CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Security Modelling in Automotive Industry

Master's Thesis in Software Engineering

Shahanas Cholayil Mayankutty
Aikaterini Sereti

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2017

REPORT NO. 2017/xxxx

Security Modelling in Automotive Industry

Shahanas Cholayil Mayankutty
Aikaterini Sereti

Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
Gothenburg, Sweden 2017

Security Modelling in Automotive Industry
Shahanas Cholayil Mayankutty - Aikaterini Sereti

Supervisor: Riccardo Scandariato, Computer Science and Engineering
Supervisor: Jörger Borg, Volvo Cars Corporation
Supervisor: Henrik Broberg, Volvo Cars Corporation
Examiner: Regina Hebig, Computer Science and Engineering

Report No. 2017/xxxx
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The sign of Volvo Cars Corporation (Copyright: Volvo Cars Corporation).

Typeset in LaTeX
Gothenburg, Sweden 2017

Security Modelling in Automotive Industry
Shahanas Cholayil Mayankutty
Aikaterini Sereti
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

As electronic systems are getting more complex and expanding into more industries, it is becoming imperative that security is increased. In order to achieve this, security needs to be considered from the early stages of a system's design and be presented in the models as well. In this report, a comparative case study has been conducted.The current state of security modeling in Volvo Cars Corporation (VCC) is considered in order to suggest an improved manner of security modelling integrated in the system models. The target is to improve the security modelling in VCC in order to reach - or reduce the gap between itself and - the state-of-art in security modelling in academia. The results are discussed and used in order to give answers to the research questions set for this report.

Keywords: security, modelling, automotive,

Acknowledgements

The authors would like to express gratitude and thanks to Volvo Cars Corporation (VCC) for supporting this thesis by offering time and resources. Particularly, the authors are grateful to Jörgen Borg and Henrik Broberg for their guidance and support during the entire project. The authors would also like to express their gratitude to all the VCC employees from numerous departments who as interviewees contributed to this project. Furthermore, the authors are grateful to the Associate professor in the Software Engineering division Riccardo Scandariato for his academic supervision and support throughout the thesis. Finally, the authors are grateful to the Assistant Professor in the Software Engineering division Regina Hebig (examiner) for her contribution during the half-time presentation with valuable comments.

Shahanas Cholayil Mayankutty - Aikaterini Sereti, Gothenburg, June 7, 2017

Contents

List of Figures

List of Tables

Abbreviations

ADAS Advanced Driver Assistance Systems. 1
AUTOSAR AUtomotive Open System ARchitecture. 12

CAN Controller Area Network. 1

ECU Electronic Control Unit. 1, 2, 37

GWfM Global Workflow Model. 10

ISSRM Information System Security Risk Management. 9

LIN Local Interconnect network. 1
LWfM Local Workflow Model. 10

MBSE Model Based Security Engineering. 6, 7, 55
MDS Model Driven Security. 6, 7
MOST Media Oriented Systems Transport. 1

OEM Original Equipment Manufacturer. 37, 39
OTA-SWDL Over The Air - Software Download. 25, 38, 39, 44–46, 49, 57

RBAC Role Based Access Control. 9
RVDC Remote Vehicle Data Collection. 24, 25, 39, 46, 57

SDLC Software Development Life-Cycle. 2, 6, 7, 11
SIM Security Infrastructure Model. 10
SOA Service-Oriented Architecture. 9, 10, 27
SoaML Service-Oriented Architecture Modelling Language. 10

UML Unified Modelling Language. 12
UML4SOA Unified Modelling Language for Service-Oriented Arcitecture. 10

V2X Vehicle To (Anything Surrounding it). 1
VCC Volvo Cars Corporation. iii, 10–13, 25, 26, 29, 30, 55–57

1

Introduction

Software is now, an inevitable part of the automotive industry and is the key driver behind innovations like Autonomous drive, Connectivity and ADAS. Today cars operate using hundreds of on-board computers called Electronic Control Unit (ECU) that communicate/collaborate with each other to deliver functionalities and services within a car. Premium segment cars also have V2X connectivity, comprising of in-vehicle connectivity (Bluetooth, Wi-Fi), mobile network connectivity (telematics) and sensors (cameras, laser, radar, ultra-sonics etc.) which help them to interact with brought-in devices, other vehicles and infra-structure. Additionally, cars also generate and store data like user information, vehicle data, diagnostics data and location data. The car has evolved from being merely a simple mode of transportation into an information kernel due to the fact that they generate, process, exchange and store large amount of data [32]. The complex automotive network both in-vehicle and off-board is plagued to vulnerabilities analogous to its dependency on extensive use of software components. There is an inherent risk that these systems can be hacked and the data contained can be manipulated or stolen [32].

In-vehicle communication is handled by ECUs that monitor and control different subsystems. The ECUs communicate with each other by sending messages through a bus - a physical connection among them. The most common automotive buses are the CAN, LIN, MOST, Flex Ray and Ethernet. Many researchers have already shown that these protocols are prone to vulnerabilities [14], [13], [21] and [29]. It is possible for a hacker to gain local access to the in-vehicle network and update or even reprogram ECUs [37]. These malicious ECUs could flood the bus with messages of higher priority and prevent legitimate ECUs from sending messages. This causes serious problems to the functionality and safety of the car. [37]

The advent of connected and autonomous cars have further intensified the concerns of security experts. When the car connects itself to the Internet, there opens a whole new spectrum of security issues. The possibility that the in-vehicle network can be remotely accessed, as a consequence of interfaces that enable both wired and wireless communication, gives rise to many attack vectors coming from the outside world [37]. The ability of cars to communicate with the grid or cloud has added it to the hit-list of cyber attackers. New generation cars have the ability to communicate with the main servers via Internet in order to upgrade ECUs and system functionalities. This Over The Air (OTA) communication needs to be

designed taking into consideration the security aspect as well. If the design is not secure enough, the communication channel becomes vulnerable to hackers and can be breached. In such a case, viruses will threaten the servers and/or the car. The threat rises even more due to the fact that by gaining control of the server, the attacker can potentially gain control of all the cars as well [15].

The discussions above are some examples that illustrate the importance of security in the automotive industry. Security is a quality attribute that directly impacts the reliability and dependability of the car. A hacker can potentially take control of a critical functionality like brakes, that could be life threatening. Vehicle data can be used to build a profile of car owners that violates the privacy principle. An interesting fact is that there could also be diverse intentions behind a security attack ranging from theft, sabotage, stealing intellectual property, electronic returning of the vehicle data (mileage, diagnostic warning indicators) or simply the thrill of hacking.

1.1 Aim and Intended Contributon

Understanding the importance of integrating security in the automotive industry leads to the need for adopting the most effective techniques to do so. Modeling is an important aspect of the software engineering discipline [22]. Using models especially for big and complex systems increases awareness of the system in general and also, its details. A model illustrates the strengths and weaknesses of a system. Having demonstrated them, architects can foresee actions that are necessary to improve the system and its security by addressing the demonstrated threats/vulnerabilities. In overall, modeling techniques provide a structured way to understand the system under development.

However, a modeling technique alone may not be enough in regards with security modeling. Academia has been investigating and improving security frameworks and notations [38], [6], [28]. The motivation behind this is to ensure that security is integrated in the models and is considered throughout the entire SDLC [6].Nevertheless, the application of these notations to real-life scenarios and case studies has not been investigated to a corresponding extent [6]. This project aims to investigate the applicability of existing notations to addresses security at the design level in automotive domain.

Alongside the academic contribution of building confidence in existing security notations for industrial application, the project also aims to assist the automotive industry in listing some selection criteria while adopting a security notation.

The remaining of this document is structured as follows: Chapter 2 discusses subjects related to the thesis topic to set a proper background. Chapter 3 describes in detail the thesis design presenting every step. Chapter 4 thoroughly presents the process to select the candidate notations for the case study. Chapter 5 provides a

brief introduction to the selected notations. Chapter 6 presents the scenarios which were used for this case study. Chapter 7 discusses how the comparison was executed. Finally, Chapter 8 presents the results in regards with the research questions.

2

Background

In this section some concepts related to the project are introduced. This will contribute on understanding the scope of this research and its intended contribution.

2.1 Security and Related Terminology

Security ensures that a system has the ability to defend any attacks that may occur due to existing vulnerabilities [1]. Security is related with different aspects. Data that is sensitive requires to be secured which is defined as information security. Communication links between components in a system may require to satisfy a certain security level which corresponds to secure communication. Components themselves may contain sensitive information or functionality and, therefore, a security level needs to be met in them as well. Information security is usually realized by the CIA triad where the alphabets stand for Confidentiality, Integrity and Availability. According to [31], [20] the security terminologies are defined as follows:

- **Confidentiality**: This concern aims to ensure that only authorized personnel can view data labelled as confidential. Certain mechanisms are deployed to prevent interception of sensitive data by an attacker (unauthorized person).
- **Integrity**: This concern ensures that the data transmitted between a sender and a receiver has not been tampered with. By implementing mechanisms to achieve Integrity, the receiver would be able to confirm whether the data is accurate and complete.
- **Availability**: This concern ensures that processes and data needed for proper functioning of a system would be available to authorized users.
- **Authentication**: Defined as the ability to confirm the identity of the source of data. An attacker can threaten a system by transmitting malicious packages to the receiver pretending they are sent from a valid source. The ability to validate the sender can prevent such attacks.
- **Auditability**: This concern requires that a system is able to keep track of

all actions that have taken place. The assets of a system are audited to keep track of deviations from the agreed security policies.

- **Privacy**: Privacy is related to situations where personal data of a user may be exposed to or used by a company. Privacy supports mechanisms to prevent inappropriate use of entrusted data without prior approval.

2.2 Model Based Security Engineering (MBSE)

The importance of integrating security in a system has been emphasized by researchers [16], [28], [38]. The increasing significance of security for the automotive industry is described in the previous chapter. There is a need to consider security since the early stages of SDLC as according to [38]: "[...] for development teams to take security seriously it must be integrated into their everyday activities, i.e. security must be concomitant with software engineering practices [...]". This also calls for a structured approach using tools, techniques and methods for developing secure software systems. Both these demands motivate the necessity to integrate security into SDLC phases, as shown in Figure 2.1. According to [38] "security engineering" consists of different processes which respond to the aforementioned necessity of merging security concerns into SDLC. It consists of the following steps closely aligned to the SDLC stages.

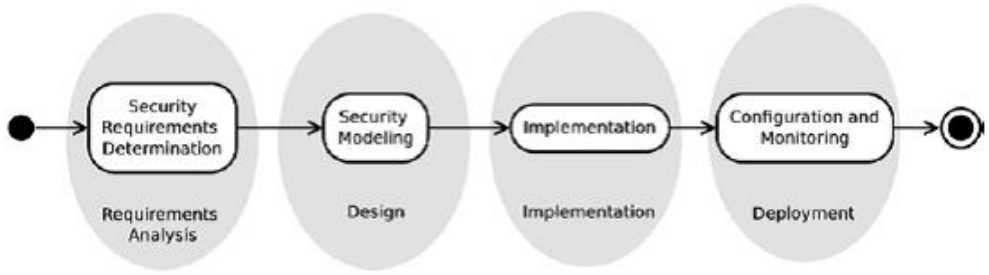

Figure 2.1: SDLC with integrated security as shown in [38]

In order to address security risks as early as possible, it is essential to be able to analyze the level of security embedded into a system early on. Using models (which are at a higher abstraction level than code) makes validation and verification possible prior to implementation. A striking social benefit of using models would be that they permit easier, faster and efficient communication. However, the lack of security constructs in generic modeling languages results in post-hoc treatment of security concerns which have negative impact on applications.There are two ways to incorporate models into the security enhanced SDLC, by adhering to Model Driven Security (MDS) or Model Based Security Engineering (MBSE). The two concepts differ in the way models are used throughout the development process. In case of

MBSE, models are not essentially the central artifacts, they are used for documentation purposes only. However, in MDS models drive the development process. This involves the use of models at every stage of the secure SDLC as well as applying Model-To-Model and Model-To-Code transformations between the stages. In this thesis, the focus will be to investigate a notation that will help the organization to represent security in system models at design level.

The paragraphs below discuss briefly the various steps involved in Security Engineering:

- Security Requirements: The trustworthiness of a system depends on how well it captures and fulfills its requirements. Requirements are elicited from stakeholders ranging from users, developers, owners, organization policies (Business cases), regulations by government and state (Legal requirements). The process of gathering security requirements in itself is a discipline consisting of various steps like identifying system assets, vulnerabilities of the system, threat modeling and risk assessment which help to extract good security requirements.

- Security Modeling: This stage of the SDLC consists of system design. At this point the architecture is implemented by adopting a particular modeling language. Integrating Security modeling, as Figure 2.1 shows, implies that additional activities are performed to visualize security properties within system models. Capturing requirements correctly and ensuring that security properties are properly visualized contributes to correct implementation of design decisions as well as test generation. Explicit modeling of security requirements into system design is crucial at this stage otherwise there is a possibility of missing important security considerations that were agreed upon by designers/architects. Having good notations that help in this task even makes verification feasible at design level. Modeling allows capture of design decisions in a graphical fashion which makes it easier to comprehend and communicate design flaws. Finally, it encapsulates the reasoning and judgment of designers which would evaporate over time if not documented [6].

  As further explained later in this document, several techniques currently exist in academia which aim to model security properties. Many of them are organized and presented briefly in [6] and [38]. UMLsec [17], [18], [7], [35], [19], SecureUML [5], [4], [8], and SecureSOA [24], [25] are some examples. These are security notations which model security properties and address the architecture and design level of SDLC. The scope of this work revolves around proper representation of security requirements in system design.

- Implementation: This is the stage where implementation of all the requirements and design decisions made so far is initiated. It is done either manually, or automatically - using code generation. The importance of representing security related information in design becomes even more pronounced when associated with implementation. Missing a security related design decision would mean incomplete or incorrect implementation thereby introducing vulnerabil-

ities in the system. Using a good notation helps to overcome the problem of loss of information from design to implementation stages.

- Configuration and Monitoring: The final stage is responsible for maintaining consistency of the final product with the initial requirements and design, as well as offer support during system run-time. Configuration focuses on documenting the system (and all its previous stages) and capturing any changes done to it in alignment with the existing documentation. From this process, the software's version is also managed. The monitoring process ensures that the product has continuous support and maintenance whenever required.

  Since hazards and threats can be evolved by potential attackers, security can benefit significantly from this process by, for instance, keeping track of versions and possessing the ability to update them. Another example of how security can benefit from this stage is that, by monitoring, threats and vulnerabilities appearing during run-time can be tracked and addressed.

Developing a secure system is challenging. It is in the requirements analysis and architecture phases where 70% of embedded software system errors are introduced. On the other hand, it is during or after integration when 80% of them are found [26]. Introducing security into each stage reduces the risk of producing low-quality vulnerable software. During each stage, threats and risks can be identified and resolved instead of putting more effort and resources to identify and resolve them in the end of the project, during deployment.

