with small probabilities and this probability rapidly decreases with increasing fraction of watermarked data (r). Also, the probability to identify more than 30% of the watermarked locations is significantly low even when the malicious SPs collude. Notably, we show that when $r = 0.025$ (which means 200 watermarked data points on a data of size 7690, and hence preserves more than 97% of data utility), even when 6 malicious SPs collude, the probability to recover more than 30% of the watermark locations is very small. In other words, under this attack model, when $r = 0.025$, the proposed scheme is p-robust against f-watermark inference for $f = 0.3$ and $p \leq 10^{-1}$.

Correlation attack: To evaluate the security of the proposed scheme against the correlations in the data, we compared two techniques presented in Sections 4.1 and 4.2. In this analysis, we focused on a data length (ℓ) of 100 in our dataset. We find each pairwise correlation $Pr(x_i = \alpha | x_j = \beta)$ between these 100 data points, where $\alpha, \beta \in \{0, 1, 2\}$. To consider only strong correlations (and to avoid the noise that arise due to weak correlations), we only consider the ones above a threshold τ (we selected $\tau = 0.9$). Note that the correlations in the data are not symmetric. That is, $Pr(x_i = d_i | x_j = d_j)$ being high does not mean that $Pr(x_j = d_j | x_i = d_i)$ is also high.



(a) Correlations in the data are not considered when selecting the data points to be watermarked (i.e., technique proposed in Section 4.1 is used for watermarking).



(b) Correlations in the data are considered using the proposed algorithm when selecting the data points to be watermarked (i.e., technique proposed in Section 4.2 is used for watermarking).

Figure 6: Inference probability to identify different fractions of the watermarked positions in the single SP correlation attack. r represents the fraction of watermarked data.

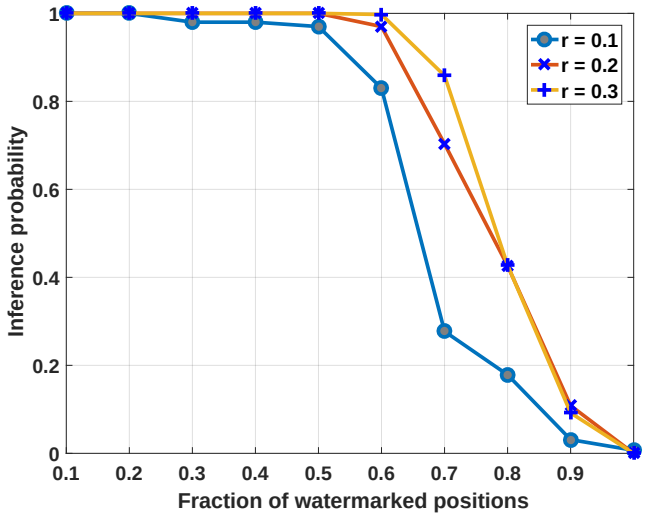
First, we compared two schemes for a single SP attack in terms of the probability of the malicious SP to identify different fractions of the watermarked positions. Note that in this attack, the malicious SP also utilizes its knowledge of correlations in the data.[2] In Figure 6 we show this comparison for different r values. We observed in Figure 6a that as r increases, the inference probability of the malicious SP increases for the technique presented in Section 4.1. This is expected since (i) if correlations are not considered while selecting the watermarked positions, the probability of the attacker to identify the watermarked positions also increases, and (ii) as more data points are watermarked in this way, the attacker can identify more watermarked position. However, when we consider the correlations in the data when selecting the watermark locations, the inference probability of the malicious SP significantly decreases as shown in Figure 6b. Also, in this scenario, inference probability decreases with increasing r value as expected. For instance, when $r = 0.3$, the watermarking scheme is p-robust against f-watermark inference for $f = 0.2$ and $p \simeq 1$ when the correlations in the data are not considered. When we consider the correlations in the data using the proposed watermarking algorithm, it becomes p-robust against f-watermark inference for $f = 0.2$ and $p \simeq 0$.

Collusion and correlation attack: We also compared two techniques presented in Sections 4.1 and 4.2 to show the resiliency of the proposed watermarking scheme against both collusion and correlation attacks at the same time. In this attack, each malicious SP first runs the correlation attack independently. As a result of this part, each malicious SP detects a number of watermarked points. For the advantage of the malicious SPs (and to consider the worst case scenario), we consider the outcome of the malicious SP with the highest number of correct detections. Let the number of watermarks detected by this malicious SP be m as a result of the first part.[3] Then, to detect the remaining $w - m$ watermarked points, malicious SPs run the collusion attack.

