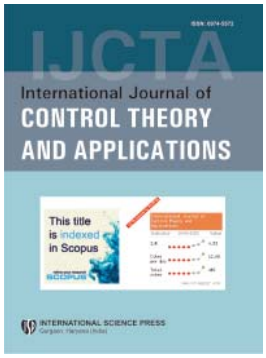

International Journal of Control Theory and Applications

ISSN : 0974–5572

Volume 9 • Number 42 • 2016

Object Oriented Metric based Analysis of RSA Digital Signature to Authenticate Mark Sheet in E-learning

Soumendu Banerjee[a] and Sunil Karforma[a]

[a]Department of Computer Science, The University of Burdwan Golapbag, West Bengal, India

E-mail: bansoumendu@gmail.com, sunilkarforma@yahoo.com

Abstract: E-learning is now-a-days spreading worldwide as a new trend of learning. In this electronic learning system, all the transmissions of documents between the three main participants of *e*-learning system, *i.e*. teacher, developer and learner are done via Internet. Due to the public access of Internet, during the transmission of important documents like mark sheet via internet, hacker can make change or destroy it. RSA Digital Signature is one of the very popular security schemes, through which the e-learning institutions can provide the security regarding the transmission of important documents. In this paper, we will calculate the values of some metrics based on the class diagram of transmission of mark sheet from the developer to learner based on RSA digital signature.

Keywords: *RSA digital signature, class hierarchy diagram, CK metrics, MOOD metrics.*

1. INTRODUCTION

E-learning is Information and Communication Technology (ICT) based education system. Like any other kinds of learning processes, mark sheet is a very important document for any learner in an *e*-learning system. If hackers can reach the mark sheet, while transmitting from developer to learner, they can change or destroy the mark sheet. To provide security regarding transmission, the developer of the *e*-learning institution may use the RSA digital signature to provide the security issues mainly authenticity, integrity and non-repudiation[1]. Digital signature is the technique through which, the sender can authenticate the sending material and after receiving, the receiver can also verify the signature for its originality. If the signature is matched, then receiver will accept, otherwise reject and request for resending. This verification is done by comparing the hash values. RSA digital signature is a digital signature scheme which is based on the public key cryptography RSA.

If the developer sends the mark sheet, digitally signed, then it will authenticate the institution and if it is changed or altered during transmission, that means the integrity is hampered and in that case, this tampering can be checked while the verification of the digital signature will be done at the learner's end. Non-repudiation means after sending the mark sheet to the learner, the developer can't deny about the transmission, which is sometimes very necessary in security aspect. Digital signature is gaining importance now-a-days as the government has approved the equal significance of digital signature and handwritten signature[2,3].

Object oriented implementation of any system provides the advantages over data redundancy, code reusability and also helps in reducing the maintenance cost. This approach is better than the traditional approach mainly due to the real world implementation by using classes and objects and this approach makes the system more reliable and flexible. Object oriented metric analysis helps in evaluating the effort of development and testing of an system[4]. It helps in better understandability, maintainability and also in reusability.

In this paper, we calculate and analyze the values of two basic object oriented metrics: Chidamber and Kemerer (CK metric) metric and Metric for Object Oriented design (MOOD metric) metric based on the class hierarchy diagram of RSA digital signature regarding the transmission of mark sheet from developer to learner in an *e*-learning system.

Section II will cover the class hierarchy diagram of RSA digital signature in respect of the transmission of mark sheet from developer to learner. We divide section III in two parts, first part contains the brief discussion on the object oriented metrics and in second part we analyze the metric values based on our proposed model in details. Finally, we conclude in section IV by showing some future scopes.

