Software Engineering Department

Ort Braude College

Capstone Project Phase A

22-1-D-11

Securing Medical Institutes Data, Encryption using E-ART Algorithm.

Supervisor: Reuven Cohen

Students:

- Wael Hamada
- Ahmad Haj

Contents

Abstract

On average, one of every 60 Israeli organizations or firms is targeted every week with ransomware attacks, an increase of 30% over the rate in 2020. The most targeted sectors around the world, including Israel, are education and research institutes, followed by government and security organizations, and health institutes**.**

Health institutions store sensitive medical data, this work focus on keeping this data safe and secured from cyber-attacks, using a new, fast, and reliable encryption algorithm called E-ART.

In this project, we will develop a robust application, with an easy-to-use interface. The application will include encryption-decryption capability for medical data, and more other features which will help keep the medical data secured and strong resisting against cyber-attacks.

1. Introduction

Cyber security is a domain that is rapidly rising over the last couple of years, and systems are required to defend themselves against cybercriminals attacks.

Cyberattacks have been rated as the fifth top rated risk in 2020 and data security became the new norm across public and private sectors. This industry continues to grow in 2021 as IoT (Internet of things) cyberattacks specifically are expected to double by 2025.

This project target is to develop a strong resistance system for securing medical data, this system will aim to prevent cyberattacks and defend if an attack happened.

The system will be based on a new algorithm called E-ART; this algorithm fulfills the sever criterions of encryption algorithms. Developing such system that is based on the E-ART algorithm provides us with a robust and fast system despite dealing with big data.

Medical staff are the main users of our application, the easy-to-use interface provides many actions that can be done by the medical staff such as: saving, deleting, inspecting, and editing files. Meanwhile the system runs the E-ART algorithm with each action.

The system will also include some features that will significantly strengthen it:

- **Backup** – in any case of losing files like deleting by mistake or someone trying to sabotage, every file will be saved in a set of external memory, only security manager can read and restore the files.
- **Data report** – this is a feature dedicated to the security manager that can select two different dates and the system will show data report between these dates, examples of data that will be shown: encryption and decryption speed, memory usage, number of deleted files, and number of added files.

2. Background and Related Work

2.1 Common Encryption Algorithms

In cryptography, encryption is the process of encoding a message or information in a way that only authorized parties can access it and those who are not authorized cannot.

Encryption uses an algorithm to scramble, or encrypt, data and then uses a key for the receiving party to unscramble, or decrypt, the information. The message contained in an encrypted message is referred to as plaintext. In its encrypted, unreadable form it is referred to as cipher text.

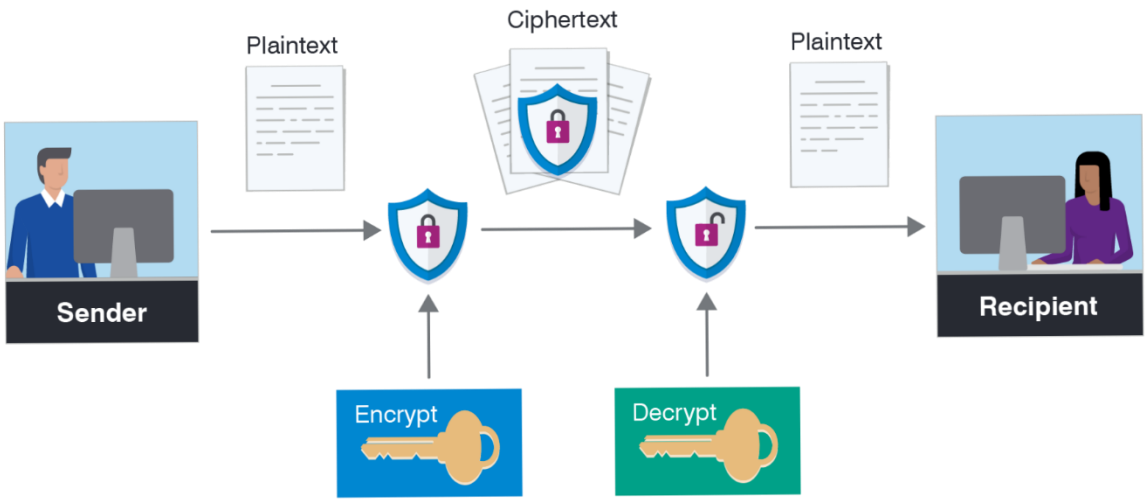


Figure 1 Encryption/Decryption

The two main advantages of the new E-ART algorithm over the commonly used encryption algorithms are the process time and memory usage. We are working with medical institutes, where large-sized data and real time applications are involved. The encryption algorithm needs to be fast, secure, and light in memory usage.

2.1.1 DES:

Data Encryption Standard was defined by IBM researchers originally in the early **1970s**. It was adopted in **1977** by government agencies to protect sensitive data and was officially retired in **2005**.

Block cipher is an encryption method that applies a deterministic algorithm with a symmetric key to encrypt a block of text, rather than encrypting one bit at a time as in stream ciphers.

DES is a block cipher algorithm that encrypts data in blocks of size of **64 bits**, producing **64 bits** of cipher text. The same algorithm and key are used for encryption and decryption with minor

differences. The key length is **56 bits**, which is relatively small, and over the years became vulnerable to brute-force attacks.

Brute force attack is a form of an attack where attacker submits many passphrases with the hope to eventually guess correctly.

