

**A PRIVACY-PRESERVING SOLUTION FOR  
STORAGE AND PROCESSING OF  
PERSONAL HEALTH RECORDS AGAINST  
BRUTE-FORCE ATTACKS**

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  
Saharnaz Esmaeilzadeh Dilmaghani  
September 2017

A Privacy-Preserving Solution for Storage and Processing of Personal  
Health Records against Brute-Force Attacks  
By Saharnaz Esmailzadeh Dilmaghani  
September 2017

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)

---

Abdullah Ercüment Çiçek

---

Ali Aydın Selçuk

Approved for the Graduate School of Engineering and Science:

---

Ezhan Kardeşan  
Director of the Graduate School

# ABSTRACT

## A PRIVACY-PRESERVING SOLUTION FOR STORAGE AND PROCESSING OF PERSONAL HEALTH RECORDS AGAINST BRUTE-FORCE ATTACKS

Saharnaz Esmailzadeh Dilmaghani

M.S. in Computer Engineering

Advisor: Erman Ayday

September 2017

There is a crucial need for protecting patient's sensitive information, such as *personal health record* (PHR), from unauthorized users due to the increase in demands of electronic health records. Even though cryptography systems have been significantly developed, cyber attack is dramatically increased during the last couple of years. Although using high entropy passwords in the encryption methods can decrease the success of an adversarial attack, it is not popular among the users to choose such passwords. However, using a weak password makes the system vulnerable to brute-force attacks. Towards this end, we present a new framework as a solution for a secure storage of PHR data regardless of the password entropy.

Our system is an application of *Honey Encryption* (HE) scheme which is a new approach that provides a security beyond the brute-force bound and therefore dominates the *Password Based Encryption* (PBE). We utilize almost 10K patients' information from various datasets in order to construct a precise encoder/decoder model as a core element of HE. By providing the proposed model, we ensure that the encryption with invalid keys yields a valid-looking but incorrect health information of a patient to an adversary. The previous applications of HE are mainly on the static datasets that are not changing over the time. However, we were able to design an HE based model on a highly dynamic dataset of PHR. To the best of our knowledge, we are the first to provide a robust password based framework against brute-force attacks of health records regardless of the password entropy.

The results of the comparing our proposed encoding method with the direct

application of the PBE scheme show that it is almost impossible for an adversary to eliminate any wrong password. We also consider real-life scenarios for different attacks with side information about a patient's health related attributes. We implement a robust and concrete framework for storing and processing the PHRs that is also a novel, practical solution for protecting PHR data.



*Keywords:* Security and Privacy, Personal Health Record (PHR), Honey Encryption.

## ÖZET

# KİŞİSEL SAĞLIK VERİLERİNİN KABA GÜÇ SALDIRILARINA KARŞI GÜVENLİ SAKLANMASI VE İŞLENMESİ

Saharnaz Esmacilzadeh Dilmaghani  
Bilgisayar Mühendisliği, Yüksek Lisans  
Tez Danışmanı: Erman Ayday  
Eylül 2017

Elektronik sağlık kayıtlarına olan taleplerin artması nedeniyle, kişisel sağlık kaydı gibi hassas bilgilerin yetkisiz kullanıcılardan korunmasına çok önemli bir ihtiyaç vardır. Kriptografi sistemleri önemli ölçüde geliştirilmiş olsa da, siber saldırılar son iki yılda büyük ölçüde artmıştır. Şifreleme yöntemlerinde yüksek entropiye sahip parolalar kullanmak olası saldırıların başarısını azaltabilirse de, kullanıcılar arasında böyle şifreleri seçmek popüler değildir. Bununla birlikte, zayıf bir şifre kullanmak, sistemi kaba kuvvet saldırılarına açık hale getirir. Bu amaçla, bu çalışmada şifre entropisine bakılmaksızın kişisel sağlık kaydı verilerinin güvenli bir şekilde depolanabilmesi için yeni bir sistem sunuyoruz.

Sistemimiz, kaba kuvvet sınırının ötesinde bir güvenlik sağlayan ve bu nedenle parola tabanlı şifrelemeye üstünlük sağlayan yeni bir yaklaşım olan Honey Encryption (HE) şemasının bir uygulamasıdır. HE'nin temel unsuru olarak kesin bir kodlayıcı/kod çözücü modeli oluşturmak için çeşitli veri setlerinden yaklaşık 10,000 hasta bilgisi kullanıyoruz. Önerilen modeli sağlayarak, geçersiz anahtarlarla yapılan şifrelemenin saldırgan, hastanın geçerli görünümü ancak yanlış sağlık bilgilerini vermesini sağlıyoruz. HE'nin daha önceki uygulamaları genellikle zaman içinde değişmeyen statik veri kümeleriyle ilgilidir. Ancak biz kişisel sağlık kayıtları içeren oldukça dinamik bir veri kümesinde HE tabanlı bir model tasarladık. Edindiğimiz bilgiler doğrultusunda, parola entropisine bakılmaksızın sağlık kayıtlarının kaba kuvvet saldırılarına karşı gelebildiği parola tabanlı ilk sistemi önerdik.

Önerilen kodlama yönteminin, parola tabanlı şifreleme şemasının doğrudan uygulanmasıyla karşılaştırılmasının sonuçları, bir saldırganın herhangi bir yanlış

şifreyi elemesinin hemen hemen imkansız olduğunu göstermektedir. Aynı zamanda, bir hastanın sağlıkla ilgili özelliklerine dayanan yan bilgiler içeren farklı saldırılar için gerçek hayat senaryolarını ele alıyoruz. Kişisel sağlık kaydı verilerini depolamak ve işlemek için sağlam bir sistem uyguluyoruz. Sistemimiz kişisel sağlık kaydı verilerini korumak için yeni ve pratik bir çözümdür.



## Acknowledgement


I would like to thank all people who supported me during the last two years and who contributed in the work that is described in this thesis. First and foremost, I would like to thank my supervisor Dr. Erman Ayday for giving me the position in his group, for his invaluable guidance and encouragement through my M.Sc. study and research. I would like to thank Dr. Abdullah Ercüment Çiçek and Dr. Ali Aydın Selçuk for being in my thesis committee and contributing in the validation survey for this research project.

I want to thank my lovely friends Anisa H., Didem D., and Nora V. for all their friendliness, for helping me to survive all the stress, and for their support and suggestions through the process of doing research and writing this thesis, and for being my best friends in Turkey. Many thanks to my dear friend Mina E. for her spiritual support and compassion. My sincere thanks also go to my lovely, precious friends Bahareh F. and Hanieh K. for all their unfailing support and love. I would like to thank Ehsan K. for his encouragement and his kindness throughout the last two years.

I would also like to thank all my friends and colleagues at Bilkent University, especially Maryam S., my old classmate Iman D., Nuoshin F., Pezhman E., Nima A., Mohammad M., Nazanin J., Ehsan Y., Zeinab E., and Fatemeh E. for the valuable friendship and support, for the stimulating discussions, and for all the fun we have had together during the last two years. I would also like to thank Ms. Ebru Ateş for all her support in the department and for her emotional kindness.

Last not least, I wish to express my profound gratitude to my family who encouraged me throughout my life. My father and mother taught me precious lessons of life that made me strong enough to take my own journey in life. My lovely sisters Sarah and Sevda who always stood by my side and not letting me give up. I kindly appreciate my dear uncle Jafar Sadegh for all his support and his suggestions. This accomplishment would not have been possible without them.

# Contents



<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background and Related Work</b>	<b>4</b>
2.1	Brute-force Message-recovery . . . . .	4
2.2	Honey Encryption (HE) . . . . .	5
2.3	Password-based Encryption (PBE) . . . . .	8
2.4	Modified Paillier Cryptosystem . . . . .	8
2.4.1	Homomorphic Properties of Paillier Cryptosystem . . . . .	9
2.4.2	Partial Decryption . . . . .	9
2.5	Related Work . . . . .	10
<b>3</b>	<b>Proposed Solution</b>	<b>13</b>
3.1	Problem Formulation . . . . .	13
3.1.1	Data Representation . . . . .	14
3.1.2	System Model . . . . .	14



3.1.3	Threat Model . . . . .	16
3.2	Proposed Solution . . . . .	17
3.2.1	PHR Retrieval . . . . .	17
3.2.2	PHR Update . . . . .	27
<b>4</b>	<b>Evaluation</b>	<b>33</b>
4.1	Data Model . . . . .	33
4.2	Correlations between the Values . . . . .	35
4.3	Performance . . . . .	37
<b>5</b>	<b>Security Analysis</b>	<b>41</b>
5.1	Measure for DTE Security . . . . .	41
5.2	Security under Brute-force Attacks . . . . .	45
5.3	Security Analysis with Side Information . . . . .	46
5.3.1	Physiological Variables . . . . .	49
5.3.2	Drugs List . . . . .	52
5.4	Discussion . . . . .	53
<b>6</b>	<b>Conclusion and Future Work</b>	<b>55</b>

# List of Figures

2.1	Encoding/Decoding before Encryption/Decryption of Honey Encryption. . . . .	6
2.2	A DTE to map a message space of disease to a seed space. . . . .	7
3.1	The proposed system model for privacy-preserving storage and retrieval of PHR data. . . . .	16
3.2	System model for PHR data storage and retrieval algorithm. . . . .	18
3.3	Calculating <b>avail</b> and <b>alloc</b> subspaces in the DTE. . . . .	21
3.4	A toy example of the encoding process. . . . .	24
3.5	System model for updating health records. . . . .	29
4.1	Average age trajectories of eight physiological attributes for males and females [1]. . . . .	35
4.2	The relationship between different drugs and age. . . . .	36
4.3	Pairwise correlations of drugs. . . . .	37
4.4	Performance of the PHR Retrieval algorithm on the physiological variables and drugs list. . . . .	38

5.1	Games defining DTE goodness. . . . .	43
5.2	Game defining MR security. . . . .	44
5.3	A simple brute-force attack to compare the conventional PBE and our proposed system. . . . .	46
5.4	Evaluation of adversary's advantage with blood pressure level as the side information. . . . .	49
5.5	Evaluation of adversary's advantage with cholesterol level as the side information. . . . .	50
5.6	Evaluation of adversary's advantage with blood pressure and the cholesterol level as the side information. . . . .	51
5.7	Evaluation of adversary's advantage with drugs list as the side information. . . . .	52

# List of Tables

2.1	Notations and definitions. . . . .	5
4.1	Performance of the PHR Update algorithm physiological variables and drugs list DTEs with different number of attributes. . . . .	39
4.2	Improved performance after reorganizing the physiological vari- ables DTE. . . . .	40

# Chapter 1

## Introduction

The transformation from paper-based health records to a digital format gathers all the information from various doctors' office in a single file called Personal Health Record (PHR) [2]. It includes information from a variety of sources, including health care providers. They can provide medical history, lab results, record health vitals, and track progress [3]. The national push to digitize the health data in USA raise the concerns of privacy and security for safeguarding medical information. To that end, in 1996, Health Insurance Portability and Accountability Act (HIPAA) [4] standardized electronic transactions in the health care sector and regulated the use of health data. HIPAA regulated the privacy and security of health data.

Even though people embrace the digitalization of the records, they have serious concerns about the privacy and security of their health records [5] and some prefer to be consulted before any releasing of their information [6]. Even so, a lot of data breaches reported during the last years. According to a report by the American National Standards Institute (ANSI), the health information privacy of nearly 18 million Americans have been breached from 2010 to 2012 [7]. Around three billion digital medical data records have been compromised since 2013, according to IBM. A meager four percent of that data was encrypted, though, meaning those credit card numbers, user names and passwords, and social security numbers passed easily onto dark-web criminal exchanges [8]. Yet in 2014 cyber attacks dramatically increased to 72% [9]. It didn't become any better in

the last couple of years. A 566 percent increase in data breaches reported, that means 12 million records were compromised in the healthcare industry just in 2016 [10]. Furthermore, a total of 37 serious healthcare breach incidents were reported to the department of Health & Human Services (HHS) or the media in the month of May 2017 alone [11]. Digital health data also couldn't survive from ransomware attacks [12].

The key subjective view to take into consideration is how health data breach can affect individuals' life. These attacks can affect an individual's life in a way that s/he may get fired from his work or feel ashamed in front of his family [13]. Above all, there are also people who suffer from illness, however, they do not attempt treatment because of privacy concerns [14]. That is to say, protecting PHR from cybercriminal attacks is an undeniable fact. The available evidence seems to point that even though cryptography systems have been significantly improved, cyberattack is dramatically increased and yet most of the encrypted databases used for electronic medical records leak information [15, 16].

The current existing encryption-based methods are highly dependent on an  $n$ -bit key while the size of the key is an important feature in the security of an encryption method, whereas the passwords that are difficult to guess by an attacker are also not easy to remember [17]. Hence, users are willing to use an easy-to-remember passwords [18, 19] which lead to a successful brute-force attacks.

*Honey Encryption* (HE) [20] is recently proposed by Juels *et al.* A new encryption tool which provides security by adding a new layer to the conventional encrypted methods. Most of the current encryption schemes use a key, where the increase of encryption security is dependent on the size of the key. Unlike the traditional *Password-based Encryption* (PBE) [21] methods, HE is not dependent on the password entropy. Using the HE, encrypting a ciphertext with a wrong key by an attacker represents a plausible looking message yet incorrect information. This property of HE provides a strong defense wall against an adversary who may try to attack a database by examining all possible passwords. In this case, the adversary is deceived by the system and he cannot eliminate his options in the password pool.

Having said that, the HE relies on a distribution-transforming encoder (DTE)

to transform the message space into a uniform seed space. On the other hand, constructing a good DTE to perfectly match to the dataset is not an easy task which makes HE not practical to implement on any domain that is one of the limitations of HE approach. Furthermore, since most of the datasets in real-world are changing over the time, constructing a DTE on a dynamic dataset is another limitation of HE. It is challenging to provide an efficient solution for a dynamic dataset. This is what we address through this study.

Our solution is an application of HE scheme on the PHR data. We utilize HE to provide a secure for the storage and data retrieval of the PHRs. In this framework, the PHR data is first encoded and then encrypted by a patient's password. Notably, the system does not depend on the encryption method, either the password complexity. A patient's password can be of any size, even an easy to guess password (or low entropy password) which occurs with high probability in real-world is not going to bother the system in the privacy and security aspects [17]. While decrypting the message, an authorized user gets the true message, however, an adversary ends up with a valid-looking message without understanding whether it is the correct one. Hence, the system prevents brute-force attacks.

Our main contributions through the study are as follows:

- We propose a new model to protect PHRs against brute-force attacks.
- The proposed method addresses some of the limitations of HE such as providing a model for dynamic dataset (e.g., medical records).
- We implement our proposed method and examine the system by providing security tests.

The structure of the thesis is as follow. Chapter 2 describes some background information regarding the concepts and theories that we have used during this study along with a review of related research in the area. In Chapter 3, the problem formulation and a detailed information of the proposed system is provided. Chapter 4 discusses the evaluations on the data model and the performance of the system. The proposed system is evaluated against different attacks in Chapter 5, and the details regarding the security analysis are investigated through this chapter. Finally, Chapter 6 concludes the thesis by discussing the future works.

# Chapter 2

## Background and Related Work

In this chapter, we outline some required background and main concepts of encryption methods and tools that we have employed during this study. Then, we discuss some of the related studies. Furthermore, for the simplicity, we gathered all frequently used notation that we have used in this study along with their definitions in Table 2.1.

### 2.1 Brute-force Message-recovery

In a brute-force attack, an attacker tries as many password s/he can in order to find the correct one. Assuming that a message  $M$  is encrypted under a key  $K$  (considering that  $M$  and  $K$  are from a predefined distribution), it gives a ciphertext  $C$  that is  $C = Enc(K, M)$ , an adversary's goal is to recover  $M$ . Trying all possible keys to decrypt  $C$ , finally, message  $M$  should appear as one of the decrypted messages results. Note that in a system which is secured by conventional password-based encryption (PBE) [21] method, an attacker can easily delete an incorrect password with a high probability.

Considering the above argument, PBE method does not provide enough security for the data. Besides, the fact that users choose simple passwords [18] threatens the systems that are based on PBE.



$\mathcal{M}$	Message space
$M$	A message sequence of a PHR, $M \in \mathcal{M}$
$p_m$	Original message distribution
$\mathcal{K}$	Key space
$p_k$	Password distribution
$\mathcal{C}$	Ciphertext
$\mathcal{S}$	Seed space
$S$	A seed in DTE, $S \in \mathcal{S}$
$p_d$	Message distribution in DTE
$P$	A password chosen by user
$\langle m \rangle$	A message that is encrypted by Paillier Cryptosystem
$Enc(m)$	Password-based encryption of message $m$
$Dec(m)$	Decryption of message $m$
$PK$	The public key of Paillier Cryptosystem
$x$	The secret key of Paillier Cryptosystem
$KDF$	The Key Derivation Function for

Table 2.1: Notations and definitions.

