ECAI 2015 - International Conference – 7th Edition
Electronics, Computers and Artificial Intelligence
25 June -27 June, 2015, Bucharest, ROMÂNIA

HAZOP-based Security Analysis for Embedded Systems: Case Study of Open Source Immobilizer Protocol Stack

Jingxuan Wei, Yutaka Matsubara, Hiroaki Takada
Graduate School of Information Science
Nagoya University
Nagoya, Japan
{jx_wei, yutaka, hiro}@ertl.jp

Abstract – **Nowadays, with the introduction of network connectivity both inside and outside modern vehicles, researchers have identified that the system is actually fragile if an attacker could locate any security vulnerabilities of the system. Although security analysis techniques prospered in the industry, still a general, compatible, and effective one remains uncertain. This paper aims to transplant the safety analysis technique HAZard and OPerability studies (HAZOP) into an appropriate security analysis technique. By conducting a case study of security analysis for Open Source Immobilizer Protocol Stack, we demonstrate the usability of the proposed technique and discusses results of the analysis.**

Keywords- Safety Analysis, Security Analysis, HAZOP, Open Source Immobilizer Protocol Stack

I. INTRODUCTION

In recent years, integration of Electronic Control Units (ECUs) in all kinds of machinery has brought us great convenience as well as a remarkable increase of production and operation efficiency. Automobiles, for example, have taken great advantages of ECUs, such as instructing drivers to drive safely and comfortably, as well as assisting technicians to conduct proper diagnosis when the vehicle is under maintenance. These digital components oversee a broad range of functionalities such as the drive train, brakes, lighting and entertainments, etc. In fact, few operations are not controlled by the computers embedded in a modern vehicle. It is said that a modern luxury automobile contains up to 70 different ECUs within millions of lines of software code [1]. In order to cooperate all the ECUs, embedded system software plays a significant role. From the beginning of the vehicle's design, no matter it is speeding on the road, providing entertainment contents to passengers, or even communicating with outer network to upgrade the software, the embedded system never be absent and is always running in the background across the entire life cycle of an automobile.

The complexity of safety-critical embedded systems such as areophane and automotive control systems has increased rapidly. Until recently, such kinds of industry have mainly focused on the safety design of an embedded system. In fact, although several international standards for functional safety such as IEC 61508 [2] and ISO 26262 [3] have been published, security of the safety-critical systems is not mentioned in such standards. The guarantee of every functionality's proper running is the primary task all the way from the initial design, even to the maintenance during the vehicle's service. Techniques have been introduced to designers to prevent every piece of equipment in the vehicle from fatal malfunction, in order to protect passengers' lives. Failure Mode and Effect Analysis (FMEA) and Fault Tree Analysis (FTA) are two general techniques that could help analyzers to consider about many of the safety concerns thoroughly and efficiently. However, with the debut of network connection availability for embedded systems, highly digitalized modern vehicles are exposed to information attacks via all kinds of unauthorized, either wired or wireless connection. It is not good enough anymore for the vehicle to just function fine as initially designed, it is also required that the vehicle possesses the ability to defend itself from attacks coming from outside connection. Security vulnerabilities have been detected among modern vehicles, if intentionally exploited by attackers, making the vehicle itself as well as passengers and their properties inside the vehicle exposed in danger [4].

Conducting a safety analysis only eliminates concerns that without people's intention to deliberately act as villains. While considering the intentionality, despite the assurance that vehicle will be working properly by itself, there could also be a security concern that someone may be exploiting certain security vulnerability to deliberately launch attacks toward the vehicle and causing the vehicle to be compromised eventually. To address such problem, a more systematic approach of security analysis is often needed.

This paper presents an analysis technique to help embedded system designers conducting security analysis at the phase of system design. During the analysis, threats should be eliminated as much as possible by implementing related countermeasures. Based on the fundamental idea from HAZOP, this paper refines a new security analysis technique by changing the original guidewords into 8 actions extracted from the attack taxonomy of the taxonomy Computer Emergency Response Team (CERT) [5]. This HAZOP-based security analysis technique uses these actions as the new guidewords to examine a given system architecture and uncover the security vulnerabilities from the design.

During the security analysis, applying the Guidewords to the given system architecture will help us to find out unconsidered deviations (unexpected

functionality, unwanted connection, etc.) of the system. To consider the causes and the consequences of certain deviation, this paper discusses the deviation's local effects (affecting system's normal functionality) as well as the global effects (keeping the user from properly using the system). On summarizing all the analysis result entries into one overall table, this paper also issues a severity value to each of the result entry in order to conduct a risk evaluation and argue about whether or not further precautions should be implemented during the system design. To visualize the given system architecture, in this paper, we use a sequence diagram to assist analyzing the security concerns.

