Model Driven Security in a Mobile Banking Application Context

Şerafettin Şentürk
R&D Center
Kuveyt Türk
Kocaeli, Turkey
serafettin.senturk@kuveytturk.com.tr

Hasan Yaşar
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, U.S.A
hyasar@cmu.edu

İbrahim Soğukpınar
Computer Engineering
Gebze Technical University
Kocaeli, Turkey
ispinar@gtu.edu.tr

ABSTRACT

As there are growing number of mobile devices worldwide, the applications running on the mobile hand-helds have great impact on the human life. One of the biggest factors for the usage of mobile applications is security and privacy since there are lots of personal and sensitive information for the individuals which are stored in these mobile devices. Because the mobile devices interact with many other devices and run on different kinds of communication protocols, the complexity and integration of mobile applications with the other digital entities increases much more ever than before. That is the reason the security and privacy issues for the mobile clients should be considered in very early steps of their application development phase which is exactly the analysis and design steps. In this study some of the security and privacy by design methodologies and toolsets have been explored. In the phase of UML modelling and workflow definition parts of the application development life cycle, some appropriate techniques have been used. From early stages of designing to test case generation and test execution steps have been covered, so that end to end secure mobile application development life cycle has been realized.

KEYWORDS

security by design, authentication, authorization, secure UML, UMLSec, Graphwalker

1 Introduction

A mobile network can be described as a collection of smart devices interacting in collaboration. Mobile technology and application development is a challenging concept containing a wide ecosystem of interconnected services and devices, such as sensors, consumer products and everyday smart home objects, cars, and industrial and health components. Mobile device deployments come along with different processing and communication architectures, technology standards, design methodologies in various environment at scale. There are collection of mobile connected devices on the network. The mobile network together with RFID technology and growing availability of Wifi, 4G-LTE wireless internet access generates enormous amounts of data. Therefore, storing, monitoring, presenting those data in a seamless and secure way is needed.

Because of limited computing power of such mobile devices, standard security countermeasures and privacy enforcements cannot be applied to mobile application environments. From the security perspective, authentication and authorization mechanisms should be developed by design. Authentication using secret keys stored in non volatile memories involves so many vulnerabilities, because of active attacks, and passive attacks. Protection mechanisms against these attacks are expensive and not adapted to devices with constrained size and energy. Security mechanisms in terms of authentication and authorization for the mobile technologies should provide features like scalability and interoperability. On the other hand, privacy is an important concern in most mobile system use cases, especially when the managed data is sensitive.

Because of all above mentioned reasons, some lightweight and interoperable security solutions should be developed for mobile application development life cycle. And those security and privacy concerns should be embedded into the system early in the design phase of the system. Therefore security and privacy by design concepts play important roles for modelling the most important security properties for the mobile applications like authentication, authorization, confidentiality and so on.

2 Model Based Security Approaches

Model building is a standard way in software engineering. Constructing models in requirements analysis and design phase

ARES '19, August 26–29, 2019, Canterbury, United Kingdom

ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3340529

of the system development process is essential in order to get highly qualified and non-error prone systems. The systems where models are in the center can be defined in an hierarchical way. Firstly, the most general one is called Model Based Engineering in which models are not mandatory for development processes but can be used in supportive activities like documentation and likewise. The second is the one where models drive the development process in every step, which is called Model Driven Engineering. To name it shortly, MDE(Model Driven Engineering) can be seen as a subset of MBE(Model Based Engineering). MDE is really mode driven in each task of software engineering process. From this perspective, MDD(Model Driven Development) is the subset of MDE, in which models are used only in the development tasks. In MDD, code generation can be done from the development design models directly. On the other hand MDS(Model Driven Security) includes the models which are much more security oriented[1].

Model Driven Security is a particular model driven engineering topic for design and development of the secure systems. In Model Driven Development or Architecture, standard functional properties of a software system is modelled mainly by using UML models. Whereas in Model Driven Security one non-functional property which is security is concerned as a design issue. Some of the security concerns that are addressed in the design phase are authorization, authentication, confidentiality, integrity and availability. Security concerns are handled either separately from business logic or together with the system and business models. The modelling approach where security issues are taken into account separately from business models is called Aspect Oriented Modelling(AOM). The concept in which security concerns are handled separately is named as Separation of Concerns.