2.1.2 AES:

On October 2, 2000, NIST announced that AES is the next standard of encryption algorithm.

AES (Advanced Encryption Standard) is a block cipher that encrypts data in blocks of size of **128 bits**, with three different key lengths of: 128, 192, and 256 bits.

The design and strength of all key lengths of the AES algorithm are sufficient to protect classified information up to the SECRET level. The more classified the data is, the longer key length is used.

2.2 The Proposed Algorithm

2.2.1 E-ART:

In this work, we will use the E-ART "Encryption technique based on ASCII values of Reflection Tree". This is a new algorithm that uses a 128-bit symmetric key, a key that is used both to encrypt and decrypt information, the key will be divided into a static and a dynamic part.

The encryption algorithm will combine a symmetric structure used in the construction of block ciphers called Feistel network, AES with substitution boxes.

First the input file is divided into several equally sized blocks, and then each block is split into plaintext and key parts.

2.2.2 E-ART Structure:

The main novelty that the E-ART algorithm revolves around is the use of the reflection property of a balanced binary tree data structure to enhance data search efficiency. Binary trees can be explained as follow. The root node has a value X. The left subtree contains all values that are smaller than X, while the right subtree contains all values greater than X. Searching for a particular value has the complexity of **O(log(n))**.

Let us explain how the reflection tree works. Consider for example a text containing the character "X", whose ASCII code is 88, This character is on level 4 of the binary tree, we can reach it by going in the path of right left right (1R1L1R). It's reflected character is at (1L1R1L), which is 40, having the ASCII code for the character "(". This technique ensures that any character in the plaintext can be encoded using a reflection tree. Having high search efficiency. We can also compute the initial reflected value as follows:

$$Val_{Initial\,ref} = (Len_{max} - Val_{org}) + 1$$

There are some issues with the initial reflected value:

1) Characters that range from 0 to 32 in the ASCII table are not addressed.
2) It is vulnerable to cryptanalysis attacks, duo to the one-to-one mapping of characters.

To solve these issues, the algorithm proposes the addition of multiple offsets to the initial reflected value. First in the case of non-printable ASCII characters, a constant offset value of 32 is added to the initial reflected value. To prevent a cryptanalysis attack we propose another offset, called variable offset. It is computed based on the E-ART tree's root node R, left node, and right node.

The general equation for the reflected value is

$$Val_{Ref} = \begin{cases} (X \,\%\, Len_{max}) + Offset_{const} & , X > Len_{max} \\ X & , X \leq Len_{max} \end{cases}$$

where $X = Val_{intial\,ref} + Offset_{var} + Offset_{const}$.

The algorithm also uses a dynamic offset based on the characters' position in the plain text. The dynamic offset is generated for each character using a pseudo-random generation with the character's position as the seed. This adds more randomness between the plaintext and the cipher text.

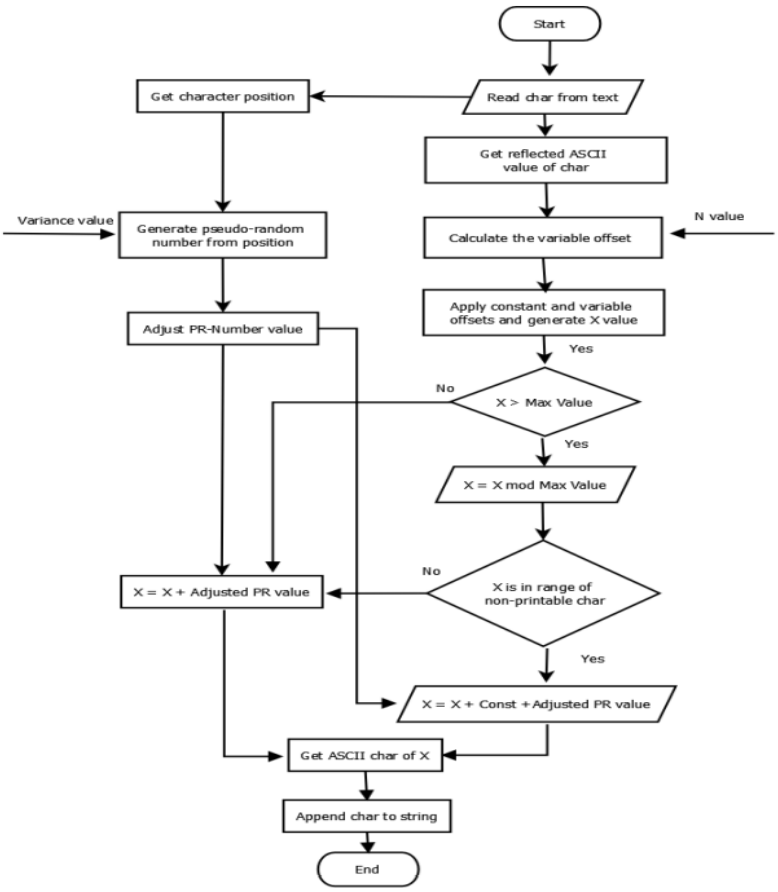


Figure 2 E-ART Structure

2.2.3 Key Derivation:

The key derivation process is simple, two offsets which are constructed from the initial key, these offsets are used during the encryption process. The first offset called the variable offset remains static during both the encryption and decryption; the second offset is a dynamic one that changes with each character value.

