PRIVACY-PRESERVING COMPUTATION AND ROBUST WATERMARKING TECHNIQUES FOR HEALTHCARE DATA

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

By

Sinem Sav

July 2018

Privacy-Preserving Computation and Robust Watermarking Techniques for Healthcare Data
By Sinem Sav
July 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Erman Ayday(Advisor)

Uğur Doğrusöz

Atila Bostan

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

PRIVACY-PRESERVING COMPUTATION AND ROBUST WATERMARKING TECHNIQUES FOR HEALTHCARE DATA

Sinem Sav
M.S. in Computer Engineering
Advisor: Erman Ayday
July 2018

Health and genomic data is sensitive in terms of carrying private information about individuals. One can infer inherited/genetic disorders, their occurrence probabilities, information about race, and kinship by analyzing an individual's genomic data. Furthermore, health data which is mostly collected by hospitals or other health institutions carries private information about individuals including the diseases they have at present or indicators of future diseases/disorders. While protecting such data, it is important to show that its utility is preserved and maximized since the data is used in researches. Regarding these facts, homomorphic encryption-based scheme (using Paillier cryptosystem) for the protection of health data and a novel watermarking scheme based on belief propagation algorithm for the genomic data is proposed in this work. Homomorphic encryption is used for the health data to show the ability of performing mathematical operations on the encrypted data without decrypting it with a real-life use-case. We show its practicality with the correctness and performance results. In the second part of this thesis, a watermarking scheme for genomic data is proposed to overcome the liability issues due to unauthorized sharing by service providers (SPs). Robust-watermarking techniques ensure the detection of malicious parties with a high probability and we show the probabilistic limits of this detection with different experimental setups and evaluation metrics. Lastly, this scheme guarantees the following with a high probability: (i) the utility is preserved, (ii) it is robust against single or colluding SP attacks, and (iii) watermark addition is compatible with the nature of the data as the proposed method considers auxiliary information that a malicious SP may use in order to remove/modify watermarked points before leaking the data.

Keywords: Watermarking, privacy, homomorphic encryption, genomic data.

ÖZET

SAĞLİK VERİLERİ İÇİN GİZLİLİĞİ KORUYAN HESAPLAMA VE GÜÇLÜ FİLİGRAN TEKNİKLERİ

Sinem Sav
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Erman Ayday
Temmuz 2018

Sağlık ve genetik veriler bireylerin kişisel bilgilerini taşıması açısından çok hassas verilerdir. Genetik verilerden kalıtsal/genetik hastalık veya bunların oluşma olasılığına dair bilgi, ırk veya akrabalık ilişkilerine dair çıkarım yapılabilir. Bunun yanı sıra, çoğunlukla hastaneler veya diğer sağlık kuruluşları tarafından toplanan sağlık verileri de bireylerin şu anda veya gelecekteki hastalıklarına dair gösterge olabilir. Bu tarz verileri korurken, veriler araştırmalarda kullanıldığı için verinin yararlılığını (utility) korumak çok önemlidir. Tüm bunları göz önünde bulundurarak, bu çalışmada sağlık verilerinin korunması için Paillier Cryptosystem kullanılarak homomorfik şifreleme ve genetik veri için yeni bir filigran¸şeması(watermarking scheme) kullanılmasını öneriyoruz. Homomorfik şifreleme ile sağlık verileri üzerinde yapılabilecek matematiksel işlemlerin, şifrelenmiş veri üzerinde, şifreyi bozmaksızın yapılabileceğini gerçek hayattan bir senaryo ile gösteriyoruz. Bu tezin ikinci kısmında, genetik verilerin yetkisiz servis sağlayıcıları tarafından paylaşılmasını engellemek için filigran şeması kullanıyoruz. Güçlü filigran şeması tekniği, kasıtlı şahısların yüksek olasılıkla tespit edilmesini sağlıyor ve bu tespitin olasılıksal limitlerini farklı deney düzenekleri kullanarak gösteriyoruz. Son olarak, bu şema şunları yüksek olasılıkla garanti ediyor: (i) verinin yararlılığı korunur, (ii) bir veya birden çok servis sağlayıcı tarafından saldırılara karşı dayanıklıdır, ve (iii) önerilen şema, kasıtlı servis sağlayıcılarının veriyi sızdırmadan önce filigranlı noktaları değiştirmek/kaldırmak için kullanacağı yardımcı bilgileri kullanarak filigranı verinin yapısı ile uyumlu hale getirir.

Anahtar sözcükler: Filigran, mahremiyet, homomorfik şifreleme, genetik veri.

Acknowledgement

First of all, I would like to thank my advisor Asst. Prof. Erman Ayday for his constant support and encouragement. He was always patient and guiding, being a role model for my future academic career. I am very grateful that he helped me whenever I have a question about my projects and steered me in the right direction, yet allowed me the flexibility while conducting this research.

I would like to thank my classmates A.Çağlar Öksüz for helping me understanding and suggesting new articles related to my work and C.Kaan Akyol for giving me the assistance and comments. I am particularly grateful for having a supportive friend Sevde Karayılan, she always studied beside me and encouraged me whenever I ran into trouble.

I would like to offer my special thanks to Doğukan Oğuz Sert for supporting me with all his patience, time and effort. Thank you for your unfailing support and continuous encouragement throughout my studies.

The first chapter of this work was supported by HAVELSAN Inc. which I appreciated their help and support.

Finally, I would like to express my very deepest gratitude to my parents and the person that I care the most, Simay Sav for being my sister. I am grateful for all their support, love, and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis introduces privacy preserving, secure schemes for the protection of healthcare and genomic data. The first scheme that is applied on health data is the homomorphic encryption and the second is novel watermarking technique based on belief propagation algorithm.

Health data and genomic data is sensitive in terms of carrying private information about individuals. One can infer inherited/genetic disorders or their occurence probabilities, information about race, and kinship by analysing an individual's genomic data. Similarly, healthcare data mostly collected by hospitals or other health institutions also carries private information about individuals including the diseases they have at present, or indicators of future disorders/diseases.

While collecting and analysing the genomic or health data, protecting individuals' privacy remains as a big issue. Hospitals and research centers need healthcare data for diagnosis, better treatment, improving treatment effectiveness, health care management, drug discovery. These institutions also need genomic data for disease susceptibility researches, control-case group analysis, and population statistics. Making the database public is one extreme example of leaking private information about patients. However, even making the database private within the hospital or medical center does not solve the privacy problem as one of the people having access to database within the center can sell the information to a third party, or can be just curious which implies semi-honest model for the problem. Therefore, it is important to anonymize medical databases to prevent leakage before publishing the database. But, anonymization comes with its flaws including linkage attacks. It is both theoretically and practically shown that de-anonymization is possible via linkage attacks where some attributes from one database is matched with the other even when personal information is hidden (even with having one attribute common and unique in two databases) [1]. One of the famous linkage attacks by Sweeney was on voter list and medical database aiming de-anonymization of the medical database [2]. Thus, the main aim of this work is to allow parties to perform statistical operations on the medical database which is kept encrypted and operations are carried out without decryption with the help of homomorphic methods. The application of the homomorphic encryption on health databases is carried out together with Havelsan Inc. and it constituted the first part of this thesis.

In the second part of this thesis, we tried to address liability issues that arises from unauthorized sharing of genomic data. For this, we proposed the use of watermarking. Watermarking is a well-known technique generally used for multimedia data. It is used for detecting the malicious parties that leak the data without consent [3]. This technique includes selecting some pixels and changing their values to create a pattern before sharing the data so that in case of a leakage, data owner can detect which service provider leaked the data according to associated watermarking patterns.

As far as genomic data is concerned, we still need a robust way of watermarking for the same purpose with multimedia sharing; to detect the malicious service providers (SPs) in case of a leakage. As genomic data reveals sensitive information about individuals; it is very important to ensure the privacy of the individuals sharing their sequential data with the SPs. Further, it is important to preserve utility of the data. For example, as genomic data is generally used for population genetics or detecting anomalies/disease susceptibility of individuals, the utility of the data is very important and any proposed technique for genomic privacy should preserve the utility at its highest level. In Chapter 4, we indicate the reasons why traditional watermarking techniques for multimedia data cannot be trivially adopted for genomic data as well as the main contributions of the proposed scheme, and explain the details of the proposed method.

The rest of this thesis is organized as follows: We give the related work on both security and privacy of healthcare and genomic data, digital watermarking as well as the literature given for using watermarking in genomic data in the next section. Chapter 3 gives the details of homomorphic encryption project conducted with Havelsan Inc., Chapter 4 gives details of proposed watermarking method for sequential data. We give the conclusion in Chapter 5.

Chapter 2

Related Work

2.1 Security and Privacy of Healtcare and Genomic Data

Recently, many researchers in the field of security and privacy put effort in discovering privacy-preserving techniques for health data or more generally for any kind of biological data carrying sensitive information about individuals including genomic data, or health/disease information. Some of the methods adopted for preserving privacy in medical data includes anonymization of collection of medical reports using k-anonymity, l-diversity, or t-closeness [4] [5] [2][6]. Other methods include de-identify, perturb, swap, recode, anonymize, or even simulate raw data before conducting data mining or publishing [7] [8]. However, all these methods decrease the utility of the data as stated in [6] and it is important to achieve maximum utility as health is a sensitive field which requires correctness. In addition, it is shown that they provide limited privacy protection and the proposed techniques are prone to inference attacks.

Ayday et al. [9] conducted a study of privacy preservation techniques for disease susceptibility evaluation which is a similar work to this work as the operations and value of the information is similar. This method relies on obfuscation method where they divide the output domain into evenly partitioned sections and outputs one section instead of the exact value in that partition. It is clear that these type of operations may reduce the utility of the data to a great extent.

Research on genomic data also reveals valuable information including drug discovery, disease susceptibility researches, control-case group analysis, and population statistics. However, genomic data also carries very sensitive information about individuals such as race, kinship, genetic disorders etc. Thus, security and privacy of genomic data has gained significance over the last few years. Several attacks have been proposed showing vulnerability of genomic data and countermeasures are proposed. Firstly, it has been shown that standard anonymization techniques cannot be used on genomic data since inference attacks still successful in these types of databases [10][11]. Also, Humbert et al. showed that the kin genomic privacy of an individual is threatened by his/her relatives sharing their genomic data[12]. Several protection mechanisms against these types of attacks have been proposed. One of them is to use cryptographic solutions to process genomic data [13][14][15]. Baldi et al. and Ayday et al. proposed techniques for the privacy-preserving use of genomic data using crypthographic techniques such as homomorphic encryption, Private Set Intersection (PSI), Private Set Intersection Cardinality (PSI-CA), and Authorized Private Set Intersection (APSI) [16][17]. Furthermore, Karvelas et al. proposed the use of oblivious RAM to access genomic data[18]. Huang et al. proposed an information theoretical scheme for the storage of genomic data [19].

As also stated in the review of Aldeen et al. on privacy preserving data miningv (PPDM) techniques [20], there are numerous other methods to preserve privacy including homomorphic encryption strategies. All methods come with their pros and cons but, it is important to consider the problem/application domain and the threat model to choose the most appropriate technique [21]. Last but not least, homomorphic encryption scheme is adopted in numerous works such as [22] on genomic, clinical, and environmental data or [23] to perform scientific investigations on integrated genomic data which are similar to the application domain of this project, and in which privacy is preserved while maintaining the utility by letting the specific statistical operations on the encrypted data. Thus, based on the problem statement which is given in Chapter 3, homomorphic encryption seems as the most appropriate solution where the medical databases are concerned. It is different from existing works as it does not decrease the utility of the data when statistics tests are concerned. Our difference from the existing work is that we show the practicality of homomorphic encryption applied on health data with a real-life use-case conducted with the industry.

2.2 Digital Watermarking

Digital watermarking is inserting a message or pattern to a digital signal such as image, song, or video [24]. It is closely related to steganography but the purpose of watermarking and steganography is different. The purpose of watermarking is to embed a code or pattern into the signal so that detection of the malicious parties is possible in case of a leakage/sharing without consent. However, steganography is the act of hiding an information or message that is not related to actual signal.

Digital watermarks are generally used for copy protection. Bloom et al. and Maes et al. proposed a system to protect copyrights on multimedia content in digital video disk (DVD) [25][26]. In these works, watermarking is used to encode copy information and behaves as a counter when the copy is made. If the counter reaches (by decrements in each copy) to a predefined limit, the system ensures that the hardware cannot create further copies. Memon and Wong proposed a buyer-seller watermarking protocol that uses watermark to detect illegal copy by inserting unique watermark pattern to seller copy before giving it to the buyer [27]. In image watermarking, frequently used methods are modifying the pixel values, using signal transforms such as wavelet [28] or Fourier transforms [29]. However, these are not directly applicable to sequential data because of the characteristics of the data that are explained in the previous section.

One of the most well-known fields that watermarking is applied is audio signals. Substitution of the least significant bit (LSB) for steganography [30] or adding echo [31] (compatible with the audio using the same characteristics) are notable time-domain watermarking techniques used in audio watermarking. In [30], authors propose hiding a message in the LSB of audio signal and although it is proposed for steganography, they highlight its importance for privacy and copy protection. However, changing the LSB and hiding message is easily detectable. The results show that changing up to three LSB is not detectable but after that point signal changes and it becomes detectable. Authors also showed that for successfully embedding a message, the samples of the host message should be at least eight times the number of bits in secret message. This limits the application of the method and as it aims steganography at first hand, it is not easily applicable to cases where privacy and robustness against detection is aimed. LSB is also used for image watermarking by substituting the least significant bit of the pixels [32][33]. The method proposed in [33] employs the genetic algorithm but it is computationally costly. Later, Chang et al. proposed the use of dynamic programming for LSB-substitution in images [32]. However, the main focus of the paper is getting low computational time and better quality than the former paper. Although, they get better quality images while hiding message in LSB of images and lower computational time, they did not perform any attacks on the proposed method to show the robustness against detection. Another way of adding watermark to audio is using quantization [34][35]. Note that none of these methods are directly applicable to sequential data since data characteristic is not necessarily same as the audio signal or image; meaning there is no high range of values or additions may not be possible when the data is labeled.

Inserting watermark into text documents is another field that watermarking scheme is used [36][37]. As the redundancy in multimedia data ease the watermarking (since it is hard to differentiate watermark by human eye), watermarking text document is completely different task. Such systems generally insert the watermark into the text in two ways: (i) using line-shift algorithm which moves a line upward or downward/left or right, and (ii) using word-shift algorithm which moves the words and expands the spaces between them. Another technique for watermarking text document by Atallah et al. proposes the use of natural language watermarking scheme [38][39]. They proposed watermarking natural language text by using watermark bit string in the syntactic structure of some of the sentences. In the proposed scheme, the watermark is not damageable when text is translated into another natural language. Similarly, Topkara et al. proposed a method based on features of the sentences and orthogonality between them to watermark sentences[40].

Aforementioned text watermarking works are different from our work in a way that they do not guarantee security against collusion attacks: meaning that more than one malicious parties collude and may detect the watermarked data points, spaces in texts or pixel values. In addition, the format can be changed in the watermarked text; meaning one can remove all unnecessary spaces easily.

