

Modeling Authentication in Mal Sequence Diagram

Madiha Arooj^{a,1,*}, Muhammad Yousaf^b, Rizwan Bin Faiz^d, Samir Obaid^{1,1,*}

^aFaculty of Computing, Riphah International University, Islamabad, Pakistan

^bFaculty of Computing, Riphah International University, Islamabad, Pakistan

^cFaculty of Computing, Riphah International University, Islamabad, Pakistan

^dDepartment of Software Engineering Capital University of Science and Technology, Islamabad, Pakistan

Abstract

Software Requirement Specification involves both functional and Non Functional Requirements (NFRs), defines system attributes and serves as a constraint on software design. While designing software, NFR in general and security requirements in specific are neglected or poorly designed since they are not treated as an important part of designing. As a consequence of which our software is vulnerable to security threats. Modeling software threats in early stages of software engineering i.e. design can help developers to identify security threats and design their corresponding mitigation. We therefore in this paper model authentication threats e.g. skimming and replay attack in mal sequence diagram. In order to do so an extended meta-model of sequence diagram designed based upon which profile mal sequence diagram is designed. Both skimming and replay attack and their corresponding mitigation is successfully modeled upon ATM various usage scenarios of ATM case study.

Keywords: NFR modeling, security, sequence diagram, UML profiling, object constraint language

1. INTRODUCTION

Non-functional requirements (NFR) is a blanket word form all requirements not only for functional requirements [1]. Modeling of NFR is very complex and not as simple as modeling of functional requirements [2]. NFR needs to concentrate at an early stage of developing software, if not, then they might be proving very costly, time-consuming, and difficult to manage at later [2]. To value software, it should meet functional as well as non-fictional requirements. Supakkul et al. proposed modeling language by extending the UML framework using a UML profile [25]. Modeling software threats can help developers to identify threats regarding non-functional requirements (NFR) at the early design stage and then mitigation measures can be designed to mitigate non-functional requirement (NFR) modeling threats. If software design flaws are not

found at an early stage of design and postponed being solved at the closing stage, the cost may be increased 3 times more than the original cost []. As security is the most important factor that should be considered at an early phase of design [7][8]. Security requirement is not considered separate from another requirement. MARTE has provided concepts for modeling non-functional security requirements [8]. To

* Autor en
correspondencia.

Email addresses: autor@cea-ifac.es (Madiha Arooj), autor2@cea-ifac.es (Muhammad Yousaf),
rizwan.faiz@riphah.edu.pk

(Rizwan Bin Faiz), autor@cea-ifac.es (Samir Obaid)

URL: madiha.arooj18@yahoo.com (Madiha Arooj), muhammad.yousaf@riphah.edu.pk
(Muhammad Yousaf), samir.obaid@cust.edu.pk (Samir Obaid)

¹Nota al pie para el autor 1

2

develop software, UML is considered a de facto standard. Unified Modeling Language (UML) is a standard for developing software [4][3].

As NFR is usually abstract, they are not formally specified. NFR conflict with each other e.g., performance vs. safely [6]. Hussein et al. presented a UML profile i.e. UMLIntr which specializes UML notations for the context of intrusion specification. There is no need to learn attack language to describe the attack. This tool given in the paper [6] helps to avoid conflicts e.g., security vs. usability, ambiguous, and redundant requirements.

Sequence diagrams are the most widely used UML models in the software industry [5]. Sequence diagrams are used for modeling a dynamic aspect of the system, they can also be used for model-based testing. An only sequence diagram can show messages exchanged between the object classes in order in which the messages occur from 13 models used in UML 2.0 [5]. This paper extends the Unified Modeling Language (UML) profile for modeling UML sequence diagram for security NFR. Moreover, we introduced a malicious sequence diagram (MSD) where we draw security threats and their respective mitigation measures by introducing new symbols or icons with the help of UML profiling. The rest of the paper is structured as follows: Section 2 presents a Related work of security modeling. Section 3 presents a UML profiling for behavior diagram. Section 4 introduces the proposed solution. Section 4 discusses the case study to describe an application of Mal sequence diagram. Section 5 shows a comparison with related work. Finally, section 6 concludes the paper.

2. RELATED WORK

Traditionally it is a challenging thing to incorporate the security into software. Software security is a key to engineering software so that the system works properly under malicious attacks [31]. Usually, we first make the system and on top of that we apply security controls, security itself is not integrated into the system. This makes the system insecure. Basically, We implement

functionality into software and added security on top of the system and this approach generated a weak software.

The new approach which is known as a security engineering approach or system engineering approach where we design a system with security. We incorporate the security into system design. Where functional requirements are focus which then passes to design, then development team make a test case on the basis of this design and then pass on to testing and at the end, software is produced. Which will make software product more secure.



Figure 1. Security software life cycle

In [16], the authors extended use case to misuse case to model security requirements. In [19], the authors explained that misuse case can help to elicit requirements for different types of system for functional but also for non-functional requirements e.g. reliability, safety, and security NFRs. The proposed approach discover more detailed use and misuse case threats and mitigation. Authors proposed a new extension for misuse case to elicit security requirements. In [20], the authors also explained how misuse case diagrams could help in the modeling of security risks and security countermeasure to mitigate these risks. In [21], the authors discussed modeling security requirements when it comes to nonfunctional requirements they introduced misuse case in contract to modeling functional requirements through

3

use case. In [22], the authors introduced an extension for the use case to the model and analyze security requirements which are known as an abuse case. In [23], the authors presented a language to specify Misuse cases in the executable form with attacks and their mitigations to model security requirements. In [24], the authors presented a UML extension of use cases for the development of a security-critical system and security modeling known as UMLsec.

In [25], the authors showed mal activity by introducing a new notation for capturing security threats and compares misuse case and mal activity, results showed that mal activity has strengths in many aspects than misuse case. In [27], the authors extended UML 2.0 for modeling security through activity diagrams. In [28], the authors explained that security engineering is the most important concept in system development. In the paper, authors focused on modeling security engineering through Mal activity diagram at the design stage. In [29], the authors presented an extension of 2.0 UML activity diagram for security requirements.

In [18], the authors extended the existing security modeling technique Misuse sequence diagram to support failure and effect analysis to another level known as the failure sequence diagram. In [26], the authors proposed a technique failure sequence diagram for modeling failure and their effects through interaction between system component.