To demonstrate the applicability of the proposed technique, this paper describes a simple case study of Open Source Immobilizer Protocol Stack (OSIPS) developed by Atmel® [6]. In the case study, this paper analyzes security vulnerabilities within the OSIPS, to generate a comprehensive and accurate analysis results. By comparing the final analysis results to detected security vulnerabilities [7] of OSIPS, this paper discusses effectiveness and applicability of the proposed security analysis technique.

This paper makes the following contributions:

- A new HAZOP-based security analysis technique
- An analysis sheet containing threats and hazards found in given system architecture
- A case study for the OSIPS

This paper is organized as follows: in Section 2, we introduce some related works including the safety analysis techniques as well as some security analysis techniques. In Section 3, we present a detailed security analysis flow and the proposed security analysis technique to find threats in the system design phase. In Section 4, we describe the results of case study of security analysis for the OSIPS. In Section 5, we discuss effectiveness and applicability of the proposed method. Finally, in Section 6, we conclude this paper.

II. RELATED WORKS

A. Safety Analysis Techniques

Before talking about security analysis, the fact could not be ignored that for many years safety analysis techniques have been able to keep the machinery from malfunction, preventing humans from getting injured or even fatal casualties. Three of main safety analysis techniques used in the industry are listed as,

FMEA, treats each component that makes up the system as the analysis object. Starting from these components, safety analysis experts are focused to list up all the failure modes that may occur in each component of the system. Then, experts will consider about a chain of all possible effects led by the particular failure mode, and at last evaluate the severity of those effects that will eventually cause accidents to either the system, or the user.

FTA, is a technique that firstly takes each ultimate fault of the whole system as the root of a tree, and then, to grow leaves on such tree, starts to consider all the causes that will ultimately lead to such fault. Presenting a tree with a root node of system failure or hazard and leaf-nodes of all kinds of faults of the system that lead to such system failure [8].

HAZOP, takes up a whole new viewpoint neither from the reasons causing any system failure nor the supposed consequences of any system faults, but takes the execution flow as the analysis object during a simulation of the system's running. Mostly used for building chemical plants, in order to uncover deviations as much as possible, a team of experts performing HAZOP technique will systematically consider each process unit in the plant, such as pipelines, tanks, or reactors, and each hazard one by one, raising queries about the design. Based on the description of the chemical plant design, experts will use the guidewords such as NO(NOT, NONE), MORE, LESS, AS WELL AS, PART OF, REVERSE and OTHER THAN to examine every variables of interest such as flow in a pipeline, temperature of a reactor, pressure of a tank, level of composition, or time of a reaction. Each guideword will be applied to every single line in the design one by one, to examine if there would be any deviations away from the original design intentions [9]. However, the guidewords are not suitable to analyze the software architecture or data flow.

B. Security Analysis Techniques

Now that, security problems have become a general concern when developing an embedded system, and certain techniques assisting system designers to perform a security analysis have also been introduced to the industry. Such as,

Attack tree. This technique can be used for representing attacks against a system. The nodes in the tree represent the different possible steps in the attack, with the top-node as the goal of the attack. All nodes below the top-node can be assigned quantitative or qualitative values. If the former type of values are assigned, it is possible to do various calculations of the top-node as well. A textual list structure is recommended if the attack trees become too complex [10]. However, such analysis technique relies heavily on the related experience of the analyzer, and may not be a suitable choice for people who is still not quite familiar to security concerns.

STPA-Sec. System-Theoretic Process Analysis for Security, is a top-down, system engineering technique modified from safety analysis technique STPA. STPA-Sec identifies security vulnerabilities and requirements as well as scenarios leading to violation of security constraints. Then uses the results to refine system concept and makes it to be more secure [11]. The hazard analysis process in STPA-Sec consists of five phases, which are a) Determining unacceptable losses. b) Creating a model of the high level control structure-HLCS. c) Identifying unsafe/unsecure control actions. d) Developing security requirements and constraints. e) Identifying casual scenarios [10]. By conducting this 5 phases, STPA-Sec can address technical and organizational issues and supports a security-driven concept development process where vulnerability

analysis influences and shapes early design decisions, also iterated and refined as concept evolves [11].