2.3 Related Work

Even though this is only an upcoming research area, there has been substantial contributions towards the body of knowledge. One of the most prominent work is UMLsec [17], that discuss "an extension of the de-facto industry standard for object oriented modeling-UML" [18], with security concerns. UMLsec is very mature and uses Use-case diagrams, Activity diagrams, Class diagrams, Sequence diagrams, State-Chart diagrams and Deployment diagrams for describing different views of a system from a security perspective [23] and [18]. The popularity of UMLsec stems from the fact that it is not domain specific and it can address a variety of security concerns (Confidentiality, Integrity, Access Control, Authentication and Cryptography [6]) that are defined as UML profile extensions using stereotypes, tagged values and constraints [6], [17] and [23] . [7] presents some interesting results obtained when UMLsec was applied for security analysis of a search engine in the intranet of a German car manufacturer. [35] and [19] discuss a model based approach that can preserve security requirements during system evolution.

Another popular contribution is SecureUML [5], that helps in defining static RBAC concepts. The meta-model of SecureUML is based on RBAC model and introduces concepts like User, Role, Permission and relationships between them, which

constitute its abstract syntax. The concrete syntax of SecureUML is based on UML and is applied to extend only the profile of UML class diagrams [23] using stereotypes. SecureUML addresses the solution domain related to enforcement of RBAC mechanisms for ensuring confidentiality [23]. The fact that this notation can define assets but cannot express attacks to these assets could be seen as a disadvantage for its widespread application. [5] provides detailed explanations on development of both the abstract and the concrete syntax of SecureUML. Furthermore, [4] enhances SecureUML by combining it with ComponentUML-a system design language and discuss automated analysis of security properties in these integrated models.[8] also explores the possibility of representing security risks in a SecureUML model. The result of this work is a mapping between model elements of SecureUML and constructs of the ISSRM domain model that would help designers consider security risk at the system design stage.

A modeling language that was not included in any of [6], [38] or [28] is SysML-Sec. SysML is an extension of UML designed to address embedded systems. According to [2] and [33], SysML-sec adds security related modeling elements to SysML. An advantage seen with this modeling notation is that it follows the V-model for system development including hardware and software partitioning stages which are exclusive to an embedded domain. Papers [2] and [33] discuss elaborate examples that showcase capabilities of SysML-Sec to model security requirements, attack trees, architectural assets and design of software components. The authors have also developed an open source toolkit named TTool that has been validated using a reference automotive system developed by the EVITA Consortium. The tool can also provide formal proofs for security properties (confidentiality and authenticity) in models using ProVerif. In the current literature studied for the thesis, one comparative case study was found focusing on UMLsec and SecureUML. This research however focused on how both notations addressed Role Based Access Control and is based only on evidence from literature [23].

One not as popular as the previous two, however, with high coverage of security concerns is SecureSOA [24], [25]. SecureSOA addresses in total four security concerns, which are Confidentiality, Integrity, Authentication and Cryptography [6]. The meta-model created in this methodology acts as a "base". This "base" is extended accordingly to address the concerns including the appropriate abstract elements. The annotation that is used in this methodology is pictorial, meaning that symbols are used to describe the security intentions in the concrete syntax [25].

Paper [27] presents a tooling framework based on SOA. This framework addresses the security concerns for SOA-based applications in an effort to make them more comprehensive to all users. The framework requires as an input a complete UML model - with the security modeling integrated - and uses libraries to transform the model to configuration files for the target system. The main target by using their suggestion is to create Web Services Security configurations based on Model Driven Architecture focusing on message protection. Through this effort, the authors have tried to address the complexity issue of generating such configurations. A similar approach is shown in [36] which suggests a framework to configure the au-

thentication security concern in Web Services Security. In the framework suggested here, the security qualifier and the SIM are linked together by the transformation of the former to a security policy for a specific platform. Although [36] focuses only on one security concern (authentication), the authors have set as future work the integration of more security concerns into their framework.

Paper [12] analyses an additional methodology related to SOAs. The authors of this paper focus on integrating the visualization of secure object flows in process-driven service-oriented architectures. Their suggested meta-model guarantees confidentiality and integrity for every secure object flow demonstrated in the model. As the other methodologies that focus on SOAs, this one, also, uses a UML extension for the meta-model. More specifically the authors define a new UML meta-model package. For the models, in addition to the newly defined package, they use SoaML and UML4SOA. The diagrams used are activity diagrams and the authors have also managed to integrate their approach with the Eclipse tools.

Another effort to link UML-based models to web-services security is described in [10]. The authors of this paper suggest a framework which was developed as part of the SECTINO project. Some of the expected benefits from the SECTINO project are "*1. Early integration of security into the engineering process, [...] 2. Correct implementation of security, [...]*" [11] which are in alignment with this project's targets. The framework which is named *Global Workflow* consists of two different models: *Interface View* and *Workflow View*. Global Workflow is a network which contains many partners who collaborate with each other in the following manner: they call services and send documents to each other. Interface View is used to represent each partner's interface regardless of how the specific partner is intended to be used. The Workflow View is separated into two sub-models: the GWfM and LWfM. The former models the message flow between partners who collaborate with each other while the later models the processes within each particular partner. Activity diagrams are used to model GWfM, class diagrams to model Interface View, class diagrams and activity diagrams to model LWfM and the target architecture is represented using XML.

This project aims to offer a different level of comparison between two notations by implementing them on industrial cases. In addition, since the thesis is conducted in cooperation with VCC, its aim impacts the organization as well by providing a security notation from the already existing ones in academia and investigating its applicability to the industry. The search of appropriate candidate notations can also provide insights to a significant question: What are the factors that guide the choice of notation?

2.4 Modeling Landscape at Volvo Cars Corporation (VCC)

Volvo Cars Corporation (VCC) is a large organization with a long history in the car manufacturing industry. Managing and conducting different activities in an organization of such scale is challenging. This section provides a detailed description of how VCC manages its necessary activities.

2.4.1 System Development Process

The corporation follows the V-model for system development. Since the system of interest within the corporation is an embedded one, the model and its concepts are adjusted accordingly. In general, the V-model includes two phases. The first, on the left is the Verification phase consisting of the following steps (from top to bottom):

- Requirements Analysis
- System Design
- Architecture Design
- Module Design and
- Implementation at the base of the V-model.

The Second, on the right is the Validation phase which includes the following steps (from bottom up):

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing

This type of model is in alignment with the secure SDLC approach that was previously discussed. It addresses all the stages in detail and provides the ability for Software Verification and Validation.

Each stage is dependent to the next and affected by the previous. The first and most important step is to define the basic functionality of the final product. After this, it is important to conduct an analysis in order to specify the requirements and move on to the design. At this point the architectural approach is decided and then a more detailed design is implemented. Completing these processes, the code

generation is initiated - either manually or automatically. Finally, another crucial part involves the testing. Testing begins with units and gradually advances to the entire system.

It is important to mention that the organization does not only employ V-model for system development but also use several models for functional deployment at different abstraction levels. The following section discusses the modeling languages that are used and purpose of the models as well.

2.4.2 Use of models in the Development Process

The two more popular modeling languages utilized in VCC are *Simulink* and *UML*. Another one that seemed to be appealing, with some efforts to establish it already in progress, is SysML.

Simulink was developed by MathWorks. It is a graphical programming language. Within the organization, though, it is applied as a modeling language as well. In the highest level of the V process, it serves the purpose of representing generic concepts of the target-product. The Simulink models created in this level are not used for code generation. Their purpose is to analyze and understand those concepts and their functionality. These models are communicated within the stakeholders in the current level, but not to other abstraction levels.

Simulink is also employed in the lowest level of the process of V-model. At this stage, the models address the implementation details of the functionality that they represent. From this abstraction level, following a specific process application code is generated and written upon AUTOSAR platform.

The next widely used language is Unified Modelling Language (UML). The architecture and design of the system/functionality is deployed using UML. Some of the implemented diagram types are Sequence, State-chart, and Object diagrams. At this level, the purpose remains the same as with the highest. The diagrams are used for better understanding and analyzing the system or sub-system of interest. In this case, as well, the models remain within the department and are not the main choice to convey the information to stakeholders outside the department.

As observed there is a disconnect between the highest and lowest abstraction levels. At the design levels the modeling language employed is UML. The popularity and maturity of this language for modeling software accounts for the choice that was made. However, the difficulty faced here is the fact that there exists no mapping from UML to Simulink.

Another significant modeling tool to mention is an in-house developed one which is well established within the organization. Its purpose is to create textual models in order to represent the requirements. It is a legacy tool with the ability to link related requirements as well as track different versions of requirements- if there

exists more than one.

2.4.3 Security as an attribute at VCC

Being a premium sector car manufacturer whose core value has been "Safety" since its early years, security has not received the corresponding attention. When compared to the strict and structured process followed in the organization for safety assurance of a vehicle, there is some ground for security to cover in order to follow a similar process. It is widely known that there is a strong relationship between safety and security because violation of security could lead to violation of safety related functionality. Given the current advancements in the automotive industry which opens up the closed in-vehicle network to a multitude of potential threats, VCC are in the process of ensuring unbiased treatment of security.

There are security objectives, security design objectives and security mechanisms/security controls in place to guide the architecture and design for enabling secure communication between the vehicle and off-board systems. All these are recorded in the form of documents. Every function in the vehicle follows these guidelines and align the subsystems and components needed to realize the functionality in accordance with this specification. When a security related functionality is to be developed, attack surfaces and attack scenarios are brainstormed and are recorded in the form of text or models. Security centric activities like Threat modeling, asset identification, capability analysis of attackers etc are carried out in a structured manner. The most common practice is to perform an FMEA to identify possible design flaws which are recorded in an Excel sheet. The same approach is used for functions that have security requirements as well.

3

Methodology

The target of this effort was to explore the different options available in academia and identify one that best suit the needs of the automotive industry. The sections that follow thoroughly explain the case study approach. Every step demonstrates an analysis of why a task was initiated and how it was executed. In this chapter, the research questions are also stated for which answers will be discussed later.

3.1 Research Questions

The research questions were directed such that they can address the aim and intended contribution of this project. After discussing the same with the academic and industrial supervisors, it was decided to approach them from two perspectives. Initially, an organization's viewpoint on the criteria to consider in order to select a candidate security notation and, then, on what criteria a security notation should fulfill in order to be applicable in the automotive industry. Therefore, the following research questions were formulated:

- **RQ1**: What criteria do companies consider important when assessing/adopting a security modeling notation?
  - **SQ1.1**:What are the necessary criteria for a security notation to fit a company and which of the existing notations in academia could fit a company?
- **RQ2**: What is the maturity level (tools available, competence required, competence available) of different security modeling notations and what is the gap to application at Volvo Cars?

3.2 Design of the Case Study Approach

The design to conduct the comparative case study includes 6 different steps. They are presented and discussed in the following sub-sections:

3.2.1 Step I - Literature Review

In order to get accustomed with the subject of the thesis, an obvious first step was to conduct a literature study. There was a need to obtain clear understanding of taxonomies used within the research area. It was also required to have an extensive comprehension of available modeling methodologies combined with or including security notations. The literature study was guided by two main sources. First, a set of three systematic literature reviews (SLR) provided by the thesis supervisor. Second, results obtained after searching on-line databases.

The systematic literature review papers [6], [38] and [28] guided the search for more papers on existing security methodologies and notations, which was a great contribution to acquiring prerequisite knowledge. Alongside the three SLR, a search was conducted in which search strings were used to query on-line databases like IEEE database, Science Direct, Citeseer and Google Scholar, the approach is summarized in Figure 3.1. The following list presents keywords and key expressions that worked as search strings to retrieve relevant papers.

- security model* automotive
- security model*
- security automotive

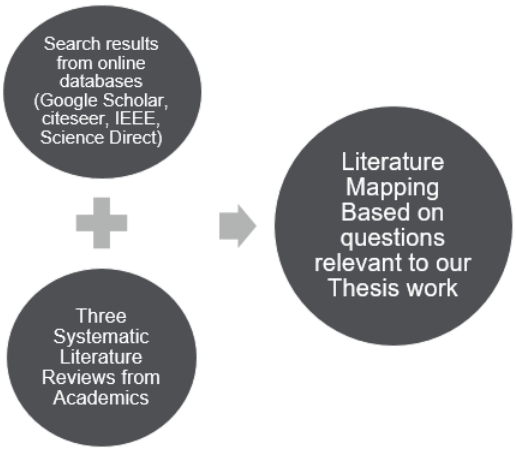

Figure 3.1: Approach for search of related literature

A total of 75 papers were retrieved from both sources. At this stage, it was

important to sort these papers according to their content and subsequent relevance to the thesis. Skimming through each of these papers, it was concluded that out of 75 only 49 papers were relevant to the thesis topic. There were many papers that had overlapping content and some papers were not directly related to the subject. A literature mapping was performed to align the 49 results with the area of inquiry. The following four questions were extracted from the research questions discussed in Section 3.1. The set of papers obtained were then scanned through to answer these questions with YES, NO or PARTLY depending on the degree to which that particular paper provided information pertaining to the question.

- **Does the paper discuss a security modeling methodology/notation?**
- **Does the paper discuss applicability of a security modeling methodology/notation?**
- **Does the paper discuss a comparison of security modeling methodology/notation?**
- **Does the paper present evidence of tools available for a security modeling methodology/notation?**

This mapping was found to be effective in helping organize and structure the pool of knowledge, an excerpt of which can be found in he following Table. The reflections/comments column contains an outline of what each paper has to offer making recall easier at later stages of this work. As a result of Step I, a notation named SysML-Sec was also found which has not been mentioned in any SLR.

