

Software Design Specification - Image Encryption

Karan Pittie (13BCE0360), Balaji Shanmugam (13BCE0821)

February 2016

Contents

1	Introduction	3
1.1	Goals and Objectives	3
1.2	General Statement of Scope	3
1.2.1	General Requirements	3
1.3	Major Constraints	3
2	Data Design	5
2.1	Database Description	5
3	Architectural Level Design	6
3.1	Program Sturcture	6
3.1.1	Overall	6
3.1.2	Encryption	7
3.1.3	Decryption	7
3.2	Description for Components	7
3.2.1	Encryption	7
3.2.2	Decryption	8
4	User Interface Design	9
4.1	Description of the User Interface	9
4.1.1	Screen Images	9
4.1.2	Panels	11
4.1.3	Triggers and Events	13
4.2	Interface Design Rules	13
5	Restrictions, Limitations and Constraints	13
6	Testing Issues	14
6.1	Classes of Test	14
7	Appendices	15

1 Introduction

This section describes the design for the Image Encryption System using the MATLAB tool by MathWorks.

1.1 Goals and Objectives

The main purpose of this system is to enable and increase the security standards of the image encryption systems. Its main aim is to encrypt the image in a form such that only the person for whom the image is intended to be seen can only be able to access the image.

- Encrypt the image using the El Gamal encryption algorithm.
- Decrypt the image using the El Gamal encryption algorithm.

1.2 General Statement of Scope

1.2.1 General Requirements

- To provide an efficient algorithm for image encryption.
- To increase the security standards of image encryption.
- To provide the users with a simple and easy way to transmit the images securely over the internet.
- To provide the users with a simple and easy way to read and decrypt the image using the decryption algorithm.
- To provide the users to work with an efficient algorithm and modify and improvise it and use it in further research.
- Help in the form of documentation shall be attached to the system as a readme file which the users can go through and look up for any help that they require to use the software.

1.3 Major Constraints

The major constraints that will be faced during the execution and implementation of this project would be as follows.

Time

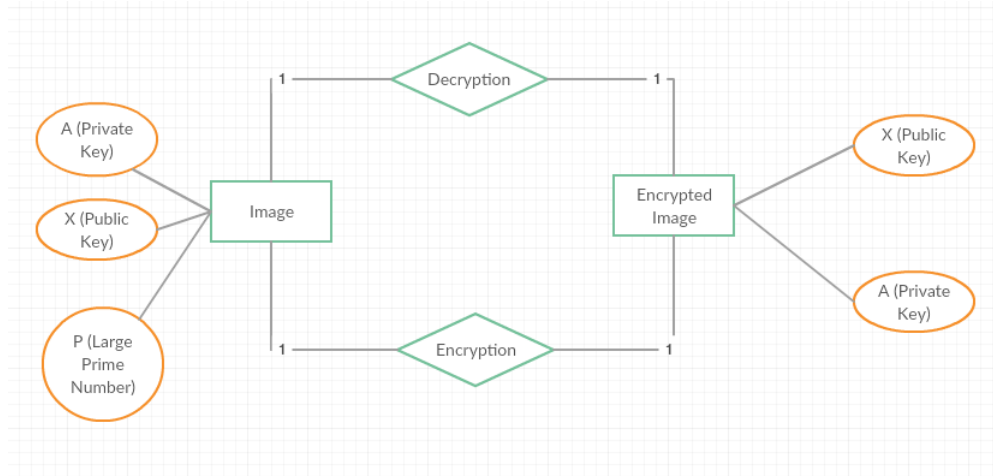
Due to the limited time available, there might be issues that may arise when the coding and the implementation part of the project starts. The algorithm is huge and complicated and will require time to be analysed and coded as per the current industry requirement.

Resources

There are very few people working on this project and there is a large amount of work that needs to be done in order to complete the project on time. Additionally the resources that are required to complete the project require paid licences from the system and hence financial aid would be required in order to take the project to a hundred percent completion stage.

