Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 9 (2014) pp. 009-018

An Empirical Validation of Object Oriented Design Security Quantification Model

Suhel Ahmad Khan[1], Raees Ahmad Khan[2]

[1, 2] Department of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University)
Vidya Vihar, Lucknow 226025, India
ahmadsuhel28@gmail.com[1], *khanraees@yahoo.com*[2]

Abstract: **Software security is a multifaceted and comprehensive property, which can be properly captured only through many different quality attributes. The idea of software security covers both conventional security attributes and classical dependability attributes. Software security involves multiple attributes such as authentication, authorization, confidentiality, integrity, availability and non repudiation. The values of security are not identified by single step. It can be measured through the whole development process by collective values of its attributes. Security quantification models have been developed on the basis of established relationship between complexity factors and security attributes and validated through proper data set for model acceptance. The aim of addressing security at design phase is to defend software from the external threats and attacks.**

Keywords: Software Security, Security Quantification Model, Design Complexity.

I. Introduction

Security estimation of software may heavily affect security of the final product. The analysis of security parameters and their impact on security will ease up to uncover the strengths and weakness of the software and provide the basis for carrying out cost and benefit analysis[1]. Security flaws are a part of design that can cause the system to violate its security requirements and result in unauthorized disclosure, destruction or modification of data[2]. In particular, over 90% of software security incidents are caused by attackers exploiting known software defects. Failures due to security breaches can endanger human lives and environments, implying serious damage to industrial and social infrastructures, jeopardizing confidentiality and privacy, or undermine the viability of whole business sector. Therefore developing high quality and secure software applications is essential in maintaining a competitive edge in today's market place[1, 2]. The design of secure software is not an easy task. It certainly requires deep understanding of various aspects of security, like security measurement, security categories, security policies etc. Security represents one of the most interesting characteristics of software products and along with it several other measures also depend on security which reflects a new face of the software. Security team can collaborate during design phase to make software secure. Using the concept of software security estimation during development of software, security can be measured by analyzing the design activities, measurement of security attributes and its impact on software. A quantitative approach can be much better than conceptual method to develop and deliver a truthful technique which can assess the actual level of security measure in newly developed software as well as in existing. Without quantification nothing can be predicted. Therefore, quantification of security has become an urgency to predict the immunity and resilience of the software.

II. Security at Design Phase

E-development is responsible to facilitate humanity in a better way, but due to increased network connectivity, bigger system support, complex software design software's are sometimes not performing their responsibilities according to their intended functionality. Increasing complexity, connectivity and extensibility of ever developing software security technologies has unique challenges for organizations to protect their resources. Ineffective software security management is responsible for financial loss as well as reputation damage for software security industries. These are indirectly associated with software organization and will severely impact performance and their market valuation[3, 4]. Security is multidimensional, emergent and irreducible concept and the unfortunate side effect of security produces inherent design complexity[5]. *Security risks can be hidden in the jungle of complexity, not coming in the light until it is too late*[6]**.** Quantitative evaluation of security is a vital process.

The latest work related to security quantification was published by Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz, of Horst Görtz Institute for IT-Security and Ruhr-University Bochum, Germany. The work entitled 'Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns' was published by ACM in November 2013 highlighting widely adopted Graphical passwords system for Android Unlock Pattern. Graphical passwords were proposed as an alternative to overcome the inherent limitations of text-based passwords[20]. A Ph. D thesis entitled "A framework and theory for cyber security assessments", submitted by Teodor Sommestad, in 2012 contributed a modeling framework and a theory to support cyber security vulnerability assessments. It has a particular focus on SCADA systems. The result is a decision support tool called the Cyber Security Modeling Language

(CySeMoL). This tool produces a vulnerability assessment for a system based on an architecture model of it[21]. In 2011, Jana Sedláčková, in his research contribution entitled "Security Factors in Effort Estimation of Software Projects" deals with problems related to an effort estimation of the software projects which are connected to the development of secured products. The aim of this contribution is to present an option of an effective consideration of the effort which is connected with the software products development[22]. In 2010, Shari Lawrence and Robert K. Cunningham emphasized the need and importance of relative metrics to make a significant progress towards rigorous, practical and effective security measurement[23]. Shari L. Pfleeger, in 2009, made an effort in showing how some existing metrics can help depict the system's immunity and resilience[24].

The current study produces major contribution in the area of security quantification including many macro level direct or indirect findings. The estimation practice at early stage is beneficial for secure software development. The object oriented technology provides flexible development environment due to its simple designing nature. Design phase that provides the complete idea of design by preparing the blue prints or comprehensive detailing at early stage is highly responsible for the viable changes that required for secure design. A security quantification models through complexity revealed many things including the persistent need for good security estimation models and the non availability of any standard methodology/framework for such development. Researcher made an attempt for the development of such models with requisite activities having sound bearing in the literature and context.

III. Security and Complexity

According to security expert Guru Gary McGraw "Software security is about understanding software-induced security risks and how to manage them. Good software security practice leverages good software engineering practice and involves thinking about security early in the software lifecycle, knowing and understanding common problems (including language-based flaws and pitfalls), designing for security, and subjecting all software artifacts to thorough objective risk analyses and testing". They recommend software security as a knowledge intensive field[4,5.]