In [30], the Author uses a state machine diagram to represent dynamic behavior of security specification at the design level. As from above-related work, we can conclude that security engineering is a key to design software at the design level. Different behavior diagrams are used for security modeling. Still, there is no evidence for malicious sequence diagram so in the paper below we will present our work which has been demonstrated through a case study of ATM.

3. UML PROFILING FOR BEHAVIOR DIAGRAMS

design stage. In [29], the authors presented an extension of 2.0 UML activity diagram for security requirements.

In [18], the authors extended the existing security modeling technique Misuse sequence diagram to support failure and effect analysis to another level known as the failure sequence diagram. In [26], the authors proposed a technique failure sequence diagram for modeling failure and their effects through interaction between system component.

In [30], the Author uses a state machine diagram to represent dynamic behavior of security specification at the design level. As from above-related work, we can conclude that security engineering is a key to design software at the design level. Different behavior diagrams are used for security modeling. Still, there is no evidence for malicious sequence diagram so in the paper below we will present our work which has been demonstrated through a case study of ATM. III. UML PROFILING FOR BEHAVIOR DIAGRAMS Unified Modeling Language (UML) is considered a de facto standard for modeling software systems [11][14]. UML modeling is done in all phases of software development from gathering requirement to maintenance [11]. With time, complexity and size of software system increase, due to which model of systems changed separately and several approaches, have been proposed to come-over these conflicts. One of the solutions proposed by authors is the UML profile for modeling these different models by using graphical syntax in UML editor [12].

Authors introduced a framework which describes UML profile e.g UMLIntr for describing intrusion scenario. They specified the use of UML for specifying security in intrusion scenarios. They generated attack signatures using their framework for users than to use a separate framework. The limitation of their work is not to specify disturbed attacks using their UML profile and need to change their profile [3]. UML is a standard for specifying and documenting systems. UML is introduced for extension mechanisms for everyone to customize own syntax and semantics. UML profile is adding new elements or constraints in existing elements; UML itself provides extension mechanics [13]. In model-driven engineering, modeling languages for security have been introduced. Authors presented MARTE, which is a modeling language for modeling security in embedded systems [15]. CASE tool [17] is an integrated modeling language by extending UML with NFR framework using a UML profile. Where authors define a metamodel to represent the concepts in the NFR framework and identify the extension points for the integration of two notations. In [10], the authors defines a UML 2.0 profile for security assessment. UML 2.0 profile provides a mapping of classes in the metamodel for UML modeling elements by

defining stereotype and introduced special icons for representation the stereotypes in the UML diagram. Authors proposed a profile for modeling in a generic and extensible way. They presented a UML profile for modeling the non-functional requirements they added an additional element to UML profile to fulfill the requirements of relevant stakeholder [9].

4

Ref#	NFR (Security)	Threat Description			Mitigation Description			Behavior Model/Diagram				Validation		Tool
		Replay Attack	Skimming Attack	Others	Natural Language	OCL	Use Case	Sequence	Activity	State Machine	Case Study	Example		
16	Security general	in			Security Threats	✓		✓					ATM	
18	Security general	in						✓				✓		
19	Security general	in			Security Threats	✓		✓					Car Security	
20	Integrity, Confidentiality, Availability				Tamper Attack, Encrypt communication, Prevent password attack, wiretap Attack	Identify and authenticate, validate data, lockdata		✓					Accounting software package	

Ref#	NFR (Security)	Threat Description			Mitigation Description			Behavior Model/Diagram				Validation		Tool
		Replay Attack	Skimming Attack	Others	Natural Language	OCL	Use Case	Sequence	Activity	State Machine	Case Study	Example		
21	Authentication Privacy, Integrity Access control				Spoofing	✓								
22	Security general	in			Script Kiddies	One-time control		✓					Laboratory	
23	Authentication Suitability, Executability				Active Attack/ Close-in, Passive Attack	Encryption		✓					Voting System and Train Control System	UCSIM
24	Integrity, Authentication					Cryptographic		✓					Internet-based business application	JMI interfaces generated by the MDR Library

5

Ref#	NFR (Security)	Threat Description			Mitigation Description			Behavior Model/Diagram				Validation		Example
		Replay Attack	Skimming Attack	Others	Natural Language	OCL	Use Case	Sequence	Activity	State Machine	Case Study	Example		
26	Security general	in						✓					Air traffic management	
27	Privacy Integrity Access Control					✓			✓	✓			Health-care institution	
28	Confidentiality					✓			✓				Online banking system	
29	Access Control Integrity Privacy					✓			✓				Distribution of electricity	
30	Authentication				Dictionary attack	✓	✓			✓			Authentication mechanism	

5

4. PROPOSED MAL SEQUENCE DIAGRAM

In this section, we will introduce a malicious sequence diagram for the first time. We model security at the design level in UML behavior diagram. The given below tree presents the concept of UML diagrams into subcategories.

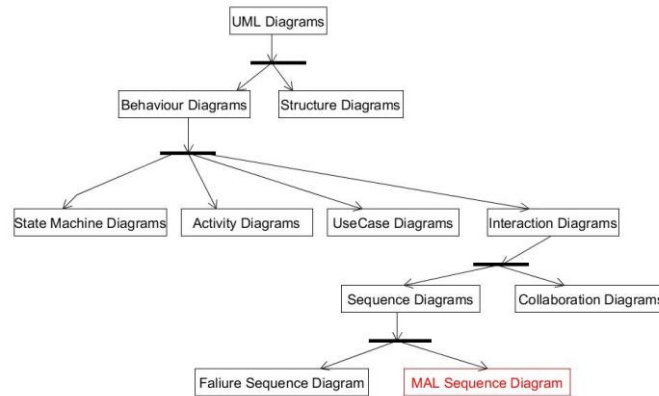


Figure 2. Tree Diagram for MAL sequence In exiting Behavior Diagrams

UML has been divided into two different categories; structural UML diagram and behavior UML diagrams. UML behavior diagrams are used to document the dynamic behavior of the system. They also visualize, specify, and construct dynamic behavior of System. Further behavior diagram is categorized as follows: Use case diagram, StateChart diagram, Activity diagram, and Interaction diagram. Interaction diagram is used for modeling: 1) The control flow by time order by the sequence diagram 2) The control flow of organization using collaboration When applying threats on sequence diagram, we will model a new diagram known as Mal sequence diagram. In this paper we have introduced a Mal sequence diagram where we use tool eclipse for profiling of new UML symbols and introducing them in sequence diagram. In this paper, we draw Mal sequence diagrams for ATM system against specific non-function requirement (NFR) that is security in term of authentication. We will draw a normal flow of ATM transactions and then introduces a threat and their respective mitigation by introducing a new symbol in diagrams.