SafSec, in which, Saf is for Safety and Sec is for Security. As displayed in the name, this is a method of managing both safety and security risks in a system development project, especially those of advanced avionics architectures (AAvA) or integrated modular avionics (IMA). Consisting of Guidance Material [12] and Standard [13], the SafSec methodology can be used to ensure that the assurance provided through safety and security certification is met efficiently with minimum rework to enable IMA to be realized and cost benefit provided [14]. However, such method focused mainly on the analysis of modularized avionics production, and the application of automobile embedded system software still remains unclear.

Despite the prosperity of security analysis techniques, a general, compatible, and effective technique for analyzing security vulnerabilities in automobile embedded system still remains uncertain in the industry. This paper tries to redirect the safety analysis approach of HAZOP, to the security analysis for designing automobile embedded systems. Now, based on the fundamental idea of applying guidewords from HAZOP, this paper takes efforts to transplant this safety analysis technique into a security analysis technique.

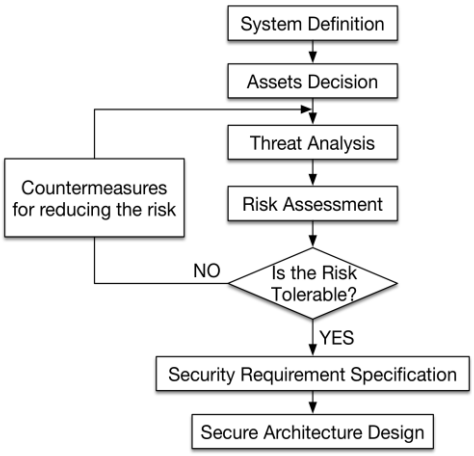
III. Security Analysis

A. Security Analysis Flow in System Development Phase

It is often considered impossible to develop a perfect embedded system with no security flaws. In fact, even if such system does exist, it would probably be high in cost but low in efficiency. However, there is a practical engineering approach suggesting to predefine the value of every asset that needs protection, and decide which asset to protect according to their values. If an attacker would have to pay much more expensive efforts but only to gain little, he would probably give up on launching such attack. Thus the asset is considered as relatively secured.

Process of developing embedded system software is much more like the development of a consumer software product. During the phase of product definition, engineers discuss the concept of operations to clarify the system's goal: what to do with this system. On top of the decision of the system concept, engineers sketch out system requirements and architecture to further discuss the system functionality. And it is this phase during which a security analysis should be conducted in order to eliminate or mitigate potential security treats as much as possible. To examine such issue, Figure 1 shows a practical security analysis flow. Starting from the choosing which asset to protect, security analysis will begin and further argue about threats that will cause the assets to be compromised. The fact that certain assets have been compromised will be considered as a risk, and will be assessed by experts to evaluate how hazardous this risk is going to be. At this point, qualitative calculation for a severity value as well as the occurrence rate will be issued for the evaluation, in order to argue about whether this risk is

Figure 1. Security Analysis Flow



tolerable. If such risk is tolerable and could be ignored during the design, then it is safe to establish the security requirement specification concerning the current asset and continue the design. On the other hand, if such risk is not tolerable, and is considered as hazardous enough to cause serious accidents in the future affecting user's life or properties, then it is necessary to consider practical countermeasures, which also need to be at an acceptable cost, for minimizing the risk. In order to make sure that whether the countermeasures are coming into force or not, threat analysis and risk assessment will be performed again until risks found during the analysis are finally tolerable.

B. Threat Analysis

1) Objectives: Threat analysis is conducted to locate potential security concerns in the system. Environmental threats, for example, a sudden power failure will cause the system to black out and stop functioning, or an unexpected struck by lightning causing a short inside the system. On the other hand, artificial threats are brought by humans with or without intentionality. Intentional artificial threats are brought by people who are deliberately exploiting system's vulnerability for malicious gain, such as eavesdropping on the communication, or falsification of the data during the transportation in order to send fake information. However, non-intentional artificial threats are due to negligence during the system operation, such as a misoperation or malfunction. These are all considered as potential threats in the system, and precautions should be implemented into the design before the system's roll out. So, here we just list up a few of all the threats that may be concealed in the system design at an early stage, and here's is the question: how can we guarantee that the conducted threat analysis is exhaustive and comprehensive? Can we assure that all the potential threats have been pointed out and that the whole system is free from security concerns?