## 2.2 Honey Encryption (HE)

Honey encryption [20] is recently proposed by Jules and Ristenpart in 2014. The word *honey* usually refers to a mechanism in computer security detecting attempts of unauthorized use of data. In another word, it holds data which appear to be legitimate in order to chase or bait an adversary [22].

The method is in fact based on security schemes in which the purpose is deception and luring attackers. HE provides *honey messages* through a brute-force attack and deceives an attacker in a way that s/he cannot distinguish messages from correct ones.

HE has the same syntax and semantic of the PBE scheme, in addition, HE has an extra hedge, which is encoding/decoding process, to protect data from data breaches. That is to say, HE provides a security beyond the brute-force bound and it makes an attack unsuccessful for relatively low-entropy passwords, by constructing *honey messages* for each possible password.

To put it in another way, an HE setup  $HE = (HEnc, HDec)$  is a pair of encryption and decryption algorithms. Let  $\mathcal{M}$  and  $\mathcal{K}$  be two sets that represent the message space and key space. We choose a message  $M \in \mathcal{M}$  encrypt it under a key  $K \in \mathcal{K}$  and the output is a ciphertext  $C = HEnc(K, M)$ . Decryption of a ciphertext  $C'$  under a key  $K'$  yields a message  $M' = HDec(K', C')$  that is a incorrect message from the same message space  $\mathcal{M}$ .

Figure 2.1 illustrates the HE method. Given that  $S$  is a seed,  $r$  is an  $n$ -bit random string used for the encryption. Note that the encoding process is probabilistic presented as  $\$,$  however, the decoding process is deterministic.

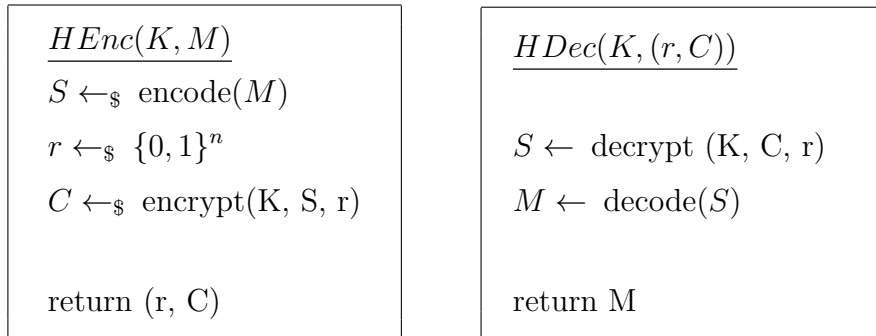


Figure 2.1: Encoding/Decoding before Encryption/Decryption of Honey Encryption. Encoding is probabilistic (implies with  $\$$ ), and decoding is deterministic [20].

The core concept behind HE is that it maps a non-uniform message space to a larger uniform and provides a  $S \in \mathcal{S}$ . This is a new method of message encoding which is called *Distribution-Transforming Encoder (DTE)* that is represented in HE.

## Distribution Transforming Encoder (DTE)

DTE is one of the main elements in HE to model the message space. The DTE consists of two steps, **encode** and **decode**. The DTE maps  $M$  to a seed space  $\mathcal{S}$ .  $M$  is chosen with a probability distribution  $p_m$  from a set of message space  $\mathcal{M}$ . A DTE then encodes  $M$  to a seed  $S$  which randomly is assigned to  $M$ . Therefore, the encoding is not necessarily unique. The decoding process, on the other hand, is deterministic. Given a seed  $S$  we can generate the message  $M$ .

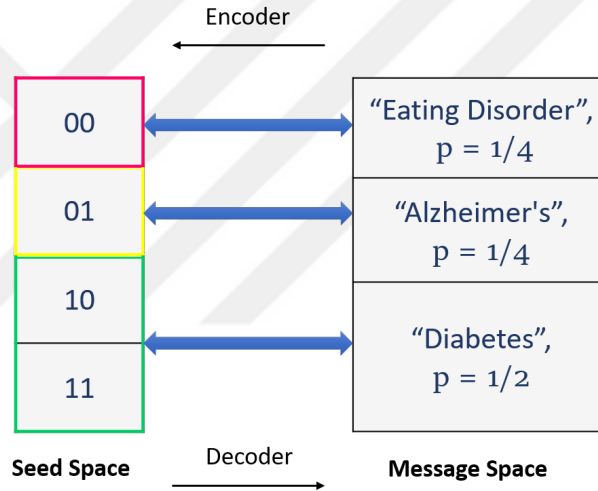


Figure 2.2: A DTE to map a message space of disease to a seed space. Message space  $\mathcal{M}$  consists of diseases and seed space  $\mathcal{C}$  is 2-bit strings. Considering the probabilities that are assigned to each disease, we can map each disease to a seed range.

Figure 2.2 illustrates a basic example of a DTE. Message space includes different diseases in this case  $M = \{\text{Eating Disorder}, \text{Alzheimer's}, \text{Diabetes}\}$  with a probability distribution  $p_m$ . Through a knowledge about some population's diseases, the probabilities of each disease are generated. We consider a 2-bit string for seed space and partition the range to different portions based on the probabilities of each disease.

One of our main contributions in this study is to construct a DTE for PHR data, which is a non-uniform dataset.

## 2.3 Password-based Encryption (PBE)

Password-based encryption (PBE) [21] is a symmetric-key (that relies on a single key to perform both encryption and decryption on the same data) generation model that transforms an input string (a password) into a encryption key using various techniques.

PBE is typically implemented using standard hashing algorithms, such as the PKCS #5 standard of RFC2898. These algorithms often use a key derivation function (KDF) to strengthen the encryption.

KDF takes as inputs a password and derives a secret key by using a pseudo-random function. The main purpose of using KDFs is to derive keys from secret passwords, which typically do not have the desired properties to be used directly as cryptographic keys. Such use may be expressed as  $DK = (P, Salt)$  where  $DK$  is the derived key,  $KDF$  is the key derivation function,  $P$  is the original password that is chosen by a user,  $Salt$  is a random number which acts as cryptographic salt.

The derived key,  $DK$  is used instead of the original password in the system. The value of the salt is stored with the hashed password or sent as plaintext with an encrypted message.

## 2.4 Modified Paillier Cryptosystem

During this study, we benefit from Paillier cryptosystem [23] to apply some of the encryption methods such as homomorphic encryption [24] and partial decryption. Paillier is a probabilistic asymmetric algorithm for public key cryptography that supports some homomorphic properties. The scheme works by generating a public key that is illustrated as:

$$PK = [b, u, h = (g^x)] \quad (2.1)$$

The public key in Equation (2.1) is composed of different components.  $b$  represents a strong secret key that is equal to  $pq$  with  $p$  and  $q$  chosen randomly from large prime numbers, a random number  $t$  is of the order  $(p-1)(q-1)/2$ , and the weak secret key  $x$  which belongs to the set  $[1, b^2/2]$ .

### 2.4.1 Homomorphic Properties of Paillier Cryptosystem

Homomorphic encryption [25] allows applying operations on a ciphertext without decrypting it. The homomorphic properties are one of the important features of the Paillier cryptosystem. The homomorphic scheme holds the following properties that we also benefit from them through our study.

- **Addition and Subtracting**

The product of two ciphertexts ends up to the encryption of sum of the plaintext of the same messages as follows:

$$Dec(Enc(m_1) \times Enc(m_2)) = m_1 + m_2.$$

Likewise, the subtraction operation follows a similar structure.

- **Multiplication**

A ciphertext raised to the power of a plaintext will decrypt to the product of the two plaintexts as follows:

$$Dec(Enc(m_1)^c) = m_1 \times c.$$

We applied homomorphic properties on the encrypted PHR data in order to update them without revealing any information.

### 2.4.2 Partial Decryption

Using the partial decryption we divide the secret key  $x$  into two separate key such that  $x = x_1 + x_2$ . Each key belongs to a party that is allowed to partially decrypt the data.

We benefited from partial decryption to decrypt data by involving two parties in the system. This way, we make a secure protocol by preventing to give the whole key to one party only. This encryption method is applied during the PHR update process to update the PHR of a patient in the hospital database. Hospital and patients are responsible to decrypt part of a data and after applying some operation, the data will be stored in the hospital database (The process is described later in Chapter 3 with more details).

## 2.5 Related Work

In the last few decades, using PHRs increases the concerns regarding privacy, security, and processing of healthcare data. Significant efforts have been done to provide security and privacy for PHR data [26, 27].

A couple of recent studies [28, 29, 30] investigated methods of security and privacy in electronic health records and classify them from different points of view, an overview of security and privacy requirements of e-health solutions, the privacy and security concerns of electronic health records system, and the system architecture. In another particular study from IBM [31], the authors focused on the algorithms that are developed for publishing patient data in a privacy preserving way.

Some of the studies focused on using rules and standards such as HIPPA [4] that defines the rules of privacy in USA health information. Others, propose pseudo anonymity techniques along with encryption [32, 33]. In a study by Demuyneck *et al.* [34] a system that provides access control for patients to choose who should have access to the health records. These system are patient-centric model. Li *et al.* [35] also propose a patient-centric framework in a public key cryptosystem and a mechanisms for data access control to PHRs which is stored at a third-party service provider. Recently, in another study [36] the authors provide an access control framework that uses hybrid cryptography and a two-factor authentication method for a secure protocol. [37] is another study on determining a secure system by limiting patients to share partial access rights to others.

The majority of researchers concentrate on encryption methods to increase the

security of the health data by using the symmetric key and public key techniques. Lee *et al.* [38] proposed a protocol based on symmetric keys that are stored in patient's smart card, hence, the presence of the smart card is required for each access. Narayan *et al.* [39] construct a secure and privacy-preserving EHR system in a public key cryptosystem (asymmetrical cryptography) by using the attribute-based encryption (ABE) method, and users are responsible for providing a secure mechanism in order to ensure the security and privacy of data. Some of the studies applied Homomorphic encryption strategy in order to protect genomic, clinical, and environmental data [40] or to perform scientific investigations on integrated genomic data [41]. Other approaches [42, 43], security is provided by hiding a search pattern and storing data in a third-party such as cloud.

However, little attention has been devoted to the impact of brute-force attacks and the solutions which can reduce the risks of revealing the health data after data breaches. To the best of our knowledge, we are the first to provide a privacy preserving password-based framework against brute-force attacks of health records regardless of the password entropy.

There are some studies in the literature that applied security schemes in which the purpose is deception and luring attackers. Honeytokens [44] and Honey-pots [22] are used to detect, deflect, and respond unauthorized usage attempts of information systems. Honeywords [45] is a solution that is to thwart attackers who look to avoid authentication schemes by cracking hashed passwords. By using honeywords, an attacker that has obtained a file of hashed passwords and inverts the hash function cannot tell if he or she has found a user's actual password or a honeyword. The honey solutions are used in industry [46] as well.

A recent solution for deceiving the attacker is proposed by Juels *et al.* [20] as honey encryption. Some of the studies benefited from honey encryption to deceive an attacker and provide a security beyond the brute-force attacks on different domains [47]. Among those is the application of HE on credit cards numbers which are highly sensitive information, using honey encryption method an incorrect key input in the system results is a valid message. In another application, honey encryption is applied on a simple question and answer messaging domain. While in a more complicated domain Huang *et al.* [48] propose their model for a secure storage of genomic data by using honey encryption. They construct an HE model on a dataset that is, despite the other application domains of HE, a non-uniform

dataset. Also, Yoon *et al.* [49] utilize HE in another data types of 2D images. Moreover, there are also other application of HE on Instant messaging system [50], and in natural language processing [51, 52] that are recently published.

Considering the applications of HE, none of the studies focused on a dynamic dataset which changes over time and specifically on the personal health records domain. Nonetheless, we were able to address this limitations of HE through this study and used this approach on a dynamic dataset.





# Chapter 3

## Proposed Solution

We design a system for privacy-preserving storage and retrieval of a patient's health information considering Personal Health Record (PHR) data that contains sensitive information such as health-related attributes. We benefit from honey encryption (HE) [20] approach in order to construct our framework. In this chapter, we investigate details of the proposed method. We start with the problem formulation in Section 3.1, discussing our assumptions for the proposed solution. Then, a general overview of the system along with the attack scenarios is introduced. In Section 3.2, technical details regarding the implementation of the system are specified.

### 3.1 Problem Formulation

In this model, PHR is a sequence of sensitive and important attributes that are recorded by a health service provider such as hospital. A PHR includes values of health-attributes (e.g., blood pressure), disorders and diseases, a list of corresponding drugs, symptoms, and treatments. PHR is the input of the system, which stores the data for later access and process.

### 3.1.1 Data Representation

We decompose a complete PHR of a patient into sequences of sensitive health-attributes, in 4 classes: **(i) Physiological Variables** which basically consists of test results such as blood pressure, cholesterol level, blood glucose, and diagnosis (disease), **(ii) Drugs** that is a list of drugs prescribed for a special disease of a patient, **(iii) Symptoms** of a disease such as fatigue, vision problems, numbness for MS disease, and **(iv) Treatment** that encompasses activities to care of a patient in order to combat a disease or disorder such as corticosteroids and physical therapy for MS disease. The message  $M$  is a sequence of health attributes values that is categorized in these categories. Hence,  $M$  is constructed as follows:

$$M = \left\{ \left\{ \text{blood pressure, cholesterol, blood glucose, disease, etc.} \right\}, \right. \\ \left. \left\{ \text{Drugs List} \right\}, \left\{ \text{Symptoms List} \right\}, \left\{ \text{Treatments List} \right\} \right\}. \quad (3.1)$$

We consider a separate PHR per disease of a patient, therefore, each person might have more than one PHR in her/his health documents.

In general, we assume  $M$  as a sequence of different attributes such that  $M = \{a_1, a_2, \dots, a_n\}$  and  $M_{i,j}$  is a subsequence of the message  $M$  that includes all elements from the  $i$ -th element until  $j$ -th.

### 3.1.2 System Model

As shown in Figure 3.1, our model consists of six parties: the adversary, the patient, the hospital, hospital staffs (e.g., doctors or nurses), the trusted authority (TA), and users that can be patient or hospital staffs as well. TA is in charge of generating and distributing public and secret keys. It randomly divides the secret key into two keys and sends it to the hospital and patient in order to not to give a full decryption access to any of the parties. Hospital is responsible for data collection, storage, and processing of the data. Data is encrypted under the patient's password which we assumed it is an easy-to-remember password since it is a common scenario in real life [17].

The main purpose of the system is to provide a privacy-preserving solution for storage, retrieval, and update of the PHR data. By this method, we store the PHR data in the hospital database and retrieve them whenever necessary. Moreover, the PHR can easily get updated if some of its attributes need to be updated. We benefit from honey encryption (HE) [20] approach in order to construct our framework. We also utilize Paillier cryptosystem [23] for updating PHR.

Even though the PHR data is highly dynamic and may change gradually, the DTE (distribution-transforming encoder) of HE is limited to datasets that do not change over time unless the data is completely decrypted, and encrypted again after the data is updated. Whereas this solution is not desirable for PHR data since it should be in clear frequently. To address this issue, we provide a protocol in order to update the data in a secure way without reconstructing the DTE or decrypting the whole data.

In a nutshell, our framework consists of two cornerstones; **PHR Retrieval** protocol in which a data retrieval request is sent by the user for accessing a PHR information, and **PHR Update** in which some of the attributes of a PHR are updated.

During the PHR retrieval process, a user who wants to access the data enters her/his password in the system. After authentication, the user requests for data access and the hospital provides the data.

When a patient revisits the hospital, the corresponding staff requests some of the information regarding the patient's health record to update her/his status of health-related attributes (e.g., blood pressure), if necessary. The responsible person applies some cryptographic operations on the data and updates the PHR. The old PHR then is replaced with the new one in the hospital database. Meanwhile, the hospital stamps the old record with date and keeps it in the archive for later accesses of a patient's medical history.

We assume that the end-user in the hospital (e.g., nurse) does not have full access to the health records, however, s/he is responsible for updating the data. Therefore, s/he uses this protocol in order to update a PHR without accessing it.

