2nd International Conference on Science, Technology and Management
University of Delhi (DU), Conference Centre, New Delhi (India)
27 September 2015, www.conferenceworld.in, www.icstmdu.com
978-81-931039-6-8

OBJECT ORIENTED METRIC BASED ANALYSIS OF ELGAMAL DIGITAL SIGNATURE ALGORITHM FOR STUDY MATERIAL AUTHENTICATION

Soumendu Banerjee[1], Dr. Sunil Karforma[2]

[1]Research Scholar, [2]Associate Professor, Department of Computer Science,
The University of Burdwan (India)

ABSTRACT

Digital Signature provides some kinds of assurance in the aspect of authentication issue in e-learning transaction. ElGamal digital signature is one kind of digital signature, which may be used in any e-learning system to provide authentication during transmission of study material from developer to student. Object Oriented Analysis and Design of any system makes it better and very much adjustable with the real world. A well-designed object-oriented system should have the characteristics like, encapsulation, abstraction, high cohesion and low coupling. Chidamber and Kemerer metric (CK metric) and Metric for Object Oriented Design (MOOD metric) are the two main object oriented metrics. We will perform analysis of the different metric values of ElGamal digital signature algorithm based on the class hierarchy diagram for signature generation and verification regarding the transmission of study material from the developer to the student in an e-learning system.

Keywords*: *E-learning, ElGamal Digital Signature, Class Hierarchy Diagram, CK-metric, MOOD Metric

I. INTRODUCTION

In our modern era, e-learning system is growing up in a gentle speed. The total e-learning process is fully dependent on the Internet and the Internet, being an open access to all, security is the main issue for any e-learning system. Digital Signature is such a process through which authentication can be achieved in any e-learning system during transaction through online. Suppose the developer, in an e-learning system, sends a set of study material to any student along with a signature made by digitally. Now, after receiving the material, student will verify the signature for authentication. If the signature is matched, then the student will accept the material, otherwise reject it, since, if the hackers make any to change the study material during transmission, then the signature will be altered and do not match. If the signature is mismatched, then the student will request the developer to send the material again. So, using Digital Signature, we can achieve the security issues like privacy, integrity, authenticity and non-repudiation[1,2] in an e-learning system and help to make it better. There are so many digital signature algorithms like, DSA, RSA digital signature, ElGamal digital signature, GOST digital signature, KCDSA(Korean certificate based digital signature algorithm) etc. Among these digital

signatures, ElGamal is better than RSA digital signature, for its discrete logarithm problem in cyclic group, which is very tedious and difficult to solve[3].

Here we will discuss on Object Oriented Analysis and Design metrics based on the class hierarchy diagram of ElGamal digital signature algorithm. There are some kinds of benefits in using object oriented analysis and design over traditional structured system analysis and design[4,5] like reducing the cost of maintenance, real world modeling, improvement in reliability and flexibility, code reusability etc.

In section II, we will design the class hierarchy diagram of ElGamal digital signature algorithm in the basis of transmitting study material for signature creation and verification. In section III, we will discuss on two most important object oriented metrics: CK-metrics and MOOD metrics etc. Section IV will cover the computation tables and diagrams regarding the metric analysis. Finally, we will end up in section V giving a brief conclusion.

