IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-4, Special Issue-1 E-ISSN: 2347-2693

Object Oriented Metrics Based Analysis of DES algorithm for secure transmission of Mark sheet in E-learning

Soumendu Banerjee*
Research Scholar
Department of Computer Science
The University of Burdwan
Burdwan, India
bansoumendu@gmail.com,

Dr. Sunil Karforma
Associate Professor
Department of Computer Science
The University of Burdwan
Burdwan, India
dr.sunilkarforma@gmail.com

Abstract: **Now-a day e-learning is getting popularity than the traditional learning methods. Internet or Intranet is the main communication media in case of e-learning. Since, the access of Internet is very easy now-a-daysand the new generation is so much efficient about the using of Internet, so, the security plays a vital role in case of e-learning. To provide authenticity, we can apply DES algorithm during the transmission of mark sheet from developer to student, two main components of e-learning. The object oriented analysis and design of any system, makes a better understanding of the system and adjustable with the real world. The fundamental characteristic of object oriented analysis is to provide data hiding, encapsulation, data abstraction etc.We can define so many types of metrics like size metric, coupling metric, cohesion metric, inheritance metric etc. in metric analysis. Other than CK metric and MOOD metric, there are also some other object oriented metrics like Lorenz and Kidd, Harrison, Counsell and Nithi etc. We will find the values of different metrics in respect of the DES algorithm regarding the transmission of study material from developer to student.**

Keywords: *E-learning, DES Algorithm, Class Hierarchy Diagram, Object-Oriented Metrics*

I. INTRODUCTION

Though the new trend of learning, e-learning, is growing at a gentle rate, it is becoming quite difficult to provide security during the ongoing process of this system. The total e-learning system is fully dependent on the Internet and Internet is publicly accessible field and current generation is too much efficient in this field. During the transmission of any kind of material through Internet, it is possible for the hackers to reach that documents and can make changes or destroy it. Mark sheet is an important document, which should be delivered from the sender (here developer) to the recipient (here student) secretly[1]. To provide authenticity, the developer can apply Data Encryption Standard (DES) algorithm[2,3] while sending the mark sheet to the student. It is one of the fundamental cryptographic algorithms, which is used to encrypt and decrypt the document to provide security. DES algorithm may be applied in any kind of transmitting document, but, in this paper, we will restrict our discussion only on the transmission of mark sheet.

In this paper, we will discuss on object oriented metric based analysis of DES algorithm[4,8]. Mainly there are two object oriented metric analyses which arefrequently used:

i) Chidamber and Kemerer metric (CK metric) and ii) Metric for Object Oriented Design (MOOD metric).

Except these two metrics, there are some other metric analyses[5] procedures advised by Lorenz and Kidd, Henderson, Harrison, Lee, Braind etc., some of these metrics are included in this paper. The main characteristics of the object oriented analysis and design are inheritance, encapsulation, data abstraction, reuse of codes, data hiding, polymorphism, high cohesion and low coupling. We can also categorize these metric analyses in respect oftheir characteristics[6] like size, cohesion, coupling, inheritance etc. Object oriented analysis and design is always provide some kind of benefits over traditional system by reducing the code, improving the portability, flexibility[7] etc.

In this paper, we have analyzed the metrics based on the class hierarchy diagram of DES algorithm in respect of sending mark sheet from developer to any student in case of an e-learning system. In section II, we will discuss on the class hierarchy diagram of DES algorithm in respect of transmitting mark sheet from developer to student including encryption and decryption. Section III has two parts. The first part will cover the different kinds of metrics which are analyzed in detailed in part two. Finally, we will conclude in section V by showing some future scope.

II. Class Hierarchy Diagram

The class diagram of DES algorithm for mark sheet encryption and decryption during the transmission from developer to student is shown below.