A. Security Attributes

The growing computing system becomes larger and complex and more difficult to handle and manage. Without any qualitative or quantitative assessment, it's not worthy effort to explore the issues of security, its attributes at design time. Security design flaws could be exposed using STRIDE approach. Different types of threat issues like spoofing, tampering, information disclosure, denial of service and elevation of privilege are directly associated with security properties like confidentiality, integrity, availability, authentication and authorization. This approach facilitates to establish a relationship between security attributes and the components for susceptibility to the threat in application. This may provide a strong basis to select security attributes to discover quantification models for security evaluation with design complexity keeping separate threat issues in mind[7]. The identified security attributes for this research work is confidentiality, integrity, availability, authentication and authorization. Confidentiality protects data and information from unauthorized user access. Data Integrity of software insures that the protected data has been modified only by authenticated and authorized person. Availability of the system is to ensure that the services for authenticated and authorized user will be available. Therefore, Confidentiality, Availability, Integrity, authentication and authorization has been taken as a commonly accepted set of security factors to be addressed while quantifying security during the course of study[7, 8.]

B. Complexity Factoring

Abstraction could be found in many more different ways than it can comprehend at one time. By hiding the inside view of abstractions, design complexity is managed by encapsulation. A hierarchy is often manufactured by a set of abstraction and by identifying these understanding of problem got very simplified for us. The definition of hierarchy comes: Hierarchy is a ranking or ordering of abstractions. The two most important hierarchies in a complex system are its class structure (the "is a" hierarchy) and its object structure (the "part of" hierarchy)[6, 9]. "The more complex the system, the more open it is to total breakdown". The failure to master the complexity of software results in late, over budget, and deficient in their stated requirements in accordance to projects. For example, a CPU typically comprises of primary memory, an arithmetic/logic unit (ALU), and a bus to which peripheral devices are linked. Each of these parts may be further divided. An ALU further split into registers and random control logic, which themselves are constructed from even more primitive elements, such as NAND gates, inverters, and so on. As Brooks suggests, "The complexity of software is an essential property, not an accidental one". It observes that this inherent complexity derives from four elements: the complexity of the problem domain, the difficulty of managing the development process, the flexibility possible through software, and the problems of characterizing the behavior of discrete systems[6].

In the line of the nature of this complexity, it concludes that there are five attributes common to all complex systems. Till the lowest level of elementary components a complex system is made of interrelated subsystems that together form subsystems, which intern made the complexity hierarchical. The pursuit of the nature of the primitive components of a complex system, the choice of what components in a system are primitive is relatively arbitrary and is largely up to the discretion of the observer of the system. Intra component linkages are generally stronger than inter component linkages. This fact has the effect of separating the high-frequency dynamics of the components. This involves the internal structure of the components from the low frequency dynamics involving interaction among components. Many complex systems are implemented with an economy of expression. This further stipulated that complex systems have common patterns[6, 10]. A simple system that worked is further makes the ground for a complex system design from scratch is never conducive and cannot be made to work. A Simple system is the platform from the beginning in the process start over. As the systems evolve, objects that were once considered complex

become the primitive objects on which more complex systems are built. These primitive objects cannot be crafted correctly. This should be use in accordance of the first and then should be improved by the real behavior of the system[10, 11]. The observation regarding identification of complexity factors is done by regress analysis of secure design best practices in the light of complexity. An effort is made to identify complexity factors at design phase with strong relation with object oriented design parameters and given security attributes. The recognized factors of complexity are fully well-suited with object oriented design parameters and having a strong impact to minimize/control the design effect. The identified factors are mentioned as follows:

- Coupling Function
- Total Supporting Services
- Minimum Privilege between Services and Requests
- Maximum Depth of Hierarchy
- Higher Level of Abstraction

C. Establishing Correlation between Complexity Factors and Security Attributes

Security is multidimensional attribute. The values of security are not identified by single step. It can be measured through the whole development process by collective values of its attributes. Computer security is frequently associated with three core pillars which can be conveniently summarized by the acronym 'CIA' but without including authentication and authorization, design security of object oriented software's could not properly estimated. The established correlation between complexity factors with security attributes are depicted in figure 1. The details of security attributes which can be included for research are summarized as follows.

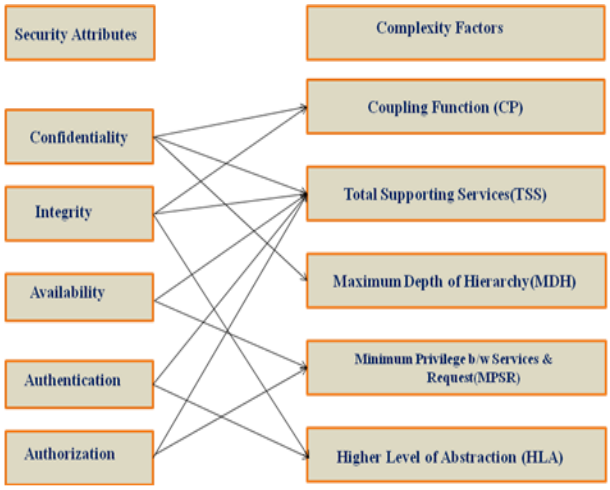

Figure 1. Correlation b/w Complexity and Security Attributes

D. Establish Measures and Strategy

The correspondence and mapping between the identified security, complexity and design constructs revealed that all metrics have relevance with respect to a class. This indicates that 'class' is the fundamental concept of object oriented software and hence all the metrics should eventually conduct measures taking class as a basis. The proposed metrics will be used to compute security with the help of complexity using the class diagram. The strategy is to first identify design constructs having relation with complexity factors and identified metrics. Then complexity factors having impacts on security attributes are computed and correlated with security to derive a single integrated measure for object oriented design security. Total supporting services is union of behavior of class elements and efforts to provide protection to the basic components of object oriented design. To gain maximum strength of protection it is mandatory to keep design complexity low by preventing unnecessary privilege grant to services. Privileges should be minimal according to interaction between services and requests. Most of the services are holding the dynamic behavior. The behavior of components is analyzed by counting services at run time environment when they demonstrate polymorphic behavior.