4.1. Profile Diagram

Following figure define UML profile for sequence diagram which extend UML sequence diagram for security related requirements.

Profile Diagram A Profile is a restricted form of meta model that can be used to extend UML. UML profiling is defined under following labels: Name: name of a label in meta model. Description: a summary of the role played by the label in meta model. Diagram: list of links to diagrams in which the label appears. Semantics: purpose of

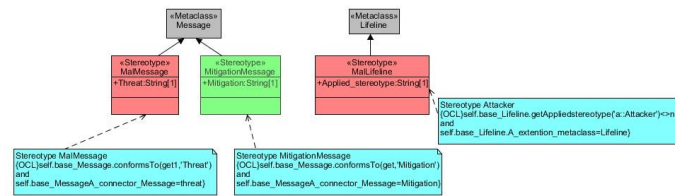


Figure 3. Profile Diagram

the label that is being used in a model Notation: symbol of a label. Generalization: a list of links from which that label is generalized. Association: label that is associated to a particular label each specified by its name, type, and multiplicity, any additional properties Operation: methods that are associated with the label Constraints: restrictions for that label

4.2. UML 2.5.1 Extension guideline template for UML Profiling

Mal-Lifeline:

1. Description

A Mal lifeline represents an Attack object in the Interaction.

2. Diagram

Profile diagram is used to extend current UML meta-class. Notation

A Mal Lifeline is shown using a symbol that consists of a red rectangle forming its head followed by a vertical line (which may be dashed) that represents the lifetime of the participant.

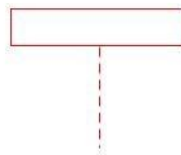


Figure 4. Mal Lifeline

Semantic

When attacker or some malicious object try to attempt malicious activity in sequence.

3. Generalization

Lifeline (from BasicInteractions, Fragments)

4. Associationlifeline

5. Operations

Mal-lifeline is a part of profile diagram and is directly generalize from Lifeline Meta-class.

7

6. Constraints

When two or more than two interaction operate on one lifeline than every lifeline on which operation is performing must appear in interaction. Attack must have a name which will identify the attack in an interaction. If reference is multivalued then selector for lifeline must be specified. (self.selector-&isNotEmpty() implies not self.represents.isMultivalued()) or (not self.selector-&isNotEmpty() implies self.represents.isMultivalued()) Threat Message

1. Description

Threat Message defines a particular threat between Lifelines of an Interaction.

2. Diagram Profile diagram is used to extend current UML meta-class. Notation A Threat Message is a NamedElement that defines specific kind of threat name in an Interaction and shown by red arrow.



Figure 5. Threat Message

Semantic Whenever attacker or malicious object try to attempt some malicious activity in sequence the threat name is mentioned on this message.

3. Generalization Message (from BasicInteractions)

4. Association Sending OccurrenceSpecification Receiving OccurrenceSpecification

5. Operations Threat message is a part of profile diagram and is directly generalize from message Meta-class.

6. Constraints Threat Message has sending occurrence specification and receiving occurrence specificationself.source= sending occurrence specification self.target= receiving occurrence specification Mitigation message:

1. Description

Mitigate Message defines a particular mitigation between Lifelines of an Interaction.

2. Diagram

Profile diagram is used to extend current UML meta-class. Notation

A Mitigation Message is a NamedElement that defines specific kind of threat in an Interaction and shown by green dotted arrow. Semantic



Figure 6. Mitigation Message

An event that occurs to prevent a potential attack or cure the effect of an attack.

3. Generalization

Message (from BasicInteractions)

4. Association

Sending OccurrenceSpecification Receiving OccurrenceSpecification

5. Operations

Mitigation is a part of profile diagram and is directly generalize from message Meta-class.

6. Constraints

Mitigation Message has sending occurrence specification and receiving occurrence specification self.source=

sending occurrence specification self.target= receiving occurrence specification

4.3. Tool Support

All UML models are defined with the help of a special tool. Since models are created in tools our work additionally needed a UML compatible tool which provide the facility for creating a UML profile and writing OCL for the requirement. And additionally, provide a facility of applying it on our model. It enabled to create a model and their extension in the tool. Eclipse is one of the tool that support to model the extended diagrams of UML but for this purpose Eclipse needs to upgrade with Papyrus model and OCL [32]. Eclipse is development tool mainly design for developing an application in java. But it also supports to develop an application in other languages. It also provides a facility to add libraries for OCL and Papyrus [32]. The Classic core OCL component provides the following capabilities to support OCL integration: It defines APIs for evaluating and analyzing OCL constraints in UML models. It also defines UML operations of the OCL abstract syntax model, including support for calibration of analyzed OCL expressions. It also provides a visitor API for examining/transforming the AST model of OCL expressions. It also provides an extensibility API for clients which allow us to modify the analyzing and evaluation environments used by the parser [33] [32]. It is an element of the model development tool which provides an integrated and user-friendly environment for creating, editing models particularly UML, SysML, and MARTE. It is the base platform for several industrial modeling tools [32].

4.4. ATM Case Study

ATM (Automated Teller Machine) allows users to perform basic transactions [34] such as transfer cash, deposit cash, withdraw cash, authentication, registration, and PIN reset. For a demonstration of our suggested work, we have used the ATM SRS document [34]. For withdrawal cash, if entered amount greater than balance, then it will display a message, enter a smaller amount.

Security Requirements in ATM

Authenticating a User

1. Based on the account number and PIN [34].
2. Bank's account information database stores an account number, a PIN and a balance [34].

4.5. Class Diagram for ATM

A class diagram is chosen by software engineers to map the structure of the system because they clearly show classes with their attributes and operations and the relationship between objects. In our figure above, each class is labeled with its name and attributes and operations are mentioned under the respective labeled class.