II. CLASS HIERARCHY DIAGRAM

2.1 Class Diagram of Proposed System

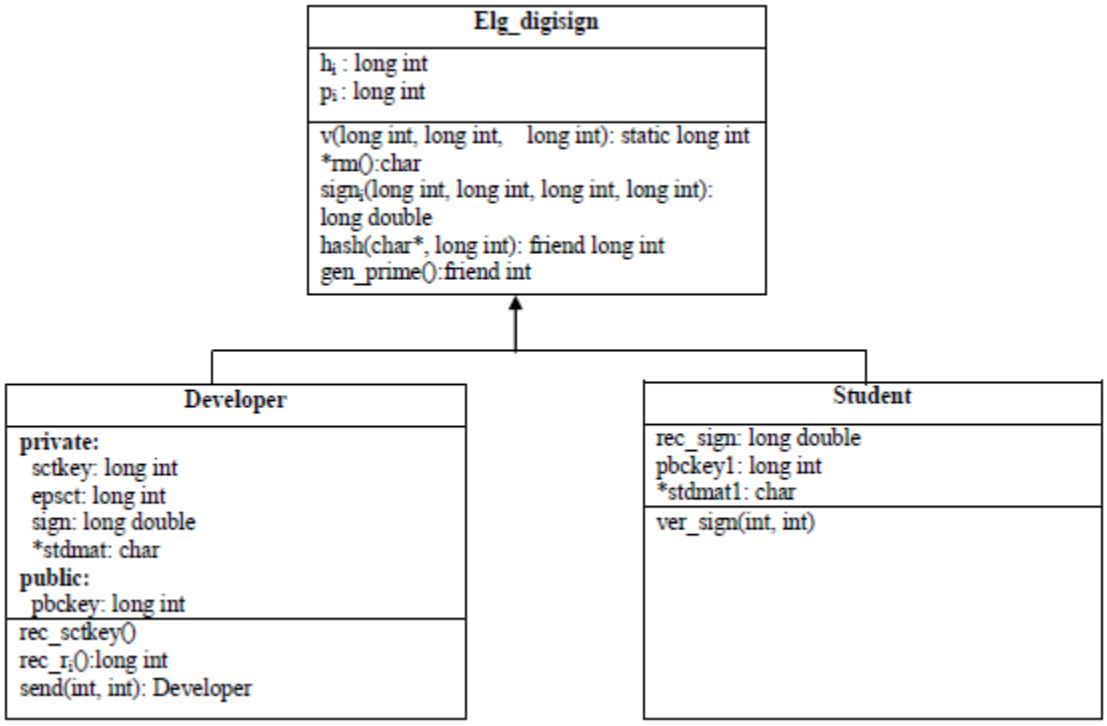

Fig 2.1: Class Diagram of ElGamal Digital Signature for Signature Creation and Verification

Class diagram is a part of Unified Modeling language (UML) in software engineering. It shows the structure of a system by showing the classes, their attributes, operations and relationship between objects[6,7]. In the above diagram, Fig 2.1, we have used three classes, which are necessary for the signature creation and verification during the transmission of study material from developer to student, using ElGamal digital signature, in an e-learning system[8,9].

The base class is Elg_digisign, which does not contain any object but mainly used for making inheritance. It consists of two data members and the five member functions are inherited publicly by two other classes: Developer and Student. Developer class contains four private and one public data member and also three public

member functions. Similarly, Student class contains three public data members and one member function. Using this class diagram, we will analysis the values of different metrics.

III. OBJECT ORIENTED METRICS

Object Oriented Analysis and Design of software system has many advantages like maintenance cost become less, codes can be reused, data can be made hidden from the outside world, decomposition of problem into easily understood object etc. Metrics are units of measurement. There are five characteristics[10] of object oriented metrics:

i) **Localization**: This process place items in close physical proximity to each other

ii) **Encapsulation**: This process used to bind data and functions into a single class-type variable[11]. When an object communicates with the outside world only through messages, is known as encapsulation.

iii) **Information hiding**: Information hiding means to design modules such a way that the data of a particular module can only be accessible only where it is required, but not from all the others.

iv) **Inheritance**: Through this process object of one class can acquire the properties of objects of another class.

v) **Abstraction**: The process through which, a class is developed in terms of its functionality and interface, rather than its implementation details.

Though there are so many object oriented design quality measuring methods are available now-a-days, but we will discuss our proposed diagram using the two most popular object oriented metrics: Chidamber and Kemerer Metrics (CK-Metrics) and Metric for Object Oriented Design metrics (MOOD Metrics). These two metrics are complement to each other.

Ck-metrics include six metrics for the object oriented design[12]:

i) **Weighted Methods per Class (WMC)**: It counts the total number of methods of a class. The measure of WMC should be kept down.

ii) **Coupling Between Object classes (CBO)**: CBO is the count of other classes to which it is coupled. CBO is the sum of all non-inheritance-related class dependencies. It is expected that the value of CBO should be kept as low as possible.

iii) **Depth of Inheritance Tree (DIT)**: The depth of inheritance tree is measured by the depth of the class in the inheritance hierarchy. The DIT value of base class is 0 and its immediate children have a DIT value of 1 and its grandchild DIT value is 2 and so on. If the value of DIT is high, then it becomes difficult to maintain[13].

iv) **Number Of Children (NOC):** Like DIT, the value of NOC is also related with the inheritance. The number of NOC of a class is the number of subclasses directly inherits from that class.

v) **Response For a Class (RFC):** The value of RFC of a class is sum of the number of methods of the class and the number of methods called by any of those methods[14]. Through the value of RFC, we measure complexity of the class. If the number of invoked methods is high, for a message then complexity increases and maintainability decreases, thus the quality of the software decreases[15].

vi) **Lack of Cohesion in Methods (LCOM):** The value of LCOM, measures the dissimilarity of methods in a class via instanced variables[15].