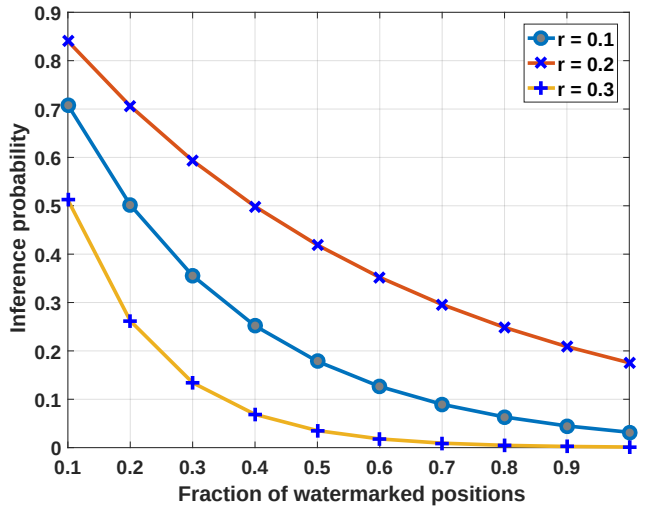
In Figure 7 we show this comparison for different r values when the number of colluding malicious SPs $h = 6$ (and data has been shared for 6 times). We observed that when the correlations are not considered in the watermarking algorithm, malicious SPs can identify more than half of the watermarked data locations with high probability as shown in Figure 7a. However, when we consider the correlations to select the data points to be watermarked, the inference probability of the malicious SPs significantly decreases (as in Figure 7b). For instance, when $r = 0.3$, the watermarking scheme is p-robust against f-watermark inference for $f = 0.5$ and $p \simeq 1$ when the correlations in the data are not considered. When we consider the correlations in the data using the proposed watermarking algorithm, it becomes p-robust against f-watermark inference for $f = 0.5$ and $p \leq 0.1$. This shows that the proposed watermarking scheme provides security guarantees against both collusion and correlation attacks with high probabilities even when all the SPs that receive the data are malicious and colluding (as in this experiment). Note that in Figure 7b, the reason inference probabilities for $r = 0.2$ is larger than the ones for $r = 0.1$ is due to the result of the optimization problem.

[2] We assume that knowledge of the malicious SP about the correlations is the same as the knowledge we utilized while adding the watermark in Section 4.2.

[3] As discussed, malicious SPs may detect less than w watermarked points as a results of the correlation attack.



(a) Correlations in the data are not considered when selecting the data points to be watermarked (i.e., technique proposed in Section 4.1 is used for watermarking).

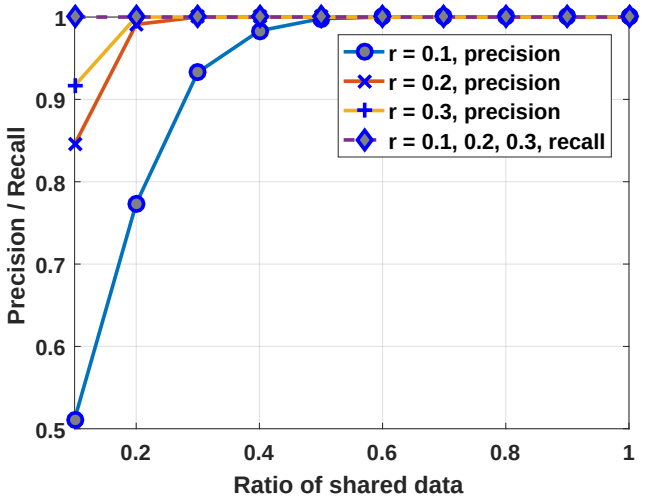


(b) Correlations in the data are considered using the proposed algorithm when selecting the data points to be watermarked (i.e., technique proposed in Section 4.2 is used for watermarking).

