ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Confidentiality Estimation Model: Fault Perspective

Anshul Mishra
Research Scholar
School of Computer Application
Babu Banarasi Das University
Lucknow, India

Dr. Devendra Agarwal
Director (Engg.), BBDNIIT
Babu Banarasi Das University
Lucknow, India

Dr. M. H. Khan
Professor, Department of Computer Science & Engineering
Institute of Engineering and Technology, Sitapur Road
Lucknow, India

Abstract: Securing software is an on-going procedure which never closes. Software security highlights must be coordinated in each level of software development life cycle. Confidentiality is key security factor to quantity of object oriented software at an early stage of software development process at design phase for high secure product. Metric based model for **"Confidentiality Estimation Model"** has been proposed by establishing the correlation between confidentiality and fault attributes constructs. Object oriented design metrics are used to quantify security attributes. Later confidentiality estimation model is empirically validated and statistical significance of the study considers the high correlation for model acceptance. The aim of this research work is to encourage researchers and developers for inclusion of the **"Confidentiality Estimation Model (CEM^{OODF})"** to access and highly secure at design time.

Keywords: Software Security, Confidentiality, Fault Attributes, Object Oriented Design Characteristics, Design Metrics

I. INTRODUCTION

A need to protect some assets, information, data and software creates a demand for security. Security is a peak significant quality attribute in the pitch of software engineering [1]. The software security concentrates on the exertion and cost spent in lateral stages is significantly greater than the initial phase of software development process. Software security vulnerabilities emerge from various poor improvement practices, new method of attacks, unsecured connections between frameworks and poor design. Confidentiality is a one of the most vital attribute of software security for conveying high secure software.

It is also an important factor to security estimation of object oriented software at an early phase of software development life cycle. Design time is most appropriate phase to estimate security of software, because this phase is the first step towards problem domain to solution domain [3]. It always supports developer for improved software design at early stage of software development life cycle, means design phase has positive impact on the overall security cost and effort. There is a need for software engineers to understand how various components of a design interact in order to secure and enhance the reliability of software during software development process. According to statistical information, more than 80% of all software delivered in the United States is not reviewed for defects, at a cost to the state economy of tens of billions of dollars each year [4]. Encrypted form of data is also used to maintain the confidentiality of data when it's transferred or stored [2]. ISO/IEC 9126 [5] defines security as follows: "The capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access to them." This study aims to produce a model to deal in a word of fault factors at design time and quantify security. This paper is organized in such a, approach that it initially describes the confidentiality security factor and describes **Confidentiality Estimation Model (CEM^{OODF})** development with their mapping, highly correlation establishment and also highlights data for statistical significance of model that are important for further study. The presented model has been validated and the research paper concludes with industry utility for project ranking in conclusion section.

II. RELATED WORK

A. Fault at Design Phase

Software fault prediction is the most efficient methodology to improve the quality of the software. To enhance the quality it is essential to discover the error or fault as quick as could be expected under the possible. To decide and progress the development of product there are diverse predict methodologies are accessible like expectation, effort prediction, and software fault prediction (SFP) and security prediction [8]. The success of the software product depends upon the quality, which measures how good the product is designed and furthermore the necessity configuration meet the final output. Also it is related with cost effective and security predictions.

Fault can be start at different abstraction levels depending on the data information available about the framework [6]. Software security is the main aspect to predict the fault in object oriented based software systems. Predict the fault treated as factors at initial stage of design. Java based object oriented software system fault prediction response is based on density based spatial clustering of application with noise [7].

The estimation of the fault should be done in the initial stage to effectively allocate effort for fixing the faults issues. A variety of researchers in the area suggested that fault measurement should be done at design phase and their view is summarized in table1.

TABLE I. A Critical Look of Fault Consider by Various Expert

Study/Author	*Year*	*Software Development Phase*
K. Emam & W. Melo [10]	2001	Design Phase
G. Denaro & M.Pezze[13]	2002	Design Phase
T. m. Khoshgoftaar & N. Seliya [14]	2003	Early phase in development life cycle
J. Aidemark [6]	2005	Design Phase
S. Kaur & D. Kumar[7]	2007	Design Phase
Dr. S. Ravichandran [11]	2007	Design Phase
R. A. Khan & K. Mustafa [9]	2008	Design Phase
R. Sharma, N. Budhija & B.singh [15]	2012	Design Phase
P. Mittal, S. Singh and K. S. Kahlon [16]	2013	Design Phase
Anjali Verma [12]	2015	Design Phase

III. CONFIDENTIALITY: SECURITY FACTOR

Security Estimation is necessary at early stage of software development process. A number of methods and techniques are used for security estimation at the time of delivered the software product. Estimation analysis always provides the accurate assessment of software product and their limitations. Confidentiality requires that information be kept private [2]. The software product is said to be highly secure if it posses security property including confidentiality. In effect, access to important information should be restricted only to those individuals who have a specific require to see or use that information.