2 Data Design

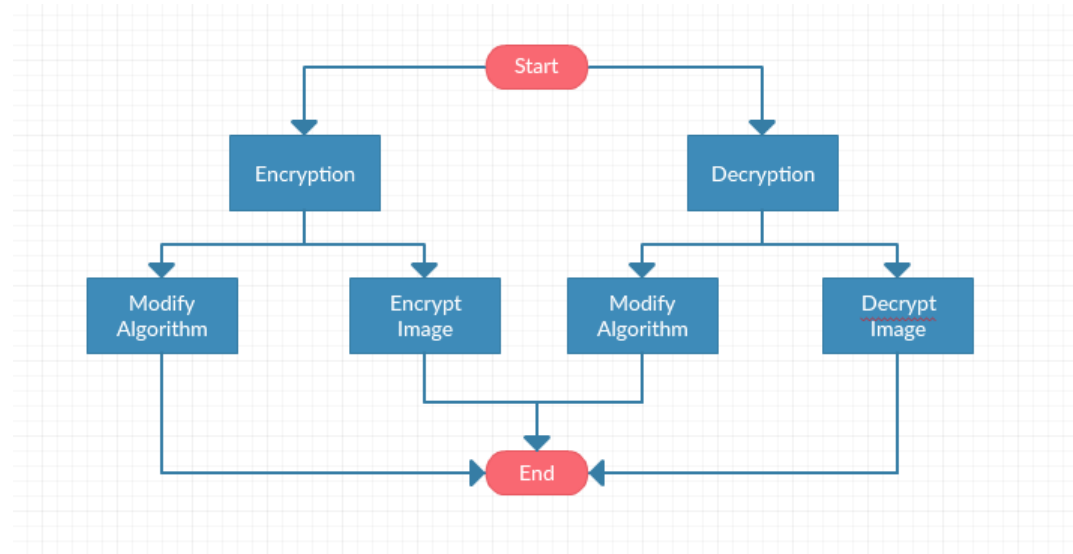
2.1 Database Description



3 Architectural Level Design

3.1 Program Sturucture

3.1.1 Overall



Menu Items

The following shows the architecture of the main menu:

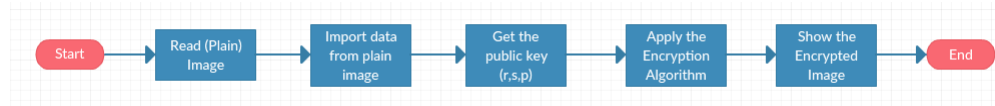
Encryption:

- Modify the Encryption algorithm
- Encrypt the image using the encryption algorithm

Decryption:

- Modify the Decryption Algorithm
- Decrypt the image using the decryption algorithm

3.1.2 Encryption



3.1.3 Decryption



3.2 Description for Components

3.2.1 Encryption

- Step1: Start
- Step2: Read the plain image into its corresponding matrix (call it M) using MATLAB such that each element m_{ij} in M does not exceed the prime number p.
- Step3: Let the public key (r, s, p).
- Step 4: For all the element m_{ij} in the matrix, select one random integer k with, $1 \leq k \leq p-2$.
- Step 5: Compute $X \equiv r \cdot k \pmod{p}$.
- Step 6: Compute $y_{ij} \equiv (m_{ij} * s \cdot k) \pmod{p}$
- Step 7: Show the encrypted image Y.

3.2.2 Decryption

To decrypt the cipher image, the private key a and X are necessary to be known by the receiver.

The process as the following (done by MATLAB):

- Step1: Import to data (Y) from the encrypted image.
- Step 2: Restore the plain image M , such that $M \equiv [Y ((X) a) - 1] \pmod{p}$
- Step3: Obtain the original image (decrypted image).

4 User Interface Design

There will be about 3 interfaces in the program. We can't design on the exact number of it yet, because there might be a few things needed after the final implementation has been done.

4.1 Description of the User Interface

After the program or the system is started then the program fires up and takes you to a main home page with the functions of MATLAB on the left pane and a command line GUI at the bottom of the screen.