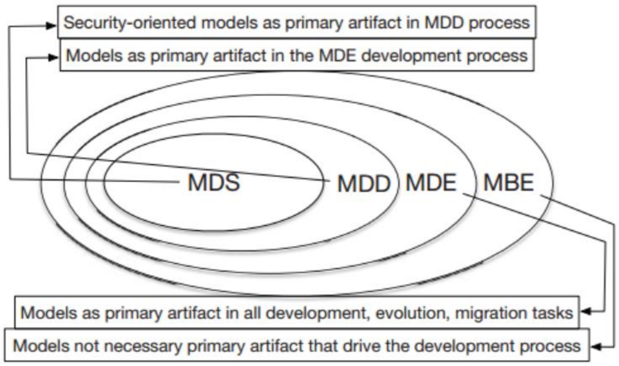

Security-oriented models as primary artifact in MDD process
Models as primary artifact in the MDE development process
MDS
MDD
MDE
MBE
Models as primary artifact in all development, evolution, migration tasks
Models not necessary primary artifact that drive the development process

Figure 1: Relations of MBE, MDE, MDD and MDS

Whereas in non-AOM approach, security concerns are modelled with business models in each and every modelling step. In both AOM and non-AOM methods, UML diagrams, even UML profiles which is an extension mechanism for UML and some kind of DSLs are used. As the relation between different model driven engineering and development types can be seen in Figure 1[1], in model driven architecture and specifically in model driven security model to model transformations are very often in place. Especially a matter of fact that the system should be independent from platform related details, there are first of all platform independent models(PIM) in model driven security. After drawing the PIMs, some kind of models which will include the platform specific details should be developed or transformed from the PIMS. The models that are derived from platforms independent models are called as Platform Specific Models(PSM). Therefore capability of transforming PIMs to PSMs is very important in MDS. For model to model transformations, some engines are used like ATL, QVT, and Graph based MMTs. After the concerns of MMT, model to text or code generation is also a step which should be taken into consideration. Generating the whole system containing the security system infrastructure or only building the security configuration or infrastructure is two main pillars in MTT.

For defining the security concerns in UML models, model driven security approaches use UML extension mechanism which is known as UML profile. In a UML profile approach, stereotypes and additional tags are used to declare the security issues, goals and requirements. UML profiles and other types of DSLs(Domain Specific Language) are developed to better capture the security semantics of the system. Apart from UML profiles, separate DSL definitions is also possible to define the security concerns.

In model driven security there are various types of solutions. Especially mobile application adaptation of the model driven security is also existing in some ways and its security related extensions. In this paper, the five main model driven security approaches have been examined because of their well known properties and popularities.

2.1 Sectet

SECTET emerged for securing web services by the help of Object Constraint Language(OCL) for making use of RBAC(Role Based Access Control). In SECTET, all security infrastructure for instance XACML policy file is generated. other version of SECTET with the name SECTET-PL has been developed for model driven security of B2B workflows. In this framework, constraint based RBAC has been defined and transformed into web service artefacts.

Generally speaking, SECTET focuses on RBAC as its security concern and generating security infrastructure, not all of the source code. There are also recent studies on flexible security configurations for SOA platforms[1].

2.2 Secure Data warehouses

This approach is the MDS method for specifically securing the databases and data warehouses. For secure database development UML and OCL has been used in an extended way. As Model driven approaches imply, PIM and PSM concepts have been extensively used, like security concerns have been embedded into platform independent models. Such security enriched PIMs is transformed to PSMs by using QVT rules. Those PSMs then can be used to generate security specific configuration files. Development of secure XML data warehouses has also been studied[1].

2.3 SecureMDD

SecureMDD is mainly built for secure design and development of smart card applications. In SecureMDD, UML class diagrams are used for static aspects of the system, on the other hand activity and sequence diagrams are used to model the dynamic behaviors of the smart cart applications. From the platform independent model of the system, abstract state machine specification and Java card code are generated. SecureMDD can also be used large and complex smart card applications[1].

2.4 SecureUML

Mainly UML models and UML profiles are used for authorization and access control mechanisms. Role Based Access Control is applied for designing the security issues of the system. Based on this Basin et al. [2] propose a UML -based language (UML profiles) with different dialects, which forms modelling languages (such as SecureUML + ComponentUML) for designing secure systems.