RFC is strongly recommended metric to measure Total Supporting Services, because it provides a cumulative measure of Encapsulation & Coupling aspects of object oriented design. As the number of methods increases, design complexity increases. Total supporting services is union of behavior of components/classes and the applied strength for maximum protection. The class behavior may be risky, sensitive, vulnerable, protective, and healthy according to the nature of requirement. The methods communicate with others at different levels to invoke responses of objects which lead greater design complexity. Response set of classes provides combined set of metric value of weighted methods per class and coupling of methods[12]. Decomposition is the process of defining the generalizations and classifications that compose an abstraction[13]. Keeping in mind this assumption, decomposition is merged with higher level of abstraction to maintain the theoretical basis that larger the number of methods invoked from an object, greater will be the design complexity. The motivation of hierarchical decomposition of design is to provide free space and allows the designers to take design decisions independently to distribute complexity across multiple components with less interdependence. This regress review validates the theoretical aspects of the design complexity factors and used metrics for security calculation.

IV. Quantification

This phase actually correlates the complexity factors with object oriented design characteristics. Metrics are derived for each identified complexity factors in order to compute them to get the numeric value. Further, security factors are correlated with the complexity factors. Six models are developed to quantify confidentiality, integrity, availability, authentication and authorization as well as security. The developed models use validated metric set to generate the values for design[12, 13, 18]. At last, software security is quantified in terms of confidentiality, integrity, availability, authentication and authorization.

V. Models for Security Quantification

Six multiple linear regressions are established between identified security attributes and complexity factors. These regression equations help in quantifying Confidentiality, Integrity, Availability, Authentication and Authorization by computing complexity factors using derived object oriented design metrics. Six models, Confidentiality Quantification

Model for Object Oriented Design (CQM^{OODC}), Integrity Quantification Model for Object Oriented Design (IQM^{OODC}), Availability Quantification Model for Object Oriented Design (AQM^{OODC}), Authentication Quantification Model for Object Oriented Design ($AUQM^{OODC}$), Authorization Quantification Model for Object Oriented Design ($AZQM^{OODC}$) and Security Quantification Model for Object Oriented Design (SQM^{OODC}) are developed in the following section. Based upon the relationship of the *security factors (Confidentiality, Integrity, Availability, Authentication and Authorization)* and complexity factors, the relative significance of individual factors that has major impact on security at design phase is weighed proportionally[16-19, 21]. Established relationship between different security attributes and development of security estimation model with feasible object oriented design metrics are depicted from Fig 2 to Fig 6. A multiple linear regression technique has been used to get the coefficients. This technique establishes a relationship between dependent variable and multiple independent variables.

A. Development of Confidentiality Quantification Model (CQM^{OODC})

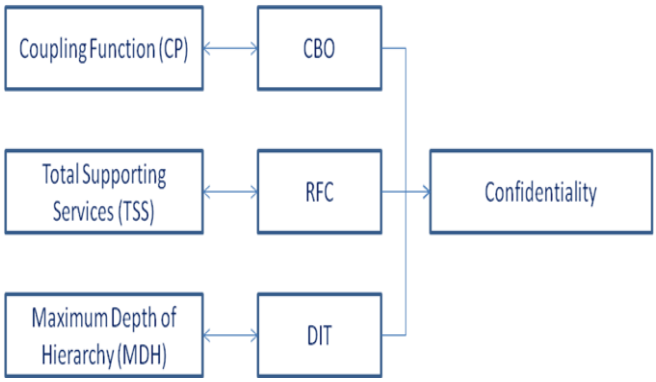

Figure 2. Confidentiality relation diagram

$$\text{Confidenti ality} = (0.599) - (0.623 * CP) + (0.431 * TSS) - (1.25 * MDH) \quad \text{--(1)}$$

B. Development of Integrity Quantification Model (IQM^{OODC})

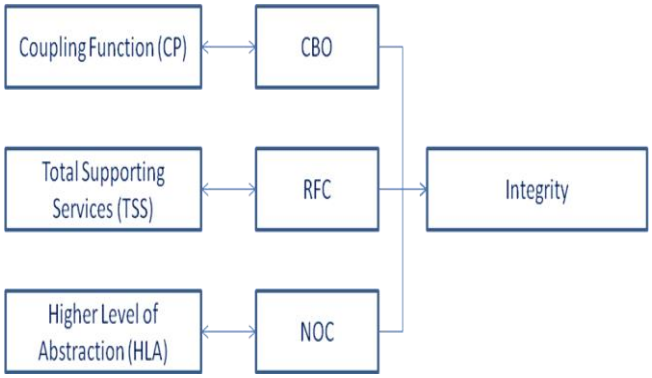

Figure 3. Integrity relation diagram

$$\text{Integrity} = (0.578) + (0.071 * CP) - (0.119 * TSS) + (1.07 * HLA) \quad \text{--(2)}$$

