International Journal of Security and Its Applications
Vol.8, No.5 (2014), pp.213-226
http://dx.doi.org/10.14257/ijsia.2014.8.5.20

Authorization Estimation Model: An Object Oriented Design Complexity Perspective

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

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

Abstract

Software security is one of the most considerable domains of software development. It provides a strong mechanism to manage and incorporate security features for precious estimation at design phase. The structural and behavioral design properties of classes, objects and their relationships are evaluated to develop metric for authorization. The assessment of security using the model is more appropriate and its validation signifies the valid impact of structural and functional information of object oriented design software. The authorization quantification model is developed using multiple linear regression technique on object oriented design constructs. The applied statistical analysis on this study concludes its statistical significance remarked that calculated data is highly acceptable. A strong theoretical basis has been developed for designing the metrics required for complexity factors as well as security attributes.

Keywords: *Authorization, Authorization cogency, security attributes, complexity factors, OO design metrics*

1. Introduction

According to McGraw, software has many advantages. It provides its services which cover areas from running car controlling, cell phone and performing financial system and services in banks. However developer should always keep in their mind the security provision because this volatile area needs highest concern. Therefore the software developer society must focus on it as much as they could. If they do not do so it will result risky and hazardous. While developing software developer should always pay attention to its security measures. After all, its vulnerability and shortcomings might destroy all our hopes and aspirations. Our internet services are catered to the drivers of the whole world through different software programs. So prevention measures should always be through as our highest concerning areas [1].

Whenever software design experts are going to install the components in system, they should ensure that the buildup is done systematically and orderly without any replication of mistakes. Some points should be kept in mind while developing a secure software control point should not only be parasite less but also numerous and multiple. Integration, consistency and synchronization of independent and individual parts should be developed with minimum error or fault. Every measure must be adapted to generalize the complication and complexities. As the commercial and other business are increasing, so require of software assurance is tremendously felt. As it known so far that software is as fragile as glass. If it doesn't handle carefully, it will mar our hopes during its execution resulting in failure of the

ISSN: 1738-9976 IJSIA

system functionality. The size of software should be so that it may acquire innumerable data and information in spite of its transit size.

Authorization is a practice to verify whether a user on the system is allowed to execute an explicit action inside a trust realm. The used mechanism in concert with authorization as a component of an access control policy. Authentication ascertains who a user is, and authorization agrees on what that user is permitted to do. Software should be designed and evolved in such manner that can be minimize or maximize its impact according to intended formula or proposed models. There have been no scientific studies that validate the claim to quantify authorization of object oriented software at design stage. It is nonetheless a strongly held conclusion among various practitioners that object-oriented software is easier to change than conventional software. Quantitative assessment of class hierarchy for authorization security attributes provides a technique, how to manage/control resources for security improvement. Thus reliability, safety and security are a must for software development. These inevitable components will enhance user faith in manufactured or produced software's. Its resistance capability must also be maintained at all costs to deliver defect free products [2].

2. Literature Review

It is evident from literature survey that considerable efforts are being made towards acceptable metrics for authorization. The following section briefly describes some of the pertinent contributions made by the researchers and practitioners. Most of the work carried out in the area strengthens the need and importance of authorization mechanism. A research paper entitled 'Identification and Implementation of Authorization and Authorization Patterns in the Spring Security Framework', by Aleksander Dikanski, Roland Steinegger, Sebastian Abeck, Research Group Cooperation & Management (C&M), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany contributed his effort in the area of software security. In this paper, the Spring Security framework is analyzed with an objective of identifying supported Authorization and authorization patterns. This research paper uses a real world case study to implement security requirement in a web application. By using this framework an implementation of patterns is identified. This will help to bridge the gap between pattern based security design and implementation to implement high quality security functionality in software systems. Security was examined in its support for common security patterns for Authorization and authorization. Three types of Authorization mechanism are being used and analyzed. RBAC, ABAC and username/password based Authorization is being used with appropriate best-practice to implement templates for Spring Security. These templates can be used as a reference to execute the declared patterns in further projects. This technique is very much fruitful for reuse-based security engineering process [3].

Another contribution 'Role-Based Authorization Constraints Specification Using Object Constraint Language', by Gail-Joon Ahn, Department of Computer Science, University of North Carolina at Charlotte, Michael. E. Shin, Department of Information and Software Engineering, George Mason University presents a brief review arguing that no access right is leaked to an authorized user. Constraints are an important aspect of role-based access control (RBAC) that works as a specification language for secure systems development. An approach is being used in this research paper that works on the object constraint language (OCL). Object Constraints Language (OCL) that is part of the Unified Modeling Language (UML) and has been used in object-oriented analysis and design. The paper demonstrates role-based authorization constraints using an industry standard constraint specification language, OCL. It specified separation of duty constraints, prerequisite constraints and cardinality constraints.