2.2.4 Initial Key:

The initial key is a secret key shared between legal entities, it consists of two values, N and Variance, both are used to generate the offsets. N represents an integer value composed of 64 or 128 bits; this value will be the parameter for calculating the variable offset. Variable offset is calculated mathematically using the proposed tree properties.

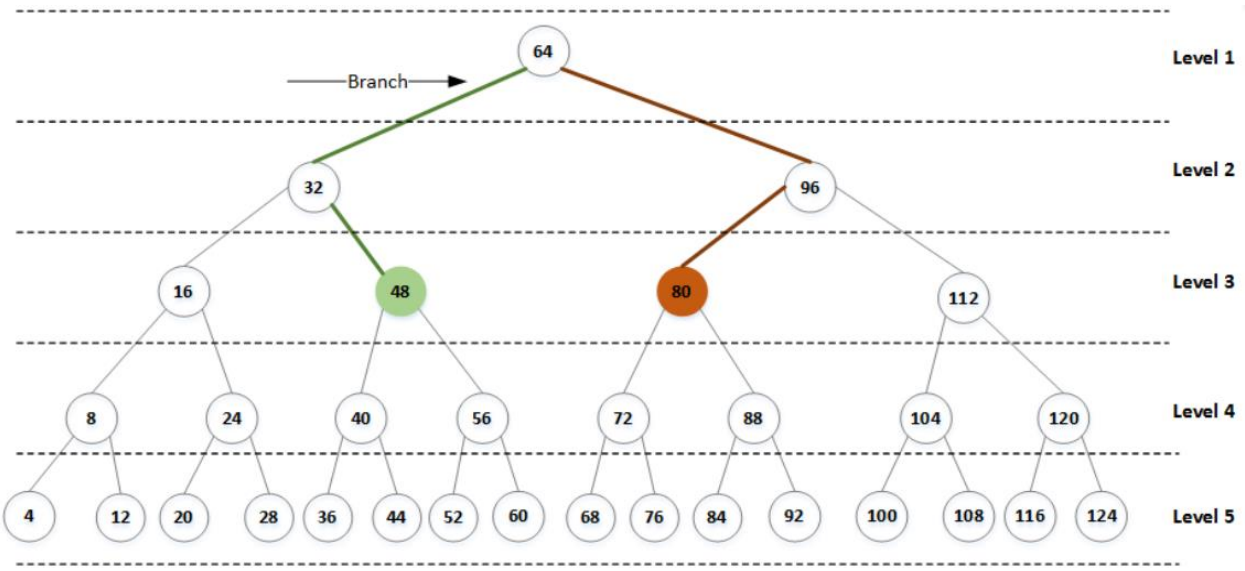


Figure 3 Proposed Tree

Looking at the figure above we can see that each character is represented as a node, and each node has its reflection node.

N(L) is calculated using the N value derived from the initial key.

N(R) is the reflection node of N(L).

Offset(var) will be calculated as follows:

$$N_L = N \bmod Len_{max}$$

$$N_R = (Len_{max} - N_L) + 1$$

$$Offset_{var} = \begin{cases} \mathrm{R} \times N_L \bmod N_R & \text{if } N_L < \mathrm{Root} \\ \mathrm{R} \times N_R \bmod N_L & \text{if } N_L > \mathrm{Root}. \end{cases}$$

This encryption transformation adds more complexity and prevents cryptanalysis attacks that take advantage of one-to-one mapping that is simply mapping distinct elements to distinct elements.

A pseudo-random number and the second part of the initial key (Variance) are both used to automatically produce the Dynamic offset. Each character's position in the text is used as a seed to generate a pseudo-random number of 64 or 128 bits. The pseudo-random number is then adjusted using the Variance value.

$$\text{Dynamic offset} = (\text{Pseudo}) \bmod \text{Variance}.$$

This offset is added in the last step to produce the final encrypted characters and is changed for each character.

2.3 Experimental Evaluation

Every strong encryption algorithm must satisfy some essential security criteria. We will evaluate E-ART's performance and efficiency by checking some of the following criteria:

- The avalanche effect
- Frequency analysis
- Performance analysis
- Randomness verification

2.3.1 Avalanche Effect:

High-quality ciphers satisfy a desirable property of cryptographic algorithms, called the "Avalanche Effect", it simply means that a slight change in the input will cause a significant change in the output.

The hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are different. In our work the avalanche effect is measured using the hamming distance as follows:

$$AvalancheEffect = \frac{HammingDistance}{TotalNoCharacters} \times 100$$

We will compare the avalanche effect between E-ART, AES, and DES.

2.3.2 Frequency Analysis:

In cryptanalysis, frequency analysis is similar to counting letters, simply how many times each letter appear in both the cipher text and the plain text. Attackers can use frequency analysis to extract and obtain the key or the plaintext. This type of attack is called statistical attack.

We will generate two histograms. One for the plain text and another for the cipher text, the goal is to have a uniformly distributed cipher text histogram.

2.3.3 Performance Analysis:

Top tier encryption algorithms have a certain balance between processing time and encryption strength. In the case of dealing with big data, we aspire to have a decent processing time.

We will calculate the processing time for different sizes of files (ranging from 200KB to 2000KB) and will compare the results with two well-known symmetric encryption algorithms AES-128 and DES.