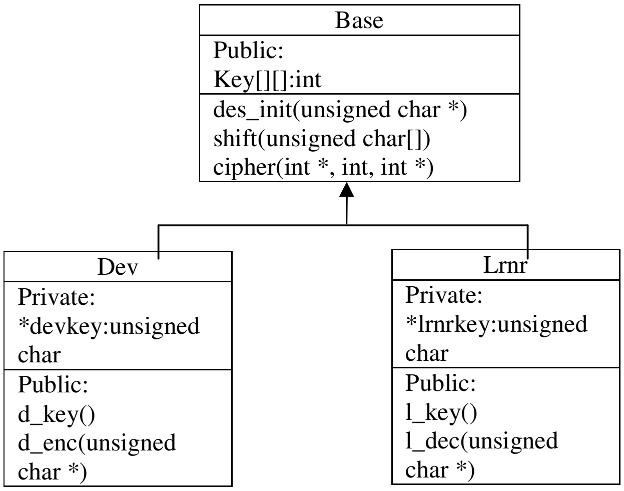

Fig: 2.1.Class hierarchy diagram for DES algorithm for mark sheet encryption and decryption.

The class hierarchy diagram is a tool of Unified Modified Language in case of object oriented Software Engineering. In this diagram, we have used three classes: Base, Dev and Lrnr. The Base class contains one attribute and three public member functions. This class is inherited by the other two classes. The Dev class contains one private data member and two public member functions. Similarly, the other class also has one private data member and two member functions. The Base class is used to generate the key. Dev class is used to encrypt the study material and Lrnr class will decrypt the encrypted mark sheet.

III. OBJECT ORIENTED METRICS BASED ANALYSIS

In this section, we will discuss on some object oriented metrics[9], which values are analyzed for the above class hierarchy diagram. Some standard metrics are discussed in section III.1 and performance of Object Oriented Analysis of the proposed model are shown in section III.2.

III.1. SOME STANDARD METRICS

Though there are some traditional metrics like Lines of code (LOC), Comment Percentage (CP) etc.[13], here we will discuss only on the object oriented metrics. The advantages of object oriented metrics are to have a good understandability of the proposed model, assess the effectiveness of the process, to make an improvement in the quality of work performance[10] etc. The object oriented analysis and design of a system reduces the cost of maintainability of the system, help in reusing of codes and also helps to hide the code from the outside world. First of all we want to discuss on some attributes on the basis of which we measure the value of the metrics:

- Class: Class is used as template to create or instantiate objects to serve our purpose in object oriented programming[11].
- Coupling:In case of the design of software engineering models,coupling is used to make sense of the interdependency of the part of modules of any system, with others. Low coupling is a good characteristic for any object oriented design.
- Cohesion: Cohesion, in case of software engineering, indicates that the part of design modules is how much tightly bonded with others. High cohesion is expected in any design.
- Information Hiding: Information hiding is one of main property of any object oriented design[12]. Using this property, the designer can make their code hidden from the outside world.

- Inheritance: This property of object oriented approach helps the designer to create classes which can acquire the properties of another class, known as base class.
- Polymorphism: It is used as an object oriented feature, through which a variable or a function can take multiple forms and it helps designer to implement a design in general form, not in specific.

On the basis of these features, we will discuss on some object oriented metrics, whose values will be calculated in respect of our proposed class diagram which is successfully wrapping DES algorithm. The well known metrics like some metrics of CK and MOOD metrics and also some other metrics are discussed below:

- Number of Attributes per class (NOA): The value of NOA is the total number of attributes defined in a class.
- Number of methods per class (NOM): The value of NOM is the total number of methods defined in a class.
- Depth of Inheritance tree (DIT): Its value is defined by the length of the node from the root of the tree. In case of the base class, the value of DIT will be 0 and its immediate children will have the value 1 and so on.
- Coupling between Objects (CBO): Its value for a class is equal to the number of other classes to which it is coupled.
- Number of children (NOC): Its value for a class is equal to the number of directly inherited sub-classes of the class.
- Response for a class (RFC): It is equal to the number of methods that can be invoked in response to a message in a class.
- Method hiding factor (MHF): Itis a measure of encapsulation which states the sum of the invisibilities of all methods in all classes, where the invisibility of a method is the percentage of the total class from which the method is hidden[mukta].
- Attribute Hiding Factor (AHF): It is also a measure of encapsulation in object oriented design which is calculated by the sum of invisibilities of all attributes in all classes, where the invisibility of an attribute is the percentage of the total class from which this attribute is hidden.
- Method Inheritance Factor (MIF): It is related with inheritance. MIF is defined as the ratio of the sum of the inherited methods in all classes of the system to the total number of methods which are available for all classes.
- Attribute Inheritance Factor (AIF):It is also related with inheritance. It is the ratio of the sum of inherited attributes in all classes of the system to the total number of attributes which are available for all classes.

III.2. ANALYSIS OF PROPOSED MODEL

In this section, we will find the values of the object oriented metrics, we have discussed in the above section in respect to our proposed class hierarchy diagram regarding the encryption and decryption of the mark sheet from developer to learner[14]. First, we use a table to show the values of CK and two other metrics below:

NOA= number of attributes in the class
NOM= count of methods in the class
CBO= number of other classes to which the class is coupled
DIT= maximum path from the node to the root in the inheritance tree
NOC= number of subclasses inherit the methods of parent class
RFC= {M} U all i {Ri}, where where{ Ri} = set of methods called by method *i*and { M } = set of all methods in the class.

Table 4.1: Metrics of mark-sheet authentication using DES algorithm

OO Metrics	Classes of proposed system		
	Base	Dev	Lrnr
NOA	1	1	1
NOM	3	2	2
CBO	2	0	0
DIT	0	1	1
NOC	2	0	0
RFC	7	5	5

Now, we will design graphs based on the values, we have shown in the above table of the metrics and discuss about the values.

From the value of NOA and NOM, we can make an estimate of the required time and effort to develop and maintain a class. Fig 4.1 and Fig 4.2 show the graphical representation of NOA and NOM, which should be kept down. So, the values of NOA and NOM of our proposed systemare ok.

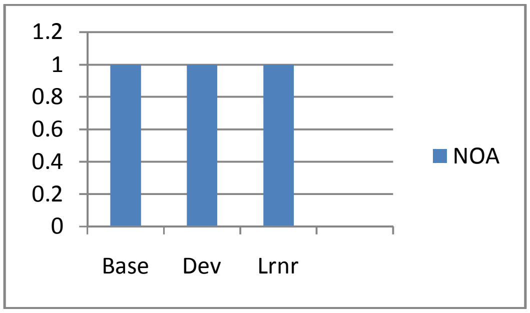

Fig 4.1: NOA

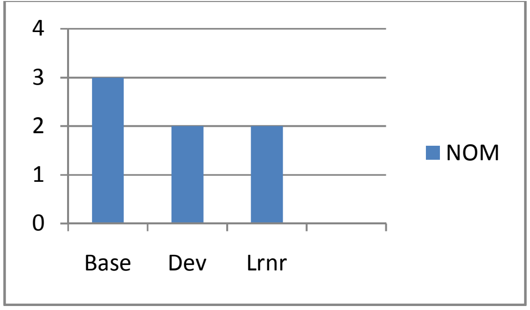

Fig 4.2: NOM

Fig 4.3 graphically represents the value of CBO metrics, which represents the coupling, so we try to keep it low in our system.

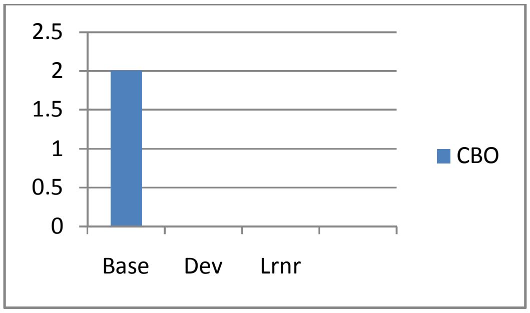

Fig 4.3: CBO

In Fig 4.4, we graphically represent the value of DIT, which represents the complexity behavior of the class. If the value of DIT increases, that means many methods might be reused. Here the maximum value of DIT is 1 that means our system is easy to maintain.