C. Development of Availability Quantification Model (AQM^{OODC})

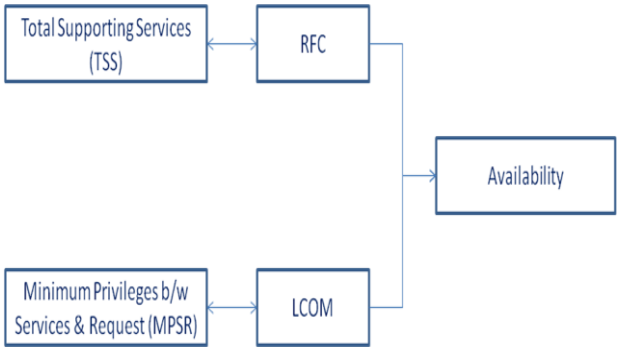

Figure 4. Availability relation diagram

$$\text{Availabili ty} = (0.654) + (0.048 * TSS) - (0.180 * MPSR) \quad \text{---(3)}$$

D. Development of Authentication Quantification Model ($AUQM^{OODC}$)

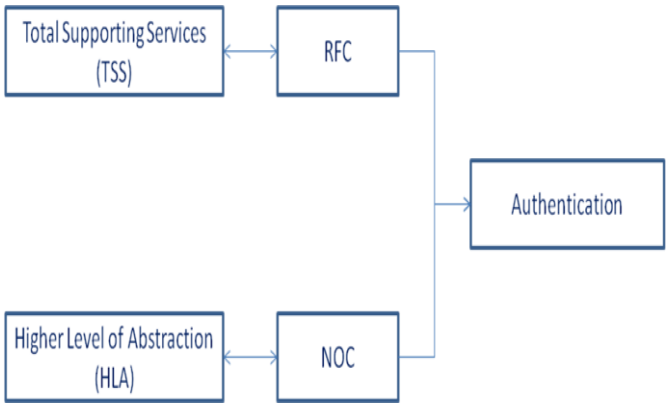

Figure 5. Authentication relation diagram

$$\text{Authentica tion} = (0.011) + (0.119 * TSS) + (0.658 * MPSR) \quad \text{--(4)}$$

E. Development of Authorization Quantification Model ($AZQM^{OODC}$)

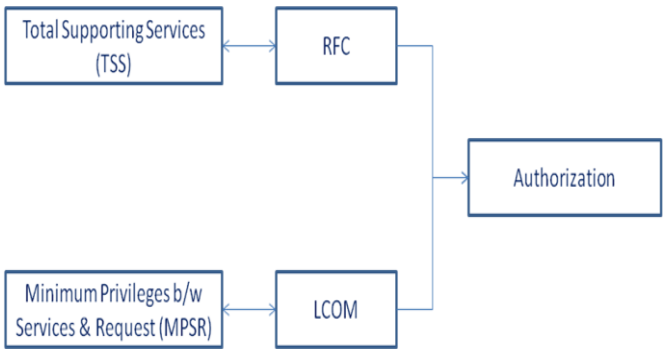

Figure 6. Authorization relation diagram

$$\text{Authorizat ion} = (-0.208) + (0.134 * TSS) + (0.143 * MPSR) \quad \text{--(5)}$$

F. Development of Security Quantification Model (SQM^{OODC})

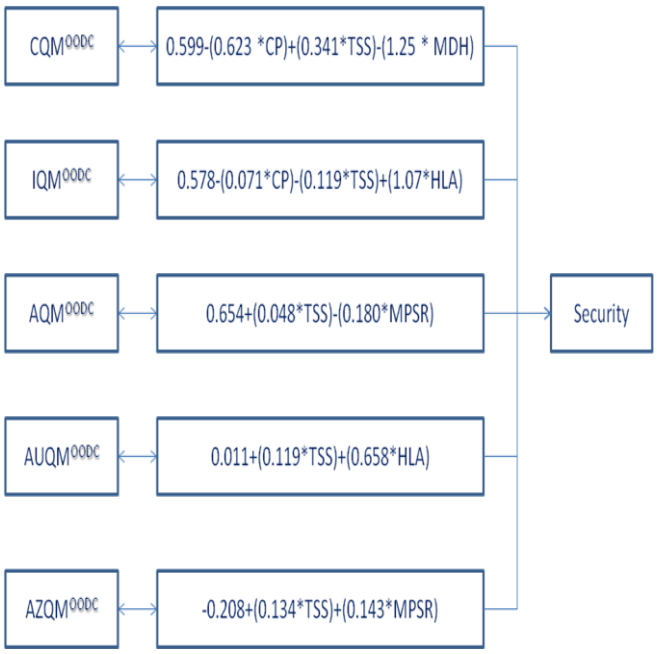

Figure 7. Security estimation model

$$\text{Security} = (-0.104) - (0.094 * \text{CQM}) + (0.431 * \text{IQM}) + (0.733 * \text{AQM}) - (0.067 * \text{AUQM}) + (0.188 * \text{AZQM}) \quad ---(6)$$

VI. Empirical Validation

Validation techniques are the best suitable mechanism to investigate the performance and usefulness of proposed methodology by collecting data from expertise knowledge. The validation process of frameworks or models are organized by collecting statistical data from practical testing that brings a scientific foundation to the engineering of anticipated learning. It accepts that recommended technique is a reasonable demonstration of the actual scheme with adequate reliability to assure analysis objectives. The basic concept for the development of frameworks or models is to analyze the detailed problem with different level of abstraction. There is some possibility that different subsection of models may have dissimilar level of validity to gain the complete boundary of system behavior. The model validation is categorized into three considerable features which are assumption, input parameter values and distributions and output values and conclusions.