2.3.4 Randomness Verification:

In our project we will use the NIST () suit, one of the most frequently used test batteries. The NIST Statistical Test Suite is a statistical package consisting of 15 cryptographic random number generators. For our experiment, we selected five tests for testing the randomness and non-uniformity attributes of our dynamic key. The selection was based on the recommendation of input size stated by NIST user's guide. Our dynamic offset will be restricted to 128 bits; thus, the following test are applicable to our key length:

- Frequency test
- Block frequency test
- Runs test
- Cumulative sums forward test
- Cumulative sums backward test

3. Expected Achievements

The main purpose of our product is to reach a high level of security when dealing with big data and in addition to have a good encryption time.

We are looking to evaluate the success of our product, by applying a variety of tests, that will give us a precise evaluation of every aspect of the product.

What's unique in our product is the easy-to-use application interface, The main users of the application are medical staff, that don't have the highest level of dealing with technology, thus having an easy-to-use application with a friendly interface is a must.

4. Research / Engineering Process

4.1 Process

4.1.1 Stages of the development process

- Gathering information and reading about other commonly used encryption algorithms, pros, and cons.
- Selecting the proper tools for implementing the system.
- Implementing the new E-ART algorithm.
- Preliminary design and visualization of the system.
- Formulation of the architecture and advanced design of the system.

4.1.2 Development process challenges

Challenges relevant to the project – includes collecting dataset, implementing efficient algorithm, running efficient testing.

Dataset collecting challenge

Our system targets medical institutes, thus we are dealing with big medical data, collecting this kind of data might be a little bit challenging, since medical data is sensitive.

Efficient algorithm challenge

The algorithm implementation might have bugs, it is very hard to examine the encryption algorithm, it needs time, a lot of data, and most importantly experience.

Efficient testing challenge

Most of the tests are already planned, but we have the concerns if these tests are enough? Are the results good enough?

4.1.3 Software Development Methodology

After examining the most common development methodologies, we found that the best fitting methodology was the **Agile Software Development**.

Reasons for choosing Agile:

- We want a good interaction between developing and testing.
- Direct communication and constant feedback that will leave no space for any guesswork in the system.
- Agile methodology has an adaptive approach that can respond to real time changing requirements.

Figure 4 Agile Software Development

4.1.4 GitHub:

GitHub is the perfect platform for us to use in the implementation phase, we are planning to use **GitHub** to stay tuned with our work, **GitHub** also provides backup feature, we can save our work in versions, making it easier for us to access previous versions.

We can also interact with the **GitHub** community for new ideas, the **GitHub** community also provides assistance in coding and bugs.

4.2. Product

4.2.1 Project Architecture:

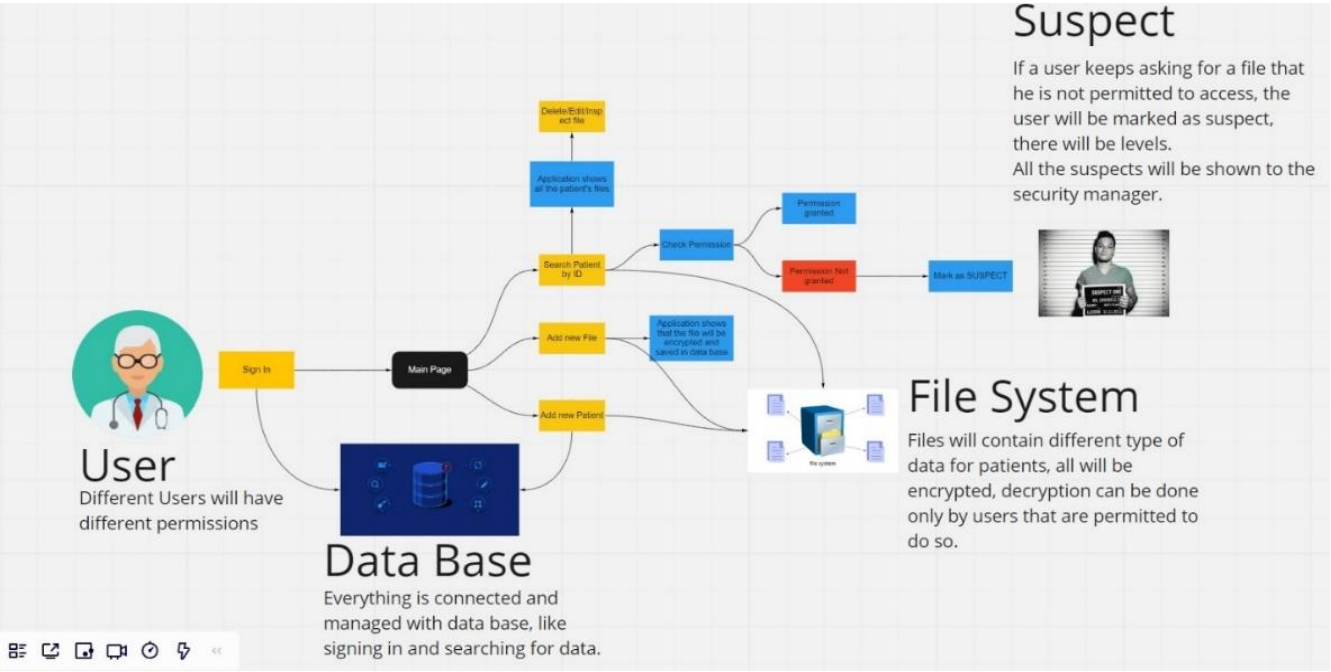


Figure 5 Project Architecture

Our system connects the model to the data base and the file system:

- **The model** enables the user to sign in, add new patient, add a new file, search for a file. Encryption-decryption will be done to save or display files.

