情報保護學會論文誌
第 22 卷 第 5 號, 2012. 10

CAN 버스 공격에 안전한 메시지 인증 및 키 분배 메커니즘*

조 아 람,† 조 효 진, 우 사 무 엘, 손 영 동, 이 동 훈‡
고려대학교 정보보호대학원

A Message Authentication and Key Distribution Mechanism Secure Against CAN bus Attack*

A-Ram Cho,† Hyo Jin Jo, Samuel Woo, Young Dong Son, Dong Hoon Lee‡
Division of information security, Korea University

요 약

차량 기술이 발전함에 따라 차량 내부에는 많은 수의 ECU(Electronic Control Unit)가 탑재되고 있다. 차량 내부에 탑재된 ECU간의 통신은 CAN(Controller Area Network)을 통해 이루어진다. CAN은 높은 신뢰성을 갖기 때문에 안전한 차량통신을 지원한다. 하지만 별도의 보안메커니즘이 적용되어 있지 않기 때문에, 많은 취약점을 내포하고 있다. 최근 연구에서는 CAN의 취약점을 이용한 공격이 제시되고 있다. 본 논문에서는 이동 통신망을 이용한 차량 내부 네트워크에 대한 원격공격 모델을 제시한다. 또한 차량 내부 메시지의 기밀성과, 무결성을 보장하면서 동시에 리플레이 공격을 방지할 수 있는 안전한 차량 내부 네트워크 메시지 인증 메커니즘을 제시한다.

ABSTRACT

According to advance on vehicle technology, many kinds of ECU(Electronic Control Unit) are equipped inside the vehicle. In-vehicle communication among ECUs is performed through CAN(Controller Area Networks). CAN have high reliability. However, it has many vulnerabilities because there is not any security mechanism for CAN. Recently, many papers proposed attacks of in-vehicle communication by using these vulnerabilities. In this paper, we propose an wireless attack model using a mobile radio communication network. We propose a secure authentication mechanism for in-vehicle network communication that assure confidentiality and integrity of data packets and also protect in-vehicle communication from the replay attack.

Keywords: CAN message authentication, CAN bus attack, In-vehicle Security, Key Management

I. 서 론

IT 기술의 발전과 함께 차량에도 다양한 IT 기술이 적용되고 있다. 전자식 연료분사장치가 등장하고, ABS(Anti-Lock Braking System), 차체자세제어장치와 같은 전자식 제어장치가 등장하였다. 최신식의 차량은 기계 장치뿐 아니라 100MB이상의 바이너리 코드(Binary Code)와 함께 수많은 컴퓨터 - Electronic Control Unit(ECU)를 탑재하고 있다〔1〕. 이를 통해 차량 조작의 대부분을 전자식으로 제어하고 에어백과 같은 안전장치, 오디오나 에어컨과 같은 편의장치 역시 전자식으로 제어하고 있다. 차량

접수일(2012년 4월 18일), 수정일(2012년 8월 14일), 게재확정일(2012년 8월 15일)
* 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. 〔KI002113, Car-헬스케어 보안 기술개발〕
† 주저자, docso21@hanmail.net
‡ 교신저자, donghlee@korea.ac.kr

의 제어가 전자식으로 전환됨에 따라 차량 운전자의 편의성은 비약적으로 증대되었다. 하지만 과거 폐쇄망 성격을 갖던 차량 내부 네트워크가 차량 내부의 OBD-Ⅱ(On-Board Diagnostic 2)단자를 통한 연결이나, V2I, V2V와 같은 통신기술의 발전에 따라 외부에 개방되었고, 이에 따라 악의적인 해커에 의한 외부 공격 위협성 역시 증가 하게 되었다[2]. 과거 30년간의 차량 보안 기술은 물리적인 차량 보안 - 도난방지 시스템(Anti Thief System)에 집중되어 있었고, 차량 내부 네트워크에 대한 보안 기술은 거의 전무한 상태라 할 수 있다.

차량 내부에 탑재되는 ECU들은 CAN(Controller Area Network), LIN(Local Interconnect Network), FlexRay, MOST(Media Oriented Systems Transport)등을 이용하여 상호간의 통신을 수행한다. CAN 프로토콜은 한 쌍의 꼬임선 으로 구성되어 있는 물리적인 특성 때문에 외부 전자파나 노이즈에 강한 장점을 가지고 있다[3]. CAN 프로토콜은 엔진이나, 브레이크와 같은 차량의 구동에 관련한 제어와 오디오, 내비게이션과 같은 멀티미디어 기기의 제어에 핵심적인 역할을 한다. CAN 프로토콜은 국제표준화기구(ISO)와 자동차 엔지니어협회(SAE)에 의해 국제표준화가 되어있지만, 현재 CAN 프로토콜에는 어떠한 보안 메커니즘도 적용되어있지 않다[3].

최근 연구에서는 CAN의 취약점을 이용한 공격 모델을 제시하고 있다. 더불어 취약점을 보완할 수 있는 방법 역시 연구가 진행되고 있으나 근본적인 보완책은 아직 미비한 상태이다[1,2,4,5].

본 논문에서는 CAN의 취약점을 이용한 공격모델과, 이를 보완하기 위한 CAN 통신 보안 메커니즘을 제시한다. 2장에서는 관련 연구를 소개하고, 3장에서는 위협모델을 제시한다. 4장에서 키 분배 및 메시지 인증메커니즘을 제시한 후 5장에서는 본 논문에서 제시한 메커니즘에 대한 안전성 분석과 효율성 분석을 한다.

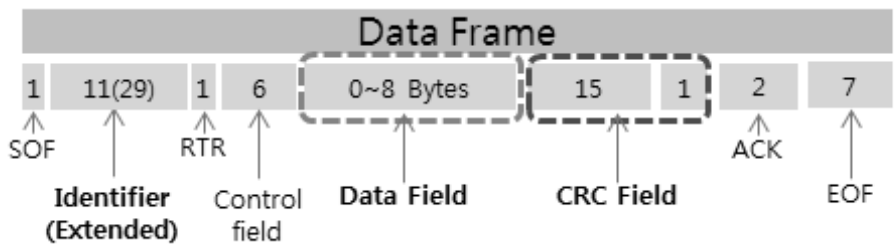
II. 관련 연구

2.1 CAN

1970년대 이후 차량의 고성능화, 지능화와 함께 차량 내에는 많은 수의 ECU가 탑재되었다. CAN은 높은 신뢰성을 갖는 Serial Bus System으로써 실시간 제어능력이 중요시되는 차량용 네트워크에 적합하기 때문에 여러 가지 차량 제어 프로토콜 중 사실상 표준으로 사용되고 있다[3,6]. CAN은 메시지의 전송속도에 따라 High-Speed CAN과 Low-Speed CAN으로 나뉜다. 전자는 주로 비교적 세밀한 제어를 해야 하는 엔진이나 브레이크의 제어를 위해 사용되고, 후자는 전조등이나 멀티미디어 기기의 조작과 같은 편의 사항을 제어하는데 사용된다[6].

