ECU 보안성 유지를 위한 인증서 기반 암호시스템 설계

유요셉, 김기천*
건국대학교 컴퓨터• 정보통신공학과
e-mail : josephyoo94@konkuk.ac.kr, kckim@konkuk.ac.kr*

A Design of Certificate-based Encryption System for Maintaining ECU Security

Joseph Yoo, Keecheon Kim*
Dept. of Computer Science, Konkuk University

요 약

기존의 엔진제어기(ECU)는 주요 mapping data(ECU 에 대응되는 값)들에 대해서 기밀성과 무결성을 제공하는 보호 기법의 부재로 인해 임의적인 튜닝이 가능하다는 특징이 존재하였다. 이로 인해 자칫 잘못된 튜닝이나 악의적인 조작이 발생할 수 있는 여지가 있으며, 이는 차량 엔진 및 조작의 안정성을 떨어뜨림과 동시에 운전자 및 보행자들의 안전을 위협할 수 있다는 문제가 있다. 이에 본 논문에서는 ECU 에 적용되는 Firmware 의 주요 mapping data 를 안전하게 암호화하는 방식을 제안하며, 이 과정에서 차량의 식별 및 ECU 에 mapping 되는 data 의 무결성 검증을 위해 인증서를 사용하는 방식을 제안한다. 본 논문의 제안을 통해 주요 mapping data 를 안전하게 보호하는 기술을 통해 차량의 안정성을 유지할 수 있다.

1. 서론

과거에 기계적으로 동작하던 핸들과 브레이크와 같은 장치들은 오늘날 전자적인 제어장치로 대체되었다. 특히, 차량의 엔진은 엔진제어기(Engine Control Unit, ECU)를 통해 제어가 이루어진다. [1] ECU 의 동작은 firmware 에 지정된 데이터를 기반으로 RPM 조절과 공기 유입량과 같은 제어를 하게 된다. 이러한 ECU 의 특징으로 인해 데이터의 보안이 이루어지지 않고 있는 현재의 엔진은 불법적인 임의의 튜닝이 가능하다. 이러한 튜닝 중 ECU 의 소프트웨어적인 데이터를 수정하는 과정을 ECU Remapping 이라고 부르며, 주로 차량의 엔진 출력 및 연비를 향상시키기 위한 목적으로 사용된다. 자동차 제조사에서는 엔진의 성능뿐 아니라 내구성에 대한 고려를 하고 엔진을 제작하게 되며, 이에 적합한 공연비(공기와 연료의 비율)와 같은 값을 mapping 한다. 이를 사용자가 임의의 Remapping 을 통해 변경할 경우, 차량이 승인 받은 안정성을 보장할 수 없으며, 배기가스 배출 역시 변화가 발생할 수 있다는 문제점이 존재한다. 더 나아가, 악의적인 공격자가 ECU Mapping 데이터를 변경할 경우, 설정된 값에 따라 주행 중 매우 위험한 상황이 발생할 수 있다. [2] 따라서 ECU 의 Mapping 데이터를 안전하게 보안해야 하며, 튜닝이 필요할 시에는 정식으로 인증된 기관에서만 수행할 수 있도록 하는 방안에 대한 연구가 필요하다. 본 논문에서는 ECU firmware 데이터 보안을 위해 인증서를 사용한 데이터 암호화 방식을 제안하며, 이를 통해 승인되지 않은 ECU Remapping 을 차단하는 구조를 디자인하였다.