Boneh and Shaw proposed a general fingerprinting for digital data that is secure against collusion attacks [41]. To guarantee the security against collusion attacks, they provide a lower bound for the length of the fingerprint(l) as $l >= 1/2(c-3)log(1/c\epsilon)$ for construction of a fingerprinting scheme that is c-secure with ϵ code. Scheme that is c-secure with ϵ code is defined as the probability of detecting a user from a coalition (colluding) c users is at least $(1-\epsilon)$. Thus, the application of this scheme where the number of malicious parties is too high becomes harder. Also, they do not consider the utility loss of the data.

2.3 Watermarking Genomic Data

More similar to our work, literature for watermarking genomic data shows that the work so far aims unauthorized use of genetically artificial organisms with dummy strands or genetically modified organisms (GMOs) protected by patents[42][43]. Liss et al. describe a method for embedding messages (watermarks) within open reading frames of synthetic genes by translating messages into a binary string, and representing in the reading frame by synonymous codon choice to this string [43]. If the genetic data is considered as codons, utility preservation will be less of a problem when the watermark is applied. However, this scheme is not directly applicable to any sequential data as the it considers genetic data in particular and employs the codon regions. In addition, stenography's aim is different than watermarking. Therefore, only a small portion of data requires changing for the message to be hidden. If we apply the same procedure for watermarking sequential data directly, security against collusion attacks will decrease. Small proportion of change means very small area to compare multiple samples. It will result with instant detection of watermark and its whereabouts. As the data turns into multimedia using the above method, watermarks are still vulnerable to insertion and deletion attacks.

Iftikhar et al. proposes a distortion-free genetic data watermarking method for the database and embedding a watermark string into a feature selected according to their mutual information [44]. However, they do not consider any collusion attacks and the method is applicable to whole dataset since it is based on features. Thus, in a scenario where the user herself wants to protect single data before sharing, the method is not simply applicable. Boujdad et al. proposes composition of watermarking with encryption, fragmentation, and client-side computations [45] for the privacy of shared genetic data which makes more complex model for computation and usage compared to our method.

In this work, different from all previous works, we propose a novel watermarking technique that addresses the liability issues in case of an unauthorized sharing of genomic data, which is extendable to any sequential data. Differently from the watermarking techniques used for genomic data, our proposed scheme that is applied on genomic data in Chapter 4 does the following:

- We apply watermarks directly to SNPs meaning, single data points, explained in the use case section, not only genes or codons. Thus, it is extendable to any use for sequential data.
- We consider correlation attacks, family attacks, and collusion attacks all together. The attacker model employs all auxiliary information to remove/modify watermarked data points.
- Utility is also considered and optimized during watermarking with the use of utility factor nodes in belief propagation algorithm.

Chapter 3

Privacy-Preserving Medical Databases

3.1 Problem Definition

One of the applications of data mining and data analysis is in the field of healthcare. Valuable information and statistics are gathered from this type of data by researchers, or even by hospital staff. However, medical databases carry private information about individuals. Thus, protecting the data and ensuring a secure environment while preserving the utility of the data is important in this field. For this purpose, homomorphic encryption which allows performing basic computations on the encrypted data without decryption is proposed in this work.

Mining health data or performing statistics on this data results in valuable information including diagnosis, better treatment, treatment effectiveness, health care management, and drug discovery [6]. Therefore, database keeping medical data is generally shared with many parties including researchers, doctors, hospital staff for accounting, registration or recording purposes, or even with public sometimes.

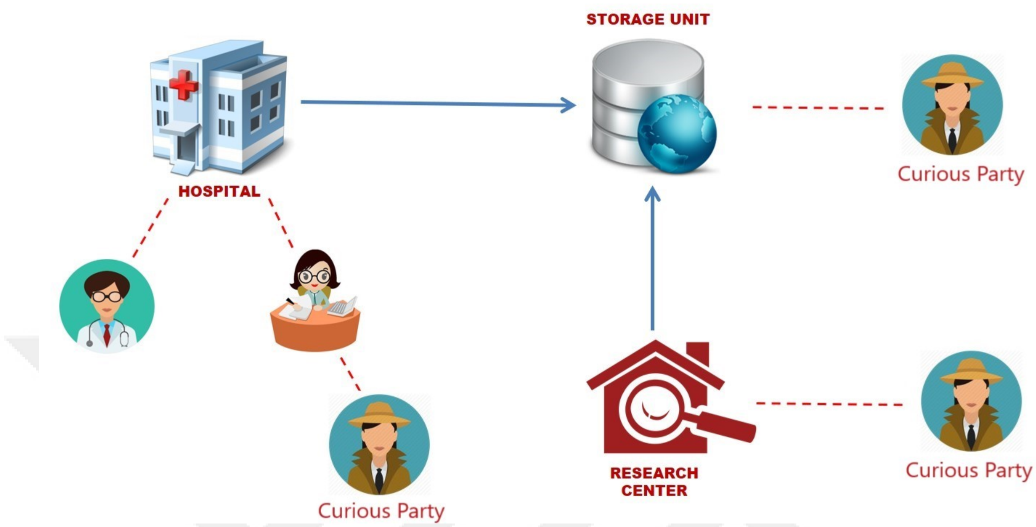


Figure 3.1: Overview of the system and threat models.

3.1.1 System Model

We consider a system between a hospital, a research center, and a storage unit whose access can be outsources to third party (i.e. companies that ensure the security). The data flow and the parties of our system model is shown in Figure 3.1. The authorized person from the hospital sends the data to the storage unit and research center has a direct access to this unit for the research purposes. However, this system poses a threat for individuals when there is a curious party that may leak the private information of a patient.

3.1.2 Threat Model

As mentioned, medical data has a valuable information for researchers, and institution. However, it carries sensitive information and in this work, it is assumed that the parties that are willing to conduct statistical tests on the data for research or aforementioned purposes, and parties in the storage unit or hospital are semi-trusted or semi-honest where a curious party can leak the underlying information as the definition of the semi-honest protocol implies that the corrupted parties follow the protocol specification but they are curious, having passive adversarial power. Thus, it is important to protect the privacy of patients while preserving the utility of the data. As the application domain is medical data, the threat model becomes having any curious party having access to the content of the database either in research center or any department other than the authorized ones (probably doctors). Figure 3.1 shows the high-level threat model. Because of this threat model, keeping the sensitive data all encrypted in the database and giving only the results of statistics tests seems as one appropriate solution. Thus, privacy-preserving scheme; namely, homomorphic encryption, is proposed throughout this chapter similar to the work described in [22].

3.2 Proposed Solution

In this section, we briefly give the preliminaries that we facilitate the discussion, and the details of the proposed solution for the threat model given in Figure 3.1

3.2.1 Preliminaries

In this section, we briefly introduce homomorphic encryption, and Paillier Cryptosystem.

3.2.1.1 Homomorphic Encryption

Application of homomorphic encryption to medical databases is the main aim of this work as it allows basic operations to be carried on ciphertext without decryption and when decrypted, it matches exactly the result of the operation. There are two types of homomorphic encryption: (i) Fully homomorphic encryption where it allows the full required functionality, and (ii) Somewhat homomorphic encryption allowing limited evaluation. In this work, we used one of the variants which is Paillier cryptosystem and this cryptosystem supports following operations on encrypted data:

- *Addition operation:* The multiplication of two ciphertext that is encrypted using Paillier cryptosystem gives the encrypted version of the addition of corresponding plaintexts.
- *Multiplication with a constant operation:* The k-th power of ciphertext that is encrypted using Paillier cryptosystem gives the encrypted version of the multiplication of the plaintext with the constant k.

In this work, we also use ***subtraction*** which is extended from aforementioned operations; meaning that the pre-condition of subtraction is the first number is always bigger than second (since there is no representation of negative values) and the number is subtracted from the former one is multiplied by -1 using *modInverse* and then addition operation is used.

3.2.1.2 Paillier Cryptosystem

Paillier Cryptosystem which is a public-key cryptosystem enables the basic operations on encrypted data without revealing the actual values to the client [46]. As it is an algorithm exactly for the main purpose of this project, it will be adopted and the following representation which is the modified Paillier cryptosystem in [22] shows how the encryption/decryption protocol works.

Key Generation: Public-key for this problem is represented as $(n, g, h = g^x)$ where the strong key is the factorization of $n = zy$ which will be used for encryption purpose. z and y are two large primes and the weak secret key is $x \epsilon [1, n^2/2]$, and g of the order $(z-1)(y-1)/2$. By selecting a random a such that $a \in \mathbb{Z}^*_{n^2}$, g computation of g becomes $g = -2a^{2n}$ Then, encryption and decryption steps become as follows:

Encryption: Let $[m]$ denote the encrypted version of a message m. To encrypt the message $m \in \mathbb{Z}_n$, we first select a random $r \in [1, n/4]$ and generate the ciphertext pair (C_1, C_2) as:

$$C_1 = g^r \bmod n^2 \qquad \text{and} \qquad C_2 = h^r(1 + mn) \bmod n^2 \tag{3.1}$$

Decryption: The decryption of message $[m]$ computed as:

$$m = \Delta C_2 / C_1^x \qquad \text{where,} \tag{3.2}$$

$$\Delta u = (u - 1) \bmod n^2/n \qquad \text{for all,}$$
$$u \in \left\{u < n^2 \mid u = 1 \bmod n\right\}$$

Apart from key generation, encryption, and decryption schemes that are defined above, proxy re-encryption is also possible as defined in [22] with modified Paillier cryptosystem as below:

Proxy re-encryption: Proxy re-encryption allows one party to change the encrypted ciphertext without seeing the content of it and another party can decrypt it with a private key [47]. If we randomly split the secret key in two shares $x1$ and $x2$, such that $x = x1 + x2$;then, an encrypted message (C_1, C_2) can be partially decrypted to a ciphertext pair (C_1', C_2') using $x1$ and exploiting following equations:

$$C_1' = C_1 \qquad \text{and} \qquad C_2' = C_2 / C_1^{x_1} \bmod n^2 \tag{3.3}$$

Then, (C_1', C_2') can be decrypted using x_2 with the decryption function given in formula (2) to recover the original message. This type of protocol can be adopted to medical database privacy problem in the future easily in case of involvement of a third party (i.e.patient involvement). For now, as it is allowed for client to decrypt the message using private key, usage of proxy re-encryption remains in the background.

3.2.2 Method

Figure 3.2 shows the proposed solution, regarding the threat and system models explained in Section 3.1.1 and 3.1.2, in high-level. We propose the following solution for basic statistics tests on the encrypted medical data. Below, a few scenarios where a statistic test that is required for the institution is represented. These scenarios represent different processing of the encrypted data, from simple to complex ones. In our data model:

- All results coming from blood tests and including any health status are encrypted.
- Demographic information of the individuals including age, gender, nation, or their name/surname are not encrypted.

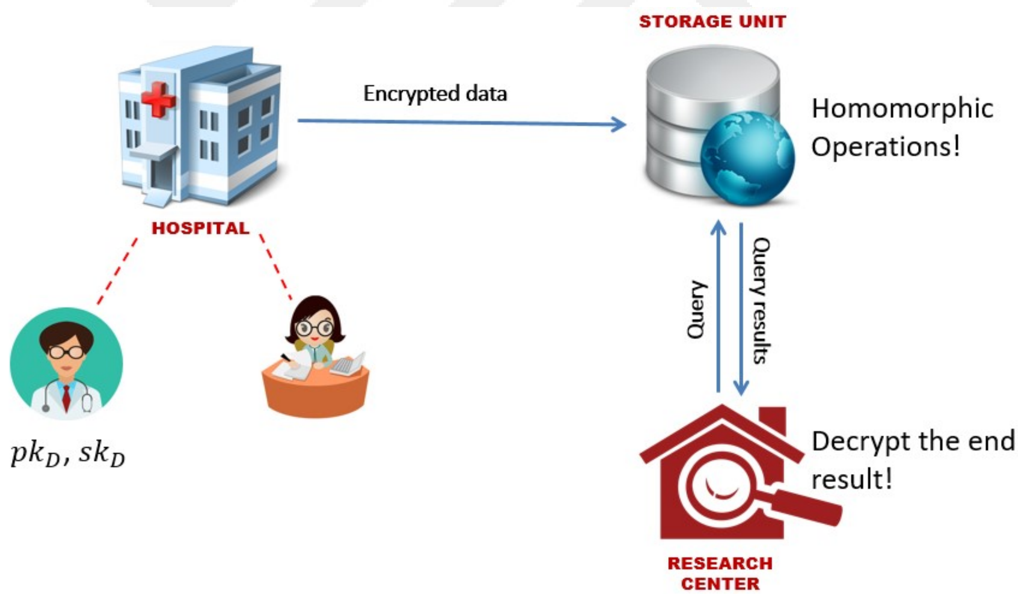


Figure 3.2: High-level Representation of the Solution

3.2.2.1 Simple Queries

1. The number of patients having HIV+ status

The Server: Has the encrypted version of the HIV+ column where plaintext 0 indicates HIV- and 1 indicates HIV+. Say $[m_1], [m_2], ..., [m_n]$ represent the encryption of $m_1, m_2, ..., m_n$ which are the HIV column values.Then, the server computes

$E(m) = [m_1] * [m_2] * ... * [m_n]$ where,

m becomes $m = m_1 + m_2 + .. + m_n$ in plaintext by the definition of Paillier cryptosystem.

The Client: The client gets encrypted $[m]$ and recovers the number of patients having HIV+ by decrypting it.

2. The average LDL cholesterol of the female patients

The Server: Has the encrypted version of the raw medical data. Say $[m_1], [m_2], ..., [m_n]$ represent the encryption of $m_1, m_2, ..., m_n$ which are the columns representing LDL cholesterol level of the female patients (gender is not encrypted). The server computes

$E(m) = [m_1] * [m_2] * ... * [m_n]$ where,

m becomes $m = m_1 + m_2 + .. + m_n$ in plaintext by the definition of Paillier cryptosystem.

Let $[t]$ denote the encryption of total number of patients with the given condition. Then, the server sends $[m]$ and $[t]$ to the client as there is no known way that provides division operation on ciphertext in Paillier cryptosystem.

The Client: The client gets encrypted $[m]$ and $[t]$ and recovers the average by decrypting them using the private key to get plain m and t. Then, applies division operation to get the desired average.

3.2.2.2 More Advanced Queries

Above schemes are examples of the simplest statistics tests; but, other types of solutions are needed for more complex ones. For example, the queries including comparisons. For those tests like *"The ratio of the patients having LDL cholesterol <100 mg/dL to all the patients."*, one can implement a new protocol that allows comparison of the numbers as expressed in [48] or other solutions that are described below for the simplicity.

1. **The number of the patients having LDL cholesterol >100 mg/dL**

The Server: Has the encrypted version of the raw medical data. Say $[m_1], [m_2], ..., [m_n]$ represent the encryption of $m_1, m_2, ..., m_n$ which are the columns representing LDL cholesterol level similar to mentioned test. However, in this case, we have one additional column showing if LDL is above or below a threshold (100 in this case). Thus, this column is again binary and multiplication over that column will give the addition of the number of patients having LDL >100 mg/dL.

The Client: The client gets encrypted sum $[m]$ and recovers the query result by decrypting it.

Possible Problems: This type of solution solves the single problem; where we have a fixed threshold. If one would like to query "number of patients having LDL >50 mg/dL" with this scenario, the scheme is need to be changed and comparison protocol will be needed.