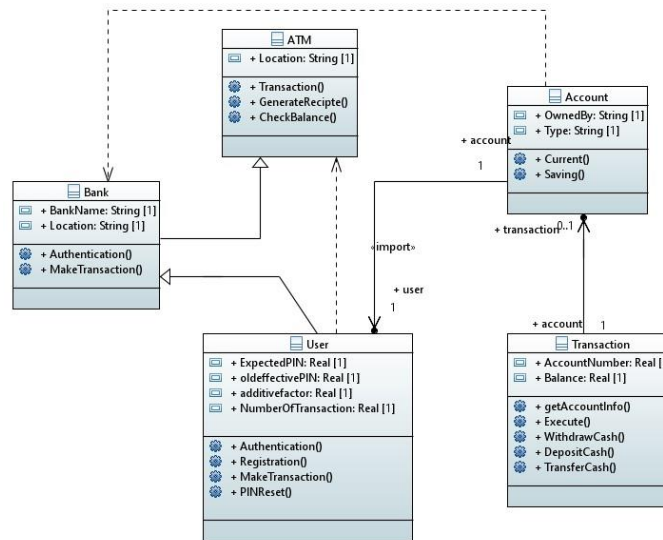


Figure 7. Class Diagram of ATM Machine

4.6. Why UML Class Diagram?

In order to model security the primary step was to design the class diagram for the system. Here the question arises that when we are designing security in mal sequence diagram than why it is necessary to design a class diagram for it. Simply because we are extending sequence diagram through stereotypes and OCL. Stereotypes do not require any special requirement but OCL required attributes/properties. Sequence diagram does not provide any facility to define these attributes. There is only one way to add these attributes that are in of class diagram. Once these attributes are defined in the class diagram then it is easy to access it in the sequence diagram. These attributes can be accessed by creating the sequence diagram from the essential class and then create context between the designed model and the essential class whose attributes are required to be used. Security Specific Authentication

PIN and attempts should be according to requirements. i. Threat Steal PIN. ii. Mitigation

$P2 = P1 + A$, which means effective PIN == expected PIN and attempt_i ≤ 3

OCL

Self.P2 = P1 + A and self.attempt_i = 3

iii. Threat Modeling of Authentication in terms of security

4.7. Description

This diagram model Authentication in terms of Security in Mal sequence diagram. The figure above shows the Flow of user login, possible attacks on login and their associated mitigation. This extended notation consists of two parts. One part consists of standard UML notation like user, lifelines, messages etc and the other part consists of extended notation like an attacker, threats, notes, and mitigation. This model specifies two types of attacks on authentication that is Replay attack and Skimming attack. Then these attacks are mitigated in OCL by restricting login attempt to three-time False Attempt. This means that when a person makes three false attempts the login feature for the user will be locked. OCL is a language that requires an attribute to operate. We cannot write OCL directly in the sequence diagram. OCL required attributes which cannot be defined in the sequence diagram. There is only one way to define these attributes that are in the class diagram. These attributes can be accessed by creating the sequence diagram from the essential class and then create context between the model and the required class. And then by using these attributes, we restrict our login attempt to three false attempts. That is the mitigation of replay attack and skimming attacks.

```

if[P2 = P1 + A] =
true thenlogin =
true elselogin =
false endif
    
```

This OCL indicates that if the new PIN that is P2 is correct and the user has not attempted more than three false Login attempt. Then the login will be successful otherwise the login feature will be blocked for the current user. After writing OCL the next step is to validate our OCL for this purpose we use OCL validation.

5. Case Study For Describing Application Of MAL Sequence Diagram

The use case are modeled using UML. Use cases are defined as the interaction between external actor and the system to fulfill goals. In the below use case diagram of ATM, the system provides bank customer or user and bank with access to core function like transfer cash, deposit cash, withdraw cash, authentication, registration, and PIN reset.

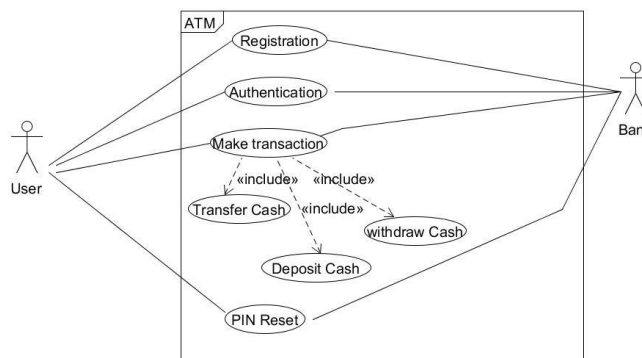


Figure 8. Usecases of ATM Machine

For a better understanding of the Mal sequence diagram for ATM skimming attacks, we have made sequence diagrams for every scenario of ATM. Recall that the proposed solution would explain the scenarios of the normal flow of cash transaction e.g cash withdraw, PIN reset, user's registration, user's authentication.

In this case study user need to register before using the ATM card. Then the user has to complete the authentication process before any transaction or PIN reset process or balance inquiry. Users would have to register with registration server with the help of helpline and customer support staff with full security so that PIN will be secured from any unauthorized person or from an attacker. The second step to register with registration server through ATM with the help of consortium. For authentication, the user has to insert a card in ATM and enter a PIN.

For introducing threats in ATM we use to draw two scenarios: one is to places a skimming device at ATM and second is for attacker which is using an ATM as a normal user. For mitigation, we introduced a solution where we add an additive factor with every PIN. With every transaction, the new effective PIN is changed.

5.1. Security Specific Modeling(Authentication)

As the Attacker in Skimming and Replay attack can access to the system by captures users information including their PIN & network traffic when an unauthorized user sends a communication to the original destination so for the mitigation of this attack we have restricted PIN by Running PIN Number. Where Attacker or unauthorized user enters an old effective PIN which was changed with every transaction and after 3 incorrect PIN attempts, the card is held by ATM.

1. Requirement:

User enter correct PIN.

2. Threat:

Skimming Attack and Replay Attack.

3. Mitigation through OCLself.expectedPIN=old effective PIN + additive factor (password is valid) self.expectedPIN!=old effective PIN + additive factor (password is invalid)

If an attacker gets to know about the old effective PIN in addition to additive factor, the attacker will not even use. ATM account because every time user make a transaction, the PIN will change, and a new PIN will be depending on a number of transactions. As

Self.expectedPIN = oldeffectivePIN + numberoftransaction* additivefactor (password is valid)

5.2. Sequence Diagram

The sequence diagram is a type of interaction diagram. These diagrams are used for understanding requirements for the new system by software developers. A sequence diagram is also known as an event diagram. Sequence diagrams are drawn in order to show details of UML use case, show how objects interact with each other and understand the detailed functionality of existing or future scenario. Below we will draw sequence diagrams for flows of ATM.

