AIMS | African Institute for Mathematical Sciences CAMEROON

Development of an encrypted patient database including a doctor user interface

Eric NIZEYIMANA (eric.nizeyimana@aims-cameroon.org)
African Institute for Mathematical Sciences (AIMS)
Cameroon

Supervised by: Prof. Dr. Axel Schumann
Technische Hochschule Millelhessen(THM), Germany

23 June 2015

Submitted in Partial Fulfillment of a Structured Masters Degree at AIMS-Cameroon

Abstract

Communication and storage of data is very essential to everyone all over the world, and some data to be shared and stored must be secured. In this masters thesis, we will develop the proof of concept of an encrypted patient database including a doctor user interface. It will help the doctor to insert a new patient, search informations related to a specific patient and send some information to the physician. It will use mathematical cryptography methods to generate the common key infrastructure which will be needed for the storage, retrieval and update of diverse patient data in the encrypted patient database. The common key infrastructure will be also useful in decrypting the snap-ins images that reside on physicians computer. Thus, data of a patient to be shared must be well encrypted and transmitted in a secure way.

Declaration

I, the undersigned, hereby declare that the work contained in this essay is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

ERIC NIZEYIMANA, 23 June 2015.

Contents

1. Introduction and literature review

1.1 Background on cryptography

Cryptography is the science of sending a message or information in a secret manner. The use of codes and ciphers to protect secrets began many years ago around 2000 BC by the old Kingdom of Egypt with non-standard hieroglyphs carved into monuments. In 500 BC to 600 BC Hebrew scholars made use of simple mono-alphabetic substitution ciphers. The ancient Greeks also adapted ciphers in the Spartan military. The Romans knew something of cryptography by the Caesar cipher and its variations.In this classical cryptography the method of encryption uses pen and paper or perhaps simple mechanical aids[9].

The development of cryptography has been paralleled by the development of cryptanalysis, the breaking of codes and ciphers. David Kahn notes in the "Code breakers" that modern cryptology originated among the Arabs, the first people to systematically document the methods of cryptanalysis. In 800 AD the frequency-analysis techniques for breaking mono-alphabetic substitution ciphers was invented by Arab mathematician Al-Kindi[9].

Even though cryptography has long and complex history, it wasn't until the 19^{th} century that it developed anything more than ad-hoc approaches for either encryption or cryptanalysis. In the early 20^{th} century, the invention of complex mechanical and electromechanical machines such as Enigma rotor machine provided more sophisticated and efficient means of encryption. In the World war I the Admiralty's room 40 broke Germany naval codes and played an important role in several naval engagements during war, notably in detecting major German sorties into the North Sea. In 1917, Gilbert Vernam proposed a teleprinter cipher which led to the development of electromechanical devices such as cipher machines, and to the only unbreakable cipher, the one time pad. Therefore, Germans lost the first World war due to cryptanalysis[9].

Later on, the Germans made heavy use, in several variants, of an electromechanical rotor machine known as Enigma. In 1932, a mathematician in Poland's cipher bureau (Marian Rejewski) deduced the detailed structure of German Army system Enigma, using mathematics and limited documentation supplied by captain Gustave Bertrand of the French military intelligence. This led to the defeat of German and Japanese in the second World war, due to the high cryptanalysis used by Britain (British typex) and US Navy (American SIGABA) by using Allied cipher machines[9].

The encryption in modern times is achieved by using algorithms that have a key for encryption and another key for decryption of the information. These keys convert the message and data into "digital gibberish" through encryption and then return them to the original form through decryption. The longer the key is, the more difficult it is to crack the code. It started by using symmetric key algorithms where only one cryptographic key is needed by both the sender and recipient, and they must keep it secret[9].

Modern ciphers like AES(Advanced Encryption Standard) and higher quality asymmetric ciphers are widely considered unbreakable. However, poor designs and implementations are still sometimes adopted and there have been important cryptanalytic breaks of deployed crypto-system in recent years. None of the mathematical ideas underlying public key cryptography has been proven to be unbreakable[9].

Nowadays, cryptography has gone from an art form that dealt with secret communication for military to a science that helps to secure systems for ordinary people all across the globe. This also means that cryptography is becoming a more and more central topic within computer science[17].

In this thesis, we will use asymmetric key to encrypt a database of patients. The doctor will have a user interface. In case the doctor needs to share data with the physician, a public key is needed and is known by everyone but a private key for each is needed in order to decrypt information. No need to secure the channel like in symmetric key.

1.2 Problem statement

Nowadays, everyone needs to have an electronic medical record at the hospital. Since these data can be sent from one doctor to another or from a physician to a doctor through the network or stored on hospital computers, they can be captured by unauthorized persons. Thus, patients fear for the security of their data because of the increasing vulnerability of that technology. The aim of this thesis is to ensure the security of patients' medical records by developing an encrypted patient database which cannot be easily broken by anyone.

1.3 Scope of project

This thesis starts in March,2015 up to May,2015. It will be oriented on a database of patients data in the medical field only. It will not deal with images or other things. Due to time constraints, we will implement the prototype of system and not a ready-to-use system.

1.4 Database Encryption concepts and terminology

Encryption: is the conversion of a plain-text or data into unintelligible form by means of reversible transformation. In other words, encryption is the process of taking some data (plain-text) and obfuscating it(using the key) so that it can not be read by others (those who do not have the correct key).

Decryption: is the translation of encrypted text or data called cipher into original text or data (plain-text).

Key: is a piece of information or parameter (string of bits) used by cryptography algorithms or ciphers to transform a plain-text into a cipher-text (encryption) and /or to transform a cipher-text in a plain-text (decryption).

Key infrastructure: is an entity in charge of the management of keys used for encryption and decryption.

Authentication and authorization: Is the way of checking if the user has rights to the system. Every time, the database should authenticate the user and then make sure that the user has permission to perform a particular action such as encryption or decryption of data and generate or use of a key[26].

1.5 Aim of project

1.5.1 Main objective.

- Development of an encrypted patient database including a doctor user interface.

1.5.2 Specific objectives.

- Description of mathematics used by cryptography Rivest-Shamir-Adleman (RSA) algorithm.
- Development of an experimental prototype of a full encrypted patient database using MYSQL with exemplary patient data.
- Development of an experimental prototype of a common doctor user interface for input and output of encrypted and decrypted data to and from the patient database.
- Development of a key infrastructure for storage, retrieval and update of diverse information in the encrypted patient database.
- Development of a common key infrastructure for the exchange of data between doctor user interface, patient database and medical image database together with the developer of the medical image database.

1.6 Literature review on existing systems

1.6.1 Overview of cryptography algorithms.

Since the beginning of modern cryptography, there are two types of encryption algorithms:

- **Symmetric algorithm:** this algorithm uses a secret key which is the same for both encryption and decryption. This key is randomly generated with n_bit string, is simple to generate and has no special properties.
- **Asymmetric algorithm:** this algorithm uses two keys, one (the public key) for encryption and the other (the private key) for decryption. The sender of the message uses the receiver's public key to encrypt, while the receiver uses his/her own private key to decrypt. These keys have a special structure, and are expensive to generate[36].

1.6.2 Public key cryptography.

The introduction of public-key encryption marked a revolution in the field of cryptography. Until that time, cryptographers had relied exclusively on shared, secret keys to achieve private communication [17]. It was a problem when you want to send a key to another person, because you can't be sure of the security of the transmission mode [21]. Therefore, Diffie-Hellman created a key exchange strategy for the private key.

Symmetric algorithms are composed by algorithms such as Data Encryption Standard (DES) and Advanced Encryption Standard (AES). They use small key size and the two parties should agree on the key before starting the communication.

Public-key techniques, in contrast, enable two parties to communicate privately without having agreed on any secret information in advance [17].

There are two classes of public key algorithms commonly encountered:

- Algorithms based on the difficulty of factoring.
- Algorithms based on the difficulty of discrete logarithm problem [1].

The RSA algorithm is based on the difficulty of factoring large numbers into two prime factors. RSA has been implemented in Hardware like Network cards and smart cards, and Software such as Microsoft product, Apple and Novel[2]. This shows how RSA has proven to be a more secure algorithm than others.

The RSA algorithm is an example of a "trapdoor function" where an operation is easy to do but hard to undo without additional information. RSA uses the fact that multiplying large numbers is much easier than factorizing them[28]. Therefore, the security of RSA is built on the complexity of factorizing these large numbers.

Elliptic curves cryptography(ECC) algorithm is based on the difficulty of discrete logarithmic problem. The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that $kP = Q$, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve[6].

RSA is simpler than Elliptic curve cryptography on the point of view of mathematics. Many engineers feel they understand RSA more than Elliptic curve. ECC has the characteristics of using short keys, low CPU consumption and low memory usage. Public key operations for RSA are faster than the ones for Elliptic Curve Cryptography[27].

1.6.3 Electronic Medical Records.

Nowadays, technology has shown a big impact in the development of the World. In the field of healthcare, many systems have been developed such as electronic medical records [32, 35], medical imaging systems [32, 30], homecare systems[8, 12], patient physiological signal recording systems[24, 25]. These systems transmit data through networks so that users can instantly and conveniently access them when necessary[32].

However, the most needed thing is to secure these data for external agents in order to maintain the privacy of patient's data. This is why we need to secure these data.

They are multiple parallel efforts to modernize medical record to ensure patients safety and patients privacy[7, 18, 20].

A secure Elliptic Curve Cryptography-based Electronic Medical Record System is a crypto-system built to allow the authorized personnel to access electronic medical records any time and anywhere with shorter keys and can be also used by portable devices. It employs a cloud database, an Elliptic curve cryptography integration unit, a smart card and portable devices to provide users with a secure environment for Electronic Medical Records transmission[32].

A centralized database stores a medical record for each individual to improve the quality and security of ongoing and later treatment[14]. Since all data are saved in the same central database, the problem will be on images which will need high amount of storage and take a lot of time in encryption and decryption.

Authors in [5] designed a self-protecting system to Secure Electronic Medical Records using Attribute-Based Encryption on Mobile devices. This system allows healthcare organizations to export Electronic

Medical Records to locations outside of their boundary.

1.7 Summary

In this chapter, we have introduced cryptography. We have presented some terminology of cryptography. We have presented the comparison of algorithms used in cryptography. In this thesis, we are going to use asymmetric algorithms since they have a stronger security in the process of sharing data through any network than symmetric algorithms. In the next chapter, we will see more deeply RSA algorithm due to its security which has been proven by mathematical ideas to be unbreakable. This algorithm will be the one that we will use for both encryption and decryption to secure patients database.

2. Fundamental mathematical tools for cryptography

2.1 Introduction

In this chapter, we will present the mathematical aspect of cryptography that will be used to encrypt and decrypt patient database and snap-ins images from the console of the physician. We will start by prime numbers which have an inner impact in cryptography, then we follow with primality test and modular arithmetic. We will also present RSA encryption and decryption algorithms. Finally, we will introduce the system tools that will be needed in the implementation of our system. All algorithms except RSA algorithm used in this chapter are referenced in [29].

2.2 Prime numbers

Since the start of modern cryptography, prime numbers have a big role in the creation of keys in public key crypto-systems in general, especially in RSA based crypto-systems.

2.2.1 Definition. [15] A positive integer p is prime if it has exactly two positive divisors which are 1 and p.

Conventionally, the number 1 is not prime since it doesn't have two distinct positive divisors[15].

A number different from 1 which is not prime is called *composite* [29].

By the fundamental theorem of Arithmetic, every positive integer number N greater than 1 can be represented in a unique way as the product of prime numbers.

$$N = \Pi_i^k P_i^{e_i}$$

where for all $i \in \{1, 2, , \cdots, k\}$, $\quad P_i$ are distinct primes and $e_i \geq 1$.

"There are two facts about the distribution of prime numbers among integers. The first is that they are the most arbitrary and ornery objects studied by mathematicians: they grow like a weeds among natural numbers, seeming to obey no other law than that of chance, and nobody can predict where the next one will sprout. The second fact is even more astonishing, for it states just the opposite: that the prime numbers exhibit stunning regularity, that there are laws governing their behaviour, and that they obey these laws with almost military precision" [29].

These numbers are distributed randomly in integers without any formula to prove the interval between two of them[29].

Divisibility: An integer n is a divisor of m written as $n|m$ if m can divide n without leaving a remainder [10].