It may be difficult to guarantee a system that has no security flaws, however, with proper security analysis technique applied, we are still able to sweep out the threats as much as possible. And when most of the security concerns have been controlled at a minimum

level, making the effort to attack such system to be at an extremely expensive level, then, it is suitable to say that the system now is under a relatively safe environment.

2) Target Diagram: Diagrams that visualize the design and correctly explain all the functionality as well as data flows are necessary to perform a proper security analysis. One of the popular diagram choices is Unified Modelling Language (UML). The standard UML diagrams are: class diagram, object diagram, use case diagram, sequence diagram, state chart diagram, activity diagram, collaboration diagram, component diagram, and deployment diagram. In this paper, we use sequence diagrams to depict sending and receiving messages among devices.

3) Guidewords: A modified attack taxonomy in [5] provides actions related to security attacks. As shown in Table I, they are PROBE, SCAN, FLOOD, AUTHENTICATE, SPOOF, BYPASS, MODIFY, and READ. The detail definitions of the actions are found in [15]. We employ the actions as guidewords for threat analysis in the system architecture and data flow, and derive deviations from the requirements and designs of the system by applying the guidewords.

We divided the guidewords into 2 groups, Primary Guidewords and Secondary Guidewords. Primary Guidewords include actions to acquire information about the target:

- Probe
- Scan
- Read.

And secondary Guidewords include actions to utilize the information acquired by Primary Guidewords and execute high-level attacks:

- Flood
- Authenticate
- Spoof
- Modify
- Bypass.

A special case for Primary Guidewords is that, attacker with concrete knowledge or former experience of the target could skip attempting the actions in Primary Guidewords to investigate the target and commence attack right away with actions in Secondary Guidewords. Therefore, when conducting a detailed analysis on the message level to examine the data flow, first of all, each guideword from 2 groups will be applied to the messages to conduct the analysis. Then one guideword from Primary Guidewords and another guideword from Secondary Guidewords will be combined to apply to the messages and begin the analysis one more time.

TABLE I. GUIDEWORDS FOR DETAILED SYSTEM LEVEL ANALYSIS, WITH THEIR MEANINGS.

Guideword	Meaning
PROBE	access a target in order to determine its characteristics
SCAN	Access a set of targets sequentially in order to identify which targets have a specific charateristic
FLOOD	Aacess a target repeatedly in order to overload the targets capacity
AUTHENT-ICATE	Present an identity of someone to a process and, if required, verify that identity, in order to access a target
SPOOF	Masquerade by assuming the appearance of a different entity in network communications
BYPASS	Avoid a process by using an alternative technique to access a target
MODIFY	Chage the content or characteristics of a target
READ	Obtain the content of data in a storage device, or other data medium

4) Analysis Sheet: With the application of each guideword (or combination of two guidewords), entries of analysis result will be generated one by one. To summary all the entries as a single table, this paper creates an analysis sheet as shown in Table II. Each guideword will be applied to the data flows in turn and generates deviations leading to local effects and global effects. And a severity value will also be issued according to the after-defined severity table to the global effects. At last, there will also be a list containing all possible attacks that may eventually cause the effects. The combination of 2 guidewords from different group will also be applied to the analysis objects. Starting from 8 cells that represent the status (whether or not has been applied as the guideword. •as yes, and - as no) of the guideword up ahead. Here as shown in the sample is an entry of applying the combination of 2 guidewords, which are Read and Flood.

C. Risk Assessment

A proper risk evaluation should be conducted by considering the combination of threat severity, the threat occurrence probability, as well as the threat success probability. With the given formula (1),