2. **The number of the patients having HIV- status**

The problem in this case is that as the pre-proposed scheme has 0 or 1 in one column indicating HIV status, summing over that column will only give total number of HIV+ patients. Thus, we need to encode the HIV status as [01] or [10] and encrypt each bit seperately in seperate columns. Thus,

The Server: Has the encrypted version of HIV status in two columns; namely HIV_NEG and HIV_POS columns holding binary values. Say $[m_1], [m_2], ..., [m_n]$ represent the encryption of $m_1, m_2, ..., m_n$ which are the columns representing HIV_NEG and $[k_1], [k_2], ..., [k_n]$ represent the encryption of $k_1, k_2, ..., k_n$ which are the columns representing HIV_POS. Then, server simply computes the sum over either HIV_NEG or HIV_POS depending on the query; namely, HIV_NEG in this query.

The Client: The client gets encrypted sum $[m]$ and recovers the query result by decrypting it.

Possible Problems: This type of solution increases the space complexity as we add new column. The problem becomes bigger if we consider that we will have tens of similar laboratory results. Therefore, we can implement below alternative instead of column addition; but, encoding scheme is implemented for this project for simplicity (as the number of columns is not too large).

Alternative Solution:

The Server: Has the encrypted version of the HIV+ column where plaintext 0 indicates HIV- and 1 indicates HIV+ similar to described scheme under Example Queries. Then, to get the number of HIV- patients, the server computes

$$[HIV-] = \sum_{i=1}^{n}([x_i] * [-1])^{-1} \tag{3.4}$$

where $[x_i]$ represents the encrypted column value and $[-1]$ represents encryption of -1. With this scheme, as the multiplication will give addition and power will give multiplication with a constant in plain domain, 0s will be converted to 1 and 1s will be converted to 0s.

The Client: The client gets encrypted sum [HIV-] and recovers the query result by decrypting it.

3. The number of the patients having HIV+ and coronary heart disease(CHD)

Assuming both HIV and CHD columns hold encrypted binary values 0 indicating negative and 1 indicating positive values, the problem becomes finding rows where both HIV and CHD columns are 1. To get those rows, we can multiply each row value and if the multiplication is 1, it means the patient has both HIV and CHD. Thus, the reduced problem is to perform multiplication in the cipher

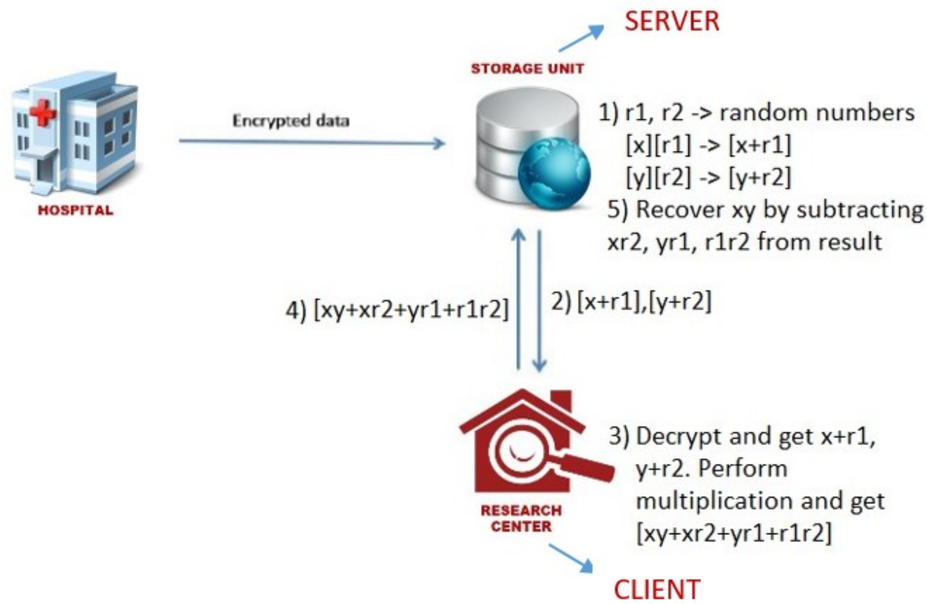


Figure 3.3: Multiplication operation with user involvement

domain. Figure 3.3 shows the described protocol for multiplication. The solution with the *user involvement:*

The Server: Chooses two random values r_1 and r_2, and computes $[x][r_1]$ and $[y][r_2]$ where x and y represents HIV and CHD column values respectively. This multiplication gives addition as $[x + r_1]$ and $[y + r_2]$ and server sends it to the client.

The Client: The client decrypts $[x + r_1]$ and $[y + r_2]$ using private key and performs multiplication operation on plain domain (As r_1 and r_2 are only known by server, client cannot infer x and y values. By multiplication, client has $xy + xr_2 + yr_1 + r_1r_2$ and sends it to server.

The Server: As the server knows r_1 and r_2, it can compute xr_2, yr_1, and r_1r_2 and by subtracting from the value obtained from the client (using aforementioned subtraction protocol), server can find multiplication of x and y.

Possible Problems: Performing this kind of operation for each row with the user involvement can be costly in terms of time. Efficiency of the protocols will be given in the evaluation section.

3.3 Evaluation

3.3.1 Dataset

As the project was an ongoing project with the **HAVELSAN Inc.**, the tests will be done on the generated data similar to real (collected from HAVELSAN Inc.), partial dataset gathered from the organization. It consists of 5 necessary tables for the example statistics of top priority with the sensitive data to be encrypted using mentioned scheme. Starting from these tables and implementing the most general and necessary operations on this dataset, more complex operations are handled using the mentioned scheme.

3.3.2 Hardware/Software Tools

In this project, we used Oracle Database 12c Release as database and implemented the protocols in Java. For the evaluation of this scheme, we used the computer having the following specifications: Intel Core i7 6700HQ 2.60GHz x64, 16GB 1 TB HDD + 128GB SSD.

3.3.3 Correctness of the Method

To see the correctness of the method, experiments on both encrypted and unencrypted data are performed. The purpose is to see if the results on encrypted data match the ones gathered from unencrypted data. The results show that the proposed method and protocols work with 100% correctness and the results are exactly same as the results of unencrypted data.

3.3.4 Computational Complexity

size/time(sec)	Unencrypted Table		Encrypted Table	
Table Size(#rows)	Binary Column	Non-Binary Column	Binary Column	Non-Binary Column
100	0.012s	0.011s	0.007s	0.122s
1000	0.072s	0.015s	0.245s	0.120s
10000	0.132s	0.138s	1.658s	2.242s
40000	0.292s	0.224s	3.262s	10.325s
100000	11.024s	11.025s	22.132s	32.128s

Table 3.1: Performance Evaluation of Paillier Cryptosystem

To see the computational complexity, we performed different tests with different table sizes for both encrypted and unencrypted data. First results showed that the time is increasing too much with the increasing data size. However, as the expectation looking at the protocols is linear time complexity, we thought that it could be something related to computer or database. Thus, the experiments are conducted by taking data chunks of size 1000 from database. 3.1 shows the performance results binary column and non-binary column operations. These tests are "the number of patients having HIV+ status" and "the average LDL cholesterol of the female patients" respectively. The key size used in these experiments is 1024 and we took the average of 20 experiments.

Chapter 4

Watermarking For Sharing Genomic Data

The adoption of watermarking prevents unauthorized sharing of genomic data by the SPs as mentioned in the introduction of this thesis. In such a case, the data owner (or a third party) can associate the source of the leakage to the corresponding SP (or SPs) by looking at watermarking patterns. Watermarking is already commonly used to prevent illegal copies of multimedia data. The main contributions of the proposed work are summarized as follows:

- The proposed method uses the auxiliary information that SPs may use to detect the watermarked positions before unauthorized sharing. This technique considers all auxiliary information before inserting watermark. The proposed scheme minimizes the probability of the identifiability of the watermark by a single SP. Set of SPs can detect the watermarked positions only by colluding their data and we show that even in that case, we identify the set of colluding SPs with high probability.
- While providing these robustness guarantees, the proposed algorithm also minimizes the utility loss in the genomic data by using utility factor nodes in the belief propagation algorithm.

- We evaluate our method with different attack models and show that the set of malicious SPs can be detected with high probability in different scenarios(removing or modifying data points). We also show the relationship between the probabilistic limits of this detection, watermark length, and the number of shared data.

- We also showed the application of the proposed method on genomic data and use of auxiliary information in watermarking. These information are MAF values of single-nucleotide polymorphism (SNPs), inherent correlations and genomic information of family members.

4.1 Problem Definition

Here, we first give why traditional watermarking techniques for multimedia cannot be trivially used for sequential data. Further, we describe the data model, system model, threat model, and objective of our system. Frequently used symbols and notations are presented in Table 4.1.

4.1.1 Differences Between Watermarking Digital and Sequential Data

It is very different watermarking sequential data and multimedia data for the following challenges:

- **Noise Tolerance/Identifiability:** As human eye cannot differentiate the watermark in multimedia data easily, this type of data is noise tolerant. Any well-known watermarking schemes explained in the related work and used for multimedia data (such as [27][28][29][30][31][32][36][37]) takes the advantage of the high range of values and redundancy in the data. However, it is not necessarily the case for sequential data where data may have low range of values; even only 3-4 values which is the case for genomic data. This fact makes the watermark easily detectable and simple modification/perturbation/addition techniques do not apply to sequential data for this reason.

- **Correlations:** There may be inherent correlations in the sequential data. For example, because of the linkage disequilibrium (see Section 3.2.1) there are important correlations within the genomic data. Thus, inserting watermark to one position may imply that some data points having correlation with that point should be modified accordingly. Otherwise, malicious SPs can easily detect watermarked points by looking at the correlations and remove them before sharing. Some multimedia data such as images or videos have correlations too. However, even there are correlations in the data, watermarking can be easily implemented without affecting the correlations by adding information and taking the advantage of redundancy in the data (i.e Oh et al. proposes echo addition to audio signals [31]). Adding information is not easily applicable to sequential data if the data is labeled and redundancy is low(i.e. genomic data, as the SNP ids are unique, adding more SNPs with different ids is easily detectable). Thus, one need to modify/perturb existing values without affecting the correlations.

- **Utility:** Another challenge for watermarking sequential data is the identifiability of the watermark which have an impact on the utility of the data. Thus, utility/privacy trade-off is an important concern while watermarking sequential data. Because of the aforementioned characteristics of the multimedia data (redundancy), watermarking does not affect the utility of the data too much. However, watermarking sequential data where the data have few states (low range of values) may affect the utility too much. For example, changing a single-nucleotide value in a genomic data have a direct impact on the study using that data. Utility is very important for sequential data since it is used for research purposes with all its details such as genomic or location data. Thus, one should be very careful while watermarking this type of data not to affect utility too much.

- **Existence of Auxiliary Information:** The use of watermarking is for the protection of the data and detection of malicious parties in case of leakage.

Thus, the aim of malicious party is to find the watermark pattern to remove or modify it before sharing. While doing this, the use of auxiliary information may help identifying the watermarked points. Auxilary information that can be used in the multimedia data is very limited (i.e. investigating the frequency) and generally depends on the perception of the human eye. However, in the sequential data, there may be too much information that an attacker may use. Taking the genomic data for example, malicious parties may look at the population statistics (frequencies of nucleotides), familial genomic information, phenotypes and correlations. Thus, proposed watermarking technique should consider all these auxiliary information and optimize the watermark pattern before sharing.

Considering the facts that are mentioned above we propose a novel watermarking technique for sequential data where robustness against attacks and utility is preserved with a high probability. The adoption of watermarking prevents unauthorized sharing of sequential data by the SPs. In such a case, the data owner (or a third party) can associate the source of the leakage to the corresponding SP (or SPs) by looking at watermarking patterns.

4.1.2 Data Model

Our data model consists of SNP values having the states $0, 1$ or 2. Data is abbreviated as $x_1, x_2, ..., x_d$ where x_i shows the value of the data point. The length of the data is d and w is used to show the watermark length that is the number of points that will be watermarked at the end of the algorithm. For the remaining notations, please refer to Table 4.1.

4.1.3 System Model

In our proposed system, data owner (Alice) shares personal sequential data with multiple service providers as shown in Figure 4.1. Before sharing, she uses unique

$x_1, x_2, ..., x_d$	Set of data points
$y_1, y_2, ..., y_z$	Possible values (states) of data points
d	Length of the data
w	Watermark length
h	Number of total sharing of the data
Z_h	Watermark pattern when data is shared with h^{th} SP
W_h	Set of data points shared with h^{th} SP

Table 4.1: Frequently Used Symbols and Notations

watermark patterns to ensure that in case of a leakage, malicious SPs can be detected with a high probability. Note that the shared data with each SP is different from each other and from the original data. As the data is changed, the utility decreases and we propose use of utility nodes in belief propagation algorithm to ensure that utility loss is minimized.

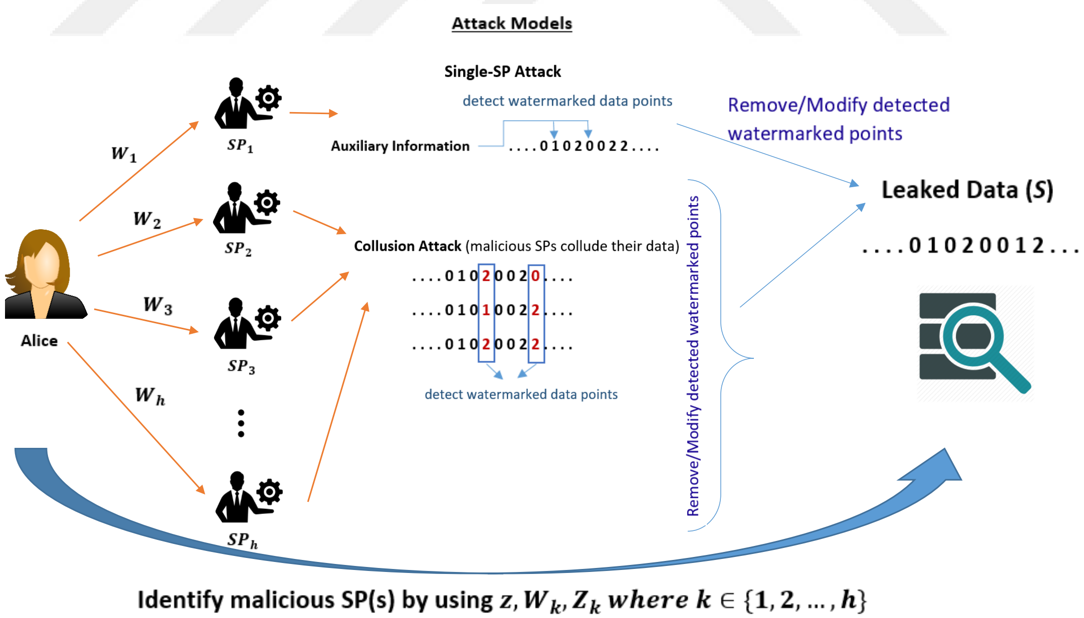


Figure 4.1: Overview of the system and attack models

4.1.4 Threat Model

In the threat model, the goal of the attacker is to share the data without being detected by the owner of the data. For this, we introduce different attack models where attacker tries to find watermarked points and remove/modify them before sharing.