2. ECU Remapping

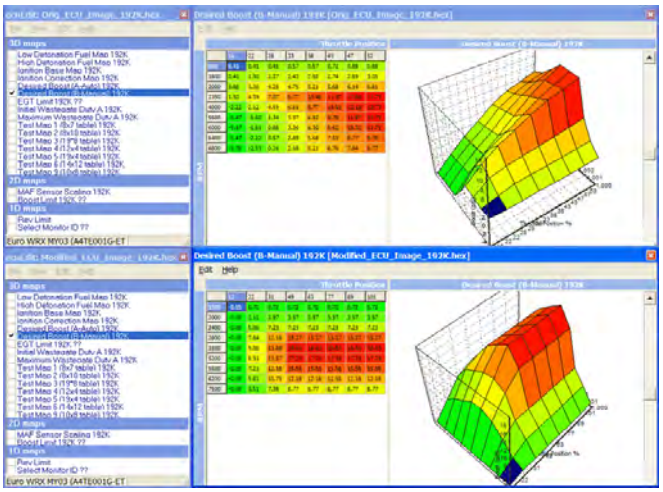
엔진제어기(ECU)는 차량의 엔진을 제어하는 핵심 장치로써, firmware 에 mapping 된 데이터를 기반으로 엔진의 동작을 관리한다. 자동차 제조사는 차량의 생산 시, 엔진의 기종과 차량의 제원에 따라 안정성과 내구성을 고려한 ECU mapping 을 수행한다. 이 mapping 은 실제 엔진이 보여줄 수 있는 성능에 비해 낮게 맞춰져 있다. ECU mapping 의 제약으로 인해 사용자가 차량 튜닝 시 부품의 교체만으로는 성능의 향상이 두드러지지 않는다. 따라서 요즘 크게 성장하는 튜닝 분야는 ECU Remapping 이라는 방식이다. 이는 ECU 에 mapping 되어있는 기존의 데이터를 새로 변경시키는 방법으로, 제조사에서 제한을 걸어둔 엔진의 성능을 벗어나서 끌어올릴 수 있다. [3]

현재 미국 등 일부 해외에서는 이러한 ECU Remapping 이 활발하게 이루어지고 있으며, 국내도 그 빈도가 늘어나는 추세이다.

그러나 ECU Remapping 의 문제 역시 존재한다. 국

*) 교신저자

내에서는 배출가스 증가 및 엔진의 내구성 저하라는 이유로 불법 개조로 분류가 되고 있으며, 자동차 제조사들도 Remapping 된 엔진에 대해서는 보증을 하지 않는다. 또한 잘못된 설정으로 Remapping 을 진행할 경우, 변경된 엔진의 설정 값으로 인해 주행 도중 사고를 유발할 수도 있는 만큼 안정성을 보장하기란 쉽지 않다. [4]



[그림-1] ECU Remapping Program

3. 인증서 기반의 ECU Firmware data 암호화 및 firmware 버전 관리 제안

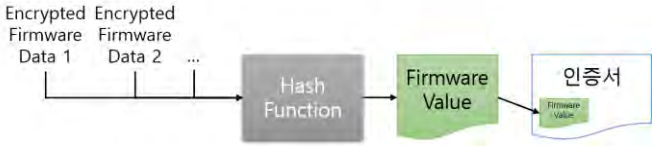
본 장에서는 불법적인 ECU Remapping 을 차단하기 위해 firmware 의 데이터를 암호화하는 기법을 제안한다. 기존의 암호화 되어있지 않은 ECU firmware 의 데이터는 프로그램만 이용하면 간단하게 Remapping 을 수행할 수 있었다. 이 데이터를 특정 key(K_1)를 사용하여 암호화하여 보관을 하고, 차량의 구동 시(시동이 걸리는 시점)에 복호화를 하여 mapping 을 시키는 방안을 제시한다. 본 장에서 디자인한 인증서 발급 및 갱신 시 이루어지는 firmware 데이터의 암/복호화 방식은 다음과 같다.

I. 차량 출고 시 차량용 인증서를 발급 후 적재
II. Firmware 관리 주체인 제조사는 Firmware 의 data 를 암호화하는 비밀키(K_1)를 임의로 생성하여 데이터 암호화 적용 후, 차량에 적재
III. Firmware 의 암호화된 주요 mapping 데이터와 firmware version number 를 hash 함수의 input 값으로 해서 Firmware Value 라는 해시 값(H_1) 생성 후 인증서의 내부에 저장
IV. 적재된 차량용 인증서 전체의 hash 연산을 통해 K_1 의 암호화 및 복호화에 사용하기 위한 키(K_2)를 생성하여 K_1을 암호화(K_1')
V. 차량 구동 시에는 암호화된 Firmware 데이터의 hash 연산을 통해 K_2 를 생성하여 K_1'의 복호화 후, K_1 을 통해 Firmware 데이터를 복호화

이 구조에서는 인증서 발급기관과 차량이 구성요소로써 존재한다. 주요 firmware 데이터는 차량에 항상 암호화되어 존재하며, firmware 데이터의 암호화 및 복호화에 사용되는 key(K_1)를 암호화 및 복호화하는 key(K_2)는 인증서 전체(뒤에 서술되는 'Firmware Value' 포함됨)를 hash 연산한 값으로 사용한다.