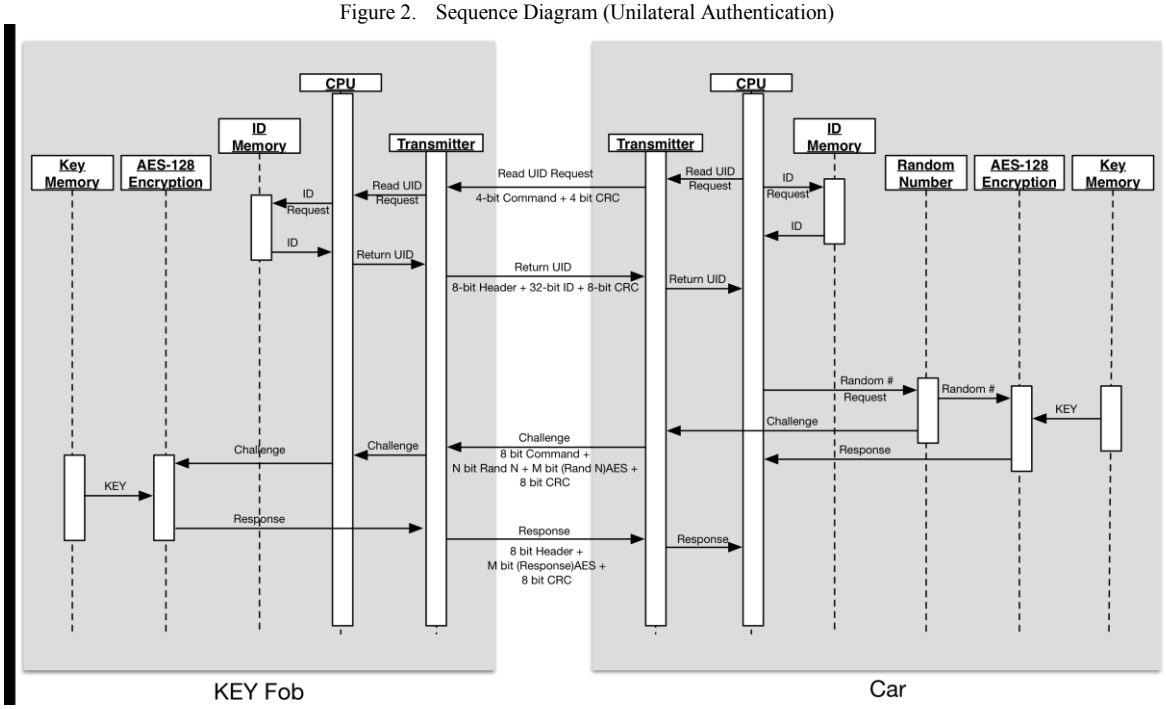
$$\text{risk} = \text{severity} \times \text{threat\_occurrence\_probability} \times \text{threat\_success\_probability} \quad (1)$$

a risk value should be easily calculated to help identifying whether such risk is tolerable or not. For example, a risk value larger than 5 suggests that the risk

TABLE II. GUIDEWORDS FOR DETAILED SYSTEM LEVEL ANALYSIS, WITH THEIR MEANINGS.

Primary Guideword			Secondary Guideword					Deviation	Local Effect	Global Effect	Possible Attacks
Probe	Scan	Read	Flood	Authenticate	Spoof	Modify	Bypass				
-	-	•	•	-	-	-	-				

Figure 2. Sequence Diagram (Unilateral Authentication)



is non-tolerable and proper countermeasures should be implemented to eliminate such risk; while a risk value less than 4 suggests that the risk is tolerable and could be ignored during the execution of the system.

IV. A CASE STUDY

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. Open Source Immobilizer Protocol Stack(OSIPS)

The OSIPS is developed by Atmel®. This stack is intended to be used in conjunction with the automotive transponders chips of Atmel®, but also in general, could be deployed in any other compatible transponders chips from other manufacturers as well. Specifications, designs and source-code of the OSIPS can be obtained as open-source system [6].

This stack comes with a variety of commands issued by the reader, and sent to the key fob. During a normal use of the immobilizer, the car will act as the reader sending commands to authenticate with the registered key fob. However, during a diagnostic or maintenance, the reader could also be used as a programming device for the car manufacturer or distributor. The communications between the car and the key fob are implemented at a low frequency of 125 kHz, which is a very limited range, practically just a few centimeters [7].

The command set out in the protocol stack's specification encompasses eleven commands. They include reading of the key fob's unique ID (UID) and error status, initiation of authentication, setting of the used secret keys, initiation and leaving of the so-called enhanced mode (for RF communication powered by the battery), a request to repeat the last response, reading and writing of user memory as well as setting memory access protection to certain memory sections. Authentication can be configured to be as: [7]

- Unilateral, only key fob authenticates itself to the reader.
- Bilateral, both key fob and reader authenticate themselves to each other.

This paper uses the unilateral authentication as the analysis object for a case study of the proposed HAZOP-based security analysis technique. The authentication follows a general pattern of exchanging challenge and response information between the communicating two devices. A challenge will be sent by the verifying party to the participant party often containing a randomly generated number. Then, on receiving the challenge, the participant encrypts it with a key that is previously shared with both parties, and sends it back as the response. The verifying party compares the response to its own encryption result with the same key, if matched, then authentication succeeds.

Figure 2 shows a conceptual diagram demonstrating the unilateral authentication process of the OSIPS. Base on the specification, this paper tries to analyze the security concerns from the messages level using the sequence diagram shown in Figure 2. Detailed data flows will be examined one by one using the pre-defined new guidewords, in order to perform a deeper security analysis.