Now we will briefly discuss on the six different metrics of MOOD metrics which examine the encapsulation, inheritance, polymorphism and message passing of object oriented design:

i) **Method Hiding Factor (MHF):** The method hiding factor is a measure of encapsulation which states the sum of the invisibilities of all methods in all classes.

ii) **Attribute Hiding Factor (AHF):** This attribute hiding factor is also a measure of encapsulation in object oriented design which is calculated by the sum of invisibilities of all attributes in all classes.

iii) **Method Inheritance Factor (MIF):** The method inheritance factor is a measure of inheritance. MIF is defined as the ratio of the sum of the inherited methods in all classes of the system.

iv) **Attribute Inheritance Factor (AIF):** The attribute inheritance factor is also a measure of inheritance. The value of AIF, 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.

v) **Polymorphism Factor (PF):** By literature, 'poly' means 'many' and 'morphism' means 'forms'. That means the ability to take many forms. It is measured by the number of actual methods overridden divided by the maximum number of possible methods override[16].

vi) **Coupling Factor (CF):** Coupling factor measures if the design is a low coupled or tight coupled. The value of coupling factor is measured by the division value of actual couplings value by the maximum possible coupling values.

IV. ANALYSIS OF PROPOSED MODEL

In this section, we will analyze the class diagram of transmitting study material from developer to student using ElGamal Digital Signature, shown in Fig-1.1, in respect of two above discussed object oriented metrics: CK-metrics and MOOD metrics. The above class diagram consist only three classes: Elg_digisign, Developer and Student. First, we will show the values of the metrics of CK-metrics in respect of these three classes, in the following table:

- WMC= count of methods implemented 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 {R_i}, where where { Ri} = set of methods called by method *i* and { M } = set of all methods in the class.

Classes of proposed diagram	**Object-Oriented Quality Metrics**				
	WMC	CBO	DIT	NOC	RFC
Class Elg_digisign	5	2	0	2	9
Class Developer	3	0	1	0	8
Class Student	1	0	1	0	6

Table 4.1: Metrics of Signature Generation and Verification using ElGamal Digital Signature

Based on the analyzed data shown in the table 4.1 of five metrics of Chidamber and Kemerer WMC, CBO, DIT, NOC and RFC metrics, we can design graphs and make some discussion on the values.

- From the value of **WMC**, shown in the graph (Fig 4.1) below, we can predict how much time and effort will be required to develop and maintain a class. As the value of WMC should be kept down, so, the value of WMC of our proposed system is ok.

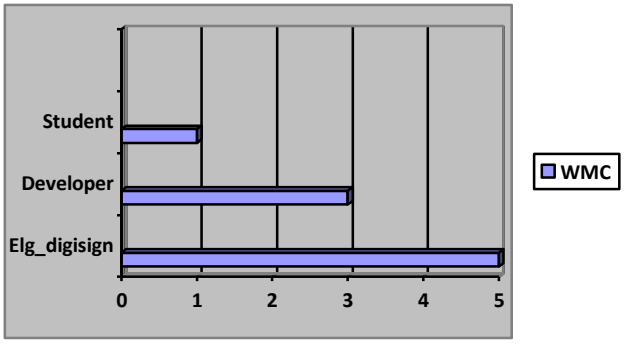

Fig 4.1: WMC

- Fig 4.2 shows the graphical representation of the values of three classes in respect of **CBO** metrics, through which we can measure the coupling level of the proposed system. In our proposed system we have preserved the value of CBO as low as possible.