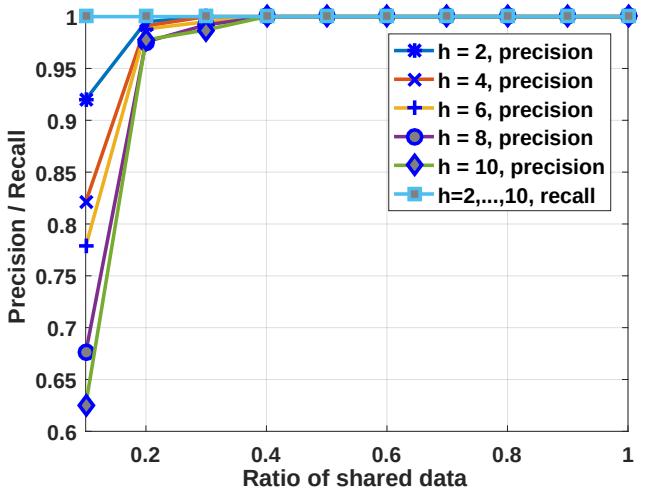
Figure 7: Inference probability to identify different fractions of the watermarked positions in collusion attack (when $h = 6$) in which the malicious SPs also use the correlations in the data. r represents the fraction of watermarked data.

5.2.2 Robustness against watermark modification

Here, we evaluate the robustness of the proposed scheme against watermark modification.



(a) Different fractions of watermarked data (r) when data has been shared with $h = 4$ SPs.



(b) Alice shares her data with h SPs when fraction of watermarked data $r = 0.2$.

Figure 8: Precision and recall values for the data owner to detect the malicious SP when the malicious SP partially shares Alice's data.

Partial sharing: We evaluated the detection performance (and robustness against watermark modification) of the proposed watermarking scheme when a malicious SP partially shares Alice's data. In this scenario, we assume that Alice has shared her data (same data portion at each sharing) with h SPs ($SP_1, \cdots, SP_h$). The malicious SP, rather than sharing the whole data with a third party without Alice's authorization, shares different fractions of the data to avoid being detected by Alice. As we have shown in previous experiments, the probability for a malicious SP to detect the watermarked data points is significantly low for our proposed scheme (even in the existence of collusion attack). Thus, we assume that the malicious SP randomly selects different fractions of data points to share with the third party. Here, we assume the mali-

cious SP does not further modify Alice's data before sharing it with a third party as such modification would degrade the credibility of the data (as we discuss in Section 6). We also consider and extensively study the impact of such modification to the detection performance later in this section.

We quantify the robustness against watermark modification under this attack using precision and recall metrics. Alice constructs a set $\mathbf{S}$ that includes the malicious SPs detected by her. We define true positive as a malicious SP that is in set $\mathbf{S}$, false positive as a non-malicious SP that is in $\mathbf{S}$, true negative as a non-malicious SP that is not in $\mathbf{S}$, and false negative as a malicious SP that is not in $\mathbf{S}$. In Figure 8, we show the precision and recall values for varying ratio of shared data by the malicious SP and for different r and h values. Following our definition of robustness against watermark modification (in Section 3.4), under this attack model, the proposed scheme is ρ/ζ-robust against watermark modification with $\zeta = 1$ for all considered values of h and r. Furthermore, when $r \geq 0.2$, $h \leq 10$, and the ratio of shared data by the malicious SP is more than 0.2, the proposed scheme is ρ/ζ-robust against watermark modification with $\rho \simeq 0.97$ and $\zeta = 1$. We conclude that the data owner can associate the source of the leakage to the corresponding SP with high probability in most of the cases, except when the malicious SP shares very small portion of user's data with a third party. However, this particular case would also reduce the benefit of the malicious SP (due to the unauthorized sharing) significantly. Furthermore, such partial sharing may degrade the credibility of data.

Watermark modification: Finally, we studied a stronger attack in which malicious SP (or SPs) modify the data in order to damage the watermark (and hence, it becomes harder for the data owner to detect the source of the data leak). Note that in practice, such modification of data not only reduces data utility (as we show in our experiments), but it also degrades data credibility while the malicious SPs share the data with a third party. Here, malicious SPs (or SP) try to remove or damage the watermark by (i) changing the states of data points that are different when they aggregate their data (i.e., when they detect a data point with multiple states in the aggregate data, they change its state to the majority of the observed states), and (ii) adding noise to other data points (i.e., changing states of other random data points). Eventually, data leaked by the malicious SPs has a watermark pattern represented as Z_α. Using Z_α and unique watermark patterns of the SPs (that previously received the data), Alice constructs the set $\mathbf{S}$ that includes the malicious SPs detected by her. As before, we evaluate the success of the detection via precision and recall metrics. For all following experiments we set the watermark ratio (r) to 0.05.