2. CLASS HIERARCHY DIAGRAM

The class diagram of RSA digital signature generation and verification for mark sheet transmission from developer to learner is shown in Figure1[5,6]:

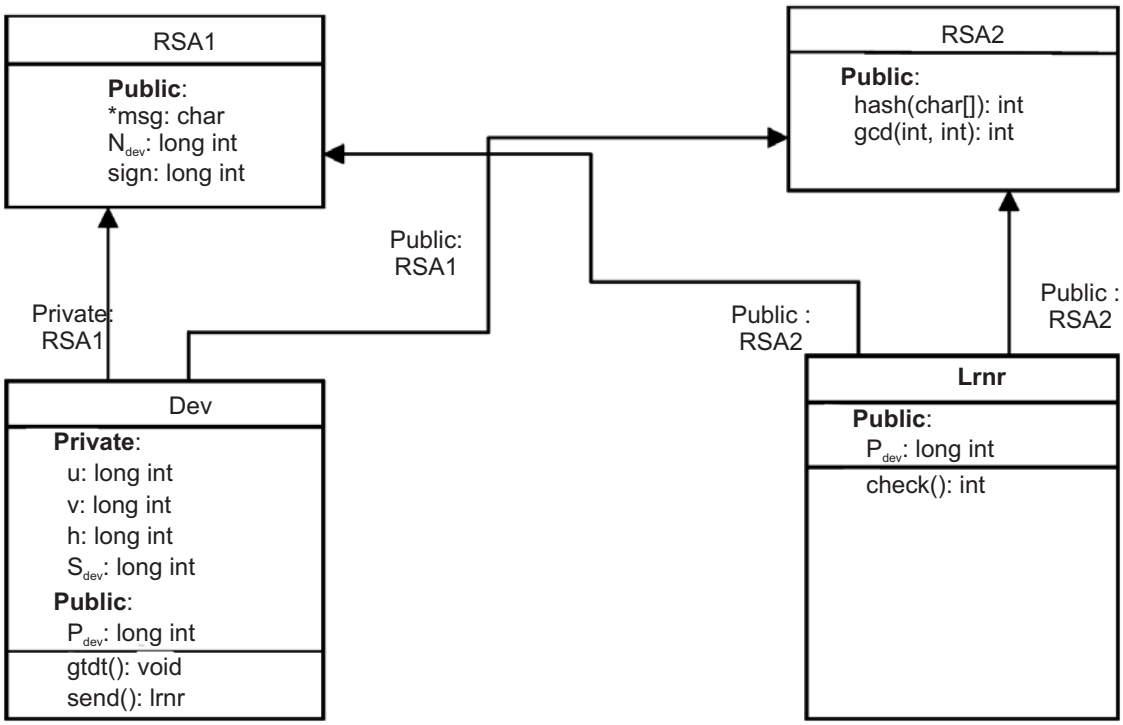

Figure 1: Class diagram of RSA digital signature regarding the transmission of mark sheet from developer to learner

Here four classes are used: RSA1, RSA2, Dev and Lrnr. RSA1 and RSA2 are used as the base classes. In the following section, covers a brief description on the data members and member functions of the class hierarchy diagram:

Class RSA1: char *msg;//Here msg is the mark sheet to be sent by an object of the class Dev
long int N_{dev}; //it contains the product of two prime numbers
long int sign;// it represents the digital signature

Class RSA2: int hash(char[]); //this is used to create hash value
int gcd(); //used to find the gcd of two numbers

Class Dev: long int *u*,*v*; //*u* and v are two prime numbers
long int *h*; //*h* is used to store hash value
long int S_{dev}; //it stores the secret key of class Dev
long int P_{dev}; //it stores the public key of class Dev
void gtdt(); //it is used to receive mark sheet and signature from Dev
Lrnr(send);// it is used to send the mark sheet along with the signature to the learner from the developer

Class Lrnr: long int P_{dev}; //it also used to store public key of class Dev
int check(); //it is used to verify digital signature at the learner's end

3.1. Object Oriented Metrics

The first section contains a brief discussion on the properties of object oriented metrics. The advantages which can be achieved through object oriented analysis of any system are reduction of maintenance and development cost, reduce the effort of maintenance, code reusability and better understandability. Before analyzing the values of the object oriented metrics, we must aware about the attributes on the basis of which we will measure the values of the metrics[7,8,9]:

1. **Class:** In object oriented programming, the class is used as a template for creating or instantiating objects within a program[10]. Each object is created from a class but one class can be used to instantiate multiple objects.
2. **Localization:** This process place items in close physical proximity to each other
3. **Coupling**: It is used to make sense of the interdependency of the part of modules of any system.
4. **Cohesion:** It indicates that the part of design modules of any system.
5. **Encapsulation:** This process used to bind data and functions into a single class-type variable[11].
6. **Information hiding:** It 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[12].
7. **Polymorphism:** Using this object oriented feature, a variable or function can be represent in multiple forms.
8. **Inheritance:** Through this process object of one class can acquire the properties of objects of another class.
9. **Abstraction:** The process through which, a class is developed in terms of its functionality and interface, rather than its implementation details.

