# Chapter 1

# System Design

System Design describes the system architecture which includes the overall structure of the proposed system. The E-R diagrams, data ow diagrams, UML diagrams play an important role in the system designing of the project.

Section 4.1 describes the system architecture of the proposed system. E-R Diagrams are discussed in Section 4.2. Section 4.3 describes the data flow diagrams. Interface design is discussed in Section 4.4. Section 4.5 describes the UML diagrams. Summary is described in the Section 4.6.

## 1.1 System Architecture

Software architecture is the development work product that gives the highest return on investment with respect to quality, schedule and cost. Software architecture alludes to the overall structure of the software and the ways in which that structure provides concep- tual integrity for a system. In its simplest form, architecture is the hierarchical structure of program components (modules), the manner in which these components interact and the structure of data that are used by the components the architectural design representation defines the components of a system (e.g., modules, objects, filters) and the manner in which those components are packaged and interact with one another. The architectural design description should address how the design architecture achieves requirements for performance, capacity, reliability, security, adaptability, and other system characteristics. The architecture of the proposed system is shown in the Figure 4.1.

## 1.2 E-R Diagrams

The entity relationship data model is based on a perception of a real world that consist of a collection of basic objects called entities, and relation among these objects.

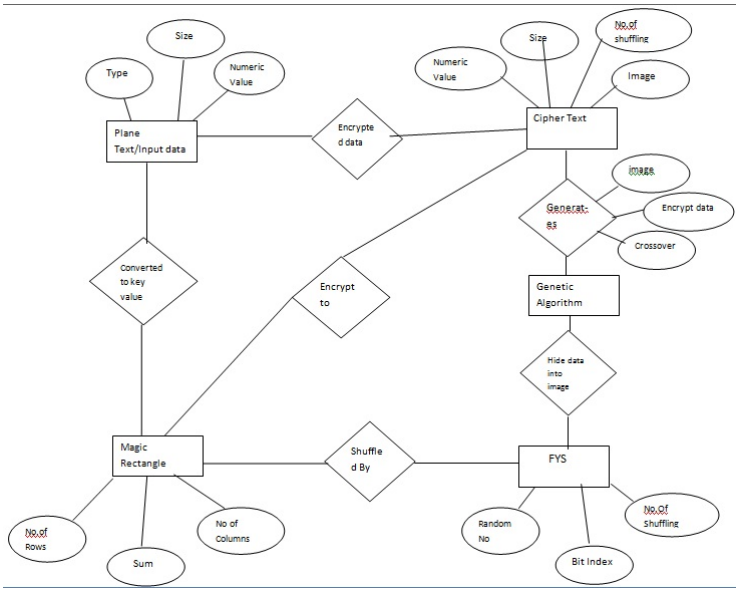


Figure 1: Usecase Diagram

Figure 4.2 shows the E-R Diagram for the Proposed System. E-R Diagram for Server is depicted in Figure 4.3. Figure 4.4 shows the E-R Diagram for Client.

## 1.3 Data Flow Diagrams

A DFD is a graphical technique that depicts the information ow and the transformation that we have applied as the data moves from input to output. A data ow diagram may be used to represent a system or software at any level of abstraction. The data flow diagram can be completed using only four simple notations i.e. special symbols or icons and the annotation that with a specific system.

Data flow diagrams are the basic building blocks that define the flow of data in a system to the particular destination and difference in the flow when any transformation happens. The data flow diagram serves two purposes:

- To provide an indication of how data are transform as the moves through the system.
- To depict the function that transforms the data flow.

The level 0 DFD of the system is shown in Figure 4.5. Figure 4.6 shows the level 1 DFD of the system. The level 2 DFD of the system is shown in Figure 4.7.

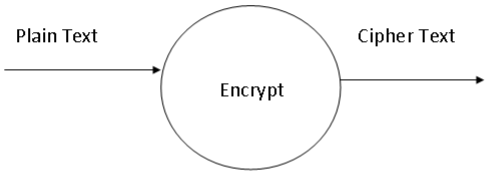


Figure 2: Data Flow Diagram(DFD0)

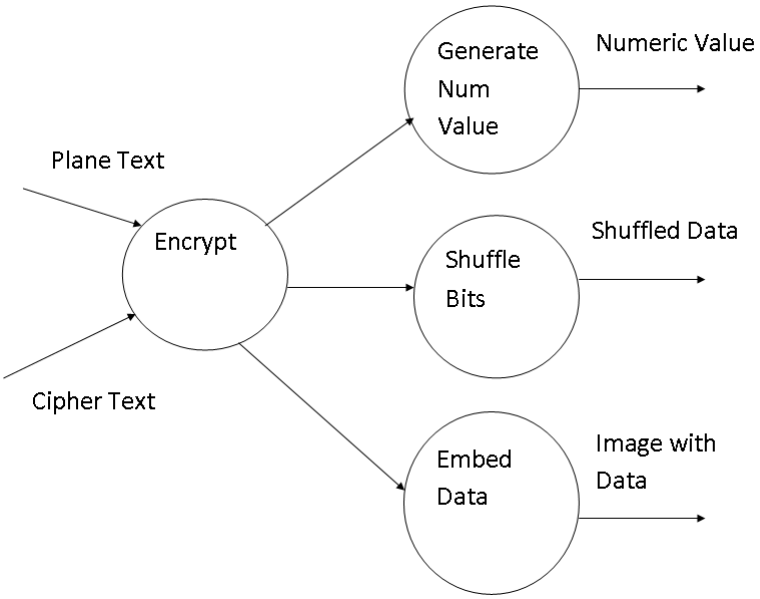


Figure 3: Data Flow Diagram(DFD1)

## 1.4 Interface Design

Interface design focuses on three areas of concern:

- The design of interfaces between software components.