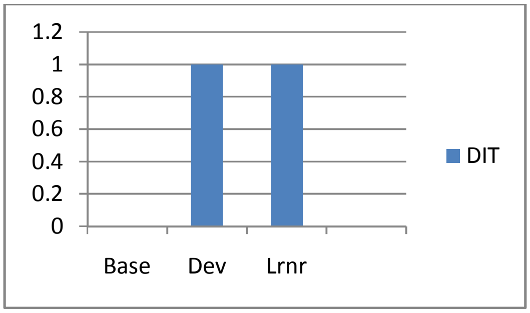

Fig 4.4: DIT

Fig 4.5 shows the metric NOC, whose maximum value in our proposed system is 2, which is the maximum value of any object oriented system.

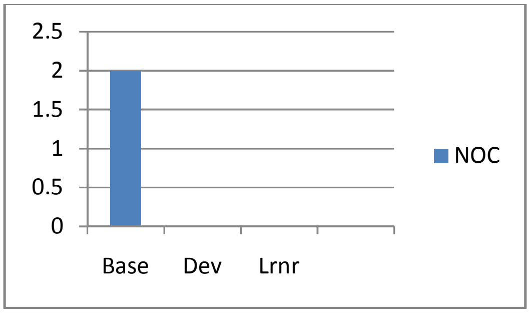

Fig 4.5: NOC

Fig 4.6 graphically represents the value of RFC. If the value increases then it becomes difficult to understand the complexity of the class and if it keeps low, then the polymorphism becomes greater. Here the optimal value of RFC is found.

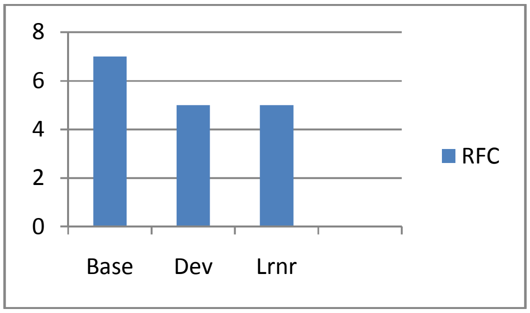

Fig 4.6: RFC

Now we will discuss on some metrics, which are under the section of Metric for Object Oriented Design (MOOD) metrics related to the mark-sheet authentication using DES algorithm from developer to learner in an e-learning system.

Equation for MHF=$\sum_{i=1}^{TC} M_h(C_i)/ \sum_{i=1}^{TC} M_d(C_i)$ //TC means total number of class
Where $M_d(C_i)$= $M_v(C_i)$+ $M_{h`}(C_i)$ where $M_d(C_i)$= methods defined in class C, $M_v(C_i)$= methods visible in class C and $M_{h`}(C_i)$= methods hidden in class C

Table 4.2: MHF

	Classes of proposed system			
	Base	Dev	Lrnr	Summation(∑)
$M_{h`}(C_i)$	0	0	0	0
$M_{v`}(C_i)$	3	2	2	7
$M_{d`}(C_i)$	3	2	2	7
MHF	0/7=0			

Table 4.2 shows the value of the MHF metric of our proposed system. This value is low, which means insufficiently abstracted implementation, i.e., very simple design.

Equation for AHF=$\sum_{i=1}^{TC} A_h(C_i)/ \sum_{i=1}^{TC} A_d(C_i)$
$A_d(C_i)$= $A_v(C_i)$+ $A_h(C_i)$, where $A_d(C_i)$= total attributes defined in class C, $A_v(C_i)$= Attributes visible in class C and $A_h(C_i)$= attributes hidden in class C

Table 4.3: AHF

	Classes of proposed system			
	Base	Dev	Lrnr	Summation(∑)
$A_{h`}(C_i)$	0	1	1	2
$A_{v`}(C_i)$	1	0	1	1
$A_{d`}(C_i)$	1	1	1	3
AHF	2/3=0.67			

Table 4.3, contains the value of the AHF metric. Here the value of AHF is 0.67, which lies between 1 and 0, which is quite ok.