4.1.1 Screen Images

Main screen

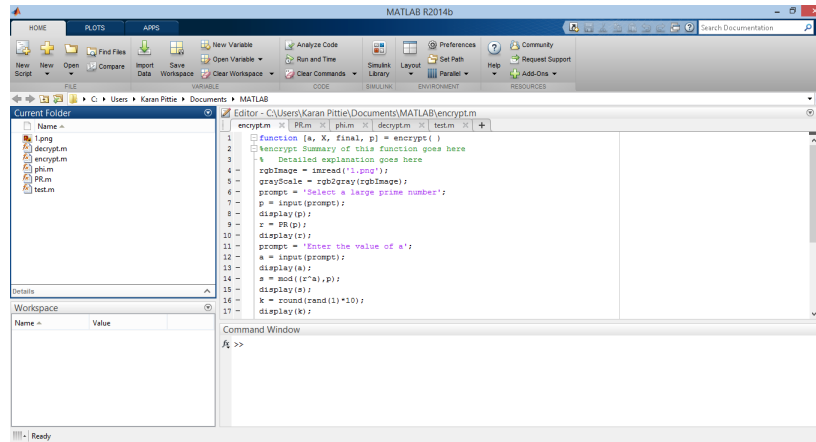
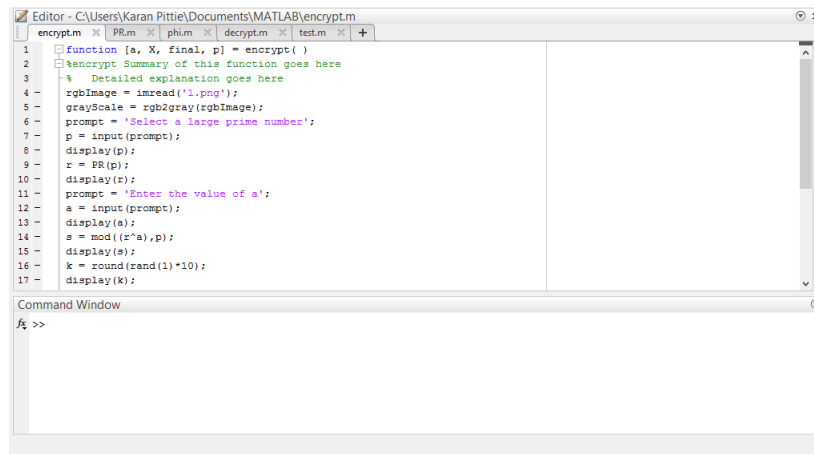


Figure 1: This will be the main User Interface of the system

Encryption screen



```

1 function [a, X, final, p] = encrypt( )
2 %encrypt Summary of this function goes here
3 % Detailed explanation goes here
4
5 rgbImage = imread('1.png');
6 grayScale = rgb2gray(rgbImage);
7 prompt = 'Select a large prime number';
8 p = input(prompt);
9 display(p);
10 r = PR(p);
11 display(r);
12 prompt = 'Enter the value of a';
13 a = input(prompt);
14 display(a);
15 s = mod((r^a),p);
16 display(s);
17 k = round(rand(1)*10);
18 display(k);

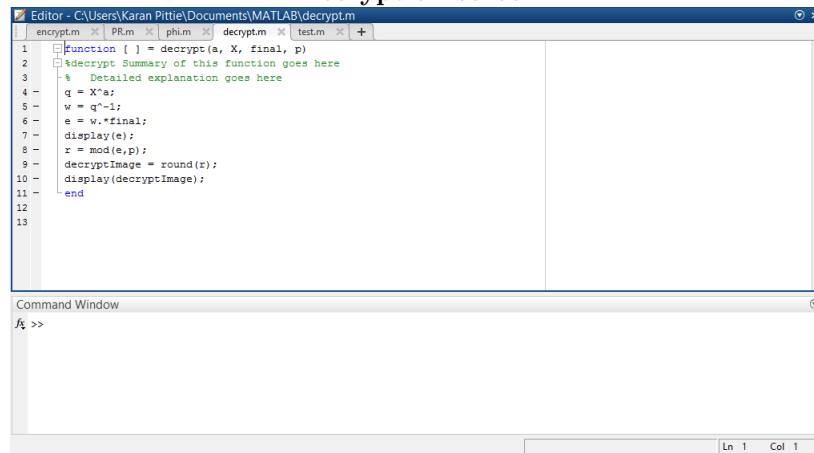
```

Command Window

`>>`

Figure 2: This will be the screen where the user can encrypt the image and also can make necessary modifications to the algorithm as per convenience