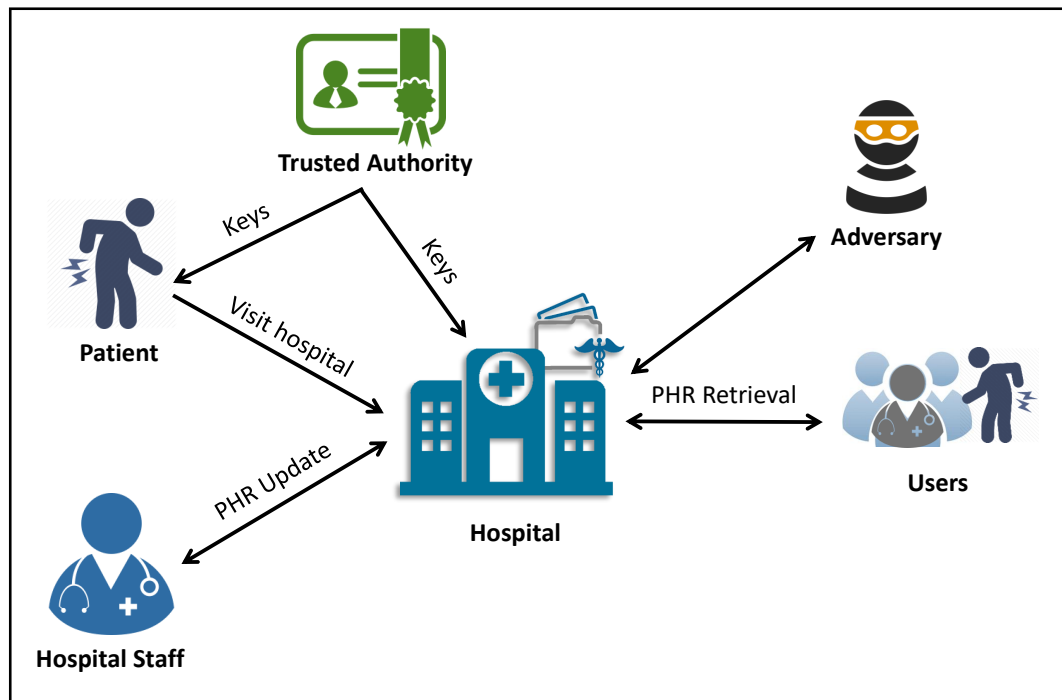


Figure 3.1: The proposed system model for privacy-preserving storage and retrieval of PHR data. six parties: the adversary, the patient, the hospital, hospital staff, the trusted authority (TA), and users who can be patient that can be patient or hospital staffs as well.

### 3.1.3 Threat Model

We assume two types of adversaries in the model: **(i)** an outsider attacker who obtains the encrypted database after hacking the system and then tries to decrypt the information, and **(ii)** an insider attacker (e.g., hospital staff) that has access to the encrypted database. The main purpose of the adversary is to obtain the health records via a brute-force attack that is repeatedly trying different passwords with the hope of eventually finding the correct one. Bearing in mind the fact that users are using easy-to-guess and low entropy passwords [18, 17], brute-force attack is a practical way to obtain information regarding a patient. We also consider that the demographic information of a patient (e.g., gender, age) is already disclosed to the attacker.

We also take into account a stronger attack in which an adversary has some

side information of a patient’s health record. We deliberate different scenarios such that an attacker might be a person from the hospital (e.g., a doctor, or a nurse) who knows about the health attributes (e.g., blood pressure). In another scenario, we presume the adversary as a pharmacist who knows the drug usage pattern of a patient. These attacks are described in more detail in Chapter 5.

Herein, we focus on protecting the data from attacks that might happen from inside of the hospital or an adversary who has stolen the encrypted database. Moreover, the outer-layer protection that includes decisions about various permissions for each user, will not be discussed during this study.

## 3.2 Proposed Solution

Our framework is a solution for secure storage, retrieval, and update of PHR data. As mentioned earlier in this chapter, the system consists of two principal algorithms. Herein, we first describe the **PHR Retrieval** algorithm and provide an example, and then we go through the **PHR Update** protocol.

### 3.2.1 PHR Retrieval

PHR retrieval provides a secure way for accessing the PHR data. We benefit from HE in this protocol so that the system always outputs a valid-looking message for every decryption result even for any wrong password. The algorithm consists of three main blocks: *Encoding*, *Decoding*, and *Encryption/Decryption*. We follow the HE approach and design a DTE for encoding and decoding. Furthermore, we utilize password-based encryption (PBE) [21] for encryption and decryption of the encoded data. Note that during this subsection, by the encrypted message we mean the password-based encryption of the message.

Figure 3.2 illustrates the steps through the PHR retrieval protocol. The patient visits the hospital and a specialist records his health-attributes as a PHR data (Step 1). The PHR is then encoded (Step 2) and encrypted by the patient’s

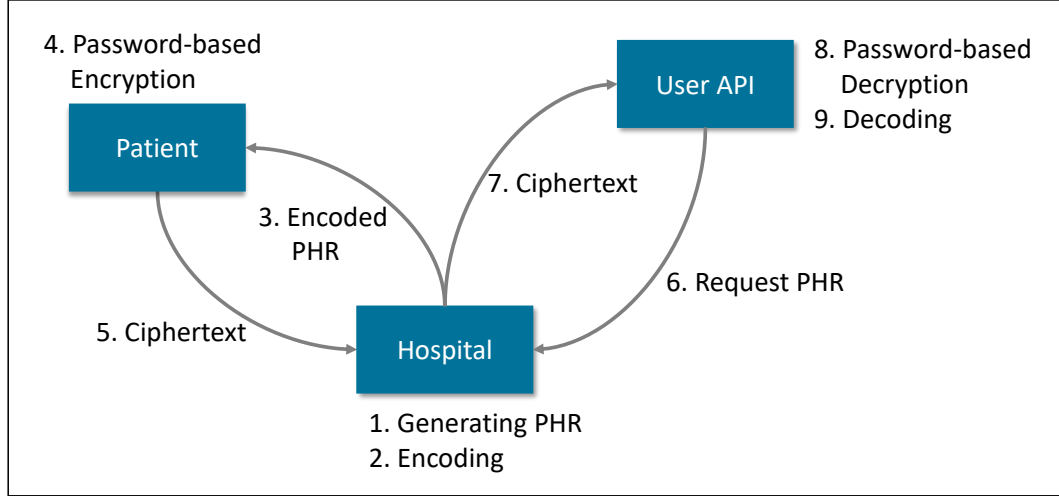


Figure 3.2: System model for PHR data storage and retrieval algorithm. A patient visits the hospital and hospital generates his/her PHR sequence. After encoding the PHR, the user encrypts it under his/her chosen password and sends the ciphertext to the hospital. When a user asks for the data, the cipher text is sent to the user after decrypting and decoding the user obtains the original data.

provided password (Steps 3 and 4). The encrypted PHR is then stored in the hospital database for later access (Step 5). During the retrieval process, a user (can be patient himself or a hospital staff) requests the PHR and enters her/his password to the system (Step 6). The hospital retrieves the corresponding ciphertext and sends it to the user (Step 7). The ciphertext is first decrypted under the user-provided password and then decoded to a PHR sequence (Steps 8 and 9).

Next, we explain the main blocks of the PHR retrieval algorithm.

### 3.2.1.1 Encoding

Applying the HE method, we need to construct a DTE to encode the PHR sequence into an integer called *seed*. In another word, our main objective is to provide an efficient way to transfer the non-uniform distributed message space  $\mathcal{M}$  to a uniform seed space  $\mathcal{S}$ , and map any message  $M \in \mathcal{M}$  to a seed  $S \in \mathcal{S}$ .

It is important to consider all the possible relationships between different attributes of a PHR to create a precise and good DTE model. Different studies [53, 54] discuss the relationships between health attributes (e.g., blood pressure) and demographic attributes (e.g., gender). We studied the possible correlations in real datasets and considered them while constructing the DTE.

We estimate the conditional probability of a PHR given all other attributes in a message  $M$ . We define  $P(a_i|M_{1,i-1})$  as the conditional probability of the  $i$ -th attribute given preceding ones. The probability of a complete message  $M$  can be calculated as follows:

$$p_m(M) = P(a_1)P(a_2|a_1) \dots P(a_{n-1}|M_{2,n-2})P(a_n|M_{1,n-1}). \quad (3.2)$$

The encoding approach for such a sequence that consists of different health-attributes works by assigning subspaces of  $\mathcal{S}$  to the prefixes of  $M$  that is the subsequences of  $M$ . Suppose a message with four elements:  $M = \{a_1, a_2, a_3, a_4\}$ , its prefixes are  $\{a_1, a_1a_2, a_1a_2a_3, a_1a_2a_3a_4\}$ .

We construct a tree-based structure DTE to encode PHR data. Each message  $M$  is represented by a branch in the tree with a subspace  $\mathcal{S}_M$  that is assigned for that branch. Then, a seed from this subspace will be attached to the message  $M$ .

For each category of health attributes (e.g., physiological attributes) that is defined in Subsection 3.1.1, we build a DTE, hence, we end up with four types of DTE at the end. The encoding algorithm takes as input a message  $M$  and generates a seed from each tree as an output:  $S_P$  for Physiological Variables,  $S_D$  for drugs,  $S_S$  for symptoms, and  $S_T$  for treatments. The main output (seed) at the end is concatenation of all four seeds such that:

$$S = \{S_P||S_D||S_S||S_T\},$$

hence,  $S$  is the encoded  $M$  using DTEs.

The DTE construction is a straightforward approach. We use a tree structure with  $n$  levels that each level is assigned to a health-attribute in  $M$  and different

nodes at each level represents all possible values of that attribute. For instance, suppose that the first level of the tree represents the blood pressure level, the nodes in that level then should show the possible values of the blood pressure level for human. We then divide the seed space  $\mathcal{S}$  in different subspaces by using the conditional probabilities of Equation (3.2). Each node at  $i$ -th level of the tree and the  $j$ -th order is represented by  $node_{i,j}$ .

The total seed space size is assigned to the root node with the interval  $[L_0^0, U_0^0]$  (note that the root is at level 0). This is the available seed space that is going to be divided into portions for nodes at the next levels. The available seed size is stored in a variable called **avail** and the available seed space for a  $node_{i,j}$  is calculated as  $\mathbf{avail}_{i,j} = U_i^j - L_i^j + 1$ . While the algorithm proceeds to the next level of the tree, **avail** value is divided into different seed subspaces by using the conditional probabilities. The subspace seed that is allocated to a node by its parent node is called **alloc** variable. Put it differently, the total allocated subspaces of all children nodes is equal to the available seed space of the parent node, hence, assuming that  $node_{i,j}$  has  $c_{i,j}$  children at level  $i + 1$  we have:  $\sum_{j=1}^c \mathbf{alloc}_{i+1,j} = U_i^j - L_i^j + 1$ .

The main purpose is to reach the leaf node by calculating the allocated seed subspaces and narrowing down the root interval until the leaf node. To this end, we need to calculate the allocated seed for each child node in order to find its interval. Suppose the average number of children of each node in the tree is  $b$ , the conditional probability of  $node_{i,j}$  is represented by  $P_{node_{i,j}}$ , and  $c_{i,j}$  is the number of children that belong to  $node_{i,j}$ . The allocated seed subspace for an attribute is calculated as follows:

- for  $t \in \{1, 2, \dots, c - 1\}$

$$\mathbf{alloc}_{i+1,t} = \begin{cases} \lceil b \rceil^{n-i-1} & \text{if } \frac{P_{node_{i+1,t}}}{\sum_{j=1}^c P_{node_{i+1,t}}} < \frac{\lceil b \rceil^{n-i-1}}{\mathbf{avail}_{i,j}}, \\ \lceil P_{node_{i,t}} \cdot \mathbf{avail}_{i,j} \rceil & \text{otherwise,} \end{cases}$$

- for  $t = c$  (if  $node_{i+1,t}$  is the last node.)

$$\mathbf{alloc}_{i+1,t} = \mathbf{avail}_{i,j} - \sum_{t=1}^{c-1} \mathbf{alloc}_{i+1,t} \quad (3.3)$$



Figure 3.3 shows the above-calculations on the nodes of a tree. The purpose is to calculate the allocated seed space for the children nodes of  $node_{i,j}$ , which has  $c_{i,j}$  children (in order to make it simple we represent this as  $c$ ), and its available seed subspace is equal to  $U_i^j - L_i^j + 1$ . The allocated seed space for each child is calculated based on the conditions in (3.3) and the corresponding conditional probabilities (e.g.,  $P_{node_{i+1,cj}}$ ). The algorithm proceeds by choosing the corresponding node (suppose the node  $4j + 2$ ) and take its allocated seeds as available seed space for the next step, hence,  $\mathbf{avail}_{i+1,4j+2} = \mathbf{alloc}_{4j+2}$ .

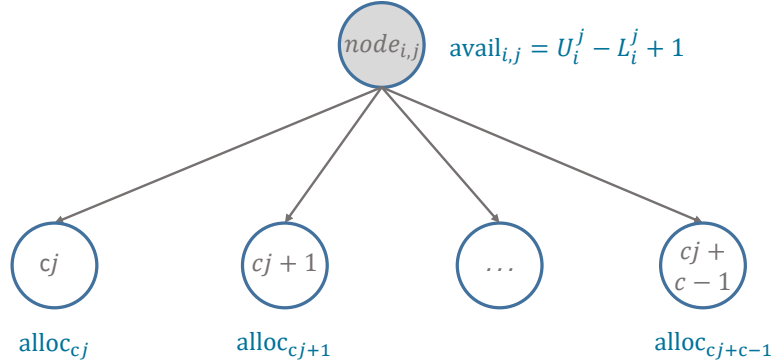


Figure 3.3: Calculating **avail** and **alloc** subspaces in the DTE.

The intuition behind Equation (3.3) is to allocate at least one seed for each sequence. Hence, as we move down to a branch of the tree we ensure that the interval size of this branch is at least equal to the total number of children nodes belonging to this branch. Considering this assumption, we initialize the available seed space as  $[0, 2^l - 1]$  in which  $l$  is the number of bits that is required to encode one sequence by DTE. Assuming that  $h_i$  is the number of nodes at level  $i$ ,  $l$  is calculated as  $\lceil \log_2(h_1 \times h_2 \times \dots \times h_n) \rceil$ .

Thus, assuming  $node_{i,j}$  has  $c$  children nodes, the intervals of its belonging children are calculated as follows:

- $[L_{i+1}^{c_j}, U_{i+1}^{c_j}] = [L_i^j, L_i^j + \mathbf{alloc}_{i,c_j} - 1]$
- $[L_{i+1}^{c_j+1}, U_{i+1}^{c_j+1}] = [L_i^j + \mathbf{alloc}_{i,c_j}, L_i^j + \mathbf{alloc}_{i,c_j} + \mathbf{alloc}_{i,c_j+1} - 1]$
- $[L_{i+1}^{c_j+2}, U_{i+1}^{c_j+2}] = [L_i^j + \sum_{t=0}^1 \mathbf{alloc}_{i,c_j+t}, L_i^j + \sum_{t=0}^2 \mathbf{alloc}_{i,c_j+t} - 1]$

...

- $[L_{i+1}^{c_j+c-1}, U_{i+1}^{c_j+3}] = [L_i^j + \sum_{t=0}^{c-2} \mathbf{alloc}_{i,c_j+t}, U_i^j]$

The interval is calculated at each level and encoding algorithm chooses a node based on the input message  $M$  at each level to expand and move forward. The algorithm will stop at a leaf node and returns a seed from its interval. We note that the above calculation is similar in all trees.

### Encoding (Example)

To give an illustration of what we have described until now, let's investigate the encoding process through an example. For the sake of the simplicity, we describe the proposed scheme over a PHR data with blood pressure, cholesterol level, and disease along with her/his drug lists. Suppose the following message  $M$  as the encode algorithm input:

$$M = \{ \{ \text{BP2, Chol4, Breast Cancer} \}, \{ 5 - \text{Fluorouracil, Doxorubicin, Cyclophosphamide} \} \},$$

that is PHR of a *female* patient whose *age* is in the range of 30 to 40.

We constructed two DTEs, illustrated in Figure 3.4, with similar tree structures: one for physiological attributes that is presented in Figure 3.4(a) and another DTE for the drugs list which is shown in Figure 3.4(b).

The DTE for physiological attributes (Figure 3.4(a)) consists of three levels, one level per each attribute in the message  $M$ . In this example the first level is for blood pressure (represented as  $\text{BP}_i$ ), the second level represents the cholesterol level ( $\text{Chol}_i$ ), and the third level for diseases.

Likewise, the drug tree (shown in Figure 3.4(b)) is constructed by considering the drugs sequences. The first level consists of a complete list of drugs, each node is expanded to other drugs node. If there isn't any further sequence for a specific

drug, we labeled its child node as *NaN* which means that the sequence no longer continues.