2.2.2 Lemma. [10] If $n|m$ and $m|l$ then $n|l$.

Proof. [10] If $n|m$ that means that there exists an integer s such that $m = ns$, and if $m|l$ also there exist integer t such that $l = mt$, this implies that $l = nst = n(st)$; therefore n is a divisor of l. □

2.2.3 Lemma. [10] Let n be a positive integer greater than 1. Let d be the smallest divisor of n that is greater than 1. Then d is prime.

Proof. [10] We are going first to check if d is well defined. And to do this, let's assume that n has no smallest divisor, this means that the lemma has no sense. Since we already know that d is a divisor of n greater than 1, so 1 is another divisor of n.

Now, let's assume that d is composite, that means , there exists another integer e such that $1 < e < d$ where e is a divisor of d. Since d divides n $(d|n)$ that implies that e also divides n. And this is a contradiction since d is the smallest divisor of n. Thus, d must be prime. □

2.2.4 Theorem. *[10] There are infinite numbers of primes.*

Proof. [10] Let's consider k finite numbers of primes $p_1, p_2, \cdots, p_k$. Let's define

$$n = \Pi_{i=1}^{k} p_i + 1.$$

Let's consider $d > 1$ as a smallest divisor of n. So d is prime and d divides n. By the way none of the prime in our k list primes is a divisor of n. All are divisors of $n-1$, and if you divide n by one of p_i's in the list, you are left with 1 every time. So, d is prime but it is not in the list. Which is contradiction, therefore prime numbers are infinite. □

2.2.5 Theorem (Division theorem)**.** *[15] Let a and b be positive integers with $a > 0$. Then there are integers q , r with $0 \leq r < a$ such that*

$$b = aq + r.$$

Where q is quotient and r is remainder of division of b by a.

Proof. [15] If $a > b$, then $q = 0$ and $r = b$. This leads us to suppose that $a \leq b$. By considering the set A of all non-negative differences between b and integer multiples of a:

$$A = \{b - ak : b - ak \geq 0, \text{ k is a positive integer}\}.$$

This set A is non-empty since it contains $b = b - a \cdot 0$, A contains a least element $r = b - aq$. If r is not strictly smaller than a, then we would have $r - a \geq 0$, and therefore

$$r - a = (b - aq) - a = b - a(q + 1).$$

So $r - a$ would be a member of A strictly smaller than r, contradicting the minimality of r . Hence r does satisfy $0 \leq r < a$. □

This proof shows that every positive integer can be written as a linear combinations of the divisor, the quotient and the remainder.

2.2.6 Definition (Greatest Common Divisor)**.** [22, 29, 15] Let $a_1, \cdots, a_r$ be positive integers, then their greatest common divisor gcd, is the positive integer d with the property that $d|a_i$ and whenever c is an integer with $c|a_i$ for each i, we have $c|d$.

2.2.7 Theorem (GCD)**.** *[22] If $A = \{a_1, a_2, \cdots, a_n\}$ is any finite subset of Euclidean domain D. Then A has a gcd, which can be expressed as a linear combination $\Sigma\lambda_k a_k$ of the a_k's.*

Proof. [22] Let $d \neq 0$ and $d|a_i$, for $i = 1, \cdots, n$. Then $a_i = q_i d + r_i$ by theorem 2.2.5. That implies $r_i = a_i - q_i d$, and if $r_i = 0$ that is $a_i = q_i d$. Thus d is a common divisor of a_i.

Now, if e is any other common divisor of a_i's, that is $a_i = q_i e$, since we know that d is linear combination of the a_i's. Therefore, there exists some coefficients λ_i for $i = 1, \cdots, n$ such that

$$d = \Sigma \lambda_i a_i = e \Sigma_{i=1}^{n} \lambda_i q_i$$

which is a multiple of e. Thus d claims a gcd for $a_1, a_2, \cdots, a_n$. □

2.2.8 GCD Algorithm. This algorithm computes the $gcd(a, b)$, where a and b are two integers.

1. Assume $a \geq b \geq 0$. We have $gcd(a, b) = gcd(|a|, |b|) = gcd(|b|, |a|)$. Since we have assumed that $a, b \geq 0$, then we can replace absolute values by a and b. If $a = b$, then we output a and terminate. If $b = 0$, we output a.
2. Quotient and remainder. By using theorem 2.2.5, we write $a = bq + r$ with $0 \leq r < b$ and $q \in \mathbb{Z}$.
3. Finished? If $r = 0$, then $b|a$, therefore, we output b and terminate.
4. Shift and repeat. Set $a \leftarrow b$ and $b \leftarrow r$, and go back to step 2.

2.2.9 Example.

Set $a = 30$ and $b = 4$ and go step by step.

$$\begin{aligned} 30 &= 4 \cdot 7 + 2 \qquad gcd(30, 4) = gcd(4, 2) \\ 4 &= 2 \cdot 2 + 0 \qquad gcd(4, 2) = gcd(2, 0) = 2. \end{aligned}$$

2.2.10 Theorem (Euclidian)**.** *[15] Let a and b be positive integers. If a divides b, then a is the greatest common divisor otherwise applying theorem 2.2.5 repeatedly, define a sequence of positive integers $r_1, r_2, \cdots, r_n$, by*

$$\begin{aligned} b &= aq_1 + r_1 \quad (0 \leq r_1 < a) \\ a &= r_1 q_2 + r_2 \quad 0 \leq r_2 < r_1 \\ r_1 &= r_2 q_3 + r_3 \quad 0 \leq r_3 < r_2 \\ &\cdot \\ &\cdot \\ &\cdot \\ r_{n-2} &= r_{n-1} q_n + r_n \quad (0 \leq r_n < r_{n-1}) \\ r_{n-1} &= r_n q_{n+1}. \end{aligned}$$

Then r_n is the greatest common divisor of a and b.

2.2.11 Remark:.

- The extended Euclidean algorithm helps us to find integers a, b such that $gcd(m, n) = am + bn$ [22, 15, 10], for any m, n. Its implementation is the backward recurrence, which is suitable for hand computations.
- Let a, b be integers. If $d = gcd(a, b)$ then d divides all linear combination of a and b.

2.2.12 Extended Euclidean Algorithm.

Let's consider two integers a and b and consider that $a > b \geq 0$. And also let's consider $k = gcd(a, b)$. This algorithm computes $k, \quad x, \quad y$ such that $ax + by = k$.

1. Initialize. Let's set $x = 1, \quad y = 0, \quad r = 0, \quad s = 1$
2. Finished? If $b = 0$, set $k = a$ and terminate.
3. Quotient and remainder. Use theorem 2.2.5 to write $a = bq + c$ with $0 \leq c < b$
4. Shift and repeat. Set $(a, \quad b, \quad r, \quad s, \quad x, \quad y) = (b, \quad c, \quad x - qr, \quad y - qs, \quad x, \quad y)$ and go back to step 2.

We say that two positive integers a and b are relatively prime (coprime) [22] if their greatest common divisor is 1, that is, $gcd(a, b) = 1$.

2.3 Modular arithmetic

Let's consider an integer n greater than 1, and let a, b be integers, we say that a is congruent to $b(mod \quad n)$ if a and b have the same remainder by the division of n. That means, if and only if n divides $a - b$. [15] We write it as

$$a \equiv b(mod \quad n).$$

Let's fix an integer n greater than 1 and let a be any integer. The congruence class of a modulo n is the set of all integers which are congruent to a modulo n [15].

$$[a]_n = \{b : b \equiv a(mod \quad n)\}.$$

We consider the set of all congruence classes modulo n, that is referred to as the set of all integers modulo n and is denoted $\mathbb{Z}_n$ or $\mathbb{Z}/n\mathbb{Z}$.

2.3.1 Example.

$\mathbb{Z}_4 = \{[0]_4, [1]_4, [2]_4, [3]_4\}$.

Let's fix an integer n greater than 1, and let a be any integer. We say that $[a]_n$ is invertible if there is any integer b such that $[a]_n[b]_n = [1]_n$. And this happens if and only if $gcd(a, n) = 1$. [15]

2.3.2 Theorem (Solving linear congruences)**.** *[15] The linear congruence*

$$ax \equiv b(mod \quad n) \tag{2.3.1}$$

has a solution if and only if the greatest common divisor d of a and n divides b. If d does divide b there are d solutions up to congruence modulo n, and these solutions are congruent modulo n/d.

Proof. [15] The idea of proof comes from [15]. We are going to show the first part of our theorem which states that the linear congruence has a solution if and only if the greatest common divisor d of a and n divides b.

Let's consider c as solution of equation 2.3.1. Then since $ac \equiv b(mod \quad n)$, we have n divides $ac - b$.

So, there exists integer k, such that $nk = ac - b$, and this means

$$b = ac - nk. \tag{2.3.2}$$

Since d is the gcd of a and n, using remark 2.2.11, there are integers k and t such that:

$$d = ak + nt \tag{2.3.3}$$

By multiplying 2.3.3 by e we get

$$ed = eak + ent,$$

that means

$$b = a(ek) + n(et)$$

This gives,

$$a(ke) \equiv b(mod \quad n).$$

So, the congruence has a solution, ke, as required. Therefore, our first assertion of the theorem has been proven.

We are going to show the second part of our theorem 2.3.2. Consider now that c is a solution of $ax \equiv d(mod \quad n)$. So, before we have $ac = b + nk$ for some integer k by the above, d divides b, and therefore, we can divide this equation by d to get the equation in integers.

$$(a/d)c = b/d + (n/d)k.$$

Thus,

$$(a/d)c \equiv b/d(mod \quad n/d).$$

That is, every solution of the original congruence is also a solution of the congruence

$$(a/d)x \equiv b/d(mod \quad n/d).$$

In the contrary, by reversing steps we see that every solution to this second congruence is also a solution to the original one. So, the solution is really a congruence class modulo n/d. Such a congruence class splits into d distinct congruence classes modulo n. Namely if c is a solution then congruence classes of $c, c + (n/d), c + 2(n/d), c + 3(n/d), ..., c + (d-1)(n/d)$ are distinct solutions modulo n, and are all the solutions modulo n.

□

2.3.3 Inverse modulo n Algorithm.

Let's consider two integers a and n and consider $gcd(a, n) = 1$. This algorithm finds x such that $ax \equiv 1(mod \quad n)$.

1. Compute extended gcd. Use extended Euclidean algorithm to compute integers x, y such that $ax + ny = gcd(a, n) = 1$
2. Finished.Output x

2.3.4 Example. Solve $13x \equiv 1(mod 29)$.

We use extended Euclidean algorithm to find x, y such that $13x + 29y = 1$.

$$29 = 2 \cdot 13 + 3 \quad \Rightarrow 3 = 29 - 2 \cdot 13$$
$$13 = 4 \cdot 3 + 1 \quad \Rightarrow 1 = 13 - 4 \cdot 3 = 1 \cdot 13 - 4(29 - 2 \cdot 13) = 1 \cdot 13 - 4 \cdot 29 + 8 \cdot 13 = 9 \cdot 13 - 4 \cdot 29$$

Since $a = 13$ and $n = 29$; therefore, $x = 9$

2.3.5 Lemma. [19] Let a, b be any two integers and let $d = gcd(a, b)$. The set $a\mathbb{Z} + b\mathbb{Z} = \{ar + bs : r, s \in \mathbb{Z}\}$ is equal to $d\mathbb{Z}$ where $d = gcd(a, b)$.

Proof. [19] The idea of the proof comes from [19]. We are going to show that $a\mathbb{Z} + b\mathbb{Z} = d\mathbb{Z}$. Let $c \in a\mathbb{Z} + b\mathbb{Z}$, we want to show that $c \in d\mathbb{Z}$ that means there is an integer k such that $c = dk$.

Also, $c \in a\mathbb{Z} + b\mathbb{Z}$ implies that there are some integers r and s such that

$$c = ar + bs.$$

Since $d = gcd(a, b)$, then d divides $ar + bs = c$. Therefore, there is some integer $k \in \mathbb{Z}$ such that $c = dk$. Thus,

$$a\mathbb{Z} + b\mathbb{Z} \subseteq d\mathbb{Z}.$$

Let $e \in d\mathbb{Z}$ implies $e = dt$ where t is an integer.

Since $d = gcd(a, b)$ there are $m, n \in \mathbb{Z}$ such that

$$d = am + bn.$$

That means

$$dt = a(tm) + b(tn).$$

Therefore,

$$e = a(tm) + b(tn).$$

Hence, $e \in a\mathbb{Z} + b\mathbb{Z}$.