- Verifying a user when signing in, adding new patient, and searching for data is managed with the **data base**.

- Adding, deleting, inspecting, and editing files are managed by a class that saves all these changes to the **file system**.

4.2.2. Use Case:

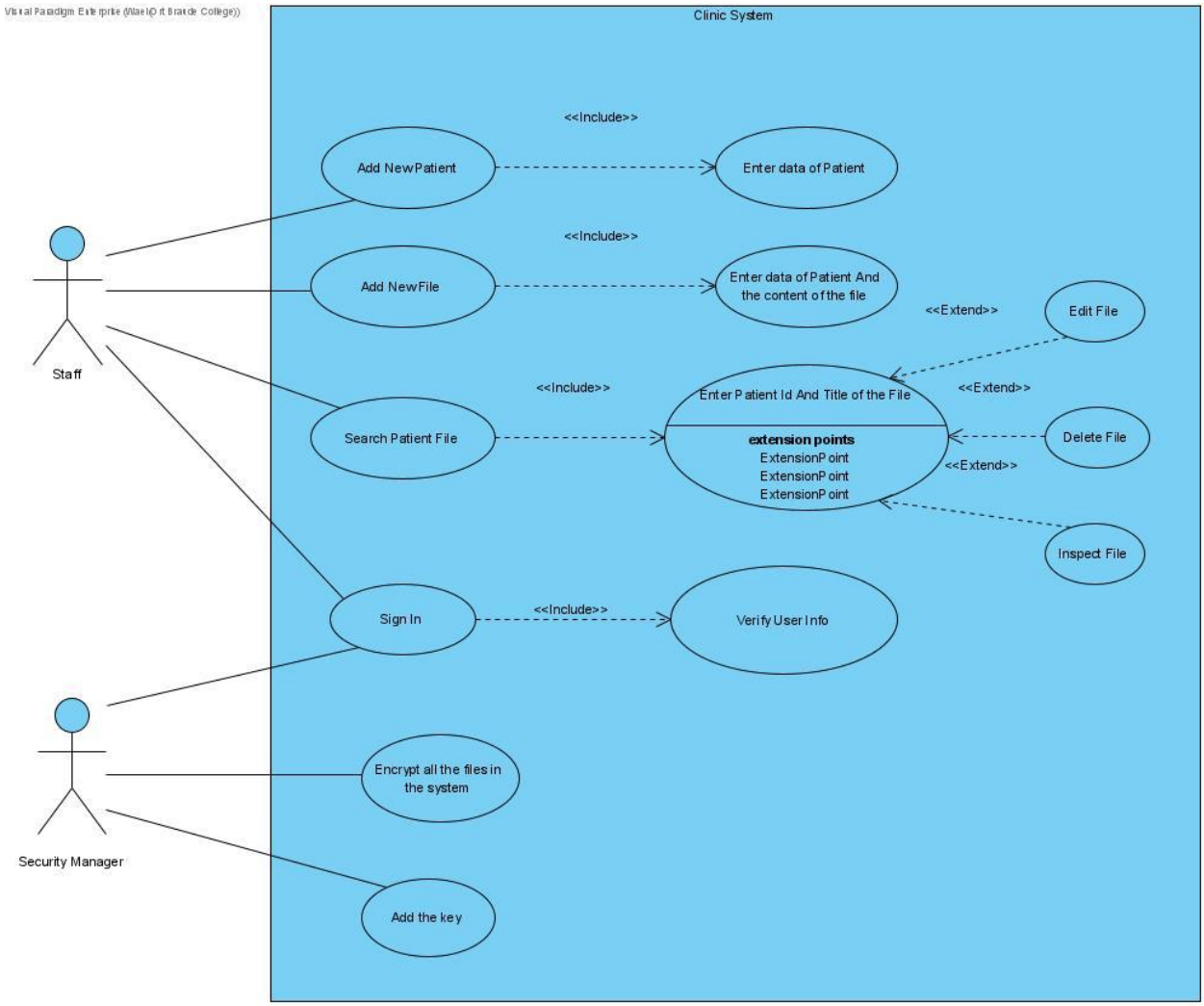


Figure 6 Use Case Diagram

Sign in:

Done by the user (Doctors and Manager) by inserting username and password.

Verify user info:

The system automatically verifies the user's data.

Add new patient:

Staff Adding new patient by inserting the appropriate data.

Add new file:

Doctor can add a new file by inserting the information of the file.

Search patient's file:

Doctor can search patient's file by entering the patient's id and the title of the wanted file, after that the system will display the file and doctor can edit, delete, or just inspect.

Encrypt all the files in the system:

When installing the application to a new institute, security manager can encrypt all the existing files that are not encrypted.

Adding the key:

Security Manager needs to add the key that is used in the encryption algorithm, he can also change the key at any time.

4.2.3. Manager Activity Diagram:

When installing the application to a new institute, security manager can encrypt all the existing files that are not encrypted.

First, he needs to log in, and then he can select the directory that contains the files, the system will then begin encrypting all the files one by one.