Paper's Title	Authors	Does the paper discuss a security modelling methodology/notation?	Does the paper discuss applicability of a security modelling methodology/notation?	Does the paper discuss a comparison of security modelling methodology/notation?	Does the paper present evidence of tools available for a security modelling methodology/notation?	Reflections/Comments
Design Notations for Secure Software_A Systematic Literature Review	Berghe, Scandariato, Yskout, Joosen	Yes	Yes	Yes	Yes	This paper is a systematic literature review. There exists evidence related to our questions, however one has to refer to other papers to find them.
Engineering Security into Distributed Systems- A Survey of Methodologies	Uzunov, Fernandez, Falkner	Yes	Yes	Yes	Yes	The paper is systematic literature review. It discusses many methodologies comparing them with the use of tables with regards to specific subjects/issues. Whenever one or more tools are available for a methodology they are mentioned.
An extensive systematic review on the Model-Driven Development of secure systems	Nguyen, Kramer, Klein, LeTraon	Yes	Yes	Yes	Yes	The paper is a systematic literature review that discusses only Model-Driven Security Engineering Methodologies.
A Case Study of FMVEA and CHASSIS as Safety and Security Co-Analysis Method for Automotive Cyber-physical Systems	Schmittner, Ma, Schoitsch, Gruber	Yes	Yes	Yes	No	Detailed description of FMVEA and CHASSIS, description of a case study, quick comparison within one section.
A Formal Methodology Applied to Secure Over-the-Air Automotive Applications	Pedroza, Sabir Idrees, Apvrille, Roudier	Yes	Yes	No	Yes	The paper proposes to extend AVATAR to support both safety and security during all methodological stages, and in the same models.The paper applies the extended AVATAR to an over-the-air protocol for trusted firmware updates of in-vehicle control units, with a special focus on design and formal verification stages.
Model-Driven Engineering for Designing Safe and Secure Embedded Systems	Apvrille, Li, Roudier	Yes	Partly	No	Yes	Detailed description of the methodology, reference to application cases
An Access Control Concept for Novel Automotive HMI Systems	Ganzel, Schnitzer, Gilbeau-Hammoud	No	No	No	No	The paper explains an access control model which can be used for safety-critical automotive HMI systems. The model supports hierarchical granting of display permissions and allows applications to be dynamically added and removed during runtime without modifying the access control layer. They have implemented a proof-of-concept prototype which demonstrates the feasibility to implement the model .
An Active Trust Model based on Zero Knowledge Proofs for Airborne Networks	Namuduri	No	No	No	No	The paper presents a trust model. Trust is a vital concept as it enables collaboration and cooperation among any nodes in any network. This trust model gathers constantly information in order to decide what the trust level is of each of the network's nodes.
Complexity Measures for Secure Service-Oriented Software Architectures	Liu, Traore	No	No	No	No	The paper presents a sample metric for service complexity.The authors argue that software complexity is detrimental to software attackability. They have managed to set the direction towards proof of the negative impact of complexity on attacks and confirmed this to a certain extent.
SysML-Sec A Model Driven Approach for Designing Safe and Secure Systems	Roudier, Apvrille	Yes	Yes	No	Yes	This paper illustrates the application of the whole methodology of SysML-sec, along with the evaluation of a security mechanism added to an existing automotive system. The fact that integrating security requirements into the system architecture can impact the system safety is exploited by the model driven approach of SysML-sec. It promotes a collaboration between system designers and security experts at all design and development stages, e.g., requirements, attacks, partitioning, design, and validation.
SecUp: Secure and Efficient Wireless Software Updates for Vehicles	Steger, Boano, Karner	No	No	No	No	This paper presents a new idea called SecUp- which can be used for secure wireless updates. The concept uses symmetric and assymmetric keys to ensure integrity and confidentiality of data flashed inot ECUs. The framework was evaluated using STRIDE and its applicability is discussed using an example.
Securing Vehicles against Cyber Attacks	Larson, Nilsson	No	No	No	No	A good paper that enlightens on dangers of allowing wireless access to the in-vehicle network.This is a placement paper that motivates the need for enhancing security in a connected car environment with a focus on cyber security challenges.
Framework for Security and Privacy for Automotive Telematics	□Duri, Gruteser, Liu, Perez, Singh, Tang	No	No	No	No	The paper presents a Data Protection Framework (DPF) to enable building telematics computing platforms that can be trusted by both users and service providers. It introduces a framework not a methodology/notation.
Towards a Cooperative ITS Vehicle Application Oriented Security Framework	Moalla, Lonc, Labiod, Simoni	No	No	No	No	The paper discusses a new framework, CIVAS, targeted to address the entire SDLC. In the paper the framework is integrated into the ETSI reference architecture and as a future work the CIVAS prototype implementation will be completed.
Model-based Security Evaluation of Vehicular Networking Architectures	M¨uter, Freiling	No	No	No	No	The paper introduces a method to evaluate a security model. Although it is not relevant to our thesis subject, it can prove a helpful paper with regards to our decision of a good security modelling methodology.
Secure Benchmarking in the Cloud	Schroepfer, Schaad, Kerschbaum, Boehm, Jooss	No	No	No	No	The paper although relevant to security, it introduces an encrypted benchmark. It is not related to security modelling, however, it is an intersting work as a prespective of a future work when it comes to security modelling and at which state each company is comparing to its peer group.
Security and Privacy for In vehicle Networks	Schweppe, Roudier	No	No	No	No	The paper presents the concept of a new methodology as well as demonstrates its implementation, however it is not a security modelling methodology/notation.It shows how automotive communication systems can be fitted with taint tracking framework that allow to monitor data flows within and between control units to achieve elevated security and privacy.

Towards Automatic Security Management: A Model-Based Approach	Chen, Abdelwahed, Monceaux	Yes	Partly	No	No	The paper presents a methodology to model security in an automated manner. It introduces a framework which reads data in real-time, filters the relevant data and models the security. An example is used to present the methodology but nothing is mentioned regarding tool support.
Security Issues and Vulnerabilities in Connected Car Systems	Becsi, Aradi, Gaspar	No	No	No	No	Not relevant to our study. But gives good explanations of vulnerabilities and threats in a connected car. There are discussions related to vulnerabilities in ECU, Vehicular Network, Gateway, external communication channels and connected mobile devices
Model Driven Security: from UML Models to Access Control Infrastructures	Basin, Doser	Yes	Yes	Yes	No	The paper introduces the concept of Model Driven Security. It exemplifies how to extend the abstract syntax, concrete syntax and semantics of SecureUML to generate new system design languages called ComponentUML and ControllerUML. The applicability is discussed using examples.
A Meta Model for Authorisations in Application Security Systems and their Integration into RBAC Administration	Kern, Kuhlmann, Kuropka, Ruthert	Yes	Partly	No	No	The paper introduces a complex methodology to model security for specifically application systems. There exists the use of examples in the paper but no indication of tool support.
A feature-based approach for modeling role-based access control systems	Sangsig, Dae-Kyoo, Lunjin, Suntae, Sooyong	Yes	Yes	No	Yes	The paper discusses an approach which consists of feature-based security modelling. It demonstrates two case studies one as a simple example and the other as a real life scenario. There is also discussion with regards to tool support for this approach.
Supporting Security Assurance in the Context of Evolution: Modular Modeling and Analysis with UMLsec	Ruhroth, Jurjens	Yes	Partly	No	Partly	The paper discusses the combination of a lightweight and heavyweight annotation of UML for security which is the extension of UMLsec and is called UMLsec2.0. The lightweight annotation is supported by tools but not the heavyweight. There exists the use of an example to present the proposed modelling technique.
A Comparison of SecureUML and UMLsec for Role-Based Access Control	Matulevicius, Dumas	Yes	Yes	Yes	No	The paper presents a comparison of two security modelling languages- UMLsec and SecureUML. The comparison is based on literature study and a small example.
Developing secure data warehouses with a UML extension	Fernandez-Medina, Trujillo, Villarroel, Piattini	Yes	Yes	No	Yes	The paper discusses the development of a UML extension in order to design and model security for Multidimensional Databases and Data Warehouses. It presents an example followed later in the paper by a case study. It also discusses the development of a tool to support the security modelling methodology.
UML specification of access control policies and their formal verification	Koch, Parisi-Presicce	Yes	Partly	No	Yes	The paper describes a UML annotation used to model security. There exists an example based on real life and also indication about tool support.
Using UMLsec and Goal Trees for Secure Systems Development	Jurjens	Yes	Yes	No	No	The paper discuss the integration of use-case driven approach for realizing functional requirements and Goal driven approach for realizing security requirements. There is a very simple example illustrated to show how this can be done. The example is NOT a real case.
Model-based Security Engineering of Distributed Information Systems using UMLsec	Best, Jurjens, Nuseibeh	Yes	Yes	No	Yes	The paper presents the results of the security analysis of a corporate meta search engine in the intranet of a German car manufacturer. The security critical parts of the system were analyzed using UMLsec. It is an industrial case study that demonstrates the use of UMLsec to show its benefits and limitations.
Incremental Security Verification for Evolving UMLsec models	Jurjens, Marchal, Ochoa, Schmidt	Yes	No	No	Yes	The paper discusses the use of a further extension of UMLsec and a validation approach. Although it uses some simple examples to elaborate on the description they are not used in real life. There is also a section dedicated to tool support.
Modeling Complex Systems by Separating Application and Security Concerns	Gomaa, Eonsuk Shin	Yes	Partly	No	No	This paper presents a new methodology to model security by separating the security attributes from the business attributes. This methodology is addressed to complex applications and it is useful as it supports reusability of the security models for other applications as well.
Security in Model Driven Development: A survey	Jensen, Gilje Jaatun	Yes	Partly	Yes	No	This paper is a systematic survey and presents several existing security modelling methodologies. Each methodology is briefly discribed and the authors discuss its strength and weaknesses.
A Review of Approaches to Model Security into Software Systems	Hussain, Rasool, Atef, Shahid	Yes	No	Yes	Yes	The authors make a distinction between external and internal security. The paper mainly address methodologies that are concerned with internal security. The survey has focussed on four parameters to classify the methodologies: model driven methodologies, methodologies having automatic tool support, methodologies having no tool support and methodologies based on formal methods. A critical analysis of the methodologies is also presented.
A Survey of Modelling and Analysis Approaches for Architecting Secure Software Systems	Dai, Cooper	Yes	Partly	Yes	Partly	The authors of this paper have conducted a survey and present categorized security modelling methodologies. The paper presents briefly the methodologies and mentions examples or real life situations they were used in. Tool support is mentioned for some of the methodologies.
Sound Methods and Effective Tools for Model Based Security Engineering with UML	Jan Jurens	Yes	Yes	No	Yes	The work aims to contribute towards usage of UML for secure systems development in practice by offering automated analysis routines connected to popular CASE tools. The approach was applied to a real industrial biometric authentication system, where it found and corrected several serious design flaws.
Basic Concepts and Taxonomy of Dependable and Secure Computing	Avi zienis, Laprie, Randell, Landwehr	No	No	No	No	This paper gives definitions relating to dependability and Security. The definitions are explained clearly along with additional definitions that address threats to dependability and Security. It presents a clear understanding of classifications of these threats and definitions of mitigation techniques. Good paper to get an overview of all definitions and terms used in this area.

Automated Analysis of Security-Design Models	Basin, Clavel, Doser, Egea	Yes	Partly	No	Yes	This paper uses an already introduced methodology, SecureUML, and automates the analysis of the models derived from SecureUML. It describes in detail the steps to be followed for an automated process and uses examples to demonstrate it. To carry out with modelling and analysing the examples the authors have developed the SecureMOVA tool which is an extension of the MOVA tool.
Using Aspects to Design a Secure System	Georg, Ray, France	Yes	No	No	No	This paper introduces an aspect oriented methodology which uses role models to model security. The role models it uses are IRM (Interaction Role Models) and SRM (Static Role Models). The paper does not use any examples to support the methodology description but the authors are currently developing a tool prototype to support their suggested methodology.
A system for Visual Role Based policy modelling	Giordano, Polese, Scanniello, Tortora	Yes	Yes	No	Yes	This paper presents a suite of visual languages to specify access and security policies according to Role Based Access Control (RBAC).The system provides a set of tools to enable a user to visually edit security policies and to successively translate them into (eXtensibleAccessControlMarkupLanguage) XACML code, The visual approach is tested using a usability study.
Model-Driven Security Engineering for Trust Management in SECTET	Alam, Breu, Hafner	No	No	No	No	This paper presents the SECTET-framework which aligns high-level security objectives identified in the Business Services with their implementation as Privacy Security Trust (PST) technologies. In addition, the authors specify dynamic constraint policies using language SECTET-PL.
Modeling and Performance Analysis for Security Aspects	Dai, Cooper	Yes	Yes	No	Yes	The paper introduces Formal Design Analysis Framework (FDAF) which is an aspect-oriented approach that supports the design and analysis of multiple non-functional properties for distributed, real-time systems. In this paper, a security attribute, data origin authentication, is defined as a reusable aspect based on its security pattern definition.The paper does discuss about tool support for FDAF
Using FDAF to bridge the gap between enterprise and software architectures for Security	Dai, Cooper	Yes	Yes	No	No	This paper discusses the framework FDAF (Formal Design Analysis Framework) using it in an empirical study for an online banking system. In this paper the framework includes the use of extended UML annotations to cover the security aspect in the derived models.
A model-driven approach for the visual specification of Role-Based Access Control Policies in web systems	Diaz, Aedo, Sanz, Malizia	Yes	Partly	No	Yes	This paper is related to the ADM-RBAC approach. The authors very briefly describe the approach and focus on validating the models derived by applying the approach. There is a tool, AriadneTool, mentioned indicating its support to ADM-RBAC.
A Security Meta-Model for Service-oriented Architectures (Secure-SOA)	Menzel, Meinel	Yes	Yes	No	No	The model in this paper lays the foundation to describe and implement a model driven transformation of security intentions to enforceable security configurations. The approach is based on the integration of security annotations in visual modelling notations.
Adding Authentication to Model Driven Security	Satoh, Nakamura, Ono	Yes	Partly	No	No	This paper introduces a framework which addresses the difficulties that can occur while using a complex security configuration. TThe paper focuses on the authentication configuration and uses an example to present the proposed framework.There are indications that the framework is tool-independent.
SecureSOA - Modelling Security Requirements for Service-oriented Architectures	Menzel, Meinel	Yes	Yes	No	No	This paper introduces the abstract and concrete syntax of Secure-SOA. There is an explanation of how Secure-SOA can be integrated with FMC (Fundamental Modelling Concepts Block Diagram). The applicability is discussed by using a small web shop example.
Model-Driven Security Based on a Web Services Security Architecture	Nakamura, Tatsubori, Imamura, Ono	Yes	Partly	No	Yes	The paper proposes a framework to model security. They have also developed a tool which is friendly to non-security experts and in the paper they use an example for demonstration purposes.
Security modeling for service-oriented systems using security pattern refinement approach	Memon, Menghwar, Depar, Jalbani, Mashwani	Yes	Partly	No	No	The paper proposes the refinement of a security pattern. In the paper an example is used to demonstrate each step suggested.The pattern the authors propose includes stages of the entire SDLC from modeling to transformation to executable code. No tool support is mentioned for the modeling part of the pattern, it is mentioned, however, for the transformation from model to executable code.
Modeling and enforcing secure object flows in process-driven SOAs: an integrated model-driven approach	Hoisl, Soberning, Strembeck	Yes	Yes	No	Yes	The paper proposes a security notation which focuses on Service Oriented Architectures.There is a detailed explanation of how the authors reached at the proposed notation and how the user can use the notation. The paper also discusses a case study conducted in health industry and the available tool support through an existing tool.
Modelling Inter-organizational Workflow Security in a Peer-to-Peer Environment	Hafner, Breu, Breu, Nowak	Yes	Yes	No	No	This paper presents another security notation for Service Oriented Architecture systems. The paper presents the notation and its structure and also discusses how the notation can be used. Although there is not a specific case study in the paper, the notation itself was developed as part of an industrial project, the SECTINO project, which could act as a case study.

3.2.2 Step II - Stakeholder Interviews

The interviews were conducted in a semi-structured format. Despite being semi-structured, some questions were prepared - shown later in this Sub-Section - in order to guide the interviewers and the interviewees. The interview questions were formulated in accordance to the research questions discussed in Section 3.1. More specifically, the interview questions 1 with its sub-questions, 2, and 4 with its sub-questions provide answers that can be cautiously generalized to answer research questions RQ1 and SQ1.1. RQ2 can be answered after the comparison is done. The interviews were conducted in two phases as described below:

3.2.2.1 Phase I

It was of great significance to get acquainted with the organizational structure and ways of working within the company, as this knowledge was required to propose an appropriate modeling notation. A good way to comprehend the company's structure was to conduct interviews with people from different abstraction levels. In total, 5 interviews were held serving this purpose. The discussions focused on the company's approach that lead to the production of an effective system. The content of these discussions were very generic and provided guidance on adapting to the organization's attitude. As a consequence, there was made an initial mapping of how security is represented in each abstraction level as well as what are the main concepts.