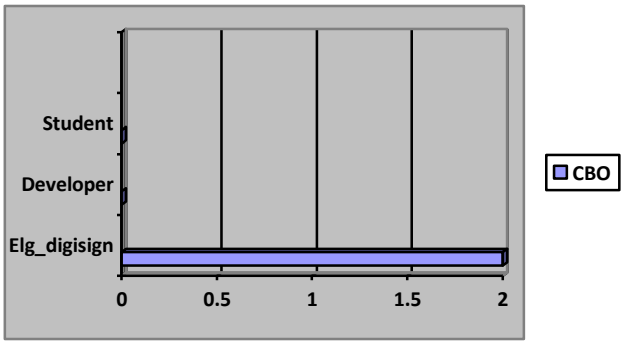

Fig 4.2: CBO

- **DIT** represents the complexity of the behavior of a class. If the value of DIT increases, it means that more methods are to be expected to be inherited but a large DIT value indicates that many methods might be reused. Here the maximum value of DIT is 1, that means, the system is easy to maintain.

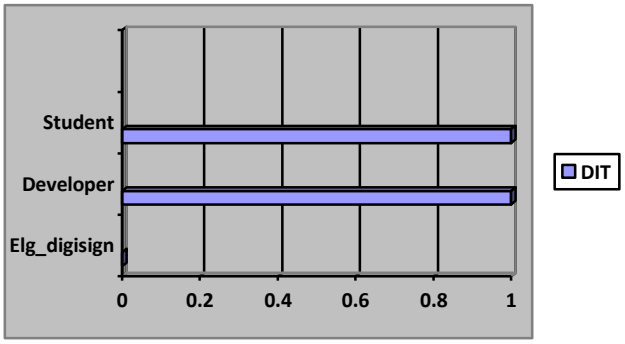

Fig 4.3: DIT

- The graph of the value of the metric **NOC** is shown in the fig 4.4. NOC shows the level of reuse. The maximum value of NOC of our system is 2, which is the maximum possible value of any object oriented system.

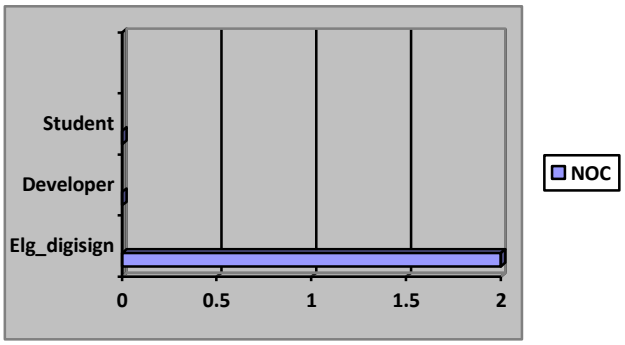

Fig 4.4: NOC

- The value of **RFC** metrics is shown graphically in Fig 4.5. If the value of RFC increases, then it becomes difficult to understand the total complexity of the class and if the value is very low, then polymorphism becomes greater. Here the optimal value of RFC is found.

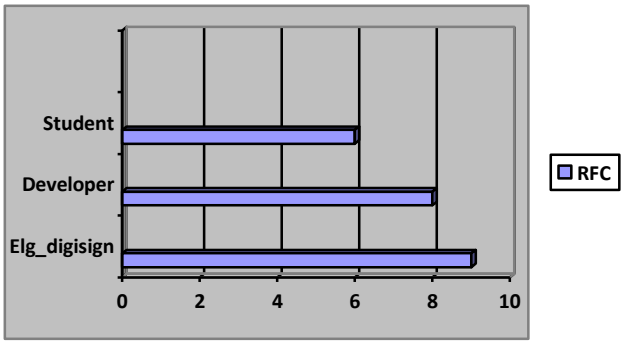

Fig 4.5: RFC

Now we will discuss on the metrics under **MOOD metrics** related to the transmission of the study material using ElGamal digital signature from developer to student 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

	Classes of proposed system			
	Elg_digisign	Developer	Student	Summation(∑)
$M_h(C_i)$	0	0	0	0
$M_v(C_i)$	5	3	1	9
$M_d(C_i)$	5	3	1	9
MHF	0/9=0			

Table 4.2: MHF Metrics of Proposed System

Table 4.2 shows the value of the MHF metric under MOOD metric of our proposed system. The value of low MHF means insufficiently abstracted implementation, which makes our design very simple.