차량의 CAN은 메시지 전송을 위해 Broadcast방식의 통신 구조를 채택하고 있다. CAN의 메시지 전송에는 Data 프레임, 리모트 프레임, 에러 프레임, 오버로드 프레임이 사용된다. ECU간의 실질적인 통신은 [그림 1]과 같은 Data 프레임을 통해 이뤄진다. Data 필드에 0~8byte의 메시지를 담고 있으며, Identifier 필드의 값을 통해 수신 노드에서 수신 여부를 결정하고, CRC 필드를 통해 Data 프레임의 오류 여부를 판별한다. Identifier 필드의 크기는 CAN 2.0A 버전에서는 11bit, 2.0B 버전에서는 29bit이고, CRC 필드의 크기는 16bit이다. 이 외의 SOF(Start-of-Frame) 필드, RTR 필드, Control 필드, ACK 필드, EOF 필드의 설명은 본 논문에서 제시하는 메커니즘과는 밀접한 연관이 없기 때문에 생략한다.

CAN의 네트워크 토폴로지는 기본적으로 버스형을



[그림 1] CAN Data 프레임

채택한다. 메시지 전송 방법은 임의의 ECU에서 전송할 메시지가 있을 때 먼저 버스의 상태를 확인하여, Idle상태일 경우 메시지를 전송하고, 그렇지 않을 경우 수신모드에 들어간다. 만약 메시지를 전송하는 도중에 충돌이 발생할 경우에는 Data 프레임내의 Identifier 필드 값을 통해 우선순위를 확인한다. 메시지를 수신할 때에는 Data 프레임내의 Identifier 필드의 값을 통해 수신여부를 판단하며, 이 때 Identifier 필드 값이 작을수록 더 높은 우선순위를 갖는다.

2.2 CAN의 취약점을 이용한 공격과 그 보안책

2.2.1 CAN의 취약점

K. Koscher 등은 CAN의 취약점을 세 가지 특성을 들어 분석하였다[1]. 먼저 CAN의 메시지 수신방법인 Broadcast방식 때문에 공격자는 물리적으로 모든 노드에서 메시지를 얻을 수 있고, 이로 인해 공격자의 도청에 취약하다고 언급하였다. 또한 ECU가 버스상의 메시지 중 ID값이 작거나, 많은 메시지를 발생시키는 우세한(Dominant)상태의 메시지를 수신할 확률이 높기 때문에 이로 인해 서비스 분산공격에 취약하다고 언급하였다. 마지막으로 Data 프레임 내에 메시지를 송신하는 ECU를 인증할 수 있는 별도의 필드가 없는 문제를 지적하였다. 이 외에도 S. Ravi 등은 CAN의 취약점을 기밀성, 무결성, 인증여부, 가용성, 부인방지의 다섯 가지 측면에 대해 분석하였다[7].

2.2.2 CAN의 취약점을 이용한 공격

D. K. Nilsson 등은 먼저 CAN에 대한 일반적인 공격 방법으로 Read, Spoof, Drop, Modify, Flood, Steal, 그리고 Replay의 8가지로 정의하였다. 또한 이를 이용하는 CAN의 논리적/물리적인 공격을 제시하였다[8].

K. Koscher 등은 실제 차량의 CAN에 대한 실험을 수행한 후 그 결과를 제시하였다[1]. 공격 실험은 실험실 내에서의 차량의 특정 ECU를 통한 공격 실험과, 실제 주행 중인 차량에 대한 공격 실험의 두 가지로 진행되었다. 먼저 실험실 내에서의 공격을 위해 차량의 Data 프레임을 수집하고, 이를 퍼징(Fuzzing)이나 역공학(Reverse engineering)을 통해 분석하여 특정 동작을 수행하는 실험을 진행 한 후, 같은 실험을 주행 중인 차량에 대해 수행하였다. K. Koscher 등은 위의 실험들을 통해 CAN Data 프레임을 재전송하거나, 특정 동작을 수행하는 Data 프레임을 CAN에 전송함으로써, 차량의 물리적인 구동을 일으킬 수 있었다.

T. Hoppe 등은 네 가지의 공격 시나리오를 정의한 후, 각 시나리오에 대한 공격방법을 제시하고, 해당 공격방법이 목적으로 하는 CAN의 취약점과, 해당 공격으로 인해 예상되는 실질적인 피해를 제시하였다[4].

2.2.3 CAN 보안방법

다수의 연구에서, CAN의 취약점에 대한 보안 방법을 제안하였다[1,2,4,5,8]. T. Hoppe 등은 CAN 공격에 대한 침입 탐지 기법을 제안하였다[4]. 침입 탐지를 위한 패턴으로써, 공격을 위해 CAN 버스에 주입되는 Data 프레임의 빈도를 통한 탐지, 사용되지 않는 메시지 ID를 포함하는 Data 프레임을 통한 탐지, 물리적인 전위 특성을 통한 탐지를 제시하였다. M. Wolf 등은 전자 서명과, 인증서를 통한 ECU 소프트웨어의 안전한 업데이트를 제안하였다[2]. 제안된 방법은 소프트웨어 업데이트를 통한 공격자의 공격에 대한 보안방법을 제시하였다. 하지만 T. Hoppe 등의 연구는 공격에 대한 직접적인 보완책이 아닌 사후 대처 방법이며, M. Wolf 등의 연구는 메시지의 도청이나 위/변조에 대한 보완책이 아닌 소프트웨어 업데이트시의 공격에 대한 보완책이므로, 앞서 언급한 CAN의 취약점을 보완하지는 못한다. CAN 메시지 보호를 위해 Dennis K. 등은 4개의 Data 프레임에 대한 메시지 인증코드를 생성한 후, 생성된 메시지 인증코드를 4등분하여 다음 4개의 Data 프레임에 삽입하는 메커니즘을 제안하였다[5]. 제안된 보안방법은 Data 프레임의 인증을 위해 4개의 메시지를 기다린 후 처리해야 하므로 차량의 실시간적인 데이터 처리 환경을 만족시킬 수 없는 단점을 지닌다.