As described in [6], the authentication is initiated by sending a Read UID command from the car. The key fob receives such command, and read the UID information from its own memory and send it back to

TABLE III. ANALYSIS SHEET

P*	Secondary Guideword					Deviation	Local Effect	Global Effect	Possible Attacks
R*	F*	A*	S*	M*	B*				
Read UID									
•	•	-	-	-	-	Flooding the KEY Fob with Read-UID-like request, which makes the KEY Fob unable to receive and deal with connections any more	The KEY Fob will not be able to send the UID information to the car.	Failure of the exchange of UID information between a registered KEY Fob and the car. The authentication will not be triggered regardless of the user's request.	Denial-Of-Service Attacks
•	-	-	•	-	-	ransponders can be used to relay the communications between the car and the key fob.	Without the genuine key fob being in the communication range, the car will be tricked to send a Read UID request to a transponder near the car.	Without user's intention, the key fob will receive a Read UID request through a transponder near the key fob. Person with the KEY Fob that is associated with a specific UID could be tracked down for their whereabouts.	Tracking
•	-	-	-	•	-	Falsification of data during the transportation.	A non-Read UID request will be sent to the key fob.	Unauthorized falsification will be ignored by the verification of CRC checksum.	Unauthorized Falsification
Return UID									
•	•	-	-	-	-	Flooding the car with Return-UID-like information, which makes the car unable to receive and deal with connections any more.	The car will not be able to receive the UID information from KEY Fob.	Failure of the exchange of UID information between a registered KEY Fob and the car. The authentication will not be triggered regardless of the user's request.	Denial-Of-Service Attacks
•	-	-	•	-	-	The car will receive UID information within the the Return UID from Unregistered key fob or unknown device.	By constantly sending Return UID until a challenge is received, the attacker may be able to acquire the UID information stored in the car.	With the UID information in hands, it just extends the possibility to launch all kinds of attack.	Relay Attack
•	-	-	-	•	-	Falsification of data during the transportation.	A non-Return UID request will be sent to the key fob.	Unauthorized falsification will be ignored by the verification of CRC checksum.	Unauthorized Falsification
Challenge									
•	•	-	-	-	-	Flooding the KEY Fob with challenge-like information, which makes the KEY Fob unable to receive and deal with connections any more.	The KEY Fob will not be able to receive the challenge information from the car.	Failure of the exchange of challenge information between a registered KEY Fob and the car. The authentication will fail regardless of the user's operation.	Denial-Of-Service Attacks
•	-	-	•	-	-	Attacker could pretend to be the car and send challenges to the key fob.	Challenge information recorded from eavesdropping on a genuine authentication will be sent to key fob.	Attacker can get himself/herself authenticated with the recorded challenge information.	Replay Attack
•	-	-	-	•	-	Falsification of data during the transportation.	A non-Challenge request will be sent to the key fob.	Unauthorized falsification will be ignored by the verification of CRC checksum.	Unauthorized Falsification
Response									
•	•	-	-	-	-	Flooding the car with response-like information, which makes the car unable to receive and deal with connections any more.	The car will not be able to receive the response information from KEY Fob.	Failure of the exchange of response information between a registered KEY Fob and the car. The authentication will fail regardless of the user's operation.	Denial-Of-Service Attacks
•	-	-	•	-	-	The car will receive a response from Unregistered key fobs or unknown devices.	The car will receive a fake response.	Without the genuine KEY to correctly encrypt the information, this fake response will be rejected at the car.	Tolerable
•	-	-	-	•	-	Falsification of data during the transportation.	A non-Response request will be sent to the key fob.	Unauthorized falsification will be ignored by the verification of CRC checksum.	Unauthorized Falsification

the car. The car will verify whether the UID is the same with the UID information in its own memory. If the UID matched, the car will start the authentication by sending a challenge mainly composed of a randomly generated number. The key fob received such challenge, read the encryption key from its own memory, and use it to encrypt the challenge as the response and sent it back. The car verify whether this response is same result as its own encryption result. If the result matched, then at last the authentication succeeds, and user may use the key fob to start operating the car.