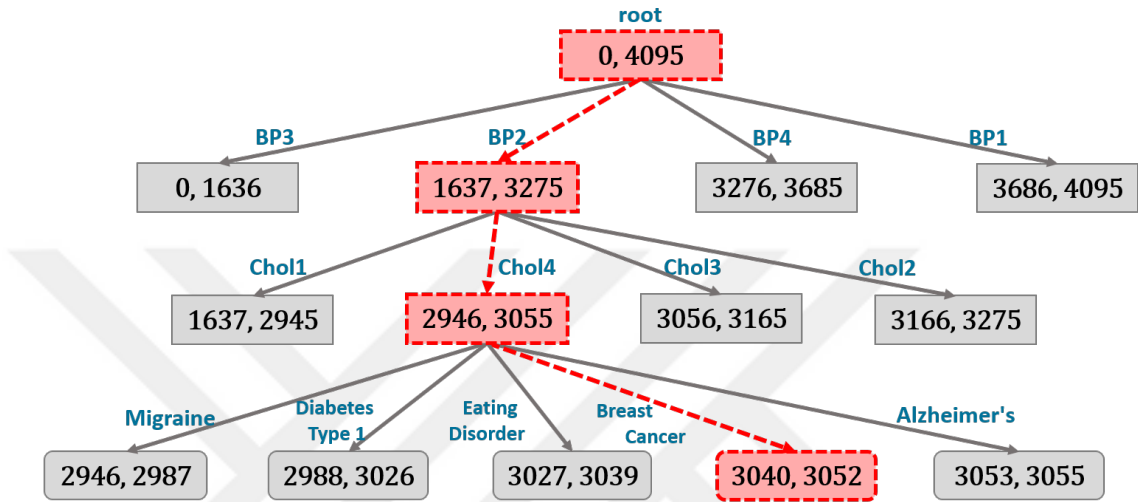
The purpose is to encode the message  $M$  with the following conditional probabilities, by taking into account that the patients' age are divided into 7 classes each consists of 10 years, such that *Age2* is the range of 30 – 40 years old.

$$\begin{aligned}
 (i) \quad & P(m_1 = \text{BP2} \mid \text{Female}, \text{Age2}) = 0.40 \\
 (ii) \quad & P(m_2 = \text{Chol14} \mid \text{Female}, \text{Age2}, \text{BP2}) = 0.67 \\
 (iii) \quad & P(m_3 = \text{Breast Cancer} \mid \text{Female}, \text{Age2}, \text{BP2}, \text{Chol14}) = 0.12 \quad (3.4)
 \end{aligned}$$

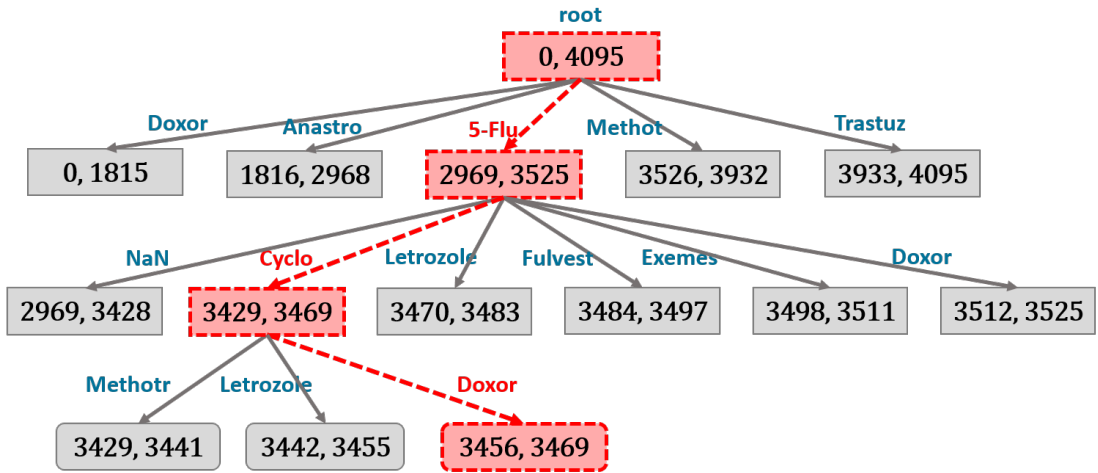
Considering  $a_1 = \text{BP2}$ , we start from the root node and move to node  $\text{BP}_2$  in the first level with the probability of 0.40. That is to say, the  $\text{BP}_2$  node's seed space is 40% of the root node. The algorithm proceeds to the second level and chooses **Chol14** with the value of 0.67. The last level is for diseases, in which the leaf node **Breast Cancer** is chosen with the probability of 0.12. Finally, the algorithm stops in this level, and returns a random integer as the seed from the leaf node's interval ([3040, 3052]). In this example the seed for physiological attributes is  $S_P = 3047$ .

Similarly, the encoding process proceeds for the drugs list (shown in Figure 3.4(b)). By knowing the probability of a drug given the proceeding ones, we trace the tree until we reach a leaf node. The tree in Figure 3.4(b) includes three levels and each level represents drug names of breast cancer. As mentioned, the drug's sequence to be encoded in this example is {5 – Fluorouracil, Doxorubicin, Cyclophosphamide}. We have considered the following conditional probabilities to trace the tree until a leaf node.

$$\begin{aligned}
 (i) \quad & P(a_1 = \text{'5 – Fluorouracil'}) = 0.13 \\
 (ii) \quad & P(a_2 = \text{'Doxorubicin'} \mid \text{'5 – Fluorouracil'}) = 0.07 \\
 (iii) \quad & P(a_3 = \text{'Cyclophosphamide'} \mid \text{'5 – Fluorouracil'}, \text{'Doxorubicin'}) = 0.33 \quad (3.5)
 \end{aligned}$$



(a) The DTE for health attributes: blood pressure, cholesterol, disease.



(b) The DTE fo drugs list.

Figure 3.4: A toy example of the encoding process. The message is for a female patient with the age range of 30 to 40 who suffers from breast cancer. The path through the seed is represented by a red dashed-line. When it reaches the leaf, we randomly choose a seed from the leaf interval as the seed in each tree. (a) Main tree that includes the health attributes and disease of a patient. (b) Drug tree that indicates the drug list of a patient.

Finally, the algorithm returns seed value of the drug tree,  $S_{Drugs} = 3461$ .

Note that in this example, we have chosen a sequence of three drugs, since it is the most common number of drugs for a patient in datasets that we have analyzed, however, the tree can be expanded easily by adding more levels.

The main seed at the end is the concatenation of  $S_H$  and  $S_D$ .

$$S = \{S_P || S_D\} = 30473461$$

After encoding process finishes, the seed is given to a password-based encryption (PBE) [21] function which encrypts the seed as a plain text under a patient-defined password.

### 3.2.1.2 Decoding

When a user sends a PHR retrieval request, the hospital resends the encrypted seed to the user API. The encrypted seed is first decrypted under the patient's provided password (Step 8 in Figure 3.2) and then fed into the decoding algorithm to generate the PHR. Therefore, the decoding algorithm takes  $S \in \mathcal{S}$  as an input and results a message  $M \in \mathcal{M}$  as the output.

Unlike encoding, decoding is a deterministic function that follows a similar process of the encoding algorithm. We first decompose the original seed and extract each seed (e.g.,  $S_D$ ) from the main seed. The system then feeds each seed into the corresponding DTE tree. Starting from the root of a tree (e.g., treatment), at each level the algorithm calculates the intervals based on the conditional probabilities (the same conditional probabilities that is introduced in the previous section). Therefore, the algorithm moves down the tree until it reaches the last level.

At each level of the tree, the algorithm compares the seed with each node's interval in that level. If the seed belongs to a node's interval that node is chosen to expand. That is to say, the algorithm chooses a  $node_{i,j}$  in a tree (e.g., treatment tree) if  $L_i^j \leq S_T < U_i^j$ . The process ends when the algorithm reaches a leaf node

and the output is the path from the root node to this leaf node.

### Decoding (Example)

Considering the previous example in Figure 3.4(a) that resulted in  $S = 30473461$ , we now decode the seed to find the corresponding message. First, the algorithm splits the seed into  $S_H = 3047$  and  $S_D = 3461$ , and then feeds each seed into the corresponding tree ( $S_H$  to the physiological attribute tree and  $S_D$  to the drug tree).

Starting from the root of the physiological attribute tree, the algorithm follows the conditional probabilities as in (3.4) and calculates the intervals for the first level of the tree. In this level,  $BP_2$  is chosen since  $L_1^2 \leq S_H < U_1^2$ . Next, the process continues on  $node_{1,2}$  by expanding its children nodes until the last level of the tree and ends in a leaf node. Finally, the decoding algorithm recovers the message by returning the path from root to the leaf node that is:

$$M_H = \{BP1, Chol4, Breast\ Cancer\}.$$

Similarly, the decoding process decodes  $Seed_D$ , by using the probabilities in Equation (3.5), which ends up with message:

$$M_D = \{5 - Fluorouracil, Doxorubicin, Cyclophosphamide\}.$$

Put it all together, decoding algorithm gets  $S = 30473461$  as an input and maps it to the corresponding message, hence, it produces  $M$  as follows:

$$M = \{\{BP2, Chol4, Breast\ Cancer\}, \{5 - Fluorouracil, Doxorubicin, Cyclophosphamide\}\}.$$

### 3.2.1.3 Encryption/Decryption

We use password-based encryption/decryption [21] for the system by following the standard PKCS #5 [55]. This method uses (i) HMAC-SHA-1 for the underlying pseudorandom function, (ii) a key derivation function,  $KDF$ , to generate a 128-bit key,  $DK$  for a given password  $P$ , and (iii) a 64-bit random salt  $R$ . The key derivation function is as follows:

$$DK = KDF(P, R)$$

$DK$  is used as a key for an AES block cipher that encrypts the seed in CBC mode.

### 3.2.2 PHR Update

As described earlier in this chapter, a PHR is mapped to a seed by using our proposed DTE. After encrypting the seed (under patient's password), it is stored in the hospital's database. Later, when a patient revisits the hospital, some of his data in the PHR might need to be updated. However, the DTE in HE does not support updating a data without fully decrypting the message or reconstructing the DTE. Furthermore, we assume that the hospital staff are not trustable and we are not willing to give them sensitive information regarding the patient's health-care unless they are authorized to have access. Due to this issue, we develop a protocol to update the attributes of a patient's PHR without leaking other sensitive information (e.g., diagnosis). To this end, we address one of the limitations of HE that is constructing a DTE for dynamic datasets.

The main purpose of PHR Update algorithm is to give permission to the staff of the hospital (e.g., radiologist, nurse, and who needs to update some attributes of PHR but does not have full access to it) to update the attributes without giving access to read the PHR.

We utilize Paillier cryptosystem, that we have described it in Chapter 2. to implement PHR Update algorithm. Two encrypted versions of a seed are stored

in the hospital database: with *homomorphic encryption* [25] and with *password-based encryption (PBE)* [21]. PBE encrypts the PHR under patient’s password that is most probably a low-entropy password [17]. Unlike PBE, homomorphic encryption uses a high entropy password that is generated by TA. We represent homomorphic encryption of a message  $m$  under patient’s secret key as  $\langle m \rangle$  and the PBE version under patient’s password as  $Enc(m)$ . From now on, we refer to  $S$  as the current seed and we represent the new seed as  $\hat{S}$ .

Each time that a seed  $S$  is generated by the DTE during the encoding process, a set  $V = \{\langle v_1 \rangle, \langle v_2 \rangle, \dots, \langle v_t \rangle\}$  is also constructed for each seed. Each  $v_j \in \{0, 1\}$  belongs to a leaf node in the DTE tree (which has  $t$  leaves at the last level  $n$ ), such that:

$$\langle v_j \rangle = \begin{cases} \langle 1 \rangle & \text{if } S \in [L_n^j, U_n^j], \\ \langle 0 \rangle & \text{otherwise.} \end{cases} \quad (3.6)$$

For instance, suppose that a DTE has 8 leaves, and the seed has been generated by the 3rd node. Hence,  $V$  is equal to:

$$V = \{\langle 0 \rangle, \langle 0 \rangle, \langle 1 \rangle, \langle 0 \rangle, \langle 0 \rangle, \langle 0 \rangle, \langle 0 \rangle, \langle 0 \rangle\}.$$

Assume that one of the staff of the hospital is authorized to access level  $i$  of the DTE tree. For example, a nurse who can update cholesterol level of a patient might access 2nd level of the tree (considering the example of Figure 3.4(a)). After measuring an attribute (e.g., cholesterol level), the new value of the attribute should be identified in the DTE. Hence, the new node that contains the new value of the attributes is recognized.

Figure 3.5 illustrates a general overview of the proposed PHR Update algorithm. The hospital sends encrypted information about the current seed (e.g., set  $V$ ) of a patient to the hospital staff (Step 1) who is responsible to update the patient’s PHR. After measuring the attribute (e.g., blood pressure), the responsible person should generate the encrypted (homomorphic version) of the new seed by applying some cryptographic operations on the encrypted information that s/he receives from the hospital at Step 1. S/he generates the homomorphic encryption



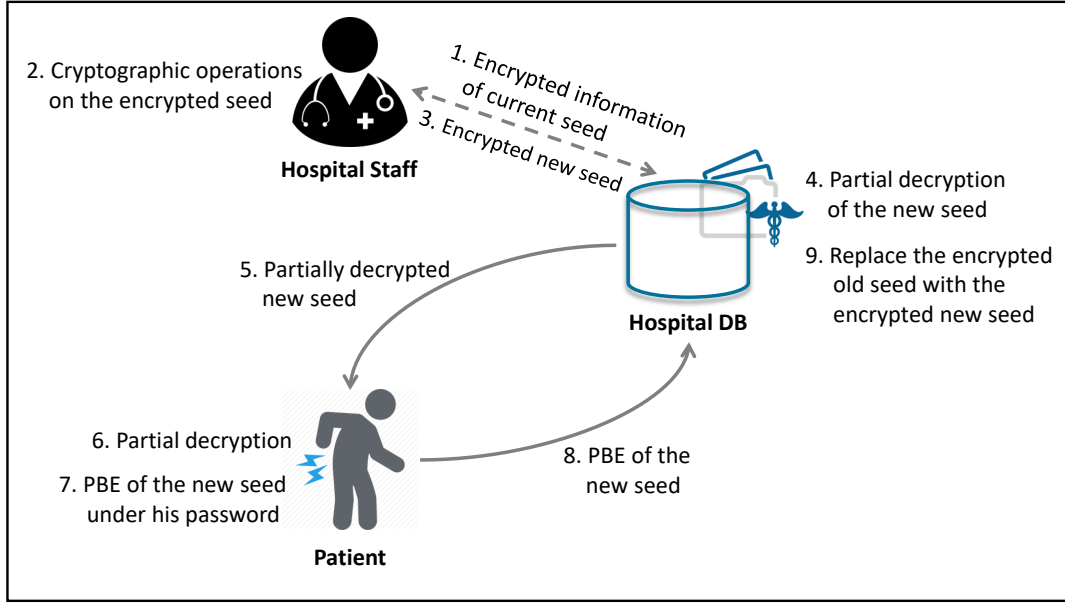


Figure 3.5: System model for updating health records.

of new seed,  $\langle \hat{S} \rangle$  then, sends it back to the hospital's database (Steps 2 and 3). Nonetheless we should also update the PBE version as well. In order to do so, the hospital partially decrypts the  $\langle \hat{S} \rangle$  under its own key and resends it to the patient (Steps 4 and 5). Similarly, the patient decrypts his own part and obtains the new seed (Steps 6). The patient then encrypts  $\hat{S}$  under his password to obtain  $Enc(\hat{S})$  (Steps 7).  $Enc(\hat{S})$  is then sent to the database and takes place of the previous seed (Steps 8 and 9).

We note that in a trivial solution the patient may receive the homomorphically encrypted new seed from the hospital staff and encrypt it under his provided password and then send it to the hospital database. However, we do not want to trust a single party for decrypting the seed. Hence, we benefit from partial decryption here to distribute the trust between the hospital and the patient so that there will be no single party who can decrypt the cipher text under Paillier.