III. 위협모델

본 장에서는 먼저 공격자와, 공격자의 능력을 정의한 후, 공격 목적을 제시한다. 마지막으로 CAN의 취약점을 이용한 공격 시나리오를 제시한다.

〔표 1〕 공격자의 능력 정의

물리적인 접근 능력	특별한 물리적 접근 능력을 필요로 하지 않음
CAN 관련지식	CAN Data 프레임에 대한 분석/재조합을 할 수 있음
소프트웨어 관련 지식	스마트 폰 어플리케이션의 제작 및 운용을 할 수 있음

3.1 공격자 분류

본 논문에서 제시하는 공격자는 〔표 1〕과 같은 능력을 갖는 것으로 정의한다. 또한 공격자는 D. K. Nilsson 등이 제시한 차량 내부네트워크에 대한 공격방법〔8〕 중 Data 프레임에 대한 Read, Replay, 그리고 Flood의 수행만으로 공격을 수행할 수 있다.

3.2 공격목적

공격자는 공격을 통해 아래 제시하는 두 가지 목적을 달성할 수 있다.

① CAN 버스 상의 차량관련 정보 수집 및 개인 정보 도청
② 원격에서의 차량 조작으로 인한 정상적인 차량 기능의 마비

특히, 향후 차량에서 다양한 서비스(금융결재, Healthcare 서비스 등)를 지원하였을 때, 공격자는 개인 금융정보나, 개인 병력과 같은 개인 정보를 도청을 통해 취득할 수 있다.

3.3 공격 시나리오

최근 스마트폰을 이용하여, 차량 운전자에게 편의성을 제공하는 서비스가 활발하게 제공되고 있다. 그 중 한가지인 차량진단 어플리케이션은 차량의 OBD-Ⅱ단자와 스마트폰을 유/무선으로 연결하여 사용자에게 차량의 현재 상태를 간단하게 확인 할 수 있게 한다. 본 논문에서 제시하는 공격시나리오는 최근 많은 차량운전자들이 사용하는 차량 진단 어플리케이션을 통한 공격으로써, 〔그림 2〕와 같은 환경을 갖는다. 상세한 공격 시나리오는 다음과 같다.

① 공격자는 어플리케이션 사용자의 동의 없이 CAN에 메시지를 송/수신하는 기능을 수행하는 악의적인 목적의 차량 진단 어플리케이션을 제작하여 배포한다.
② 공격대상이 되는 일반적인 차량운전자는 스마트폰을 통해 악의적인 어플리케이션을 사용한다. 어플리케이션 실행 시 사용자는 정상적인 서비스를 제공받으며, 공격자의 악의적인 행위를 인지할 수 없다.
③ 공격자는 차량 진단 어플리케이션과 이동통신망을 이용하여 원격으로 공격 대상 차량의 CAN의 Data 프레임을 도청하는 동시에, CAN에 메시지를 송신하여, 차량 내부 ECU에 대하여 자신이 원하는 제어를 수행한다.

〔그림 2〕 공격 시나리오

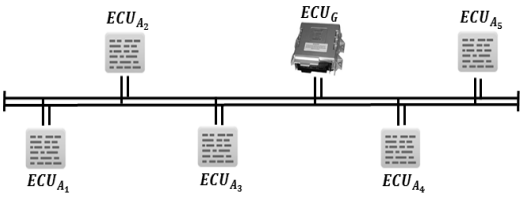
Ⅳ. 제안하는 메커니즘

본 장에서는 시스템 모델과 보안 요구사항 그리고 제약사항을 제시한다. 마지막으로 제시한 보안 요구사항을 만족하는 차량용 CAN의 보안 프레임워크(Framework)를 제시한다. 제안하는 프레임워크는 크게 인증서 관리 기술과, 인증서를 이용하는 키 분배와 키 업데이트 기술, 마지막으로 CAN 메시지의 기밀성과 메시지 인증을 제공하는 CAN 메시지 인증 프로토콜로 구성된다.

4.1 시스템 모델 및 보안 요구사항

4.1.1 시스템 모델

차량 내부에는 〔그림 3〕과 같이 다양한 기능을 수행하는 다수의 ECU와 1개의 Gateway ECU가 탑재되어 있다. Gateway ECU는 전체 ECU에 대한 인증서 관리와 키 분배 및 키 업데이트를 수행한다. 각 ECU에는 기기 인증서가 탑재되어 있으며, 차량이 출고되기 전 차량에 탑재된 모든 ECU는 Gateway ECU에 자신의 인증서를 등록한다. 등록과정에서 필요한 외부통신은 SSL(Secure Socket Layer), TLS(Transport Layer Security)와 같은 안전한



〔그림 3〕 차량의 CAN 환경

채널을 통해 이뤄진다. 차량 시동 시, Gateway ECU는 ECU간의 안전한 통신을 위해 사용될 비밀값을 생성하여 각 ECU에게 안전하게 전송한다. 전송받은 비밀값을 통해 각 ECU는 암호화 세션키, 메시지 인증키, 키 업데이트키를 생성한다. 생성된 키들을 통해 ECU는 Data 프레임 내의 Data 필드를 암호화한 후 메시지 인증코드를 생성하고 이를 Data 프레임에 조합하여 CAN 버스를 통해 타 ECU에 송신한다. 이 때 공격자의 재전송 공격을 방지하기 위해, 메시지 인증 코드 생성 시 카운터(Counter) 값을 메시지 인증키와 함께 사용한다. 각 ECU는 자신의 저장 공간에 자신의 ID와 자신이 메시지를 수신 받는 다른 ECU ID에 해당하는 카운터 값들을 저장한다. 또한 키의 안전성을 보장하기 위해 Gateway ECU는 일정 주기마다 키 업데이트 과정을 수행한다. 제안하는 메커니즘은 〔표 2〕와 같은 표기법을 사용한다.