No matter how powerful a theoretical result may be, it needs to be empirically validated to establish its practical use, effectiveness and efficiency. This is true in all Engineering disciplines, including Software Engineering. Therefore, in addition to the theoretical validation, an experimental tryout is equally important in order to make the claim acceptable. In view of this fact, an experimental validation of the proposed models, Confidentiality Quantification Model for Object Oriented Design (CQM^{OODC}), Integrity Quantification Model for Object Oriented Design (IQM^{OODC}), Availability Quantification Model for Object Oriented Design (AQM^{OODC}), Authentication Quantification Model for Object Oriented Design ($AUQM^{OODC}$), Authorization Quantification Model for Object Oriented Design ($AZQM^{OODC}$) and for Security Quantification Model for Object Oriented Design (SQM^{OODC}), have been carried out using sample tryouts. Following sections describes the details of validations. The viable experiments are helpful to validate proposed model to establish the effectiveness and efficiencies for its practical usefulness. Therefore, validation of Security Quantification Model for Object Oriented Design has been carried out using sample tryouts. For the purpose, ten versions of class diagram of online purchase system are taken[23]. The results obtained on implementing the same are shown in table 1. Table 2 produces the standard values and calculated values of confidentiality, integrity, availability, authentication and authorization[16-19, 21].

Class Diagram	CP	TSS	MDH	HLA	MPSR
Design 1	3.30	6.44	0.44	0.30	2.20
Design 2	2.60	5.80	0.18	0.36	1.00
Design 3	2.80	4.90	0.18	0.32	1.60
Design 4	1.50	2.40	0.20	0.12	1.60
Design 5	1.60	2.50	0.16	0.16	1.70
Design 6	1.00	3.45	0.54	0.36	1.54
Design 7	2.00	3.80	0.30	0.30	1.20
Design 8	3.10	5.20	0.20	0.10	1.00
Design 9	2.06	4.40	0.20	0.18	1.33
Design 10	1.60	3.40	0.24	026	1.90

Table 1. Metric values for security quantification models

Class Diagram	Conf_Stand	Conf_Cal	Int_Stand	Int_Cal	Aval_Stand	Aval_Cal	Auth_Stand	Auth_ Cal	Az_Stand	Az_Cal
Design 1	0.30	0.189	0.428	0.366	0.583	0.567	0.70	0.974	0.833	0.940
Design 2	0.733	0.732	0.461	0.457	0.692	0.752	0.70	0.940	0.733	0.712
Design 3	0.416	0.325	0.538	0.536	0.636	0.601	0.56	0.804	0.692	0.677
Design 4	0.30	0.232	0.60	0.527	0.50	0.481	0.45	0.380	0.40	0.342
Design 5	0.272	0.254	0.636	0.565	0.545	0.468	0.45	0.413	0.454	0.370

Design 6	0.50	0.477	0.60	0.623	0.545	0.542	0.45	0.658	0.545	0.474
Design 7	0.333	0.273	0.66	0.588	0.692	0.620	0.45	0.660	0.538	0.472
Design 8	0.222	0.190	0.333	0.286	0.777	0.723	0.70	0.631	0.666	0.603
Design 9	0.636	0.566	0.428	0.395	0.733	0.622	0.45	0.419	0.583	0.562
Design 10	0.538	0.462	0.692	0.593	0.545	0.484	0.45	0.527	0.545	0.546

Table 2. Security quantification model summary

A. Hypothesis Testing for (CQM^{OODC}) Model

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model[15]. A hypothesis test based on 2-sample t test is being performed and confidence interval is being observed by the difference of two standard mean. The t test history of confidentiality is given in Table 3. It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model.

Ho: (Null Hypothesis): The impact values derived from CQM^{OODC} cannot significantly reflect the risk posture of the threat element with existing approach.

H1: (Alternate Hypothesis): The impact values derived from CQM^{OODC} can significantly reflect the risk posture of the threat element with existing approach.

t Test for Confidentiality data

	Mean	Std. Div	Std. Error	No. of Samples	Pearson Coff.	Deg. of freedom	t- Values
Conf_Old Values	0.425	0.170	0.053				
				10	0.982	9	4.89
Conf_New Values	0.370	0.181	0.057				

Table 3. t-Test for Confidentiality

To find out the significance of the difference between the means of old Conf_values and Conf_values, the means of both old and new confidentiality impact is calculated. Pearson coefficient of correlation comes out to be 0.982. The coefficient shows that the old Conf_values before treatment and new values of Conf_values after treatment are highly correlated. The degree of freedom for both confidentiality values is 9. This test provides the ground for applicability of t-test. The t value comes out to be 4.89. As the value exceeds the t critical value of 2.26 for a two tailed test at the 0.05 level for 9 degree of freedom, thus the null hypothesis H_{01} is strongly rejected and the alternate hypothesis H_{11} is accepted. The impact values derived from CQM^{OODC} can effectively reflect the risk posture of the threat element with available methodology. The obtained equation to quantify Confidentiality using design parameters is highly acceptable.

B. Hypothesis Testing for (IQM^{OODC}) Model

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model. A hypothesis test based on 2-sample t test is being performed and confidence interval is being observed by the difference of two standard mean. The t test history of integrity is given in Table 4.

t Test for Integrity data

	Mean	Std. Div	Std. Error	No. of Samples	Pearson Coff.	Deg. of freedom	t- Values
Int_Old Values	0.537	0.119	0.037				
				10	0.945	9	3.57
Int_New Values	0.493	0.112	0.035				

Table 4. t-Test for Integrity

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model.