Though there are some traditional object oriented metrics like, Line of code (LOC), Comment Percentage (CP) etc, but the two basic object oriented metrics are CK metrics and MOOD metrics, our discussion will include most of the metrics from these two and some of the others:

1. **Number of Attributes (NOA)**: The value of NOA is the total number of attributes defined in a class.
2. **Number of methods (NOM)**: The value of NOM is the total number of methods defined in a class.
3. **Depth of Inheritance tree (DIT):** Its value is defined by the length of the node from the root of the tree.

4. **Coupling between Objects (CBO)**: Its value for a class is equal to the number of other classes to which it is coupled.
5. **Number of children (NOC)**: Its value for a class is equal to the number of directly inherited sub-classes of the class.
6. **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.
7. **Method hiding factor (MHF):** It is 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[13].
8. **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.
9. **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.
10. **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.
11. **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[14].

3.2. Analysis of Object Oriented Metrics

Now, based on the class diagram shown in fig.1, we will analyze the values of the object oriented metrics discussed above.

To represent the values of the metrics in respect of the above classes, we will use the following table:

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 {R_i}, where where { R_i}
= set of methods called by method i and {M}
= set of all methods in the class.

Table 3.1: Metrics of RSA digital signature

OO Metrics	*Classes of proposed system*			
	RSA1	*RSA2*	*Dev*	*Lrnr*
NOA	3	0	5	1
NOM	0	2	2	1
CBO	2	2	0	0
DIT	0	0	1	1
NOC	2	2	0	0
RFC	0	5	4	3

Now, we will plot the values of the above data analysis of object oriented metrics to draw graphs and make some discussion on these values in tabular form below.

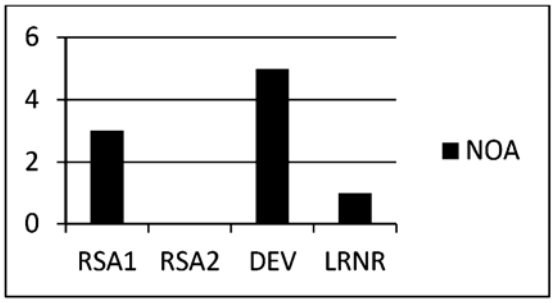

Figure 3.1: NOA

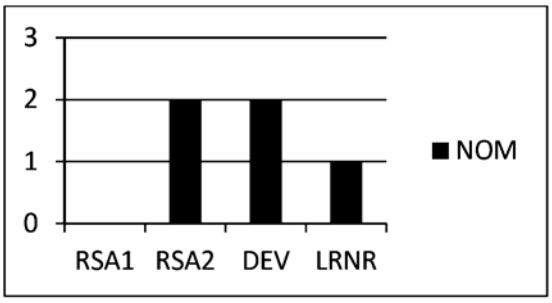

Figure 3.2: NOM

Fig. 3.1 and fig.3.2 shows the total number of attributes and the total number of methods per class respectively by using graphs. These values are helpful to estimate the time and cost management of any system. If these values are kept low, which means the proposed model is easy to maintain. Here the maximum value of NOA is 5 and the maximum value of NOM is 2, which means it is easy to maintain.