By conclusion, $a\mathbb{Z} + b\mathbb{Z} = d\mathbb{Z}$. □

2.3.6 Lemma. [19] If $gcd(a, n) = 1$ then, there is an integer a^{-1} such that $a \cdot a^{-1} \equiv 1(mod \quad n)$.

Proof. [19] By using lemma 2.3.5, the set $a\mathbb{Z} + n\mathbb{Z}$ is equal to $\mathbb{Z}$, the set of all integers. In other words, this means that there are integers r, s such that $ar + ns = 1$. Since we are working in modulo n, this becomes $a \cdot r \equiv 1(mod \quad n)$. □

2.3.7 Theorem (Chinese remainder theorem[19, 15, 29]). *Let $n_1, n_2, \cdots, n_k$ be numbers, no two of them which have a common factor. For any number $a_1, a_2, \cdots, a_k$, the system of congruences*

$$\begin{aligned} x &\equiv a_1 (mod \quad n_1) \\ x &\equiv a_2 (mod \quad n_2) \\ &\cdot \\ &\cdot \\ &\cdot \\ x &\equiv a_k (mod \quad n_k) \end{aligned}$$

has a solution. Any two solutions x_1, x_2 *are congruent mod* $n_1 n_2 \cdots n_k$.

Proof. [19, 15] Let's denote m_i the product of all elements of the set $\{n_1, n_2, \cdots n_k\}$ except n_i. Let's consider that $gcd(m_i, n_i) = 1$. Then lemma 2.3.6 implies that there is a number r_i such that $m_i r_i \equiv 1 (mod \quad n_i)$. Now, let $x \equiv \Sigma_{i=1}^{k} a_i m_i r_i$ let's check if x satisfies the given system of congruences. If x_1, x_2 both satisfy the given system of congruences, therefore $x_1 - x_2$ is divisible by each of $n_1, n_2, \cdots, n_k$. Since, these numbers have no common factors, we may conclude that $x_1 - x_2$ is divisible by $n_1 n_2 \cdots n_k$. □

2.3.8 Definition (Euler function)**.** [29]

Euler's $\varphi-$function: for $n \in \mathbb{N}$, let $\varphi(n) = \#\{a \in \mathbb{N} : a \leq n \quad \text{and} \quad gcd(a, n) = 1\}$, if p is a prime number then,

$$\varphi(p) = p - 1 = \#\{1, 2, 3, ..., p - 1\}.$$
$$\varphi(p^n) = p^n - p^{n-1}$$

2.3.9 Example.

$\varphi(5) = \#\{1, 2, 3, 4\} = 4$

$\varphi(20) = \varphi(5)\varphi(2^2) = (5 - 1)(2^2 - 2) = 8$

2.3.10 Theorem. *[29] If* $gcd(x, n) = 1$ *then* $x^{\varphi(n)} \equiv 1 (mod \quad n)$.

Proof. [29] Let $K = \{a : 1 \leq a < n \text{ and } gcd(a, n) = 1\}$. Since the reductions modulo of elements of $xK = \{xa : 1 \leq a < n \text{ and } gcd(a, n) = 1\}$ are the same as the reductions of elements of K, thus, $\Pi_{a \in K}(xa) \equiv \Pi_{a \in K} a (mod \quad n)$. Since the products are over the same numbers modulo n, now we can cancel the a' s on both sides to get $x^{\#K} \equiv 1 (mod n)$. □

2.3.11 Huge power algorithm. Let m be a non-negative integer. This algorithm writes m in binary, so, it finds $\varepsilon_i \in \{0, 1\}$ such that $m = \Sigma_{i=0}^{r} \varepsilon_i 2^i$

1. Initialize. Set $i = 0$
2. Finished? If $m = 0$, terminate.
3. Digit. If m is odd, set $\varepsilon_i = 1$, otherwise $\varepsilon_i = 0$, increment i
4. Divide by 2. Set $m = \lfloor \frac{m}{2} \rfloor$, the greatest integer $\leq m/2$. Go to step 2.

2.3.12 Compute power algorithm. Let a and n be integers and m a non negative integer. This algorithm computes a^m modulo n.

1. Write in binary. Write m in binary by using algorithm 2.3.11, so $a^m = \Pi_{\varepsilon_i} a^{2^i} (mod \quad n)$
2. Compute powers. Compute $a, a^2, a^{2^2} = (a^2)^2, a^{2^3} = (a^{2^2})^2, \cdots, a^{2^r}$, where $r+1$ is the number of binary digits of m.
3. Multiply powers. Multiply together the a^{2^i} such that $\varepsilon_i = 1$, always working modulo n

2.3.13 Example. We can compute the last 2 digits of 7^{91}, by finding $7^{91}(mod 100)$.

Since the $gcd(7, 100) = 1$, we have the Euler's theorem 2.3.10 that is $7^{\varphi(100)} \equiv 1(mod 100)$.

Since φ is multiplicative, then

$$\varphi(100) = \varphi(2^2.5^2) = (2^2 - 2)(5^2 - 5) = 40.$$

Thus,

$$7^{40} \equiv 1(mod 100),$$

hence

$$7^{91} \equiv 7^{40+40+11} \equiv 7^{11}(mod 100).$$

We now compute $7^{11}(mod 100)$ using algorithm 2.3.11. First, we write 11 in binary by repeatedly dividing by 2:

$$\begin{aligned} 11 &= 5 \cdot 2 + 1 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 1 \cdot 2 + 0 \\ 1 &= 0 \cdot 2 + 1. \end{aligned}$$

So, in binary, $(11)_2 = 1011$. Next we compute a, a^2, a^4, a^8 and output $a^8.a^2.a$ we have

$$a = 7, a^2 = 49, a^4 = 49^2 \equiv 1, a^8 = 1^2 = 1$$

$$7^{91} \equiv 7^{11} \equiv a^8.a^2.a \equiv 1.49.7 \equiv 43(mod 100).$$

2.4 Primality testing

It turns out to be remarkably easy to test whether a number is prime or not. At least, it seems easy to factor a number and search its prime divisors. There are many tests to prove that a number is prime but all are probabilistic [10]. They are not perfect. There is a certain probability they give the incorrect

answer. But by running several times the same test, we can reduce the probability of getting a wrong answer to an acceptable level.

In this thesis, we are using the **Rabin-Miller** primality test. Its purpose is to verify whether an odd integer number n is prime.

We choose a random value a less than n, called the basis. We check the property of $(a(mod \quad n))$ that always holds when n is prime. There are 25% chances that n is not prime. But by repeating this test for different random values of basis a, we enlarge our confidence in the final result [10]. We will use five iterations to check if a number is prime.

2.4.1 Lemma. [19] If p is prime, then every pair of integers a, b satisfies

$$(a+b)^p \equiv a^p + b^p (mod \quad p).$$

Proof. [19] By using binomial theorem

$$(a+b)^p = \Sigma_{k=0}^{p} \binom{p}{k} a^k b^{p-k}.$$

Every term in this sum is divisible by p except the $k = 0$ and $k = p$ terms.

□

2.4.2 Theorem (Fermat's little). *[19, 29, 10, 13] The number n is prime if and only if the congruence*

$$x^{n-1} \equiv 1 (mod \quad n)$$

is satisfied for every integer x between 0 and n.

Proof. [13, 19] The ideas of proof come from [13, 19].

We assume that $x^{n-1} \equiv 1(mod \quad n)$ for all x between 0 and n, and we show that n is prime.

By contradiction, let's assume that n is not prime.

- If $n = 1$, this is the contradiction because x^{-1} is not congruent to $1(mod \quad n)$.
- If $n \neq 1$. Since $n \neq 1$ and n is not prime there is an integer $n > d > 1$ such that d divides n. Since the number d^{n-1} is divisible by d, then d^{n-1} is not congruent to $1(mod \quad n)$. So, n is prime.

  Conversely, let's assume that n is prime and show that $x^{n-1} \equiv 1(mod \quad n)$. Let A be a set of all integers x which satisfy $x^n \equiv x(mod \quad n)$. This set has as element $x = 1$, and it is closed under addition and subtraction by lemma 2.4.1. Thus, every integer x belongs to A.

Then, let's consider x as any integer not divisible by n. The fact that $x \in A$ means that $n|x^n - x = x(x^{n-1} - 1)$. Since n is prime and x is divisible by n, this means $n|x^{n-1} - 1$, that is $x^{n-1} \equiv 1(mod \quad n)$. □

We use theorem 2.4.2 to detect that a given $p \in \mathbb{N}$ is not prime.

$$a^p \equiv a(mod \quad p).$$

2.4.3 Definition (Pseudo-primes [13]). The number $a \in \mathbb{N}$ is called witness for non-primality of $p \in \mathbb{P}$, if $a^p \not\equiv a(mod \quad p)$ is valid.

The number $a \in \mathbb{N}$ is called prime to basis a if $a^p \equiv a(mod \quad p)$.

Proof. If p is prime, then nothing to prove on the theorem. Otherwise, if p is composite, there exist a divisor a of p with $2 \leq a < p$. And if $a^{p-1} \equiv 1(mod \quad p)$, then $p|a^{p-1}-1$. Since $a|p$, we have $a|a^{p-1}-1$, hence there exists an integer k such that $ak = a^{p-1} - 1$. By subtraction we have $a^{p-1} - ak = 1$. So, $a(a^{p-2} - k) = 1$, this implies that $a|1$, which is a contradiction since $a \geq 2$. □

2.4.4 Theorem (Carmichael numbers). *[13]*

A number $p \geq 4$ is a Carmichael number if and only if

- $p = p_1...p_n$
- $p_k - 1|p - 1$ *for all* $k = 1, 2, ...n$.

They have the property that $a^{n-1} \equiv 1(mod \quad n)$ for all a with $gcd(a, n) = 1$

These Carmichael numbers are composite but they pass the Fermat test for all basis. This is why we are going to use Rabin-Miller test for primality.

2.4.5 Definition (Rabin - Miller test). [13, 29]

An odd number $p \in \mathbb{N}$ with representation $p - 1 = 2^t u$ where u is odd, is called strong pseudo-prime to basis a, if $gcd(a, p) = 1$, $a^u \equiv 1(mod \quad p)$ or there is an $s \in \{0, 1, ..., t-1\}$ for which $a^{2^s n} \equiv 1(mod \quad p)$.

This test proves that either a number n is not prime or it is probably prime.

Once p is not strict pseudo-prime to basis a, a is called a witness for the non-primality of p, which means that p is not prime.

2.4.6 Rabin - Miller algorithm. Given an integer $n \geq 5$ this algorithm outputs either true or false. If it outputs true, then n is probably prime, and if it outputs false, then n is definitely composite.

1. Split off power of 2. Compute the unique integers m and k such that m is odd and $n - 1 = 2^k m$.
2. Random base. Choose a random integer a with $1 < a < n$.
3. Odd power. Set $b \equiv a^m(mod \quad n)$. If $b \equiv \pm 1(mod \quad n)$ output true and terminate.
4. Even powers. If $b^{2r} \equiv -1(mod \quad n)$ for any r with $1 \leq r \leq k - 1$, output true and terminate. Otherwise, output false.

Proof. [29]

We are going to prove that if the algorithm outputs false, an integer n is really composite. Let's assume that

$$a^m \not\equiv \pm 1(mod \quad n),$$

and for all r with $1 \leq r \leq k-1$ we have

$$a^{2^r m} \not\equiv -1 (mod \quad n).$$

Since n is prime and $2^{k-1}m = (n-1)/2$, definition 2.4.3 implies that

$$a^{2^{k-1}m} \equiv \pm 1 (mod \quad n),$$

by hypothesis

$$a^{2^{k-1}m} \equiv 1 (mod \quad n).$$

This implies that

$$(a^{2^{k-1}m})^2 \equiv 1 (mod \quad n)$$

and we have also

$$a^{2^{k-2}m} \equiv \pm 1 (mod \quad n).$$

Again by hypothesis, this implies

$$a^{2^{k-2}m} \equiv 1 (mod \quad n).$$

Repeating this argument inductively, we see that

$$a^m \equiv \pm 1 (mod \quad n),$$

which is contradicting our hypothesis on a. □

The test is too fast. Unfortunately, a number which passes the test is not implied to be prime directly. Monier (1980) and Rabin (1980) have shown that, a composite number can pass the test for at most $1/4$ of the possible bases a. If N multiple independent tests are performed on a composite number, then the probability that it passes each test is $1/4^N$ or less [34]. As the probability becomes small, we get confidence to our solution of primality.