TABLE IV. SEVERITY VALUE TABLE FOR RISK EVALUATION

ID	Global Effect	User Awareness	Evaluation			Severity
			Automobile	User's Wellbing	Privacy	
1	Failure of the exchange of UID information between a registered KEY Fob and the car. The authentication will not be triggered regardless of the user's request.	Users will not notice any unusual change until they try to operate the car.	8	0	0	8
2	Without user's intention, the key fob will receive a Read UID request through a transponder near the key fob. Person with the KEY Fob that is associated with a specific UID could be tracked down for their whereabouts.	No. All communications are implemented wirelessly.	6	0	10	8
3	Unauthorized falsification will be ignored by the verification of CRC checksum.	No. All communications are implemented wirelessly.	0	0	0	0
4	With the UID information in hands, it just extends the possibility to launch all kinds of attack.	No. All communications are implemented wirelessly.	6	0	0	6
5	Failure of the exchange of challenge information between a registered KEY Fob and the car. The authentication will fail regardless of the user's operation.	Users will not notice any unusual change until they try to operate the car.	8	0	0	8
6	Attacker can get himself/herself authenticated with the recorded challenge information.	No. All communications are implemented wirelessly.	10	0	8	9
7	Failure of the exchange of response information between a registered KEY Fob and the car. The authentication will fail regardless of the user's operation.	Users will not notice any unusual change until they try to operate the car.	8	0	0	8
8	Without the genuine KEY to correctly encrypt the information, this fake response will be rejected at the car.	No. All communications are implemented wirelessly.	4	0	0	4

B. Assumptions

Alongside with the communication between the key fob and the car, this sequence diagram also indicates the internal communication within each party. This paper ignores analysis on the internal communications due to the temporary lack of physical connection. However, this does not implicit that there is no need to conduct an analysis on them, in fact, if an attacker was able to gain access of internal devices such as through OBD-II port, USB, iPod Dock, etc., there will be even more possibilities for him/her to compromise the system. In this paper, we only perform security analysis on the external wireless communication between the key fob and the car, during which command like Read UID, or the exchange of information like Return UID, Challenge or Response are sent and received.

We assume that attackers are lack of knowledge about the OSIPS and all the subjective are attackers with malice. As we have mentioned, risk assessment is evaluated by the combination of threat severity, the threat occurrence probability, as well as the threat success probability. However, during the security analysis in this case study, it is still considered as difficult to calculate both probabilities of threat's occurrence and success. Therefore, in this paper, a given severity value is solely used to conduct a simplified version of risk evaluation.

C. Threat Analysis

As shown in Table IV, analysis results of applying the combination of Primary Guidewords and Secondary Guidewords are all summarized in the analysis sheet. For column size issue in Table. IV, some table headers are shortened as following: P*: Primary Guidewords; R*: Read, F*: Flood; A*: Authenticate; S*: Spoof; M*: Modify; B*: Bypass.

As the analysis results in the analysis sheet, deviations by the application of each guideword as well as the combination of Primary Guidewords and Secondary Guidewords are listed up to consider all the possible local effects and global effects. Comparing the possible attacks reported in [7], in which the security issues include (not limited to) as relay attack with genuine key fob, tracking, denial-of-service attacks, replay attack on authentication, spoofing attack on memory access protection, and hijacking communication sessions, we could find most of attacks. This results imply the effectiveness and applicability of the proposed method in the security analysis.

Applying all Primary Guidewords to messages lead to the same result due to same interpretation of Probe, Scan, and Read, therefore all analysis results relating to Primary Guidewords are summarized under a single Primary Guideword: Read. The guidewords such as Authenticate and Bypass were not applicable for applying to the analysis objects. For example, Read UID request does not contain any specifications or data flows about authentication between the car and the key fob; or that Read UID request is required to initiate the authentication between the car and the key fob, and cannot be bypassed. The same situation also appeared in conducting analysis on Return UID, Challenge and Response. Therefore, we could omit analysis related to Authenticate and Bypass.

D. Risk Assessment

As shown in Table IV, all global effects that have been listed up in Table III are now summarized together and evaluated about their severity one by one. Noted that there is not a specific rule established on how to decide the severity value of each global effect. We believed that every company or organization has their own rules, but this paper considers severity value

evaluation from 3 prospects: Automobile, Passenger's wellbeing and Privacy. And at last, a total severity value will be generated by calculating the average of the 3 non-zero prospects' value. Value larger than 5 will be considered as risk being non-tolerable and value less than 4 will be considered as risk being tolerable. In this case study, global effect with the ID of 1, 2, 4, 5, 6, 7 are considered as dangerous hazards and should be minimized or eliminated during the system design.