Semantics of SecureUML (and ComponentUML) are provided by Brucker et al. [3] and Basin et al. [4,5] which enable formal analysis of security-design models. Based on this work, Clavel et al. show and discuss their practical experience of applying SecureUML in industrial settings[6]. Recently, the work on SecureUML has been continued by combining SecureUML + ComponentUML with a language for graphical user interfaces (GUI), namely ActionGUI[7,8]. These modelling languages with MMT enable the full generation of security-aware GUIs from models for data-centric applications with access control policies.

2.5 UMLSec

UMLSEC is one of the well known and the most mature model driven security approach. By means of UMLSEC, security requirements, threat scenarios, and various security mechanisms can be developed in design time of a system. Again here security related stereotypes and tags are extensively used. Not like SecreUML only focusing on the authorization, UMLSEC handles multiple security concerns for instance integrity and confidentiality. UMLSEC has also been deployed in some industrial context for designing distributed information systems. The approach has been used together with Secure Tropos in order to tackle security issues from requirement phase of the system.

A unique feature of UMLsec is that it is designed for more Object-Oriented (OO) and component based systems[9]. The approach adds security in general by adding two important elements. The first is method calls. With methods calls there is an added general mechanism for controlling access to data. This also adds the possibility of validating every request made to a certain subset of the system. The second element is information hiding. This mechanism is achieved by the encapsulation of data in objects. The result of this is that data within an object, can only be accessed within objects, and messages are the only way to communicate[10].

The central idea in UMLsec is using stereotypes to define new types of modeling elements that extend the semantics of existing types in the UML metamodel. To give some examples for those stereotypes; security assumptions, security requirements, security policies like fair exchange, role based access control, secure communication link, authenticity, non-repudiation are used.

3 Model Driven Secure Mobile Application

In many respects, developing mobile applications is similar to software engineering for other embedded applications. Common issues include integration with device hardware, as well as traditional issues of security, performance, reliability, and storage limitations. However, mobile applications present some additional requirements that are less commonly found with traditional software applications.

Mobile applications have potential interaction with other applications; embedded devices only have factory-installed software, but mobile devices might have various applications from varied sources, with the possibility of interactions between them. Most embedded devices use software installed directly on the device, but mobile devices usually include applications that invoke services over the telephone network or the Internet via a web browser and affect data and displays on the device[11].

From the perspective of power consumption, many aspects of an application affect its use of the device’s power and the battery life of the device. Dedicated devices can be optimized for maximum battery life, but mobile applications may inadvertently make extensive use of battery-draining resources. Also in terms of testing, while native applications can be tested in a traditional manner, mobile web applications are particularly challenging to test. Not only do they have many of the same issues found in testing web applications, but they have the added issues associated with transmission through gateways and the mobile network.

In terms of security issues, most embedded devices are “closed”, in the sense that there is no straightforward way to attack the embedded software and affect its operation, but mobile platforms are open, allowing the installation of new “malware” applications that can affect the overall operation of the device, including the surreptitious transmission of local data by such an application. Also mobile devices generally connect to the Internet, as well to PCs for software updates or media synchronization, providing convenient attack vectors. Device makers and wirelessservice providers have long focused on communications and other services, with security remaining an afterthought. In addition, said Gustavo de Los Reyes, executive director for AT&T Security R&D, “These phones are being used frequently for sensitive transactions like banking, mobile payments, and transmitting confidential business data, making them attractive targets if not protected.” [12]

The main problem of mobile security is the high interaction between humans, machines and mobile technologies with constraints in terms of connectivity, computational power, and energy. Mobile network is a dynamically changing environment and security issues require making-decision systems to change security mechanisms at runtime[13]. Therefore, it is necessary to learn and adapt for adjusting on-demand security attributes and anticipate new threats in an information system. For this reason, modelling threats and security risks are very important in the early steps of the system development lifecycle of mobile applications. Security is nothing that can only be concerned at the testing phase that comes after the development. It should be taken into consideration when creating the requirements of the systems which are particularlt called security requirements and then they should be designed in a well-defined way. From this perspective, Model-Driven Engineering (MDE) has relevant aspects that contribute to design and deploy mobile application systems considering contextual information and to adopt suitable secure solutions at runtime. Also building models in requirements and design phase of the system development process is really necessary in order to get highly qualified and non-error prone systems. Therefore some suitable approaches and solutions are pretty necessary for embedding security requirements in the designing phase of the mobile application systems. To realize these issues, and to integrate security issues in the very early steps in a mobile banking application, UMLSec and Grapwalker approaches have been used and the results have been collected.