First, we consider the single SP attack in which data has been shared with h SPs and there is a single malicious SP. Watermark length (w) is known by the malicious SP and the malicious SP randomly changes ($\pi \times w$) data points in the data and shares it. For each SP i that received her data, Alice computes $g_i = |Z_\alpha \cap Z_{I_i}|$ (Z_{I_i} is the watermark pattern of SP i) and identifies the malicious SP as the one with the highest g_i value. In Figure 9, we show the precision and recall when the data owner knows that there is a single malicious SP and for different π and h values. In this scenario, both the precision and recall values are high even when the malicious SP significantly damages the watermark. Under this attack, the proposed scheme is ρ/ζ-robust against watermark modification with $\rho = \zeta \simeq 1$ when $\pi < 13$ and $h \leq 20$ ($\pi = 13$ means a utility loss of 65%).

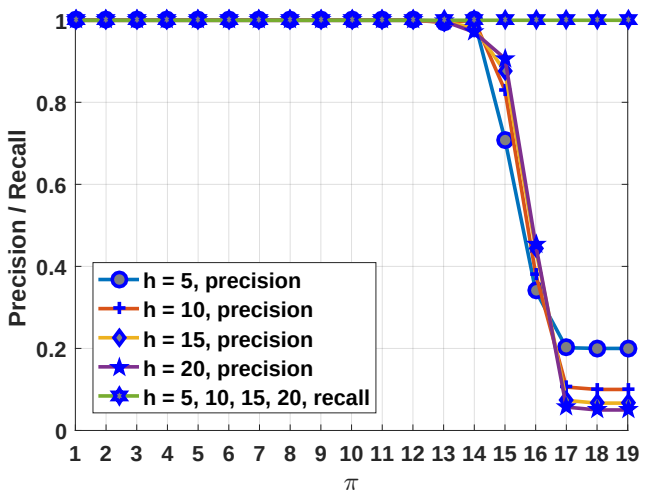
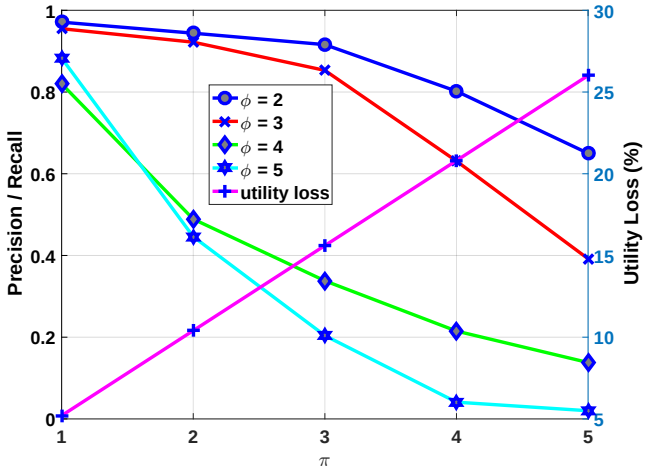
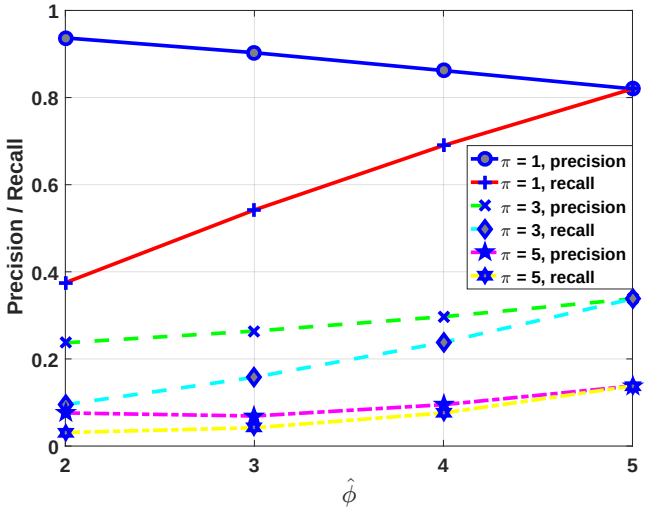


Figure 9: Precision and recall values for the data owner to detect the malicious SP in the single SP attack in which data has been shared with h SPs. Malicious SP randomly changes $\pi \times w$ data points to damage the watermark.