Single SP Attack: In this attack, one malicious SP tries to find watermarked points to minimize the probability of detection before sharing. For this purpose, s/he uses the auxiliary information (such as correlations, MAFs and familial information in genomic data). For each point s/he finds the probability of observing each state; meaning probability of $x_i = y$, for $y \epsilon \{y_1, y_2, ..., y_z\}$. If the probability of having the shared value is lower than the probability of having different value, it indicates that this point may be watermarked. We assume that malicious SP knows the watermark length and select w points having lowest probability of shared value as "watermarked points".

Collusion Attack: Set of SPs that receive the same data portion from the same data owner with different watermark patterns may collude their data to identify the watermarked points. Note that the portion of shared data with different SPs may differ in real life; we assume it is the same in our scenario for construction of a stronger model. In this 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 watermarked data points. Then, SPs may also use auxiliary information to detect further watermarked points.

After finding identified watermarked data points by single SP or collusion attack, SPs choose one the following ways:

- Remove all identified points and share the data.
- Remove fraction (r percent) of these identified points and share the data.
- Remove r percent and modify (add noise) remaining p percent of these identified points.
- Modify $w \times b$ points where $b \epsilon 1, 2, ..., 15$.

Aforementioned attack models are further explained in the use case with real data and auxiliary information that can be used in section 4.3.

4.1.5 Objective (Detection of Malicious SP(s))

The objective of the model is finding the source of leakage. After the leakage, Alice observes the leaked data; finds removed points and compares with the real data to find noisy points. Note that modified points may share the same state with the real data and in this case, Alice cannot differentiate the watermark pattern for that data point. After finding the noise and removed data, Alice chooses the SPs having the pattern for those points and mark them as malicious SPs. As the watermark pattern is unique for each SP, if Alice can identify the noisy point for sure, she can easily detect which SP has that data point in its watermark pattern. i.e. If Alice shares her genomic data with 4 SPs (SP_1, SP_2, SP_3, SP_4), watermark length is 1 and x_i, x_j, x_k, x_m are watermarked respectively. (First SP receives data where only x_i is watermarked, second SP receives data where only x_j is watermarked etc.) If one malicious SP from the set (SP_1, SP_2, SP_3, SP_4) modifies x_m by following one of the attacks explained in Section 4.1.4 before sharing, two interpretations are possible:

- Modified x_m has the same state with the real data $\rightarrow$ Alice cannot make any inference.
- Modified x_m has different state than the real data $\rightarrow$ Alice marks SP_4 as malicious as the watermark is unique. That is, all other SPs have real state for x_m and no other SP can have this modified value. In this case, Alice's decision could be true positive (TP) or false positive (FP) depending on attacker. If SP_4 is the source of leakage, the decision is TP. However, SP_1 could fix the value of x_i, and modify x_m which causes a FP decision.

4.2 Proposed Solution

Here, we first give the preliminaries that will facilitate the proposed solution. Then, we give the details of the proposed method.

4.2.1 Preliminaries

In this section, we introduce the belief propagation algorithm, and some genomics concepts.

4.2.1.1 Belief Propagation

Belief propagation is based on a message-passing strategy for performing efficient inference on high dimensional data on graphical models such as factor graphs or Bayesian networks [49]. It is typically used to compute marginal distributions of unknown variables. Belief propagation is widely used since computation of marginal distributions may require summing large number of terms; making the process computationally hard and complex. In a factor graph, there are two types of nodes: (i) Factor Nodes, and (ii) Variable Nodes which are explained in Section 4.2.2.

4.2.1.2 Genomics

Here, we briefly introduce some genomics concepts including single nucleotide polymorphism, reproduction, and linkage disequilibrium which are used throughout the use-case and evaluation sections.

Single Nucleotide Polymorphism (SNP): SNP is major contributor of human genetic variation where single nucleotide having A, T, G, or C values differs between members of the population. SNPs form 80% of of all known polymorphisms [50]. There are usually two different nucleotides which are called alleles observed at a given SNP position: (i) the major allele which is the most frequently observed nucleotide in the population, and (ii) the minor allele which is the rare nucleotide, generally causing difference in the phenotype. In this work, for a specific SNP position, we represent the major allele as B, and the minor allele as b (where both B and b are in A, T, G, C). In addition, for simplicity of calculations we use the following notation:

- homozygous-major genotype (BB) denoted as 0.
- heterozygous genotype (Bb) denoted as 1.
- homozygous-minor genotype (bb) denoted as 2.

Reproduction: The Law of Segregation which is the first law of Mendel, states that two alleles for each trait segregates during meiosis and child independently gets one allele from mother and the second from the father. According to this law, each allele inherited with the probability 0.5 from mother and father. General inheritance probabilities for the child is given in Table 4.2. Each table entry shows probability of observing phenotypes BB, Bb, and bb respectively. We use this model in showing application of the proposed scheme on genomic data and related attack models.

		Father Genotype (F)		
		BB	**Bb**	**bb**
Mother Genotype(M)	**BB**	(1,0,0)	(0.5,0.5,0)	(0,1,0)
	Bb	(0.5,0.5,0)	(0.25,0.5,0.25)	(0,0.5,0.5)
	bb	(0,1,0)	(0,0.5,0.5)	(0,0,1)

Table 4.2: Mendelian inheritance probabilities of a child given the genotypes of the parents. Each table entry shows probability of observing phenotypes BB, Bb, and bb respectively.

Linkage Disequilibrium (LD): Linkage disequilibrium [51] is the notion of having non-random association between any pair of SNPs due to recombination, natural selection, genetic drift, and mutation. Because of LD correlation, if there is a high correlation between a pair of SNPs, inferring the state of one SNP, given the state of the correlated one is possible (with high probability). This concept forces the consideration of LD correlated SNPs and their values in genomic privacy researches since the attacker could use this information to infer the values of the SNPs. We calculated the LD values on our dataset based on the correlation between each pair of SNPs and each pair of values. Let $F(SNP_i)$ represent the frequency of observing the of SNP_i in the population. Using this, $P(SNP_i)$ is calculated as:

$$P(SNP_i) = \begin{cases} 0, & \text{if } F(SNP_{i-1}) = 0. \\ F(SNP_i)/F(SNP_{i-1}), & \text{if } F(SNP_{i-1}) > 0. \end{cases} \tag{4.1}$$

4.2.2 Method

Our proposed solution employs belief propagation algorithm to consider all auxiliary information, optimize utility, and maximize the detection of malicious SPs. Belief propagation algorithm is a well-known message passing algorithm for computing the marginal distributions since it provides efficient inference from high dimensional data. Dimensions represent the relationship between variable and factor nodes. The algorithm is based on exchanging messages between nodes iteratively. The main idea is to represent probability distribution of variable nodes by factorization into products of local functions in factor nodes. As discussed, the reason that we use belief propagation algorithm over naive factorization is that this problem may require summing large number of terms for the computation of marginal distributions. Thus, to reduce the computation, we represented the problem as a factor graph.

4.2.3 Nodes and Messages

In this section, we briefly explain the variable nodes, factor nodes, and messages between them. We also give the general rules of belief propagation algorithm.

4.2.3.1 Variable Nodes

Decision nodes (variable nodes) represent the unknown variables (data points), and in each variable node we decide on the probability of watermarking that point. That is, for each node we have the marginal probability distribution of states $y_1, y_2, ..., y_z$. Each variable node x_i represent the marginal probability distributions of i^{th} unknown variable. The message $\mu^v_{i \to k}$ from variable node x_i to factor node k denotes the probability of $x_i = y$, for $y \epsilon \{y_1, y_2, ..., y_z\}$ at v^{th} iteration of message passing algorithm. Note that in our algorithm, the probability distributions show the probability of watermarking that point. To be more precise, at the end of message passing algorithm, we sort all probability values, we check if the highest probability states are different from real value of the data, and if so, we watermark that point with the highest probability state. For example, if the variable node x_i has 3 states, namely y_1, y_2, y_3 and the algorithm terminates with $P(x_i = y_1) = 0.6$, $P(x_i = y_2) = 0.2$, $P(x_i = y_3) = 0.2$, we watermark that point as y_1 as it has the highest probability (in case where the real value is different from y_1, meaning either y_2 or y_3).

4.2.3.2 Factor Nodes

These nodes represent the functions of factorized joint probability distribution of variable nodes. One example of factor nodes could be correlation nodes that shows the correlation between variable nodes. Thus, the variable node x_i is connected to correlation node c_{ik} together with variable node x_k if there is a correlation between nodes x_i and x_k. The message $\mu^v_{i \to k}$ from factor node c_{ik} to variable node x_k denotes $P(x_k = y)$, for $y \epsilon \{y_1, y_2, ..., y_z\}$ given the correlation value and the probability that $x_i = y$, for $y \epsilon \{y_1, y_2, ..., y_z\}$ at v^{th} iteration of message passing algorithm.

Detailed explanation of the nodes and messages is given in Section 4.3 through an application of the algorithm. General rules of belief propagation algorithm are as follows:

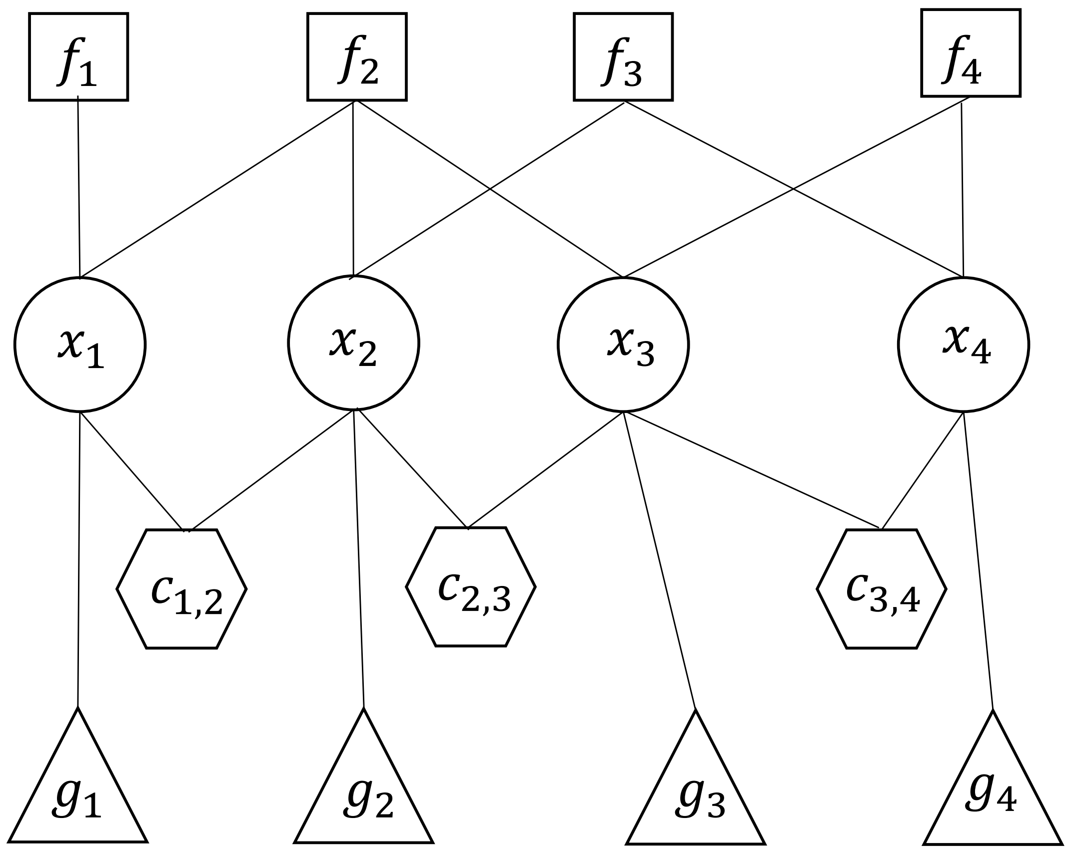


Figure 4.2: Factor graph representation with 3 types of factor nodes: f_i, $c_{i,j}$, and g_i. Variable nodes are denoted as x_i.

- The algorithm starts in variable node with initial probability distribution.
- The algorithm iteratively exchanges messages between the connected nodes for updating the probability distributions of the targeted unknown variable nodes at each iteration, until convergence.
- At each iteration, variable nodes generate the factor node messages by multiplying all incoming messages from the neighbours except the receiver neighbour.
- At each iteration, factor nodes generate the messages by using local functions and send it to necessary variable nodes.

- At the end of each iteration, marginal probability distribution of each variable node is updated by multiplying all incoming messages from neighbours.
- The algorithm is terminated when beliefs on the variable nodes converges or after fixed number of iterations.

General factor graph representation of the algorithm is given in Figure 4.2. This graph has 3 types of factor nodes connected to corresponding variable nodes denoted as x_i. Each factor node may represent a different type of relationship about the variable nodes depending on the application.

4.3 Use Case: Genomic Data

In this section, we briefly introduce a real-life application of the proposed scheme using genomic data. As our dataset is composed of SNP values as $0, 1, 2$, the variable nodes show the marginal distributions of the these SNP values. We used 3 types of factor nodes: (i) Familial Nodes (g nodes), (ii) Attack and Utility (f_g) Nodes, and (iii) Correlation Nodes (c nodes). Messages between nodes and connections are briefly explained below.

4.3.1 Messages

Below the notation for messages between factor graph nodes are given:
$\mu^v_{i\to k}$: Message from variable node x_i to factor node k at the v^{th} iteration.
$\beta^v_{i\to k}$: Message from familial node g_i to variable node k at the v^{th} iteration.
$\delta^v_{i\to k}$: Message from f_g factor node i to variable node k at the v^{th} iteration.
$\lambda^v_{i\to k}$: Message from correlation factor node $c_{i,k}$ to variable node k at the v^{th} iteration.

4.3.2 Variable Nodes

In this application, variable nodes are used for selecting SNPs to be watermarked and the state of this watermark. Each variable node (decision node) x_i represents the marginal probability distributions of the SNP_i with states $\{0, 1, 2\}$. It calculates the multiplication of the probability values from neighbors and sends it to the connected attack and utility factor nodes (f_g) and correlation nodes ($c_{i,j}$) following the belief propagation rules. The message $\mu_{i \to k}^{v} P(x_i = y)$ from variable node i to factor node k indicates that $P(x_i = y)$ at v^{th} iteration where $y \epsilon \{0, 1, 2\}$. Initially, the $P(x_i^1 = y) = 0.33$ for every $y \epsilon \{0, 1, 2\}$ since the probability of watermarking that data point as 0,1, or 2 is equally likely at the start. Note that the variable nodes do not send any message to familial nodes since the information coming from familial node does not change and does not depend on the marginal distributions of the SNPs as discussed in Section 4.3.3. Below function is representation of the message from variable node i to correlation factor node g:

$$\begin{aligned} \mu_{i \to g}^{v} P(x_i = y) = & \frac{1}{Z} \times \beta_{z \to i}^{v-1} P(x_i = y | m_i, f_i) \times \\ & \delta_{s \to i}^{v-1} P(x_i = y) \times \\ & \prod_{k=1, k \neq g}^{n} \lambda_{k \to i}^{v-1} P(x_i = y) \end{aligned} \tag{4.2}$$

where Z is the normalization constant as the sum of probability distributions for each SNP should be 1.