- The design of interfaces between the software and other non human producers and consumers of information (i.e., other external entities).
- The design of the interface between a human (i.e., the user) and the computer.

### 1.4.1 Introduction to Interface Design

User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

### 1.4.2 Component of Interface Design

A software engineer designs the user interface by applying an iterative process that draws on predefined design principles.

### 1.4.3 Need of Interface Design

If software is difficult to use, if it forces you into mistakes, or if it frustrates your efforts to accomplish your goals, you wont like it, regardless of the computational power it exhibits or the functionality it offers. Because it molds a users perception of the software, the interface has to be right.

### 1.4.4 The Golden Rules

- Place the user in control.
- Reduce the users memory load.
- Make the interface consistent.

These golden rules actually form the basis for a set of user interface design principles that guide this important software design activity

## 1.5 UML Diagrams

The UML is a language for

- ***Visualizing***-The structures which are transient can be represented using the UML.

- ***Specifying***-The UML addresses the specification of all the important analysis,design and implementation decisions that must be made in developing & deploying a software-intensive system.
- ***Constructing***-The UML is not a visual programming language,but its models can be directly connected to a variety of programming languages.
- ***Documenting***-The UML addresses the documentation of a system's architecture and all of its details. The various stuctural and behaivioural diagrams are discussed in the chapter.

### 1.5.1 Use Case Diagrams

A Use case diagram shows a set of use cases and actors and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. The Use Case diagram of the proposed system is shown in Figure 4.8.


Figure 4: Usecase Diagram

### 1.5.2 Interaction Diagrams

Both sequence and collaboration diagrams are kinds of interaction diagrams. An interaction diagram shows an interaction, consisting of a set of objects and their relationships. They address the dynamic view of a system. Figures 4.11 and 4.14 are the sequence and collaboration diagrams of the system respectively.

- A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages.
- A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive massages.

Sequence diagram and collaboration diagrams are isomorphic i.e one can be transformed into other.

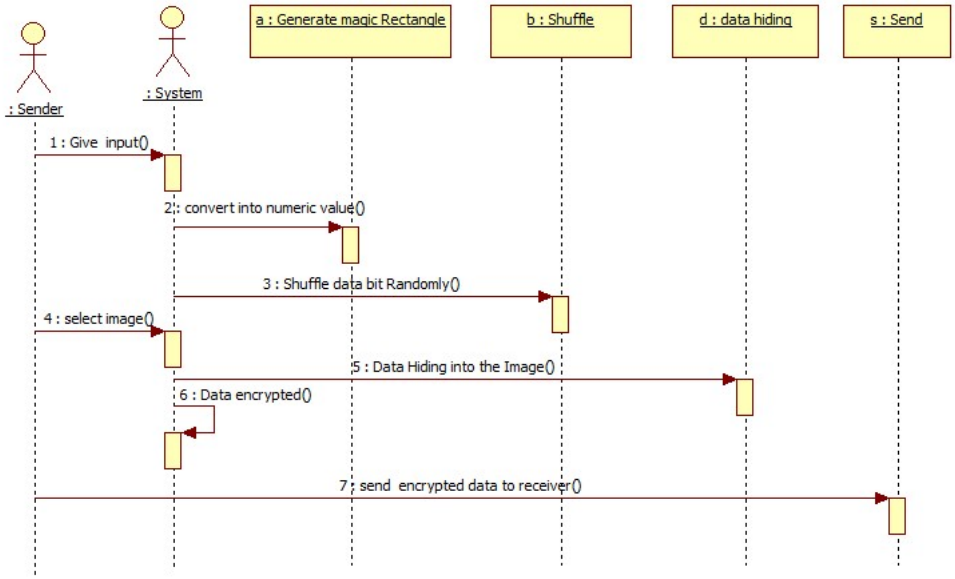


Figure 5: Sequence Diagram

### 1.5.3 Class Diagram

A Class diagram shows a set of classes,interfaces and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagram address the static design view of a system. Figure 4.15 shows the class diagram for the proposed system.