We also considered the case in which colluding malicious SPs compare their aggregated data and change the states of data points that are different, as discussed before. Colluding malicious SPs also add random noise in addition to changing the states of data points that are different in the aggregate data. We assume data has been shared with h SPs and colluding malicious SPs randomly change ($\pi \times w$) data points in the data before they leak it. The data owner Alice may or may not know the number of malicious SPs. Let the actual number of malicious SPs be ϕ and the prediction of Alice for the number of malicious SPs be $\hat{\phi}$ which can be any number from 1 to h. Alice first generates all combinations of h with $\hat{\phi}$. Then, she eliminates the combinations for which the union of the watermarked points of the SPs (in that particular combination) does not contain the watermark pattern in the leaked data (Z_α). Next, for each non-eliminated combination c_i, she computes $g_i = \sum_{j \in c_i} |Z_\alpha \cap Z_{I_j}|$. That is, she computes the sum of intersections of watermarked data points for each SP in the corresponding combination c_i with Z_α. Finally, she selects the set $\mathbf{S}$ as the most likely combination with the highest g_i value and concludes that the SP (or SPs) in the corresponding combination are malicious.



(a) Data owner knows the number of malicious SPs ($\phi = \hat{\phi}$).



(b) Data owner predicts the number of malicious SPs as $\hat{\phi}$ when the actual number of malicious SPs $\phi = 5$ and for varying π values.

Figure 10: Precision and recall values for the data owner to detect the malicious SPs in the collusion attack in which data has been shared with $h = 10$ SPs. Malicious SPs both change the states of data points that are different in the aggregated data and they randomly change $\pi \times w$ data points to damage the watermark. In (a), precision and recall curves for different ϕ values overlap. Also, in (a), we show the percentage of utility loss due to addition of extra noise by the malicious SPs.

In Figure 10, we show the precision and recall when $h = 10$, $\hat{\phi} = \phi$, and when the data owner does not know ϕ, respectively. In Figure 10a, we also show the percentage of utility loss in the data due to the noise addition by the malicious SPs (to damage the watermark). Here, the utility loss is shown when $r = 0.05$ (i.e., when 5% of original data is watermarked). As r value increases, the loss in utility (due to extra noise addition by the malicious SPs) also increases linearly. For instance when $r = 0.1$, to decrease the precision and recall values down to 0.2, half of the SPs that received the data should be malicious and they need to add noise to 50% of the original data to damage the watermark. As shown in Figure 10a, if the data owner knows the number of malicious SPs, both precision and recall of detection performance are high up to 30% of the SPs that received the data are malicious (and colluding) and up to a utility loss of 15%. That is, the proposed scheme is ρ/ζ-robust against watermark modification with $\zeta = \rho \simeq 0.9$ up to $\phi = 3$ and $\pi = 3$. Beyond this, we observed a decrease in both precision and recall with increasing π and ϕ values. This behavior gives some idea about the practical limits of our proposed scheme. When data owner predicts the number of malicious SPs (Figure 10b), we observed two cases: (i) when the added noise by the malicious SPs is less than 3 times the watermark length, the proposed scheme includes the actual malicious SPs in set $\mathbf{S}$ with a high probability. That is, the proposed scheme is ρ/ζ-robust against watermark modification with $\rho \simeq 0.7$ up to $\pi = 3$ and for all $\hat{\phi}$ values. When the added noise by malicious SPs is beyond this value, both precision and recall values start decreasing. However, adding noise beyond this value significantly reduces data utility as discussed before.

6 Discussion

Here, we discuss the potential use of our proposed scheme in real-life, its potential extensions, and future research directions.

Usability and Scalability. The proposed system detects the malicious SPs if data is leaked without the data owner's consent and if the data owner observes this leakage. Similarly, the SP that buys the data may keep the malicious SPs liable from this unauthorized sharing (with the cooperation of the data owner). It may be practically infeasible for a data owner to notice their data is leaked. Instead, this can be outsourced to a third party that continuously analyzes publicly available datasets that are made available by SPs that collect personal information.