4.3.3 Familial Factor Nodes

Familial factor nodes g_i calculates the message $\beta_{i \to k}^{v} P(x_i = y | m_i, f_i)$ for $y \epsilon \{0, 1, 2\}$ using the probabilities given in Table 4.2 and sends it to variable node x_i where m_i and f_i indicates the i^{th} SNP values of mother and father. To give an example, if mother has 0 and the father has 0 for the corresponding SNP, $P(x_i = 0 | m_i = 0, f_i = 0) = 1$. Note that the messages of familial nodes are computed beforehand, as it does not recieve any message in iterations.

4.3.4 Attack and Utility Factor Nodes

These nodes are indicated as f_g nodes since they are grouped together according to similar MAF values. The variable nodes between some MAF threshold is connected to one f_g node as indicated in Figure 4.3. The term utility is used to ensure that the data points that are modified in watermarking are the less informative points in general, meaning the points that have higher MAF values. These nodes are responsible for maximizing the utility by multiplying the overall probability of the state with the corresponding MAF values and minimizing the success probability of attacker which is defined as the probability of detection of a watermark point by collusion attacks. The intuition behind this calculation (see Equation 4.5) is the integrating the MAF values of the SNPs to the algorithm. If the MAF value is low, it means that the SNP is informative and we would like to favor uninformative ones since watermarking them does not decrease the utility too much.

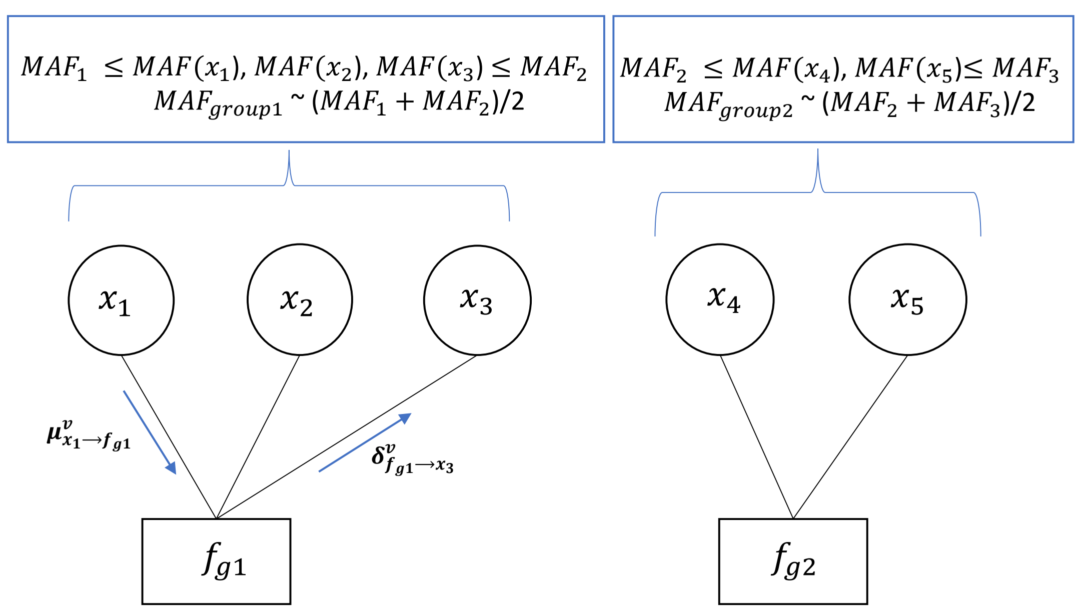


Figure 4.3: Representation of variable nodes connected to f_g nodes. Each node receives and sends message. Only one message from each message type is shown for simplicity.

The following notation will facilitate the discussion on these nodes.

- $v_{y,i}$ indicates the frequency of state y in the population for the SNP_i.
- $n^h_{y,i}$ indicates the number of occurrences of state y in the previous h sharings for SNP_i.
- $n^h_{\not{y},i}$ indicates the number of occurrences of states other than y in the previous h sharings for SNP_i.

The probability of having $SNP_i = y$ is proportional to MAF value of the SNP. That is, having the original value of the SNP in h sharing is calculated as:

$$P(x_i = y) = (MAF_i)^{n^h_{y,i}} \tag{4.3}$$

where MAF_i is MAF value of the i^{th} SNP. Similarly, having a value different than the original value of the SNP in h sharing is calculated as:

$$P(x_i \neq y) = (1 - MAF_i)^{n^h_{\not{y},i}} \tag{4.4}$$

The reason behind this is optimizing the utility according to MAF and the intuition behind this is the collusion attacks. Note that $n^h_{y,i} + n^h_{\not{y},i} = h$ in all cases. As the attacker knows that we maximize the utility based on the proposed algorithm, s/he can conclude that the data is watermarked proportional to the MAF values; as indicated in Equation 4.3 and 4.4. $n^h_{y,i}$ and $n^h_{\not{y},i}$ for $y \epsilon \{0, 1, 2\}$ values are found by the malicious attackers by colluding their data.

Based on the intuition that is explained above, these nodes computes $P(x_i = y|A\&U)$, where A and U stands for "Attack" and "Utility" respectively. This function is defined as:

$$\begin{aligned} P(x_i = y|A\&U) = v_{y,i} \times (MAF_{group})^{n^h_{y,i}} \times \\ (1 - MAF_{group})^{n^h_{\not{y},i}} \end{aligned} \tag{4.5}$$

Above formula guarantees the aforementioned higher utility and lower attack probabilities together. As f_g is responsible for more than one SNP, it calculates the above probability for using average MAF value in the group indicated as MAF_{group} and calculates this for the *real values* of the SNPs as we would like to minimize the success of the attacker in collusion attacks that is finding the real value of the SNP. Lastly, we also have multiplication with the frequency of real value of the data point to ensure that it is closer to real life case. With this, the points having 0 frequency for some states is never watermarked as that state because it is very easy to detect the watermark in these points. To give an example, if data point x_i is shared with 6 SPs as $0, 0, 1, 2, 1, 0$ respectively, there are following possibilities: (i) The data has real value 0 and watermarked 3 times, (ii) The data has real value 1 and watermarked 4 times, or (iii) The data has real value 2 and watermarked 5 times. Assuming the real value is 0, $n^h_{y,i} = 3$ and $n^h_{y,i} = 3$ in this case. As the SNPs having lower MAF values (rare SNPs) are more informative, we would like to watermark these points with least probability through multiplying with $(1 - MAF_{group})^{n^h_{y,i}}$.

4.3.5 Correlation Factor Nodes:

We use LD correlations in the genomic data to ensure that the algorithm is secure against correlation attacks; that is a malicious SP cannot make an inference about watermarked points by looking at the correlations, with a high probability. These correlations are computed for each pair of SNPs in the dataset beforehand and threshold η is used as a cutoff points (If the pair of SNPs have correlation value higher than η, they are counted as correlated SNPs). The correlation value may differ based on the state of each data point and correlation between the data points is typically asymmetric. These nodes are responsible for maintaining the correlations within the data and we have one correlation node for each pair of correlated SNPs. Message from i^{th} correlation node to j^{th} variable node, $\lambda^v_{i \to y}$ is defined as:

$$\lambda^v_{i \to j} P(x_j = y) = \gamma_{sj} * \mu^v_{s \to i} P(x_s = t) \quad y, t \epsilon \{0, 1, 2\}. \tag{4.6}$$

where x_s is the neighbour variable node and γ_{sj} denotes the correlation value of having the value y for SNP_j given that the value of SNP_s is t. Figure 4.4 shows a representation of connections between variable and correlation nodes. In this figure, if the probability of having 0 at x_1 is 0.4 and the correlation of the x_1 to x_2 for values (0,1) is 0.6, $\lambda_{c_{1,2} \rightarrow x_2}$ indicates $P(x_2 = 1|x_1 = 0) = 0.4 * 0.6 = 0.24$

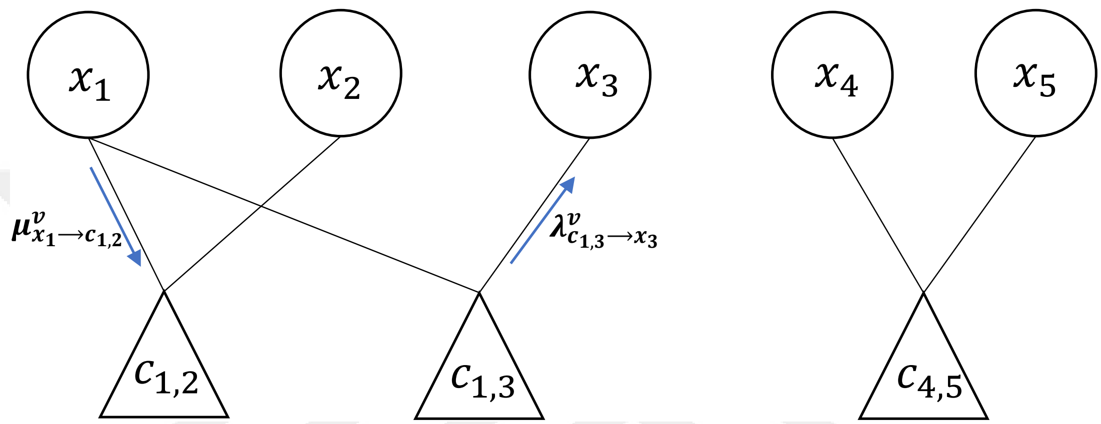


Figure 4.4: Representation of variable nodes connected to correlation nodes. Each node receives and sends message. Only one message from each message type is shown for simplicity.

At the end of the belief propagation algorithm, the probability distributions that are different from real value of the SNPs are sorted and top w points having highest probability distribution are watermarked with corresponding states. Note that the watermark pattern for each sharing is unique. Thus, in each watermarking step, we check all previous sharings and if the watermark pattern matches with one of the previous sharings, we adjust the pattern by flipping probabilities of the data points that have same pattern in previous sharing with the next highest state or point. The number of data points that are flipped depends on the output of the algorithm and our selection of a parameter denoted as m. The output of the algorithm changes according to attack probabilities computed in the f_g nodes. Thus, it may give the same watermark pattern for some points depending on the number of sharings. The parameter m shows the number of data points that is desired to have different pattern in the next sharing. When $m = w$ the whole pattern of watermark is changed in the next sharing, that is no data points is watermarked with the same state twice. If the output of the belief propagation algorithm gives n points having the same watermark pattern with previous sharing and $m > w - n$; then we flip the first $m - (w - n)$ points to change the pattern where $w - n$ shows the number of watermarked points having different patterns in the previous sharings. We make sure that the pattern is not same with any of the previous patterns by repeating this flipping process for each sharing until it has a unique pattern.

4.4 Evaluation

As mentioned, we implemented the proposed scheme on genomic data. In this section, we first provide the details of data model, experimental setup, and evaluation metrics that is used in evaluation of the proposed scheme. Then, we give the results (utility and robustness) of different experiments.

4.4.1 Data Model and Experimental Setup

We worked on a SNP data of 99 individuals from 1000 Genomes Project [52]. In the obtained dataset, each individual has 7690 SNP values meaning that we have a 99 by 7690 matrix showing the SNP values denoted by 0 (homozygous-major genotype), 1 (heterozygous genotype), or 2 (homozygous-minor genotype). MAF values and LD-correlations calculated on this dataset. We specified the correlated pairs of SNPs based on a correlation threshold specified as $\eta = 0.9$ for our experiments. As our proposed method has familial nodes; selecting 2 individuals from this dataset, we generated a family by inferring the child's SNP values based on the work proposed in [53]. After this process, we took genomes of 20 family composed of father, mother, and child, conducted all experiments on these 20 family and take the average of precision, recall, and entropy values. The process of synthetic family generation does not change the result of the application since it also uses the Mendelian probabilities given in Table 4.2. Thus, the result of synthetic family generation is compatible with the information that we employ in the algorithm and attacks.

In addition, to show the affect of auxiliary information and factor nodes used in belief propagation algorithm, we conducted every experiment with and without familial nodes and performed all of the aforementioned attacks using familial auxiliary information.

4.4.2 Evaluation Metrics

For the evaluation, we calculated precision, recall, and entropy values based on the predicted set of SPs. In the experiments, we selected random set of malicious SPs for the collusion attacks, and random SP for single SP attack. We tried to detect malicious set based on watermark patterns. For detection, Alice compares her data with the leaked data and finds the data points that differs or removed. By looking at the watermark patterns for each SP, Alice decides the source of leakage. Thus, any SP with partial watermark pattern matching the removed/modified data point is added to set of malicious SPs. Based on this prediction, we calculated the uncertainty of Alice about the source of leakage (entropy). Let SP_ϕ be the source of leakage. We quantify the uncertainty of Alice by computing the entropy as $H = -\sum_{i=1}^{h} P(SP_\phi = SP_i) \log P(SP_\phi = SP_i)$ assuming Alice knows the number of colluding SPs.

Let TP, FP, FN, and TN denote the number of true positives, false positives, false negatives, and true negatives in the predicted set respectively.Precision, recall, and accuracy values are computed as $\frac{TP}{TP+FP}$, $\frac{TP}{TP+FN}$, $\frac{TP+TN}{TP+FP+FN+TN}$ respectively.

4.4.3 Results

We evaluated the proposed scheme under attacks that are explained in Section 4.1.4. The robustness of the method is evaluated against collusion (+auxiliary information) and modification attacks. For all experiments we set $h = 20$ meaning the data is shared 20 times and give the results for $2, 4, 6, ..., 18$ colluding malicious

SPs for simplicity. In these experiments, we make the following assumptions which provides the maximum amount of information to the malicious SPs and stronger attack models:

- Watermark length w is known by malicious SPs.
- Alice shares the same data portion with all SPs.
- Malicious SPs have access to all population statistics including LD correlation values, MAF values, and frequencies of all states for all SNPs.
- Malicious SPs have the familial information (All SNPs of mother and father).

In addition, in all results where malicious SP adds noise to watermarked position, we assume that s/he modifies to point compatibly with the overall frequency in the population. To give an example if SNP_i has frequencies 0.8, 0.1, 0.1 for states $0, 1, 2$ respectively SPs mark this watermarked point as 0 to be realistic.

4.4.3.1 Single SP Attack

In this attack, one SP tries to find set of watermarked points using the auxiliary information coming from family, correlations, and MAF values. The attacker does the following for finding and modifying/removing watermarked positions:

- Using the mother and father genomic information and Mendelian inheritance probabilities, the attacker finds the probability distribution for each SNP x_i.
- Checks the LD correlations.
- Detects watermarked points based on these probabilities. For example, if both mother and father has 0 for SNP x_i, the probability of having $P(x_i = 1) = 0$ and if the watermarked point has 1, the attacker marks that point

as watermarked rigorously. Similarly, if the correlation values indicate high correlation such as $P(x_i = 1|x_j = 1) = 0.98$ and the shared data indicates $x_i = 1$ and $x_j = 2$, attacker marks x_j as watermarked.

- Based on probability distributions, attacker chooses top w points and removes or modifies them compatible with the familial information and frequencies in the population.
- As the attacker knows watermark length w, if s/he could not find w points compatible with familial or correlation information, the remaining points are selected based on highest MAF as the attacker knows watermarking algorithm and the algorithm will favor the lower MAF points as they are more informative.
- When the malicious SPs remove r percent of the watermarked points and modify p percent, we assume that it selects top r percent having the *highest* MAF value to remove, and remaining p percent to modify.