In order to generate the homomorphic encryption of the new seed, the algorithm takes as inputs the following variables, considering that  $node_{i,j}$  is the new node that is measured by a hospital staff; **(i)** a set  $V$ , which represents the index of the current seed, **(ii)** the lower boundary values of the leaves (e.g.,  $L_n^j$ ), chosen from branches that belong to the new measured node ( $node_{i,j}$ ), **(iii)** the

total number of  $node_{i,j}$ 's branches that is  $C = c_{i+1} \times c_{i+2} \times \dots \times c_n$  where  $c_i$  is the number of children belong to each node at level  $i$ , **(iv)** random integers  $\{r', r'', \dots, r^C\}$  that are within the interval sizes of  $node_{i,j}$ 's leaves, **(v)** number of nodes at level  $i$  as  $h_i$ , and **(vi)**  $n$  represents the last level of the tree. The following function calculates the homomorphic encrypted new seed given the above-mentioned variables.

$$\begin{aligned}
\langle \hat{S} \rangle &= (L_n^{Cj} + r') \times \sum_{t=0}^{h_i-1} \langle v_{Ct} \rangle \\
&+ (L_n^{Cj+1} + r'') \times \sum_{t=0}^{h_i-1} \langle v_{Ct+1} \rangle \\
&+ (L_n^{Cj+2} + r''') \times \sum_{t=0}^{h_i-1} \langle v_{Ct+2} \rangle \\
&+ \dots \\
&+ (L_n^{Cj+C-1} + r^{(C)}) \times \sum_{t=0}^{h_i-1} \langle v_{Ct+C-1} \rangle.
\end{aligned} \tag{3.7}$$

In a nutshell, the PHR Update protocol calculates  $\langle \hat{S} \rangle$  by shifting the previous seed value to a new interval. Benefiting from the above Function, homomorphic operations (multiplication and addition) are applied on a set  $V$  to obtain the homomorphic encrypted value of the new seed.

After updating the seed,  $V$  should also be updated to keep the new seed's index for later updates. We again assume that the new node that is measured by the hospital staff is  $node_{i,j}$  with  $h_i$  nodes at level  $i$  (similarly  $h_n$  nodes at the last level  $n$ ). Considering the same notations in Function (3.7),  $C = c_{i+1} \times c_{i+2} \times \dots \times c_n$  is the total number of leaves (or branches) that belong to  $node_{i,j}$ . The new set  $\hat{V}$  then is calculated as follows:

$$\text{For } k \in \{0, 1, 2, \dots, h_n\}, \quad \langle \hat{v}_k \rangle = \begin{cases} \sum_{t=0}^{h_i-1} \langle v_{Ct+k-C} \rangle & \text{if } Cj \leq k \leq Cj + C - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

We describe the PHR Update algorithm in more details in Algorithm 1 to make clear the role of each party.

---

**Algorithm 1** PHR Update

---

```

1: procedure PHR UPDATE
2: TA:
3:    $[PK, x] \leftarrow \text{KEYGENERATION}()$ 
4:    $x_1 = \text{GENRANDOM}$  from  $[0, x]$ 
5:    $x_2 = x - x_1$ 
6: Hospital:
7:    $\text{paillier} \leftarrow \text{new PAILLIERSCHEME}(PK, x)$ 
8:    $\langle \hat{S} \rangle_x \leftarrow \text{paillier.HOMOMORPHIC}(\langle V \rangle, \text{node}_{i,j})$ 
9:    $\langle \hat{V} \rangle \leftarrow \text{UPDATE}(\langle V \rangle)$ 
10:   $\langle \hat{S} \rangle_{x_2} \leftarrow \text{paillier.PARTIALDEC}(\langle \hat{S} \rangle_x, x_1)$ 
11: Patient:
12:   $\hat{S} \leftarrow \text{paillier.PARTIALDEC}(\langle \hat{S} \rangle_{x_2}, x_2)$ 
13:   $\text{Enc}(\hat{S}) \leftarrow \text{PBE}(\hat{S}, P)$ 
14:
15:   $\{\langle S \rangle, V, \text{Enc}(S)\}$  replaced by  $\{\langle \hat{S} \rangle, \hat{V}, \text{Enc}(\hat{S})\}$ 

```

---

**Step 1 (@ TA).** TA produces public ( $PK$ ) and secret ( $x$ ) keys by running KEYGENERATION function. TA then divides  $x$  into  $x_1$  and  $x_2$  for the hospital and the patient respectively in a way that  $x = x_1 + x_2$ .

**Step 2 (@ Hospital).** A responsible person at the hospital, who wishes to update the data (e.g, specialist), measures the health attributes and returns the new node of the measured attributes to the hospital. Knowing set  $V$  of the current seed, by using Paillier scheme some homomorphic operations (as shown in Function (3.7)) is applied to generate the homomorphic encryption of the new seed ( $\langle \hat{S} \rangle$ ). Moreover, in order to keep the index of the new seed for later updates, a set  $\hat{V}$  is also constructed by applying Function (3.8). Finally, hospital partially decrypts  $\langle \hat{S} \rangle_x$  using its secret key  $x_1$  (by PARTIALDEC function) and sends  $\langle \hat{S} \rangle_{x_2}$  to the patient to update  $\text{Enc}(S)$ .

**Step 3 (@ Patient).** The patient applies PARTIALDEC function and partially decrypts  $\langle \hat{S} \rangle_{x_2}$  using her/his own part of the key  $x_2$ , and obtains the decrypted value of the new seed ( $\hat{S}$ ). The patient encrypts  $\hat{S}$  under his/her password  $P$  and sends  $\text{Enc}(\hat{S})$  back to the hospital.

The update process finishes when the hospital replaces old values ( $\langle S \rangle, V$ , and

$Enc(S)$  with the new ones ( $\langle \hat{S} \rangle$ ,  $\hat{V}$ , and  $Enc(\hat{S})$ ) in its database.

Note that in real-world scenarios, the history of a patient's PHR is important for further investigations on her/his health status. With this in mind, we provide a time stamp for each seed (which represents the patient's PHR in our system) of a patient before updating it. Therefore, we keep all the seed regarding a patient in the hospital's database without keeping the whole attributes of a PHR. This is an efficient way in terms of memory complexity.



# Chapter 4

## Evaluation

In order to construct a good DTE to map PHR data to a uniform space, it is necessary to understand the message space distribution. To this end, we build our model based on various datasets in order to compute the conditional probabilities in Equation (3.2). Furthermore, we compute the correlations between the used drugs and demographic attributes (e.g., gender) of patients from the datasets, and the relationship between a drug with the other drugs.

### 4.1 Data Model

**PatientsLikeMe**<sup>1</sup>. We used PatientsLikeMe social network in which patients connect with others to evaluate their treatments. Users share their experiences with other patients who have similar diseases in order to improve their knowledge and experiences.

We crawled the social network and gathered more than 8.3K patients profiles with different diseases. The profile of users consists of various attributes such as user name of a patient, location, gender, age, condition, and a free text that a user writes about himself. Analyzing the dataset, we end up with 386 different diseases.

---

<sup>1</sup><https://www.patientslikeme.com>

We categorized the demographic attributes of individuals into different groups: **(i)** by *age* from 20 to more than 90 in 7 groups each containing a range of ten years, and **(ii)** by *gender*. Benefiting from this dataset we were able to find the conditional probability of a disease given the age range and gender of a patient. Hence, in general we build 14 DTEs for each age range and gender combination.

**The Cancer Genome Atlas (TCGA)**<sup>2</sup>. TCGA is a project that includes genetic mutations responsible for cancer in order to improve diagnose, treatment, and prevent cancer through a better understanding of the genetic basis of this disease. The project is supervised by the National Cancer Institute Center for Cancer Genomics and the National Human Genome Research Institute funded by the US government.

We have benefited from clinical drug information of TCGA breast cancer dataset of 2.4k patients. The dataset includes patients' information such as userid, gender, age, etc. We utilize this dataset to model a patient's drugs list who is diagnosed with breast cancer.

After preprocessing the dataset and extracting the unique users and drugs, we ended up with 771 patients and 54 unique drugs for breast cancer. Each patient uses at least one drug and at most 6 drugs.

**Physiological Variables Dataset.** Recently some studies [53, 54] find correlations between the personal attributes such as age and gender with physiological attributes such as blood pressure, blood glucose, etc. investigating various datasets.

Yashin *et al.* discover the patterns of individuals' aging in terms of physiological variables [1]. They provide 8 variables associated with gender and age of individuals which are illustrated in Figure 4.1. They have utilized the Framingham Heart Study (FHS) dataset, which includes detailed medical histories and exams.

As shown in Figure 4.1, the pattern of physiological attributes are dependent not only on the age, but also on the gender of an individual. We have utilized diastolic blood pressure and cholesterol level attributes of this study to model the health variables of PHRs and in order to estimate the conditional probability of

---

<sup>2</sup><https://gdc.cancer.gov/>

a blood pressure and cholesterol level given the gender and age of a patient.

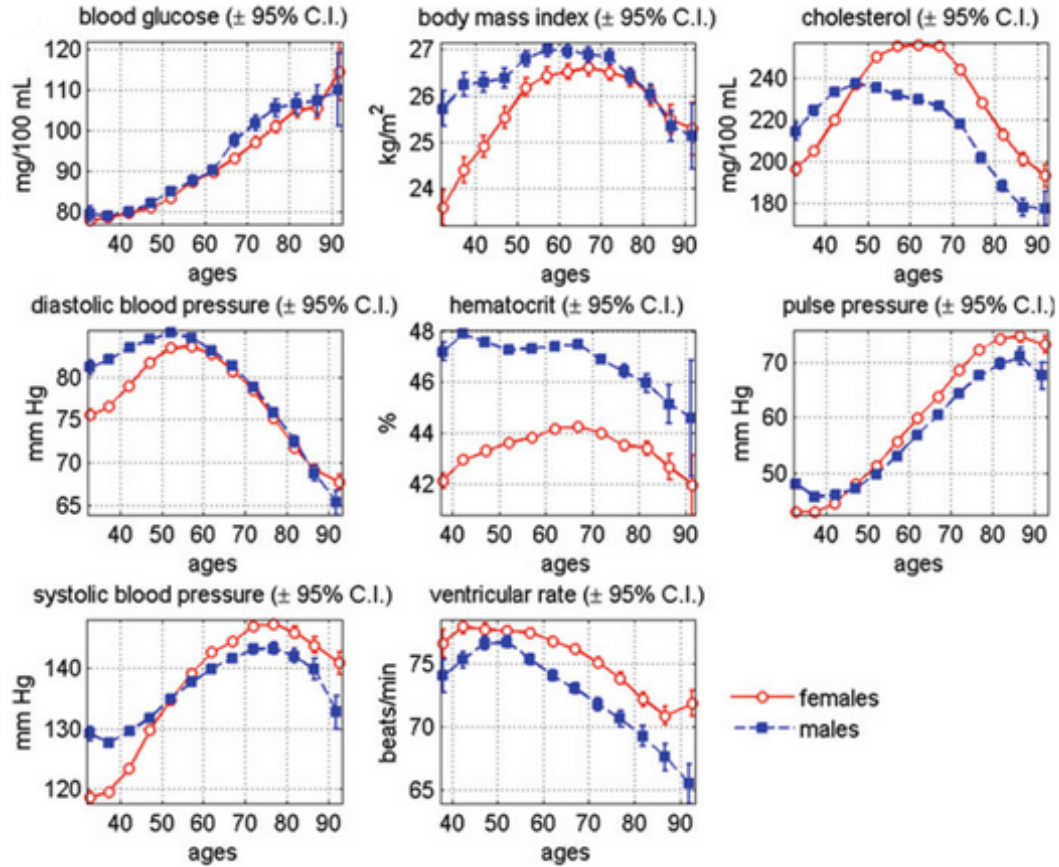


Figure 4.1: Average age trajectories of eight physiological attributes for males and females [1].

We quantized the values of blood pressure from 65 to 85 mm Hg, into 4 ranges. Similarly, we divided the cholesterol level from 180 to 260 mg/100 ml into 4 ranges.

## 4.2 Correlations between the Values

Investigating the breast cancer drug's dataset, we discovered correlations between the age of a patient and her/his drug usage. Figure 4.2 illustrates some of the

drugs' frequency within 3 age ranges ( $< 45$  &  $> 14$ ), ( $< 65$  &  $> 44$ ), ( $> 65$ ) based on the standard population metrics. As shown in this figure, there is a correlation between drugs and the age range. For instance, *Taxotere* and *Docetaxel* are mostly prescribed to patients who are under 45 years old. Nonetheless, *5-Fluorouracil* is mostly used among the mid age patients (ages between 44 and 65) and *Paclitaxel* for the older ages (more than 64).

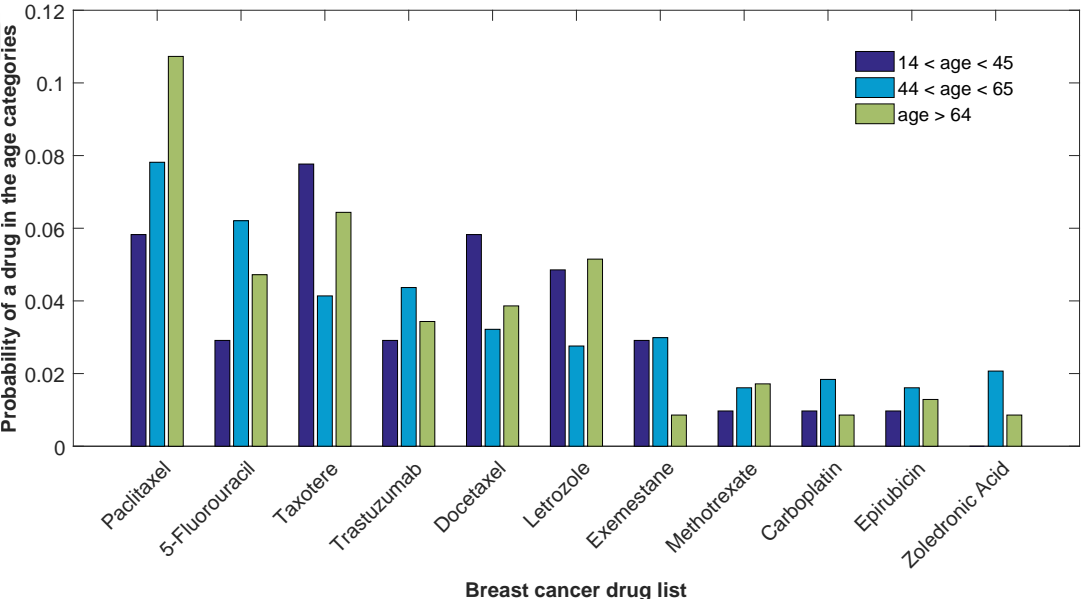


Figure 4.2: The relationship between different drugs and age.

Furthermore, we investigate the drugs combination and the pairwise correlations of drugs. Figure 4.3 illustrates the relationship of a drug with other drugs in the TCGA dataset for breast cancer. It describes the rate of two drugs combination. In another word, in the total two by two combinations of drugs, this figure represents which drug is more popular to be used with a specific drug. For instance, in 60% of the cases 5 – Fluorouracil and Letrozole are used together.



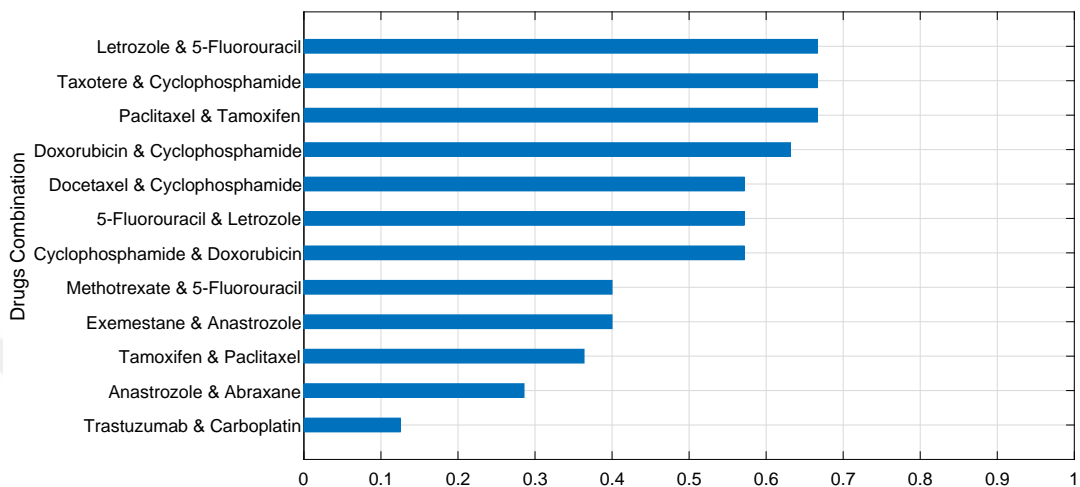


Figure 4.3: Pairwise correlations of drugs.

### 4.3 Performance

We implemented our system on Matlab (encoding and decoding), Python (encryption and decryption), and Java (Paillier cryptosystem) by using the datasets that we have described earlier. Herein, we quantize the system performance within the two algorithms that we have introduced in our model, *PHR Retrieval* and *PHR Update*. We evaluated the system on a sample of 386 different diseases and 771 patients drug usage.

The proposed system does not have a storage overhead since one of the important key feature of the structure that we used as the DTE is that we do not store the DTE tree. The public knowledge of the probabilities is the important feature to construct the DTE. Hence, given the probabilities, each time in an encoding process of a PHR, a branch of the DTE is constructed by the system to obtain the seed of the corresponding PHR. Therefore, the memory complexity is  $O(n)$  where  $n$  is the length of PHR sequence.