4 Case Study with UMLSec

In this part, secure mobile banking scenarios have been drawn with UMLSec. The reason why UMLSec has been selected is that UMLSec is the most mature secure UML modelling technique among the other approaches. And also UMLSec can handle more than one security issues in one system while others can focus only one like authorization. The main and the initial process in a banking mobile application is the login process where basically a two factor authentication is achieved. Therefore this login process should be securely designed in a well UMLSec format. In this case study fundamentally, the login process in a mobile banking application has been modelled in terms of encryption, authorization and authentication aspects of the system. Because there is a remote connection from a mobile application client to a banking web server and that should be secured, this should be stated in the design diagrams. The keyword “remote access” would be a nice matching concept for this purpose. And the communication link between mobile client and the server machine should have secure properties like encryption and integrity. For this reason the basic UMLSEC stereotype has been used.

In Figure 2, given the default adversary type, the constraint for the stereotype <<secure links>> is violated. The model does not provide communication secrecy against the default adversary, because the Internet communication link between web server and client does not provide the necessary security level according to the Threat(Internet) scenario. From this percpective threat modelling feature of UMLSec approach has been used as a better property compared to other techniques like SecureUML and others.

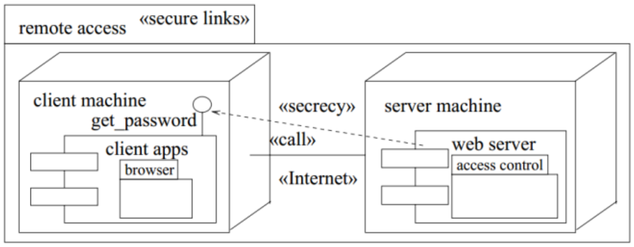


Figure 2: Secure Links usage for a basic banking scenario

On the other hand, in order to achieve the one of the most important security requirements in the design diagram, which is authentication, password based mechanism has been declared in the model. So getting password and passing it to server side for verification should be defined in the system model. the authentication Secrecy <<call>> or <<send>> dependencies in object or component diagrams stereotyped <<secrecy>> are supposed to provide secrecy for the data that is sent along them as arguments or return values of operations. This stereotype is used in the constraint for the stereotype <<secure links>>[14].

Secure dependency this stereotype, used to label subsystems containing object diagrams or static structure diagrams, ensures that the <<call>> and <<send>> dependencies between objects or subsystems respect the security requirements on the data that may be communicated along them. More specifically, the constraint enforced by this stereotype is that if there is a <<call>> or <<send>> dependency from an object C to an object D then the following conditions are fulfilled.

Additionally, the principle authorization security requirement has been defined in the banking web server side in terms of access control mechanisms. From this definition in the design phase necessary access control lists for the banking resources will be defined and later on implemented.

In a separate model, authorization issues could be designed for the specific web servers. Mainly the role based access control is used in banking systems. Just after making the right authentication, specified authorization right should be assigned to every roles in the system.

To model the encryption concepts that work in the basement of the mobile banking application, Figure 3 shows a key generation system stereotyped with the requirement <<secure dependency>>. The given specification violates the constraint for this stereotype, since the Random generator and the <<call>> dependency do not provide the security levels for random() required by Key generator[14].

Critical stereotype labels objects whose instances are critical in some way, as specified by the associated tag {secret}, the values of which are data values or attributes of the current the secrecy of which are supposed to be protected. This protection is enforced by the constraints of the stereotypes <<data security>> and <<no down - flow>> which label subsystems that contain <<critical>> objects.

This model will eventually be a role design model for all stakeholders that are involved in the software development process. In other words, it will be a kind of standardization work for the developers, analysts, testers for them to follow the defined way of making key generation in the encryption process. And for the later developments and designs, this model will be ready to reuse for other developers as a centeralized key generation library and all with its already created test cases all together. This above mentioned aprroach is one of the main motivations of this current study.