Equation for MIF=$\sum_{i=1}{}^{TC}M_i(C_i)/\ \sum_{i=1}{}^{TC}M_a(C_i)$
Where $M_a(C_i)= M_d(C_i)+ M_i(C_i)$, $M_a(C_i)$=number of methods available, $M_d(C_i)$= number of methods defined and $M_i(C_i)$= number of methods inherited

Table 4.4: MIF

	Classes of proposed system			
	Base	Dev	Lrnr	Summation(∑)
$M_d(C_i)$	3	2	2	7
$M_i(C_i)$	0	3	3	6
$M_a(C_i)$	3	5	5	13
MIF	6/13=0.46			

Table 4.4 shows the value of MIF metric. It should be followed that the value of MIF should not be too low or too much high. Here MIF=0.46. So, it is quite ok.

Equation for AIF=$\sum_{i=1}{}^{TC}A_i(C_i)/\ \sum_{i=1}{}^{TC}A_a(C_i)$
Where $A_a(C_i)= A_d(C_i)+ A_i(C_i)$, $A_a(C_i)$=number of methods available, $A_d(C_i)$= number of methods defined and $A_i(C_i)$= number of methods inherited

Table 4.5: AIF

	Classes of proposed system			
	Base	Dev	Lrnr	Summation(∑)
$A_d(C_i)$	1	1	1	3
$A_i(C_i)$	0	1	1	2
$A_a(C_i)$	1	2	2	5
AIF	2/5=0.4			

Table 4.5 shows the value of AIF of our proposed model. If the value of AIF is 0%, it means that there is no attribute exists in the class and also there is lacking of inheritance. Here the value of AIF is 0.4, which is quite ok.

REFERENCES

[1]. Weippl, R.E., Security in E-Learning (Springer, 2005)

[2]. Andrew, S. Tanenbaum (2005), Computer Networks, Pearson Prentice Hall

[3]. Behrouz, AForouzan (2006), Data Communication and Networking, Tata McGraw Hill

[4]. S. Karforma and S. Banerjee, “Object Oriented Implementation of DES forSecurity in E-Learning”, Asian Resonance, vol-III, Issue-IV, Oct-14, pp: 12-20

[5]. [http://ce.sharif.edu/courses/8586/1/ce924/resources/root/4.%20Kamandi_OOMetrics.pdf

[6]. K.K.Aggarwal, Y. Singh, A. Kaur, R. Malhotra, “Empirical study of object oriented metrics”, Journal of Object Technology, ETH Zurich, Chair of Software Engineering, Vol. 5, No. 8, November-December 2006, pp: 149-173

[7]. Rajib Mall, Fundamentals of Software Engineering (Prentice Hall of India, New Delhi, 2006)

[8]. Karforma S. and Mukhopadhyay S., A Study on the application of Cryptography in E-Commerce, The University of Burdwan, West Bengal, India, July-2005

[9]. Muktamyee S., An overview of Object Oriented Design Metrics, (Master Thesis) Department of Computer Science, Umeå University, Sweden June 23, 2005

[10]. www.psu.edu/courses/infsy/infsy570_rxo4/metrics/metrics.ppt

[11]. http://techterms.com/definition/class

[12]. Balagurusami E., Object oriented programming with C++ (Tata McGraw Hill, New Delhi, 2006)

[13]. A. Kamandi, “Object Oriented Metrics”, Sharig University of technology, spring 2007

[14]. S. Karforma and S. Banerjee, “Object oriented metric based analysis of ElGamal digital signature algorithm for study material authentication”, IJSTM, vol-4, special issue-1, sept-15, pp: 522-530