[그림-2] 인증서 기반의 K_2 생성 절차(K_1 보호)

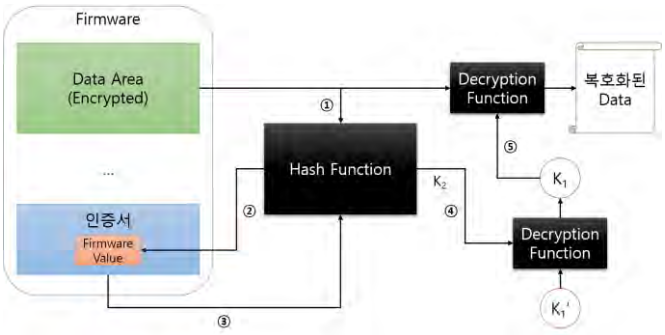
본 장에서는 또한 firmware 의 버전 관리 및 무결성 검사를 위해 'Firmware Value' 라는 값을 제안한다. 이 값은 firmware 의 주요 암호화된 mapping 데이터와 firmware version number 를 hash 함수의 입력으로 넣어 구한 해시 값(H_1)으로, 인증서의 내부의 한 영역에 저장된다.



[그림-3] Firmware Value 생성 및 저장

Firmware Value 는 차량이 구동될 시, firmware 에 존재하는 mapping 데이터들 중 암호화 된 주요 항목들을 읽어와서 생성한다. 만일 Brute Force 공격과 같은 방법으로 암호화 키를 알아내어 데이터의 값을 변형시킨다고 해도, 차량의 구동 시 Firmware Value 가 변형되어 인증서의 부분적인 변화가 발생하게 된다. Hash 함수의 특성 상, 부분의 변화는 전혀 다른 결과를 생성하기 때문에 데이터의 변형 시에는 K_2 의 값이 완전히 달라지는 결과를 얻게 된다. 따라서 ECU Mapping 데이터의 변화가 일어나게 되면 K_1'(data 암호 키)의 정상적인 복호화가 이루어지지 않아 차량의 구동을 차단할 수 있다.

차량의 구동 시, 발생하는 hash 연산 및 복호화 작업은 [그림-4]와 같다.



[그림-4] Firmware Value 및 K_2 생성 및 K_1을 통한 Firmware Data 복호화

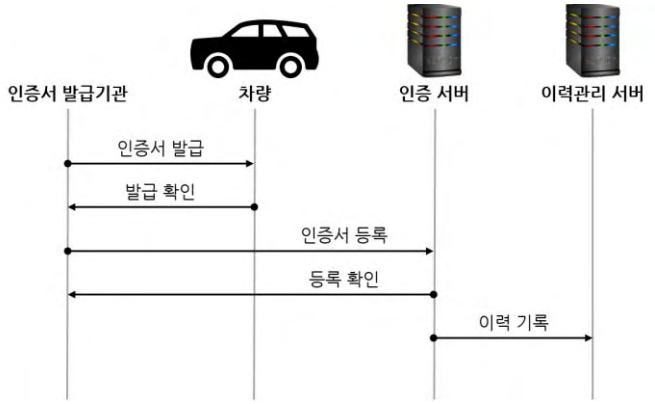
[그림-4]에서 각 단계는 다음과 같다.

① 암호화된 firmware 데이터는 hash 함수의 input
② hash 의 결과(H_1)가 Firmware Value 가 되며, 이를 인증서 내부에 업데이트
③ 인증서 전체를 hash 함수의 input 으로 넣음
④ hash 의 결과로 구한 K_2 를 key 로 사용하여 암호화된 K_1'으로부터 K_1를 복호화
⑤ K_1 을 통해 firmware 데이터의 값을 복호화

Firmware 버전 관리의 경우, 인증서를 통한 차량 인증을 인가된 서비스 센터를 통해 수행한 후, 업데이트를 수행하게 된다. 이때, 서비스 센터를 통한 정식 인증이 완료되면 차량의 인증서는 새로운 인증서로 갱신된다. 인증서 관리 부분에 관련해서는 다음 장에서 다루도록 한다.