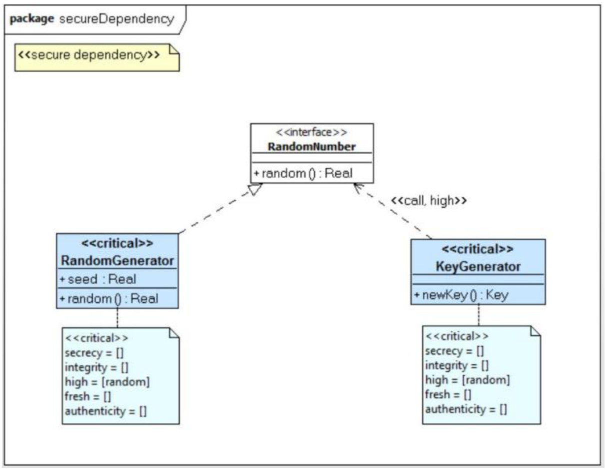


Figure 3: Secure Dependency for Key Generation

In a fundamental banking secure communication use case scenario the data and network traffic should be encrypted with a chosen crypto system. In order to make such a secure system work ahead, key generation process should be achieved. Therefore a key generation scenario has been modeled in Figure 3 by the use of secure dependency stereotype to define security properties of a running banking case.As defining the high level security issues by means of UMLSec and its provided stereotypes and tags, some security issues can be designed in work flow charts.

4.1 Graphwalker Tool

In the designing phase of the system, two-level approach has been used. Meaning that more generic features of the system like encryption, authorization have been modelled by means of UMLSec. But on the other hand more detailed steps of the running system are required. For this reason, one of the main important security aspects of the mobile communication system, which is referred to authentication has been drawn by work flows with another design tool namely Graphwalker. This tool an open source Model-based testing tool for test automation. It is designed to make it easy to design the tests using graphs. It reads models in the shape of directed graphs and generate test paths from these graphs. Therefore, by making use of Graphwalker way of designing the authentication steps will enable to generate automatically the test cases so that whole software development life cycle will be handled in model based way. In fact this was not also possible to generate test cases by making use of UMLSec method. Therefore this is the main advantage and goal of selecting Grapwalker to design models to create the detailed steps of the test cases.

In a more specific mobile banking authentication scenario, all the steps required for securely designing the system have been written down and the test cases have been generated automatically from them by using graphwalker tool. In the selected scenario, two factor authentication mechanism has been used. Firstly, the user enters his basic user credentials which are the user identification number and his password. And then secondly the user receives an SMS message where an OTP code or PIN is sent. In a specified amount of time, the user should enter his OTP code into the system authentication screens. Afterwards the user can have the right session in the mobile application and can see the right dashboard page for the application.After drawing the steps the authentication process, the proper graphwalker command has been executed and as a result the correct test cases as in the form of a path in the graph have been generated.

After drawing the steps the authentication process, the proper graphwalker command has been executed and as a result the correct test cases as in the form of a path in the graph have been generated. In the generated test cases, the items that start with "e" show that it is an edge in the graph which indicates an action in the test scenario steps. The generated items that start with "v" show the verification steps or vertices of the graph where the test result control or assertions have been done. Because the edge coverage has been set to 100, all the edges have been covered in the generation process.

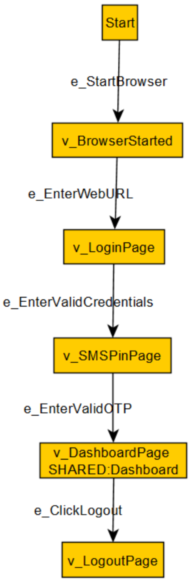
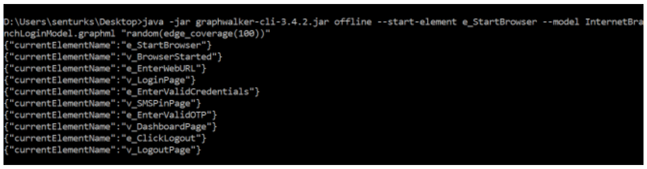


Figure 4: Model Based Way of Authentication Steps

As we have the UMLSec models designed for the system, they can be more defined as static models that are resuable components for all the stakeholders of the development process. By additionally having the graphwalker models, we have more dynamic nature in the design phase meaning that we could have chance to model more dynamic and detailed steps for our system which than will be matched to our executable test cases.