We evaluate the performance of both retrieval and update algorithms on a server consists of 478 processors each with 2.30GHz Intel Xeon CPU E5-2650 and Ubuntu 14.04.4 LTS system. We compute the time complexity of each algorithm separately as discussed below.

As for *PHR Retrieval* process, we compute the time complexity of the main four blocks (*encode*, *decode*, *PBE encryption*, *PBE decryption*) of this algorithm for two DTEs, physiological variables and drugs, considering a PHR as a record of 6 variables (three physiological variables and a list of three drugs). The average running time is reported in Figure 4.4. Overall, in both DTE the time complexity is depending on the length of a PHR data (depth of the tree) and it is not significant for a sequence of three attributes PHR.

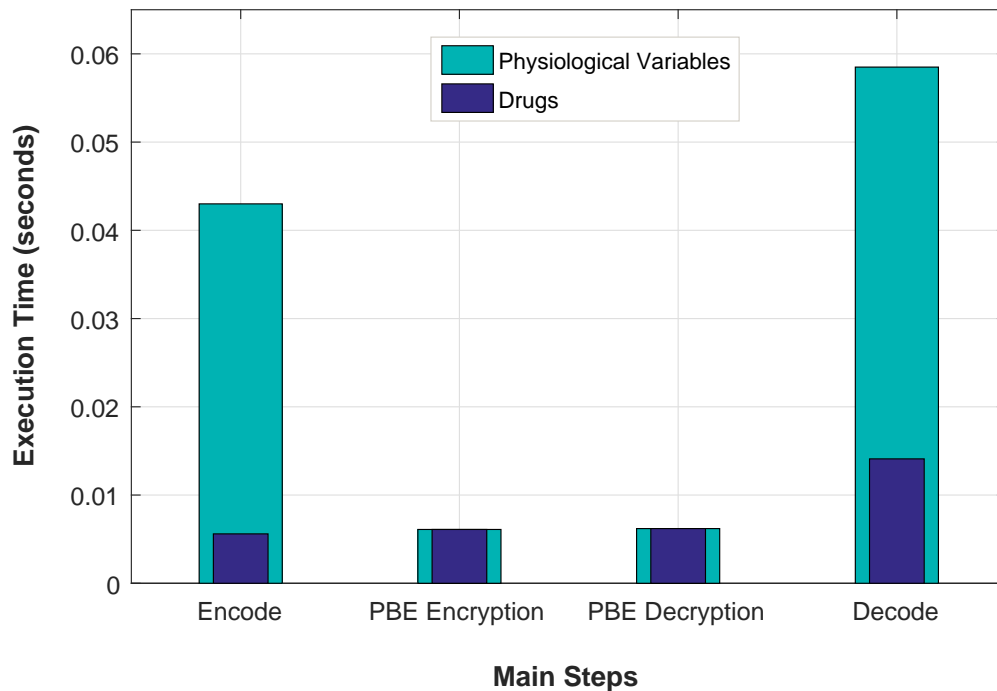


Figure 4.4: Performance of the PHR Retrieval algorithm on the physiological variables and drugs list. The figure represents the execution time for 3-layer trees of physiological variables and drugs list which described in Figure 3.4. In both DTEs encoding and decoding take more time in comparison to encryption and decryption.

Comparing the main blocks of *PHR Retrieval* algorithm, the most expensive phases are the encoding and decoding blocks due to the calculation of the intervals based on the conditional probabilities. The differences between physiological variables and drugs DTE during the encoding/decoding is due to the number of

nodes in each level of the trees. Since the drugs tree has more number of nodes in comparison with the physiological tree, the encoding/decoding processes are more costly. Moreover, by increasing the number of attributes and the tree levels, the encoding and decoding process time is increased linearly as well. Nevertheless, the encryption and decryption time remains similar since it does not rely on the DTE structure.

We compute the running time of *PHR Update* procedure on physiological variables and drugs DTEs as well. We calculate the running time and present it in Table 4.1. The execution time of this algorithm is highly depend on the cryptographic operations on encrypted data.

	1st level time (sec)	2nd level time (sec)
<b>Phys DTE - 3 levels</b>	34.01	8.35
<b>Phys DTE - 10 levels</b>	83.36	70.19
<b>Drugs DTE - 3 levels</b>	27.80	2.90
<b>Drugs DTE 6 levels</b>	39.90	0.70

Table 4.1: Performance of the PHR Update algorithm on the physiological variables (represented as Phys Var) and drugs list DTEs.

We consider different levels of a each tree in our evaluations, for instance, in physiological tree the update cost of the first level (blood pressure) is more than the second layer (cholesterol level) since the number of nodes in a branch of a blood pressure node is more than second layer for cholesterol. Hence, the running time of *PHR Update* algorithm is dependent on the number of nodes, especially the nodes on the last level of a DTE. As the number of leaves grows in a DTE the cryptographic operations take more time. As we increased the number of levels in a DTE, the running time of the PHR Update algorithm increases as well. To evaluate the performance for a DTE with more levels, we assumed a physiological tree with 10 layers of health attributes and drugs’ tree with 6 layers.

The time complexity of *PHR Update* algorithm is highly dependent on the layer which needs to be updated. As we get close to the root the number of nodes that should be computed is increased. As from the result in Table 4.1, updating

blood pressure is more time effective than cholesterol level in the physiological variable DTE since it is closer to the root. Hence, the organization of the levels is an important issue in terms of time complexity. To this end, we reorganize the tree structure by putting the static levels with more values and close to the root and recalculate the execution time again. The results are represented in Table 4.2. As it is shown in this table, the running time dramatically decreased to less than a second after the changing the tree structure.

	1st level time (sec)	2nd level time (sec)
<b>Phys DTE - 3 levels</b>	0.20	0.11
<b>Phys DTE - 10 levels</b>	0.33	0.26

Table 4.2: Improved performance after reorganizing the physiological variables DTE. The running time decreased dramatically after taking the the dynamic levels close to the root and the static levels close to the root.

Hence, in order to make an efficient system we propose an organization in which the dynamic attributes are close to the leaves and the attributes that are change rarely near to the root.

# Chapter 5

## Security Analysis

In this chapter, we analyze the security of our proposed system regarding the DTE model in Chapter 3. In the following subsections, we analyze the goodness of the proposed DTE and the security of the framework against different kinds of brute-force attacks.

### 5.1 Measure for DTE Security

At each step  $i$ , the encoding algorithm allocates a seed space of size  $\lceil b \rceil^{n-i-1}$  to a branch in that step and the next step separates an input interval into different portions for each children of the sub-tree at step  $i$ . Hence, as discussed in Chapter 3, at least one integer is assigned for the sequences under the branch.

The main goal of constructing a DTE is to transform a non-uniform distributed message to a uniform space. Therefore, a secure DTE is the one that provides a close sampling to the real one so that for any  $M \in \mathcal{M}$ ,  $Pr[\mathbf{decode}(\mathbf{encode}(M)) = M] = 1$ . Hence, the decoding function should provide a close distribution between the target message distribution  $p_m$  to the DTE distribution which we define as follows:

$$p_d(M) = P[M' = M : S \leftarrow_{\S} \mathcal{S} ; M' \leftarrow \mathbf{decode}(S)]. \quad (5.1)$$

A good and secure DTE, is the one in which  $p_m$  and  $p_d$  distributions are close to each other. Herein, we quantify the difference between these two distributions,  $p_m$  and  $p_d$  in the proposed DTE model. We define  $P_m^i$  as the original probability of the prefix sequence  $M_{1,i}$  such that:

$$P_m^i = \sum_{M' \in \mathcal{M}, M'_{1,i} = M_{1,i}} p_m(M').$$

Similarly, we define  $P_d^i$  in the distribution  $p_d$ .

**Lemma 1.** The largest difference between  $p_m(M)$  and  $p_d(M)$  bounds the DTE advantage of an adversary, hence:

$$\forall M \in \mathcal{M}, |p_m(M) - p_d(M)| < \frac{1}{2^{l-n}}.$$

The proof is similar to Lemma 1 of [48].

Therefore, Lemma 1 encloses the largest difference between  $p_m(M)$  and  $p_d(M)$ . It also results the following theorem that bounds the DTE advantage of an attacker, introduced by honey encryption. The DTE advantage is formally define by the following definition.

**Definition 1 (DTE goodness).** Let  $\mathcal{A}$  be an adversary attempting to distinguish between the two games shown in Figure 5.1. We determine the advantage of an adversary  $\mathcal{A}$  for a message distribution  $p_m$  and encoding scheme  $\mathbf{DTE} = [\mathbf{encode}, \mathbf{decode}]$  by

$$Adv_{(DTE, p_m)}^{dte}(\mathcal{A}) = |Pr[SAMP1_{(DTE, p_m)}^{\mathcal{A}} \Rightarrow 1] - Pr[SAMP0_{(DTE)}^{\mathcal{A}} \Rightarrow 1]|.$$

**Theorem 1.** Let  $p_m$  be the target message distribution and  $\mathbf{DTE} = [\mathbf{encode}, \mathbf{decode}]$  be the transformation scheme using  $l$  bits for encoding. Let  $\mathcal{A}$  be any sampling adversary, then

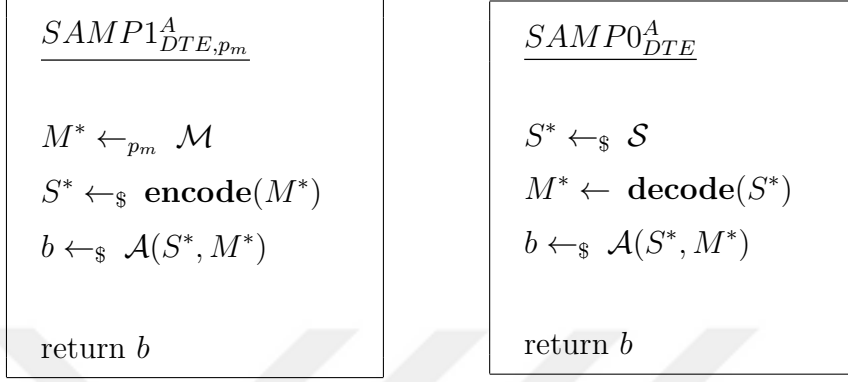


Figure 5.1: Games defining DTE goodness. In  $SAMP0_{DTE,p_m}^A$ , sequence  $M^*$  is sampled according to the target distribution,  $p_m$ . However, in  $SAMP1_{DTE}^A$ ,  $M^*$  is sampled from  $p_d$ . The adversary result in  $b$  that is either 0 or 1 which indicates his guess on whether he is in  $SAMP0_{DTE,p_m}^A$  or  $SAMP1_{DTE}^A$ .

$$Adv_{(DTE,p_m)}^{dte}(\mathcal{A}) \leq \frac{b^n}{2^l}.$$

The proof follows Theorem 6 in [20] and Theorem 1 in [48].

## Message Recovery Security

In order to formalize the security goals, we use the quantification of security against message recovery (MR) attacks. The purpose is to provide, given encryption of a message, the probability of any adversary recovering the correct message is negligible.

**Definition 2.** Let  $\mathcal{B}$  be an adversary attempting to recover the correct sequence from the given honey encryption sequence. The MR security is defined as a game shown in Figure 5.2. The advantage of  $\mathcal{B}$  against HE scheme is

$$Adv_{(HE,p_m,p_k)}^{mr}(\mathcal{B}) = Pr[MR_{(HE,p_m,p_k)}^A \Rightarrow True],$$

where,  $p_k$  is the password distribution that is non-uniform. We suppose that

$$\begin{array}{l}
\hline
MR_{DTE, p_m, p_k}^B \\
\hline
K^* \leftarrow_{p_k} \mathcal{K} \\
M^* \leftarrow_{p_m} \mathcal{M} \\
C^* \leftarrow_{\$} \mathbf{HEnc}(K^*, M^*) \\
K^* \leftarrow_{\$} \mathcal{B}(C^*) \\
\hline
\text{return } M = M^*
\end{array}$$

Figure 5.2: Game defining MR security. Given ciphertext  $C^*$  that is encrypted from  $M^*$  under  $K^*$ , adversary  $\mathcal{B}$  is allowed to predict the message by brute-force attacks. In case  $\mathcal{B}$ 's output, message  $M$  is equal to the original message  $M^*$ , then he wins the game.

the most probable password has a probability  $w$ . the following theorem is provided by using Lemma 1 and Theorem 1. The detailed information about the following theorem is available in [48, 20].

**Theorem 2.** Consider  $\mathbf{HE}[\mathbf{DTE}, H]$ , where  $H$  is the hash function, modeled as a random oracle and  $\mathbf{DTE}$  using a  $l$ -bit representation. Let  $p_m$  be the sequence distribution with maximum sequence probability  $\gamma$ , and  $p_k$  be a key distribution with maximum weight  $w$ . Suppose  $\alpha = \lceil 1/w \rceil$ . Then for any adversary  $\mathcal{B}$ ,

$$Adv_{(HE, p_m, p_k)}^{mr}(\mathcal{B}) \leq w(1 + \gamma) + \frac{x}{y}, \quad (5.2)$$

where  $\gamma = \frac{\bar{\alpha}^2}{2c} + \frac{e\bar{\alpha}^4}{27c^2}(1 - \frac{e\bar{\alpha}^2}{c^2})$ ,  $\bar{a} = \lceil 3/w \rceil$ , and  $\bar{b} = \lfloor 2/\gamma \rfloor$ .

The proof is similar to Corollary 1 in [20] and Theorem 2 in [48].



## 5.2 Security under Brute-force Attacks

Herein, we evaluate the security guarantee of our proposed system. To implement the system, we have used two encryption methods, password-based encryption (PBE) [21] and homomorphic encryption. As we discuss before, PBE encrypts a message under a weak password that is provided by the patient. Nonetheless, the homomorphic encryption encrypts the message under a high entropy key that is generated by the TA. Hence, we focus on evaluating PBE which is more vulnerable against brute-force attacks.

To this end, we prepare two experiments. Our aim is to compare the proposed system, that includes an extra hedge, with the conventional password-based encryption (PBE) [21] under brute-force attacks. As for the conventional PBE, we use our proposed DTE by setting all probabilities equal in all of the tree edges. For this PBE setting, decrypting a ciphertext returns low probability messages that are easy to classify as the wrong decrypted messages.

For both experiments, we benefit from TCGA and PLM datasets in order to evaluate the security of our system against conventional PBE. We encrypted a patient’s PHR under a given password and implemented a simple brute-force attack. We assume that the password size is 1000 (from 1 to 1000) and the patient’s password is “550”, that is the correct password for both experiments. The attacker knows about the password pool and s/he performs a brute-force attack by trying all possible passwords from the password pool.

First experiment is an implementation of direct PBE (we encode the data with a uniform DTE). Whereas, in the second experiment, we encrypted the patient’s PHR with the proposed system. Results of both experiments are shown in Figure 5.3. In both experiments, we calculated the interval size of the output sequence and interpreted the results.

As illustrated in Figure 5.3(a), the output of decryption without using DTE represents the fact that it is easy to exclude the irrelevant sequences since they have lower probabilities than the true sequence. Hence, the attacker can easily remove the password that results in wrong messages and reach the correct key. On the other hand, in the second experiment that uses our model (Figure 5.3(b)), the correct sequence is surrounded by the wrong but valid-looking sequences.

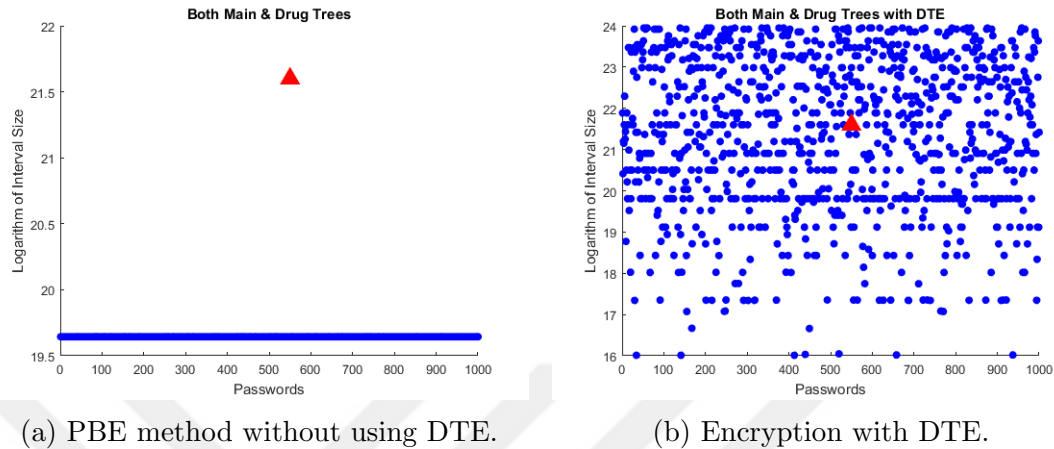


Figure 5.3: A simple brute-force attack to compare the conventional PBE and our proposed system.