The data owner can share their data with numerous SPs. The main constraint of the algorithm described in Section 4 is that the watermark pattern given to each SP should be unique. Thus, it is sufficient to change as less as one watermarked data point between two sharings of the same data with two SPs. However, as the overlap between watermark patterns increase, the precision and recall of the data owner to detect the malicious SP(s) decrease (as discussed in Section 5). We will further study this trade-off between scalability and watermark robustness in future work.

It is also important to note that the robustness guarantees of the proposed scheme may vary over time depending on the data type. For instance, via new discoveries in genomics, things that are non-sensitive today may turn out to be sensitive in the future. Similarly, new discoveries may result in new correlation models in the data. Thus, the evaluations we have

shown in Section 5 represent the robustness guarantees we can provide with today's knowledge.

Data utility. We may include w (i.e., watermark length) as one of the objectives of the optimization problem and put a limit on it. When we do so, the problem becomes a multi-objective optimization problem. Solution of a multi-objective optimization problem is non-trivial and many proposed techniques suggest converting the multi-objective problem into a single-objective one. Thus, we transform this multi-objective problem into single objective problem.

In this new formulation, there are two additions to the optimization problem introduced in Section 4.1. First, the objective function is changed as follows:

$$min\{\beta \cdot \prod_{i=0}^{h+1}(\frac{n_i^{h+1}}{n_i^{h+1}+n_{h-i+1}^{h+1}})^{n_i^{h+1}} + (1-\beta)\cdot w\}$$

We use the weighted sum of the watermark length and the inference probability as the new objective function. The weight (β) determines the tradeoff between the inference probability and the watermark length (i.e., data utility). Second, we add a new constraint as $w < w_m$, where w_m is the maximum allowed watermark length. This new constraint puts a threshold to the maximum number of watermark points. This new optimization problem guarantees the minimum weighted sum of inference probability and watermark length.

Depending on the data type, other utility constraints may also be included in the proposed algorithm. For instance, if adding watermark to two consecutive data points significantly reduces data utility, once y_i^h and $\hat{y}_i^h$ values are determined as a result of the optimization problem, watermark addition algorithm in Section 4.1 (or Section 4.2) can be tailored to take this constraint into account while adding the watermarks.

Other applications. The proposed watermarking algorithm can be applied for any type of sequential data (we describe the general framework for sequential data in Section 4). However, implementation for different data types is non-trivial. For instance, correlations in other types of data may be more complex. Furthermore, auxiliary information about the data owner may help a malicious SP to infer the watermarked positions with higher probability. To address some of these challenges, we will work on the application of the proposed scheme for location patterns as future work.

7 Conclusion and Future Work

In this work, we have proposed a scheme to share sequential data while addressing the liability issues in case of unauthorized sharing. The proposed scheme is between a data owner and one or more service providers. We have shown that the proposed watermarking scheme provides high security against collusion and correlation attacks. That is, with high probability, malicious service providers cannot identify the watermark on the data even if they collude or try to use the inherent correlations in the data. We have also shown that the proposed scheme does not degrade the utility of data while it provides the aforementioned security guarantees. We believe that the proposed work will deter the service providers from unauthorized sharing of personal data with third parties. The algorithm proposed in this paper does not consider if malicious SPs share statistics (e.g., average or median) about the data without the authorization of the data owner. Such statistics can also be shared by aggregating multiple data owners's data. In future work, we will also consider this and work to develop algorithm that also identify the unauthorized sharing in such scenarios.

References

[1] 1000gp phase 3 haplotypes. https://mathgen.stats.ox.ac.uk/impute/1000GP_Phase3.html, 2017.

[2] Single-nucleotide polymorphism. https://isogg.org/wiki/Single-nucleotide_polymorphism, 2017.

[3] André Adelsbach, Stefan Katzenbeisser, and Ahmad-Reza Sadeghi. A computational model for watermark robustness. *Proceedings of the 8th International Conference on Information Hiding*, pages 145–160, 2007.

[4] J.A. Bloom, I.J. Cox, T. Kalker, J.-P.M.G. Linnartz, M.L. Miller, and C.B.S. Traw. Copy protection for DVD video. *Proceedings of the IEEE*, 87(7):1267–1276, Jul. 1999.

[5] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, 1998.