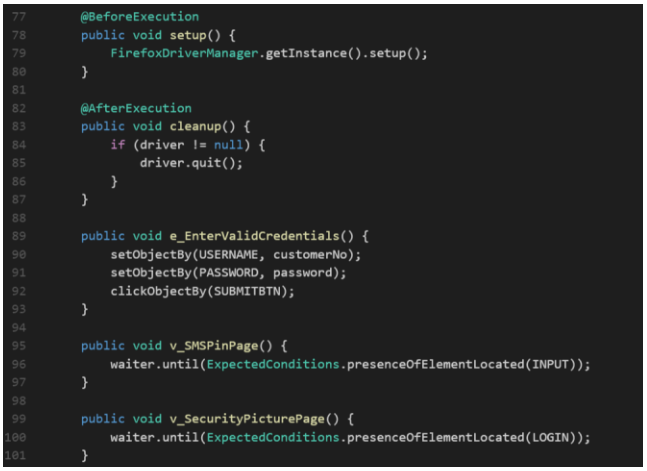
Once the test cases have been generated by using the graphwalker tool, the corresponding test code scripts have been written as well. For each vertex and edge in the graph, the right Java based test code method has been defined where the basic GUI level test controls have been handled. For every vertex where page verification has been made the concrete test code script has been written like SMS Pin Page, Security Picture Page that are all the single steps for the authentication process of the valid mobile application. The transitions that are also called edges between the vertices are the actions that can be taken by the user. And these are written as separate Java methods.

As a summary in this empirical study, encryption, authorization and all of the authentication steps of the mobile application security concerns have been modelled by using UMLSec and Graphwalker. To have the executable steps by hand, corresponding test cases and test code scripts have been generated and written. By this way main security issues at the same time would be taken into consideration in the design.



```
D:\Users\senturks\Desktop>java -jar graphwalker-cli-3.4.2.jar offline --start-element e_StartBrowser --model InternetBra
nchLoginModel.graphml "random(edge_coverage(100))"
{"currentElementName":"e_StartBrowser"}
{"currentElementName":"v_BrowserStarted"}
{"currentElementName":"e_EnterWebURL"}
{"currentElementName":"v_LoginPage"}
{"currentElementName":"e_EnterValidCredentials"}
{"currentElementName":"v_SMSPinPage"}
{"currentElementName":"e_EnterValidOTP"}
{"currentElementName":"v_DashboardPage"}
{"currentElementName":"e_ClickLogout"}
{"currentElementName":"v_LogoutPage"}
```

Figure 5: Test Case Generation by Grapwalker



```java
    @BeforeExecution
    public void setup() {
        FirefoxDriverManager.getInstance().setup();
    }

    @AfterExecution
    public void cleanup() {
        if (driver != null) {
            driver.quit();
        }
    }

    public void e_EnterValidCredentials() {
        setObjectBy(USERNAME, customerNo);
        setObjectBy(PASSWORD, password);
        clickObjectBy(SUBMITBTN);
    }

    public void v_SMSPinPage() {
        waiter.until(ExpectedConditions.presenceOfElementLocated(INPUT));
    }

    public void v_SecurityPicturePage() {
        waiter.until(ExpectedConditions.presenceOfElementLocated(LOGIN));
    }
```

Figure 6: Test Code Scripts

5 Conclusion

In this empirical case study, some of the most important security properties as the mobile application security concerns have been explored. It has been shown that security by design and privacy by design concepts are very important and such lightweight security mechanisms for confidentiality and for authorizations like access control and some other privacy issues could be embedded in to the system just in the early software development life cycle for the mobile applications. Because mobile systems have by nature heterogeneous devices and protocols, a uniform approach that covers all different complex platforms and makes an abstraction level for achieving the platform independent solutions is necessarily needed. From this perspective, securely modelled mobile application systems in the design phase of the development life cycle are crucial by starting with the platform independent models. To support that, mainly there exists UML profile mechanisms where software developers and security engineers can easily embed some security features in the mobile system modelling components. Generally speaking security requirements like authorization, authentication, confidentiality and any others are given into the requirements gathering and design phase by means of model driven security approaches.