Ho: (Null Hypothesis): The impact values derived from IQM^{OODC} cannot significantly reflect the risk posture of the threat element with existing approach.

H1: (Alternate Hypothesis): The impact values derived from IQM^{OODC} can significantly reflect the risk posture of the threat element with existing approach.

To find out the significance of the difference between the means of old Int_values and new Int_values, the means of both old and new integrity impact is calculated. Pearson coefficient of correlation comes out to be 0.945. The coefficient shows that the old Int_values before treatment and new values of integrity after treatment are highly correlated. The degree of freedom for both integrity values is 9. This test provides the ground for applicability of t-test. The t value comes out to be 3.57. As the value exceeds the t critical value of 2.26 for a two tailed test at the 0.05 level for 9 degree of freedom, thus the null hypothesis H_{01} is strongly rejected and the alternate hypothesis H_{11} is accepted. The impact values derived from IQM^{OODC} can effectively reflect the risk posture of the threat element with available methodology. The obtained equation to quantify Integrity using design parameters is highly acceptable.

C. Hypothesis Testing for (AQM^{OODC}) Model

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model. A hypothesis test based on 2-sample

t test is being performed and confidence interval is being observed by the difference of two standard mean. The t test history of availability is given in Table 5. It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model.

t Test for Availability data							
	Mean	Std. Div	Std. Error	No. of Samples	Pearson Coff.	Deg. of freedom	t- Values
Avl_Old Values	0.624	0.094	0.029				
				10	0.878	9	2.57
Avl_New Values	0.586	0.098	0.031				

Table 5. t-Test for Availability

Ho: (Null Hypothesis): The impact values derived from AQM^{OODC} cannot significantly reflect the risk posture of the threat element with existing approach.

H1: (Alternate Hypothesis): The impact values derived from AQM^{OODC} can significantly reflect the risk posture of the threat element with existing approach.

To find out the significance of the difference between the means of old Aval_values and new Aval_values, the means of both old and new availability impact is calculated. Pearson coefficient of correlation comes out to be 0.878. The coefficient shows that the old Aval_values before treatment and new values of Availability after treatment are highly correlated. The degree of freedom for both availability values is 9. This test provides the ground for applicability of t-test. The t value comes out to be 2.57. As the value exceeds the t critical value of 2.26 for a two tailed test at the 0.05 level for 9 degree of freedom, thus the null hypothesis H_{01} is strongly rejected and the alternate hypothesis H_{11} is accepted. The impact values derived from AQM^{OODC} can effectively reflect the risk posture of the threat element with available methodology. The obtained equation to quantify availability using design parameters is highly acceptable.

D. Hypothesis Testing for ($AUQM^{OODC}$) Model

It is mandatory to check the validity of proposed model for acceptance. A t-test examines whether two samples are different and is commonly used when the variances of two normal distributions are unknown and when an experiment uses a small sample size. A 2-sample t test has been introduced to test the significance of Auth_Stand values to Auth_Cal Values. A hypothesis test based on 2-sample t test is being performed and confidence interval is being observed by the difference of two standard mean. The t test history of Authentication is mentioned in Table 6.

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model.

Ho: (Null Hypothesis): The impact values derived from $AUQM^{OODC}$ cannot significantly reflect the risk posture of the threat element with existing approach.

H1: (Alternate Hypothesis): The impact values derived from $AUQM^{OODC}$ can significantly reflect the risk posture of the threat element with existing approach.

t Test for Authentication data							
	Mean	Std. Div	Std. Error	No. of Samples	Pearson Coff.	Deg. of freedom	t- Values
Au_Old Values	0.536	0.118	0.037				
				10	0.766	9	2.33
Au_New Values	0.640	0.213	0.067				

Table 6. t-Test for Authentication

To find out the significance of the difference between the means of old Au_values and new Au_values, the means of both old and new authentication impact is calculated. Pearson coefficient of correlation comes out to be 0.766. The coefficient shows that the old Au_values before treatment and new values of Au_values after treatment are highly correlated. The degree of freedom for both authentication values is 9. This test provides the ground for applicability of t-test. The t value comes out to be 2.33. As the value exceeds the t critical value of 2.26 for a two tailed test at the 0.05 level for 9 degree of freedom, thus the null hypothesis H_{01} is strongly rejected and the alternate hypothesis H_{11} is accepted. The impact values derived from AQM^{OODC} can effectively reflect the risk posture of the threat element with available methodology. The obtained equation to quantify Authorization using design parameters is highly acceptable.

E. Hypothesis Testing for ($AZQM^{OODC}$) Model

It is mandatory to check the validity of the proposed model for acceptance. A t-test examines whether two samples are different and is commonly used when the variances of two normal distributions are unknown and when an experiment uses a small sample size. A 2-sample t test has been introduced to test the significance of Auth_Stand values to Auth_Cal Values. A hypothesis test based on 2-sample t test is being performed and confidence interval is being observed by the difference of two standard means. The t test history of Authorization is mentioned in Table 7.

t Test for Authorization data							
	Mean	Std. Div	Std. Error	No. of Samples	Pearson Coff.	Deg. of freedom	t- Values
Az_Old Values	0.598	0.131	0.041				
				10	0.977	9	1.65
Az_New Values	0.569	0.176	0.055				

Table 7. t-Test for Authorization

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model.

