ISSN (Print) : 0974-6846
**Indian Journal of Science and Technology,** *Vol 8(27), DOI: 10.17485/ijst/2015/v8i27/81018, October 2015* ISSN (Online) : 0974-5645

# Proposal and Verification of Function-based Security Protocol for Vehicular Communication System

**Kun-Hee Han[1] and Woo-Sik Bae[2*]**

[1]Department of Information Communication Engineering, Baekseok University, Cheonan Chungnam, Korea; hankh@bu.ac.kr

[2]Department of AIS Center, Ajou Motor College, Boryeong Chungnam, Korea; drbws@daum.net

## Abstract

Not only M2M (Machine-to-Machine) communication but also IoT (Internet of Things) technology has emerged against the backdrop of evolving ICT industry, drawing much attention in all fields including auto industry. Being first adopted in such fields as factories, logistics, meteorology, environment, defense and agriculture and stockbreeding, M2M systems enable devices to communicate automatically with one another and to operate in response to varying conditions. As for vehicles, M2M technology is applied to interactions between internal and external vehicular devices, between vehicles, between vehicles and traffic systems and between vehicles and peripheral devices. Yet, wireless communication systems are prone to hacking attacks in transmission sections. Thus, any attacks on vehicles' brakes, multiple control systems and engine control parts will put passengers and safety at risk. In this context, many researchers explore the security measures for inter-device communication. This paper designed a protocol for safe communication between vehicular devices using hash function and complex mathematical formulae, and tested it with Casper/FDR, a formal verification tool for protocols. The proposed protocol proved to guard against diverse attacks and to be effectively applicable in practice.

**Keywords:** Authentication Protocol, Internet of things, Location privacy, Model Checking

## 1. Introduction

In addition to the concept of device-to-device and machine-to-machine communication as well as virtual interactions with information, IoT (Internet of Things) incorporates general communication where all products support embedded systems including electronic appliances and smartphones. Particularly, as inter-device automatic communication without human intervention, M2M (Machine-to-Machine) communication is widely investigated[1,2]. M2M technology is extensively applied across industries that hardly allow manual inspection and operation, e.g. military surveillance, agriculture and stockbreeding, meteorological observation, safety, environment, vehicles and factory automation[3,4]. Regarding vehicles, research is undertaken on automatic operation without human intervention in vehicle-to-vehicle, vehicle-to-device, vehicle-to-thing and device-to-device communication. Still, the wireless section for inter-device communication is potentially vulnerable to hacking attacks from intruders[5]. In case hackers tamper with the communication system between things and vehicles, they can control brakes, steering systems and engines, threatening people's lives. Hence, meeting security requirements is a significant issue, and many researchers inquire into the security in wireless communication sections. Mostly, however, previous studies were focused on theorem proving, ending up with complications unforeseen in the design process. Some protocols turned out to be incompatible with actual systems or manifest security glitches, which warrants in-depth studies[5]. As a consequence, ISO 26262, an international standard for functional safety of vehicles has recently been developed. Still, more efforts need be made regarding the threats to inter-device security[6].

This paper used Casper/FDR[7,8], widely used for formal verification, to experimentally test the proposed protocol and prove its safety in wireless communication system. This paper consists of the following chapters. Chapter 2 reviews relevant literature on threats to vehicular security

* *Author for correspondence*

and CASPER/FDR. Chapter 3 proposes an authentication protocol and describes its operation before experimentally testing it with Casper/FDR. Chapter 4 verifies the safety of the protocol. Finally, Chapter 5 presents the conclusion.

## 2. Literature Review

### 2.1 Literature Review

The following are the prerequisites for security in wireless communication[9,10].

- Authentication and integrity: Devices need be identified as authorized players. To that end, each device should have a unique ID.
- Confidentiality: Data transmitted and received between authenticated devices in vehicular communication should remain strictly confidential and protected against unauthenticated devices.
- Anonymity and privacy: A lack of anonymity in vehicular communication leads to an infringement on privacy. In case vehicular information is known to an intruder, serious problems including safety issues may occur.
- Non-repudiation: Non-repudiation technology should be applied so that any device of origin cannot deny the fact that it has sent certain data.2.2 Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

### 2.2 CASTER/FDR

Casper (Compiler for the Analysis of Security Protocols) is a compiler developed to facilitate the process of designing a protocol based on CSP (Communication Sequential Process)[11]. To design a protocol, types of variables and functions, initial states of communication agents, sequence of exchanging messages between agents, security attributes of interest, real data types and names need be declared. Then, functions used in the protocol, representation of initial states of communication agents, and intruders' initial states need be designed[12]. When the program is run, it automatically performs the conversion to a CSP document. The FDR (Failure Divergence Refinements) program tests the converted document in terms of whether it meets the security and authentication attributes. The FDR verifies the safety, deadlock and livelock of a protocol and facilitates the analysis of vulnerabilities[7,8].

## 3. Proposed Protocol

The proposed protocol was designed by using random numbers and variables varying with each session and based on vector values and hash functions.

The symbols are defined in Table 1.