To assurance under the confidentiality, communications channels must be properly monitored and controlled to prevent unauthorized access. The overall purpose of the software is to deliver secure software that is efficient in operation, easily accessible to user within specified time and given budget. Confidentiality is one of the decidedly important security indicators of object oriented software.

Here research is needed to develop a structured scientific approach to ensure that software is secure, stable and high quality. Consolidated charts for security factors identified by various experts are concluded in table2. It is visibly evident from this table that Confidentiality, Integrity, Availability, Durability and Authorization are common accepted factors at design phase.

TABLE II. A critical look of commonly accepted security factors by Various Experts

Security Factor	*Confidentiality*	*Availability*	*Integrity*	*Authorization*	*Durability*	*Stability*
Experts / Study						
A. P. Martin 2006 [19]	✓	✓	✓			
S. Chandra & R. A. Khan 2010 [21]	✓	✓	✓	✓	✓	
S. Jain & M. Ingle 2011 [27]	✓	✓	✓			
S. Chandra, R. A. Khan & A. Agrawal 2010 [22]	✓	✓	✓	✓		
R. A. Khan & S. A. Khan 2012 [17]	✓	✓	✓			
I. A. Mir & S.M.K Quadri 2012 [26]	✓	✓	✓	✓		
N.Parveen, R. Beg & M. H. Khan 2014 [23]	✓	✓	✓			
R. Kumar 2014 [20]	✓	✓	✓		✓	
Rajeev & S.A. Khan 2015 [1]	✓	✓	✓	✓	✓	
K. Sahu & R. Shree 2015 [18]	✓	✓	✓	✓		√

A. Correlations among Fault Factor and Security Factors

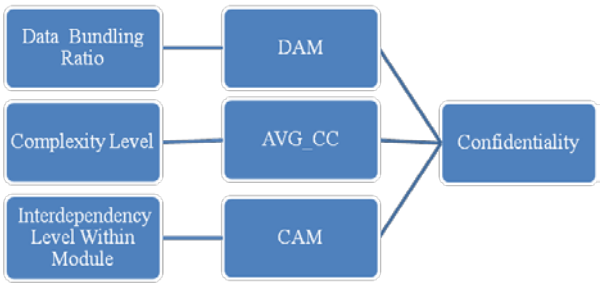

Figure. 1. Confidentiality Estimation Framework of Object Oriented Design

The figure1 describes the estimation process of confidentiality model in order to establish a multivariate model for security and OOD faults constructs. The values of these metrics can be easily identified by class diagram metrics. This metrics will play the role of independent variables while confidentiality will be taken as dependent variable. The quantifiable assessment of confidentiality is very helpful to achieve security index of software design for secure the product within time and given budget.

B. Proposed framework implemented the following steps

I. Indentify the confidentiality as a security factors
II. Indentify the fault factors at design Phase
III. Identified best suited OOD metrics for fault factors
IV. Correlation Establishment
V. Model Development for quantifying confidentiality
VI. Empirical validation for developed model

Correlation among fault factors and security factors has been established and shown in figure-1. It was observed that each fault factors affect certain security factors. As per the values of selected independent variables, namely DAM (Data Access Metrics), AVG_CC (Cyclomatic Complexity), CAM (Cohesion among method), the values of dependent variable 'Y' can be found out by using the 'Confidentiality Estimation Model.

C. Confidentiality Estimation Model Development

It is evident from literature survey that confidentiality is not a new term; rather it has been in discussion among the industry professionals at various forums, but there is no commonly accepted comprehensive and complete model or framework available to estimating the confidentiality through given fault factors at design phase, that motivate to develop **'Confidentiality Estimation Model' (CEM^{OODF})** using fault attributes and approach based on its internal design property or design diagram. This model used the low level design metrics namely Data Access Metrics, Cyclomatic Complexity, Cohesion Among Methods, to describe a range of measurement for software and defined in terms of design characteristic and also helpful for quantitative assessment of degree to which system, component or process hold a given attribute. In order to establish a model for confidentiality, multiple linear regression techniques have been used .The proposed multivariate model takes the following form:

$$Y=\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X3 + \ldots\ldots \alpha_n X_n \qquad (1)$$

Where

- Y is dependent variable
- X1, X2, X3 ... Xn are independent variables.
- $\alpha_1, \alpha_2, \ldots \alpha_n$ are the regression coefficient of the respective independent variable.
- α_0 is the regression intercept.