Researchers utilize constraints identified by a formal language such as RCL2000 when we design and analyze role-based systems. This work helps system developer understand the constraints and requirements on secure systems development. A unified way to specify authorization constraints can be investigated so researcher can apply the approach to other access control models such as MAC and DAC [4].

An effort made by Ravi S. Sandhu, Edward J. Coynek, Hal L. Feinsteink and Charles E. Youmank, in his contribution 'Role-Based Access Control Models', introduces a family of reference models for role- based access control (RBAC) in which permissions are associated with roles, and users are made members of appropriate roles. This greatly simplifies management of permissions. Roles are closely related to the concept of user groups in access control. However, a role brings together a set of users on one side and a set of permissions on the other, whereas user groups are typically depended as a set of users only. The basic concepts of RBAC originated with early multi-user computer systems. The resurgence of interest in RBAC has been driven by the need for general-purpose customizable facilities for RBAC and the need to manage the administration of RBAC itself. As a consequence RBAC facility ranges from simple to complex. This article describes a novel framework of reference models to systematically address the diverse components of RBAC, and their interactions [5].

An IEEE Transaction paper entitled 'Constructing Authorization Systems Using Assurance Management Framework', by Hongxin Hu, and Gail-Joon Ahn discusses an approach based on assurance management framework (AMF). Model-driven approach has recently received much attention in developing secure software and systems at an early stage of the software development life cycle. In this paper, we introduce a multilayered software development life cycle (SDLC), which is based on an assurance management framework (AMF), focusing on the development of authorization systems. AMF facilitates comprehensive realization of formal security model, security policy specification and verification, generation of security enforcement codes, and rigorous conformance testing. The research paper eloquent a multilayered SDLC for authorization systems based on proposed AMF, which is designed for analysis and realization of security models and policies. The proposed study discussed multilayered SDLC, toolset, RAE and RASS and constitutes a set of modules including a formal analysis tool such as alloy analyzer to facilitate the features for proposed methodology [6].

A white paper on Design Authorization Systems Using SecureUML, By Rudolph Araujo & Shanit Gupta, from Foundstone Professional Services, a division of McAfee, in February 2005 offers an exclusive combination of services and education to assist organizations constantly and evidently defend the most significant assets from the most critical threats. Through a strategic approach to security, Foundstone identifies, recommends, and implements the right balance of technology, people, and process to manage digital risk and leverage security investments more effectively. Foundstone's Secure Software Security Initiative (S3i™) services help organizations design and engineer secure software. By building in security throughout the Software Development Lifecycle, organizations can significantly reduce their risk of malicious attacks and minimize costly remediation efforts. Services include: Source Code Audits, Software Design and Architecture Reviews, Threat Modeling, Web Application Penetration Testing, Software Security Metrics and Measurement. This white paper describes the Found stone Secure UML template, a Microsoft Visio template built to model authorization systems. The tool allows architects to leverage the power and flexibility of the Visio environment while modeling their role-based access control systems. Secure UML is based on the widely known Unified Modeling Language (http://www.uml.org) [7].

The security expert and researcher Yuhui Zhong and Bharat Bhargava from Center for Education and Research in Information Assurance and Security and Department of Computer Science, Purdue University, USA developed a trust enhanced role mapping server, which can shared RBAC mechanism for authorization based on evidence and trust. The contribution entitled 'Authorization based on Evidence and Trust' presents an algorithm to evaluate reliability of evidence and role assignment policies. This works helps to quantify trust, formalization of evidence and trust, evaluation of reliability of evidences on the basis of machine reasoning and proof. This work is also beneficial in the area of e-commerce [8].