Hence, we dramatically decrease the possibility of classifying the low probability sequences and make it difficult and almost impossible for the adversary to exclude the incorrect passwords. However, having background knowledge about a PHR by an attacker can cause some privacy leakage. This scenario leads to our next experiment in which we evaluate our system against an adversary (e.g., nurse, pharmacist) who has some information of a patient’s PHR.

Furthermore, the proposed PHR Update scheme preserving security and privacy that is relying on Paillier cryptosystem. The detail discussion regarding the security preserving of Paillier Cryptosystem is described in Theorem 9 to 11 of [23].

### 5.3 Security Analysis with Side Information

A patient’s demographic attributes such as age and gender can be known by an adversary. There are other health attributes that may be exposed during the treatment process of a patient, updating the attributes in her/his PHRs.

Herein, we evaluate our proposed scheme for PHR storage against attacks in which an adversary has background knowledge about the health attributes of a PHR as side information. We assume a set of possible values of a health attribute

that are such as  $\{K_1, K_2, \dots, K_u\}$ . For instance, considering cholesterol level the possible values are  $\{\text{Chol1}, \text{Chol2}, \text{Chol3}, \text{Chol4}\}$ . Let  $P_{K_i}$  represents the prior probability of a value  $K_i$  of an health attribute. The adversary performs a brute-force attack and tries each password to decrypt the ciphertext. S/he outputs the result sequence and keeps the corresponding password if the sequence matches the victim's demographic attributes, otherwise the adversary excludes the password from the password pool. We assume that the attacker makes a binary decision on keeping the password.

The order usually represents a password's rank. Studies that have been done on real-life password distributions [56, 57] prove that password distribution follows the Zipf's law [58, 59]. Based on Zipf's law, given some corpus of natural language iteration, the frequency of any word is inversely proportional to its rank in the frequency table, hence, the rank-frequency distribution is an inverse relation (e.g., the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.). Therefore, for the password distribution within a dataset, the probability of the  $i$ -th password (password with rank  $i$ ) is as follows:

$$P_i = Zi^{-r}, \quad (5.3)$$

where  $Z$  and  $r$  are constant values depending on the dataset. This is also the password distribution,  $p_k$ , that we discuss about in previous chapters.

Assume that there are totally  $t$  unique passwords in the password pool which are sorted in descending order regarding their probabilities such that:

$$P_1 \geq P_2 \geq P_3 \geq \dots \geq P_t, \text{ and } \sum_{i=1}^t P_i = 1.$$

Let  $K^*$  be the value of a health attribute of a victim which is exposed to an adversary, and the decryption under a incorrect key outputs  $K_i$  with probability  $P_{K_i}$  (note that this assignment is independent across passwords). The probability of retaining a password is calculated by an independent Bernoulli trials across passwords computed as

$$P_{ret} = \sum_{i=1}^t P_{K_i}. \quad (5.4)$$

From Theorem 2 we know that the advantage of adversary  $\mathcal{B}$  without side information is nearly equal to  $w$  that is the maximum weight of a password in the password distribution,  $p_k$  ( $w$  is equal to the  $P_1$  that is introduced before). Suppose  $\mathcal{B}'$  as the adversary with side information  $K^*$ . The adversary first filters the passwords by computing  $P_{ret}$  in Equation (5.4) and then follows the algorithm for adversary  $\mathcal{B}$  in the message recovery game which is represented in Figure 5.2 on the remaining passwords. Suppose  $p'_k$  as the password distribution for the retaining passwords, and with the maximum weight as  $w'$ . We randomize the password pruning process as a function  $f(p_k) \rightarrow p'_k$ . Hence, adversary  $\mathcal{B}'$  computes  $p'_k$  by using randomized function  $f$  and sends it to adversary  $\mathcal{B}$ . The advantage of adversary  $\mathcal{B}'$  is called  $\text{Adv}(\mathcal{B}')$  calculated as follows:

$$\begin{aligned} \text{Adv}(\mathcal{B}') &= E_{p'_k \leftarrow f(p_k)} [\text{Adv}_{HE, p_m, p'_k}^{MR}(\mathcal{B})] \\ &\approx E_{p'_k \leftarrow f(p_k)} [w'], \end{aligned} \quad (5.5)$$

where  $E$  is the expectation over the randomized password elimination process, and we approximate  $\text{Adv}_{HE, p_m, p'_k}^{MR}(\mathcal{B})$  with the maximum weight  $w'$  in the password distribution  $p'_k$ . Next, we quantify  $\text{Adv}(\mathcal{B}')$  using real data.

In order to quantify the  $\text{Adv}(\mathcal{B}')$  with a real data, we benefit from the Zipf's model for password distribution which is proposed by Wang *et. al.* in [58], with the following settings:  $t = 486118$ ,  $Z = 0.037871$  and  $r = 0.905773$ .

We perform different experiments for the attributes that a PHR holds. We start with physiological variables and evaluate the privacy loss by considering each variable as known information for the adversary. Similarly, we do the experiment with drug information.

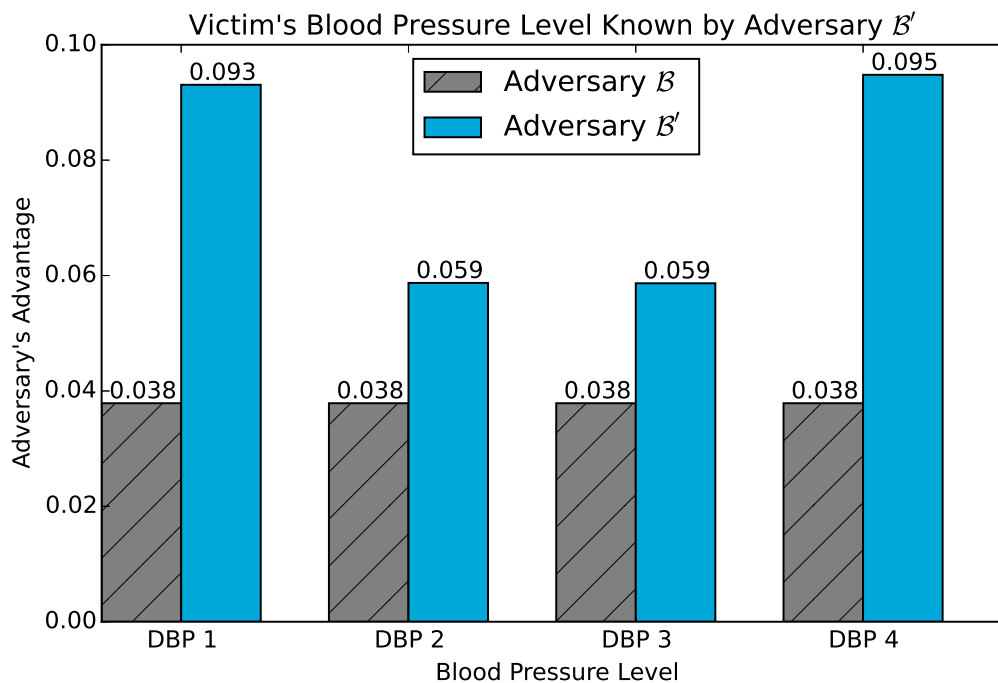


Figure 5.4: Evaluation of adversary’s advantage with blood pressure level as the side information. Adversary  $\mathcal{B}$  has no side information and his advantage is approximately  $w = 0.0379$  ( $w$  is the maximum weight of a password in the password distribution,  $p_k$ ) while adversary  $\mathcal{B}'$  has the blood pressure as the side information.

### 5.3.1 Physiological Variables

Suppose that one of the hospital staff has some background about a patient’s PHR during the measurements and tests in the hospital. Herein, we consider blood pressure and cholesterol level as the attributes that are exposed to the adversary. We conduct the Bernoulli trials with corresponding  $P_{ret}$ , as in Equation (5.4), on the password pool. We consider the adversary’s advantage without side information as  $w$  that is the maximum weight of a password in the password distribution,  $p_k$ , which is approximately equal to  $w = 0.038$ , and then we calculate  $\text{Adv}(\mathcal{B}')$  in Equation (5.5) by repeating the whole experiment 1500 times for each blood pressure (or cholesterol) level and we illustrate the average as the result for each attribute.

First, we calculate the average of adversary advantage when the adversary only

knows about the blood pressure of a patient. The prior probabilities for blood pressure ranges are as follows:

$$(K^*, P_{K^*}) = \{(\text{BP}_1, 0.10), (\text{BP}_2, 0.40), (\text{BP}_3, 0.40), (\text{BP}_4, 0.10)\}.$$

Figure 5.4 represents the privacy loss after assuming that the blood pressure attribute is exposed to the adversary. The worst cases among the blood pressure ranges are  $\text{BP}_1$  and  $\text{BP}_4$  information in which the adversary's advantage increased from 0.038 (that is the advantage of the adversary without any information) to 0.095. The reason for observing this increase for these blood pressure values is due to the low prior probability which results in a small  $P_{ret}$  per each of the values.

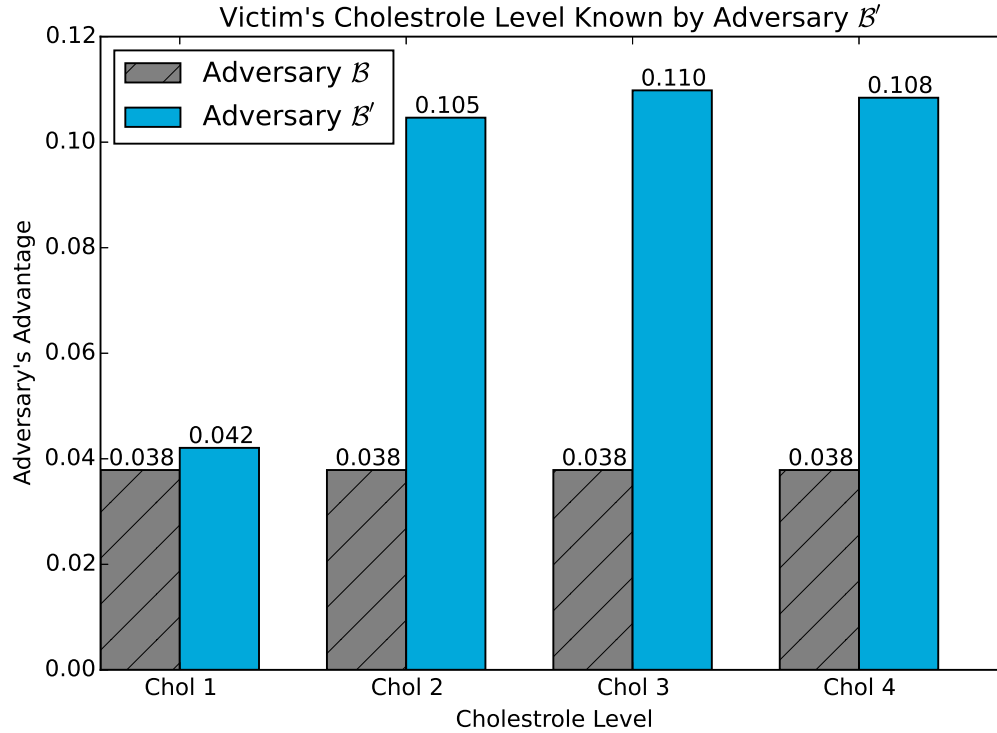


Figure 5.5: Evaluation of adversary's advantage with cholesterol level as the side information. Adversary  $\mathcal{B}$  has no side information and his advantage is approximately  $w = 0.0379$  ( $w$  is the maximum weight of a password in the password distribution,  $p_k$ ) while adversary  $\mathcal{B}'$  has the cholesterol as the side information.

Similarly, we quantify the privacy loss considering that cholesterol level is revealed for the adversary. Suppose that the prior probabilities of different cholesterol ranges are as follows:

$$(K^*, P_{K^*}) = \{(\text{Chol}_1, 0.8), (\text{Chol}_2, 0.06), (\text{Chol}_3, 0.06), (\text{Chol}_4, 0.06)\}.$$

The adversary's advantage who knows about the cholesterol level of a patient increases as shown in Figure 5.5. We observed that the advantage is increased more for  $\text{Chol}_1$ ,  $\text{Chol}_2$ , and  $\text{Chol}_3$  due to the low prior probability that they have. However, it is difficult for the adversary to remove the messages with the value of  $\text{Chol}_1$  as cholesterol level attribute.

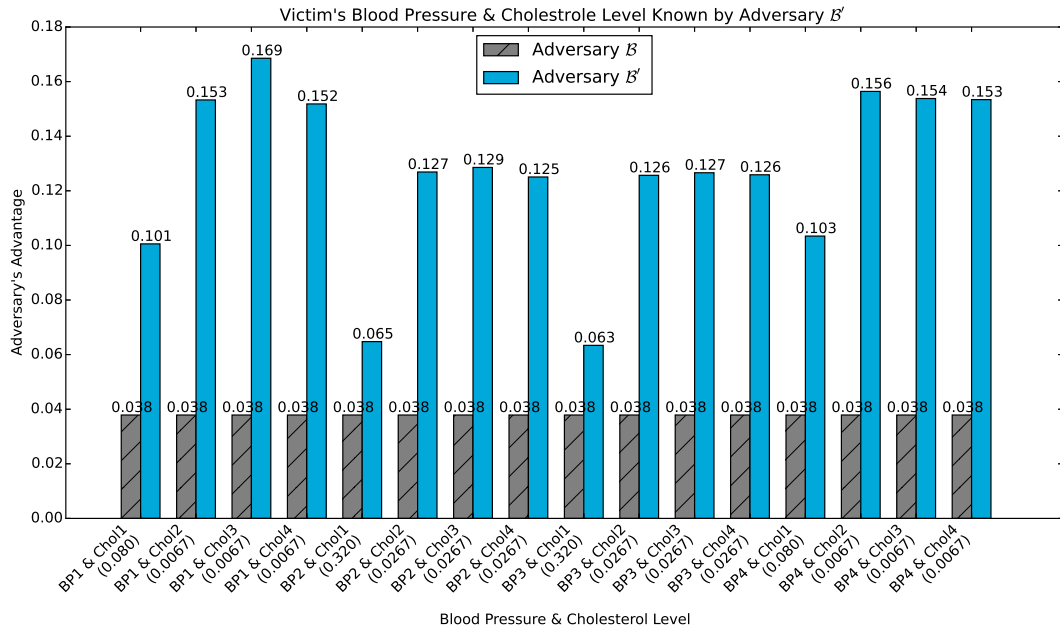


Figure 5.6: Evaluation of adversary's advantage with blood pressure and the cholesterol level as the side information.

Moreover, we also calculate the privacy leakage with the assumption that the adversary knows about both blood pressure and cholesterol level of a patient. The outcome is represented in Figure 5.6. For the patient's with blood pressure and cholesterol levels equal to  $\text{BP}_1$  and  $\text{Chol}_2$ , the risk of brute-force attacks increases up to 12%, that is the worst case among the other cases.

### 5.3.2 Drugs List

An attacker can be a curious pharmacist who knows some information about a patient’s drug usage. Suppose that the patient is diagnosed with breast cancer and a doctor prescribed her a list of drugs in which one of the drugs is disclosed to the adversary. Following the previous set-up, we quantize the privacy loss for this condition and represent the result in Figure 5.7.