The accuracy in detection of source of leakage when single SP shares the data is 100% with 0 entropy in both cases (with and without familial node usage in belief propagation algorithm) where malicious SP *removes* some portion of the detected watermarked points. The reason is that at least small portion of watermark pattern remains in the shared data when the watermark length is at least 10 (0.01% of the data as used in all experiments). However, modifying too much points decreases the precision and it also decreases the utility of the data too much. The reason that modification drops the precision whereas removing does not affect it too much is selected scheme for the attack. As the attacker modifies data points using highest frequencies in the population, the detection becomes harder. Figure 4.5 shows the precision results as the attacker increases the number of modified points; that is, decreases the utility (by increasing data loss). Here, by utility we imply the data loss, that is we assume all points are equally informative. As the data size is 1000, when $15w$ points modified for $w = 40$, the utility drops to 40% (assuming all SNPs have equal importance) since 600 points modified out of 1000 points. Note that there is a significant drop in the precision where $w = 30$ or $w = 40$. This is because the number of added noise also significantly increases in these cases. It is expected that when attacker decreases the utility too much by increasing the number of noisy points, the detection of malicious SP becomes harder. For all experiments given in Figure 4.5, recall value was always 1 since we successfully detected the malicious SP in all predicted sets but we suspected other SPs too, causing predicted set to become larger and precision to drop.

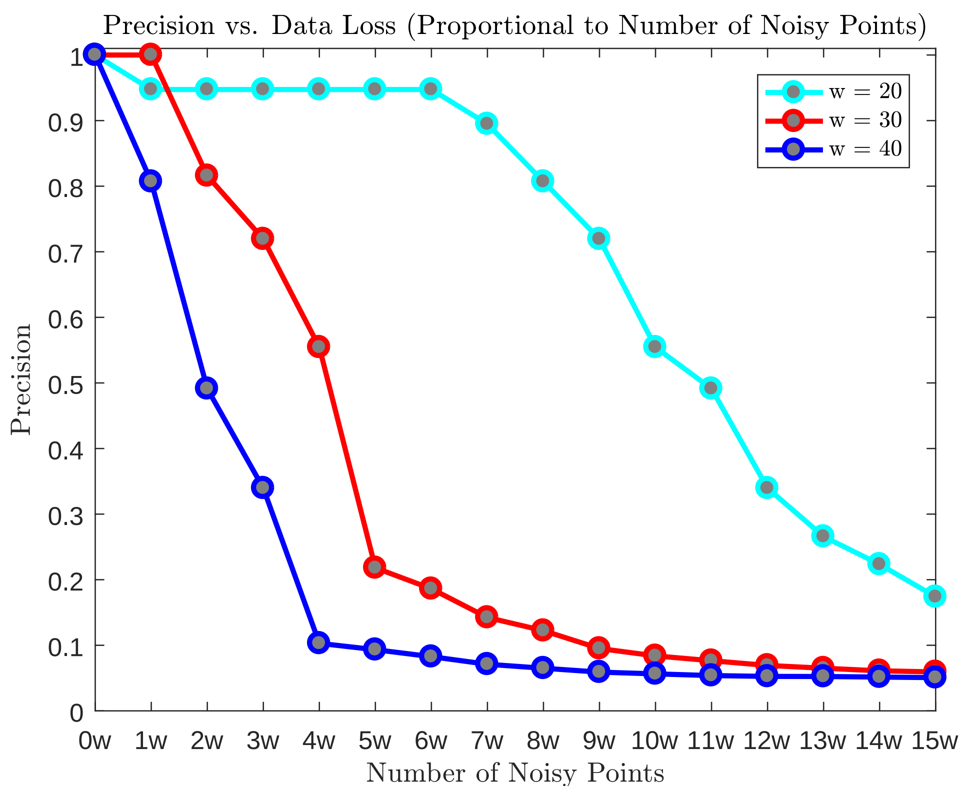
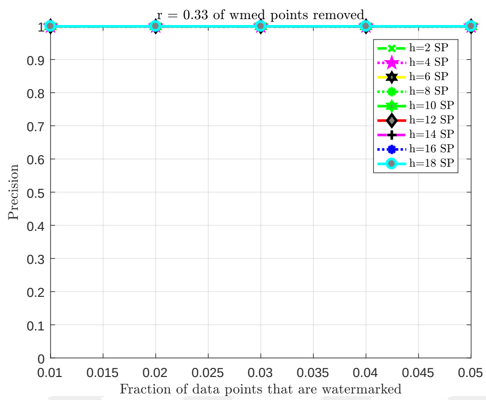


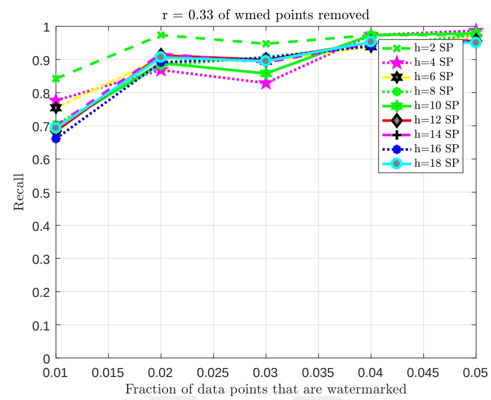
Figure 4.5: Precision vs. Data Loss results, x-axis shows the number of noise added points by the attacker as proportional to watermark length(w); data loss is proportional to number of noisy points.

4.4.3.2 Collusion Attacks

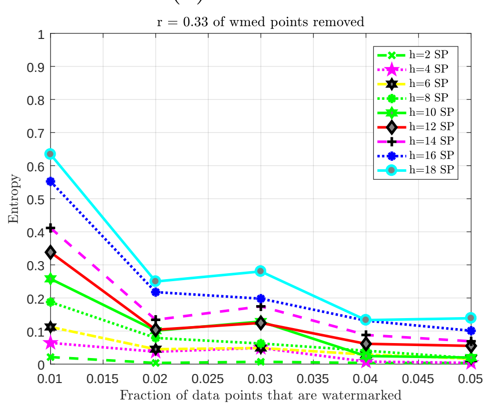
In these attacks, as described in Section 4.1.4, number of malicious SPs collude and vertically align their data. The points that differ in positions are marked as



(a) Precision

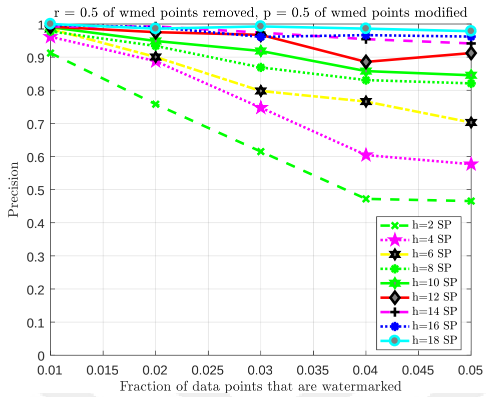


(b) Recall

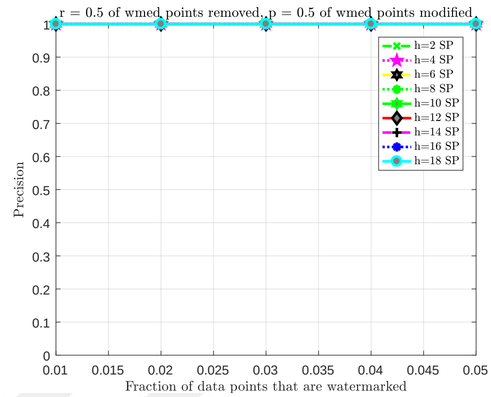


(c) Entropy

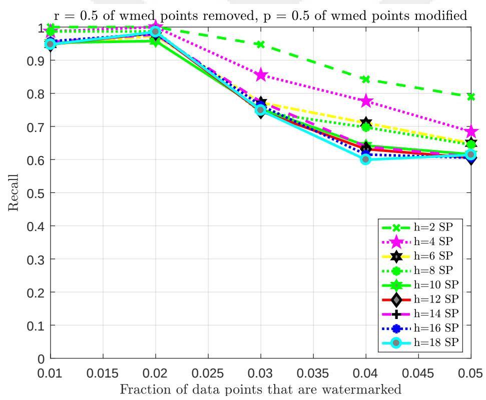
Figure 4.6: Precision, Recall, and Entropy results when malicious set of SPs remove 33% of watermarked points and familial nodes are used in belief propagation algorithm.



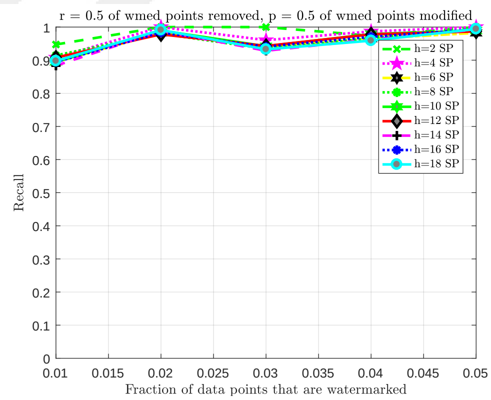
(a) Precision without familial nodes



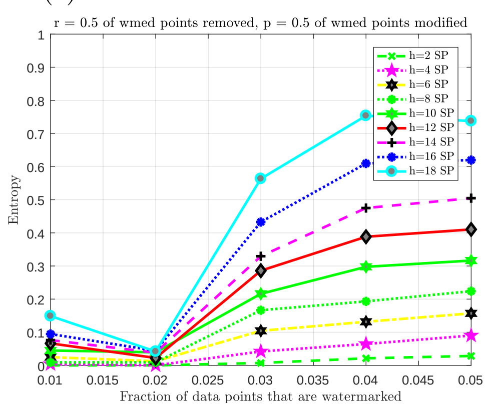
(b) Precision with familial nodes



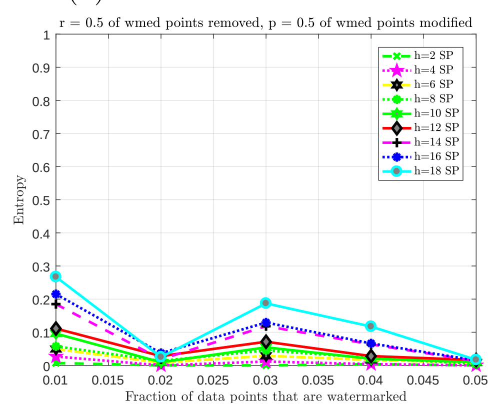
(c) Recall without familial nodes



(d) Recall with familial nodes



(e) Entropy without familial nodes



(f) Entropy with familial nodes

Figure 4.7: Precision,Recall, and Entropy results when malicious set of SPs remove 50% of watermarked points and modify remaining 50%. 4.7a, 4.7c, 4.7e shows the results when familial nodes are excluded from belief propagation algorithm and 4.7b, 4.7d, 4.7f shows the results when familial nodes included.

watermarked points. As the watermark pattern is unique for all SPs, when they collude the *maximum* number of points found as watermarked is (hxw) where h is the number of SPs that are colluding, w is the watermark length. When the number of points found from collusion attack is less than (hxw), SPs try to find additional watermarked points based on auxiliary information similar to single SP attack. When set of malicious SPs remove all found watermarked points and do not use any modification on the data points, the accuracy is still 100%. The results for one extreme case where malicious SPs remove only small portion (33% of watermarked points) is given in Figure 4.6.

Figure 4.7 shows the precision, recall, and entropy results of collusion attacks where the malicious set of SPs remove 50% of detected watermarked points and modify(add noise) to remaining 50%. The left column shows the results without use of familial nodes and the right column shows the results with familial nodes.

As it can be seen in Figure 4.7, the familial nodes increased the success of Alice in detection of the source of leakage dramatically as this auxiliary information is considered in belief propagation algorithm with family factor nodes. We also tried removing all watermarked points, modifying all watermarked points, and with different r and p values (0.25, 0.33, 0.5, 0.6, 0.7 and 0.75). As the trend of the graphs are similar, they are not included. However, in case of using familial factor nodes detection of source of leakage is always 100% accurate when malicious SPs remove all watermarked points, and remove 80% and modify 20% cases. After this point, if attacker increases the number of points that are modified and decrease number of removed points, the precision drops slightly since modification is very compatible with population statistics and familial information. In the worst case, the minimum precision value found was 0.61 where all watermarked points are modified by attacker.

4.5 Discussion

In this section, we discuss the given results, potential applications, and future directions.

Results: As it can be seen in previous section, the results are better in the use of auxiliary information in belief propagation algorithm. In the cases where adversarial model removes all or some fraction of watermarked points, the precision results are still very high as even small portion of remaining watermark pattern allows Alice to find source of leakage with the highest certainty. When some portion of watermarked points are modified; however, precision or recall values start to fall. The reason for that is noisy points may indicate some other watermark pattern or the real value in the original data.

Another thing to mention about result graphs is that the small peaks in some points. An example for this is the small increase in point where 0.03 of the data is shared (see Figure 4.6b and 4.6c). The reason behind this is increasing watermark length sometimes reveals better information about watermarked points to adversary. As the watermark pattern is unique for each SP and it depends on MAF, familial information, and correlations; increasing watermark length after this point indicates more points fall into next MAF window (factor nodes that are grouped together). The watermarked points with highest MAF when 0.03 of data is watermarked mostly in the same window; meaning adversary can infer them better and added noise hides the true pattern correctly. One solution for that is increasing number of f_g nodes and adjusting the average MAF value (thus the multiplier of local function).

Usability: We propose a scheme for detection of malicious SPs sharing the data without consent by minimizing the utility loss. This scheme could be outsourced for the potential use. It is also extendable to any sequential data with auxiliary information to be used in belief propagation algorithm.

Future Directions: For the privacy of genomic data, we included limited number of factor nodes; meaning limited auxiliary information. However, adversary model could use additional information such as phenotype of the individual. Thus, the method could be extended and phenotype factor nodes could be used. Further, integrating differential privacy concept [54] and making the watermarks naturally act like noise could be investigated in the future.

Chapter 5

Conclusion

In this work, we tried to address the protection of health and genomic data. We have proposed the use of homomorphic encryption for allowing secure and privacy preserving operations on health data. The results show that with a little computational burden, the use of homomorphic encryption is practical in real-life. For the future, more complex queries may require the use of new protocols. We used Paillier cryptosystem for this work and the correctness of the algorithm matched the ground truth which is taken as the result of computed queries on unencrypted data.

We have also proposed a novel watermarking scheme based on belief propagation algorithm to mitigate the liability issues arising from unauthorized sharings. The proposed system tries to minimize utility loss and it is applicable to any sequential data with a little range of data values and existing auxiliary information. The results of different attacks showed that the data owner can identify the source of leakage with high probability. We believe that the proposed scheme may serve as deterrence for malicious parties.

Bibliography

[1] M. M. Merener, "Theoretical results on de-anonymization via linkage attacks," *Transactions on Data Privacy*, vol. 5, pp. 377–402, 2012.

[2] P. Samurai and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," *Technical Report, SRI International*, 1998.

[3] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications.* CRC Press, 1 ed., 2004.

[4] N. Li and T. Li, "t-closeness: Privacy beyond k-anonymity and l–diversity," *Proceedings of the 32nd international conference on very large databases*, pp. 139–150, 2006.

[5] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, 2007.

[6] Somy.M.S, Gayatri.K.S, and Ashwini.B, "Privacy preserving health data mining," *International Journal of Computer Science and Technology*, vol. 6, no. 4, pp. 52–56, 2015.