2.1 Concluding Remarks and Problem Formulation

Considerable efforts are being made by the researchers and industry professionals to design and build up secure software. However it may usually experience from delayed security assessments, which affect the security assessment process. An early and accurate estimation of security provides a way to determine what to measure, how to measure and makes the variables more manageable, meaningful to build up repeatable formulas that show the status of security services and how it changes over time. Design complexity participates as an active member for constructing secure software [9-11]. The literature survey reveals the fact that authorization defines which users are allowed access to the system, as well as the privilege of user for which they are eligible. This survey concludes that authorization is the process of determining whether a user on the system is permitted to perform a specific operation within a trusted domain. A viable quantitative model is needed to address design security through complexity. One of the best approaches may be to correlate complexity factors with security attributes authorization in order to quantify authorization in terms of complexity with the help of object oriented design characteristics. Based on the description of the above problem there may be a vast set of research questions that may need to be addressed. Some of the pertinent ones are stated as follows:

- How user can access the resources to perform actions for authorization in case of object oriented methodology.
- Which are the factors that directly influence the authorization security attributes with respect to object oriented design complexity?
- Can authorization of object oriented software be estimated successfully at the design level?
- Can authorization metric usage be really useful and reliable, without assuring theoretical and empirical validity?
- Is it possible to extract quantitative features from the representation of a software design to enable us to predict the authorization of software?
- What properties of software measures are required in order to determine the authorization of a design through design complexity?
- How general are the lessons learned in this study? Can they be applied in situations involving other metrics, or to organizations, which have different operational contexts?

3. Authorization: Security Attribute

Authorization is the function of specifying access rights to resources, which is related to information security and computer security in general and to access control in particular. Resources include individual files' or items' data, computer programs, computer devices and functionality provided by computer applications [12]. The accessing of resources by users through authorization focuses on *roles and permission* of software applications. If applying any level of security, you should define an application class for every search definition. The application class is responsible for fetching a list of runtime values based on the security attributes [13]. There are various prescribed designs are available for access control systems including discretionary access control, mandatory access control, and role-based access control. In addition, numerous technologies are available for centralizing access control into various frameworks, operating systems, and libraries. Because of the complexity of different access control schemes, it's best to initiate by looking at authorization from a broad perspective. Authorization process is just like to show the identity at the time of checking. In multi user computer system, authorization provides the privileges to access process or resources preceded by authorization [14, 15, 26, 27].

3.1 Authorization Cogency of Object Oriented Class Hierarchy

The correspondence and mapping between the identified security attribute, authorization, 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 classes as a basis. The proposed metrics will be used to compute authorization cogency of OO design with the help of complexity using the class diagram. It has been observed that each of the design construct *i.e.,* Encapsulation, inheritance, cohesion, coupling, & polymorphism having an impact on certain complexity factors and security attributes [18, 19]. The significance of this study is to estimate authorization cogency with an optimized set of object oriented class design features with their relationship in depicted in Figure 1.

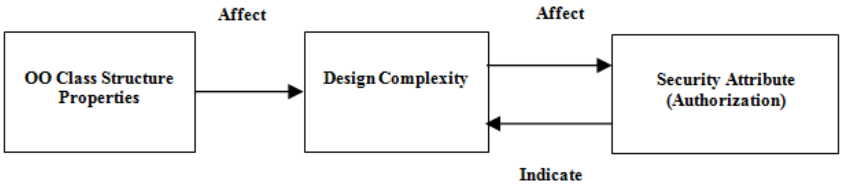
Affect
Affect
OO Class Structure Properties
Design Complexity
Security Attribute (Authorization)
Indicate

Figure 1. Theoretical basis for the Development of Authorization Security Model through OO Design Complexity

Objects access the resources to perform their task via appropriate services. The predefined user acts according to their role with legal permission to access the resources for object oriented design hierarchies. In case of object oriented design methodology the key entity like objects, attributes, methods and classes are arranged to form the design of the software through design characteristics including inheritance, coupling, cohesion, encapsulation, polymorphism, *etc.,* In this respect the object attributes acts as *role and permissions* behave as performing actions for classes. In order to measure the authorization cogency of object oriented design, the main focus should be on the decision taken for accessing protected resources that work on trust and appropriateness. A decomposition of elementary security objectives with corresponding threat tree is depicted in Table 1 [16, 25]. The threat trees are categorized into a tree structure with generic threat class to its root in STRIDE. The evolution regarding authorization strengthen the proposed study that trust and appropriateness can be maintained through providing minimum privilege between services and request through enforcing maximum strength of protection of resources.