But for almost all model driven security approaches, multiple security concerns that are handled in the design phase is very low. Among all model driven security ways, the most popular one which is taken into account is authorization. And the number of security issues that are studied at the same time together are only three like authorization, authentication and confidentiality. One open issue to study further in this research area is to find more sophisticated ways to work on more multiple security concerns for designing and deploying secure mobile systems.

Secondly, privacy by design issues are also crucial in mobile ecosystem. With privacy it is meant concealment of personal information as well as the ability to control what happens to this information[15]. Regarding the devices and objects collecting various data within mobile applications, we wish to avoid tracking or following individuals without their knowledge and/or their data being carelessly handled like e.g. being left out in any capacity in cyberspace.

Also by designing the security features of the mobile application context by providing some good samples from the UML modelling and its corresponding profile extensions, some further detailed security work flows and models around them are crucial and has been worked in this study. As an example, important authentication steps with the Graphwalker tool has been extensively designed for the mobile banking application scenario. For the future work, the other remaining security properties like authorization and privacy could be in detail designed and some automatic security test cases can be generated by using the same approach and modelling steps.

REFERENCES

[1] Nguyen, P. H., Kramer, M., Klein, J., & Le Traon, Y.(2015). An extensive systematic review on the ModelDriven Development of secure systems. Information andSoftware Technology, 68, 62-81

[2] D.Basin, J.Doser, T.Lodderstedt, Model driven security:from UML models to access control infrastructures, ACMTrans. Softw. Eng. Methodol. 15 (2006) 39–91, doi:10.1145/1125808.1125810

[3] A.Brucker, J.Doser, B.Wolff, A model transformationsemantics and analysis methodology for SecureUML, in:Lecture Notes in Computer Science, Springer, Berlin,Heidelberg, 2006,pp.306–320.

[4] D.Basin, M.Clavel, J.Doser, M.Egea, A metamodel-basedapproach for an-alyzing security-design models, in:G.Engels, B.Opdyke, D.Schmidt, F.Weil(Eds.), ModelDriven Engineering Languages and Systems, Volume 4735of Lec-ture Notes in Computer Science, Springer, Berlin,Heidelberg, 2007, pp.420–435, doi 10.1007/978-3-540-75209-7_29.

[5] D.Basin, M.Clavel, J.Doser, M.Egea, Automated analysis ofsecurity-design mod-els, Inf. Softw. Technol. 51 (5)(2009)815–831, doi: 10.1016/j.infsof.2008.05.011

[6] M.Clavel, V.Silva, C.Braga, M.Egea, Model-driven securityin practice: an in-dustrial experience , in: Model DrivenArchitecture–Foundations and Applica-tions, in: LectureNotes in Computer Science, Springer, Berlin, Heidelberg,2008,pp.326–337

[7] D.Basin, M.Clavel, M.Egea, Model-driven development ofsecurity-aware GUIs for data-centric applications,Foundations of Security Analysis and DesignVI, SpringerBerlin Heidelberg, 2011, pp.101–124

[8] D.Basin, M.Clavel, M.Egea, M.A.G. de Dios, C.Dania, Amodel-driven method-ology for developing secure datamanagement applications, IEEE Trans. Softw.Eng.40(4)(2014)324–337

[9] Jürjens, J. (2005). Secure Systems Development with UML.Springer-Verlag, Berlin/Heidelberg

[10]Chambwe, K. K. (2018). Model-based Secure SoftwareEngineering using UMLsec applied to Assisted Living and Home Care (Master's thesis)

[11] Wasserman, T. (2010). Software engineering issues for mobile application development. FoSER 2010

[12] LEAVITT, Neal. Mobile security: finally a serious problem?. Computer, 2011, 44.6: 11-14

[13] Weber, Stefan G., et al. "Towards trustworthy identity and access management for the future internet." Proc. 4th International Workshop on Trustworthy Internet of People, Things & Services (IoPTS). Vol. 29. 2010.

[14] Jürjens, J. (2002, September). UMLsec: Extending UML for secure systems development. In International Conference on The Unified Modeling Language (pp. 412-425). Springer, Berlin, Heidelberg

[15] Weber, R. H. (2010). Internet of Things – New security and privacy challenges. Computer Law & Security Review, 26(1):23–30