2.5 RSA crypto-system

RSA encryption has proven its potential. It is one of the best algorithms used in modern cryptography. Its strength relies on the fact that it is difficult to decrypt the message in case you are not the recipient[16]. The user of RSA creates a pair of keys (a public key and a private key) and then publishes a public key based on two large prime numbers along with an auxiliary value.

If the public key is not big enough, only someone with knowledge of the prime numbers can feasibly decode the message. Up to now, a key of 768-bits has been broken, so it is better to use keys which are longer than this [16].

RSA algorithm involves three steps: key generation, encryption and decryption.

2.5.1 Key generation. [16]

RSA involves a public key which is known by everyone and used for encrypting message while a private key is used to decrypt.

We need to choose two distinct prime numbers p and q randomly with similar bit-length, then we verify if they are prime by primality test 2.4.5. Once these numbers are prime, we compute $n = pq$, here n is used as modulo for both the public and private keys. Its length is the key length[16].

We compute

$$\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1),$$

where φ is Euler's totient function. Then, we choose an integer e such that $1 < e < \varphi(n)$ and $gcd(e, \varphi(n)) = 1$. That means e and $\varphi(n)$ are coprime. We call e the public key exponent. A short bit-length and small hamming weight of e result in a more efficient encryption. The public key is the pair of integers (n, e)[16].

We determine $x = d$ by solving the equation below,

$$x \equiv e^{-1}(mod \quad \varphi(n)).$$

This is the same as solving

$$ed \equiv 1(mod \quad \varphi(n)),$$

by using extended Euclidean algorithm. Here, d is kept as the private key exponent and must be secret [16].

2.5.2 Encryption. [16]

The sender of the message M needs to transmit the public key (n, e) to the receiver and keeps the private key d secret. We first decompose the message M into blocks if M is too big and we encrypt each block. Then, computer converts blocks of message into integers m, such that $0 \leq m < n$ by American Standard Code for Information Interchange (ASCII).

Then, we get cipher-text which is the original message already encrypted.

$$c \equiv m^e(mod \quad n).$$

Then we shall delete p, q, φ[16].

2.5.3 Decryption. [16]

The cipher-text c will be transferred to the receiver. Since it was already encrypted the receiver needs to decrypt it in order to get the original message m.

The receiver recovers the message m from c by using the private key exponent d via computing [16],

$$m \equiv c^d(mod \quad n).$$

Then, when you get a message m which is an integer, computer uses again the ASCII to get the text message. Finally combine all blocks to get the whole message.

2.5.4 RSA algorithm. This algorithm shows the encryption and decryption of the message.

1. Find two primes p, q randomly. The primes must be big enough at least 769 digits.

2. Compute $n = pq$.
3. Find $\varphi(n) = (p-1)(q-1)$.
4. Find e such that $gcd(e, \varphi(n)) = 1$.
5. (e, n) is public key.
6. $d \equiv e^{-1}(mod \quad \varphi(n))$, that is $d \cdot e \equiv 1(mod \varphi(n))$.

   Remark: d is the secret key, must be kept well. A cipher-text is only secured if the public key can be revealed while keeping the message secret[31].
7. $p, q, \varphi(n)$ shall be deleted.
8. For a message N, we encrypt it like this: $C \equiv N^e(mod \quad n)$. where C is called cipher text. If N is long, we make blocks with length of m such that $0 \leq m < n$. Then we encrypt m.
9. For decrypting message $N \equiv c^d(mod \quad n)$.

2.5.5 Theorem. *[33]* $d(e(m)) \equiv m(mod \quad n)$ *whenever* $gcd(m, n) = 1$.

Proof. [33] Let's recall that

$$e \equiv d^{-1}(mod \quad \varphi(n)),$$

so,

$$ed \equiv 1(mod \quad \varphi(n))$$

or in other words $ed - 1$ is a multiple of $\varphi(n)$, we say $k\varphi(n)$. Then we can find message by calculating

$$d(e(m)) \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\varphi(n)} \equiv m(m^{\varphi(n)})^k \equiv m.1^k \equiv m(mod \quad n)$$

where we have used theorem 2.3.10 to conclude that $m^{\varphi(n)} \equiv 1(mod \quad n)$. □

2.6 System tools

In order to achieve our needed results, we use some hardware tools and software tools.

2.6.1 Hardware.

- **Processor:**Intel Core i-5-3340M CPU @ 2.70GHz x 4
- **Random Access Memory:** 4 GB
- **Operating System type:** 64-bit
- **Graphics:** Intel Ivybridge Mobile

2.6.2 Software.

- **Operating System:** Linux Ubuntu 14.04
- **Programming language:** Python 3.0 with idle editor 3.4
- **Database:** MYSQL-Database (version-5.5 5.5.41).

These are the system tools that will be used in this project of developing the encrypted patient database including the doctor user interface. Every doctor will have his own public key and private key. The doctor will publish his public key to all physicians. Physicians can use the doctor's public key to encrypt patients images. Only the doctor having the corresponding private key will be able to decrypt it.

3. System analysis and design

In this chapter, we will present how the system is designed to solve our problem defined in section 1.2. Generally, the system is a collection of components that work together to realize some objectives [3]. We will present the system architecture which will describe graphically, the way we will solve our problem. Finally, we will describe in details our system that will be coded in computer programming language (Python 3) in order to implement it.

3.1 System Architecture

Every system is designed to solve one or more problems. This system will help the doctors to save encrypted patient data and also to decrypt images from physicians as a solution to the problem described in section 1.2. This system will keep patients data safely.

In figure 3.1, the doctor has right for registering a new doctor with all attributes shown. A doctor can access the system by using the local key. He is able to access the database where all patients data are saved. By using the common key infrastructure, he is able to encrypt and decrypt patients data including images file from a physician. In this thesis the word treatment has the same meaning as illness.

3.2 System design

3.2.1 Modelling Language (UML).

UML is the graphical modelling language which is helpful in developing, understanding and communicating different views of the system[11]. There are many types of diagrams. In this thesis, we will present three of them; namely the use case digram, the data-stream diagram and the class digram.

3.2.2 case diagram.

A use case diagram is the UML diagram which describes the system by actors and use cases. An actor starts a use case[11].

In our system the doctor is an actor. He receives information from patients and entered them in the system. A use case can be defined as a sequence of interactions between the doctor and the system[11]. For instance, if we consider the physician in figure 3.2, he can decrypt the patient data which are in the system, so he is an actor.

Figure 3.2 shows three actors. The first one is the patient. He is there only to give information to the doctor. He is *an external actor* and will not do anything to the system. The doctor can't enter a patient's data to the system without him.

Once the doctor inputs information, he can save them in the system. He is able to encrypt and decrypt the saved patient data. He can edit and view patients data. He can send patients data to the physician for getting the corresponding image information. The doctor also have the right for registering new doctors in the system.

The last actor of our system is the physician who is able to decrypt patients data from the doctor and send an encrypted image information to him. The doctor can snaps-in image from physician console.