Table 1. Threat to Security Objectives

STRIDE threat trees	Elementary security objectives
Spoofing an external entity or process	Authorization
Tampering with data stores	Integrity of stored data
Tampering with data flow	Integrity of transmitted data
Tampering with a process	Integrity of application
Repudiate message	Non-repudiation
Repudiate transaction	Auditability
Information disclosure of data stores	Confidentiality of stored data
Information disclosure of data flow	Confidentiality of transmitted data
Information disclosure of a process	Confidentiality of application
DoS against data stores	Availability of stored data
DoS against data flow	Availability of transmitted data
DoS against a process	Availability of application
Elevation of Privileges for Processes	**Authorization**

Metric experts and practitioners have different opinions on the measurement/estimation of security of software using single and integrated metrics. However, the researcher strongly feels that a single integrated object oriented security metric may lead to a unified measure of security, rather than by doing the qualitative and inevitably subjective interpretations for security through a range of single metric, which generally provide component-wise measurements. It is strongly felt that industry people as well as the practitioners do not want to have different component security measures but are more interested in a unified measure. Security metrics is the measure of security policies, processes and products. Security managers look for a magic formula that calculates risk and effectiveness in reducing risk, but the reality is that security metrics aren't that simple [24].

In general, no pertinent solution is available to correlate authorization with object oriented design complexity. Authorization is confirmation that an authenticated principal, a user, a computer, a network device, or an assembly has permission to perform an operation. Protection allows only appointed users to perform certain actions and it prevents malicious acts. Authorization estimation techniques are very much useful to control data tampering and information disclosure threats. Design application authorization to limit access to application resources or to implement business rules based on the user’s role within the application. For example it checks for approval of expense claim within the application or retrieving sensitive columns from database in .Net environment [17].

$$\textbf{Authorization Cogency} = \frac{\textbf{CMPA}}{\textbf{TC}}$$

Where, AC= Authorization Cogency

TC= Total number of classes where user is authorized to access the resources.

CMPA= Classes with minimal proof of authorization to access resource through vulnerable set.

Calculation of Authorization Cogency (AC) of design requires calculation of identifying set of classes, member or actors which going to access resources to perform a particular task. On the basis of the above criteria, select all relative classes that directly or indirectly associated with above identified classes. This brings the Total number of classes where the user is authorized to access the resources. As per security concern, the extensible critical classes with vulnerable attributes and methods may enhance the possibility of attack and exploitation. A role assigned to valid user for accessing services through applicable permission using aggregation and generalization may increase the rate of exploitation or attack surface with sharing sensitive information among classes. This count will provide the Total number of classes that access the resource to perform task via proper authorization with containing sensitive or vulnerable entry. This adequate information is helpful to estimate authorization cogency of object oriented design in terms of protection allows only authorized user to perform certain action to prevent malicious acts.

4. Metric Implementation

Key values for authorization cogency are the authorized classes that accesses resources to complete task. The inputs are the authorized entry that performs an operation with vulnerable entity. A case study of internet banking system in Figure 2 is taken to estimate authorization cogency of design. In this case study customer class is authorize to avail the services through association to view his cheque status, account statement and balance enquiry. The customer can perform a legal action like to withdraw or deposit methods for transaction with web service class facility. The total number of classes including user which authorized to access the resource are: *{Customer, Chequestatus, account statements, balance enquiry, transaction, web services, accountdetails}.* Classes that hold exploitable information like user id, passwords, transaction id, account operations, pin value etc. are having the possibility of exploiting due to vulnerable attributes or methods. Total numbers of classes that access the resource via proper authorization with vulnerable entry are {*customer, transaction, account details, web services*}.

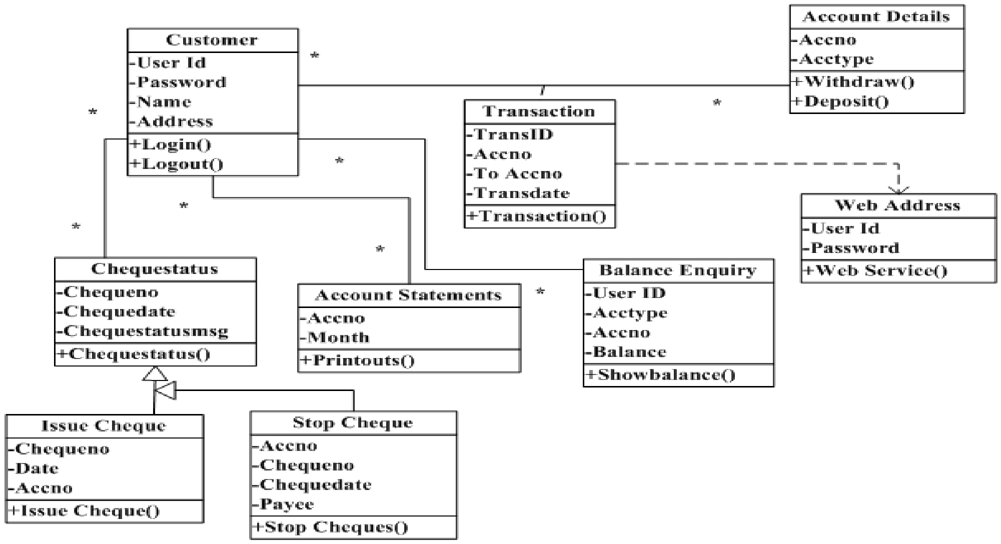

Figure 2. Claas Hierarchy of Internet Banking System

The metric calculation for internet banking design is as follows:

1. Total Count (TC) = 9

Total Count (CMPA) = 4

Then, Authorization Cogency (design online internet banking) = 4/9=0.444

2. Contextual Interpretation

If, CMPA/TC=0 & CMPA/TC=1➔

Using above inequality, the value of AC of a design will always be greater than 0 and less than or equal to1.

$$\boxed{0 \leq AC \leq 1}$$

i.e., Higher the AC, Lower the chance of exploitability of design.

Therefore, it may be interpreted that the authorization cogency of design (internet banking) is moderate and tolerable, if the calculated values of authorization cogency is evaluated on the scale where lower values of AC indicated higher rate of exploitable stage for resources and higher index on AC indicated lower rate of exploitation for resources.

5. Development of Authorization Quantification Model

The process of authorization permits that recognized entity has right to use a particular resource. The object oriented technology classes are defined in terms of their attributes, methods and relationships. The classes interact with each other through generalization, aggregation, coupling or cohesion. The behavior of classes is defined according to role and permission. Services are provided by classes to perform or complete a particular task for a user or administrator. The services are extremely security concern issues because their execution highly depends upon priority and one line action from entry to end point for each authorized system. An example of banking authorization system is taken from [6] that endorses general security rationales such as separation of duty and least privilege to provide support for each object. The participating actors are working according to their role with corresponding permissions on bank objects. The role based assignment to construct authorization system resolve the basic idea to quantify authorization to measure overall services for the user to complete the operations in the design hierarchy is depicted in Figure 3. The overall description for authorization is mentioned in Table 3.

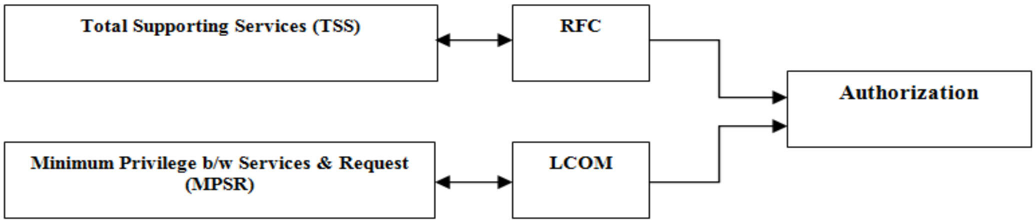
Total Supporting Services (TSS)
RFC
Authorization
Minimum Privilege b/w Services & Request (MPSR)
LCOM

Figure 3. Relation Diagram

Table 3. Description Table for Authorization

Description	Quantitative analysis of software security at early stage enables the evaluation and assessment of security and provides the basis for assessment of security technologies. In order to improve efficiency of the software from the security point of view security estimation is required. Moreover, security expert, Gary McGraw, strongly recommended making an endeavor to quantify security trade-offs [1]. An effort has been made in this regard to develop an approach to address Authorization mechanism through complexity perspective.
Audience	Security Management Groups, Security Operations, Academicians and Researchers working in the area of security quantification
Question	Security goals pertain to such question as to what we are going to secure. What should we do to secure our software? How user can access the resources to perform actions for authorization in case of object oriented methodology. Which are the factors that directly influence the security of software in design complexity