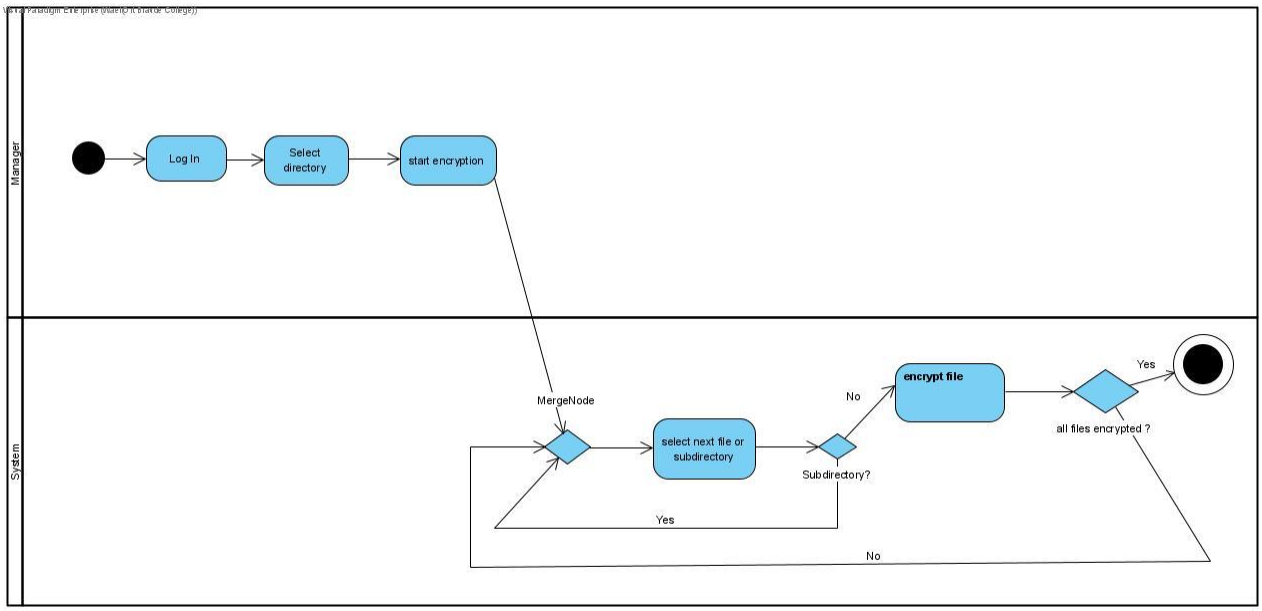


Figure 7 Manager Activity Diagram

4.2.4. Edit Activity Diagram:

Staff can search and edit files.

First step is logging in, then the doctor can choose to search patient by id, then he can choose the file that he wants to edit.

In order to display the file, the system will decrypt it, and after the doctor is done editing and saving the file, the system will encrypt the file.

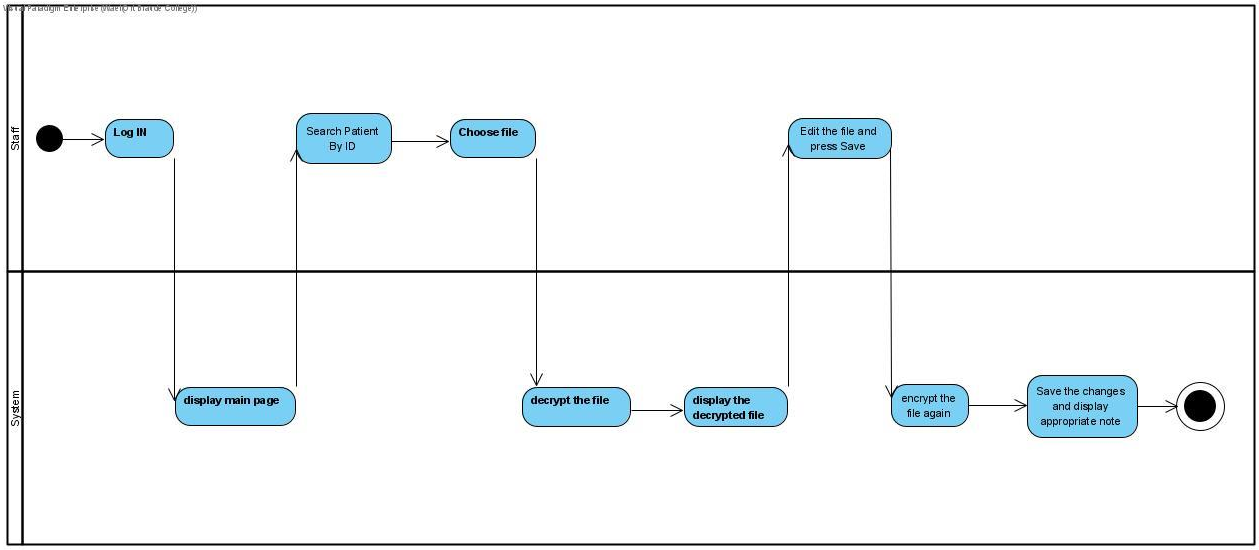


Figure 8 Edit Activity Diagram

4.2.5. Class Diagram:

The class diagram includes two parts:

Model part:

We have the E-ART class that includes the encryption-decryption algorithm that is used to encrypt and decrypt the targeted information.

Searching for files is done with MySearch class.

Reading, writing, and updating files are done with FileSystem class.