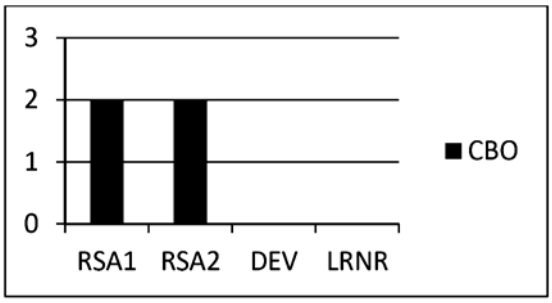

Figure 3.3: CBO

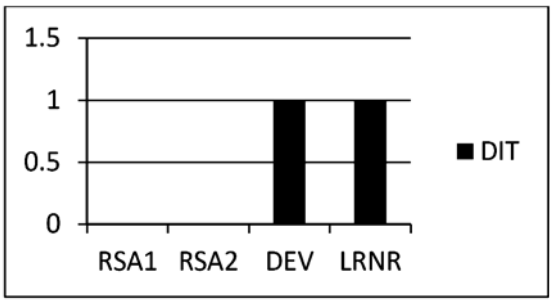

Figure 3.4: DIT

Fig.3.3 shows the value of CBO, that means coupling between objects. As we know in software engineering, better implementation can be achieved by low coupling. In our proposed model, the value of CBO is also kept low.

Fig.3.4 shows the value of the DIT graphically. Here, RSA1 and RSA2 are two base classes, so, their value is 0 and for the other two classes this value is 1. DIT is used to represent the complexity level of any design. Here this value is kept low, which means the system is easy to simple.

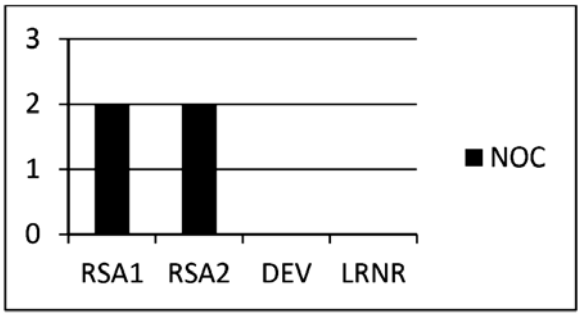

Figure 3.5: NOC

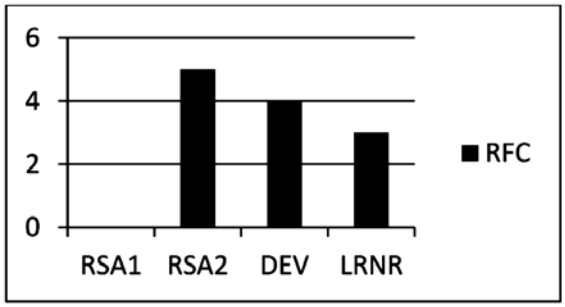

Figure 3.6: RFC

Fig.3.5 shows the values of NOC of our proposed model. Here, the value is 2, which is quite ok.

Fig 3.6 shows the graphical representation of the value of the RFC metric. The increasing of this value makes the system difficult to understand and if it keeps vary low, and then polymorphism increases. Here, it is an optimal value of RFC.

Now, we will find out the values of the metrics which is under MOOD metrics based on the above class diagram.

1. Equation for **MHF (Method Hiding Factor)**

$$= \sum_{i=1}^{TC} M_h(C_i) / \sum_{i=1}^{TC} M_d(C_i) \text{ //TC means total number of class}$$

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 3.2
MHF metrics of proposed system

	Classes of proposed system				
	RSA1	*RSA2*	*Dev*	*Lrnr*	*Summation(∑)*
$M_h(C_i)$	0	0	0	0	0
$M_v(C_i)$	0	2	2	1	5
$M_d(C_i)$	0	2	2	1	5
MHF			0/5 = 0		

Table 3.2 shows the value of the MHF metric under MOOD metric of our proposed system, which is low, that means insufficiently abstracted implementation, which makes our design very simple.