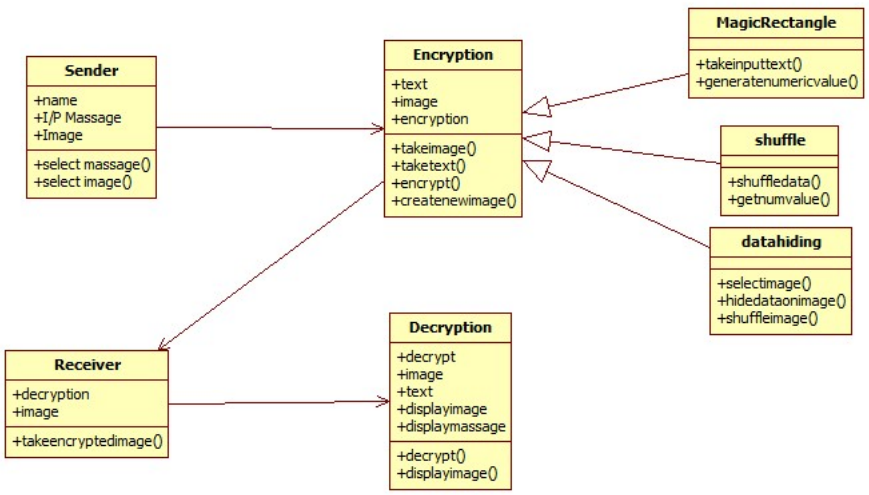


Figure 6: Class Diagram

### 1.5.4 Component Diagram

A component diagram shows the organization and dependencies among a set of components. Component diagrams address the static implementation view of a system. The component diagram for the proposed system is shown is Figure 4.16.

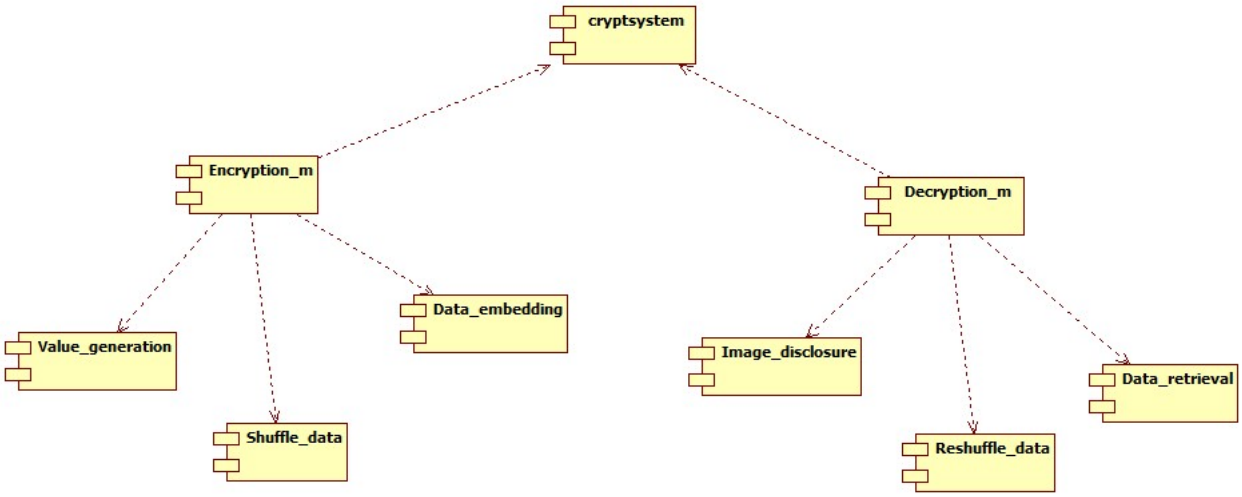


Figure 7: Component diagram

### 1.5.5 Deployment Diagram

A deployment diagram shows the con

guration of run-time processing nodes and the components that live on them. Deployment diagram address the static deployment view of an architecture. Deployment diagram for the proposed system is shown in Figure 4.17.

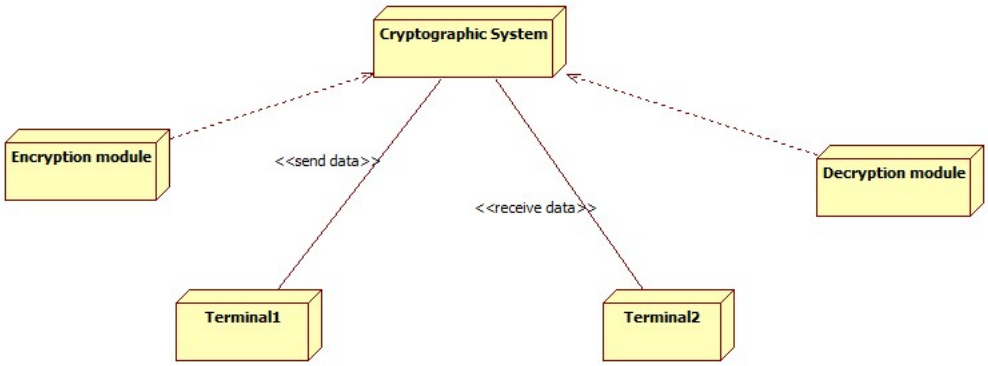


Figure 8: Deployment Diagram

## 1.6 Summary

In this chapter, the proposed system design is described. In the next chapter, Implementation is discussed.