〔표 2〕 제안하는 메커니즘에서 사용하는 표기법

ECU_{A_i}	i번째 ECU
ECU_G	Gateway ECU
$PK^E_{A_i}, SK^E_{A_i}$	i번째 ECU의 암호화용 공개키, 비밀키
$PK^S_{A_i}, SK^S_{A_i}$	i번째 ECU의 서명용 공개키, 비밀키
$Cert^E_{A_i}, Cert^S_{A_i}$	i번째 ECU의 암호화용 기기 인증서, 서명용 기기 인증서
S_i	i번째 세션의 세션키를 생성하기 위한 비밀값
EK_{S_i}	i번째 세션의 암호화 세션키
AK_{S_i}	i번째 세션의 인증키
KEK_{S_i}	i번째 세션의 키 업데이트 키
CTR_{A_i}	i번째 ECU의 메시지 카운터 값

4.1.2 보안요구사항

안전한 CAN 메시지 인증 프로토콜은 다음의 보안 요구사항을 만족하여야 한다. 먼저 악의적인 공격자의 도청을 통한 메시지의 수집 및 분석을 방지하기 위해 CAN Data 프레임의 기밀성을 제공해야만 한다. 또한 Data 프레임의 무결성을 제공하여 위/변조 여부를 파악할 수 있어야 하고, 추가적으로 기 수집된 Data 프레임을 이용한 리플레이(Replay) 공격을 방지할 수 있어야만 한다. 악의적인 노드에 의해 발생할 수 있는 비정상적인 트래픽을 이용한 CAN에 대한 DoS(Denial of Service) 공격은 침입 탐지 시스템과 같은 방법을 이용하여 이상 징후를 탐지한 후, 사후 대처를 해야 한다. 하지만 본 논문에서는 사후 대처방법인 침입탐지시스템을 다루지 않는다. 〔표 3〕은 CAN 메시지 인증 프로토콜의 보안 요구사항을 나타낸다.

4.2 인증서 관리

앞서 제시한 보안 요구사항과 차량용 CAN의 제약사항을 만족시키기 위해서 대칭키를 사용하는 CAN 메시지 인증 프로토콜이 설계되어야 한다. 이를 위하여 각 ECU들과 Gateway ECU는 비밀키를 공유하여야만 한다. 본 논문에서 제안하는 메커니즘에서는 안전한 비밀키 공유를 위하여 ECU의 기기 인증서를 사용한다.

차량에 장착되는 ECU는 공인된 인증기관(Certification Authority)의 인증을 받은 제조사에 의해 제작되며, 제작 시 각 ECU 및 Gateway ECU에 서명용 기기 인증서($Cert^E_{A_i}$)와, 암호화용 기기 인증서($Cert^S_{A_i}$)를 탑재한다. 이후 차량이 제작업체에서 출고되거나, 또는 차량에 탑재된 ECU를 변경할 때 기기 인증서 등록절차를 수행한다. 기기 인증서의 탑재과정과 등록과정은 일반적인 IT시스템에서의 과정과 같이 안전한 채널을 통해 이루어진다. 본 논문의 주안점은 앞 절에서 제시한 제약사항을 고려한 CAN 메시지 인

〔표 3〕 CAN 메시지 인증 프로토콜의 보안 요구사항 및 설계 요구사항

보안 요구사항	데이터 기밀성	도청을 통해 데이터를 분석할 수 없어야 한다.
	데이터 무결성	데이터의 위/변조를 감지 할 수 있어야 한다.
	데이터 인증	정상적인 데이터를 생성하여 전송할 수 있다.
	재전송 공격 방지	리플레이 공격을 방지할 수 있어야 한다.
설계 요구사항	실시간 처리	데이터 처리가 실시간적으로 수행되어야 한다.

증 프로토콜에 있기 때문에 기기 인증서의 탑재, 등록 그리고 관리에 대한 일반적인 내용은 생략한다.

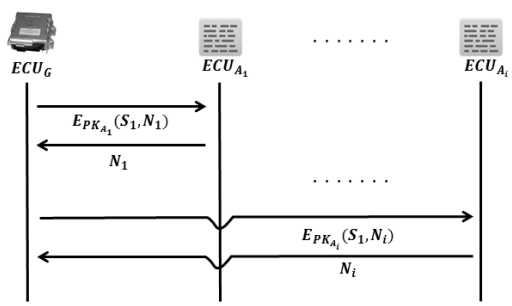
4.3 키 분배 및 키 업데이트

제안하는 CAN 메시지 인증 프로토콜을 위해서는 차량 환경에 적합한 키 관리 기술이 필요하다. 제시하는 키 관리 기술은 크게 차량의 ① 최초 시동 시에 수행되는 키 분배 과정과, ② 안전도를 높이기 위한 키 업데이트 과정으로 나눌 수 있다.

4.3.1 키 분배

차량의 시동 시, ECU_G는 n개의 ECU가 있는 환경에서 키 분배를 위한 〔그림 4〕와 같은 과정을 수행한다. 키 분배의 자세한 과정은 다음과 같다.

① ECU_G는 비밀값 S_1과 각 ECU의 키 전송을 확인하기 위한 n개의 Nonce 값 $N_1,\cdots,N_n$를 생성한다.
② 생성한 S_1과 N_i를 각 ECU_{A_i}의 공개키 PK_{A_i}로 암호화하여 전송한다. ECU_G는 키 확인을 위해 각 ECU에게 보낸 Nonce값을 테이블로 관리한다.
③ ECU_{A_i}는 전송받은 메시지를 자신의 비밀키 SK_{A_i}로 복호화하여 비밀값 S_1와 N_i를 얻는다.
④ ECU_{A_i}는 전송받은 비밀값 S_1을 S_1',S_1'',S_1'''으로 나눈 뒤, SHA-1 알고리즘과 같은 KDF(Key Derivation Function)를 통해 암호화를 위한 세션키 EK_{S_1}과, 메시지 인증 코드를 생성하기 위한 메시지 인증키 AK_{S_1}, 그리고 키 업데이트를 위한 키 업데이트 키 KEK_{S_1}을 생성한다.