[7] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques.," *In Proceedings of the Third IEEE International Conference on Data Mining*, pp. 1–9, 2003.

[8] S. Oliveira and O. Zaane, "Protecting sensitive knowledge by data sanitization.," *In Proceedings of the Third IEEE International Conference on Data Mining*, pp. 613–616, 2003.

[9] E. Ayday, J. L. Raisaro, J. Hubaux, and J. Rougemont, "Protecting and evaluating genomic privacy in medical tests and personalized medicine," *In Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society (WPES '13)*, pp. 95–106, 2013.

[10] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.

[11] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays," *PLOS Genetics*, vol. 4, pp. 1–9, 08 2008.

[12] M. Humbert, E. Ayday, J.-P. Hubaux, and A. Telenti, "Addressing the concerns of the lacks family: Quantification of kin genomic privacy," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, (New York, NY, USA), pp. 1141–1152, ACM, 2013.

[13] M. Blanton and M. Aliasgari, "Secure outsourcing of dna searching via finite automata," in *Data and Applications Security and Privacy XXIV* (S. Foresti and S. Jajodia, eds.), (Berlin, Heidelberg), pp. 49–64, Springer Berlin Heidelberg, 2010.

[14] Y. Chen, B. Peng, X. Wang, and H. Tang, "Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds," 01 2012.

[15] J. R. Troncoso-Pastoriza, S. Katzenbeisser, and M. Celik, "Privacy preserving error resilient dna searching through oblivious automata," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, (New York, NY, USA), pp. 519–528, ACM, 2007.

[16] E. Ayday, J. L. Raisaro, J. Hubaux, and J. Rougemont, "Protecting and evaluating genomic privacy in medical tests and personalized medicine," *In Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society (WPES '13)*, pp. 95 – 106, 2013.

[17] P. Baldi, R. Baronio, E. D. Cristofaro, P. Gasti, and G. Tsudik, "Countering gattaca: Efficient and secure testing of fully-sequenced human genomes," *CoRR*, vol. abs/1110.2478, 2011.

[18] N. Karvelas, A. Peter, S. Katzenbeisser, E. Tews, and K. Hamacher, "Privacy-preserving whole genome sequence processing through proxy-aided oram," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, (New York, NY, USA), pp. 1–10, ACM, 2014.

[19] Z. Huang, E. Ayday, J.-P. Hubaux, J. Fellay, and A. J. Genoguard, "Protecting genomic data against brute-force attacks," in *Proceedings of IEEE Symposium on Security and Privacy*, 2015.

[20] Y. A. A. S. Aldeen, M. Salleh, and M. A. Razzaque, "A comprehensive review on privacy preserving data mining," *SpringerPlus2015*, vol. 4, 2008.

[21] H. W. Jint, "Practical issues on privacy-preserving health data mining," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 64–75, 2007.

[22] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J.-P. Hubaux., "Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data.," *In Proceedings of the 2013 USENIX Conference on Safety, Security, Privacy and Interoperability of Health Information Technologies*, 2013.

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

[24] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2 ed., 2008.

[25] J. A. Bloom, I. J. Cox, T. Kalker, J.-P. Linnartz, M. L. Miller, and C. Traw, "Copy protection for dvd video," vol. 87, pp. 1267–1276, 1999.

[26] M. Maes, T. Kalker, J.-P. Linnartz, J. Talstra, F. Depovere, and J. Haitsma, "Digital watermarking for dvd video copy protection," *IEEE Signal Processing Magazine*, vol. 17, pp. 47–57, 2000.

[27] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 10, pp. 643–649, 2001.

[28] F. Huo and X. Gao, "A wavelet based image watermarking scheme," in *2006 International Conference on Image Processing*, pp. 2573–2576, Oct 2006.

[29] B. Chen, G. Coatrieux, G. Chen, X. Sun, J. L. Coatrieux, and H. Shu, "Full 4-d quaternion discrete fourier transform based watermarking for color images," *Digital Signal Processing*, vol. 28, pp. 106 – 119, 2014.

[30] M. Asad, J. Gilani, and A. Khalid, "An enhanced least significant bit modification technique for audio steganography," in *International Conference on Computer Networks and Information Technology*, pp. 143–147, July 2011.

[31] H. O. Oh, J. W. Seok, J. W. Hong, and D. H. Youn, "New echo embedding technique for robust and imperceptible audio watermarking," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 3, pp. 1341–1344 vol.3, 2001.

[32] C.-C. Chang, J.-Y. Hsiao, and C.-S. Chan, "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," *Pattern Recognition*, vol. 36, no. 7, pp. 1583 – 1595, 2003.

[33] "Image hiding by optimal lsb substitution and genetic algorithm," *Pattern Recognition*, vol. 34, no. 3, pp. 671 – 683, 2001.

[34] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, pp. 1423–1443, May 2001.

[35] V. Bhat K, I. Sengupta, and A. Das, “An audio watermarking scheme using singular value decomposition and dither-modulation quantization,” *Multimedia Tools and Applications*, vol. 52, pp. 369–383, Apr 20110.

[36] J. Brassil, S. Low, and N. Maxemchuk, “Copyright protection for the electronic distribution of text documents.,” *Proceedings of the IEEE*, vol. 87, pp. 1181–1196, 1999.

[37] J. Brassil, S. Low, N. Maxemchuk, and L. O’Gorman, “Hiding information in document images,” *Proceedings of Conference on Information Sciences and Systems*, pp. 482–489, 1994.

[38] M. J. Atallah, C. J. McDonough, V. Raskin, and S. Nirenburg, “Natural language processing for information assurance and security: An overview and implementations.,” *Proceedings of the 2000 Workshop on New Security Paradigms*, pp. 51–65, 2000.

[39] M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik., “Natural language watermarking: Design, analysis, and a proof-ofconcept implementation,” *Proceedings of 4th International Workshop on Information Hiding*, 2001.

[40] M. Topkara, U. Topkara, and M. J. Atallah, “Words are not enough: Sentence level natural language watermarking,” in *Proceedings of the 4th ACM International Workshop on Contents Protection and Security*, MCPS ’06, (New York, NY, USA), pp. 37–46, ACM, 2006.

[41] D. Boneh and J. Shaw, “Collusion-secure fingerprinting for digital data,” in *Advances in Cryptology — CRYPT0’ 95* (D. Coppersmith, ed.), (Berlin, Heidelberg), pp. 452–465, Springer Berlin Heidelberg, 1995.

[42] D. Heider and A. Barnekow, “Dna watermarks: A proof of concept,” *BMC Molecular Biology*, vol. 9, p. 40, Apr 2008.

[43] M. Liss, D. Daubert, K. Brunner, K. Kliche, U. Hammes, A. Leiherer, and R. Wagner, “Embedding permanent watermarks in synthetic genes,” *PLOS ONE*, vol. 7, pp. 1–10, 08 2012.

[44] S. Iftikhar, S. Khan, Z. Anwar, and M. Kamran, "Geninfoguard–a robust and distortion-free watermarking technique for genetic data," *PLOS ONE*, vol. 10, 02 2015.

[45] F.-z. Boujdad and M. Südholt, "Constructive Privacy for Shared Genetic Data," in *CLOSER 2018 - 8th International Conference on Cloud Computing and Services Science*, Proceedings of CLOSER 2018, (Funchal, Madeira, Portugal), pp. 1–8, Mar. 2018.

[46] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 1592, pp. 223–238, 1999.

[47] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[48] F. Kerschbaum, D. Biswas, and S. de Hoogh, "Performance comparison of secure comparison protocols," *Database and Expert Systems Application, 2009. DEXA '09. 20th International Workshop on*, 2009.

[49] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[50] A. A. Komar, *Single Nucleotide Polymorphisms: Methods and Protocols*, vol. 578. Humana Press, 2009.

[51] D. S. Falconer and T. F. Mackay, *Introduction to Quantitative Genetics(4th edition)*. Harlow, United Kingdom: Pearson Education Limited, 1996.

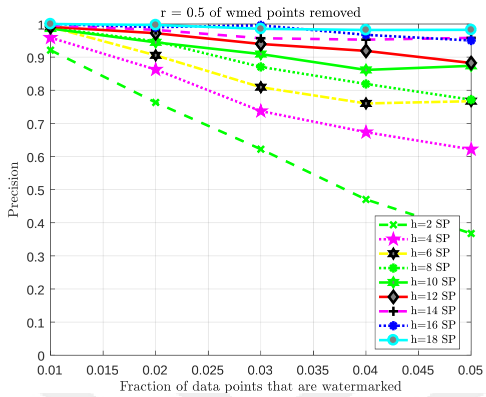
[52] "1000gp phase 1 haplotype." `https://mathgen.stats.ox.ac.uk/impute/1000GP_Phase3.html`. Accessed: 2017.

[53] I. Deznabi, M. Mobayen, N. Jafari, O. Tastan, and E. Ayday, “An inference attack on genomic data using kinship, complex correlations, and phenotype information,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2017.

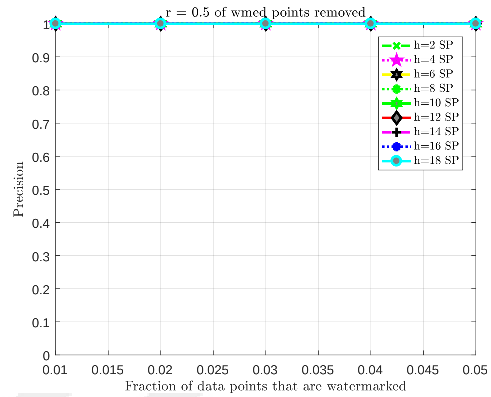
[54] C. Dwork, *Differential Privacy*, pp. 338–340. Boston, MA: Springer US, 2011.

Appendix A

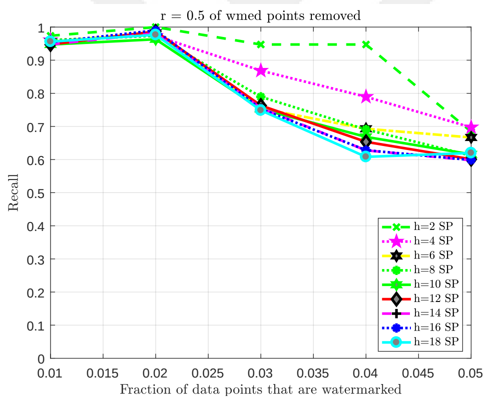
Supplementary Figures



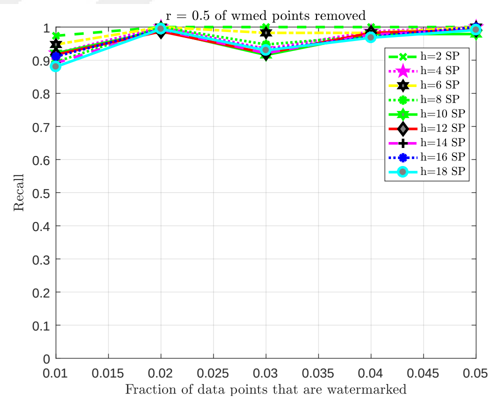
(a) Precision without familial nodes



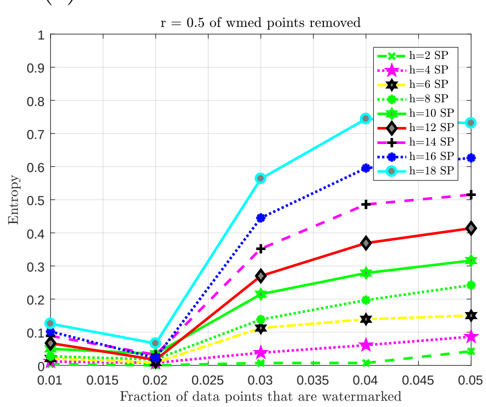
(b) Precision with familial nodes



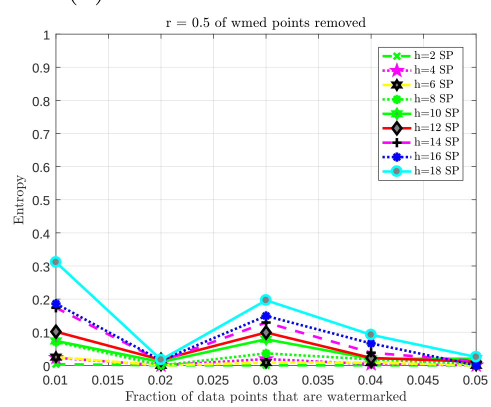
(c) Recall without familial nodes



(d) Recall with familial nodes

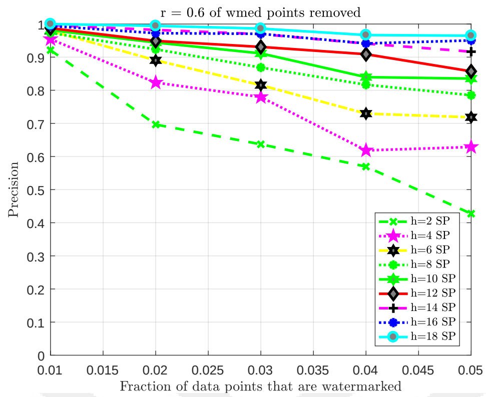


(e) Entropy without familial nodes

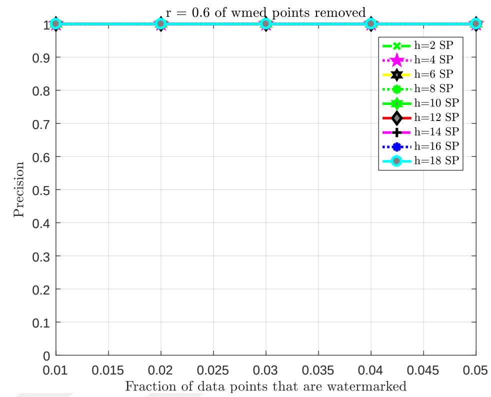


(f) Entropy with familial nodes

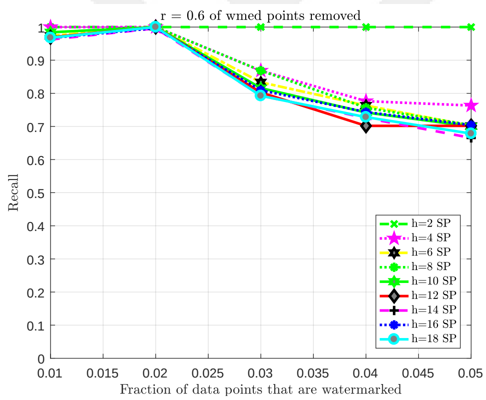
Figure A.1: Precision, Recall, and Entropy results when malicious set of SPs remove 50% of watermarked points. A.1a, A.1c, A.1e shows the results when familial nodes are excluded from belief propagation algorithm and A.1b, A.1d, A.1f shows the results when familial nodes included.