Equation for **AHF (Attribute Hiding Factor)** $= \sum_{i=1}^{TC} A_h(C_i) / \sum_{i=1}^{TC} A_d(C_i)$

$A_d(C_i) = A_v(C_i) + A_h(C_i)$,

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 3.3
AHF metrics of proposed system

	Classes of proposed system				
	RSA1	*RSA2*	*Dev*	*Lrnr*	*Summation(∑)*
$A_h(C_i)$	0	0	4	0	4
$A_v(C_i)$	3	0	1	1	5
$A_d(C_i)$	3	0	5	1	9
AHF			4/9 = 0.44		

In the table 3.3, we analyze the value of the AHF metric in respect to our proposed model. If this value is 100%, then 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 (Method Inheritance Factor)**

$$= \sum_{i=1}^{TC} M_i(C_i) / \sum_{i=1}^{TC} M_a(C_i)$$

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 3.4
MIF metrics of proposed system

	Classes of proposed system				
	RSA1	*RSA2*	*Dev*	*Lrnr*	*Summation(∑)*
$M_d(C_i)$	0	2	2	1	5
$M_i(C_i)$	0	0	2	2	4
$M_a(C_i)$	0	2	4	3	9
MIF			4/9 = 0.444		

From table 3.4, we can see that the value of MIF of our proposed system is 0.316, which is not too much high or too much low, which 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 attributes available,

$A_d(C_i)$ = number of attributes defined and

$A_i(C_i)$ = number of attributes inherited

Table 3.5
AIF metrics of proposed system

	Classes of proposed system				
	RSA1	*RSA2*	*Dev*	*Lrnr*	*Summation(∑)*
$A_d(C_i)$	3	0	5	1	9
$A_i(C_i)$	0	0	3	3	6
$A_a(C_i)$	3	0	8	4	15
AIF			6/15 = 0.4		

Table 3.5 shows the AIF value of our proposed system, which is 0.4. This value is not too much high or not too much low, which indicates that our system is quite ok.

Coupling factor (CF) = Actual coupling/Possible coupling

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

Table 3.6
CF metrics of proposed system

Classes	*Classes of proposed system*				
	RSA1	*RSA2*	*Dev*	*Lrnr*	*Summation(∑)*
RSA1	X	0	1	1	2
RSA2	0	X	1	1	2
Dev	1	1	X	0	2
Lrnr	1	1	0	X	2
TC = 3	Total number of coupling = 8 Possible number of coupling = 12				
CF	8/12 = 0.67				

From the above table 3.6, 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.

3. CONCLUSION

This paper contains the object oriented metric based analysis for improvement of quality and authenticity of the system by using signature generation and verification along with RSA digital signature regarding the transmission of mark sheet from developer to learner. These analyses are based on Chidamber and Kemerer metrics (CK metrics) and Metric for Object Oriented Design metrics (MOOD metrics). Similar kind of transmissions of other documents in e-learning system like study material, certificates are also possible using this proposed model. More authenticity can be achieved by implementing digital watermarking and digital right management, which is beyond the scope of this paper.

REFERENCES

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

[2] http://www.drdobbs.com/rsa-digital-signatures/184404605

[3] https://en.wikipedia.org/wiki/Digital_signature

[4] Dr. R.V. Krishnaiah, B.S. Prasad, "Analysis of object oriented metrics", ,IJCER, vol-2,issue-5, issn-2250-3005(online),pp:1474-1479

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

[6] Karforma S. and Banerjee S., "Object oriented modeling of RSA digital signature for security in e-learning", IJATES, ISSN: 2348-7550, vol-02, special issue-01, September-14, pp-283-290

[7] Edward V. Berard, "Metrics for Object-Oriented Software Engineering", The Object Agency, Inc.

[8] Karforma S. and Banerjee S, "Object oriented metric based analysis of ElGamal digital signature algorithm for study material authentication", IJSTM, vol-04(spl-01), sept-15, pp: 522-530

[9] Karforma S. and Banerjee S, "Object oriented metric based analysis of DES algorithm for secure transmission of mark sheet in e-learning", IJCSE, ISSN: 2347-2693, vol-4,(spl-01), 2016, pp:93-98

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

[11] Balagurusami E., "Object oriented programming with C++", Tata McGraw Hill, New Delhi, 2006

[12] A. Kamandi, "Object Oriented Metrics", Sharig University of technology, spring 2007

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

[14] Rajib Mall, "Fundamental of Software Engineering", Prentice hall of India Pvt. Ltd., 2004