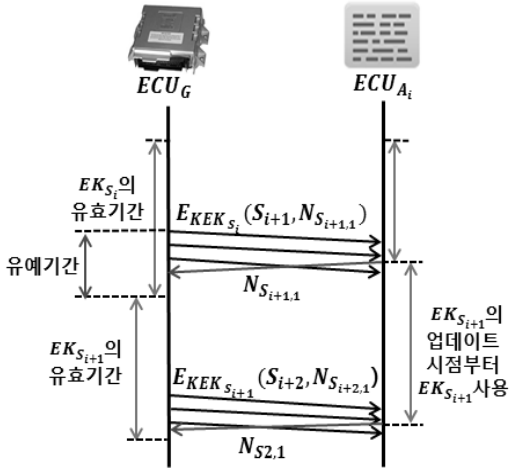


〔그림 4〕 키 분배 과정

⑤ ECU_{A_i} 키 생성 확인을 위해 전송받은 N_i를 ECU_G에 전송한다.
⑥ ECU_G는 일정시간 후 키 확인 메시지를 전송하지 않은 ECU에 대해 암호화된 첫 번째 세션의 비밀값과 Nonce값을 재전송한다.
⑦ 각 ECU는 카운터 값 CTR_{A_i}를 0으로 초기화 한다.
⑧ 모든 ECU의 키 전송이 확인된 후 정상적인 CAN 통신을 시작한다.

4.3.2 키 업데이트

키가 노출 될 경우의 안전성을 보장하기 위하여 〔그림 5〕와 같은 일정한 주기를 갖는 키 업데이트가 수행된다. 키 업데이트는 다음과 같이 수행된다. 차량 시동 이후 일정 주기가 지나면 ECU_G는 업데이트할 새로운 세션키를 위한 비밀값을 생성한다. 이 때 최초 키 분배 과정과 마찬가지로 각 ECU의 키 전송 확인을 위해 n개의 Nonce값을 함께 생성한다. 예를 들어, i번째 세션이 끝나고 i+1번째 세션이 시작된다면, ECU_G 생성된 비밀값과 Nonce값을 이전 세션의 키 업데이트 키(KEK_{S_i})로 암호화 되어 각 ECU에 전송한다. i+1번째 비밀값을 전송받은 각 ECU는 최초 시동 시의 키 분배과정과 동일한 과정을 수행하여 새로운 세션키 $EK_{S_{i+1}}$, $AK_{S_{i+1}}$, $KEK_{S_{i+1}}$를 생성하고, ECU_G에게 Nonce값을 전송한다. ECU_G는 키 전송이 확인되지 않은 ECU에게 키 업데이트 메시지를 재전송 한다. 키 업데이트가 수행할 때 기 정의된 유예 기



〔그림 5〕 키 업데이트 과정

Num	Time	ID	DATA
3281	1287.064	0080	00 17 4E 18 1F 17 1F 14
3306	1297.048	0080	00 17 4E 18 1F 17 1E 15
3335	1307.117	0080	00 17 5C 18 1F 17 1E 16
3357	1316.76	0080	00 17 58 18 1F 17 1D 57
3384	1327.128	0080	00 17 32 18 1E 18 1D C8
3409	1336.813	0080	00 17 24 18 1E 18 1C B9
3436	1347.095	0080	00 17 10 18 1C 18 1C 1A
3462	1357.25	0080	00 17 10 18 1C 18 1B 1B
3487	1367.191	0080	00 17 FC 17 1C 18 1B 7C

Num	Time	ID	DATA
351	137.783	018F	00 2B 20 00 00 5C 00 00
375	147.169	018F	00 2B 21 00 00 5C 00 00
400	157.196	018F	00 2B 22 00 00 5C 00 00
425	166.753	018F	00 2B 22 00 00 6C 00 00
453	178.017	018F	00 2B 23 00 00 6C 00 00
476	186.977	018F	00 2B 23 00 00 5C 00 00
504	198.198	018F	00 2B 25 00 00 5C 00 00
529	207.2	018F	00 2B 25 00 00 5C 00 00
554	217.91	018F	00 2B 26 00 00 5C 00 00

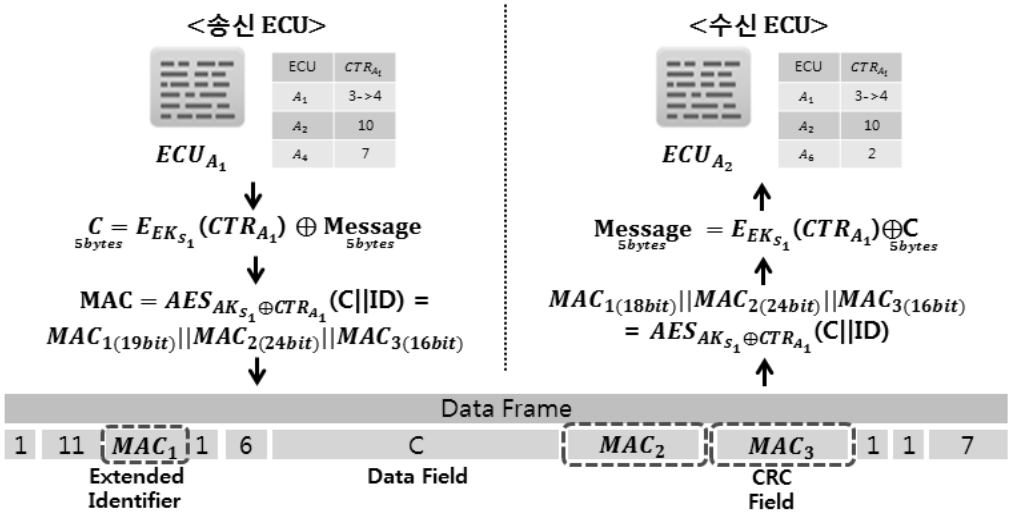
〔그림 6〕 실제 차량의 CAN 메시지 패킷

간(Grace Time)동안 이전 세션키와 새로운 세션키가 CAN 통신에 사용된다[9].

4.4 제안하는 CAN 메시지 인증 프로토콜

〔그림 6〕에서 볼 수 있듯이 실체 차량의 CAN Data 프레임 내의 Data 필드 값 8byte 중 3~4 byte는 "00"과 "FF" 같은 값이나 데이터의 카운터 정보와 같은 값들로 채워져 있다. 따라서 제안하는 CAN 메시지 인증 프로토콜에서는 Data 필드 중 3byte를 키의 안전성을 확보하기 위한 공간으로 차용한다. 또한 차량 환경에서 실제로 사용하지 않는 Extended Identifier 필드 19bit와 CRC 필드 16bit도 키의 안전성을 확보하기 위한 공간으로 차용한다. 제안하는 CAN 메시지 인증 프로토콜에서는 차용된 공간 59bit(3byte + 19bit + 16bit)를 메시지의 무결성과 인증을 위한 메시지 인증 코드를 위한 공간으로 사용한다. CRC 필드의 메시지 오류 판별은 기능은 메시지 인증코드의 기능에 포함되어 있다[5].