**Table 1.** Symbols and definition

| Symbols | Definition |
|---|---|
| Tag | Agent |
| Reader | Agent |
| S | Server |
| H | Hash Function |
| x, k, y | Nonce |
| sk 1, sk 2 | Session Key |
| Vector 1, Vector 2 | Vectors |

### 3.1 Operation

The proposed protocol operates in the following steps.

◎ (Step ① : Tag → Reader)

The Tag receives a Query from the Reader, and generates the Hash Function (Vector1) using the Vector1 value. Also, the Tag generates the Nonce and session keys prior to the concatenation with the values generated. Then, the Tag stores the values in the variable %M1, and transmits the calculated H (Vector1),{x}{sk1}%m1 to the Reader. Here, the generated value is unique that cannot be generated by another Tag. As for the hash value, the fixed-length data are hashed as follows.

For the first vector function, $\tilde{a} = (a_0,...,a_k)$ and an integer 2w are calculated as, $h_{\tilde{a}}(\bar{x})^{strong} = (a_0 \sum_{i=0}^{k} a_{i+1} x_i \bmod 2^{2w}) \div 2^w$ which is applied to the string of the transmitted data value as follows: $h_a(\bar{x}) = h_{int}((\sum_{i=0}^{k} x_i \cdot a^i) \bmod p)$ where $a \in [p]$ is uniformly random and $h_{int}$ is chosen randomly from a universal family mapping integer domain [p] → [m].

Therefore, following the hash operation, $h_a(Vectors) = h_{int}((\sum_{i=0}^{k} x_i \cdot a^i) \bmod p)$, x, sk1, and the concatenation operation, H(Vector1),{x}{sk1}%m1 is sent to the Reader.

◎ (Step ② : Reader → S)
Using the value of H(Vector1), {x}{sk1}%m1 from the Tag and its own Sk2, k, the Reader calculates {m1%{x}{sk1},sk2,k}{sk2}, which is followed by an operation with the data from the Tag and the concatenation operation. Once the {m1%{x}{sk1},sk2,k}{sk2} value is normally generated, the Reader sends it to S.
◎ (Step ③ : S → Reader)
S uses the {m1%{x}{sk1},sk2,k}{sk2} value from the Reader for an operation and mutual authentication. Using the value transmitted by the Reader, S calculates the value as {Tag,x,{k}{sk1}%m2}{sk2}. Then, S concatenates the {Tag,x,{k}{sk1}%m2}{sk2} value and transmits it to the authenticated Reader.
◎ (Step ④ : Reader → Tag)
The Reader receives the {Tag,x,{k}{sk1}%m2}{sk2} value from S and checks it for authentication. Then, the Reader performs an operation and generates m2%{k}{sk1},{x}{k},H(Vector2). Then, the Reader calculates the hash value $h_a(Vector2) = h_{\text{int}}(\ (\sum_{i=0}^{k} x_i \cdot a^i) \bmod p)$ and k, sk1, x, k and does a concatenation to generate the m2%{k}{sk1},{x}{k},H(Vector2) value. Once the value is generated normally, the Reader transmits it to the Tag for authentication.
◎ (Step ⑤ : Tag →Reader)
Finally, the Tag receives from the Reader the value of m2%{k}{sk1},{x}{k}, $h_a(Vector2) = h_{\text{int}}(\ (\sum_{i=0}^{k} x_i \cdot a^i) \bmod p)$ and compares it with the value it holds. Once the two values match, the tag uses its ID for the operation, $h_a(Vector2) = h_{\text{int}}(\ (\sum_{i=0}^{k} x_i \cdot a^i) \bmod p)$, Reader(+)H (Tag), and encryption and transmits the result to the Reader, completing its authentication session. Subsequently, the Reader receives the value of $h_a(Vector2) = h_{\text{int}}(\ (\sum_{i=0}^{k} x_i \cdot a^i) \bmod p)$, Reader (+) H (Tag) from the Tag and sends it to S. Then, S retrieves the stored value for the Tag to compare it with the value received from the Tag for authentication. Upon completion of normal authentication process, hash codes and Tag codes are checked for subsequent operations.

## 4. Conclusion

The present paper designed a vehicular security protocol and verified its safety, deadlock and live lock operations using the model checking program of FDR 2.91 released for research. In Figure 1, the source file is loaded and converted to the CSP source without any compile errors. The converted CSP format is verified by CASPER. As the CSP code is complex and prone to mistakes in manual coding, it is recommended to use FDR for the conversion.