Decryption screen



```

1 function [ ] = decrypt(a, X, final, p)
2 %decrypt Summary of this function goes here
3 % Detailed explanation goes here
4
5 q = X^a;
6 w = q^-1;
7 e = w.*final;
8 display(e);
9 x = mod(e,p);
10 decryptImage = round(x);
11 display(decryptImage);
12
13 end

```

Command Window

`>>`

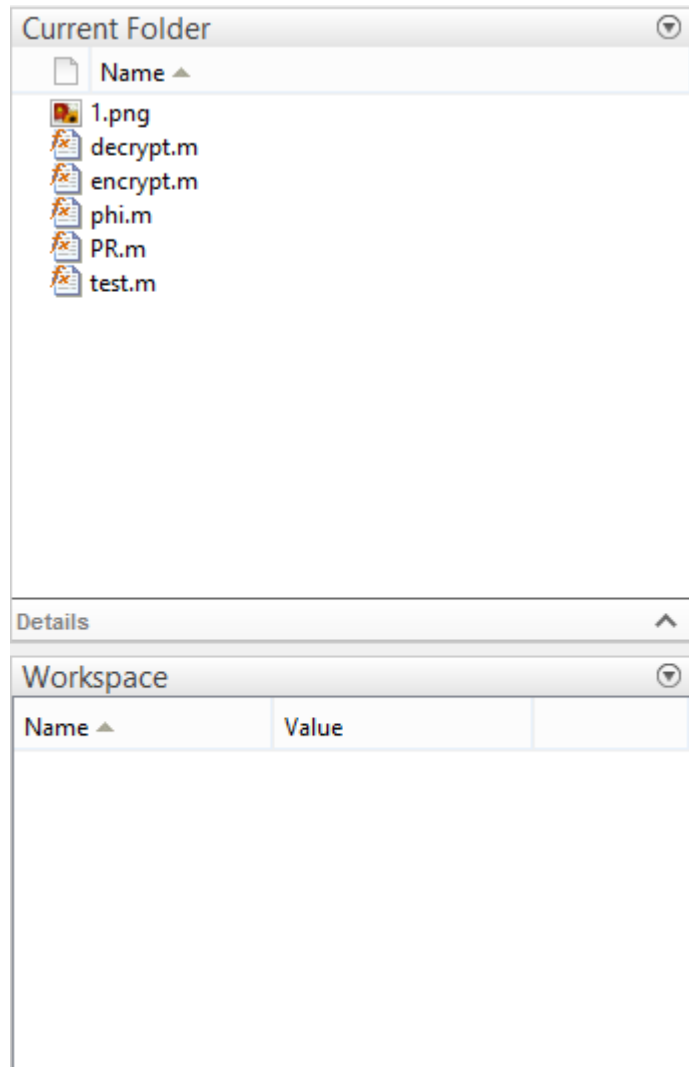
Ln 1 Col 1

Figure 3: This will be the screen where the user can decrypt the image and also can make necessary modifications to the algorithm as per convenience

4.1.2 Panels

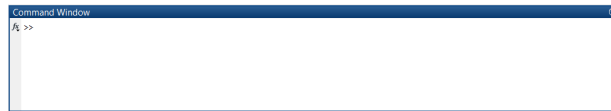
Functions Window

This is the panel that will list out all the functions that are available for the user to work with and use for implementing the functions that each one designed to perform.



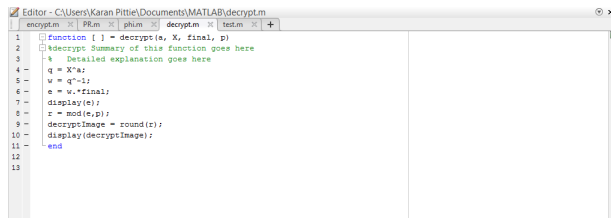
Command Window

This is the panel where the user can type in all his commands and the actions that he wants to perform with respect to the data that he has collected or wants to work with.



Coding Window

This is the panel where the user can see the code that is been written for the user, and also which will be the logic behind the implementation of the encryption and the decryption algorithm schemes.



4.1.3 Triggers and Events

Command line command

This will trigger the functions to be implemented in the system and after accepting appropriate inputs with the required criteria to be followed the encryption system will proceed with the encryption process. Similarly, if in the command line the decrypt function has been triggered then the decrypt function will be triggered and after accepting proper input from the user and working with the data supplied to the function the algorithmic logic will be applied to the image and the decrypted image will be rendered as the final output.