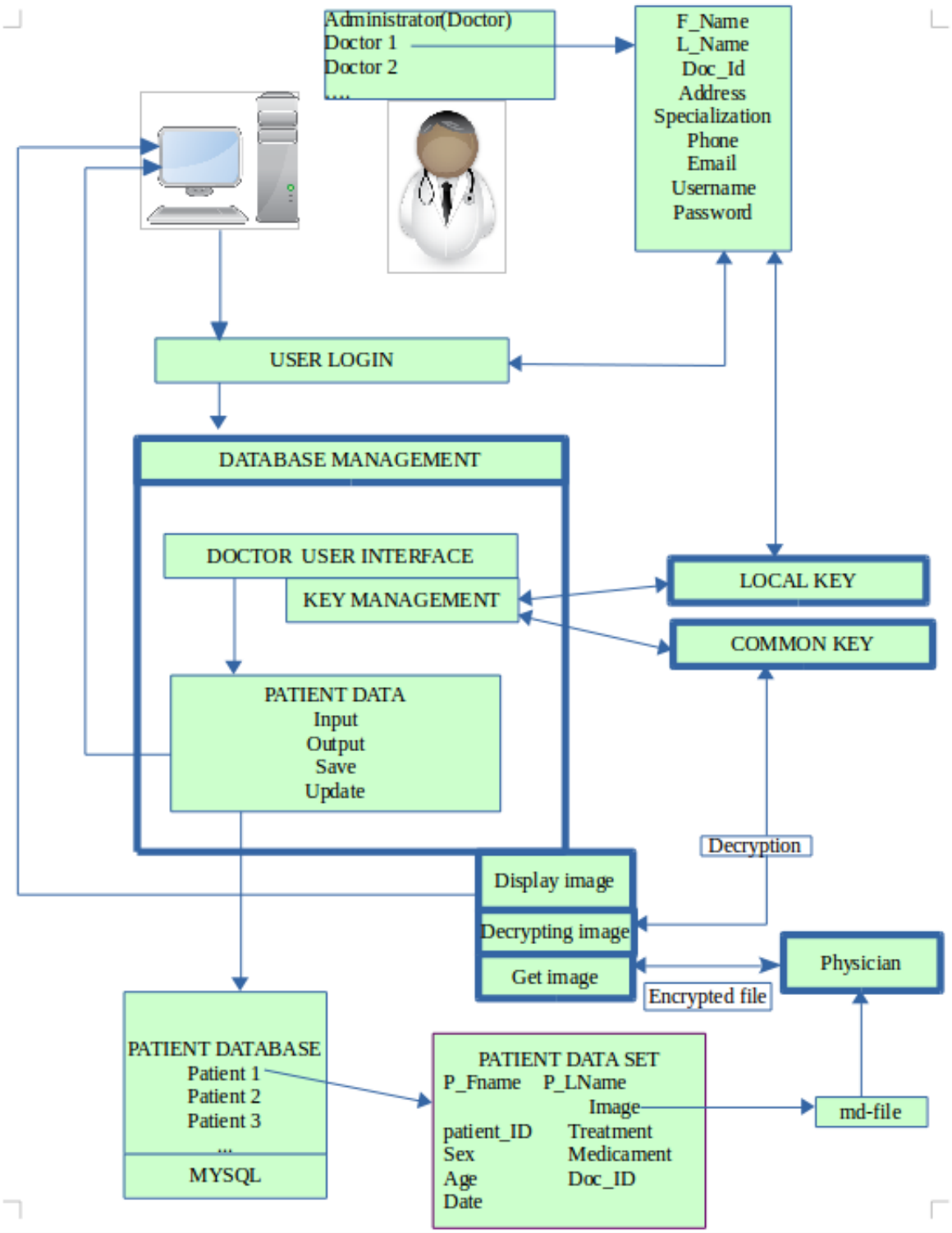


Figure 3.1: System architecture

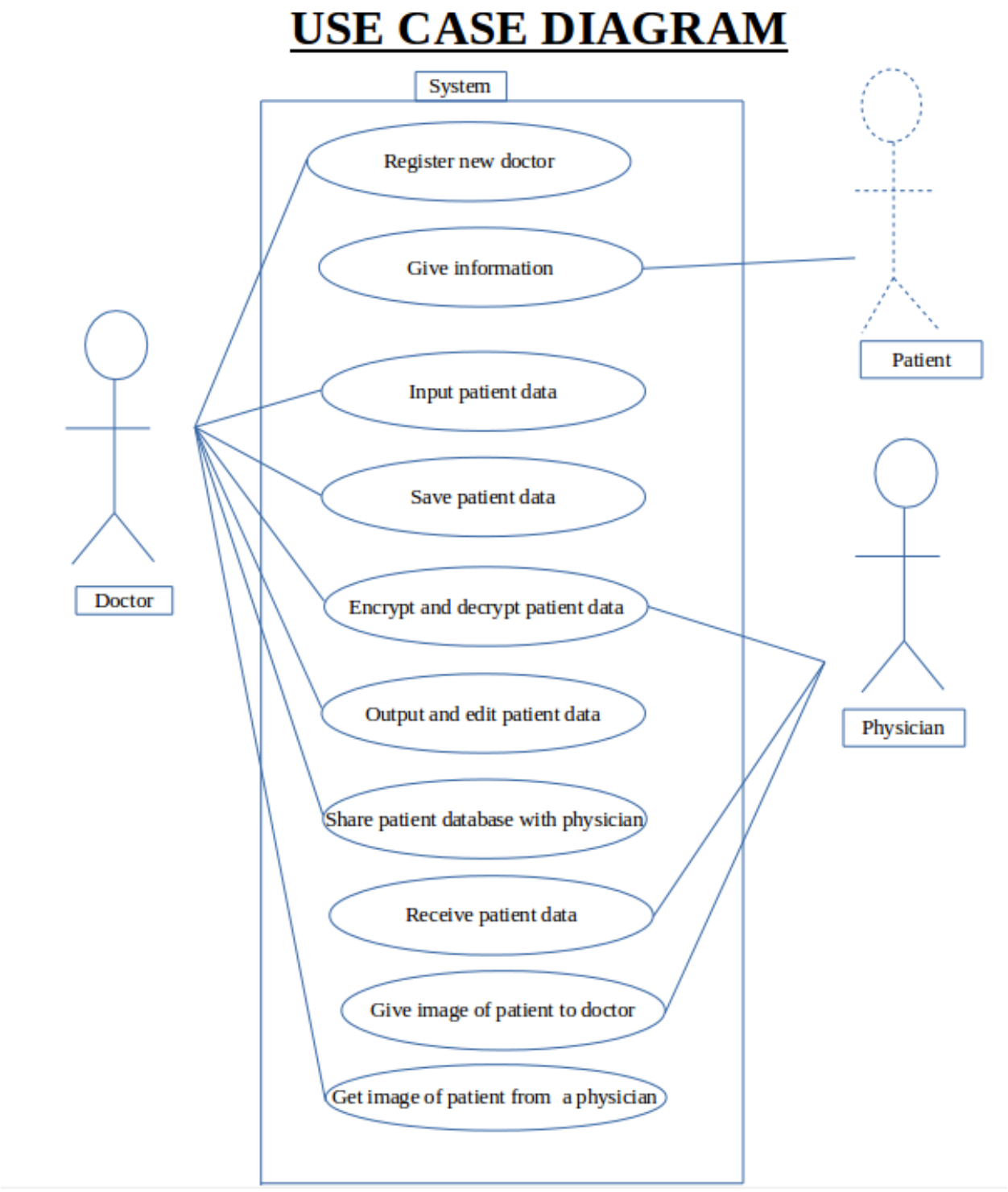


Figure 3.2: A use case diagram

Use cases description

- `Register new doctor:` This use case helps the existing doctor to register the new doctor in the system.
- `Give information:` The patient gives the information to the doctor in order to consultant him. The doctor gathers the information of any disease from patient.
- `Input patient data:` The doctor inputs the information from the patient into the system.
- `Save patient data:` The doctor saves the patient data in the system and here, the doctor must encrypt these data.
- `Encrypt and decrypt patient data:` The doctor needs to save the encrypted data in the system and decrypts these data any time he needs to output them. The physician uses the public key of the doctor who requires a specific image of patient. Then, the physician encrypts that image and sends it to the doctor. The doctor needs to use his private key to decrypt that image.
- `Output and edit patient data:` The doctor outputs patient data to the screen and decrypts them and then, updates them. The doctor saves the updated patient data in the database.
- `Share patient data with physician:` The doctor sends the patient ID to the physician for image of patient.
- `Receive patient data:` The physician receives the patient ID and finds in the medical images database the image corresponding to the received patient ID from the doctor.
- `Give image of patient to doctor:` The physician sends the encrypted patient image to the doctor. The physician uses the doctor's public key to encrypt the image.
- `Get image of patient from a physician:` The doctor gets the image from a physician and that image is encrypted. He needs to decrypt it by using his own private key.

3.2.3 Data stream diagram.

The data stream diagram shows how operations flow and the actor who is conducting them with the data in system. Diagrams show the external bodies of sending and receiving information. They show the processes that change information, the information flow itself and the place where the information will be stored [4]. Figure 3.3 shows the patients data flow. We have six operations and data.

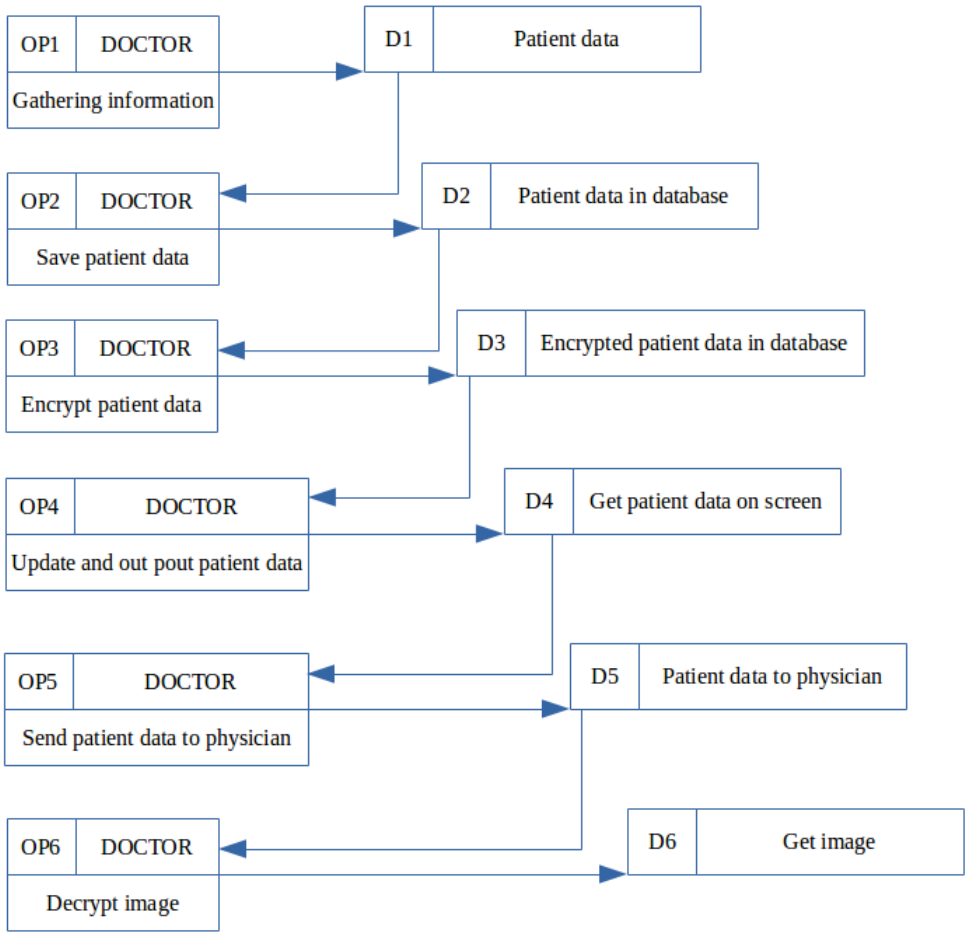


Figure 3.3: Data stream diagram

3.2.4 Class diagram.

In a class diagram, classes are presented as boxes, and the relationships between them are presented as lines connecting the boxes[11]. In this work, we have used an association diagram where the doctor class specifies many instances of patient class. Figure 3.4 shows multiplicity of an association where one doctor can consult many patients, and one patient can be consulted by many doctors.

Figure 3.4 shows all saved in our database.

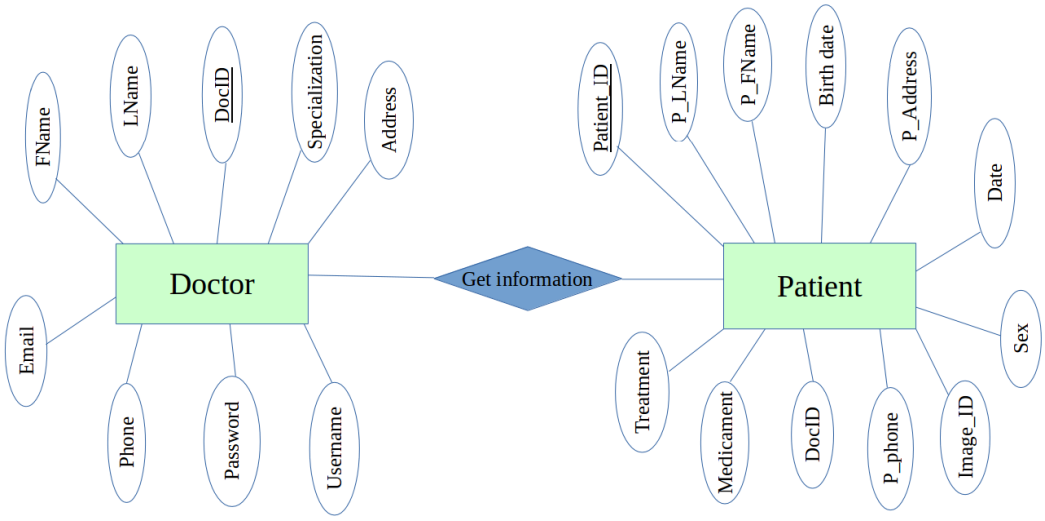


Figure 3.4: Class diagram

3.2.5 Data model in database.

The data model shows the way data are organized in different tables in the database. In figure 3.5 we have three tables. The middle table is composed by the primary keys of other two tables. The first is the doctors table, which has different attributes. In this table, the DocId attribute is considered as the primary key, and the foreign key in patients table. The last one is patients table. It has the relation of many to many with the doctors table. That means, one doctor can consult many patients. Also in reverse, one patient can be consulted by many doctors at different time intervals.

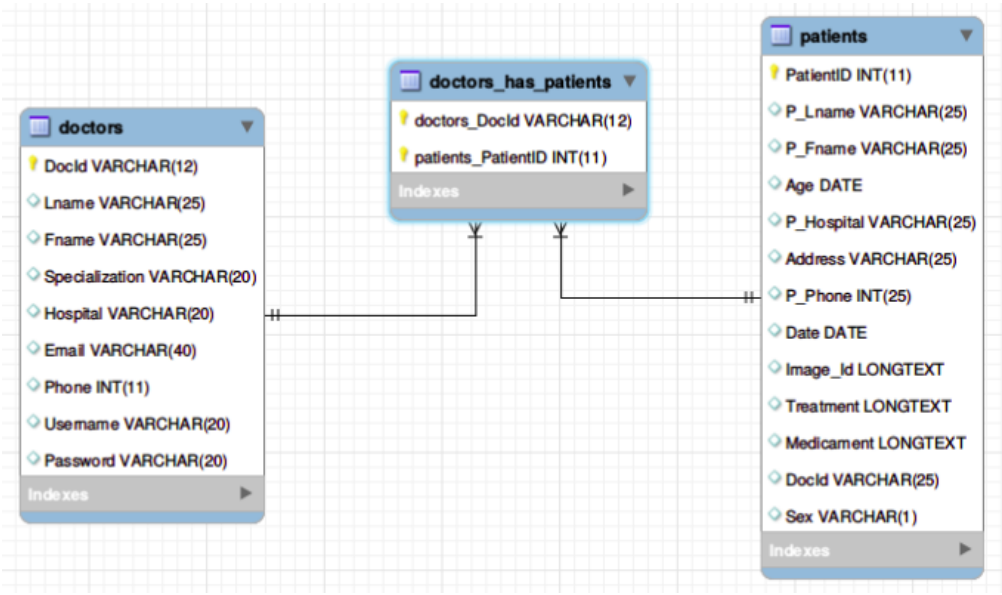


Figure 3.5: Data model diagram

In the next chapter, we will see the implementation of the correspondence between all attributes in both patients and doctors tables on the user interface.

4. Implementation and result

4.1 Introduction

In this chapter, we will apply the methodology of encrypting patients data in the database. The doctor will interact friendly with computer through the graphic user interface (GUI) of the system. A GUI demands a combination of technologies to form a friendly interaction between user and computer in order to gather and produce information. In the previous chapter, we saw the complexity of the inner system. Now, the GUI will make the system easier to use. The doctor doesn't need to know the complexity of program in order to use the system. We are going to see the implementation technologies used to develop an encrypted patients database with a doctor user interface system.

4.2 Implementation technologies

MYSQL

In this thesis, we have used MySQL for implementation of our system. MySQL stands for My Structured Query Language. It is involved in the creation, management and the manipulation of a database. The database is a structured collection of data in various tables[23]. We have used updated version of MySQL (MySQL-5.5 5.5.41-0ubuntu0.14.04.1 source package in ubuntu operating system) in order to add, edit, access and manage data in a doctor database. We have created a database called Doctor with two tables such as patients and doctors tables with their attributes. Patients table holds all information of patients where some attributes such as image ID, medicament and treatment are encrypted. The doctors table holds all information of different doctors.