- 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

	Classes of proposed system			
	Elg_digisign	Developer	Student	Summation(∑)
$A_h(C_i)$	0	4	0	4
$A_v(C_i)$	2	1	3	7
$A_d(C_i)$	2	5	3	10
AHF	4/10=0.4			

Table 4.3: AHF Metrics of Proposed System

In the table 4.3, we analyze the value of the AHF metric. If the value of AHF is 100%, that means all the methods are private and if the value is 0%, then all the methods are public. Here the value of AHF is 0.4, 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

	Classes of proposed system			
	Elg_digisign	Developer	Student	Summation(∑)
$M_d(C_i)$	5	3	1	9
$M_i(C_i)$	0	5	5	10
$M_a(C_i)$	5	8	6	19
MIF	10/19=0.53			

Table 4.4: MIF metrics of proposed system

We use table 4.4 to show the value of MIF metric, which is 0.53, of our proposed model. It should be followed that the value of MIF should not be too low or too much high. So, this value is showing that the proposed system is good.

- 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

	Classes of proposed system			
	Elg_digisign	Developer	Student	Summation(∑)
$A_d(C_i)$	2	5	3	10
$A_i(C_i)$	0	2	2	4
$A_a(C_i)$	2	7	5	14
AIF	4/14=0.29			

Table 4.5: AIF Metrics of Proposed System

In table 4.5, we calculate 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 0.29, which is quite ok.

- Equation for **PF**=$\sum_{i=1}^{TC} M_o(C_i)/ \sum_{i=1}^{TC}[M_n(C_i)*DC(C_i)]$

$M_n(C_i)$= new methods defined in the class, $M_o(C_i)$= overridden methods in the class and $DC(C_i)$= the descendants count in C_i.

	Classes of proposed system			
	Elg_digisign	Developer	Student	Summation(∑)
$M_n(C_i)$	0	3	1	4
$M_o(C_i)$	0	5	0	5
$DC(C_i)$	2	0	0	2
PF	5/(4*2)=5/8=0.62			

Table 4.6: PF Metrics of Proposed System

Here the value of PF metric of our proposed system is 0.62, which implies that our system is well designed and it contains inheritance.

- Coupling factor (**CF**)=Actual coupling/Possible coupling

The table below shows the actual coupling between the three classes of our proposed system.

Classes	Classes of proposed system			
	Elg_digisign	Developer	Student	Summation(∑)
Elg_digisign	X	1	1	2
Developer	1	X	0	1
Student	1	0	X	1
TC=3	Total number of coupling=4 Possible number of coupling=6			
CF	4/6= 0.67			

Table 4.7: CF Metrics of Proposed System

From the above table 4.7, we can find that the value of coupling factor of our proposed system is 0.67, which in between 0% and 100%, which means that our system is well-coupled.

V. CONCLUSION

We have made Object-Oriented analysis for improvement of quality and authenticity of the system of generating and verifying digital signature while transmitting study material from developer to student, using ElGamal digital signature. Object-Oriented analysis is done with the help of class diagram, Chidamber and Kemerer metrics (CK metrics) and Metric for Object Oriented Design metrics (MOOD metrics). Further improvement of the system may be done with the application of water marking technology, which is beyond the scope of this paper.

REFERENCES

[1] Weippl, R.E., Security in E-Learning (Springer, 2005)
[2] https://en.wikipedia.org/wiki/Digital_signature
[3] Hankerson D., Menezes A., Vanstone S., Guide to Elliptic Curve Cryptography (Springer)
[4] http://www.freepatentsonline.com/article/International-Journal-Business-Research/190463129.html
[5] http://www.dba-oracle.com/t_object_oriented_approach.htm.
[6] https://en.wikipedia.org/wiki/Class_diagram
[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] Karforma S. and Banerjee S., Object oriented modeling of ElGamal digital signature for authentication of study material in E-learning system, IJARSE, ISSN-2319-8354(E) Vol No.4, Special Issue (02), February 2015pp: 455-460
[10] Edward V. Berard, Metrics for Object-Oriented Software Engineering, The Object Agency, Inc.
[11] Balagurusami E., Object oriented programming with C++ (Tata McGraw Hill, New Delhi, 2006)
[12] Karforma S. and Banerjee S., DIPRM IN E-COMMERCE SYSTEM-A UML BASED APPROACH (Ph.D theses- 2012)
[13] Muktamyee S., An overview of Object Oriented Design Metrics, (Master Thesis) Department of Computer Science, Umeå University, Sweden June 23, 2005
[14] Sharma A K., Kalia A., Singh H., Metrics Identification for Measuring Object Oriented Software Quality, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231 -2307, Volume–2 , Issue-5, November 2012
[15] sdlab.naist.jp/members/camargo/presentations/CKMetrics.ppt
[16] http://www.aivosto.com/project/help/pm-oo-mood.html