	perspective? Is it possible to extract quantitative features from the representation of a software design to enable us to predict the authorization for software? Can we develop an integrated or minimal set of methods that incorporates all the aspects of the object oriented paradigm through complexity and estimate the authorization security attribute?
Answer	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. Our approach is to develop an Authorization quantification model for OO design phase assuming complexity as a key factor. This will lead to help the Authorization mechanism of a class hierarchy. It provides a comparative analysis with different approaches for Authorization quantification and reduces the effort on security assurance and avoidance of unnecessary overhead. It may help to evaluate the security of software which facilitate the estimation, and planning of new activities.
Formula	$\text{Authorization Cogency} = \frac{\text{CMPA}}{\text{TC}}$ Where, AC= Authorization Cogency TC= Total number of classes where the user is authorized to access the resources CMPA= Classes with minimal proof of authorization to access the resource through vulnerable set.
Targets	Higher values of Authorization would generally result in more vulnerable methods or attributes in the class. This number will also help organizations interpret the results of other application security metrics. Because of the lack of experiential data from the field, no strong consensus on the range of acceptable goal values for security spending exists. In general, this value should be comparable with peer organizations with similar profile and security activities [9].
Source	The primary data sources for this model as inputs are defined by the realistic dataset of OO class hierarchies.
Usage	Optimal environment always refers to lowest degree of incidents. The lower the number of incidents, the healthier the security posture would be assuming perfect detection. This model can also indicate the effectiveness of security controls that works to evaluate Authorization security attribute quantitatively with respect to OO design complexity.
Limitations	A security program may or may not have direct control over the number of incidents that occur within their environment. However, this model could be used to show that improving countermeasures and processes within operations to reduce the number of incidents that occur. Thus, Number of incidents must be considered in the context of other models.

The generic quality models [18, 20-23] have been considered as a basis to develop the authorization Quantification Model (AZQM^{OODC}) for object oriented design shown which involves the following steps. (1) Identification of complexity factors that influences authorization mechanism at design phase. (2) Identification of object oriented design characteristics. (3) A means of linking between them. Based upon the relationship of the authorization security factors and complexity factors, the relative significance of individual factors that has major impact on security at design phase is weighted proportionally. A multiple linear regression technique has been used to get the coefficients. This technique establishes a relationship between a dependent variable and multiple independent variables. The Multiple Regression equation takes the following form:

$$\left[Y = \alpha_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots\ldots\ldots\ \ldots\ldots\ldots\ \beta_n X_n\right] \qquad \textbf{(1)}$$

Where Y is dependent variable, X_1, X_2, X_3………. X_n are independent variables related to Y and are expected to explain the variance in Y, $\beta_1, \beta_2, \beta_3,\ldots\ldots\ldots \beta_n$ are the coefficients of the respective independent variables. Regression coefficient is the average amount of dependent increase/decrease when the independents are held constants, and α_0 is the intercept. Total supporting services (TSS) and Minimum Privilege b/w Services & Request (MPSR) incorporate the involvement of Inheritance and aggregation of classes. Taking into account the same, the multiple regression equation to quantify authorization of object oriented design has been established. A class hierarchy of online shopping has been used to implement the model to quantify the authorization through complexity [28]. The ten class hierarchies are used for metric calculation and extracted data is depicted in Table 4. Standard values of Authorization are (Auth_Stand) are calculated using the proposed formula in section 3.1.

Authorization= α + β1* TSS + β2 * MPSR -----------(2)

Table 4. Authorization Calculation Table

Class Diagram	Auth_Stand	TSS	MPSR
Design 1	0.70	4.88	1.98
Design 2	0.70	3.88	1.88
Design 3	0.68	4.45	1.90
Design 4	0.50	4.41	1.33
Design 5	0.58	4.00	1.42
Design 6	0.55	4.33	1.33
Design 7	0.60	4.45	1.27
Design 8	0.44	3.45	1.81
Design 9	0.44	3.72	1.54
Design 10	0.43	3.60	1.50

[Authorization= (-0.208) + (0.134 * TSS) + (0.143 * MPSR)……………(3)]

The model summary of calculated data is mentioned which discusses the statistical interpretation of used data that model is highly effective. Where, Authorization is the dependent variable, Total Supporting Services and Minimum Privilege b/w Services & Request are independent variables. These variables are expected to explain the variance in Authorization, β_1, β_2, and β_3 are the coefficients of respective variables, and α_0 is the intercept. Table 5 summarizes the results of the correlation analysis for Authorization quantification model, and shows that for all the system, all of the design constructs are highly correlated with Integrity. R is a measure of the correlation between the observed value and the predicted value of the criterion variable. The next column gives us a value of R Square, which is a measure of how much of the variability in the outcome is accounted for by the predictors and indicates the proportion of the variance in the criterion variable which is accounted for by our model. In essence, this is a measure of how good a prediction of the criterion variable we can make by knowing the predictor variables. The Adjusted R Square value tells us that our model accounts for 93% of variance in the Spelling scores a very good model. The Pearson correlation table 6 provides a strong correlation between elements of models for high acceptability. Pearson's coefficient of correlation is the most widely used methods of measuring the degree of relationship between two variables. The value of '*r*' lies between ± 1. Positive values of *r* indicate positive correlation between the two variables (*i.e.,* Changes in both variables take place in the statement direction).