Mouse click on function

This will trigger the system to output or show the logic or the algorithm that has been used to process the images. It can be used by the professors and the experts in the field to work on and implement and change the algorithm and improvise the algorithm the way in which they seem fit.

4.2 Interface Design Rules

Interface design focuses on these areas of concern:

1. The design of interfaces between software modules
2. The design of interfaces between the software and other nonhuman producers and consumers of information (i.e., other external entities)
3. Easy to Learn
4. Readability
5. Easy navigate between interfaces

5 Restrictions, Limitations and Constraints

Time

Due to the limited amount of time available for the development phase of the project there are many parts of the projects that have been ignored so that at least the implementation of the algorithm can be completed to the full.

Skills limitation

The people working on this project have very little knowledge on the software that is being used to implement the algorithm and hence the whole process of building a system will be a challenging task as the contributors need to understand first the working of the software being used to code the algorithm and then implement the algorithm.

Insufficient Resources

The resources that are required to take this project to completion are not available at hand for usage. The software itself that is being used to implement the algorithm is a paid one and the one version that is currently being used to develop the system is just a trial version.

6 Testing Issues

To validate the software we need to test the software. During the testing we will be concerned about the inputs and their expected outputs. We emphasize on the testing where we will input the data and will compare the output with the expected results. At this stage, we are not concerned about the process; we are only looking for the correct outputs.

6.1 Classes of Test

The software has many functions in it. We will go through each of the software function to describe different types of test performed on them.

Encryption Function

The whole encryption process will be the implementation of the El Gamal encryption algorithm, which is being attached to the end of this document for reference. We have tried and tested the software for Black and white images and the algorithm seems to encrypt them as per the El Gamal encryption scheme. All the expected results and keys that need to be formed as a result of the encryption process are being generated but we are unsure if the results that are obtained by the encryption process are the ones that we should be getting as a result of the encryption scheme so we are currently working on that part. Although for RGB images, the image matrix is a three dimensional matrix and hence we have to reduce the dimensionality of the matrix to two dimensions in order to perform the encryption algorithm that we currently have coded without which the system returns a garbage value.

Decryption Function

The decryption process that has currently been coded by us is failing even for the two dimensional images that we have encrypted using the El Gamal encryption scheme. It returns a random value and we are currently working on why that error has been happening.

7 Appendices

El Gamal encryption consists of three components: the key generator, the encryption algorithm, and the decryption algorithm.

Key generation

The key generator works as follows:

- Alice generates an efficient description of a cyclic group G of order q with generator g . See below for a discussion on the required properties of this group.
- Alice chooses an x randomly from $\{1, \dots, q-1\}$.
- Alice computes $h := g^x$.
- Alice publishes h , along with the description of G, q, g , as her **public key**. Alice retains x as her **private key**, which must be kept secret.

Encryption

The encryption algorithm works as follows: to encrypt a message m to Alice under her public key (G, q, g, h) ,

- Bob chooses a random y from $\{1, \dots, q-1\}$, and then calculates $c_1 := g^y$.
- Bob calculates the shared secret $s := h^y$.
- Bob maps his secret message m onto an element m' of G .
- Bob calculates $c_2 := m' \cdot s$.
- Bob sends the cipher text $(c_1, c_2) = (g^y, m' \cdot h^y) = (g^y, m' \cdot (g^x)^y)$ to Alice.

Note that one can easily find h^y if one knows m' . Therefore, a new y is generated for every message to improve security. For this reason, y is also called an ephemeral key.

Decryption

The decryption algorithm works as follows: to decrypt a cipher text (c_1, c_2) with her private key x ,

- Alice calculates the shared secret $s := c_1^x$.
- And then computes $m' := c_2 \cdot s^{-1}$ which she then converts back into the plaintext message m , where s^{-1} is the inverse of s in the group G . (E.g. modular multiplicative inverse if G is a subgroup of a multiplicative group of integers modulo n).

The decryption algorithm produces the intended message, since

$$c_2 \cdot s^{-1} = m' \cdot h^y \cdot (g^{xy})^{-1} = m' \cdot g^{xy} \cdot g^{-xy} = m'.$$