제안하는 CAN 메시지 인증 프로토콜은 공격자의 도청과, 개인 정보의 유출을 방지하기 위하여 AES-CTR모드를 이용하여 메시지를 암호화 한다[10]. 암호화의 대상은 CAN의 Data 필드 8byte중 직접적인 패킷 정보를 포함하고 있는 5byte(40bit)로 한정한다. Data 필드의 나머지 3byte는 메시지 인증코드를 위한 공간으로 사용한다. 만약 Identifier 필드를 포함하여 Data 프레임이 암호화 될 경우에 각 ECU는 메시지 수신 여부를 결정하기 위해 CAN 버스상의 모든 Data 프레임을 복호화 하여야 하므로 제안하는 프로토콜에서는 Identifier 필드는



〔그림 7〕 제안하는 CAN 인증 프로토콜

암호화 하지 않고, Data 필드의 5byte만 암호화 한다. 상세한 ECU간의 통신과정은 아래와 같다.

① Data 프레임을 생성하는 ECU_{A_i}는 (식1)과 같이 카운터 값 CTR_{A_i}을 세션키 EK_{S_i}로 암호화한 후, 그 값 중 5byte와 XOR연산을 하여 5byte의 암호문 C를 생성한다.

② (식)2와 같이 암호문 C와 Identifier 필드를 입력 값으로, $AK_{S_i} \oplus CTR_{A_i}$을 키 값으로 하는 AES 알고리즘을 통해 메시지 인증코드를 생성한다.

③ 생성한 메시지 인증코드를 3등분 하여 Data 프레임에 〔그림 7〕과 같이 삽입한 후 CAN 버스를 통해 전송한다. 전송 후에 ECU_{A_i}는 버퍼(Buffer)의 카운터 값 CTR_{A_i}을 1 증가 시킨다.

④ 각 수신 ECU는 CAN 버스상의 메시지의 ID 값을 기준으로, 메시지 수신 여부를 결정한다.

⑤ 수신 ECU는 (식3)과 같이 자신의 저장 공간에 저장되어있는 송신 ECU의 카운터 값 CTR_{A_i}를 이용하여 메시지 인증코드를 확인한 후, 다시 카운터 값 CTR_{A_i} 를 이용해 메시지를 복호화한다.

⑥ 메시지를 복호화하여 처리한 후, 수신 ECU는 자신의 저장 공간에서 CTR_{A_i}를 1 증가시킨다.

$$C = E_{EK_{S_i}}(CTR_{A_i}) \oplus Message \quad \text{(식1)}$$

$$MAC = AES_{AK_{S_i} \oplus CTR_{A_i}}(C) \quad \text{(식2)}$$

$$AES_{AK_{S_i} \oplus CTR_{A_i}}(C) \stackrel{?}{=} MAC, \quad \text{(식3)}$$

$$Message = E_{EK_{S_i}}(CTR_{A_i}) \oplus C$$

V. 분석

본 장에서는 제안하는 보안 메커니즘에 대한 보안성 분석과 성능 분석을 제시한다.

5.1 보안성 분석

• 메시지 기밀성(Message Confidentiality) : ECU간의 통신 시 Data 필드 중 5byte의 메시지는 해당 시점의 암호화 세션키 EK_{S_i}를 통해 AES-CTR 알고리즘으로 암호화 된다. 암호화에 사용되는 세션키 EK_{S_i}를 생성하기 위한 비밀값 S_i는 최초에 각 ECU의 공개키로 암호화되고, 이 후 키 업데이트 키 KEK_{S_i}를 통해 암호화 되어 전송되기 때문에 외부에 노출되지 않는다. 따라서 세션키가 없는 공격자는 도청을 통해 실질적인 메시지의 의미를 분석하기 어렵다. 또한 수집된 메시지를 통한 전수조사를 수행하여 세션키를 알아내기 위해서는 $2^{126.1}$의 연산시간이 필요하므로 Real time 안에서의 공격이 불가능 하다〔12〕

• 메시지 무결성(Message Integrity)과 메시지 인증(Message Authentication): 제안하는 프로토콜은 메시지의 무결성을 보장하기 위해, AES-128 알고리즘을 통해 59bit의 메시지 인증 코드를 사용한다. 메시지 인증 코드를 통해 ECU는 메시지의 위/변조를 감지할 수 있다. 정상적으로 인증서 등록과정을 거치지 않은 ECU는 인증키를 알 수 없으므로, 2^{59}개의 경우의 수를 갖는 메시지 인증코드에 대해 공격자가 특정 암호문에 대한 메시지 인증코드의 충돌 쌍을 생성할 확률은 $1/2^{29.5}$이다. 차량환경을 고려하여 1ms의 빈도로 충돌 쌍을 찾기 위한 전수공격을 수행하려면 최대 8.78일의 시간이 소요된다. 따라서 제안하는 CAN 메시지 인증 프로토콜에서 8일 미만의 주기를 갖는 키 업데이트를 수행할 경우, 공격자는 업데이트 주기 내에 메시지 인증코드를 생성할 수 없기 때문에 다른 ECU와 정상적인 통신을 수행할 수 없다.

• 재전송 공격 방지(Resistant to Replay-attack) : AES-CTR 알고리즘은 메시지 생성을 위해 카운터 값이 사용되므로, 동일 세션 내의 동일한 메시지에 대해 다른 암호문이 생성되고, 다른 메시지 인증코드를 생성한다. 또한 메시지 인증 코드를 생성하기 위해 사용되는 인증키(AK_{S_i}) 역시 주기적으로 업데이트되기 때문에, 수집된 메시지를 통한 재전송 공격은 불가능하다.

• 메시지의 실시간 처리(Message Processing in Realtime) : 제안하는 CAN 메시지 인증 프로토콜은 송신할 메시지의 암호화와 메시지 인증코드의 생성에 1.9ms, 수신된 메시지의 메시지 인증코드의 확인과 메시지 복호화에 1.9ms의 시간이 소요된다. 실제 판매중인 차량의 주행 실험을 통해 분석한 차량 운행 중 메시지 최대 송신빈도는 10ms이므로 제안하는 CAN 메시지 인증 프로토콜은 메시지의 실시간 처리를 보장한다. 이에 대한 자세한 분석은 다음 절의 성능분석에서 제시한다.