The User class includs the Staff and the Security manager, along with patient.

Both the user and patient inherit the person class.

GUI part:

Our main pages are the following:

Welcome page – where the user can choose the appropriate sign in.

Sign in page – where the user can enter usernam and password.

Main page – where all the main activities will be displayed.

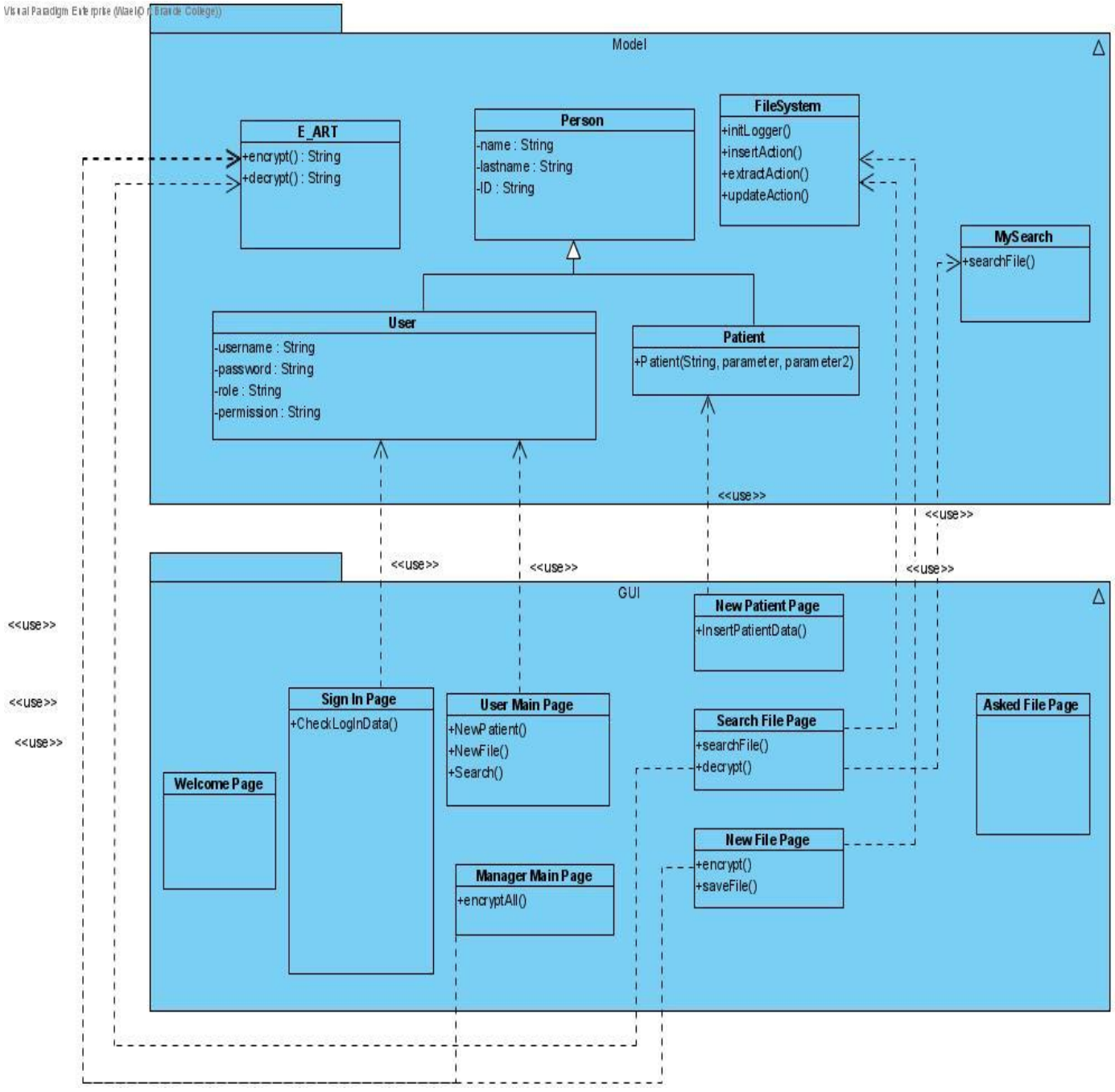
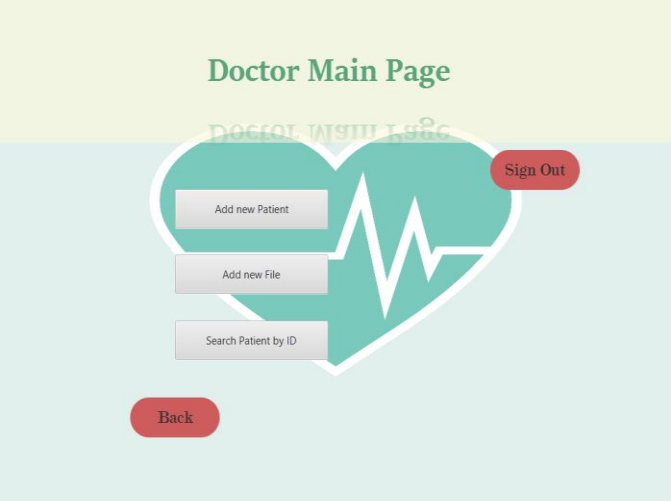
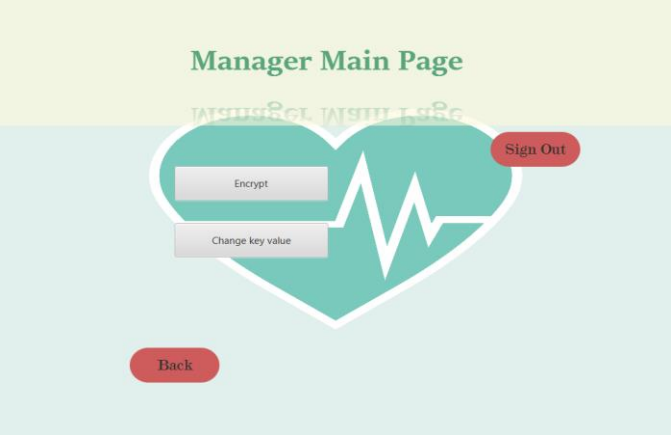