V. DISCUSSION

As explained at the previous section, we could confirm the effectiveness and applicability of our method. However, there are still several problems that need further consideration.

This combination of guidewords generates a new problem: what is the threshold of the combination? Two groups of guidewords have been combined to consider the deviations in this paper, however for example, it is not impossible to add back the Secondary Guidewords again into the combination, generating a combination of guidewords from 3 groups to commence such attacks step by step with the primary guidewords, the secondary guidewords, and the tertiary guidewords. Such as attacks in which an attacker Reads the data during the communication, Modifies them into nonsense packets and Floods them back to network; or an attacker firstly Bypasses the authentication mechanism, then fraudulently Authenticates himself/herself as a legal user, and Spoofs the whole network to communicate to or through him/her. Of course, 4 groups, 5 groups or even much more groups are also considerable for the security analysis. In fact, it is proper to say that the more groups combined the more accurate and comprehensive results can we get. We need to consider the combination threshold about where to stop as the future work.

Attacks such as session hijacking takes control of the whole system after the authentication is completed. Thus eavesdropping across the whole exchange of challenge and response information is required to succeed by such attack. Therefore, considering the relationship of the two directions such as Read UID and Return UID, also Challenge and Response, it may not be appropriate to deliberately separate them as two completely different objects to conduct the analysis. Integration of such related exchanges of messages should receive fair attention as well, and should be analyzed as a whole. Improvement such as raising the abstraction level of analysis object diagram may be one of the solutions to minimize the current drawback. Thus using not only the sequence diagram as the analysis object, but also introducing other kinds of diagrams to conduct an even comprehensive analysis on all kinds of different levels.

VI. CONCLUSION

This paper presented the HAZOP-based security analysis technique, with its new guidewords in order to perform exhaustive security analyses during the system designing. We used 8 actions words extracted from the attack taxonomy by CERT as the new guidewords and performed security analysis by applying them to embedded system designs. Although this HAZOP-based technique did succeed in locating some of the security flaws that may eventually cause an incident to the system, leading to the exposure of privacy, or loss of properties. As future works, we plan to discuss problems found in the case study, and develop a tool to support security analysis based on the proposed method.

REFERENCES

[1] R. Charette, "This Car Runs on Code", Available Online: http://www.spectrum.ieee.org/feb09/7649, Feb 2009.

[2] IEC, it IEC 615008, "Functional Safety of Electrical/electronic/programmable Electronic Safety-related Systems", Part 1-8, 2000.

[3] ISO, it ISO 26262, "Road vehicles – Functional safety –", Part 1-9, 2011.

[4] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Daany Anderson, Hovav Shacham, Stefan Savage, "Experimental Security Analysis of a Modern Automobile", IEEE Symposium on Security and Privacy, Oakland, CA, May 16-19 2010.

[5] R.R. Brooks, S. Sander, Juan Deng, and Joachim Taiber, "Automobile Security Concerns, Challenges and State of the Art of Automotive System Security", IEEE VehicularTechnology Magazine, June 2009.

[6] Atmel, "Open Source Immobilizer Protocol Stack", Available online: http://www.atmel.com/tools/opensourceimmobilizerprotocolstack.aspx, 2015.

[7] Stefan Tillich and Marcin Wójcik, "Security Analysis of an Open car Immobilizer Protocol Stack", 2012.

[8] David John Pumfrey, PhD., "The Principled Design of Computer System Safety Analyses", University of York, Department of Compurter Science, September 1999.

[9] Nancy G.Leveson, "Safeware: System Safety and Computers", Addison Wesley Professional, 1995.

[10] Christian Raspotnig, PhD., "Requirements for safe and secure information systems", Department of Information Science and Media Studies, University of Bergen, October 2014.

[11] William E. Young, Jr., PhD Candidate, "STPA-SEC for Cyber Security/Mission Assurance", Engineering Systems Division, Systems Engineering Research Lab, March 2014.

[12] Praxis High Integrity Systems, "SafSec: Integration of Safety Security Certification, SafSec Methodology: Guidance Material", 2006.

[13] Praxis High Integrity Systems, "SafSec: Integration of Safety Security Certification, SafSec Methodology: Standard", 2006.

[14] "SafSec: Integration of Safety Security Certification", Available Online: http://intelligent-systems.altran.com/en/technologies/security/safsec.html, 2015.

[15] J. D. Howard an T. A. Longstaff, "A Common Language for Computer Security Incidents", Sandian Nat. Lab., Sandia Rep. SAND98-8867, 1998.