〔표 4〕 제안하는 메커니즘과 다른 연구의 비교

구분	CAN Protocol	〔5〕 D. K. Nilsson and U. E. Larson,	제안하는 CAN 메시지 인증 프로토콜
공격자의 도청에 대한 메시지의 기밀성	X	X	O
메시지의 무결성	O	O	O
데이터 인증	X	O	O
재전송 공격 방지	X	X	O
메시지의 실시간 처리	O	X	O

5.2 성능 분석

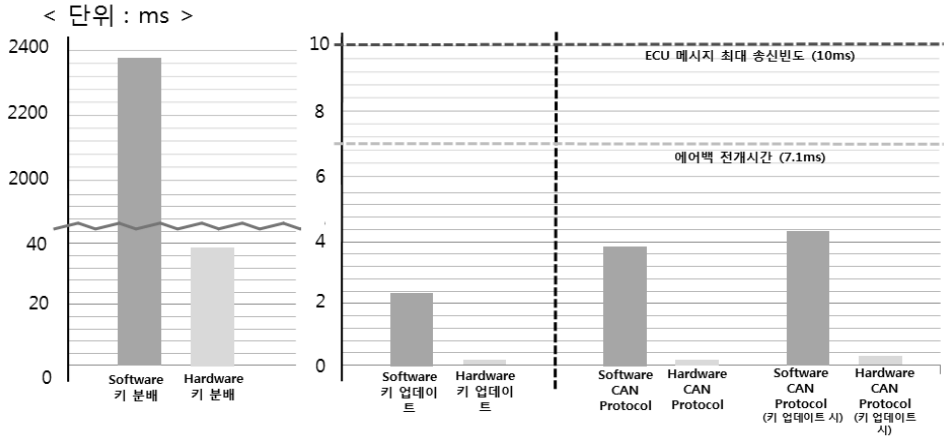
본 절에서는 제안하는 보안 메커니즘에 대해 기존 선행연구들의 성능 분석 수치를 인용하여 소프트웨어 기반의 성능 분석과 하드웨어 기반의 성능 분석을 제시한다. 최근 개발되고 있는 차량용 ECU는 평균적으로 200Mhz의 클락 스피드와 256Kbyte의 저장능력을 갖고 있다〔13〕. 소프트웨어 기반의 성능 분석에서 인용한 선행연구의 성능 분석 수치는 차량용 ECU와 유사한 프로세서를 기반으로 측정되었으며〔14,15,16,17〕, 하드웨어 기반의 성능분석은 eSTREAM project의 연구를 기준으로 하였다〔18〕.

5.2.1 소프트웨어(Software) 기반의 성능 분석

• 키 분배 : ECU_G는 각 ECU의 공개키로 비밀값을 암호화 하여야 한다. 200Mhz의 클락 스피드를 갖는 Gateway ECU는 46.06ms의 ECIES(Elliptic Curve Integrated Encryption Scheme) 암호화 수행 시간을 갖는다〔17〕. 차량에 장착된 ECU가 50개라고 가정하였을 때 각 ECU에게 비밀값을 전송하기 위해서는 약 2.3초가 소요된다. 또한 각 ECU는 전송된 비밀값을 복호화기 위해서 29.13ms의 시간이 소요된다〔17〕. 세션키를 생성하기 위한 알고리즘으로 SHA-1을 사용하였을 때 필요한 연산시간은 각 세션키 당 0.1ms씩 총 0.3ms가 소요된다〔18〕. 메시지 송/수신에 필요한 시간은 ECIES 암/복호화 속도보다 빠르고, 연산과정 중에 수행할 수 있으므로 이를 고려하지 않았을 때, 최초 시동 시 3초 이내의 시간에 키 분배가 완료된다.

• 키 업데이트 : 키 업데이트를 위해 Gateway ECU와 일반적인 ECU는 각 1회의 AES 알고리즘 연산을 수행한다. 1회의 암/복호화 연산의 소요시간은 0.961ms 이므로〔15,16〕, 메시지의 최대 송신 빈도인 10ms 보다 적다.

• CAN 메시지 인증 프로토콜 : 메시지 프레임 생성을 위해 메시지를 송신할 ECU는 메시지의 암호화와 메시지 인증코드의 생성을 위해 각각 1회의 AES-CTR 알고리즘의 수행과, AES 알고리즘의 수행이 필요하다. AES-CTR 알고리즘의 경우 병렬처



〔그림 8〕 제안하는 메커니즘의 성능 분석

리가 가능하므로, 실제 암호화를 위해서 1회의 XOR 연산이 수행된다. 따라서 송신 ECU가 메시지를 암호화하여 메시지 인증코드를 생성하기 위해 1.92ms의 시간이 소요되며, 수신 ECU가 메시지 인증코드를 확인한 후 메시지를 복호하기까지 1.92ms의 시간이 소요된다. 키 업데이트 과정 중의 수신 ECU는 i번째 세션의 인증키와 i+1번째 인증키로 메시지 인증코드를 생성하여 검증하여야 하기 때문에 추가적으로 0.96ms가 더 소요되어, 총 4.80ms의 시간이 소요된다.

차량의 기능 중 가장 실시간 적인 동작이 일어나야 하는 에어백은 탑승자의 안전을 위해 차량충돌 후 7.1ms 전개되어야 한다[19]. 제안하는 CAN 인증 프로토콜은 메시지의 암/복호화와 메시지 인증코드의 생성과 확인에 약 3.84ms가 소요되고, 키 업데이트 과정 중에는 약 4.80ms가 소요되므로, 실시간적인 차량 구동을 보장한다. 또한 향후 차량 기술의 발전으로 100개 이상의 ECU가 차량에 탑재되더라도, 브로드캐스트 통신방식을 사용하는 CAN의 특성상 송신측 ECU의 송신주기는 영향을 받지 않으며, 수신측 ECU의 경우 CAN 버스상의 메시지 중 자신이 수신할 ID를 필터링(Filtering)하여 수신하므로 ECU의 개수는 메시지 인증코드 확인 및 복호화 해야 하는 메시지 수에 크게 영향을 주지 않는다. 따라서 ECU의 개수가 증가하더라도 7.1ms 이내에 메시지 인증코드의 확인과 메시지 복호화를 수행할 수 있다.