3.2.2.2 Phase II

The motivation behind Phase II was to extract information with regards to modeling approaches followed in different departments of the organization. Another set of 8 interviews were held. These interviews were arranged with people who are responsible for realizing different functionalities. Some of these people were on a lower abstraction level and the modeling languages used did not address design level, as the modeling notations this thesis focused on. The models in those levels represented implementation algorithms and code structure to be followed. Among the people interviewed were those responsible for functional/system safety, verification and validation. It was very interesting to understand the approach in such situations as safety and security are closely related to each other. In addition, discussions had also been held with people who were at the desired abstraction level (system architecture and design)- according to the security notations in focus. These discussions provided high level details concerning overall security design objectives and mechanisms implemented to realize them. The results of Step II are treated with more detail in Section 2.4.2.

The questions as they were prepared to guide the interviews:

1. What are the types of models currently being used?
   - 1.1 What is the level of abstraction for the models?
   - 1.2 What are the visualised concepts?
   - 1.3 Which is the used notation?
     - 1.3.1 With which modeling language?
     - 1.3.2 What type of diagrams are employed?
     - 1.3.3 Which tool is used?
2. What are the security needs/requirements?
3. What is the purpose of the models?
   (a) Documentation?
   (b) Analysis?
   (c) Use during the entire development process?
4. Which of the following security concerns are required to be illustrated in the models?
   (a) Confidentiality?
   (b) Integrity?
   (c) Availability?
   (d) Auditability?
   (e) Privacy?
   (f) Access Control?
   (g) Authentication?
   (h) Logging?
   (i) Cryptography?
   - 4..1 Which security concerns are the most important?
5. Is tool support significant for the suggested notation?

3.2.3 Step III - Candidate Filtering

The interviews were not recorded however, extensive notes of all the subjects discussed were taken. To avoid any risk of omitting important information mentioned during the interviews, an after-interview document was created in Excel. This document contained a grid of various levels in the organization, combined with the names and designation of people interviewed. The grid was filled each time right after an interview with everything that was discussed.

The grid proved useful to structure and interpret the data collected. A deductive data analysis strategy was followed to extract relevant data from the interviews. Initially, the data was scrutinized in search of repetitive terms. Through repetition, one can conclude that there is a requirement related to that term. At the end of this search, an initial set of criteria was defined. The next step for data analysis was based on the descriptions added in the grid. Considering what the interviewees mentioned as being important and if - and how - each description was related to others in the grid, the final set of criteria was partly defined.

3.2.3.1 Candidate Filtering I

It was concluded from data analysis that tool support for the selected notation was an important criteria for Volvo. The motivation behind this argument was the fact that without a tool the models would remain as PowerPoint presentations which was not the intended outcome of this thesis work. As a first step, it was thus decided to consider only those notations that displayed evidence of existing tool support. It is important to note here that, notations with prototype tools are not considered. While interviewing stakeholders related to security activities(in Phase II), there was a demand that the notation should be able to satisfy, if not all at least two of the CIA (Confidentiality, Integrity and Availability) properties.From the existing pool of candidate notations it was thus decided that an initial filtering shall be done and the two questions guiding this stage of filtering would be:

- **Does the notation have tool support?**
- **Is the notation capable of addressing more than one security concern?**

The 30 security notations were put through a selection process that is explained in detail in Chapter 4, Section 4.1.At the end of this selection process only 5 notations remained namely-UMLsec, SysML-Sec, Secure-SOA, SOA by Hoisl et.al and Secure Tropos.

3.2.3.2 Candidate Filtering II

As mentioned previously, the stakeholder interviews contained more information from which another list, a more detailed and specific one, was extracted. Studying and understanding the needs of the organization with regards to security modeling led to this list. Another approach followed to enrich the second list of criteria was brainstorming. Considering a wide range of criteria relevant to the organization and academics ensured that the risk to the process would be minimized and no important criteria would be unintentionally excluded. A total of 14 criteria were decided, out of which 9 were of industrial importance (specific to Volvo) and 5 were of combined pertinence (both industrial and academic). The criteria constituting the decision matrix and the motivation behind each is discussed in Chapter 4, Section 4.2.

The five candidate notations chosen at the end of Section 3.2.3.1, UMLsec, SysML-Sec, Secure-SOA, SOA by Hoisl et.al and Secure Tropos were filtered again using the grid of 14 criteria. If there exists evidence that the notation could satisfy a given criterion, it is marked with a green dot. For partial evidence of criterion fulfillment, it is marked with a yellow dot and a red dot for negative criterion satisfaction. The total number of criteria fulfilled by each notation were added and two notations with the highest number were selected for detailed study. The selection of UMLsec and SysML-Sec, which were the modeling notations chosen for implementation, is shown in Figure 4.2.

3.2.3.3 Stakeholder Workshop

At the end of this Step, a stakeholder workshop was held to gather consensus for the methodology used for candidate filtering. A total of 6 stakeholders participated in the workshop. There was detailed discussion on the list of 14 criteria, as well as the motivation/reasoning for each criterion, which can be found in the following Chapter. As the list itself was based on stakeholder demands, there were no disagreements. One interesting argument put forth by a stakeholder was the fact that none of the stereotypes had icons attached to them (except Secure-SOA). It was unanimously agreed that icons promoted understandability and expressiveness of a notation. However, the chosen candidates UMLsec and SysML-Sec do not use icons(pictorial representations) on their stereotypes.

3.2.4 Step IV - Scenarios for Case Study

Before implementing the two chosen modeling notations, it was vital to organize and structure the approach. As a first step it was important to decide which scenarios that were met in Step II would be utilized for the case study. Although only one scenario was considered enough, it was decided to proceed with two to enforce better ground for comparison. Hence the selection of Remote Vehicle Data Collection

(RVDC) and Over The Air - Software Download (OTA-SWDL), which are explained in Chapter 6. Due to the confidentiality agreement with VCC, it was resolved to elicit some generic threats and requirements related to OTA and RVDC instead of exploiting Volvo specific information. In this case, consulting experts from VCC was essential to validate that the threats and requirements were realistic and did not disclose sensitive details. The aforementioned Chapter provides analyses of threats and requirements for each case.

3.2.5 Step V - Comparative Case Study Setting

Succeeding the definition of realistic and generic cases, was structuring the approach. Since it was agreed upon having two cases in combination with two selected modeling notations, it was opted to follow a strategy demonstrated by the Figure 3.2. There were two selected notations implemented using two selected scenarios. The resulting models were utilized to compare how the modeling notation behaved against every single security requirement for each scenario. Figure 3.3 demonstrates how the two scenarios combined with the two notations were handled. More specifically, Student 1 implemented OTA-SWDL using UMLsec and RVDC using SysMLsec. Similarly Student 2 implemented OTA-SWDL with SysMLsec and RVDC with UMLsec.

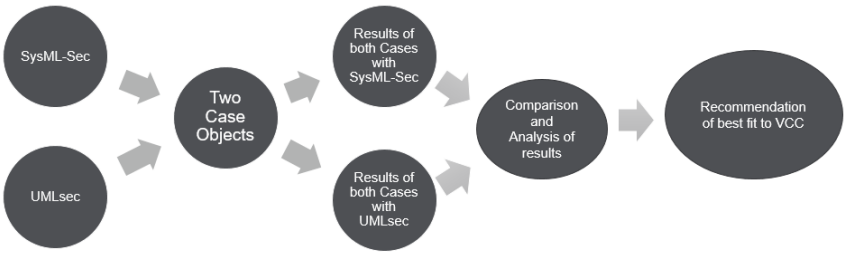

Figure 3.2: Overall approach of case study

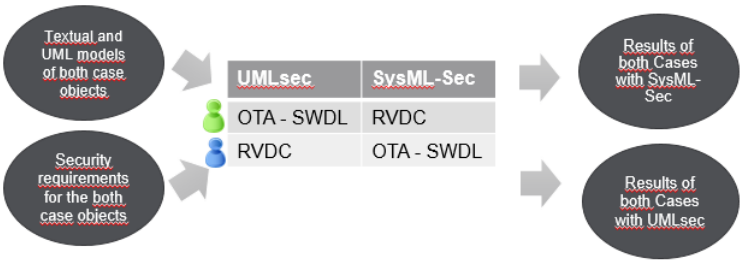

Figure 3.3: Approach for modeling the two cases

3.2.6 Step VI - Evaluation Criteria

Comparing between two notations requires well formed evaluation criteria. In an attempt to understand **how** and **what** to evaluate two strategies were adopted-

Reading other work concerning evaluation of modeling notations and questioning stakeholders about their expectation from a modeling notation.

During the stakeholder workshop, it was concurred that the two qualities expected from a notation were ease of use and expressive power. Since these two requirements were abstract in nature, it was necessary to break them down into related concepts. To help with this refinement it was decided to refer to the cognitive dimension framework [9] for evaluating modeling notations. The list of evaluation criteria were decided upon in compliance with VCC stakeholders and the output models from Step IV were compared against them. The list is given below:

1. Ease of use
   (a) Documentation to support learnability
   (b) Range of Diagrams that can be annotated using the notation
   (c) Dependencies or constraints that guide the use of symbols in the notation
2. Expressive Capability
   (a) Extend to which the notation can express security concerns and other security related information
   (b) Does the notation convey its intended meaning without confusing the reader
   (c) How obvious is the role of a symbol used in the notation

The answers to criteria 1(a), 1(b), 1(c) and 2(a) were obtained during/from the case study described in Step IV (Section 3.2.4). To answer 2(b) and 2(c) the output models from Step IV were put through an understandability exercise involving 2 security-related stakeholders from VCC. The subjects were asked to describe what they understood from the security enriched models presented to them. To avoid influencing their responses, the security notations were not explained in advance. However, the two scenarios used to conduct the case study were presented. The answers were then compared to the semantics of each notation to extract the gap in perception.

Based on the comparison mentioned above, it was possible to perform an analysis of both modeling notations. From this analysis, a best fit notation was suggested for the concerned organization.

4

Candidate Selection Process

4.1 Candidate Filtering I

Considering the two questions mentioned in subsection 3.2.3.1 several candidate notations were excluded. A detailed table is shown in Figure 4.1. As can be seen from the table only 6 notations had tool support (UMLsec, SecureUML, SysML-Sec, SECTET, SOA by Hoisl et.al and Secure Tropos). Additionally, only 11 notations could address more than one security concern. It was difficult to analyse how many security concerns were addressed by Secure Tropos. Secure Tropos was essentially a methodology that combined two modeling notations, i* for security requirements and and UMLsec for secure design. However, it was decided to include Secure Tropos as a candidate due to its mature tool support and coverage. There were only 4 notations that satisfied both the stated conditions (UMLsec, SysML-Sec, SECTET, and SOA by Hoisl et.al). On analyzing SECTET, it was found that the notation addressed security issues related to inter-organizational work-flow scenarios. Thus, it was decided to exclude SECTET in-spite of the notation being a qualified candidate.

Another notation named Secure-SOA was included as a candidate because there was an imperative demand from the organization to consider notations supporting security in Service-Oriented Architecture (SOA). This notation was chosen as it could address 4 security concerns even though it did not have tool support. This trade-off was also approved by Volvo and the final 5 notations chosen after initial filtering were UMLsec, SysML-Sec, Secure-SOA, SOA by Hoisl et.al and Secure Tropos.

4.2 Candidate Filtering II

The five notations selected after candidate filtering I were subjected to another selection process guided by a decision matrix. Taking into account both industrial and academic perspectives for adapting a notation a total of 14 criteria were elicited. It should be noted that the criteria discussed below are not ordered/rated according to their importance.

Name of the Notation	Number of security Concerns Covered	Availability of Tool Support	Comments
UMLsec	5	Yes	Chosen
SecureSOA	4	No	Chosen because there is a proposal to shift towards service oriented architectures and notations to support security in SOA
NakamuraSOA	3	No	
Hoisl-SOA	3	Yes	Chosen
Hafner-SOA	3	No	
Gomaa-UML	3	No	
SysML-Sec	2	Yes	Chosen
Medina-DB	2	No	
Memon-SECTET	2	Yes	Qualified, yet not chosen because it is used for peer-to-peer and inter-organizational workflow scenarios
Vela-DB-XML	2	No	
FDAF	2	No	
SecureUML	1	Yes	Not Chosen as it can only address RBAC
Ahn-AC	1	No	
Alam-SECTET	1	No	
AMF	1	No	
Buyens-LP	1	No	
ADM-RBAC	1	No	
Georg-AO	1	No	
Giordano-AC	1	No	
Kim-AC	1	No	
Kong-Threat	1	No	
Mariscal-AC	1	No	
PbSD	1	No	
Ray-AC	1	No	
Sohr-AC	1	No	
UML AC	1	No	
UMLS	1	No	
Xu-Petri	1	No	
Yu-AC	1	No	
Secure Tropos	??	Yes	This is a framwork comprising of two notations, i* modelling language for security requirements model and UMLsec for security enhanced design models

Figure 4.1: Initial list of selection criteria

- One most frequently suggested requirement was that the notation should provide tool support. In the absence of tool support designers could resort to using tools best suited to them. Differences in the capabilities of tools used could result in inconsistent use of the notation. Besides, to be able to annotate system design models the tool should be compatible with the tools currently used for modeling. If not, there would exist isolated models created with different tools which is not desirable.

- Adopting a notation requires that the designers learn to use it correctly. To support this, it is important that the notation has sufficient documentation providing explanation of its semantics. Documentation is essential to reduce errors as well as modeling time. Also, an increased learning curve due lack of resources might not be favourable for the industry.

- A security notation should be able to represent many security properties. The requirement put forth during stakeholder interviews was that the notation should be able to annotate, if not all, at least the CIA triad. The motivation behind this criteria is that in a company it is not desirable to use different notations to get intended coverage of security issues.

- Another demand was the possibility to trace security requirements through various activities. For Volvo, the purpose of having security enhanced system models is to support documentation and analysis of functionality. Besides, the organization employs an in-house developed tool for requirements management and traceability. During the stakeholder workshop it was discussed that the requirement management tool could be replaced and that it should not influence the choice of notation. Thus this criterion was considered to be of less importance for the concerned company but of importance from an academic perspective.

- VCC stakeholders favour that the proposed notation enables proper documentation and analysis of security requirements. Documentation is required to capture security related design decisions and analysis support helps to identify design flaws earlier that could become vulnerabilities later on.This motivation is relevant to academia too.

- Alignment of the symbols used by the notation with the representation of security requirements in the VCC requirements tool is a criteria that was considered out of scope for this work. Currently, there is no explicit representation of security requirements in the tool which renders this criterion unwarranted.

- It would be easier to adapt to a notation that is based on the modeling language currently being used in the company. A completely new dialect (a new modeling language and a notation related to it) would introduce too much overhead into the development process. For Volvo, it would be better to use notations that are build upon UML or SysML. Thus, the choice of a notation should take into account the existing knowledge base of intended users.

- It was found during stakeholder interviews that not all designers are security specialists. It is desirable that the notation is intuitive enough for non-security personnel to understand thus allowing widespread acceptance. A notation that does not require high security expertise for application can be used by all.

- There are different abstraction levels at Volvo (complete vehicle, system, component) and it would be beneficial if one notation could suit all of them. Communication between the levels become comprehensible and coherent due to use of a single notation. From the above motivation it follows that a notation should be able to annotate security related information pertaining to all abstraction levels.