PYTHON

In the implementation of the doctor user interface, we have used python-3 with the editor idle-3.4. Python helps us to make a program which leaded to the result. In this program, we have used some mathematical tools to encrypt and decrypt the input patient data on the patient interface, and generated public and private keys for registration of new doctors.

4.3 Result and evaluation

The main objective of this thesis is to develop an encrypted patient database including a doctor user interface.

When we run our program, the LOGIN form appears as shown in figure 4.1 and the doctor clicks on Login button to continue to the next form that appeared as can be seen in figure 4.2. Then, the doctor is required to enter his credentials (user name and password). The system will go to check if these credentials exist in the database (doctors table).

LOGIN

When credentials are not present in the database, the system will give the doctor the message shown in figure 4.3 by informing him that the user name or password entered is not correct and he can try again.

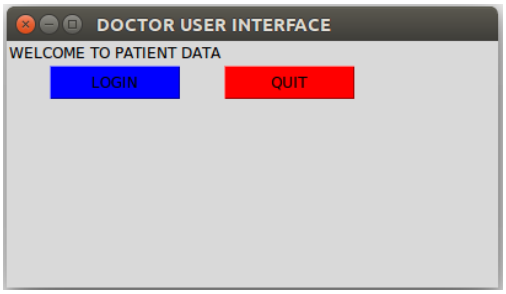


Figure 4.1: Login form

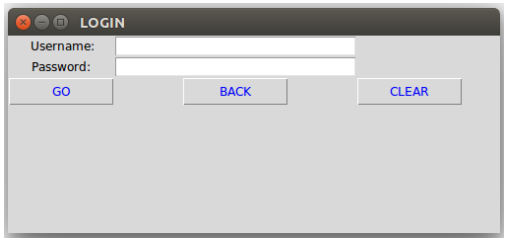


Figure 4.2: Login form

Otherwise the system will give the doctor the message of valid credentials and he needs to click on OK button to continue as it appears in figure 4.4. The system will go to SYSTEM form.

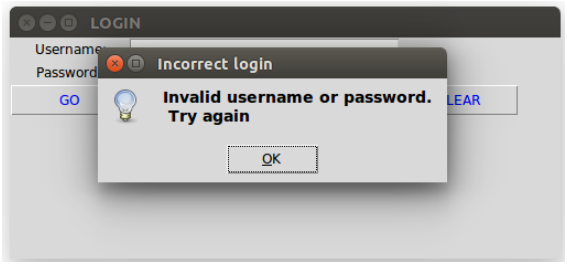


Figure 4.3: Doctor user interface

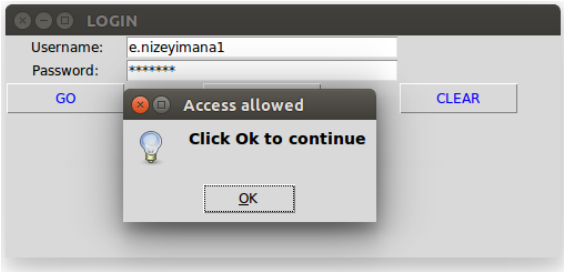


Figure 4.4: Login form

Figure 4.5 shows the system with three buttons: patient data , doctor and back buttons. Once the patient data button is clicked, the system will go to the new form which is called PATIENT DATA. When the button of doctor is clicked, the system will lead the user to the form called DOCTOR. Lastly, the button of back take the user to the LOGIN form.

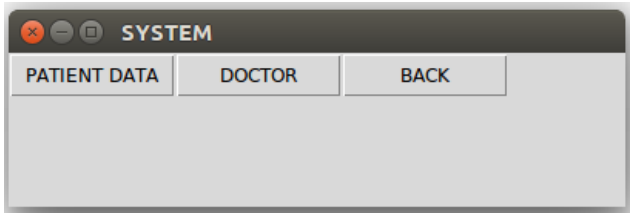


Figure 4.5: System interface

PATIENTS DATA

Figure 4.6 shows the entry of all data required for a patient as described in different labels. When the doctor opens the PATIENT DATA form, he fills all text fields accordingly. The system sets the date and the patient ID automatically. The button back can be clicked to go back to the SYSTEM form. The doctor inputs all other remaining information of patients as shown in figure 4.7. Once the doctor finishes to input all patients information, he will click on save button. The doctor can clear all information in all text fields except the date because it is set automatically.

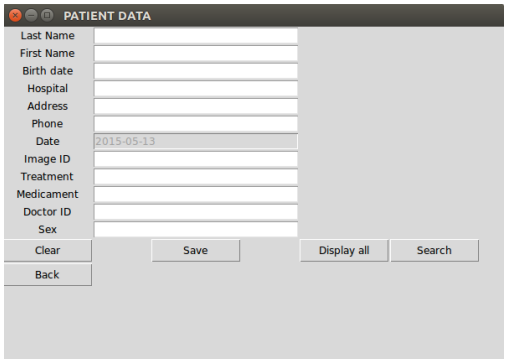


Figure 4.6: Patient interface

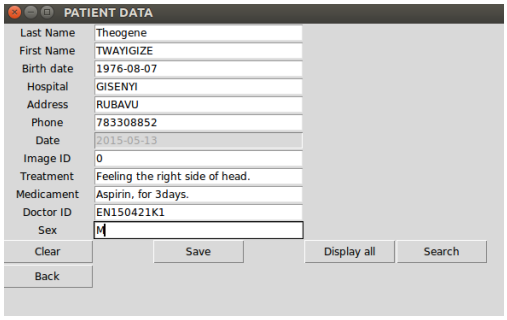


Figure 4.7: Registering new patient

SAVING ENCRYPTED PATIENT DATA

Figure 4.8 appears with the text field to input the public key which is needed for encryption of patient data. We have configured the system to encrypt the image ID of patient, the treatment and the medicament the patients should take. The doctor is required to enter his public key which the administrator has created with 1500 digits in order to encrypt patients data. When the button of encryption in figure 4.9 is clicked, all data on the Patient data form will be saved directly in the patients table in the database. The encrypted data will be saved in the database their encrypted form. Figure 4.10 appears to confirm that data are already saved in the database.

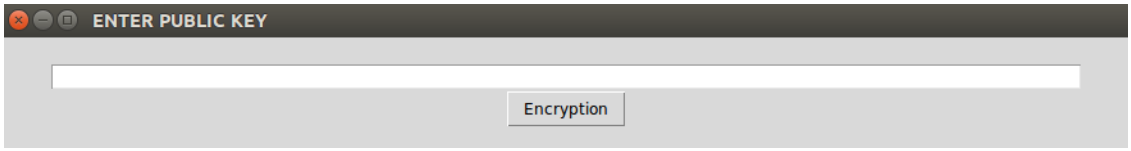


Figure 4.8: Encryption window

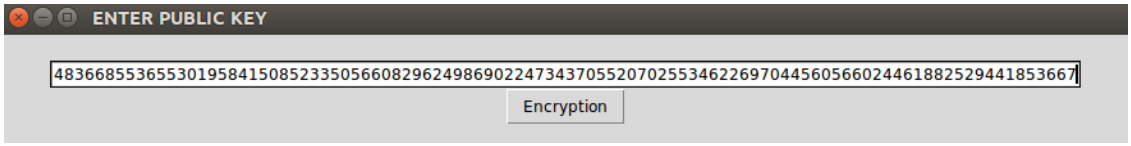


Figure 4.9: The public key

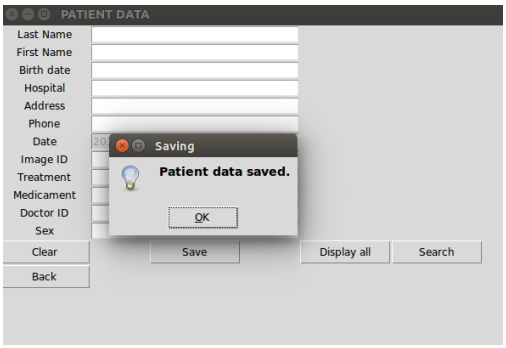
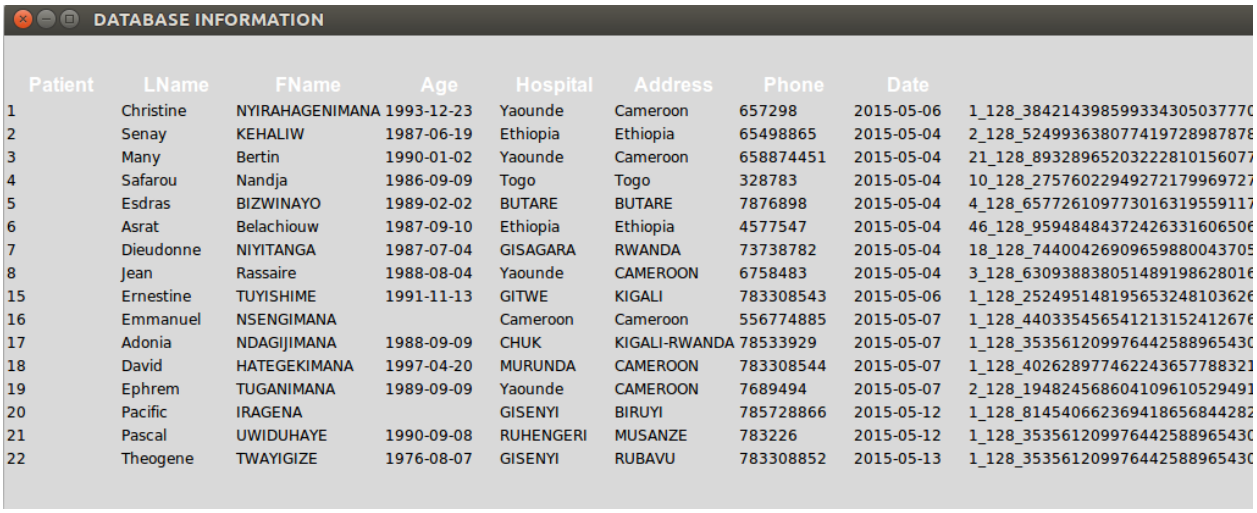


Figure 4.10: Saved data in database

DISPLAYING PATIENTS DATA

When the doctor finishes to save patients data, he is able to retrieve all the data contained in the database by clicking on display all button. The figure 4.11 displaying all patients in the system. The figure 4.11 doesn't show all data of tables because the data encrypted are too long, around 1500 digits. If the doctor needs to see all information, he uses the resize pointer which appears on that form.



Patient	LName	FName	Age	Hospital	Address	Phone	Date	
1	Christine	NYIRAHAGENIMANA	1993-12-23	Yaounde	Cameroon	657298	2015-05-06	1_128_384214398599334305037770
2	Senay	KEHALIW	1987-06-19	Ethiopia	Ethiopia	65498865	2015-05-04	2_128_524993638077419728987878
3	Many	Bertin	1990-01-02	Yaounde	Cameroon	658874451	2015-05-04	21_128_893289652032228101560777
4	Safarou	Nandja	1986-09-09	Togo	Togo	328783	2015-05-04	10_128_275760229492721799697277
5	Esdras	BIZWINAYO	1989-02-02	BUTARE	BUTARE	7876898	2015-05-04	4_128_657726109773016319559117
6	Asrat	Belachiouw	1987-09-10	Ethiopia	Ethiopia	4577547	2015-05-04	46_128_959484843724263316065067
7	Dieudonne	NIYITANGA	1987-07-04	GISAGARA	RWANDA	73738782	2015-05-04	18_128_744004269096598800437057
8	Jean	Rassaire	1988-08-04	Yaounde	CAMEROON	6758483	2015-05-04	3_128_630938838051489198628016
15	Ernestine	TUYISHIME	1991-11-13	GITWE	KIGALI	783308543	2015-05-06	1_128_252495148195653248103626
16	Emmanuel	NSENGIMANA		Cameroon	Cameroon	556774885	2015-05-07	1_128_440335456541213152412676
17	Adonia	NDAGIJIMANA	1988-09-09	CHUK	KIGALI-RWANDA	78533929	2015-05-07	1_128_353561209976442588965430
18	David	HATEGEKIMANA	1997-04-20	MURUNDA	CAMEROON	783308544	2015-05-07	1_128_402628977462243657788321
19	Ephrem	TUGANIMANA	1989-09-09	Yaounde	CAMEROON	7689494	2015-05-07	2_128_194824568604109610529491
20	Pacific	IRAGENA		GISENYI	BIRUYI	785728866	2015-05-12	1_128_814540662369418656844282
21	Pascal	UWIDUHAYE	1990-09-08	RUHENGERI	MUSANZE	783226	2015-05-12	1_128_353561209976442588965430
22	Theogene	TWAYIGIZE	1976-08-07	GISENYI	RUBAVU	783308852	2015-05-13	1_128_353561209976442588965430

Figure 4.11: All patients in system

SEARCHING AND DECRYPTING PATIENT DATA

Figure 4.12 appears when the doctor needs to search a particular information of patient. He needs to press the button search and to enter the patient ID. Figure 4.13 shows data in different text fields. Some of these data are encrypted and the doctor needs to decrypt them.