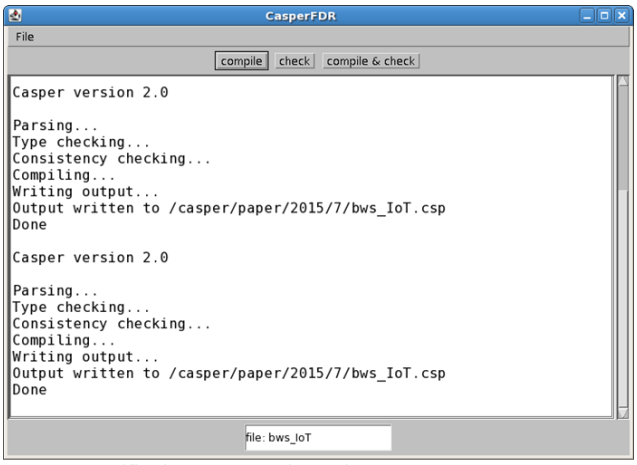


**Figure 2.** Verification Set-Up and Running.

Once the compiling is completed, an in-depth verification is performed to see whether it is safe against different types of attacks as in Figure 2. Here, the tick marks on the left side in Figure 2 indicate the attributes meet the security requirements.

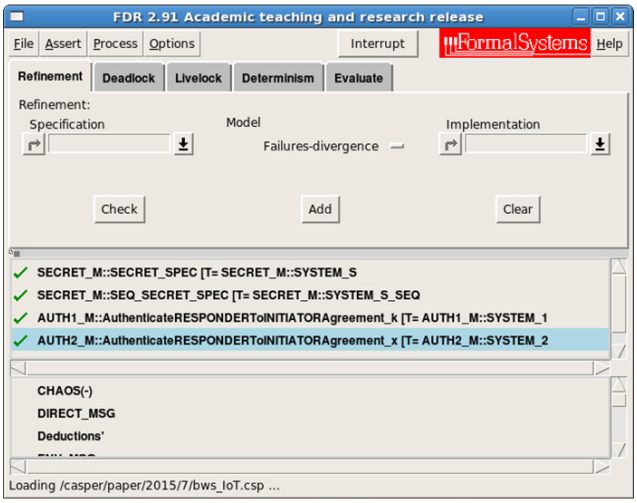


**Figure 3.** Security Verification Results of the Protocol.

Figure 2 presents 4 outputs from the verification, each of which is described below.

- SECRET_M::SECRET_SPEC[T=SECRET_M::SYSTEM_S
  This indicates the proposed protocol is secure enough to guard against intruders and attacks on the system. The communication between agents, data values,

session keys and hash functions are safe and secure.

- SECRET_M::SEQ_SECRET_SPEC[T=SECRET_M::SYSTEM - S_SEQ
  This is the result of verifying if the protocol operates normally in each step of each communication section in the system, if it falls into any infinite loop, and if it encounters any errors. The proposed protocol operates as a safe process.
- AUTH1_M::AuthenticateRESPONDERToINITIATORAgreement_k[T=AUTH1_M::SYSTEM_1
- AUTH1_M::AuthenticateRESPONDERToINITIATORAgreement_x[T=AUTH2_M::SYSTEM_2
  3. and 4. verify if the Responder and Initiator of k and x can mutually authenticate each other in a normal communication. Mutual authentication is successful in each step, indicating the protocol's safety against external attacks.

## 5. Conclusion

It was not until recently that many studies dealt with the security of vehicular inter-device communication. In near future, multiple hardware devices will be hyper-connected for communication. Yet, malevolent intruders taking advantage of the vulnerabilities of wireless sections are likely to commit attacks hindering system operation. Such vulnerabilities and resultant attacks on vehicles are significant issues directly related to people's lives. This paper proposed an efficiently secure and safe communication protocol to cope with security issues in vehicular communication. The proposed protocol is based on hash operation, public keys, vectors and random numbers, and designed to perform an operation in each communication session, yielding a unique output value for mutual authentication. The formal verification with Casper/FDR proved the safety of each item tested. The proposed protocol worked efficiently without redundant calculations, and the design was found to benefit the safe and secure vehicular communication environment. Future studies will extend the scope of communication security to a range of security gateways.

## 6. Acknowledgment

This work was supported by the research grant of Baekseok University in 2015.

## 7. References

1. Gope P, Hwang T. Untraceable sensor movement in distributed IoT infrastructure. IEEE Sensors Journal. 2015; 15(9):5340–8.
2. Niu B, Zhu Xa, Li Qc, Chen Ja, Li Ha. A novel attack to spatial cloaking schemes in location-based services. Future Generation Computer Systems. 2015; 49:125–32.
3. Lin X-J, Sun L, Qu H. Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications. Computers and Security. 2015; 48:142–9.
4. Cirani S, Picone M,Gonizzi P,Veltri L,Ferrari G. IoT-OAS: An oauth-based authorization service architecture for secure services in IoT scenarios. IEEE Sensors Journal. 2015; 15(2):1224–34.
5. Ray BR, Abawajy J, Chowdhury M. Scalable RFID security framework and protocol supporting Internet of Things. Computer Networks. 2014;67:89–103.
6. ISO 26262. Road vehicles - Functional safety, Management of functional safety and Concept phase.
7. Casper GL: A compiler for the analysis of security protocols. User Manual and Tutorial; 2009. Version 1.12. 2009.
8. Formal Systems (Europe) Ltd. Failures-Divergence Renement. Oxford University Computing Laboratory. FDR2 User Manual. 2010; p. 19.
9. Hoare CAR. Communicating Sequential Processes. Upper Saddle River, USA: Prentice-Hall; 1985.