- Volvo wishes to implement a security engineering approach in the near future. In that case, it would be convenient to keep the same notation throughout the life cycle to avoid mapping constructs of one notation (like symbols) to those of another. There is an imperative need that the notation should be able to address security throughout the system development life cycle to avoid complexities that come along with the mapping.

- Another frequently asked question was *"What are the security notations used by other companies?"* which leads to the conclusion that a popular notation is favoured for adoption. If a notation has been validated by case studies it has probably matured according to the demands put forth by the industry.

- Since the industry of interest focuses on embedded systems it would be favourable if the notation is able to support various facets of an embedded domain. However, since the notations offered by academia are not specifically designed for embedded systems, except one, this criterion is considered with caution.

- VCC is motivated to change their ways of working from document centric to model centric. For the notation to consider the intended future way of working in the organization it should support model based or model driven development activities.

- At VCC, Simulink is used at the complete vehicle and component levels. The modeling language used at design level is UML and SysML. From this scenario there emerged a criterion that the notation should support mapping to Simulink. This condition is not achievable as it is out of the thesis scope.

As seen from Figure 4.2, UMLsec satisfies 7 and SysML-sec satisfies 8 out of the 14 criteria. Thus these two notations were chosen for implementation in the case study.

		Criterias Considered for Filtering of Notations	UMLsec	SysML-Sec	SecureSOA	HoisISOA	Secure Tropos
Academic & Industrial	1	Availability of tool support	🟢	🟢	🔴	🟢	🟢
	2	Availability of proper documentation facilitiating easy learnability/understandability of the notation	🟢	🟢	🟡	🟡	NOT SURE
	3	Number of Security concerns covered	5	2	3	3	NOT SURE
	4	The notation should enable traceability of security requirements through the variuos activities.	🟡	🟡	🟡	🟡	🟡
	5	The notation should allow verification/analysis of security concerns in the system design	🟢	🟢	🟢	🟢	🟢
Industrial	6	Alignment with representation of security requirements in Elektra (Volvo Specific)	🔴	🔴	🔴	🔴	🔴
	7	The notation should take into consideration the existing knowledge base of intended users with respect to modeling (For Volvo it is UML and SysML)	🟢	🟢	🔴	🔴	🟡
	8	The notation should not require high secuirty expertise for application	🟢	🟢	🟢	🔴	🟡
	9	The notation should be capable of addressing the different abstarction levels in the organization (Volvo Specific)	🔴	🔴	🔴	🔴	🔴
	10	The notation should be able to address security throughout the entire system development life cycle (Requirement Specification, Design, Implementation)	🟡	🟡	🔴	🔴	🔴
	11	The notation should have been validated using case studies to build confidence in its real life applicability	🟢	🟢	🔴	🔴	🟢
	12	The notation should be able to support variuos facets of an embedded system domain (Volvo Specific)	🔴	🟢	🔴	🔴	🔴
	13	The notation should take into consideration the current/future way of working in the organization	🟢	🟢	🟢	🟢	🟢
	14	The notation should be mapped to Simulink, since Simulink is used in high (Complete Vehicle) and low (Component) levels. (Volvo Specific)	🔴	🔴	🔴	🔴	🔴
		Total number of criteria satisfied by each candidate	**7**	**8**	**3**	**3**	**4**

Legend: Color	Meaning
🟢	Completely satisfies the criterion
🔴	Does not satisfy the criterion
🟡	Partially satisfies the criterion
NOT SURE	Could not get enough information from literature to support satisfaction of criterion

Figure 4.2: Final list of selection criteria

5

A primer on UMLsec and SysML-sec

This section focuses on discussing UMLsec and SysML-sec. This is an introduction to the notations and how they are used in order to annotate and address security concerns in a system model.

5.1 UMLsec

UMLsec is a lightweight extension of UML. The UML diagrams are annotated with *stereotypes* and *tags* in order to integrate security concerns in the models. As UMLsec is a large notation with a lot of stereotypes, this description will focus on the stereotypes that are required to understand the following Chapters. The stereotypes and associated tags in UMLsec are discussed in Table 5.1.

Stereotype	Associated Tags	Description
«provable»	{*action*=state_of_action _to_prove}, {*cert*=expression_that _proves_action}, {*adversary*=adversary_type}	Demonstrates that there exists proof that an action in the model has occurred.
«rbac»	{*protected*=activity}, {*role*=(actor,role)}, {*right*=(role, right)}	Requires an Activity diagram to model *Role-Based Access Control* in a subsystem.
«Internet», «encrypted», «LAN», «wire»		Used on communication links in a subsystem.

«secrecy», «integrity», «high»		Used to label dependencies between elements when the data sent between them requires the respective security properties. These stereotypes are combined with the *«secure links»* and *«secure dependency»* stereotypes.
«secure links»	{*adversary*=adversary_type}	Used to ensure that the physical layer meets the security requirements on communication links. Stereotypes *«secrecy»*, *«integrity»* and *«high»* can be used with this one.
«secure dependency»		When there exist a *«call»* and *«send»* dependency between elements exchanging data, *«secure dependency»* enforces that data security requirements remain consistent by employing *«secrecy»*, *«integrity»* and *«high»* stereotypes.
«data security»	{*adversary*=adversary_type}, {*integrity*=(variable, expression)}, {*authenticity=(data, origin)*}	With respect to the type of adversary, it enforces basic data security requirements. Combined with the stereotype *«critical»*, more security requirements can be addressed.
«critical»	{*secrecy*=data}, {*integrity*=(variable, expression)}, {*authenticity=(data, origin)*}, {*high*=message}, {*fresh*=data}	Used on objects handling critical data. Combined with the stereotype *«data security»* to enforce the respective security requirements on the critical object(s) of a subsystem.

Table 5.1: UMLsec stereotypes and tags used in the project

5.2 SysML-sec

SysML-sec is a notation based on SysML. Like SysML, the notation aims to model embedded systems - its software and hardware components. With SysML-sec both Safety and Security can be integrated in the system models. SysML, being an extension of UML, models a system using Block diagrams, Activity diagrams, State diagrams, Sequence diagrams and Use-case diagrams. To annotate security on the design level, SysML-sec uses block diagrams.Two security-related elements introduced in SysML-sec are *Property Pragmas* and *Cryptoblock*.

- **Property Pragma**: Pragmas can be found in Block diagrams. A Property Pragma requires State diagrams to be combined with and addresses *Authentication* and *Confidentiality* security concerns as follows:

  - **Authenticity** is represented as follows in the Pragma: *#Authenticity b1.s1.e b2.s2.e* . *b1* and *b2* represent two blocks, sender and receiver respectively, communicating with each other to send *e*. *s1* and *s2* are the states of interest and which are shown in the corresponding State diagrams of blocks *b1* and *b2*. *e* is the element, for example a message or a file, sent from *b1* to *b2*. *Authenticity* in the Property Pragma in this case ensures that the state of element *e* **after** state *s1* of block *b1* matches the state of *e* **before** state *s2* of block *b2*. Therefore, during the transmission of *e* no changes have occurred to it.

  - **Confidentiality** is represented as *#Confidentiality blockY.attributeX* . This means that in the Block diagram there exist a block named *blockY* which contains an attribute named *attributeX*. *Confidentiality* here ensures that *attributeX* remains confidential.

- **Model Pragma**: With a *Model Pragma* the Initial Knowledge can be set which refers to the attributes which are known when the system or a session starts. To model it the *InitialSystemKnowledge* and/or *InitialSessionKnowledge* are required.

- **Cryptoblock**: is a block that includes a predefined set of cryptographic methods. A *Cryptoblock* provides *Confidentiality* to the asset it represents. In addition, the cryptographic methods can be used in its State diagram to demonstrate what actions occur during transition from one state to the next. For instance, cryptoblock *b1* has the predefined method *sencrypt(Message msg,Key k)*. Among the states of its State diagram are the following: *CreateMessage* and *TransmitMessage*. The transition arrow between these two states can be noted as *msg1 = sencrypt(m, key)* which is translated to the action that *b1* encrypts message *m* into *msg1* with the symmetric encryption method *sencrypt*.

6

Scenarios for the comparative evaluation of UMLsec and SysML-sec

This chapter intends to familiarize the reader with two selected scenarios that were used to implement the chosen security modeling notations. The following sections present a detailed description of each scenario with some potential security threats. The security requirements that can be derived are also discussed. It is important to note that the area of importance in this thesis is not threat modeling, henceforth no methodology has been followed for the same.

6.1 Over The Air - Software Download (OTA-SWDL)

Over-The-Air (OTA) is a means of communication that is getting very popular in the automotive industry of the latest technology. It is a standard employed to transmit and receive information using a wireless connection. [34] It is selected by the automotive industry to facilitate software updates to ECUs without the need for a physical connection. What is first required is to store these updates before transmitting them to the vehicles. For that, the Original Equipment Manufacturer (OEM)s maintain a software repository that holds all software update files. Whenever there is a need for an update, the vehicle requests the update from the repository and the OEMs push it to the vehicle. The updates are sourced from the repository to the vehicle through a cloud - which is maintained by the OEM as well. For better understanding of the scenario refer Figure 6.1. To establish and maintain the wireless connections, like telematics unit or infotainment unit, the vehicle is equipped with appropriate connected ECUs on-board.

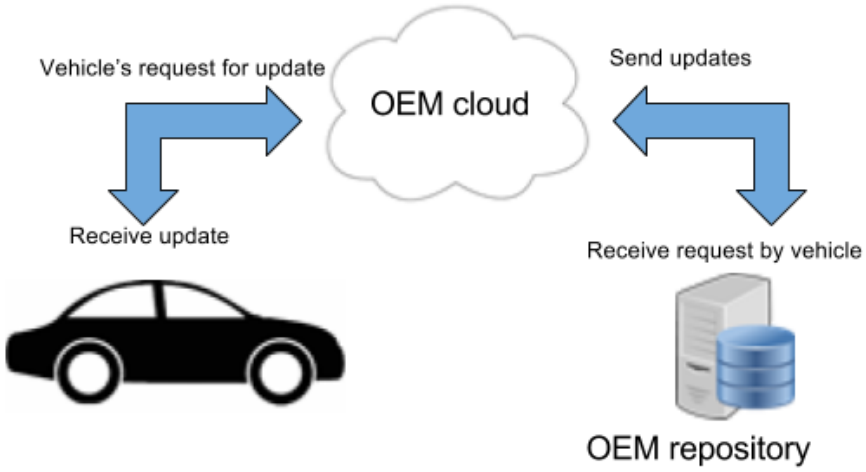

Figure 6.1: Vehicle updates through the Cloud

6.1.1 Threats identified in the scenario

As previously explained, OTA-SWDL is a wireless connection. Such a connection faces threats attempting to breach it. The following threats are identified in the context of the automotive industry taking into consideration the purpose of use of the protocol.

- Read software update files: An attacker could get access to the contents of a software update file causing theft of intellectual property. This means the attacker can use the information or possibly sell it to competitors.
- Deny software update: An attack could deny software update to a vehicle when there exists one. This would prevent the vehicle from functioning properly if the particular update was a software patch.
- Provide old updates: Another possible threat could be that the ECU is provided with a previous update or an update with known vulnerabilities instead of the latest update. This would result in denial of proper functionality to the vehicle.
- Modify software update: If an attacker is successful in overwriting the update file with his own malicious one, it would mean taking control of the vehicle and affecting its performance.
- Delete content of Software update: An attacker could simply erase the contents of an update file. A requesting vehicle can see an available update but receives an empty software update file causing it to enter an erroneous state.

6.1.2 Security Requirements

From the above threats, the security requirements that are required for the modeling implementation are presented here. The requirements address six different security concerns, which are: Confidentiality, Integrity, Authentication, Authorization, Auditability and Freshness.

R1 The vehicle shall authenticate that the updates originate from a reliable source.

R2 The OEM software repository shall be accessible only to authorized personnel.

R3 The content of the software update shall be encrypted to maintain confidentiality.

R4 The content of the software update shall be signed to ensure data integrity.

R5 A security log shall be maintained to enable auditing of update activities in the vehicle.

R6 A security log shall be maintained to audit activities of employees handling software updates in the OEM repository.

R7 The content of software update files shall fulfill freshness property.

6.2 Remote Vehicle Data Collection (RVDC)

The idea behind collecting diagnostic data from cars and using it to enhance user experience as well as vehicle performance is based on an offline tool called RVDC. Whenever a design team wants to know how often their function is used or how well it is performing, a measurement assignment (request for diagnostic data) is created and sent to the vehicle. The measurement assignment is executed whenever conditions are met and the result of data collection is uploaded to the OEM cloud when possible. Before data collection starts the user is asked for consent to participate in the process. This is an important aspect of RVDC as the output data could contain private information depending on the nature of requested diagnostic data. The collected data is then used by analysts to conclude results and decide strategies for improvement. The scenario is depicted in Figure 6.2.

6.2.1 Threats identified in the scenario

Although the RVDC scenario aims simply on serving the customer in a more efficient manner, it can still be vulnerable and target of attacks. This connection is also wireless as OTA-SWDL. Threats that are identified in this context are listed below:

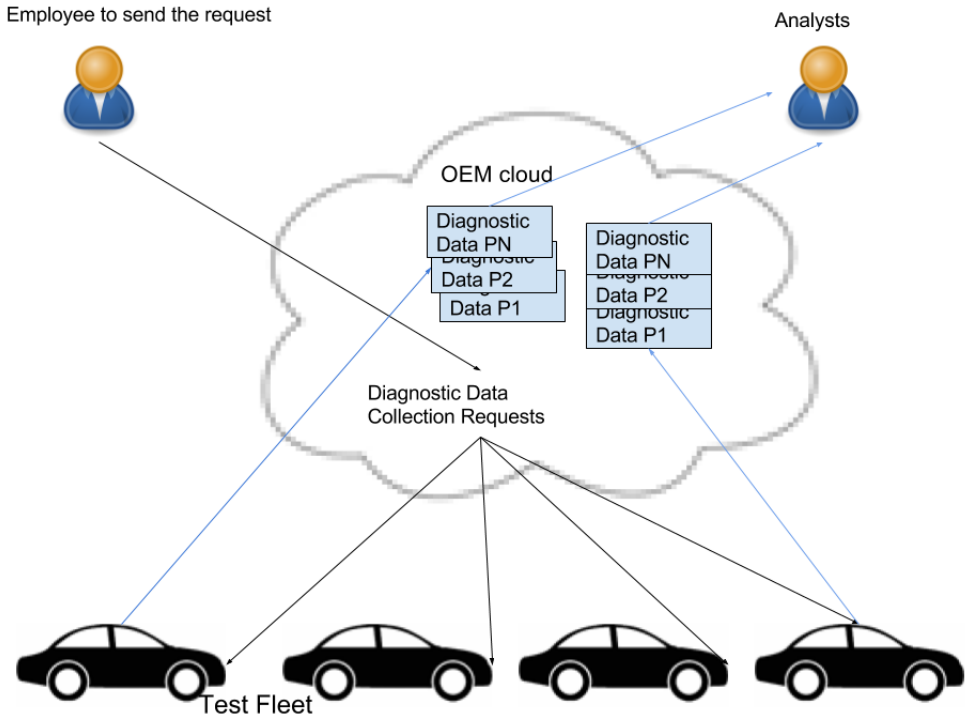

Figure 6.2: Diagnostic Data Collection

- Data collection against consent: Data might be retrieved by the organization, without the driver's permission.
- Read the content of diagnostic data: An attacker can access the content of the diagnostic data. This information can be used to cause harm to a specific driver.
- Modify the diagnostic data collection request: An attacker can have access to the request sent to the vehicle and modify it. As a consequence, the attacker can either contaminate the request or change its content - what is requested to receive.
- Modify the diagnostic data: An attacker could get access to the diagnostic data sent by the vehicle and modify its content. The organization in such a case will not serve the client as best as possible which could lead to disappointed customer.
- Request sent by unauthorized personnel: A diagnostic data request could be sent by unauthorized personnel. This might aim on retrieving sensitive information about the places the targeted vehicle has visited in order to cause harm to the driver.