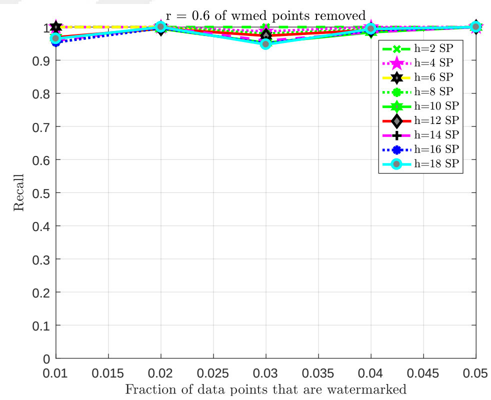
(a) Precision without familial nodes



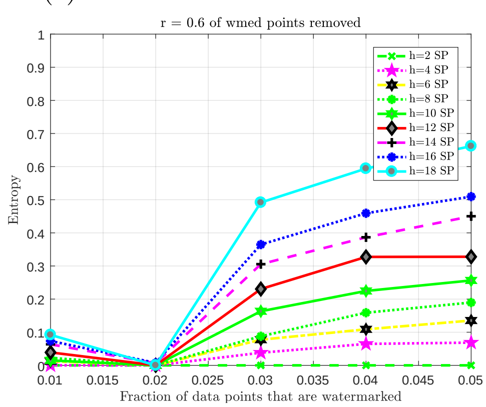
(b) Precision with familial nodes



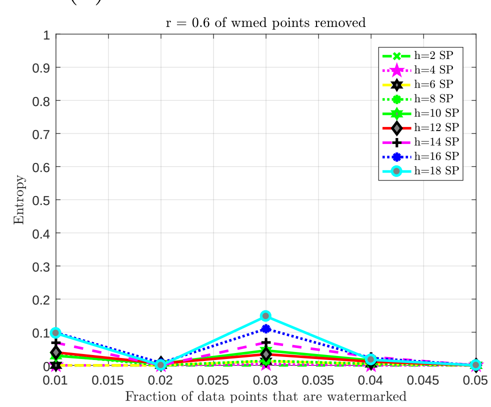
(c) Recall without familial nodes



(d) Recall with familial nodes

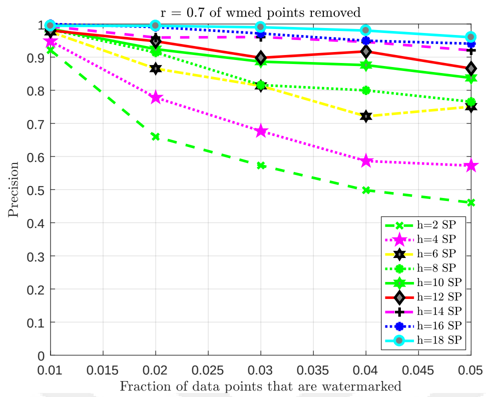


(e) Entropy without familial nodes

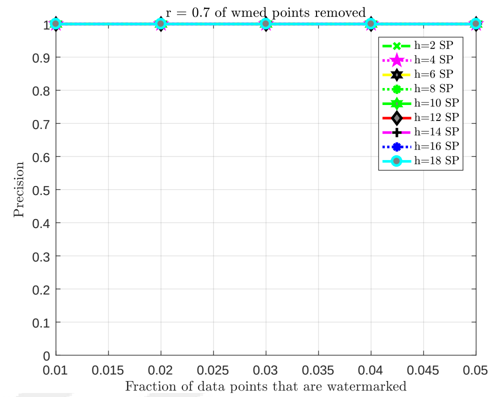


(f) Entropy with familial nodes

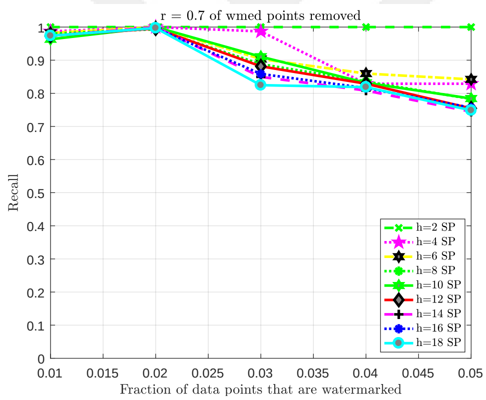
Figure A.2: Precision, Recall, and Entropy results when malicious set of SPs remove 60% of watermarked points. A.2a, A.2c, A.2e shows the results when familial nodes are excluded from belief propagation algorithm and A.2b, A.2d, A.2f shows the results when familial nodes included.



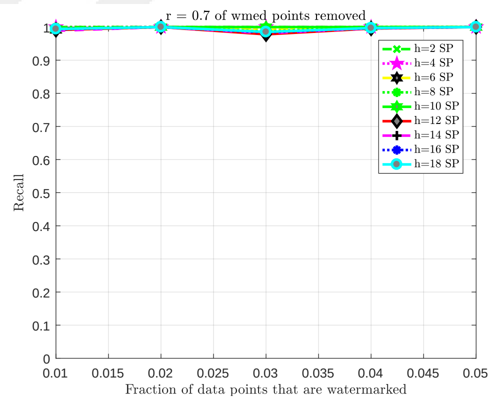
(a) Precision without familial nodes



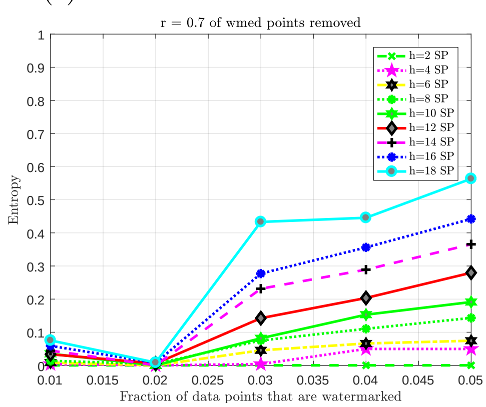
(b) Precision with familial nodes



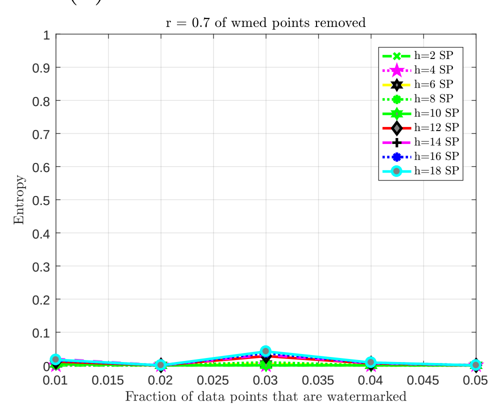
(c) Recall without familial nodes



(d) Recall with familial nodes

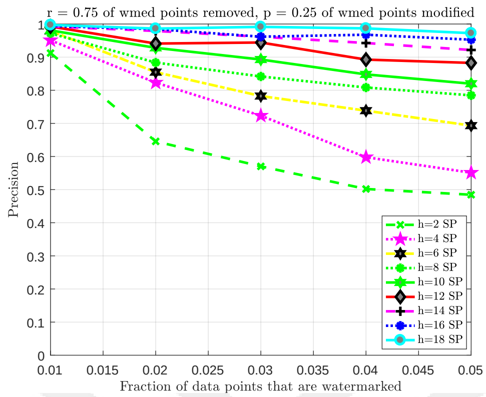


(e) Entropy without familial nodes

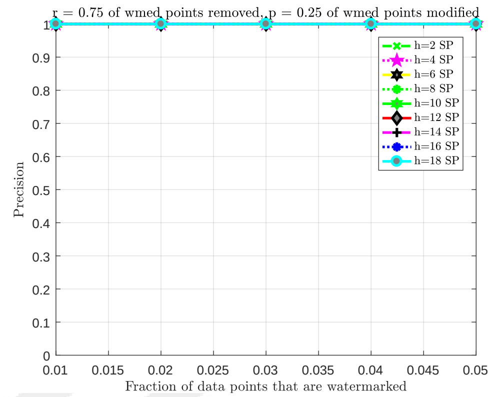


(f) Entropy with familial nodes

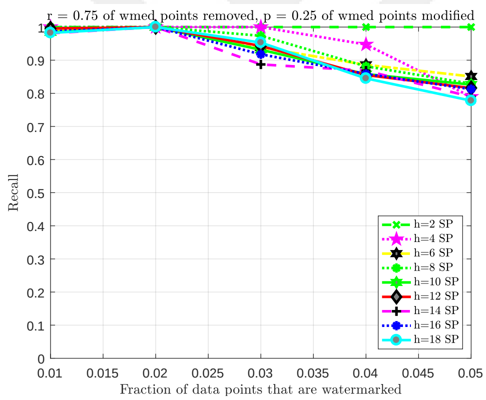
Figure A.3: Precision, Recall, and Entropy results when malicious set of SPs remove 70% of watermarked points. A.3a, A.3c, A.3e shows the results when familial nodes are excluded from belief propagation algorithm and A.3b, A.3d, A.3f shows the results when familial nodes included.



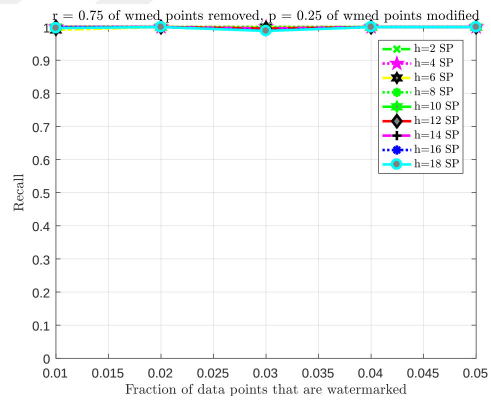
(a) Precision without familial nodes



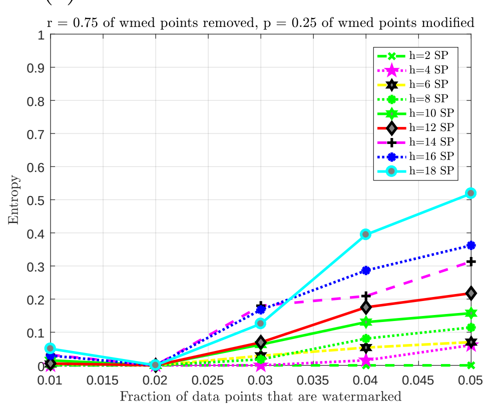
(b) Precision with familial nodes



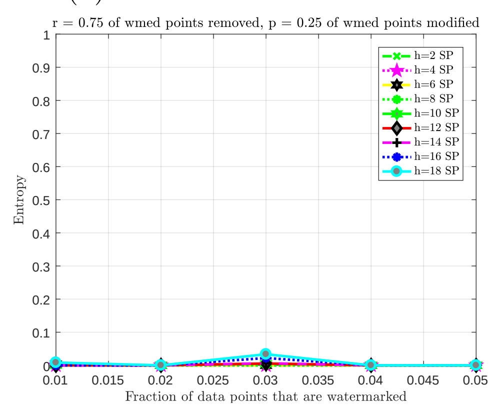
(c) Recall without familial nodes



(d) Recall with familial nodes

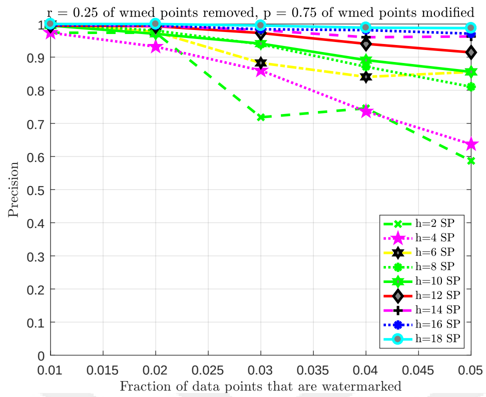


(e) Entropy without familial nodes

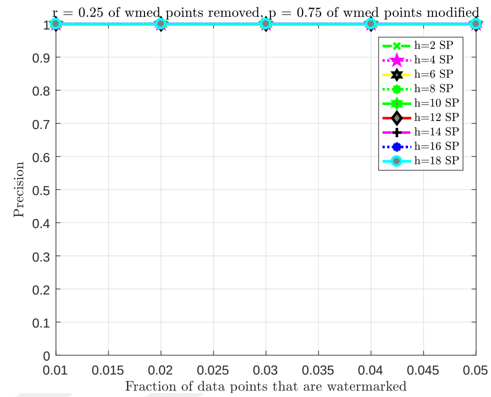


(f) Entropy with familial nodes

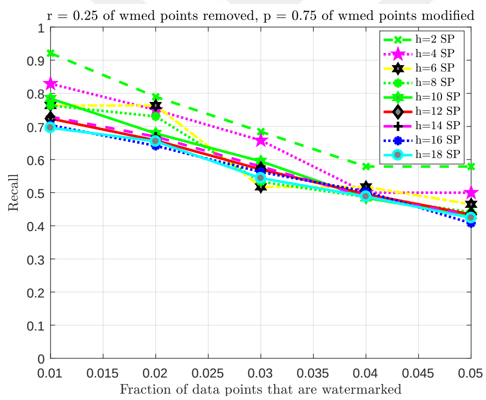
Figure A.4: Precision,Recall, and Entropy results when malicious set of SPs remove 75% of watermarked points and modify remaining 25%. A.4a, A.4c, A.4e shows the results when familial nodes are excluded from belief propagation algorithm and A.4b, A.4d, A.4f shows the results when familial nodes included.



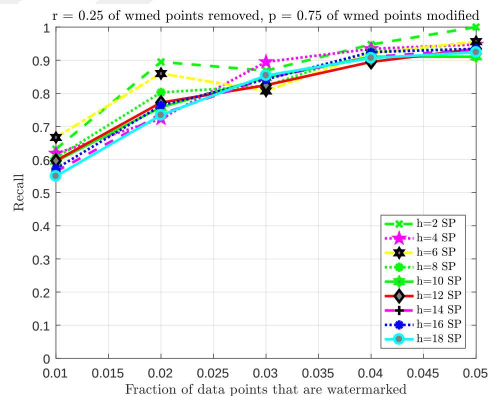
(a) Precision without familial nodes



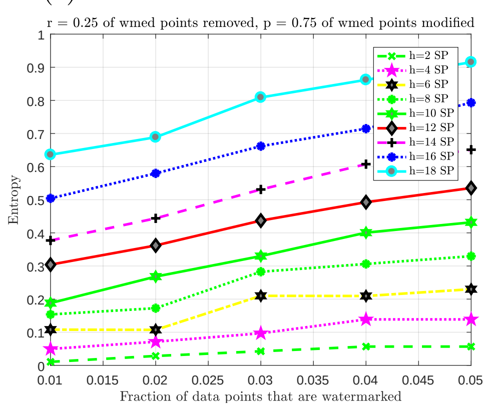
(b) Precision with familial nodes



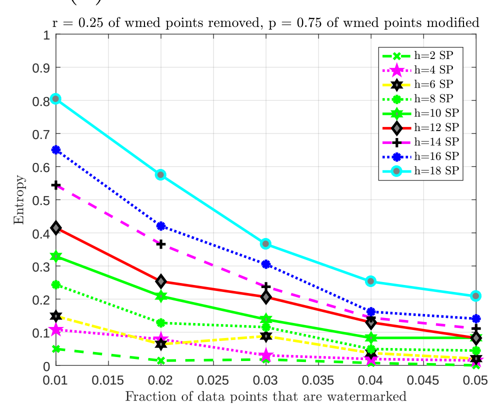
(c) Recall without familial nodes



(d) Recall with familial nodes



(e) Entropy without familial nodes



(f) Entropy with familial nodes

Figure A.5: Precision,Recall, and Entropy results when malicious set of SPs remove 25% of watermarked points and modify remaining 75%. A.5a, A.5c, A.5e shows the results when familial nodes are excluded from belief propagation algorithm and A.5b, A.5d, A.5f shows the results when familial nodes included.