Figure 9 Class Diagram

4.2.6 User Interface:





5. Evaluation/Verification Plan

5.1. System Requirements

	Type	Requirement
1	FR	System will authenticate the staff and manager login.
2	FR	System will enable the doctor to add a new patient.
3	FR	System will enable the doctor to add a new file.
4	FR	System will enable the doctor to search for files.
5	FR	System will display the patient's data on the screen.
6	FR	System will enable doctor to delete a certain file.
7	FR	System will enable doctor to edit a certain file.
8	FR	System will enable the manager to encrypt all the files in a specific directory.
9	FR	System will enable the manager to add or change the key of encryption and decryption.
10	FR	System will display "ERROR" window with appropriate note in each illegal activity. (For example, if a user doesn't fill all the Required fields the system display "You Must Fill All the Fields")
11	FR	System will display "CONFIRMATION" window with appropriate note in each update done. (Like if a doctor adds a new patient the system displays "Patient Data Saved")
12	FR	The encryption algorithm (E-ART) must satisfy the property "Avalanche Effect".
13	FR	The encryption algorithm (E-ART) must be secure against "Frequency Analysis".
14	FR	The encryption algorithm (E-ART) must have a certain balance between processing time and strength (Performance Analysis).
15	FR	The encryption algorithm (E-ART) must succeed five selected tests from the NIST suit "Randomness Verification".

5.2. Testing

Test ID	Description	Expected result	Actual result
1.a	User enters correct username and password.	System recognizes the User's data and opens main page.	
1.b	User enters incorrect username or incorrect password.	User is not recognized; system will display error window.	
2.a	Doctors insert new patient data to add him to the system.	Adding this patient to the system, system will display confirmation window.	
2.b	Doctors insert existing data of patient to add him to the system.	System will display error window.	
4.a	Doctor with permission searches patient's Files.	System will display the file.	
4.b	Doctor without permission searches patient's Files.	System will display error window.	
12	We will compare the "Avalanche Effect" between E-ART, AES, and DES.	E-ART will give better results (Bigger avalanche effect).	
13	We will compare the results of " Frequency Analysis " between E-ART, AES, and DES.	E-ART will have a uniformly distributed cipher text histogram.	

14	We will calculate the processing time for different sizes of files and will compare the results with two well-knows symmetric encryption algorithms AES-128 and DES.	E-ART will have better processing time.	
15	we will test the E-ART algorithm with five selected tests from the NIST () suit.	E-ART will pass all these tests.	

6. Conclusion

Digital data security is growing rapidly over the last few years, it is an interesting subject once you are invested in it. After exploring the internet, we finally found the E-ART algorithm, we were convinced and decided to continue with it.

With all the knowledge, skills, and the experience that we gained over the years, we were able to visualize both stages of the engineering process. The first stage was very essential for the second one, it required a lot of preparation and thinking.

To successfully finish the first stage, we had to dive deep into the internet, search and read a lot of interesting articles that helped us fully understand the small concepts of our project. After all the big winner here is us, as we gained a lot of knowledge and experience, especially learning how to prepare for a big project from scratch.

We believe that our project will be a strong security system, especially for big data systems, the system is expected to be fast and light on resources. Overall, we want a system that can be used in the modern world, so many features will be added that can strengthen the security of the system.

7. References

1- Abdullah, Beloff and white 2021, *E-ART: a new encryption algorithm based on the reflection of binary search tree*, University of Sussex. http://sro.sussex.ac.uk/id/eprint/96822/.

2- Thakkar, Jay 2020, *DES vs AES: Everything to know about AES 256 and DES Encryption*, Infosec Insights. https://sectigostore.com/blog/des-vs-aes-everything-to-know-about-aes-256-and-des-encryption/.

3- Ohri, Ajay 2021, *Symmetric and Asymmetric Key Cryptography: A Detailed Guide In 2021*, Jigsaw. https://www.jigsawacademy.com/blogs/cyber-security/symmetric-and-asymmetric-key-cryptography.

4- Kazarian, Jason Paul 2021, *Securing Big Data in the IoT Age: Why Dynamic Key Management is Key*, Teach Beacon. https://techbeacon.com/security/securing-big-data-iot-age-why-dynamic-key-management-key

5- GeeksforGeeks 2020, *Avalanche Effect in Cryptography*. https://www.geeksforgeeks.org/avalanche-effect-in-cryptography/.

6- Totorialspoint 2020, *Block Cipher*. https://www.tutorialspoint.com/cryptography/block_cipher.htm.