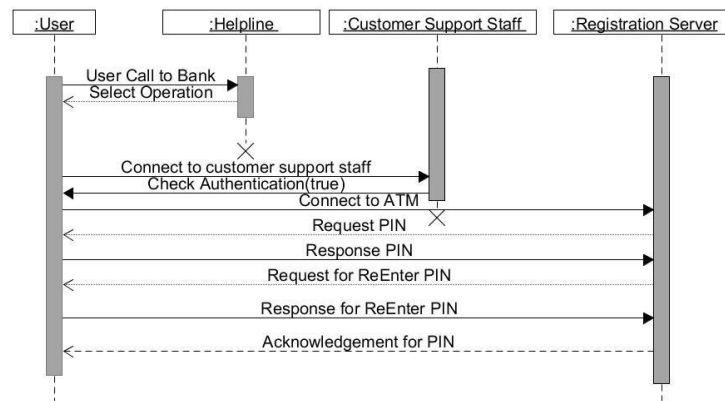


Figure 9. Sequence diagram Of Registration

When the user takes ATM card, he has to set a PIN for secure login. The PIN should be confidential so that no invalid user can access your account. For this, user contact helpline to set a PIN. Bank helpline connects a user to customer support staff where they check user’s authentication if they find a user’s validly they connect a user to ATM to set PIN where the user entered PIN twice to confirm it.

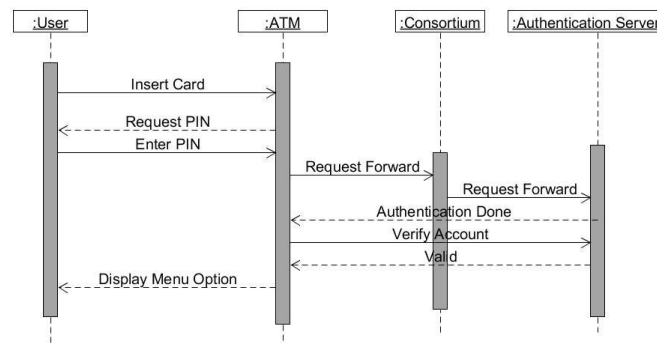


Figure 10. Sequence diagram Of Authentication

The authentication starts after a customer has entered his card in the ATM. The ATM has to check if the card is valid or not. For validation, ATM should read the PIN and personal information from card then ATM send a request to the bank to verify the user. If the user’s verification is valid, ATM will display a menu screen and the user will proceed to other activities.

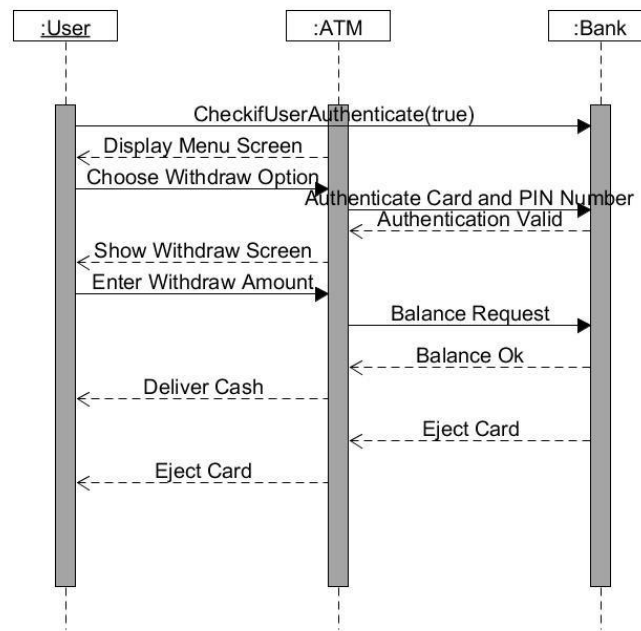


Figure 11. Cash Withdrawal

When authentication completed successfully, the user select withdraw option and entered the amount to withdraw. After this process, ATM checks balance from the user’s account. If entered amount is less or equal to available balance then ATM will deliver amount and ejected the Card. Otherwise, it will display an error message like, ”you have an insufficient amount in your account.”

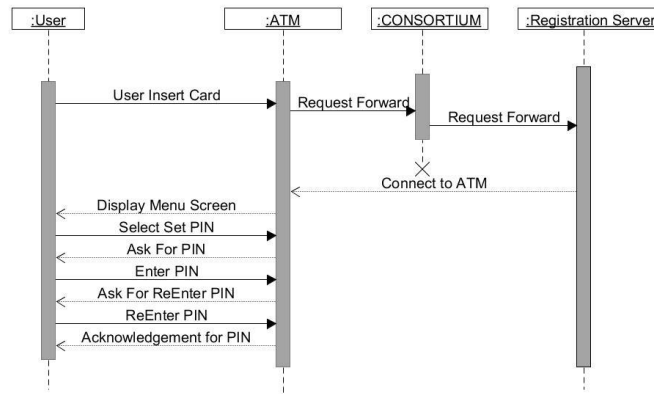


Figure 12. Sequence diagram Of PIN reset

In normal flow, the user can also change ATM PIN when the user forgets the PIN or want the more secure PIN. User insert card in ATM then consortium connects ATM with respective bank of the user. When request accepted then the bank will connect ATM to the bank. ATM shows

menu screen to the user to select PIN reset option. The user enters pin twice to confirm PIN and new PIN activated.

5.3. Mal Sequence Diagram

For system security, the malicious sequence diagrams are used to represent malicious attacks against the system in order to mitigate harmful accidents. The below Misuse case represents the attack on system.

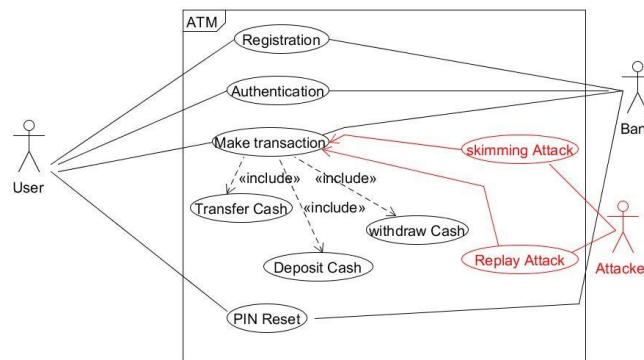


Figure 13. Mis UseCases for ATM

Misuse case is consist of two new entities in use case: one is misuse case and other is attacker. Misuse case shows a number of actions which is performed with the intention to harm the system. On another side, the attacker is the people who intentionally break into the system to harm the system.

In the above fig , we mention malicious activity in the transaction use case where attacker attacks to fetch PIN and personal information of the user by installing a skimming device and then try to attempt replay attack, which is also known as a playback attack, where attacker receive personal information between user and ATM and use it as an authentic user and connect to ATM to use it, with the help of this information.