[6] Tae-Yun Chung, Min-Suk Hong, Young-Nam Oh, Dong-Ho Shin, and Sang-Hui Park. Digital watermarking for copyright protection of mpeg2 compressed video. *IEEE Transactions on Consumer Electronics*, 44(3):895–901, 1998.

[7] Ingemar J Cox, Matthew L Miller, Jeffrey Adam Bloom, and Chris Honsinger. *Digital watermarking*. Springer, 2002.

[8] S Emmanuel, AP Vinod, D Rajan, and CK Heng. An authentication watermarking scheme with transaction tracking enabled. In *Digital EcoSystems and Technologies Conference*, pages 481–486. IEEE, 2007.

[9] Huiji Gao, Jiliang Tang, and Huan Liu. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1582–1586. ACM, 2012.

[10] John D. Hedengren, Reza Asgharzadeh Shishavan, Kody M. Powell, and Thomas F. Edgar. Nonlinear modeling, estimation and predictive control in APMonitor. *Computers & Chemical Engineering*, 70:133 – 148, 2014.

[11] Xiaoming Jin, Zhihao Zhang, Jianmin Wang, and Deyi Li. Watermarking spatial trajectory database. In *International Conference on Database Systems for Advanced Applications*, pages 56–67. Springer, 2005.

[12] Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. A review of text watermarking: Theory, methods, and applications. *IEEE Access*, 6:8011–8028, 2018.

[13] Suleyman S Kozat, Michail Vlachos, Claudio Lucchese, Helga Van Herle, and S Yu Philip. Embedding and retrieving private metadata in electrocardiograms. *Journal of Medical Systems*, 33(4):241–259, 2009.

[14] Sin-Joo Lee and Sung-Hwan Jung. A survey of watermarking techniques applied to multimedia. In *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)*, volume 1, pages 272–277. IEEE, 2001.

[15] Li Liu and Xiaoju Li. Watermarking protocol for broadcast monitoring. In *Proceedings of International Conference on E-Business and E-Government*, pages 1634–1637. IEEE, 2010.

[16] Claudio Lucchese, Michail Vlachos, Deepak Rajan, and Philip S Yu. Rights protection of trajectory datasets with nearest-neighbor preservation. *The VLDB Journal—The International Journal on Very Large Data Bases*, 19(4):531–556, 2010.

[17] Maurice Maes, Ton Kalker, J-PMG Linnartz, Joop Talstra, FG Depovere, and Jaap Haitsma. Digital watermarking for DVD video copy protection. *IEEE Signal Processing Magazine*, 17(5):47–57, 2000.

[18] N. Memon and Ping Wah Wong. A buyer-seller watermarking protocol. *IEEE Transactions on Image Processing*, 10(4):643–649, Apr. 2001.

[19] M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang. Privacy in the genomic era. *ACM Computing Surveys*, 48(1), Sep. 2015.

[20] Hussain Nyeem, Wageeh Boles, and Colin Boyd. On the robustness and security of digital image watermarking. In *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*, pages 1136–1141. IEEE, 2012.

[21] Hussain Nyeem, Wageeh Boles, and Colin Boyd. Digital image watermarking: its formal model, fundamental properties and possible attacks. *EURASIP Journal on Advances in Signal Processing*, 2014(1):135, 2014.

[22] S. S. Samani, Z. Huang, E. Ayday, M. Elliot, J. Fellay, J.-P. Hubaux, and Z. Kutalik. Quantifying genomic privacy via inference attack with high-order SNV correlations. *Proceedings of Workshop on Genome Privacy and Security*, 2015.

[23] Montgomery Slatkin. Linkage disequilibrium — understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9(6), 2008.

[24] Arezou Soltani Panah and Ron Van Schyndel. A lightweight high capacity ecg watermark with protection against data loss. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pages 93–100. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[25] Michail Vlachos, Johannes Schneider, and Vassilios G Vassiliadis. On data publishing with clustering preservation. *ACM Transactions on Knowledge Discovery from Data*, 9(3):23, 2015.

[26] Xiaohan Zhao, Qingyun Liu, Haitao Zheng, and Ben Y Zhao. Towards graph watermarks. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*, pages 101–112. ACM, 2015.

[27] Jianpeng Zhu, Qing Wei, Jun Xiao, and Ying Wang. A fragile software watermarking algorithm for content authentication. In *IEEE Youth Conference on Information, Computing and Telecommunication*, pages 391–394. IEEE, 2009.