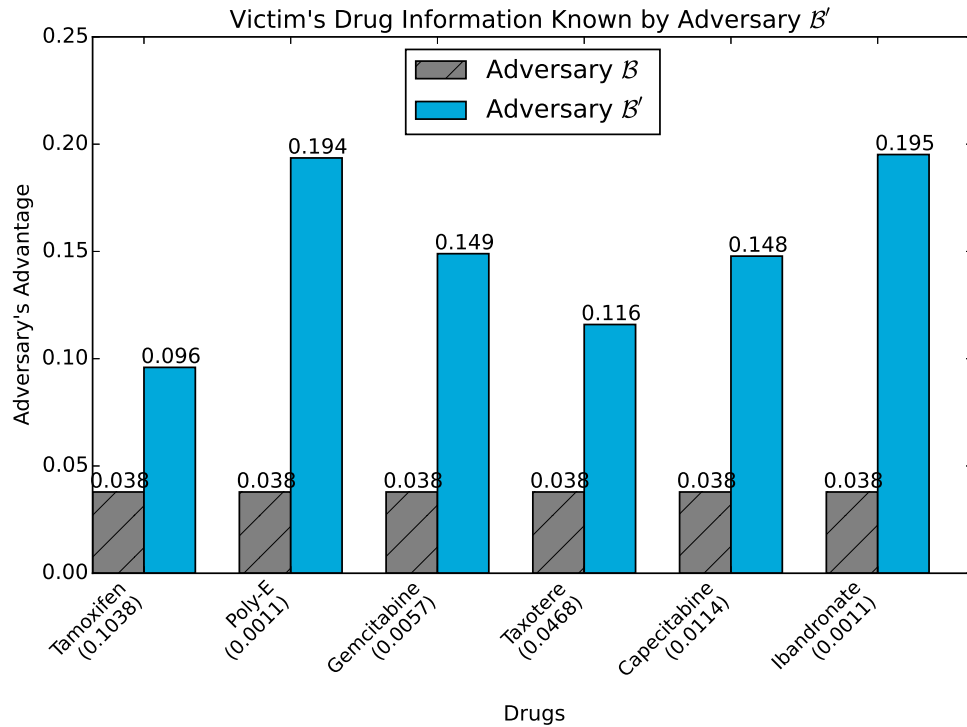


Figure 5.7: Evaluation of adversary’s advantage with drugs list as the side information. Adversary  $\mathcal{B}$  has no side information and his advantage is approximately  $w = 0.0379$  ( $w$  is the maximum weight of a password in the password distribution,  $p_k$ ) while adversary  $\mathcal{B}'$  has one of the drug’s name of the victim’s drugs usage as the side information.

As it is shown in the figure, as much as the prior probability decreases the advantage of the adversary with side information increases.



## 5.4 Discussion

Here in this section we discuss about the extensions and limitations of our proposed method.

### Dynamic Model

Health data is a highly dynamic dataset, hence storage and processing of this kind of data should be implemented in a flexible model. Our proposed model is a practical solution for the health records. We design *PHR Update* algorithm that is a solution to update the data whenever an attribute is changed in a patient health record. The time complexity of *PHR Update* algorithm is highly dependent on the layer which needs to be updated. As we get close to the root the number of nodes that should be computed is increased. As we present in Section 4.3, updating blood pressure is more time effective than cholesterol level since it is closer to the root. Hence, the organization of the levels is an important issue in terms of time complexity.

For instance assume the same example in Chapter 3 a tree with three levels for physiological variables: blood pressure, cholesterol level, and disease, respectively. Updating the blood pressure attribute takes 53 seconds, while after we change the tree organization and put the diseases in the first level before blood pressure and cholesterol, the updating process time of blood pressure dramatically decreases to 0.56 seconds, which is due to the high number of nodes at disease level. Hence, in order to make an efficient system we propose an organization in which the dynamic attributes are close to the leaves and the attributes that are change rarely near to the root.

Moreover, the algorithm makes it possible to modify the information and to add or remove new attributes to the DTE as well. It can be done simply by changing the interval sizes based on the new conditional probabilities and shift the seed.

## **Symptoms & Treatments**

As we introduced earlier, the proposed solution is a general model that consists all the health attributes and the corresponding values. However, due to the lack of existing datasets for the treatments and symptoms of diseases, we did not implement our model on real datasets for these attributes. Nevertheless, after investigating these attributes, we can easily extend our model of drugs DTE to treatments and symptoms DTE as well.

## **Legitimate User's Typo**

Honey encryption provides an incorrect but plausible-looking plain text as for each wrong passwords. This is a significant feature for security concerns. However, one of the limitations of this model for legitimate users rises when a user unintentionally tries a wrong password while entering the system. To address this issue, each user can choose a limited set of characters as a text which is not related to any of health attributes in his/her health record. This way, when a user enters the password a confirmation text will appear that the user should verify if it is his/her provided text. Otherwise, the user should reenter the password.

# Chapter 6

## Conclusion and Future Work

The security and privacy of health data has significant effect on the patients' life and the technology development of health record systems. The recent reports prove the fact that the causal security methods are not enough to protect sensitive information such as medical data. Hence, we come up with a new solution by benefiting from recent novel methods in order to construct a secure and private system for the storage and process for health records.

Brute-force attacks threaten the medical datasets and it is one of the corrupted attacks that reveal the health and medical information of a patient. Even though pseudo anonymity techniques are mostly applied to the dataset, given some background regarding a patient's attitude, an attacker can reveal a lot of information of medical history corresponding to the patient. Encouraging users to provide high-entropy passwords is one of the solutions to combat brute-force attacks. However, based on the studies users are not willing to provide complex passwords and they are using easy-to-remember passwords. To this end, we propose a system that does not rely on the password entropy, and even if an adversary tries possible passwords to break the system, the leaked data plausible-looking but incorrect message that does not reveal any information of a patient.

Our method provides a secure storage and processing for the personal health records regardless of the password entropy. Utilizing honey encryption approach, when an attacker tries different passwords to access the data, for each incorrect

password that he tries a valid-looking message resulted. Hence, he cannot significantly reduce his options in the password pool. Our security analysis proves that even side information does not cause a major security degradation in the system. Furthermore, honey encryption is generally applied on a static system where data does not change over time, nevertheless, we construct a system for dynamic datasets such as medical data. The data can be updated in an encrypted way without losing security and reconstructing the DTE.

Our proposed system is a good solution for secure storage and processing of the PHR data, however, the system can be improved in different aspects such as security and analyzing the health records.

The PHR Update algorithm is an efficient way to update the health attributes of a patient without revealing the data. We have involved the patient to this process in order to distribute the secret key and strengthen the security. Nevertheless, we are to investigate on developing a non-interactive protocol in order to eliminate the patient without degrading the security and privacy of the system.

Furthermore, during this study, we assume that the data is structured. Nonetheless, in real-world examples of health records, there are unstructured parts as well. For example, a health record might be a document that consists of some attributes related to the patient and a free text including the patient's treatments and symptoms. We investigate on extracting health attributes from a biomedical text by utilizing name-entity recognition (NER) techniques [60]. Hence, benefiting from this approach we can extend our framework to first extracting the attributes from the health document and then constructing the DTE for further processing.

Other work includes extending the study to other dynamic datasets such as location dataset in order to store and process the data through a private and secure protocol against brute-force attacks. Furthermore, improving the system in case of searching the PHR data in a secure and private protocol is also planned as the next step for this study.

# Bibliography

- [1] A. I. Yashin, E. Stallard, *et al.*, *Biodemography of Aging: Determinants of Healthy Life Span and Longevity*, vol. 40. Springer, 2016.
- [2] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands, “Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption,” *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 121–126, 2006.
- [3] “What is a personal health record?.” <https://www.healthit.gov/>. Accessed: 2017-09-05.
- [4] A. Act, “Health insurance portability and accountability act of 1996,” *Public law*, vol. 104, p. 191, 1996.
- [5] P. Chhanabhai and A. Holt, “Consumers are ready to accept the transition to online and electronic records if they can be assured of the security measures,” *Medscape General Medicine*, vol. 9, no. 1, p. 8, 2007.
- [6] R. Whiddett, I. Hunter, J. Engelbrecht, and J. Handy, “Patients’ attitudes towards sharing their health information,” *International journal of medical informatics*, vol. 75, no. 7, pp. 530–541, 2006.
- [7] “The financial impact of breached protected health information business case for enhanced phi security.” <http://webstore.ansi.org/phi/>. Accessed: 2017-06-29.
- [8] “Ibm’s plan to encrypt unthinkable amounts of sensitive data.” <https://www.wired.com/story/ibm-z-mainframe-encryption>. Accessed: 2017-08-8.

- [9] A. Shahani, “The Black Market For Stolen Health Care Data,” tech. rep., NPR, 02 2015.
- [10] “Questions loom for healthcare’s data security.” <http://www.nasdaq.com/article/questions-loom-for-healthcares-data-security-cm809113>. Accessed: 2017-06-29.
- [11] “Protecting information: The most effective practices to ensure health data privacy.” <https://datafloq.com/read/protecting-information-the-most-effective-practice/3458>. Accessed: 2017-08-10.
- [12] “Nhs cyber-attack causing disruption one week after breach.” [https://www.theguardian.com/society/2017/may/19/nhs-cyber-attack-ransomware-disruption-breach?utm\\_source=datafloq&utm\\_medium=ref&utm\\_campaign=datafloq](https://www.theguardian.com/society/2017/may/19/nhs-cyber-attack-ransomware-disruption-breach?utm_source=datafloq&utm_medium=ref&utm_campaign=datafloq). Accessed: 2017-08-14.
- [13] A. Browne, “Lives ruined as NHS leaks patients’ notes,” tech. rep., The Gaurdian, 06 2000.
- [14] “Electronic health records vs. patient privacy: Who will win?.” <http://www2.idexpertscorp.com/blog/single/electronic-health-records-vs.-patient-privacy-who-will-win>. Accessed: 2017-06-05.
- [15] J. Kirk, “Even Encrypted Medical Record Databases Leak Information,” tech. rep., IDG News Service, 09 2015.
- [16] E. Snell, “Top 4 Healthcare Data Breaches Stem from Hacking Incident,” tech. rep., Health IT Security, 10 2016.
- [17] D. Florencio and C. Herley, “A large-scale study of web password habits,” in *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, (New York, NY, USA), pp. 657–666, ACM, 2007.
- [18] J. Bonneau, “The science of guessing: analyzing an anonymized corpus of 70 million passwords,” in *Security and Privacy (SP), 2012 IEEE Symposium on*, pp. 538–552, IEEE, 2012.

- [19] H. Kim and J. H. Huh, “Pin selection policies: Are they really effective?,” *computers & security*, vol. 31, no. 4, pp. 484–496, 2012.
- [20] A. Juels and T. Ristenpart, “Honey encryption: Encryption beyond the brute-force barrier,” *IEEE Security & Privacy*, vol. 12, no. 4, pp. 59–62, 2014.
- [21] B. Kaliski, “Pkcs# 5: Password-based cryptography specification version 2.0,” 2000.
- [22] L. Spitzner, *Honeypots: tracking hackers*, vol. 1. Addison-Wesley Reading, 2003.
- [23] E. Bresson, D. Catalano, and D. Pointcheval, “A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications,” in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 37–54, Springer, 2003.
- [24] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, “Secure attribute-based systems,” *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.
- [25] C. Gentry, “Computing arbitrary functions of encrypted data,” *Communications of the ACM*, vol. 53, no. 3, pp. 97–105, 2010.
- [26] R. C. Barrows and P. D. Clayton, “Privacy, confidentiality, and electronic medical records,” *Journal of the American Medical Informatics Association*, vol. 3, no. 2, pp. 139–148, 1996.
- [27] T. Dinev, V. Albano, H. Xu, A. D’Atri, and P. Hart, “Individuals’ attitudes towards electronic health records: A privacy calculus perspective,” in *Advances in Healthcare Informatics and Analytics*, pp. 19–50, Springer, 2016.
- [28] J. L. Fernández-Alemán, I. C. Señor, P. Á. O. Lozoya, and A. Toval, “Security and privacy in electronic health records: A systematic literature review,” *Journal of biomedical informatics*, vol. 46, no. 3, pp. 541–562, 2013.
- [29] H. S. G. Pussewalage and V. A. Oleshchuk, “Privacy preserving mechanisms for enforcing security and privacy requirements in e-health solutions,” *International Journal of Information Management*, vol. 36, no. 6, pp. 1161–1173, 2016.

- [30] B. Yüksel, A. Küpçü, and Ö. Özkasap, “Research issues for privacy and security of electronic health services,” *Future Generation Computer Systems*, vol. 68, pp. 1–13, 2017.
- [31] A. Gkoulalas-Divanis, G. Loukides, and J. Sun, “Publishing data from electronic health records while preserving privacy: A survey of algorithms,” *Journal of biomedical informatics*, vol. 50, pp. 4–19, 2014.
- [32] T. Neubauer and J. Heurix, “A methodology for the pseudonymization of medical data,” *International journal of medical informatics*, vol. 80, no. 3, pp. 190–204, 2011.
- [33] C. Quantin, D.-O. Jaquet-Chiffelle, G. Coatrieux, E. Benzenine, and F.-A. Allaert, “Medical record search engines, using pseudonymised patient identity: an alternative to centralised medical records,” *international journal of medical informatics*, vol. 80, no. 2, pp. e6–e11, 2011.
- [34] L. Demuynck and B. De Decker, “Privacy-preserving electronic health records,” in *Communications and Multimedia Security*, vol. 3677, pp. 150–159, Springer, 2005.
- [35] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, “Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption,” *IEEE transactions on parallel and distributed systems*, vol. 24, no. 1, pp. 131–143, 2013.
- [36] S. Souza, R. Gonçalves, E. Leonova, R. Puttini, and A. Nascimento, “Privacy-ensuring electronic health records in the cloud,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 11, 2017.
- [37] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “Patient controlled encryption: ensuring privacy of electronic medical records,” in *Proceedings of the 2009 ACM workshop on Cloud computing security*, pp. 103–114, ACM, 2009.
- [38] W.-B. Lee and C.-D. Lee, “A cryptographic key management solution for hipaa privacy/security regulations,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 34–41, 2008.



- [39] S. Narayan, M. Gagné, and R. Safavi-Naini, “Privacy preserving ehr system using attribute-based infrastructure,” in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pp. 47–52, ACM, 2010.
- [40] 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 *HealthTech*, 2013.
- [41] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin, “A cryptographic approach to securely share and query genomic sequences,” *IEEE Transactions on information technology in biomedicine*, vol. 12, no. 5, pp. 606–617, 2008.
- [42] Q. Xue, M. C. Chuah, and Y. Chen, “Privacy preserving disease treatment & complication prediction system (pdtcp),” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 841–852, ACM, 2016.
- [43] J. Sun, X. Zhu, C. Zhang, and Y. Fang, “Hcupp: Cryptography based secure ehr system for patient privacy and emergency healthcare,” in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pp. 373–382, IEEE, 2011.
- [44] L. Spitzner, “Honeytokens: The other honeypot,” 2003.
- [45] A. Juels and R. L. Rivest, “Honeywords: Making password-cracking detectable,” in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 145–160, ACM, 2013.
- [46] B. N. Kausik, “Method and apparatus for cryptographically camouflaged cryptographic key storage, certification and use,” Jan. 2 2001. US Patent 6,170,058.
- [47] N. Tyagi, J. Wang, K. Wen, and D. Zuo, “Honey encryption applications,” *Network Security*, 2015.
- [48] Z. Huang, E. Ayday, J. Fellay, J.-P. Hubaux, and A. Juels, “Genoguard: Protecting genomic data against brute-force attacks,” in *2015 IEEE Symposium on Security and Privacy*, pp. 447–462, IEEE, 2015.
- [49] J. W. Yoon, H. Kim, H.-J. Jo, H. Lee, and K. Lee, “Visual honey encryption: Application to steganography,” in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pp. 65–74, ACM, 2015.

- [50] J.-I. Kim and J. W. Yoon, “Honey chatting: A novel instant messaging system robust to eavesdropping over communication,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 2184–2188, IEEE, 2016.
- [51] R. Chatterjee, J. Bonneau, A. Juels, and T. Ristenpart, “Cracking-resistant password vaults using natural language encoders,” in *Security and Privacy (SP), 2015 IEEE Symposium on*, pp. 481–498, IEEE, 2015.
- [52] M. Beunardeau, H. Ferradi, R. Géraud, and D. Naccache, “Honey encryption for language,” in *International Conference on Cryptology in Malaysia*, pp. 127–144, Springer, 2016.
- [53] E. Jervase, D. Barnabas, A. Emeka, and N. Osondu, “Sex differences and relationship between blood pressure and age among the ibos of nigeria,” *Internet J Biol Anthropol*, pp. 3–2, 2009.
- [54] A. I. Yashin, K. G. Arbeev, I. Akushevich, S. V. Ukraintseva, A. Kulminski, L. S. Arbeeva, and I. Culminskaya, “Exceptional survivors have lower age trajectories of blood glucose: lessons from longitudinal data,” *Biogerontology*, vol. 11, no. 3, pp. 257–265, 2010.
- [55] H. Krawczyk and P. Eronen, “Hmac-based extract-and-expand key derivation function (hkdf),” 2010.
- [56] D. Wang and P. Wang, “On the implications of zipf’s law in passwords,” in *European Symposium on Research in Computer Security*, pp. 111–131, Springer, 2016.
- [57] D. Malone and K. Maher, “Investigating the distribution of password choices,” in *Proceedings of the 21st international conference on World Wide Web*, pp. 301–310, ACM, 2012.
- [58] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, “Zipf’s law in passwords,” *IEEE Transactions on Information Forensics and Security*, 2017.
- [59] D. M. Powers, “Applications and explanations of zipf’s law,” in *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pp. 151–160, Association for Computational Linguistics, 1998.

- [60] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Linguisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