The data used for establishing confidentiality model is taken from [24] that have been collected through large commercial object oriented systems. The relationship between security factor and fault factors has been established as depicted in Figure 1. As per the mapping, Metrics are selected from [25] as independent variable to build up the confidentiality Estimation model via SPSS, values of coefficient are calculated and confidentiality model is formulated as given below.

Using SPSS software values of all independent variables (metrics), regression intercept and coefficient of the respective independent variables are calculated. Confidentiality values have selected from [17] as standard values. On the basis of this approach, the multiple linear regression confidentiality models have been developed that is given in equation (2).

Table iii. Confidentiality computed table

Project	*Standard Confidentiality*	*DAM*	*AVG_CC*	*CAM*
P_1	0.477	0.50	0.667	0.666
P_2	0.763	0.00	1.6667	0.555
P_3	0.533	0.250	1.100	0.228
P_4	0.504	0.9166	1.7083	0.185

Confidentiality= 0.3220 - 0.2140* DAM + 0.2000* AVG_CC + 0.1920* CAM (2)

D. Statistical Significance of Independent Variables

Table IV. ***Statistical Significance of Confidentiality Model***

Descriptive Statistics			
	Mean	*Std. Deviation*	*N*
Calculate Confidentiality	.59579	.118152	10
DAM	.54928	.484175	10
AVG_CC	1.75880	.758753	10
CAM	.21817	.099777	10

The descriptive table IV is very important for further research work. It gives the valuable record of descriptive statistics that are mean, standard deviation and number of samples selected for model validation.

Table V. ***Model Summary for Confidentiality Model***

Model Summary				
Model	*R*	*R Square*	*Adjusted R Square*	*Std. Error of the Estimate*
1	.999[a]	.998	.997	.006223
a. Predictors: (Constant), DAM, AVG_CC, CAM				

Summary table V for confidentiality Estimation Model proves that all the three selected metrics are statistically significant at confidence level of 95%.

IV. EMPIRICAL VALIDATION OF CONFIDENTIALITY MODEL

This section of work proves that how significant proposed study, where metrics and model are able to estimate the confidentiality security index of object oriented design at design time. The empirical validation is important phase of research to evaluate the proposed confidentiality model for high level acceptability and appropriate execution. Statistical analysis is the best practice for claiming the model acceptance [19]. To justify claiming approach for acceptance of model, an experimental validation of the proposed confidentiality estimation model through fault attributes at design time has been carried out using samples.

A. Data Set for Ten Projects

The data is taken from this model is from various versions windows application frameworks.48 releases of 15 open source projects were investigated: Apache Ant (1.3 – 1.7) [24]. Five of them are custom build solutions that had been already successfully installed in the customer environment. In view of this fact, an experimental validation of the proposed model for confidentiality evaluation has been carried out using sample tryouts. In order to validate proposed confidentiality estimation model, the value of metrics are available by using [24] data set for following 10 projects in table VI.

Table VI. ***Confidentiality Data Table***

Project	*DAM*	*AVG_CC*	*CAM*	*Known Index*	*Calculate Index*
P_1	.826	2.500	.207	.678	.685
P_2	1.000	1.634	.081	.432	.448
P_3	.000	1.273	.318	.635	.638
P_4	.667	.950	.103	.379	.389
P_5	1.000	2.304	.143	.583	.596
P_6	1.000	3.238	.214	.813	.797
P_7	.000	.909	.288	.547	.559
P_8	1.000	2.130	.145	.521	.541
P_9	.000	1.364	.362	.612	.664
P_{10}	.000	1.286	.321	.636	.641

B. Statistical Test of Confidentiality Estimation Model

It is essential to test the validity of proposed model for acceptance. 2 sample t tests apply for check the impact between standard confidentiality and calculated confidentiality. 2t-test is handy hypothesis tests in statistics when compare means.