Ho: (Null Hypothesis): The impact values derived from $AZQM^{OODC}$ cannot significantly reflect the risk posture of the threat element with existing approach.

H1: (Alternate Hypothesis): The impact values derived from $AZQM^{OODC}$ can significantly reflect the risk posture of the threat element with existing approach.

To find out the significance of the difference between the means of old Az_values and new Az_values, the means of both old and new authorization impact is calculated. Pearson coefficient of correlation comes out to be 0.464. The degree of freedom for both authorization values is 9. This test provides the ground for applicability of t-test. The t value comes out to be 1.65. As the value does not exceeds the t critical value of 2.26 for a two tailed test at the 0.05 level for 9 degree of freedom, thus the researcher is fail to reject null hypothesis $H_{01}$and it concludes the impact values derived from AZM^{OODC} cannot effectively reflect the risk posture of the threat element with available methodology. The obtained equation to quantify Authorization using design parameters is significantly produces same results as authorization standard methodology does. There is no significant difference between authorization standard methodology and authorization calculated methodology.

F. SQM^{OODC} Model Validation

In view of the fact, an experimental validation of the proposed model namely Security Quantification Model for Object Oriented Design (SQM^{OODC}) has been carried out using sample tryouts. An established method is being used to get Security_Stand values for all the ten versions of the design diagram[16-19, 21]. Due to unavailability of single security index value, the researcher gets the average values of available different index. Similarly for the same version, the proposed model (SQM^{OODC}) is being used to calculate Security _Cal values through establishing multiple liners regression equation of available security attributes. The data is depicted in table 8.

Class Diagram	Security_Stand	Security_Cal
Design 1	0.437	0.589
Design 2	0.628	0.647
Design 3	0.530	0.608
Design 4	0.466	0.493
Design 5	0.484	0.505
Design 6	0.548	0.562
Design 7	0.561	0.622
Design 8	0.444	0.603
Design 9	0.456	0.645
Design 10	0.376	0.396

Table 8. Security Data Calculation

G. Hypothesis Testing for (SQM^{OODC}) Model

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of Security _Stand values to Security _Cal Values. A hypothesis test based on 2-sample t test is being performed and confidence interval is being observed by the difference of two standard mean. The t test history of security is mentioned in Table 9.

t Test for Security data

	Mean	Std. Div	Std. Error	No. of Samples	Pearson Coff.	Deg. of freedom	t- Values
Sec_Old Values	0.493	0.073	0.023	10	0.693	9	3.46
Sec_New Values	0.567	0.079	0.025				

Table 9. t-Test for Security

It is mandatory to check the validity of proposed model for acceptance. A 2-sample t test has been introduced to test the significance of the model.

Ho: (Null Hypothesis): The impact values derived from SQM^{OODC} cannot significantly reflect the risk posture of the threat element with existing approach.

H1: (Alternate Hypothesis): The impact values derived from SQM^{OODC} can significantly reflect the risk posture of the threat element with existing approach.

To find out the significance of the difference between the means of old Security_values and new Security_values, the means of both old and new Security impact is calculated. Pearson coefficient of correlation comes out to be 0.693. The coefficient shows that the old Security_values before treatment and new values of Security_ after treatment are highly correlated. The degree of freedom for both security values is 9. This test provides the ground for applicability of t-test. The t value comes out to be 3.46. As the value exceeds the t critical value of 2.26 for a two tailed test at the 0.05 level for 9 degree of freedom, thus the null hypothesis H_{01} is strongly rejected and the alternate hypothesis H_{11} is accepted. The impact values derived from SM^{OODC} can effectively reflect the risk posture of the threat element with available methodology. This model provides a single index value of security for object oriented class hierarchies. The obtained equation to quantify Security using design parameters is highly acceptable.

VII. Implication of Study and Future Work

On the successful completion the study, the researcher found that early security estimation is highly desirable for in the area of secure software development. The knowledge gained from the above study may directly or indirectly contribute to prove the significance in the following manner:

- The developed framework may be used to validate other available models, which do not get the appropriate place in the literature due to the lack of their theoretical and empirical validation.
- The developed framework provides step by step procedure to quantify security attributes at early stage of development life cycle.

- The available framework is helpful to identify appropriate metrics for complexity factors at design phase, which brings opportunity to make viable changes for improvement of security level of design.
- The developed models are validated using structural and functional information from object oriented class hierarchies. The model's ability to estimate overall security from design information, at least for the different design hierarchies of projects has been demonstrated. Several design of different projects being used to estimate security and statistical analysis reported that model has been found significantly correlated.
- The proposed model may be used effectively in monitoring security at design phase.
- The attributes may helpful to affect the overall security ranking of the software or application.

The model proposed to quantify security of object oriented design using complexity as key attribute is highly significant and correlated with software design constructs. Though the model has been validated using try out data, but its utility may be analyzed for larger set of data. Security can be evaluated in terms of object oriented design construct. Therefore, the effect of changes on security due to design parameters is the matter of study for generating threshold values to control design complexity. A generic guideline may be produced in the form of developer's manual for designing class hierarchies based on the results of the model. Some suggestive measures may be made to the development team to revisit the design to achieve the set of security index.

