

# Improve the Security of UML Class Diagrams Through the Application of Model Refactoring

Haris Mumtaz<sup>1</sup>, Mohammad Alshayeb<sup>2\*</sup>, Mahmood Niazi<sup>2</sup> and Sajjad Mahmood<sup>2</sup>

<sup>1</sup> Electrical, Computer and Software Engineering Department  
University of Auckland, New Zealand  
*hmum126@aucklanduni.ac.nz*

<sup>2</sup> Information and Computer Science Department  
King Fahd University of Petroleum and Minerals  
Dhahran 31261, Saudi Arabia.  
*{alshayeb, mkniazi, smahmood}@kfupm.edu.sa*

**Abstract:** Security is an important software quality attribute that reflects the ability of a system to prevent malicious actions and loss of information. Model bad smells may have negative impact on software quality. There are a few studies that investigated security bad smells and the application of model refactoring for improving the security of software applications. The objective of this research is to investigate how model refactoring can be applied on Unified Modeling Language (UML) class diagrams to remove security bad smells and improve the structural design of an application. We use a genetic algorithm-based detection technique to find security model bad smells in a class diagram. Next, refactoring is applied to remove security-related model bad smells. Finally, the improvement to security is assessed through the statistical analysis of quality metrics. The empirical validations of our approaches are performed using multiple open source projects. The results show significant detection recall and correction efficacy in terms of detecting and removing the security bad smells, respectively. The statistical analyses of quality metrics allow us to conclude there is a significant improvement in security quality of the investigated class diagrams as a result of refactoring.

**Keywords:** Model Refactoring; Secure Software; Model Bad Smells.

---

\* Corresponding author