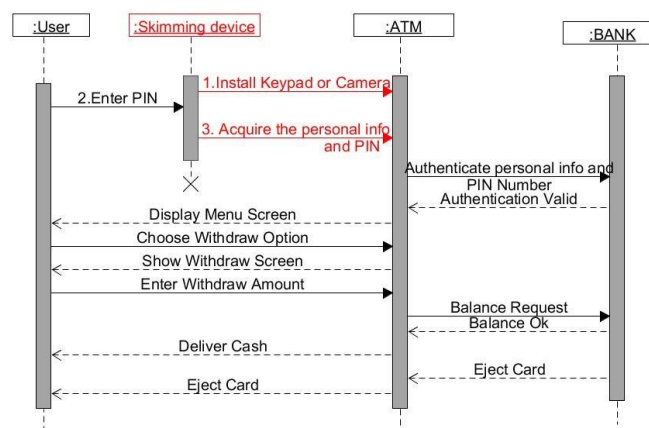


Figure 14. Mal Sequence diagram for attaching Skimming Device at ATM

In this first scenario, when an attacker wants to fetch personal data and PIN, he will go to the ATM and install a card reader and fake keypad or camera which is shown by Mal Lifeline named as Skimming device then a user inserts his card into the ATM machine and enters a PIN, all information is fetched by an attacker.

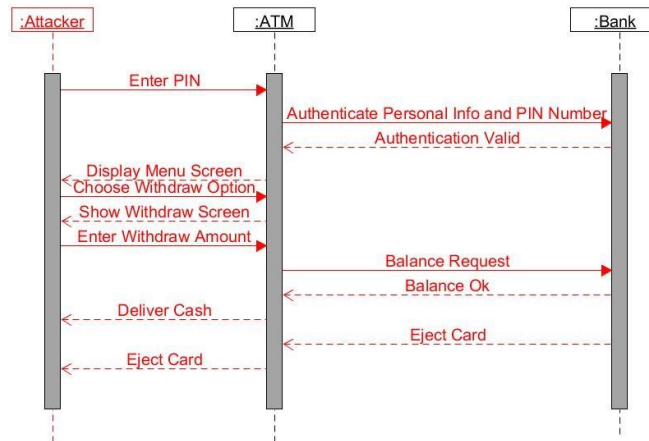


Figure 15. MAL Sequence Diagram for ATM

In the second scenario, after getting desired information, an attacker goes to the ATM which is shown by Mal Lifeline and enters the PIN and performs an activity for example in above Mal sequence diagram the attacker enters the PIN and after bank verification, ATM displays the menu screen. Attacker selects a withdrawal option and enters the amount. These messages are shown by red arrow because they are Mal message.

5.4. Mitigation with Running PIN Number

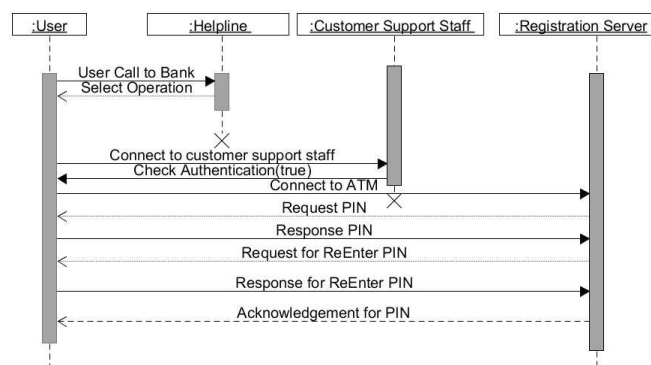
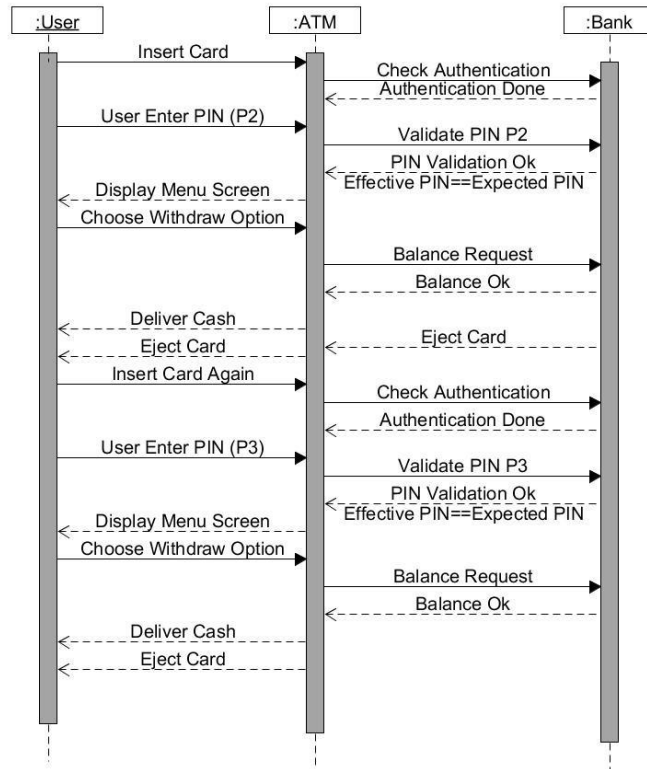


Figure 16. Modified Registration for User via Helpline with Additive Factor

In this case, a user registered from connecting bank helpline via phone, where the connection is established between the user and customer support staff, where the procedure of authentication is carried out. After authentication is successfully done, the user connects to the registration server.

Now, the user will set a PIN and incremental digit twice just like PIN and get acknowledgment message.

Figure 17. mitigation for Cash withdrawal



In the case of mitigation, When the user wants to withdraw money from ATM, user enter PIN which is set at the first time with an additive factor,

$$P2 = P1 + A \tag{1}$$

PIN changed with the number of transactions so, after the transaction at P2, a new effective PIN will be P3 which is

$$P3 = P2 + A \tag{2}$$

Every time the user makes transaction PIN will be changed accordingly.