6.2.2 Security Requirements

From the threats listed above, security requirements needed for the modeling implementation are extracted. The requirements address six different security concerns, which are: Confidentiality, Integrity, Authentication, Authorization, Privacy and Auditability.

R1 The user-driver shall provide consent to the received diagnostic date collection request for data collection.

R2 The vehicle shall authenticate that the received diagnostic date collection request is transmitted from a reliable source.

R3 The vehicle shall send encrypted data as a response to the data collection request.

R4 The content of the diagnostic data collection request shall be signed to ensure data integrity.

R5 A security log shall be maintained to audit the activities of the employees handling the diagnostic date collection requests.

R6 Only authorized personnel shall be able to send diagnostic data collection request to a vehicle.

7

Comparative analysis of UMLsec and SysML-Sec

The comparison in this section is based on the evaluation criteria described in Section 3.2.6. There are two levels of comparison, one to understand whether the security notations are easy to use and second to evaluate their expressive capability.

7.1 Ease of use (1)

To analyze the ease with which a notation can be adapted for use in the industry three main factors are considered. Firstly, documentation available to support learnability. The existence of documentation was considered important as it offers the support required by a non familiar user to comprehend the semantics of a notation and use it correctly. Secondly, flexibility of the notation with respect to the number of diagrams that can be annotated using its symbols. The more types of diagrams that can be annotated, the more cases a stakeholder can model. Thirdly, dependencies between the symbols in a notation. This is considered to affect usability of a notation because dependencies could constrain or guide a user.

Documentation to support learnability (1.a)

4 papers were referenced in order to get a clear picture of SysML-sec, [2], [33] [30], and [3]. SysML-sec proved challenging to learn as the *Pragmas* used to model Authenticity and Confidentiality were not clearly explained in these resources.

For UMLsec a book [16] was found and used in combination with 5 papers [17], [18], [7], [35] and [19]. The UMLsec profile comprises of many stereotypes and tags that express various security concerns. The challenge here was to comprehend the correct use of these stereotypes and tags, as a lot of them exist in the notation and each one can be used with specific types of diagrams.

Range of Diagrams that can be annotated using the notation (1.b)

SysML-sec proved to be limited when it came to labeling security at design level, which was the focus of this study. SysML-sec used only block diagrams for representing Authentication and Confidentiality which was achieved by using *Pragmas* or *CryptoBlocks*.

On the contrary, UMLsec supports a wider range of diagrams on which security stereotypes and tags can be used. More specifically, [16] states that the profile concerns all of UML such as Deployment Diagrams, Activity Diagrams, Component Diagrams, Class Diagrams, Sequence Diagrams and Use Case Diagrams.

Dependencies or constraints that guide the use of symbols in the notation (1.c)

There could be constraints on the semantics of a notation that could guide or restrain the use of certain symbols. Such dependencies that were discovered during implementation of the case study are described here.

In UMLsec, each stereotype has associated stereotypes and tags that have to be used together to achieve complete representation of a security property. Some examples are as follows:

- The stereotype *«secure links»* have to be used along with associated stereotypes *«encrypted», «LAN» and «wire»* to express the need for a secure connection between nodes. According to [16], it is essential that at-most one of the latter stereotype appears on a communication link in-order to make the use of the former stereotype valid. During security analysis, the combined effect of these stereotypes along with the type of *adversary* help to determine whether the connection is secure. Figure 7.1 shows how the OTA-SWDL components are deployed securely.

- Another stereotype is *«secure dependency»* which is used to denote consistency of security requirements on data being transferred between components. The semantics of this symbol follows that an object I in subsystem B contains a tag, if and only if the object I holds the same stereotype in subsystem A.Here, object I is communicated from subsystem A to subsystem B [16]. The stereotypes that can be used are *secrecy*, *integrity* and *high*. During analysis, it is possible to extract scenarios where security requirements on data are violated only if the above mentioned stereotypes and tags are used in combination. The tags on their own do not impart the same meaning. Figure 7.2 should be examined in combination with Figure 7.3 to fully understand this stereotype.

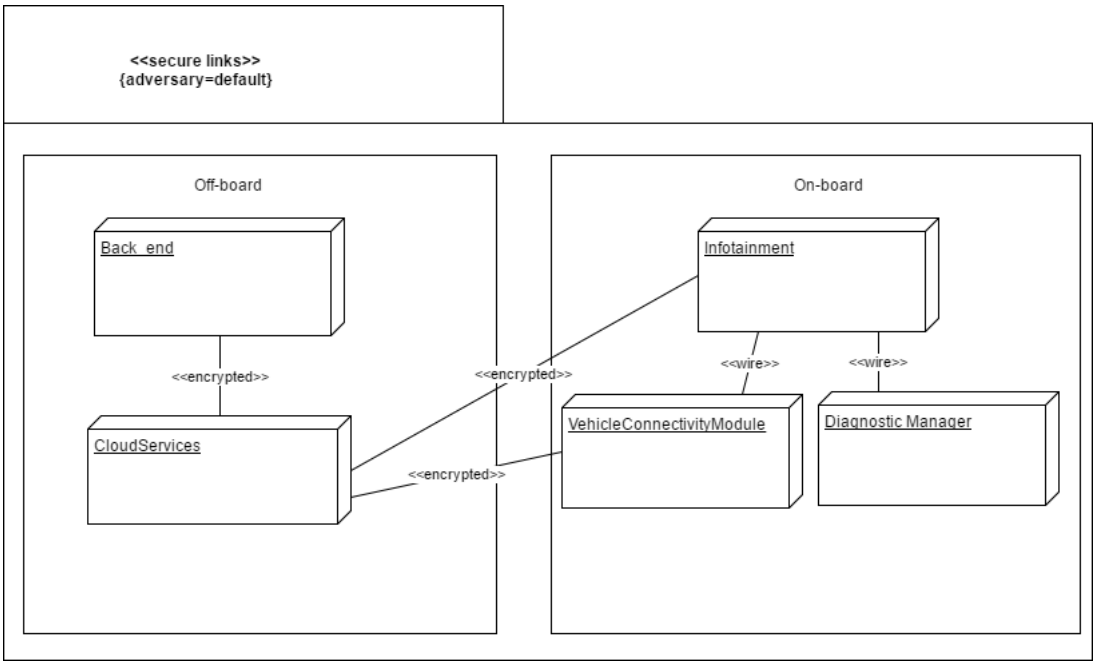

Figure 7.1: UMLsec Deployment diagram showing the type of connections between different components in OTA-SWDL

- The stereotype *«data security»* is justified only when the associated object is labeled with stereotype *«critical»* along with the necessary tags *secrecy*, *integrity*, *high*, *authenticity* and *freshness*. An example of this can be seen in Figure 7.3.

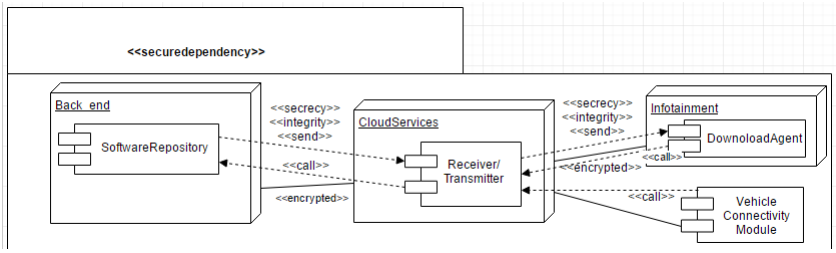

Figure 7.2: Component diagram with UMLsec showing which security concerns the dependencies should satisfy

In case of SysML-sec, each *pragma* should have a corresponding state diagram. The automated analysis of whether a security property is supported by design is possible by the tool only if the two model elements are linked together. Figures 7.6 and 7.7 show the respective state diagrams as they were modeled according to the block diagram of Figure 7.5 for OTA-SWDL.

7.2 Expressive Capability of Security Notations (2)

In order to accomplish this task it was necessary to explicate the level to which a security requirement is expressed by the notation. It was decided to examine whether a notation can illustrate only declarative security properties or provide further granularity by illustrating operational security properties as well.

Coverage of Security Concerns (2.a)

The initial focus for comparison was whether every requirement from both scenarios (discussed in Subsections 6.1.2 and 6.2.2) could be addressed using the security notations. The following two Tables 7.1 and 7.2 provide the extend to which UMLsec and SysMLsec can address the defined set of declarative security requirements.

Requirements	UMLsec	SysMLsec
R1	Yes	Yes
R2	Yes	No
R3	Yes	Yes
R4	Yes	No
R5	No	No
R6	No	No
R7	Yes	No

Table 7.1: Comparison: Coverage of Security Requirements of OTA-SWDL

Requirements	UMLsec	SysMLsec
R1	No	No
R2	Yes	Yes
R3	Yes	Yes
R4	Yes	No
R5	No	No
R6	Yes	No

Table 7.2: Comparison: Coverage of Security Requirements of RVDC

Based on this, an early conclusion can be made that UMLsec has greater coverage compared to SysMLsec. It was also decided to investigate whether UMLsec and SysML-Sec could support representation of operational mechanisms to achieve the above mentioned declarative properties. The results of this analysis are as shown below:

- **Confidentiality:** The tag *«secrecy»* associated with stereotype *«critical»* is used to express which data should be kept confidential. Confidentiality can

be achieved by cryptographic methods like encryption, however, UMLsec does not provide stereotypes to show this concept. As can be seen from Figure 7.3 the OTA SoftwareUpdateFile should be kept confidential by encryption, however, UMLsec is incapable of annotating this design decision.

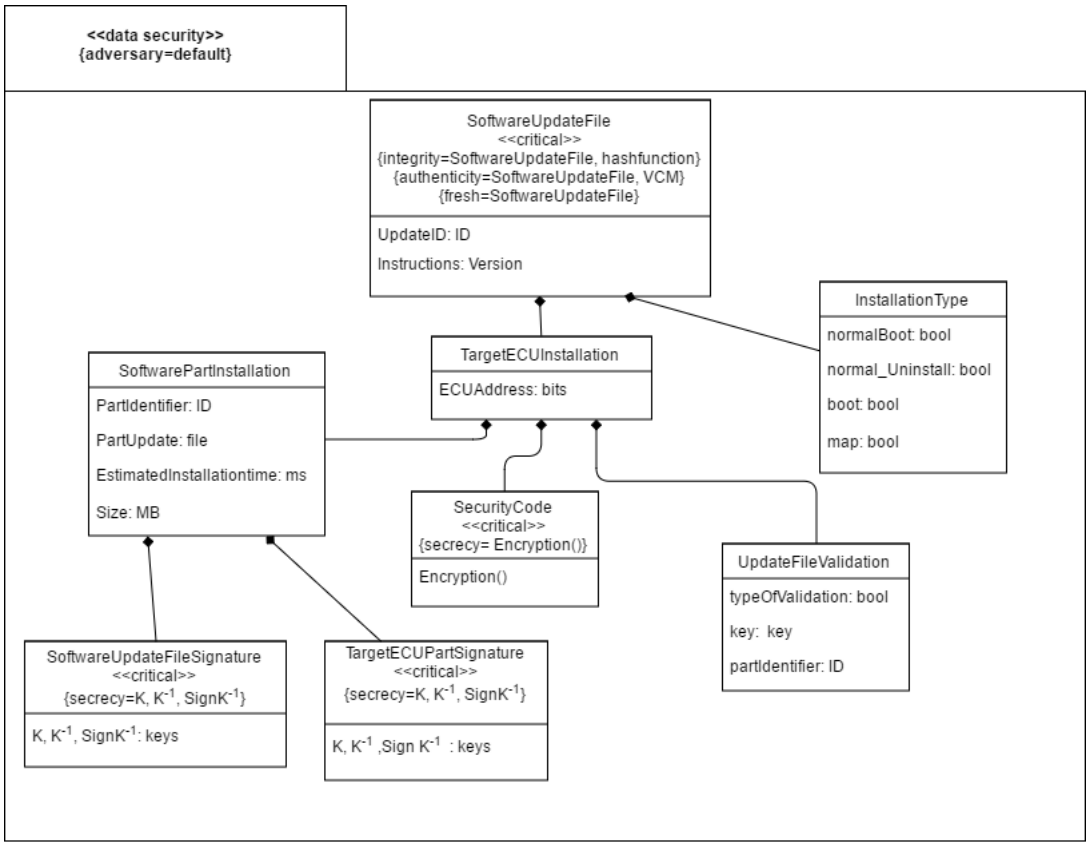

Figure 7.3: Class diagram with UMLsec capturing the structure of SoftwareUpdateFile

SysML-Sec has a different approach to represent confidentiality. The data that should be encrypted is represented as a *cryptoblock*, as in Figure 7.4, with attributes like *Message aencrypt(Message msg, Key k), Message adecrypt(Message msg, Key k), Message cert(Message msg, Key k), bool verifyCert (Message cert, Key k)*. The notation uses two concepts *Model Pragma* and *Property Pragma* that are indicative of global constraints at system level labelled as *#InitialSystemKnowledge* and constraints on data that should be kept secret labelled as *#Confidentiality* respectively. An example of how *Pragmas* are represented can be seen in Figure 7.5.

- **Integrity:** In UMLsec, the tag *«integrity»* associated with stereotype *«critical»* is used to denote integrity requirements on data. The attribute pair *(variable,expression)* related to the tag depicts the variable whose integrity should be preserved from an adversary and the range of expressions acceptable for that variable. This tag can be found in Figure 7.3. It does not show operational mechanisms to achieve integrity like hash functions or cryptography.