VIII. Conclusion

The latest issue of the research is incorporating security in development life cycle especially at design phase using the object oriented technology. The aim of the study is to quantify security at design phase. For the purpose, the framework integrates object oriented parameters and correlate with design complexity and security attributes. The different security models are helpful to generate quantitative values through complexity perspective by using object oriented parameters. It may help to evaluate the security of software and provide the basis for cost & effort estimation and facilitate for planning new activities and ideas for secure development.

References

[1] D. Verdon, G. McGraw, "Risk Analysis in Software Design", *IEEE Security & Privacy*, 1540-7793/04, IEEE, pp:32-37, 2004

[2] D. A. Ashbaugh, "Security Software Development: Assessing and Managing Security Risk", *Taylor & Francis Group*, ISBN-13: 978-1-4200-6380-6, 2009

[3] J. Zadeh, D. DeVolder, "Software Development and Related Security Issues", *IEEE*-1-4244-1029-0, pp:746-748, 2007

[4] G. McGraw, *Software Security: Building Security In*, Addison Wesley Professional, ISBN:978-0-321-35670-3, 2006

[5] J. Steven, "Adopting an Enterprise Software Security Framework", 1540-7993, *IEEE Security & Privacy*, pp:64-67, 2006

[6] G. Booch, *Object-Oriented Analysis and Design with Applications*, Pearson Education, ISBN 0-201-89551-X, 2007

[7] S. Herman, S. Lambert, T. Ostwald, A. Shostack, "Threat Modeling- Uncover Security Design Flaws Using the STRIDE Approach", Technical Report, web address:msdn.microsoft.com/en-us/magazine/cc163519.aspx, November 2006.

[8] G. H. Walton, T. A. Longstaff, R. C. Linger, "Technology Foundations for Computational Evaluation of Software Security Attributes", Technical Report CMU/SEI-2006-TR-021, Esc-Tr-2006-021, December 2006.

[9] C. J. Berg, *High Assurance Design: Architecting Secure and Reliable Enterprise Applications*, ISBN:0-321-37577-7, Addison Wesley Professional, Oct 13, 2005

[10] D. Champeaux, L. Douglas, P. Faure, *Object-Oriented System Development*, Addison Wesley, ISBN 0-201-56355-X, 1993

[11] J. Rumbaugh, M. Blaha, *Object Oriented Modeling and Design*, Pearson Education, ISBN:81-7808-738-3, 2003

[12] Dr. L. Rogenberg, D. Brennan, "Principle Components of Orthogonal Object Oriented Metrics", White Paper Analyzing Results of NASA Object oriented Data, Oct 2001

[13] K. Mustafa, R. A. Khan, "Quality Metric Development Framework", *Journal of Computer Science*, 1(3) ISSN: 1549-3636,: pp:437-444, 2005

[14] M. Dowd, J. McDonald, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*, Addison Wesley Professional, ISBN: 978-0-321-44442-4

[15] C R Kothari, *Research Methodology: Methods and Techniques*, Published by New Age International (P) Ltd, ISBN (13) : 978-81-224-2488-1, 1990

[16] S A Khan, R A Khan, "Security Quantification Model", *International Journal of Software Engineering*, ISSN: 2090-1801, Volume 6, Number 2, 2013, pp: 75-89

[17] P. Mell, K. Scarfone, S. Romanosky, "CVSS A Complete Guide to the Common Vulnerability Scoring System Version 2.", June 2007

[18] L. W. Henry, "Maintenance Metrics for Object Oriented Paradigm", *Proceeding of the First International Software Metrics Symposium*, , pp: 52-60, May 1993

[19] Computing and Information Sciences Class diagram version 1:, available at: http://people.cis.ksu.edu/~reshma/798_ClassDiagram.htm, Last Assessed:25 March 2014

[20] S. Uellenbeck, M. Dürmuth, C. Wolf, T. Holz, H. Görtz, 'Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns', CCS'13, November 2013, ACM 78-1-4503-2477-9/13/11, dx.doi.org/10.1145/2508859.2516700.

[21] T. Sommestad, 'A Framework and Theory for Cyber Security Assessments', Submitted for the Degree of Doctor of Philosophy, Industrial Information and Control Systems KTH, Royal Institute of Technology Stockholm, Sweden, 2012.

[22] J. Sedláčková, 'Security Factors in Effort Estimation of Software Projects', ACM Slovakia, Vol. 3, No. 2, 2011, pp: 12-17

[23] S. Lawrence and R. K. Cunningham," Why Measuring Security is Hard", IEEE Computer and Reliability Societies, 1540-7993/10/2010, pp: 46-54, 2010.
[24] S. L. Pfleezer, "Useful Cyber security Metrics", IT Professional, July/August, 2009.

Suhel Ahmad Khan is pursing PhD in Information Technology from Babasaheb Bhimrao Ambedkar University (A Central University), Vidya Vihar, Raebareli Road, Lucknow. He has been completed his MCA degree from Uttar Pradesh Technical University, Lucknow. Mr. Khan is young, energetic research fellow and has completed a Full Time Major Research Project funded by University Grants Commission, New Delhi. He has more than 5 year of teaching & research experience. He is currently working in the area of Software Security and Security Testing. He has also published & presented papers in refereed journals and conferences. He is a member of IACSIT, UACEE, and Internet Society.

Dr. Raees A. Khan has earned his doctoral degrees from JMI, New Delhi, India and he is currently working as an Associate Professor and Head in the Department of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, India. His area of interest is Software Security, Software Quality and Software Testing. He has published a number of National and International books, research papers, reviews and chapters on software quality and software testing.