Table 5. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Sig. F Change
1	0.930	0.865	0.842	0.0615	0.000

Table 6. Pearson Correlation

	Authorization	TSS	MPSR
Authorization	1.00	0.90	0.75
TSS	0.90	1.00	0.63
MPSR	0.75	0.63	1.00

6. Model Validation

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 model namely complexity Authorization quantification model (AZQM^{OODC}) has been carried out using sample tryouts. The following sections describe the details of validations and data regarding validation for authorization formulation is carried out from different version of class diagram of online purchase system and extracted data from designs are available in Table 7 [28].

Table 7. Authorization Data Table

Class Diagram	TSS	MPSR	Auth_Stand	Auth_Cal
Design 1	4.41	1.33	0.70	0.573
Design 2	3.10	1.50	0.375	0.421
Design 3	3.60	1.90	0.444	0.546
Design 4	3.45	1.81	0.400	0.513
Design 5	3.60	1.20	0.444	0.446
Design 6	6.44	2.20	0.833	0.940
Design 7	5.80	1.00	0.733	0.712
Design 8	4.90	1.60	0.692	0.677
Design 9	2.40	1.60	0.40	0.342
Design 10	2.50	1.70	0.454	0.370
Design 11	3.45	1.54	0.545	0.474
Design 12	3.80	1.20	0.538	0.472
Design 13	5.20	1.00	0.666	0.603
Design 14	4.33	1.33	0.583	0.562
Design 15	3.60	1.90	0.545	0.546
Design 16	2.81	1.09	0.545	0.328
Design 17	3.50	1.16	0.333	0.431
Design 18	3.30	1.00	0.250	0.381
Design 19	3.27	1.09	0.636	0.390
Design 20	3.63	1.00	0.583	0.425
Design 21	3.36	1.00	0.692	0.389

6.1 Hypothesis Testing for Authorization

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 8.

Table 8. T-Test of Authorization

	Mean	Std. Deviation	Std. Error	No. of Samples	Degree of Freedom	t-Value
Az_Old Values	0.5424	0.1488	0.0324	21	20	1.50
Az_New Value	0.5019	0.1457	0.0318			

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): There is no significant difference between Az_Std and Az_Cal values.

H1: (Alternate Hypothesis): There is significant difference between Az_Std and Az_Cal values.

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. This test provides the ground for applicability of t-test. The t value comes out to be 1.50. As the value does not exceeds the t critical value of 2.086 for a two tailed test at the 0.05 level for 20 degree of freedom, and the calculated t value fail to reject the null hypothesis H_{01} and concludes that there is no significant difference between authorization standard values and authorization calculated values. 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.

7. Conclusion

This paper developed a multivariate linear model ‘Authorization Quantification Model ($AZQM^{OODC}$)’ for object oriented software in design time. It estimates the security in terms of design complexity factors which are weighted according to their influence. A case study has been used to validate the model through appropriate statistical measures and contextual interpretation signifies the acceptance of the model.

References

[1] G. McGraw, “Software Security: Building Security In”, Addison Wesley Professional, ISBN:978-0-321-35670-3, **(2006).**

[2] M. Dowd and J. McDonald, “The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities”, Addison Wesley Professional, ISBN: 978-0-321-44442-4, **(2006).**

[3] A. Dikanski, R. Steinegger and S. Abeck, “Identification and Implementation of Authorization and Authorization Patterns in the Spring Security Framework”, Research Group Cooperation & Management (C&M), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, SECURWARE, The Sixth International Conference on Emerging Security Information, Systems and Technologies, ISBN: 978-1-61208-209-7, **(2012),** pp. 14-20.

[4] G. J. Ahn and M. E. Shin, “Role-Based Authorization Constraints Specification Using Object Constraint Language”, Proceeding of Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE **(2001)**, pp. 157-162, doi>10.1109/ENABL.2001.953406.

[5] R. S. Sandhu, E. J. Coynek, H. L. Feinsteink and C. E. Youmank, “Role-Based Access Control Models”, IEEE Computer, vol. 29, no. 2, **(1996)** February, pp. 38-47.

[6] H. Hu and G.-J. Ahn, “Constructing Authorization Systems Using Assurance Management Framework”, IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, vol. 40, no. 4, **(2010)** July.

[7] R. Araujo and S. Gupta, “Design Authorization Systems Using SecureUML”, Foundstone Professional Services, **(2005)** February.