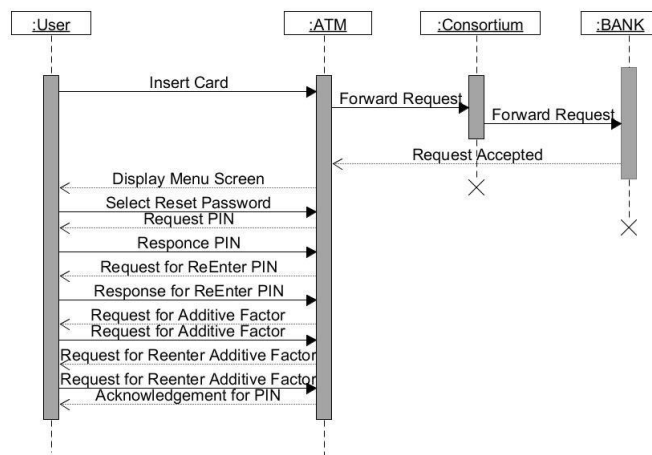


Figure 18. Reset with Additive factor

If the user forgot incremental digit or additive factor, the user will invoke PIN reset option and set a new additive factor that can be added to the user’s PIN.

5.5. Analysis Of Possible Mitigation Technique

In order to mitigate the attacks, we present mitigation against ATM cash withdrawal with the help of running PIN solution. The given below figure shows the mitigation.

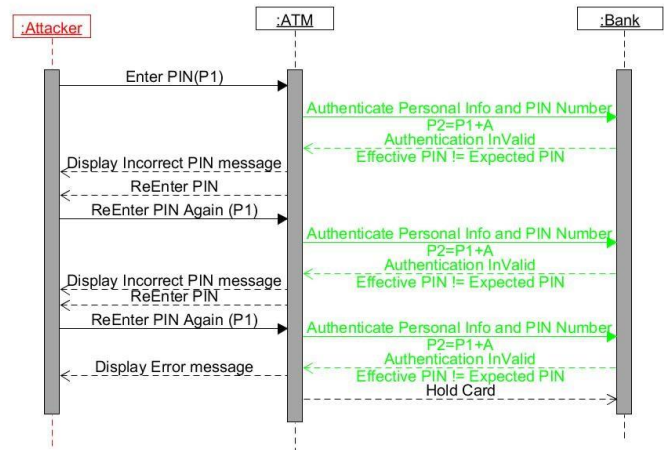


Figure 19. Attach attempt with captured PIN

18

After fetching PIN number through the fake keypad, attackers use ATM to enter into user's account when the user enters a PIN to ATM, it will not work because attacker enters previous PIN P1 which is not stored in the server. The new effective PIN is P2, which is $P2 = P1 + A$, if the entered PIN is not matched with new effective PIN = old effective PIN + additional factor

$$P2 = P1 + A \tag{3}$$

Hence, to use user's ATM account, entered effective PIN must be matched to expected PIN.

$$effectivePIN == expectedPIN \tag{4}$$

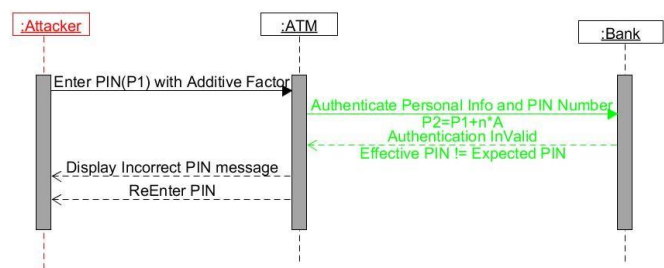


Figure 20. Cash withdrawal with known additive PIN

If an attacker gets to know about the old effective PIN in addition to additional factor, the attacker will not even use ATM account because every time user make a transaction, the PIN will change, and a new PIN will be depending on a number of transactions

As

$$P2 = P1 + n * A \tag{5}$$

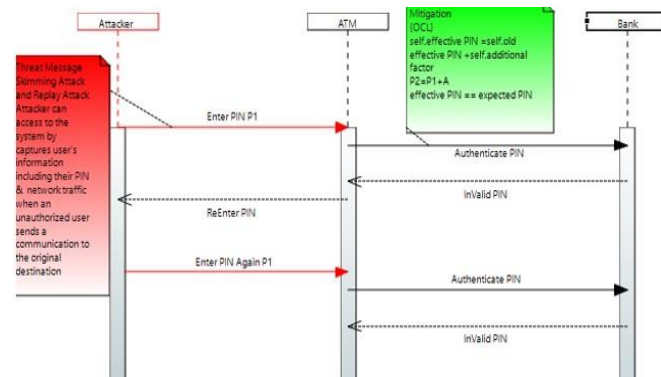


Figure 21. Cash withdrawal with known additive PIN

6. Comparative Analysis

For usability users who don't want complex systems they will set additive factor as zero with their PIN to reduce complexity. Users who are security sensitive about their accounts they will set their additive factor as a huge complex number. So both type of users can use ATM with security.

1

9 Conclusion

This work suggests extending the UML sequence diagram through a UML profile to make the extension reusable. Further, this extended sequence diagram called mal sequence diagram is used to design security at the beginning of the software development process. This will realize the developers that security is an integral part of the system. And this is required to be met during the development stage. Further mostly security is designed at the top level but ISO 25010 suggest to design security at sublevel they can be measured through metrics. For extended UML sequence diagram, we use eclipse for adding new symbols in existing UML for modeling threats or attacks on sequence diagram and introduce mitigation regarding every threat. An ATM case study is demonstrated where we draw a normal flow of ATM sequences and then draw mal sequence diagrams with threats and at the last, we draw mitigation for every threat. Although this research models authentication and authorization threats however this can further be used generate test cases through OCL to test security to gain confidence.

References

- [1] Samuel P, Joseph AT. *Test sequence generation from UML sequence diagrams*. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing 2008 Aug 6 (pp. 879-887). IEEE.
- [2] Iqbal, Muzaffar. *Nfr modeling approaches* In 2011 First ACIS International Symposium on Software and Network Engineering, pp. 109-114. IEEE, 2011.
- [3] Alexander, Ian. *Misuse cases help to elicit non-functional requirements*. Computing and Control Engineering Journal. 2003 Feb 1;volume 14 number 1, pp. 40-5.