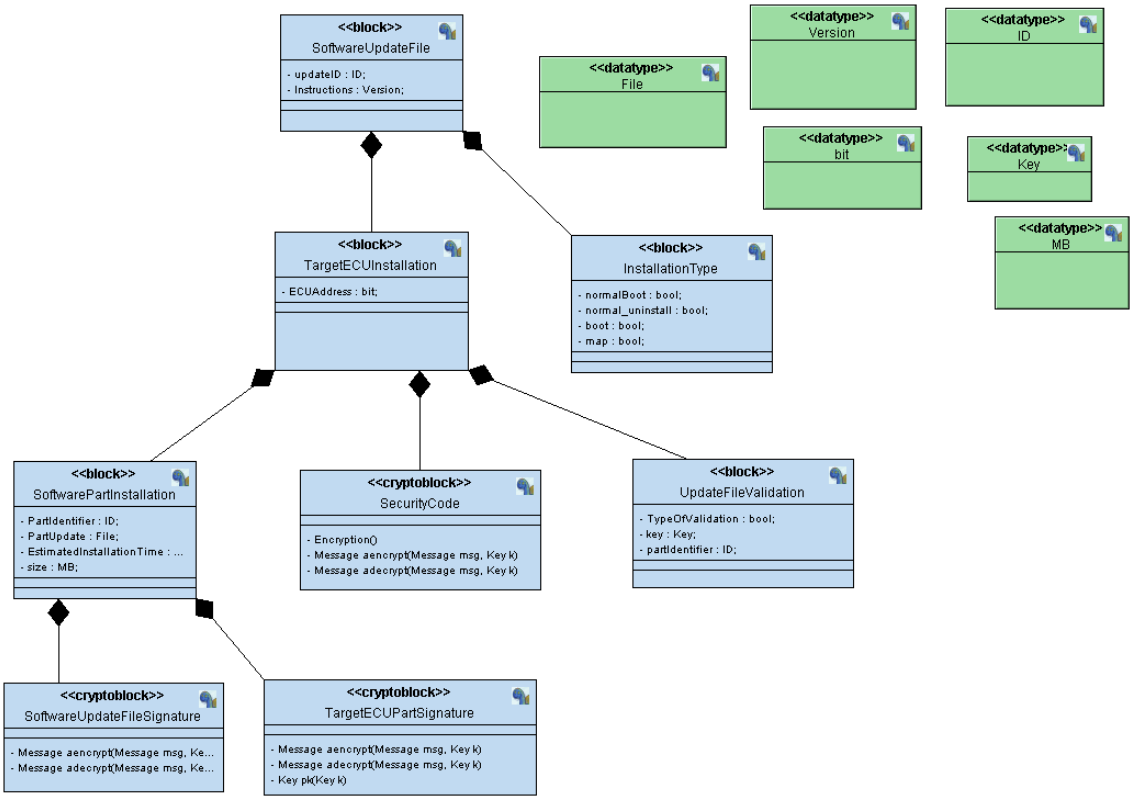

Figure 7.4: Blocks used as classes to demonstrate the structure of SoftwareUpdateFile

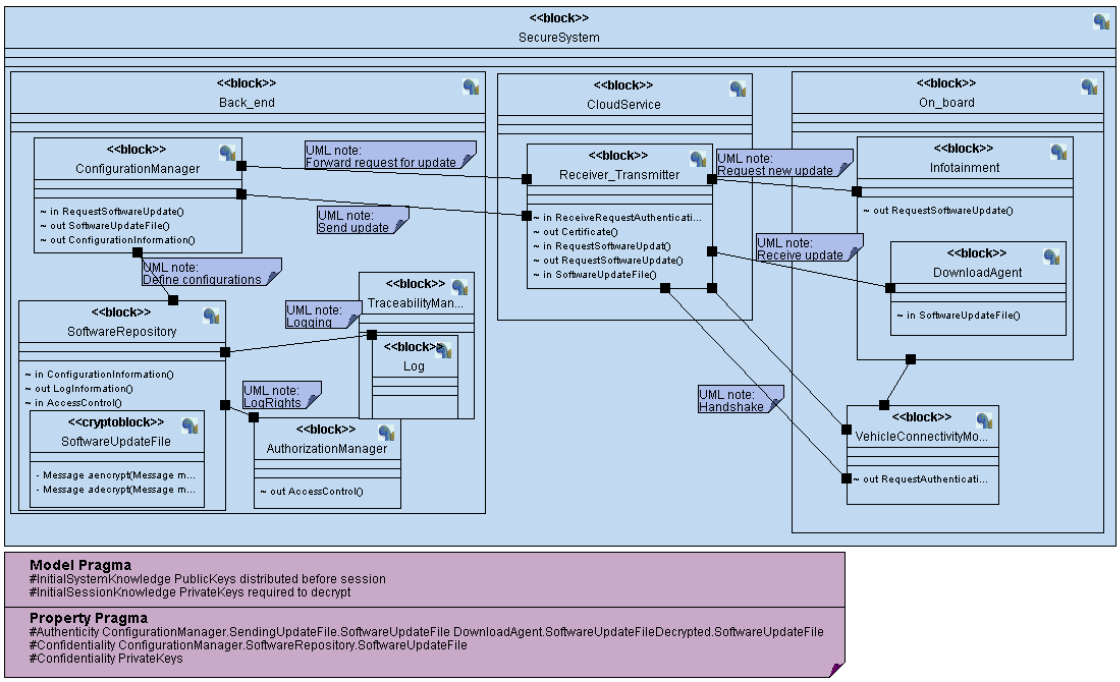

Figure 7.5: Block diargam capturing the backend (VSC), cloud and on-board (vehicle)

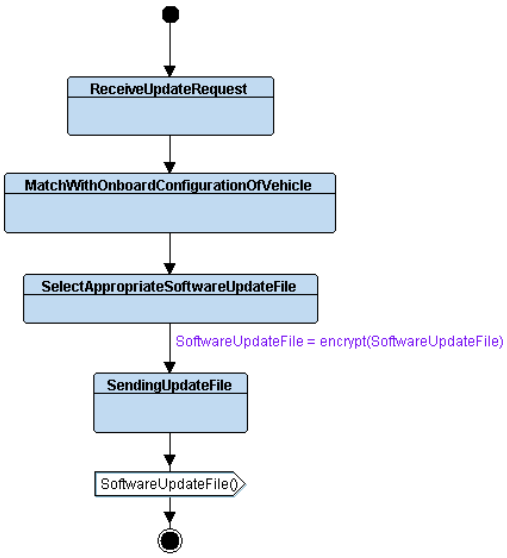

Figure 7.6: State diagram with SysMLsec showing the sender (ConfigurationManager)

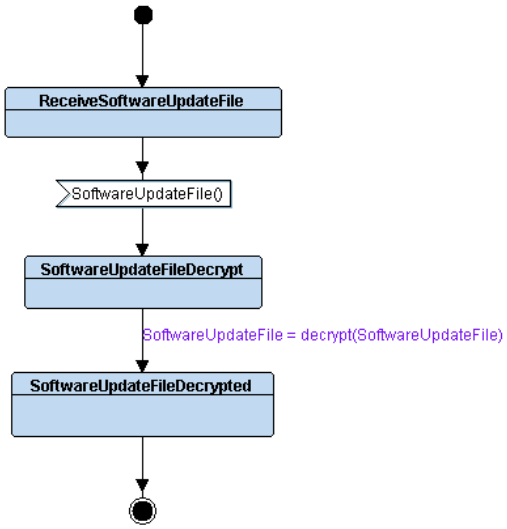

Figure 7.7: State diagram with SysMLsec showing the receiver (DownloadAgent)

- **Authentication:** UMLsec offers a tag *«authenticity»* that can be attached to an object or subsystem labelled with stereotype *«critical»*, as shown in Figure 7.3. The use of this tag along with its attribute pair *(data, origin)* is representative of data authenticity (that the *data* was sent from *origin*). Operational mechanisms that assist to achieve data authenticity, for example signatures, MACs (Message Authentication Codes) are not depicted by the notation.

  SysML-Sec uses *Property Pragma #Authenticity* to label an authentication requirement. The pragma takes as argument the source and destination of data that should be authenticated.This notation too does not specify mechanisms that help to achieve the property. In Figure 7.5 the use of *Property Pragmas* is demonstrated for OTA-SWDL.

- **Authorization:** UMLsec provides stereotype *«rbac»* to represent role based access control. The associated tags *«roles»*, *«right»* and *«protected»* are used to denote the different roles of an actor, the activities they are allowed to perform and the resources that are kept restrained respectively. Thus, UMLsec allocates a stereotype to show how an authorization policy (RBAC) is implemented on an operational level. Figure 7.8 shows in OTA-SWDL the different authorized actions for every role.

- **Freshness:** The tag *«fresh»* associated with stereotype *«critical»* is used to show that the data should not be outdated. It is of the type *data*.Operational mechanisms to accomplish freshness are not labelled by UMLsec. The use of

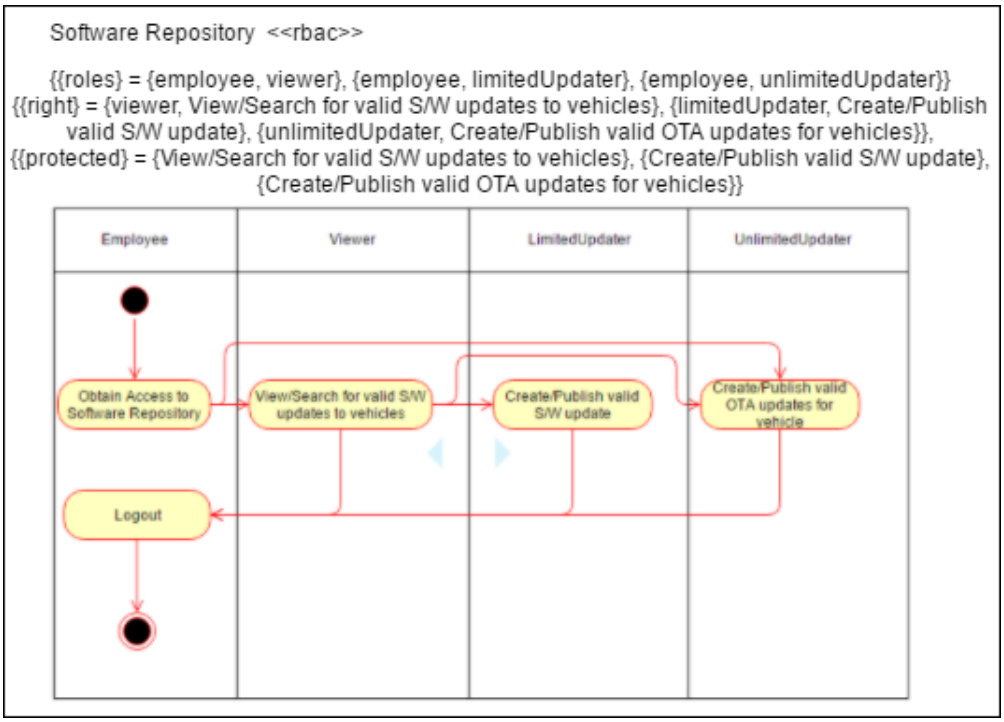

Figure 7.8: Activity diagram with UMLsec capturing the Authorization levels

the tag is shown in Figure 7.3 where it ensures that *SoftwareUpdateFile* should not be outdated.

- **Logging:** There are no stereotypes in UMLsec or SysML-Sec to label activities that should be accounted for.

  There is a stereotype *«provable»* with tags *(action, cert, adversary)* that can be used to denote a non-repudiation requirement. The tag *action* denotes a non-deniable action for which *cert* is an expression that is proof for the action to have happened. In the Figure 7.9 the stereotype *«provable»* is used to prove that a log entry is created and stored every time the *SoftwareRepository* component is accessed by an employee.

- **Privacy:** There are no stereotypes available in UMLsec and SysML-Sec to represent privacy or the ways to achieve it through appropriate design.

Along with the capability to annotate declarative and operational security properties, a broader scope of analysis was chosen to observe whether the notations can address other security related information.To conduct this comparison, the following concepts were taken into consideration:

- **Security of Physical Infrastructure:** The type of link that is used to connect subsystems/components, the support for security provided by these links and how the notations choose to represent them are observed.

  In case of UMLsec, the level of security expected from a communication link

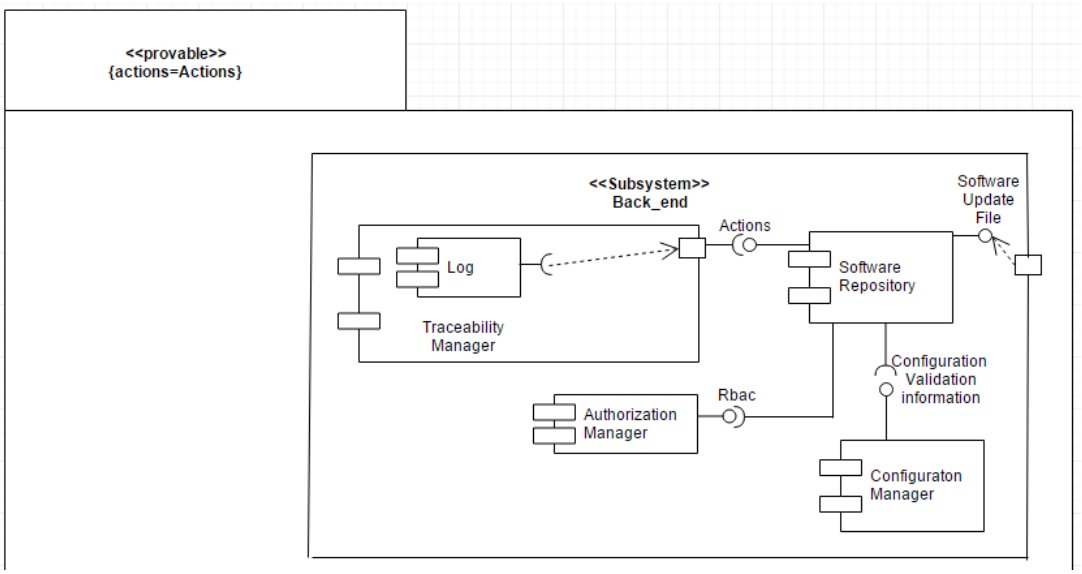

Figure 7.9: UMLsec diagram to demonstrate how Logging could be modeled with the existing stereotypes

between two nodes is illustrated by the stereotype *«secure links»*. UMLsec offers associated stereotypes namely *«internet»*, *«encrpted»*, *«LAN»* and *«wire»* to further define the type of connection that can exist between two interacting subsystems/components. Using a combination of the above stereotypes, it is possible to show that the security requirements on the communication link are met by the physical situation between nodes.

In case of SysML-Sec, there are two ways to represent communication link between two nodes namely *"public"* and *"private"*, where *private* means a link that cannot be listened to.

- **Security of Data:** The notations are inspected with respect to what they can offer to label data that should be kept secure.

  UMLsec offers the stereotype *«data security»* combined with *«critical»* and its associated tags which are *«adversary»*, *«secrecy»*, *«integrity»*, *«authenticity»*, *«high»* and *«fresh»*. The 4 tags (excluding *«adversary»*), are able to show that the data has the required security level against an adversary of specified type.

  On the other hand, SysMLsec does not offer the ability for such high precision. With the use of Property Pragmas, SysMLsec can address only two security requirements for data confidentiality and authenticity.

- **Labelling of Assets:** How each security notation represents different parts of the system that are considered as assets. It is noted that neither UMLsec nor SysML-Sec provides an explicit stereotype to label assets. Instead, the existing stereotypes are considered to be illustrative of data,components and links that should be secure.

UMLsec has the capability to annotate assets on physical and data levels using stereotypes *«secure links»*, *«data security»* and *«critical»*.

In SysML-Sec, an asset is denoted by a *"Cryptoblock"*. This stereotype can be used to denote an object, class, component, subsystem etc.On physical level, SysML-Sec denotes an asset using *"private"*.