[8] Y. Zhong and B. Bhargava, “Authorization based on Evidence and Trust”, Proceeding of 4th International Conference, DaWaK 2002 Aix-en-Provence, France, **(2002)** September 4–6, pp:94-103, doi>10.1007/3-540-46145-0_10 .

[9] B. B. Madan, K. Goˇseva-Popstojanova, K. Vaidyanathan and K. S. Trivedi, “Modeling and Quantification of Security Attributes of Software Systems”, Proceedings of the International Conference on Dependable Systems and Networks (DSN’02), IEEE, **(2002)**, pp. 505-514.

[10] G. McGraw, “Form the Ground-Up: The DIMACS Software Security Workshop”, DIMACS Technical Report 2003-13, **(2003).**

[11] D. Frank, “Agencies Seek Security Measures”, CIO Magazine, **(2000)** June 19.

[12] **(2014)** March 31, http://en.wikipedia.org/wiki/Authorization, Last Visited.

[13] http://docs.oracle.com/cd/E38689_01/pt853pbr0/eng/pt/tpst/task_WorkingwithAuthorizationandAuthorization-027f51.html.

[14] G. H. Walton, T. A. Longstaff and R.C. Linder, “Evaluation of Software Security Attribute’s”, IEEE, **(1997)**.

[15] M. Bishop, “Computer Security: Art and Science”, Addison Wesley, ISBN:0-201-44099-7, **(2002).**

[16] A. Yautsiukhin, R. Scandariato, T. Heyman and F. Massacci, “Towards a Quantitative Assessment of Security in Software Architectures”, In Proceedings of the 13th Nordic Workshop on Secure IT Systems, **(2008),** https://lirias.kuleuven.be/bitstream/123456789/.../nordsec08-artsiom.pdf.

[17] **(2014)** February 20, http://msdn.microsoft.com/en-us/library/ee817656.aspx, Last Visited.

[18] S. R. Chidamber and C.F. Kemerer, “A Metric Suite for Object Oriented Design”, IEEE Trans. On Software Engineering, IEEE, vol. 20, no. 6, **(1994)** June, available at: http://www.pitt.edu/~ckemerer/CK%20research%20papers/MetricForOOD_ChidamberKemerer94.pdf.

[19] S. A.Khan and R. A. Khan, “Securing Object Oriented Design: A Complexity Perspective”, International Journal of Computer Application, vol. 8, no. 13, **(2010)** October, pp. 8-12.

[20] C. Wang and W. Wulf, “A Framweork for Security Measurement”, Proc.of National Information Systems Security Conference, **(1997)** October 7-10, pp. 522-533.

[21] R. G Dromey, “A Model for Soft. Product Quality”, IEEE Transaction on Soft. Engg., vol. 21, no. 2, **(1995)** February, pp. 146-162.

[22] Dr. L. Rogenberg and Dinnis Brennan, “Principle Components of Orthogonal Object Oriented Metrics”, White Paper Analyzing Results of NASA Object oriented Data, **(2001)** October.

[23] J. Bansia, “A Hierarchical Model for Object Oriented Design Quality Assessment”, IEEE Transaction of Software Engineering, vol. 28, no. 1, **(2002)** January, pp. 4-17.

[24] P. Lindstrom, “Security Measuring Up”, CISSP, **(2005),** available at: http://searchsecurity.techtarget.com/tip/1289483 sid14_gcil106034900.html.

[25] S. Hernan, S. Lambert, T. Ostwald and A. Shostack, “Threat Modeling: Uncover Security Design Flaws using the STRIDE Approach”, **(2014)** March 11, web reference: http://msdn.microsoft.com/en-us/magazine/cc163519.aspx, Last Visited.

[26] L. Pesante, “Information Security Basic”, Carnegie Mellon University, **(2008)**, web reference: www.uscert.gov/site/default/files/publications/informationsecuritybasic.pdf.

[27] Y. Cherdantseva and J Hilton, "Information Security and Information Assurance", The Discussion about the Meaning Scope and Goals IN: Organizational, Legal and Technological Dimensions of Information System Administrator. IGI Global Publishing, **(2013)**, web reference: http://en.wikipedia.org/wiki/Information_security#cite_note-Cherdantseva_Y_2013-2.

[28] S. A. Khan and R. A. Khan, "Security Quantification Model", International Journal of Software Engineering, IJSE, ISSN:2090-1801, vol. 6, no. 2, **(2013)**, pp. 75-89.

Authors

Suhel Ahmad Khan, he 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.

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 Testi ng. He has published a number of National and International books, research papers, reviews and chapters on software quality and software testing.