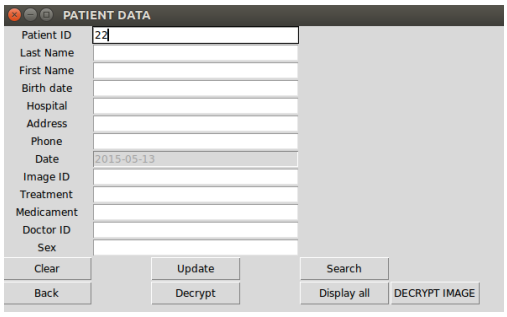


Figure 4.12: Enter patient ID

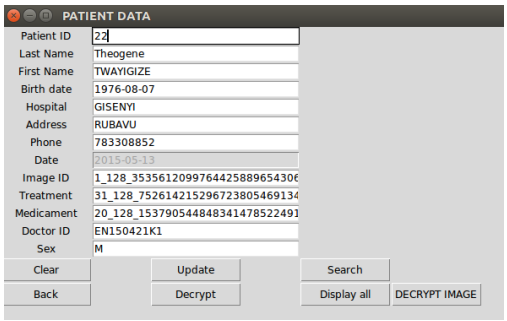


Figure 4.13: Searching a particular patient

Figure 4.14 appears when the decrypt button in figure 4.13 is clicked. The doctor needs to enter his private key on the text field as shown in figure 4.15 which is compatible with the public key that is used to encrypt. When the button decryption in figure 4.15 is clicked, all the encrypted data are decrypted directly as shown in figure 4.7.

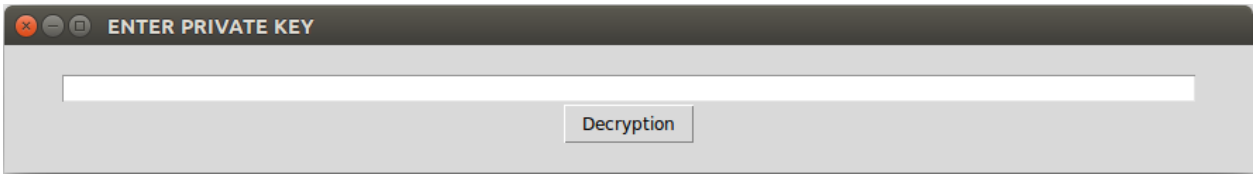


Figure 4.14: Private key

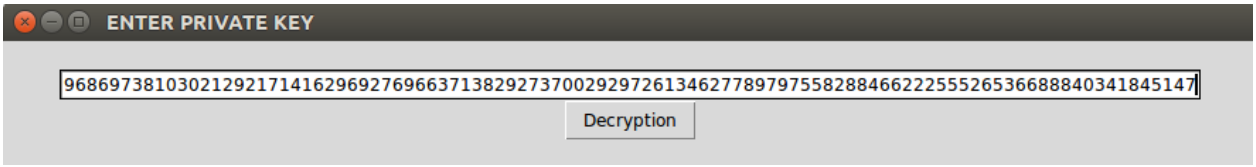


Figure 4.15: Decrypted interface

UPDATING PATIENT DATA

A doctor might need to update a patient's data for example once he finished to record data and he finds an error. He has the right to update patients data. He searches first the patients data by using patients ID and decrypts them. Figure 4.7 shows the data after the decryption. The doctor can edit any data of patient except date and patient ID as shown in figure 4.16. Then, the doctor clicks the update button and figure 4.18 appears and he enters his public key in the text field as done on saving new patient in the system. All the data with that patient ID will be replaced by the new entries. The message in figure 4.17 confirms the storage of updated data.

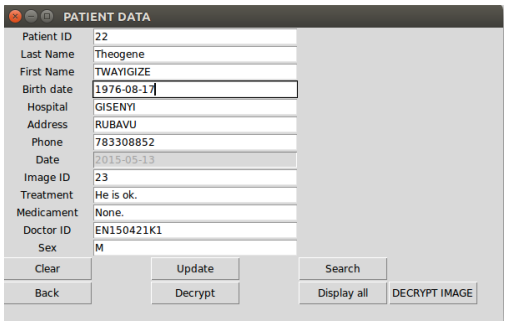


Figure 4.16: Update patient data

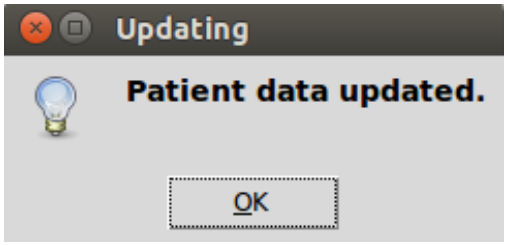


Figure 4.17: Updated patient data in database

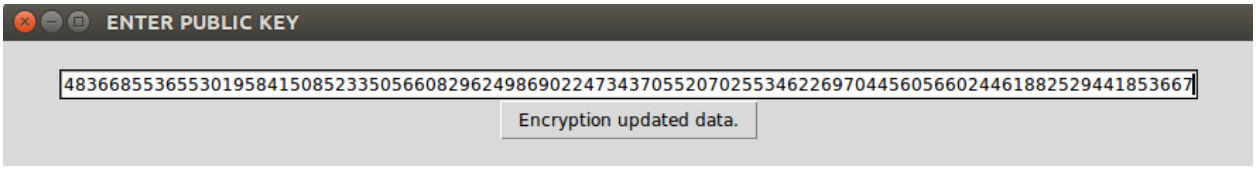


Figure 4.18: Public key to update patient data

DECRYPTION OF IMAGE

The doctor may need to have a specific image depending on the sickness. The doctor sends the patient ID to the physician. The physician takes the requested image of patient and encrypts it. Then, the physician sends the image to the doctor through email. The image from the physician is encrypted with

the public key of the doctor who requests it. The doctor needs to decrypt the image from physician by using his private key. The doctor clicks the DECRYPT IMAGE button which appears in figure 4.16. Figure 4.19 appears with two labels; one corresponding with the text field of the image message from physician and another for private key of the doctor. The doctor copies and pastes both the image message and private key in text fields respectively as shown in figure 4.20. Then, he clicks decryption image button. The figure 4.21 appears on the screen. The decryption can take some seconds of time due to the size of the image.

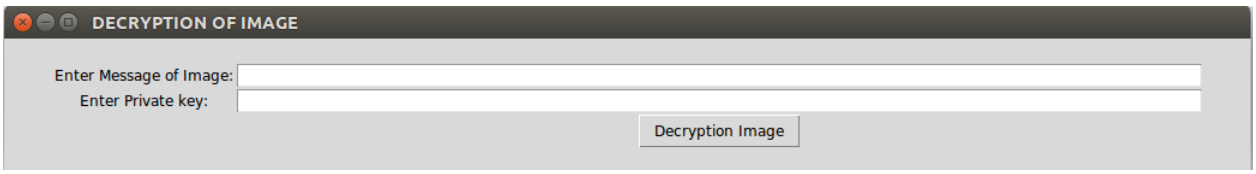


Figure 4.19: Image decryption interface

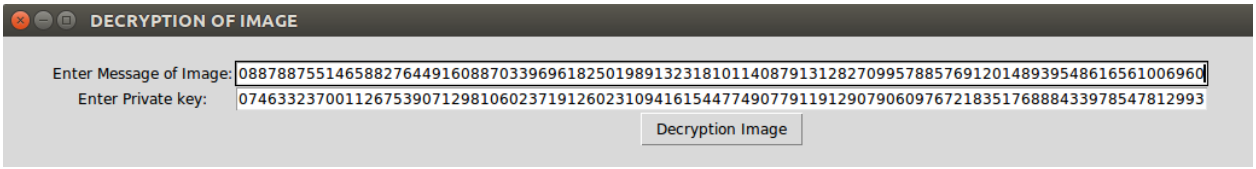


Figure 4.20: Image message and the private key

Figure 4.21: Image from physician of patient

DOCTOR REGISTRATION

The DOCTOR form gives the existing doctor the ability to register a new doctor. The administrator can update the doctor information like password and pair of keys; and searches the doctor information. Figure 4.22 appears when the button doctor from figure 4.5 is clicked. The doctor can fill text fields as shown on figure 4.23 for a new doctor. Then, the doctor clicks the register button. The confirmation message of new doctor is successful and this is shown in figure 4.24. Once the register button is clicked, the two files are created directly and hold both public key and private key of the new doctor. These

keys also will be saved automatically in the database in the table called doc. The keys generated are of size 1500 digits. The public key will be known by everyone in order to encrypt messages to send to the doctor , this key is the one that is used to encrypt patient data. On the other hand, the private key will help the doctor to decrypt data.

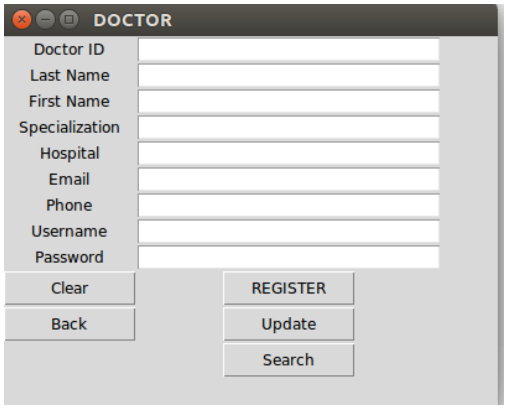


Figure 4.22: Doctor interface

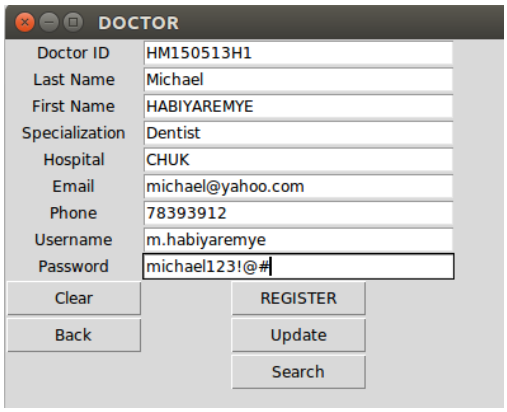


Figure 4.23: Register new doctor

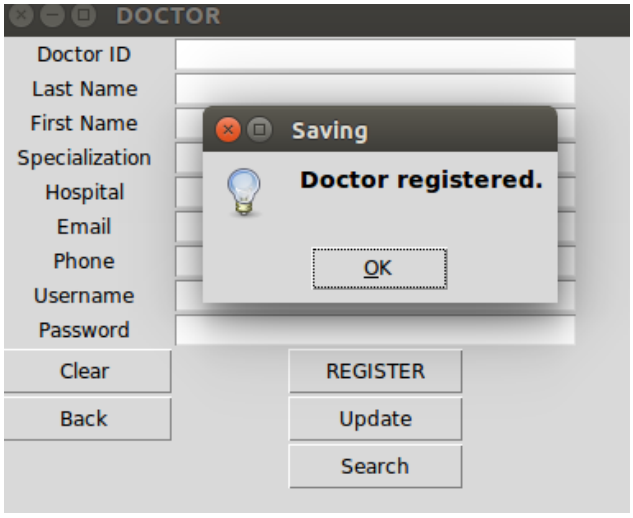


Figure 4.24: Doctor interface

UPDATE AND SEARCH FOR DOCTOR

Figure 4.22 has another button called search. This button is clicked once the administrator needs to update the doctor information especially for password or keys. The administrator enters the Doctor ID in the text field as shown in figure 4.25, and clicks on the search button. He will get all the corresponding information as seen in figure 4.23.