- [4] Ravi S, Raghunathan A, Kocher P, Hattangady S. *Security in embedded systems: Design challenges*. ACM Transactions on Embedded Computing Systems (TECS). 2004 Aug;volume 3 number 3, pp. 461-491
- [5] Saadatmand M, Cicchetti A, Sjdin M. *On the need for extending marte with security concepts*. In International Workshop on Model Based Engineering for Embedded Systems Design (M-BED 2011) 2011 Mar.
- [6] Sharbaf M, Zamani B. *A UML profile for modeling the conflicts in model merging* In 2017 IEEE 4th International Conference on KnowledgeBased Engineering and Innovation (KBEI) 2017 Dec 22 (pp. 0197-0202). IEEE.
- [7] Chaudron MR, Heijstek W, Nugroho A. *How effective is UML modeling?* Software & Systems Modeling. 2012 Oct 1;volume 11 number 4;pp. 571-80.
- [8] Gogolla M, Bttner F, Richters M. *USE: A UML-based specification environment for validating UML and OCL*. Science of Computer Programming. 2007 Dec 1;volume 69 number 1-3;pp. 27-34
- [9] Hussein M, Zulkernine M. *UMLintr: a UML profile for specifying intrusions* In 13th Annual IEEE International Symposium and Workshop on Engineering of Computer-Based Systems (ECBS'06) 2006 Mar 27 (pp. 8-pp). IEEE.
- [10] Samuel P, Joseph AT. *Test sequence generation from UML sequence diagrams* Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing 2008 Aug 6 (pp. 879-887). IEEE.
- [11] Gogolla M, Bttner F, Richters M. *USE: A UML-based specification environment for validating UML and OCL*. Science of Computer Programming. 2007 Dec 1;vol 69 number 1-3: pp. 27-34.
- [12] Sindre G, Opdahl AL. *Eliciting security requirements with misuse cases*. Requirements engineering. 2005 Jan 1;vol 10 number 1;pp. 34-44.
- [13] Alexander I. *Misuse cases help to elicit non-functional requirements*. Computing and Control Engineering Journal. 2003 Feb 1;vol 14 number 1;pp. 40-5.
- [14] Sindre, Guttorm and Opdahl, Andreas L. *Eliciting security requirements with misuse cases*. Requirements engineering; Vol 10,number 1;pp. 34-44,2005,Springer.
- [15] Raspotnig, Christian and Opdahl, Andreas L *Improving security and safety modelling with failure sequence diagrams*. International Journal of Secure Software Engineering (IJSSE).2012;vol 3,number 1,pp. 2036,IGI Global.
- [16] Alexander I. *Misuse cases help to elicit non-functional requirements* Computing and Control Engineering Journal. 2003 Feb 1;vol 14 number 1;pp. 40-5.
- [17] Okubo, T., Taguchi, K., & Yoshioka, N *Misuse cases+assets+ security goals* (2009, August). In Computational Science and Engineering, 2009.CSE09. International Conference on (Vol. 3, pp. 424-429). IEEE
- [18] McDermott J, Fox C. *Using abuse case models for security requirements analysis* InProceedings 15th Annual Computer Security Applications Conference (ACSAC'99) 1999 Dec 6 (pp. 55-64). IEEE.
- [19] Whittle J, Wijesekera D, Hartong M. *Executable misuse cases for modeling security concerns*. In Proceedings of the 30th international conference on Software engineering 2008 May 15 (pp. 121-130). ACM.

- [20] Popp G, Jurjens J, Wimmel G, Breu R. *Security-critical system development with extended use cases*. Security-critical system development with extended use cases. In Tenth Asia-Pacific Software Engineering Conference, 2003. 2003 Dec 12 (pp. 478-487). IEEE.
- [21] Firesmith DG. *Security use cases*. Security use cases. Journal of object technology. 2003 May;vol 2 number 3.
- [22] Rodriguez A, Fernandez-Medina E, Piattini M. *Towards a UML 2.0 extension for the modeling of security requirements in business processes*. In International Conference on Trust, Privacy and Security in Digital Business 2006 Sep 4 (pp. 51-61). Springer, Berlin, Heidelberg.
- [23] Chowdhury MJ, Matulevicius R, Sindre G, Karpati P. *Aligning mal-activity diagrams and security risk management for security requirements definitions*. In International Working Conference on Requirements Engineering: Foundation for Software Quality 2012 Mar 19 (pp. 132-139). Springer, Berlin, Heidelberg.
- [24] Rodriguez A, Fernandez-Medina E, Trujillo J, Piattini M. *Secure business process model specification through a UML 2.0 activity diagram profile*. Decision Support Systems. 2011 Jun 1;vol 51 number 3:446-65.

20

- [25] Khan MU. *Representing security specifications in UML state machine diagrams*. Representing security specifications in UML state machine diagrams. Procedia Computer Science. 2015 Jan 1;vol 56:pp. 453-8.
- [26] McDermott J, Fox C. *Using abuse case models for security requirements analysis*. In Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99) 1999 Dec 6 (pp. 55-64). IEEE.
- [27] Whittle J, Wijesekera D, Hartong M. *Executable misuse cases for modeling security concerns*. In Proceedings of the 30th international conference on Software engineering 2008 May 15 (pp. 121-130). ACM
- [28] Nikiforova O, Kozacenko L, Ahilcenoka D. *UML Sequence Diagram: Transformation from the Two-Hemisphere Model and Layout*. Applied Computer Systems. 2013 Jun 1;vol 14 number 1:pp. 31-41.
- [29] Minhas NM, Qazi AM, Shahzadi S, Ghafoor S. *An Integration of UML Sequence Diagram with Formal Specification Methods A Formal Solution Based on Z*. Journal of Software Engineering and Applications. 2015 Aug 13;vol 8 number 08:pp. 372.
- [30] Popp G, Jurjens J, Wimmel G, Breu R. *Security-critical system development with extended use cases*. In Tenth Asia-Pacific Software Engineering Conference, 2003. 2003 Dec 12 (pp. 478-487). IEEE.
- [31] Chowdhury MJ, Matulevicius R, Sindre G, Karpati P. *Aligning mal-activity diagrams and security risk management for security requirements definitions*. In International Working Conference on Requirements Engineering: Foundation for Software Quality 2012 Mar 19 (pp. 132-139). Springer, Berlin, Heidelberg.
- [32] Rodriguez A, Fernandez-Medina E, Trujillo J, Piattini M. *Secure business process model specification through a UML 2.0 activity diagram profile*. Decision Support Systems. 2011 Jun 1;vol 51 number 3:pp. 446-65.

- [33] Rodriguez A, Fernandez-Medina E, Trujillo J, Piattini M. *Secure business process model specification through a UML 2.0 activity diagram profile*. Decision Support Systems. 2011 Jun 1;vol 51 number 3:pp. 446-65.
- [34] Raspotnig C, Opdahl AL. *Improving security and safety modelling with failure sequence diagrams*. International Journal of Secure Software Engineering (IJSSE). 2012 Jan 1;vol 3 number 1:pp.20-36.