Table VII. ***2 t- test between Standard Confidentiality and Calculate Confidentiality***

Paired Samples Statistics					
		Mean	*N*	*Std. Deviation*	*Std. Error Mean*
Pair 1	*Calculate Confidentiality*	.59579	10	.118152	.037363
	Standard Confidentiality	.58364	10	.123599	.039086

Null hypothesis (H0): There is no significant difference between Standard Confidentiality and Calculate Confidentiality

H0: μ1-μ2 = 0

Alternate hypothesis (HA): There is significant difference between Standard Confidentiality and Calculate Confidentiality.

HA: μ1-μ2 ≠ 0

In the above hypothesis μ1 and μ2 are treated as sample means of population. Mean value and Standard Deviation value have been calculated for specified two samples and represented in table 7. Correlation comes out to be 0.991, that shows the standard confidentiality and calculated confidentiality is highly correlated. The hypothesis is tested with zero level of significance and 95% confidence level. The p value is 0.052. Therefore alternate hypothesis directly discards and the null hypothesis is accepted. The developed equation used for confidentiality estimation is accepted.

V. CONTRIBUTIONS ANALYSIS AND CRITICAL FINDINGS

This study shows the importance of confidentiality in general and as a key factor to software security for producing high class secure software at early stage of design phase. As a result we can conclude without any loss of generality that confidentiality Estimation model is essential and applicable in the security estimation.

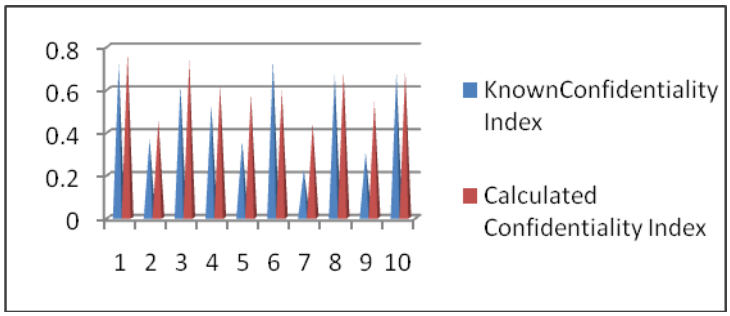

Figure. 1. Pyramid Graph Comparing between Known confidentiality and Calculated Confidentiality Index.

Confidentiality Estimation Model provides a confidentiality indexing (RI) benchmark for other researchers and designers. Developed model also provides confidentiality indexing (RI) for Industry project ranking.

VI. CONCLUSION

This paper has developed an efficient model "Confidentiality Estimation Model" through object oriented design constructs using the technique of multiple linear regressions. Statistical analysis shows that this model is statistically very much impact and acceptable. This model estimates the security in term of fault design factors which are affected to security factors. This paper also validates the quantifying ability of presented model. That Empirical validation analysis on this research work proves that Confidentiality Estimation Model is standard approach, more practical in nature and helps the software industry in project ranking.

REFERENCES

1. R. Kumar, S. A. Khan & R. A. Khan, "Revisiting Software Security: Durability Perspective", International Journal Of Hybrid Information Technology, Vol. 8, No. 2, pp 311-312, 2015.
2. M.Dowd, J. M. Donald & J. Schuh, "The Art of Software Security Assessment: Identifying and Preventing software vulnerabilities", Addison Wasley Professional, Nov 10, 2006.
3. S. A. Khan & R. A. Khan, "Integrity Estimation Model for Object Oriented Design", ACM, SIG Soft,Vol. 37, No. 2, March 2012.
4. U. Chhillar & S. Bhasin , " A New Weighted Composite Complexity Measure for Object-Oriented Systems", International Journal of Information and Communication Technology Research Volume 1 No. 3,pp: 101-108, July 2011.
5. P. Syväniemi, S. Purhonen, A. Ovaska, E. Kuusijärvi & J.; Evesti, " A. Situation-Based and Self-Adaptive Applications for Smart Environment", J. Ambient Intelligence Smart Environ, 2012.