The administrator can edit information of doctor. Then, he updates them after some editing as seen in figure 4.26. When the administrator finishes the update, the message in figure 4.27 appeared for confirmation. If he searches again he will find updated data. The doctor can search any information of other doctor but he can't see username and password as shown in figure 4.26.

The button back helps the doctor to go back to the previous form. The button quit which is on Doctor user interface form helps the doctor to quit the system.

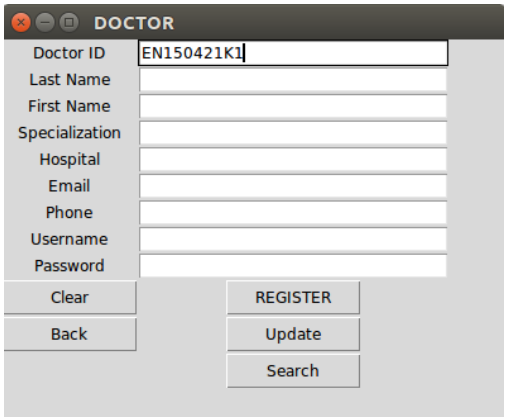


Figure 4.25: Search Doctor information

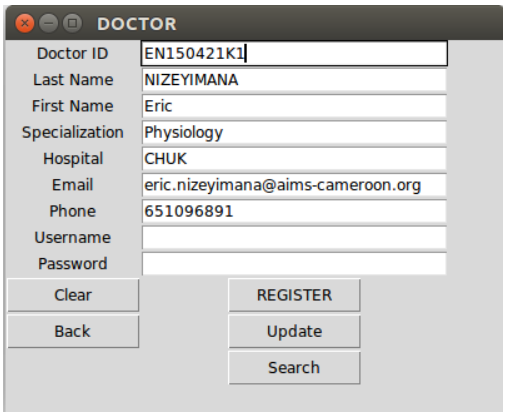


Figure 4.26: Data for doctor edited

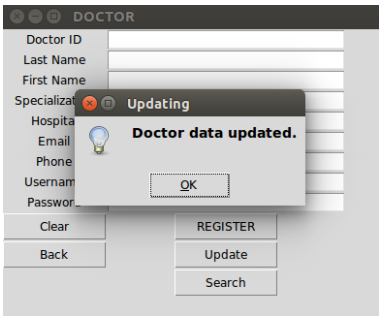


Figure 4.27: Updated data

In this chapter, we have seen the technologies that have been used to achieve our objectives. The result that we have got at the end solves our problem statement that is described in chapter one. Now, patients can't fear electronic data are more secured since they are kept with the strong encryption as shown in the above result.

5. Conclusion and recommendation

5.1 Conclusion

The purpose of this thesis is to develop an encrypted patient database with a doctor user interface. The thesis gives an impression of the need of electronic medical record of each patient at the hospital. This thesis sought to answer these questions:

• How can patients data be secured during both the sharing process of these data to doctors and their storage on hospital computers in databases?

• How does the doctor decrypt the encrypted image from physician?

RSA is a good tool for mathematical cryptography helping to get a solution of the above question. It is based on the difficulty of factoring n which is the product of two large prime numbers chosen randomly p and q. It uses public key (n, e) for encryption and private key (n, d) for decryption. RSA encryption algorithm turns a message M into block of integers m such that $0 \leqslant m < n$ and computes the cipher-text c by $c \equiv m^e (mod \quad n)$. RSA decryption algorithm takes the cipher-text c and computes the given message m by $m \equiv c^d (mod \quad n)$. This algorithm ensures the security of patient's database which cannot be easily broken by anyone as it is proven in the result of our system.

5.2 Recommendation

This thesis has been limited by the time constraint. It shows the prototype of system, not a ready-to-use system. We recommend future researchers to use the symmetric cryptography for encryption of patient data. This will allow any doctor to decrypt and encrypt data using private key. They can make also a cloud database of patient data. This will need the digital certificate for doctor who request the patient data for authentication. The security is used everywhere for keeping information, but there are many cryptanalysis techniques that try to extract the shared information and make it available to unauthorized people. Mathematicians and computer scientists should be oriented in the field of cryptography to ensure the integrity of information.

Appendix A. Attached files

Below there are files description of all programs necessary for our system.

1. **Patient_ Encryption.py** This is the main python program of the system.
2. **cryptomath.py** This is a python program which has functions like $gcd()$ and $findModInverse()$.
3. **rabinMiller.py** This is a python program which checks if an number is prime by function $isPrime()$ and generates large prime with key size of 1500 bits by using function $generateLargePrime()$.
4. **makeRsaKeys1.py** This is a python program which generates both public and private keys by function $generateKeys()$ and create files to save these keys by function $makeKeyFiles$.
   - `generateKeys(keySize)`: This function creates a public and private key pair with the keys that are keySize bits in size. Due to the length of key, it can take a while to run.
   - `makeKeyFiles()`: This function creates two files with "x_ Pubkey.txt" and "x_ privkey.txt", where x is a last name of the doctor. These files are composed by n, e and d, e integers written in them, delimited by a comma.
5. **rsaCipher1.py** This is python program which encrypts and decrypts message. It has functions like:
   - `getBlocksFromText()`: This function converts a string to a list of block integers.
   - `getTextFromBlocks()`: This function converts a list of blocks integers to the message string.
   - `encryptMessage()`:This function encrypts each block of integers.
   - `decryptMessage()`: This function decrypts each list of encrypted block integers into the original message string.
6. **rsaCipher2.py** This is a python program which works like $rsaCipher1.py$ but its functions are modified for image encryption and decryption, since it keeps the height and width of image.
7. **Doctor.sql** This is MySQL file which contains all information of the database. It is composed by three tables: patients table, doc table, and doctors table.

Acknowledgements

I would like to thank first and foremost the Almighty God for being my strength and guide in writing of this thesis. Without Him, I would not have the wisdom or the physical ability to do so.

I owe my deepest gratitude to my thesis supervisor Prof. Dr. Axel Schumann for devoting much time to find some necessary literature for this thesis, guide and support me from the initial to the final level. I highly thank him for enabling me to develop an understanding of the subject.

I would like to thank Prof. Dr. Mama Foupouagnigni and all AIMS-CAMEROON staffs for enabling me to get the best education that I have earned. My thanks goes to the whole network of African Institution for Mathematical Science (AIMS).

I express my thanks to all lecturers and tutors who delivered different courses to me which have the impact to the completion of this thesis.

I am grateful to Mrs. Nathalie Wandji for her guidance, wisdom and tireless assistance during every step of this thesis.

I am heartily thankful to Dr. Bajpai Gaurav for advising and encouraging me during the writing of this thesis and all my colleagues in general.

I thank all my family and relatives especially my mother for always being supportive of my education.

I offer my regards and blessings to all my classmates and friends who supported me in any respect during the completion of this thesis.

References

[1] Workshop on side - channel analysis: Cryptography concepts and background. http://www.cryptography.com/public/pdf/Cryptography_Reference.pdf. Accessed: March 30, 2015.

[2] Using rsa public key exchange - how it works. http://www.internet-computer-security.com/VPN-Guide/RSA.html, 2008. Accessed: April 6, 2015.

[3] Introduction to system analysis and design. http://download.nos.org/cca/cca1.pdf, 2015. Accessed on April 23, 2015.

[4] System analysis 16. http://www.betsaonline.com/SystemAnalysis/system_analysis.pdf, 2015. Accessed on April 23, 2015.

[5] Joseph A Akinyele, Matthew W Pagano, Matthew D Green, Christoph U Lehmann, Zachary NJ Peterson, and Aviel D Rubin. Securing electronic medical records using attribute-based encryption on mobile devices. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, pages 75–86. ACM, 2011.

[6] MS Anoop. Elliptic curve cryptography, an implementation guide. *online Implementation Tutorial, Tata Elxsi, India*, 5, 2007.

[7] Carol Franc Buck. Designing a consumer-centered personal health record. Technical report, Tech. Rep., California Health Foundation, 2007.

[8] Sergio T Carvalho, Leonardo Murta, and Orlando Loques. Variabilities as first-class elements in product line architectures of homecare systems. In *Proceedings of the 4th International Workshop on Software Engineering in Health Care*, pages 33–39. IEEE Press, 2012.

[9] Fred Cohen. A short history of cryptography. *Retrieved Oct*, 23, 2011.

[10] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. *Cryptography Engineering: Design Principles and Practical Applications: Design Principles and Practical Applications*. John Wiley & Sons, 2011.

[11] Hassan Gomaa. *Software modeling and design: UML, use cases, patterns, and software architectures*. Cambridge University Press, 2011.

[12] Young-Guk Ha and Yung-Cheol Byun. A ubiquitous homecare service system using a wearable user interface device. In *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on*, pages 649–650. IEEE, 2012.

[13] André Heck and Wolfram Koepf. *Introduction to MAPLE*, volume 1993. Springer-Verlag New York, 1993.

[14] AnaFred JoaquimFilipe HugoGamboa. Biomedical engineering systems and technologies.

[15] John F Humphreys and Mike Y Prest. *Numbers, groups and codes*. Cambridge University Press, 2004.

[16] David Ireland. Rsa algorithm. http://www.di-mgt.com.au/rsa_alg.html, 2015. Accessed on March 26, 2015.

[17] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. CRC PRESS, 2007.

[18] George R Kim, Christoph U Lehmann, et al. Pediatric aspects of inpatient health information technology systems. *Pediatrics*, 122(6):e1287–e1296, 2008.

[19] Bobby Kleinberg. The miller-rabin randomized primality test. http://www.cs.cornell.edu/courses/cs4820/.../MillerRabin.pdf, 2010. Accessed on May 11, 2015.

[20] Steve Lohr. Ge and intel join forces on health technologies. *New York Times*, 3, 2009.

[21] MatD. Rsa - diddife - hellman explained. http://www.mat-d.com/site/rsa-diffie-hellman-explained-in-3-minutes/, 2007. Accessed: April 6, 2015.

[22] Robert J McEliece. *Finite fields for computer scientists and engineers*, volume 23. Kluwer Academic Publishers Boston, 1987.

[23] Mike. An introduction to databases. http://www.ucl.ac.uk/archaeology/cisp/database/manual/node1.html, 2010. Accessed: May 7, 2015.

[24] Jelena Mišic. Enforcing patient privacy in healthcare wsns using ecc implemented on 802.15. 4 beacon enabled clusters. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 686–691. IEEE Computer Society, 2008.

[25] Alexandros Pantelopoulos and Nikolaos G Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(1):1–12, 2010.

[26] Erich Peterson and Siqing Li. An overview of cryptographic systems and encrypting database data. http://www.4guysfromrolla.com/articles/021407-1.aspx, 2007. Accessed: April 5, 2015.

[27] Thomas Pornin. Why is elliptic curve cryptography not widely used, compared to rsa? http://crypto.stackexchange.com/questions/1190/why-is-elliptic-curve-cryptography-not-widely-used-compared-to-rsa, 2011. Accessed: April 8, 2015.

[28] MIRA SCARVALONE. Rsa encryption and diffie hellman key exchange. 2009.

[29] William Stein. *Elementary number theory: primes, congruences, and secrets: a computational approach*. Springer Science & Business Media, 2008.

[30] Piyamas Suapang, Kobchai Dejhan, and Surapun Yimmun. Medical image archiving, processing, analysis and communication system for teleradiology. In *TENCON 2010-2010 IEEE Region 10 Conference*, pages 339–345. IEEE, 2010.

[31] Albert Sweigart. *Hacking Secret Ciphers with Python*. Amazon Distribution, 2013.

[32] Kun-Lin Tsai, Fang-Yie Leu, Tien-Han Wu, SS Chiou, Yu-Wei Liu, and Han-Yun Liu. A secure ecc-based electronic medical record system. *Journal of Internet Services and Information Security*, 4(1):47–57, 2014.

[33] David Wagner. Discrete mathematics for computer science. http://www.cs.berkeley.edu/~daw/teaching/cs70-f03/, 2013. Accessed on May 11, 2015.

[34] Eric W. Weisstein. Rabin-miller strong pseudoprime test. http://mathworld.wolfram.com/Rabin-MillerStrongPseudoprimeTest.html, 2015. Accessed on April 12, 2015.

[35] Yuanli Wu and Hongqiao Yang. An electronic medical records review system for mobile healthcare based on web services. In *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on*, pages 1040–1044. IEEE, 2012.

[36] Bill Young. Foundation of computer security. http://www.cs.utexas.edu/users/byoung/cs361/lecture44.pdf. Accessed: April 6, 2015.