4. 인증서 관리 방식

본 장에서는 차량의 고유 식별 및 firmware data 암호화 및 복호화 key 인 K_1 을 암호화하여 보호하는 key(K_2)의 source 로써 작용하는 차량용 인증서의 발급 및 관리에 대해 다룬다. 본 논문에서 제안하는 시스템에서는 중앙 인증서 발급기관이 필요하며, 차량의 출고 시 인증서의 등록이 이루어진다. 이와 함께 별도의 인증 서버에 인증서가 등록되며 모든 일련의 과정은 이력관리 서버에 저장된다. 여기서 사용되는 인증서는 차량의 인증과 firmware 의 정보 등을 관리하는데 특화된 차량용 인증서를 지칭한다. 추후 인증이 발생하는 부분은 모두 인증서버를 통해 일어나게 되며, 인증서버를 거치는 모든 통신내역 역시 이력관리 서버에 기록된다.

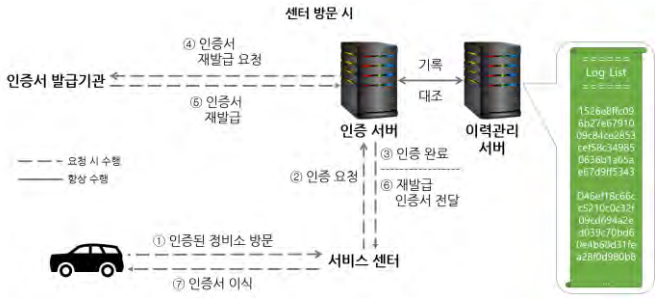


[그림-5] 차량 출고 시 인증서 발급

```
<VehicleModel>*yundai *onata Hybrid</VehicleModel>
<SerialNumber>86f7e437faa5a7fce15d1ddc</SerialNumber>
<UserData>b9eaeaea377667b8</UserData>
<...>  *****  </...>
<FirmwareValue>e9d71f5ee7c92d6dc9e92ffdad17b8bd49418f98</FirmwareValue>
```

[그림-6] 차량용 인증서 내용 예시

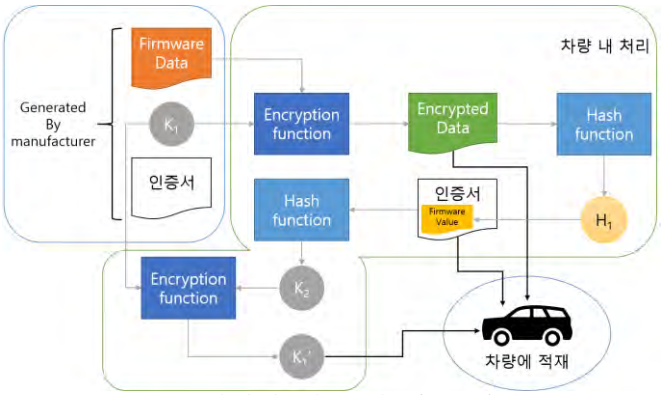
[그림-5]에서 발급된 인증서는 차량의 firmware 내부의 data 영역 중 하나를 인증서 영역으로 차지하게 된다. 이 인증서의 내부에 firmware 의 값들을 추출해서 생성한 hash 값인 Firmware Value(H_1)가 포함되며, 그 값이 포함된 인증서의 hash 를 통해 K_2 를 생성하여 K_1'을 복호화해서 firmware 데이터의 암호 키 K_1 을 구한다. 이 K_1 을 복호화 key 로 이용하여 구동 시 firmware 데이터의 복호화에 사용된다.



[그림-7] 차량 인증 및 인증서 갱신

만일 위 인증과정에서 이력관리 서버에 기록된 이력과 현재 차량의 이력이 다르게 나올 경우, 이는 인증서 혹은 데이터의 변조가 일어났다고 판단을 할 수 있다.