- **Capabilities of Adversaries:** The ability of each notation to consider different types of adversaries, their capabilities and impact on the system.

  UMLsec has imported the notion of adversary as a tag denoted by *«adversary»* wherever required. The *«adversary»* tag identifies different types of actors who can pose threats to the system. *Insider* and *default* are the two generic types of *«adversary»* identified by UMLsec. In our approach the adversary is considered as *«adversary»=default* assuming that no insiders (employees of an automotive industry) will threaten the system.

  On the other hand, SysMLsec does not have the ability to illustrate different types of adversaries for a system/subsystem.

Does it convey the intended information? (2.b)

The exercise involving stakeholders yielded results that were useful to gauge the understandability of the chosen notations. In general, the feedback obtained were quite similar to the actual semantics, however there were some suggestions put forth by the subjects. They are discussed below:

In SysML-Sec, the communication link between two components are not labelled when it is private. It would be easier to understand the nature of this link if it was annotated explicitly, instead of having to click on the connection to find out. In cases where the diagram would be printed onto PowerPoint presentations or PDFs it is not possible for a viewer to visualize the type of link. Also, the fact that there is no precise definition for a *private* channel could be confusing for a user.

It is very hard to find security related information from SysML-Sec diagrams because they are not distinct from rest of the model elements. For example, the *cryptoblock* which was used to denote an encrypted file was unnoticed by both subjects until prompted.

It would be better if security properties like Confidentiality and Authenticity were labelled on the diagram itself rather than having them attached separately in the form of *pragmas*. Tracing the information presented in a *pragma* to elements in the diagram is tedious.

Overall, UMLsec diagrams were well understood. One strong point being the fact that it is possible to annotate the type of *adversary*. Security of physical

infrastructure is also more refined and clearly shows encrypted/internet/wire links. Models representing data security and role based access control were appreciated as they convey precise information.

How obvious is the role of a symbol used in the notation? (2.c)

The stereotypes mentioned on the packages such as *«secure links»*, *«secure dependency»* and *«data security»* were rarely noticed. The subjects could comprehend the meaning of a diagram by reading the tags associated with these stereotypes alone. However, there was some ambiguity regarding the meaning of UMLsec stereotype *«secure dependency»* and the tags associated with it.

The subjects also had some nice-to-have recommendations for example, the possibility to explicitly label assets and attach CIA rating for each depending upon which security property was essential to be satisfied for that particular asset.

To recapitulate, the comparative analysis discussed above indicates that UMLsec is better documented and described than SysML-sec and can be combined with a wider range of diagrams. Concerning dependencies or constraints to guide the use of symbols or stereotypes, both notations were found to have such associations among stereotypes/tags (for UMLsec) and symbols/diagrams (for SysML-sec). Furthermore, it was observed that UMLsec addressed 5 (Confidentiality, Integrity, Authentication, Authorization, Freshness) out of 7 security concerns while SysML-sec addressed 2 (Confidentiality, Authentication) out 7. It should be mentioned that in UMLsec, with the use of a certain stereotype, Logging could be modeled despite the lack of a corresponding stereotype. UMLsec was concluded to be more accurate as well as descriptive when compared to SysML-sec in regards with Physical Infrastructure, Security of Data, Labelling of Assets and Capabilities of Adversaries. Finally, the understandability exercise indicated that more information can be elicited when employing the UMLsec notation and that certain stereotypes of UMLsec in addition to the cryptoblocks of SysML-sec were rarely noticed.

8

Revisiting the research questions

This chapter focuses on answering the research questions and discussion on which of the two selected notations is an appropriate choice. Considering that the comparative case study was conducted in one corporation, it is important to acknowledge the existence of threats to validity in the answers . In order to minimize these threats, generalization was kept strictly in alignment with the criteria and requirements that were emphasized as important by experts in the organization.

RQ1:Criteria to adopt a security modeling notation

For answering research question RQ1, the Figure 4.1 and Figure 4.2 in combination with the comparison discussed in Chapter 7 were taken into consideration. Since the early meetings held with VCC employees, it became apparent that existing tool support is among the most significant criteria. As companies shift towards MBSE or Model-Driven Engineering (MDE), reliable tool support impacts their effectiveness and productivity. In addition, tools currently being used in an organization also influence the choice of a security modeling notation by considering which notations they support.

The number of Security concerns that are covered is also a requirement for companies while selecting a security modeling notation. As shown in Table 7.1 and Table 7.2, not all security concerns can be addressed. In addition, during the understandability exercise, there existed comments indicating its importance.

Another criterion would be whether the notation can be supported by the currently employed modeling techniques. Taking advantage of the already existing knowledge base is what led to this criterion. The importance of maintaining a relatively small learning curve may allow the intended users to invest more time in applying the security notation and advancing the projects.

To identify the necessary requirements for a security framework to fit a company Section 7.1 and Section 7.2 should be considered. Based on these, a security framework should be able to address the security concerns required by a company. Therefore, the framework should define symbols/stereotypes for every security con-

cern. Moreover, detailed documentation and support for annotating a wide range of diagrams result in a powerful security framework. Finally, a security framework should verify that security properties are modeled in a system.

Considering what is discussed so far in this Chapter, it could be said that not many existing techniques could match a company's criteria. As the suggestion of an existing technique for a company matches the suggested notation for VCC, the answer for SQ1.1 is given later in this Chapter.

RQ2:Maturity level of security modeling techniques and existing gap at VCC

Through the literature review, a significant number of security modeling techniques was found. Although a lot of effort has been put on investigating and designing new notations, detailed study led to the conclusion that most of them are not yet mature enough. Either tool support or adequate documentation are missing as well as not all security concerns can be addressed. Regarding VCC, most of the design level models lacked proper representation of security properties. In cases where security concerns are modeled, the notation employed did not have an academic background supporting verification of the models. Even though UML was widely used by architects and designers, no UML based security notation had been investigated for use. An explanation of maturity regarding modeling and tools used at VCC is discussed in Section 2.4.3 which answers RQ2.Clearly, there existed a gap to application at VCC and answers to the research questions assisted in bridging the gap.

Taking into account the results of RQ1 and RQ2, it is now possible to suggest a bit fitting notation for VCC that will address the existing gap. From RQ1, UMLsec is a clear winner in terms of criteria satisfied for adopting a notation. From the list of 14 criteria described in Figure 4.2, UMLsec scores 50% (7 out of 14). The expectations from a security notation for adaptability (corresponding to subquestion 1.1)is discussed in Chapter 7. Section 7.1 shows the precedence of UMLsec over SysML-sec with respect to ease of use. UMLsec has a book to facilitate learnability and can be used on a wider range of diagrams.Literature also claims that UMLsec has mature tool support [6].However, since the tool is not open source it was not feasible to use it for the thesis work. Section 7.2 reflects that UMLsec allows expression of a variety of security related information. Alongside being able to label 5 out of 7 security requirements, the notation also address security of physical infrastructure, two types of adversaries and their capabilities. Considering the maturity of modeling and tools used at VCC (corresponding to RQ2), it is considered easier for VCC to invest in learning this UML based notation. The reason for VCC to investigate a security notation is to enable documentation and analysis of security concerns. UMLsec is a good notation considering its intended use in the concerned company. Due the fact that UMLsec is capable of representing 5 security properties and supports analysis it is an appropriate choice.

The downside of UMLsec would be that the notation has semantic constraints in the form of dependencies between symbols discussed in section 7.1.It would take some effort for the designer to understand proper placement of stereotypes and tags. But this initial investment would payoff as these dependencies guide proper representation of the intended security property.

It should also be hinted that UMLsec was still not able to fully support all the requirements of Table 7.1 and Table 7.2. According to the findings of this project, it is important for researchers to identify what security concerns the industry needs to address. It would also be significant for the existing notations to improve not only the notation but also supportive documentation in order to assist candidate users to understand it. An observation that was made while using UMLsec was that the notation allows traceability for certain security concerns through dependency stereotypes. This could be an indication of what a notation can support and extend to more security concerns.

The motivation behind this thesis is to investigate to what extend the security notations currently existing in academia can be employed in automotive industry. After studying the existing security notations, a comparative case study was conducted at VCC to identify which security notation fits the organization. To introduce minimal threats, the case study was implemented using two scenarios given by VCC. The comparison aimed to explore the characteristics of the two selected notations that affect their ease of use and expressive capability. Comparing the two notations for both OTA-SWDL and RVDC, it was concluded that between UMLsec and SysMLsec, the former proved to be more powerful notation.

References

[1] S. Aljawarneha, A. Alawnehb, and R. Jaradat, "Cloud security engineering: Early stages of SDLC", in *Future Generation Computer Systems*, Oct. 2016.

[2] L. Apvrille, L. Li, and Y. Roudier, "Model-Driven Engineering for Designing Safe and Secure Embedded Systems", in *2016 Architecture-Centric Virtual Integration (ACVI)*, Apr. 2016.

[3] ——, "Model-driven engineering for designing safe and secure embedded systems", in *Architecture-Centric Virtual Integration (ACVI), 2016*, Apr. 2016.

[4] D. Basin, M. Clavel, J. Doser, and M. Egea, "Automated analysis of security-design models", in *Information and Software Technology*, vol. 51(5), May 2009, pp. 815–831.

[5] D. Basin, J. Doser, and T. Lodderstedt, "Model driven security: From UML models to access control infrastructures", in *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 15(1), Jan. 2006, pp. 39–91.

[6] A. van den Berghe, R. Scandariato, K. Yskout, and W. Joosen, "Design notations for secure software: A systematic literature review", in *Software & Systems Modeling*, 2014, pp. 1–23.

[7] B. Best, J. Jurjens, and B. Nuseibeh, "Model-based security engineering of distributed information systems using UMLsec", in *Proceedings of the 29th International Conference on Software Engineering*, 2007, pp. 581–590.

[8] M. J. M. Chowdhury, "Security risk modelling using SecureUML", in *16th International Conference-Computer and Information Technology*, Mar. 2014.

[9] T. Green, "Cognitive dimensions of notations", in *People and Computers V*, 1989, pp. 443–460.

[10] M. Hafner, M. Breu, R. Breu, and A. Nowak, "Modelling inter-organizational workflow security in a peer-to-peer environment", in *Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, 2005.

[11] M. Hafner and R. Breu, "Sectino - A motivating case study from E-Government", in *Security Engineering for Service-Oriented Architectures*. 2009, pp. 65–70.

[12] B. Hoisl, S. Sobernig, and M. Strembeck, "Modeling and enforcing secure object flows in process-driven SOAs: An integrated model-driven approach", in *Software & Systems Modeling*, 2014, pp. 513–548.

[13] T. Hoppe and J. Dittmann, "Sniffing/replay attacks on can buses: A simulated attack on the electric window lift classified using an adapted cert taxonomy", in *2nd Workshop on Embedded Systems Security (WESS)*, 2007.

[14] T. Hoppe, S. Kiltz, and J. Dittmann, "Automotive it security as a challenge: Basic attacks from the black box perspective on the example of privacy threats", in *Computer Safety, Reliability, and Security*, 2009, pp. 145–158.

[15] ——, "Security threats to automotive can networks—practical examples and selected short-term countermeasures", *Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures*, vol. 96, pp. 11–25, Jan. 2008.

[16] J. Jurjens, *Secure systems development with uml*. Springer-Verlag Berlin Heidelberg, 2005.

[17] ——, "Towards development of secure systems using UMLsec", in *Fundamental Approaches to Software Engineering*, 2001, pp. 187–200.

[18] ——, "Using UMLsec and goal trees for secure systems development", in *Proceedings of the 2002 ACM symposium on Applied computing*, 2002, pp. 1026–1030.

[19] J. Jurjens, L. Marchal, M. Ochoa, and H. Schmidt, "Incremental security verification for evolving UMLsec models", in *ECMFA 2011, LNCS 6698*, 2011, pp. 52–68.

[20] N. Z. Khidzir, A. Mohamed, and N. H. H. Arshad, "Information assets security requirement: The relationship between confidentiality and availability of information assets in ICT outsourcing", in *IEEE Colloquium on Humanities, Science and Engineering (CHUSER)*, Dec. 2012.

[21] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile", in *2010 IEEE Symposium Security and Privacy*, 2010, pp. 447–462.

[22] J. Ludewig, "Models in software engineering – an introduction", in *Software and Systems Modeling*, Mar. 2003, pp. 5–14.

[23] R. Matulevicius and M. Dumas, "A comparison of SecureUML and UMLsec for role-based access control", in *Databases and Information Systems*, 2010, pp. 171–185.

[24] M. Menzel and C. Meinel, "A security meta-model for service-oriented architectures", in *IEEE International Conference on Services Computing*, 2009, pp. 251–259.

[25] ——, "SecureSOA - Modelling security requirements for service-oriented architectures", in *IEEE International Conference on Services Computing*, 2010, pp. 146–153.

[26] (). Model Based Engineering, Software Engineering Institute and Carnegie Mellon University, [Online]. Available: `http://www.sei.cmu.edu/architecture/research/model-based-engineering/`.

[27] Y. Nakamura, M. Tatsubori, T. Imamura, and K. Ono, "Model-driven security based on a web services security architecture", in *IEEE International Conference on Services Computing*, 2005.

[28] P. H. Nguyen, M. Kramer, J. Klein, and Y. L. Traon, "An extensive systematic review on the Model-Driven Developmentof secure systems", *Information and Software Technology*, vol. 68, pp. 62–81, Dec. 2015.

[29] D. K. Nilsson and U. E. Larson, "Simulated attacks on can buses: Vehicle virus", in *5th lASTED Int. Conf. on Communication Systems and Networks*, Sep. 2008, pp. 66–72.

[30] G. Pedroza, L. Apvrille, and D. Knorreck, "AVATAR: A SysML Environment for the Formal Verification of Safety and Security Properties", in *11th Annual International Conference on New Technologies of Distributed Systems (NOTERE)*, May 2011.

[31] G. Rathee and H. Saini, "Security concerns with open research issues of present computer network", in *International Journal of Computer Science and Information Security*, Apr. 2016.

[32] T. V. Roermund. (Jan. 2016). Security and privacy standards are critical to the success of connected cars, [Online]. Available: `https://techcrunch.com/2016/01/28/security-and-privacy-standards-are-critical-to-the-success-of-connected-cars/`.

[33] Y. Roudier and L. Apvrille, "SysML-Sec: A model driven approach for designing safe and secure systems", in *2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, 2015.

[34] M. Rouse. (May 2007). Over the Air (OTA), [Online]. Available: `http://searchmobilecomputing.techtarget.com/definition/Over-the-Air`.

[35] T. Ruhroth and J. Jurjens, "Supporting security assurance in the context of evolution: Modular modeling and analysis with UMlsec", in *IEEE 14th International Symposium on High-Assurance Systems Engineering*, 2012, pp. 177–184.

[36] F. Satoh, Y. Nakamura, and K. Ono, "Adding authentication to model driven security", in *IEEE International Conference on Web Services*, 2006.

[37] "Survey on security threats and protection mechanisms in embedded automotive networks", in *Dependable Systems and Networks Workshop (DSM-W)*, Jun. 2013, pp. 1–12.

[38] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "Engineering security into distributed systems: A survey of methodologies", *Journal of Universal Computer Science*, vol. 18, pp. 2920–3006, 20 Dec. 2012.