6. J. Vinter, J. Aidemark, D. Skarin, R. Barbosa, P. Folkesson & J. Karlsson, " A. Situation-Based and Self-Adaptive Applications for Smart Environment", Technical Report No. 05-07, Division of Computer Engineering, CHALMERS UNIVERSITY OF TECHNOLOGY, Göteborg, Sweden, 2005.
7. S. Kaur, and D. Kumar, "Software Fault Prediction in Object Oriented Software Systems Using Density Based Clustering Approach", International Journal of Research in Engineering and Technology (IJRET) Vol. 1 No. 2, March, 2012.
8. L. C. Briand, J. Daly, V. Porter & J. W, "Predicting Fault-Prone Classes with Design Measures in Object-Oriented Systems", Proc. of the 9th Int'l Symposium on Software Reliability Eng., Paderborn, Germany, pp.334-343, 1998.
9. R. A. Khan & K. Mustafa, " Fault Proneness Model For Object Oriented Software : Design Phase Perspective, Information Technology Journal, 2008.
10. K. El. Emam & W. Melo, " The Prediction of Faulty Classes Using Object Oriented Design Matrics", The Journal of System and Software, Elsevier, 2001.
11. DR. S. Ravichandran, "Design And Development Of Software Fault Prediction Model To Enhance the Software Quality Level", International Journal of Information Technology and Management Information Systems (IJITMIS), Vol. 1, pp. 01–06, 2007.
12. A. Verma, "Software Fault Prediction for Object Oriented System", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, Issue 3, March 2015.
13. G. Denaro & M. Pezze, " An Emparical Evaluation of Fault Proneness Model", ICSE, 2002.
14. T. m. Khoshgoftaar & N. Seliya, "Fault Prediction Modeling for Software Quality Estimation: Comparing Commonly Used Techniques", Journal of Empirical Software Engineering, pp 255–283, 2003.
15. R. Sharma, N.Budhija & B. Singh, "Study of Predicting Fault Prone Software Modules", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 2, February 2012.
16. P. Mittal, S. Singh and K. S. Kahlon, "Empirical Model for Fault Prediction using Object-oriented metrics in Mozilla Firefox", International Journal of Computer Technology & Research, pp. 151- 161, 2003.
17. S. A. Khan & R. A. Khan, "Confidentiality Quantification Model for Object Oriented Design", International Journal of Information and Education Technology &, SIG Soft, Vol. 37, No. 2, March 2012.
18. K. Sahu & R. Shree, "Stability: Abstract Roadmap of Software Security", American International Journal of Research in Science, Technology, Engineering & Mathematics, 2015.
19. A. P. Martin & D. Khazanchi, "Information Systems and Quantitative Analysis Faculty Proceedings & Presentations", Department of Information Systems and Quantitative Analysis, Information Availability and Security Policy, 2006.
20. R. Kumar, S. A. Khan & R. A. Khan, "Software Security Testing A Pertinent Framework**" ,** Journal of Global Research in Computer Science, Vol. 5, No. 3, March 2014
21. S. Chandra & R. A. Khan, "A Methodology to Check Confidentiality of a Class Hierarchy", Elsevier, Vol. 10, Issue 3, 2010.
22. S. Chandra, R. A. Khan & A. Agrawal, "Software Security Factors in Design phase", International Conference on Information Systems, Technology and Management, Springer, pp 339-340, March 2009.
23. N. Parvee, R. Beg & M. H. Khan, "Integrating Security and Usability at Requirement Specification Process", International Journal of Computer Trends and Technology, Vol. 10, Apr 2014.
24. M. Jureczko & L. Madeyski, "Towards identifying software project clusters with regard to defect prediction", IEEE, 2010.
25. S.P.Kadam & S. Joshi, "Secure by Design Approach to Improve Security of Object Oriented Software", IEEE, 2015.
26. I. A. Mir & S.M.K Quadri, "Analysis and Evaluating Security of Component Based Software Development: A Security Metrics Framework", I. J. Computer Network and Information Security, pp 21-31, 2012
27. S. Jain & M. Ingle, "A Review of Security Metrics in Software Development Process", International Journal of Computer Science and Information Technologies, Vol. 2, 2011.

IBLIOGRAPHY OF AUTHORS

	Anshul received the MCA degree from Dr. R. M. L. Avadh University, Faizabad, in 2008. He is enrolled as research scholar in BBDU , Lucknow. His research interests include Software testability, Software Quality Estimation, Data Dictionary.
	Dr. Devendera Agarwal is currently working as Prof. & Director (Engg.) at BBDNIIT (BBD Group), Lucknow. He has over 17 years of teaching & 5 years of industrial experience. He has done his B.Tech in Computer Science from Mangalore University in 1993, M.Tech from U.P. Technical University, Lucknow in 2006, and Ph.D. from Shobhit University, Meerut in 2013. He has over 10 research papers with 4 students pursuing Ph.D.
	Dr. M. H. Khan, Professor, Department of Computer Science and Engineering at IET, Lucknow UP. Obtained his MCA degree from Aligarh Muslim University (Central University) in 1989 .Later he did his PhD from Lucknow University. He has around 25years rich teaching experience at UG and PG level.