차량의 인증서는 K_1 의 암/복호화 key 인 K_2 의 source 로 사용되는 만큼 일정 기간마다 갱신이 필요하다. 인증서의 갱신은 인가된 서비스 센터에 방문해서 차량의 인증이 완료되면 새로 발급을 하도록 한다. 이때, 차량의 인증서가 변하므로 K_2 값이 변하게 된다. 따라서 K_1 의 값을 다시 새로운 인증서를 통해 생성한 key(new K_2, 이하 K_2)로 암호화하여 저장하는 절차가 필요하다. 이 과정에서 인증 여부, 인증서 재발급 여부, firmware 의 정상적인 변형 여부가 이력관리 서버에 기록된다. 만일 신규 firmware 버전이 있을 경우, 해당 firmware 를 다운로드 받은 후 재발급 받은 new K_1 을 통해 firmware 데이터를 암호화하여 차량에 적재한다. 이후 차량의 인증서로 생성된 K_2 로 K_1 을 암호화하여 K_1' 역시 차량에 탑재한다. 인증서의 초기 생성과 갱신, 그리고 firmware 의 변경 시 일어나는 과정은 다음 [그림-8]과 같다.



[그림-8] 데이터의 생성 및 차량 적재 과정

5. 결론 및 향후 발전방향

스마트카와 자율주행차량 기술이 발전하고 차량 간 통신이 연구됨으로 인해서 전자적으로 제어되는 장치의 보안성에 대한 관심이 증가하고 있다. 특히 차량의 핵심 부품인 엔진을 제어하는 엔진제어기(ECU)의 mapping data 를 안전하게 보호해야 할 필요성이 보고되고 있다. [5]

본 논문에서는 차량 내 ECU 의 firmware data 의 보호를 위해서 인증서 기반의 암/복호화 시스템을 제안했다. 인증서를 발급하는 기관을 통해 인증서의 생성과 재발급이 이루어지며, 이를 인증서버와 이력관리 서버에서 기록을 관리한다. Firmware Value 라는 해시값(H_1)이 내장된 인증서를 다시 hash 함수를 지나면서 생성한 key(K_2)를 통해 firmware data 의 암호화와 복호화를 수행할 수 있는 key(K_1)을 보호한다. 1 차적으로는 데이터의 암호화로 인해 사용자 임의의 수정이 불가능하고, 2 차적으로 암호가 Brute Force 공격과 같은 방식으로 노출이 되었을 때에는 데이터가 변경되어도 Firmware Value 가 함께 변경되도록 설계하여 차량의 구동을 차단하도록 하였다.

인증서는 단순히 K_1 의 암/복호화 key 로써 사용되는 것이 아닌, 차량의 식별 및 firmware 무결성 유지 방식으로도 활용된다. 이력관리 서버의 내용과 실제 인증서가 보유한 내용에 차이가 발생할 경우, 이를 서비스 센터에서 인지할 수 있으며 이를 통해 보증 정책 등의 변화를 줄 수 있다.

본 논문의 인증서 방식은 중앙의 인증서 발급기관을 두고 각 차량에 배포하는 방식이므로 향후 자율주행차량에서의 V2V 통신에서의 상호인증 수단으로 발전하여 사용할 수 있으리라 기대된다.

ACKNOWLEDGEMENT

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신 기술진흥센터의 지원(No.B0511-16-0001, 글로벌 딜리버리 클라우드 플랫폼의 대규모 OTT 서비스 적용을 위한 방송· 통신 사업자 공동의 시범 사업)으로 수행된 연구임

참고문헌

[1] 김영길. (1997). [특집] 엔진 전자제어 장치 (ECU) 의 개발 기술과 발전 방향. 오토저널, 19(2), 26-31.

[2] 이혜련, 김경진, 정기현, 최경희, 박승규, 권도근. (2013). 자동차용 ECU 의 CAN 메시지를 통한 자동차 공격 방법 연구. 한국컴퓨터정보학회논문지, 18(11), 39-49.

[3] Cains, T., Pates, D., and Warth, M., "Improving Speed and Accuracy of Gasoline and Diesel Engine Testing via Closed-Loop Combustion Control," SAE Technical Paper 2012-01-0902, 2012, doi:10.4271/2012-01-0902.

[4] Sasan Jafarnejad, Lara Codeca, Walter Bronzi, Raphael Frank, Thomas Engel. “A Car Hacking Experiment: When Connectivity meets Vulnerability”, 2015, IEEE

[5] I. Studnia, V. Nicomette, “Survey on Security Threats and Protection Mechanisms in Embedded Automotive Networks”. Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference. Budapest, Hungary, 2013