5.2.2 하드웨어(Hardware) 기반의 성능 분석

- 키 분배 : 하드웨어 기반에서의 공개키를 통한 비밀값의 암/복호화는 1회에 0.75ms가 소요된다. 차량에 장착된 ECU가 50개 일 때, ECU_G는 총 37.5ms의 암호화 연산을 수행한다[20]. 각 ECU마다 비밀값의 복호화에 0.75ms의 시간이 소요되며, 키 생성을 위한 SHA-1 알고리즘의 연산은 1회당 0.001ms씩 총 0.003ms의 연산시간이 소요된다[21]. 따라서 제안하는 메커니즘에서의 키 분배는 총 38.25ms의 시간이 소요된다.
- 키 업데이트 : 키 업데이트를 위한 AES 알고리즘의 암/복호화 연산시간은 0.071ms이다[16]. 따라서 모든 ECU는 1ms미만의 시간에 키 업데이트를 완료할 수 있다.
- CAN 인증 프로토콜 : 메시지 생성을 위해 1번의 AES-CTR 알고리즘의 수행과 1번의 AES 알고리즘의 수행이 필요하다. AES-CTR알고리즘과, AES 알고리즘 연산시간이 0.071ms 이므로[16], 송신측 ECU가 메시지를 암호화 하고 메시지 생성코드를 생성하는데 0.142ms가 소요되며, 수신측 ECU에서 메시지 인증코드의 확인 및 메시지 복호화에 역시 0.142ms가 소요된다. 키 업데이트 과정 중에는 추가적으로 0.071ms가 더 소요되어 총 0.213ms가 소요된다.

VI. 결론

본 논문에서는 차량용 CAN의 취약점을 이용한 공격시나리오를 제시하고, 해당 취약점을 보완할 수 있는 안전한 차량 내부 통신 네트워크 메커니즘을 제안하였다. 제안하는 메커니즘은 차량 ECU의 인증서를 통해 CAN 버스의 비밀값을 안전하게 각 ECU에 전송하고, 해당 비밀값을 이용하여 세션키와 인증키 그리고 키 업데이트 키를 생성하여 기밀성과, 무결성을 보장하는 ECU간의 안전한 통신을 제공한다. 또한 제안하는 메커니즘은 정상적으로 인증을 거쳐 기기 인증서를 등록한 ECU만이 인증키를 생성하여 통신을 할 수 있고, 카운터를 이용하여 메시지 재전송 공격을 방지한다.

현재까지 제안된 차량 내부 네트워크의 보안 메커니즘은 대부분 공격에 대한 침입 탐지와 같은 수동적인 보안을 제공한다. 하지만 제안하는 메커니즘은 차량 내부 네트워크의 가용성을 만족하면서도 도청이나 재전송 공격을 사전에 차단할 수 있는 능동적인 보안을 제공한다. 향후 연구에서는 제안하는 메커니즘에 대하여 임베디드 보드를 통한 CAN환경에서의 시뮬레이션 또는 CAN 전용 시뮬레이션 소프트웨어인 CANoe를 통한 시뮬레이션이 필요하며, 최종적으로는 실제 차량환경에서의 시뮬레이션이 필요하다.

참고문헌

[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, and T. Kohno, "Experimental security analysis of a modern automobile," Proceedings of the 2010 IEEE Symposium on Security and Privacy, pp. 447-462, May. 2010.

[2] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the art : embedding

security in vehicles," EURASIP Journal on Embedded Systems, vol. 2007, pp 16, Jun. 2007

〔3〕 Sato Michicho, *자동차 네트워크 시스템*, 성인당, Jan 2010.

〔4〕 T. Hoppe, S. Kiltz, and J. Dittmann. "Security threats to automotive CAN networks - practical examples and selected short-term countermeasures," Proceeding of the 27th Internatioanl Conference on Computer Safety, Reliability, and Security(SAFECOM '08), pp. 235-248, Sep. 2008.

〔5〕 D. K. Nilsson and U. E. Larson, "A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure," Journal of Networks, vol. 4, no. 7, pp. 552 - 564, Sep. 2009.

〔6〕 김강석, "CAN 통신 도청 및 조작을 통한 차량 ECU의 외부위협 가능성 분석," 석사학위논문, 고려대학교, 2011년 2월.

〔7〕 S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," ACM Transactions on Embedded Computing Systems. vol. 3, no. 3, pp. 461 - 491, Aug. 2004.

〔8〕 D. K. Nilsson and U. E. Larson, "Simulated Attacks on CAN Buses: Vehicle virus," Proceedings of the Fifth IASTED Asian Conference on Communication Systems and Networks (ASIACSN), pp. 66-72, Aug. 2008.

〔9〕 IEEE, "IEEE Std 802.16-2009," IEEE, May. 2009.

〔10〕 M. Dwokin, "Recommendation for block cipher modes of operation method and techniques," U.S. DoC/NIST, Dec. 2001.

〔11〕 FIPS Publication 197, "Advanced Encryption Standard (AES)." U.S. DoC/NIST, Nov, 2001.

〔12〕 A. Bogdanov, D. Khovratovich, and C. Rechberger. "Biclique cryptanalysis of the full AES," ASIACRYPT 2011, LCNS 7073, pp. 344-371, 2011.

〔13〕 Texas Instruments, "http://ti.com/lsds/ti/microcontroller/home.page

〔14〕 D. E. Boyle and T. Newe, "On the implementation and evaluation of an elliptic curve based cryptosystem for java enabled wireless sensor networks," Sensors and Actuators A: Physical, vol. 156, issue 2, pp.394-405, Dec 2009.

〔15〕 J. Grobschadl , S. Tillich, C. Rechberger, M. Hofmann, and Marcel Medwed, "Energy Evaluation of Software Implementations of Block Ciphers under Memory Constraints," Conference on Design, automation and test 2007, p. 1110 - 1115. Apr 2007.

〔16〕 A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks", IPSN 2008 : Proceedings of the 2008 International Conference on Information Processing in Sensor Networks, pp. 245 - 256. Apr 2008.

〔17〕 P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu. "Analyzing and modeling encryption overhead for sensor network nodes." Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 151-159, Sep 2003.

〔18〕 T. Good and M. Benaissa, "Hardware performance of eSTREAM phase-iii stream cipher candidates," State of the Art of Stream Ciphers Workshop (SASC 2008), pp. 163-173, Feb 2008.

〔19〕 서주형, 최규흠, 유재민, 오주환, "에어백 전개시간에 따른 승객 보호 성능 연구," 한국 자동차 공학회 2006년도 춘계학술대회 논문집, pp. 1199-1205, 2006년 3월.

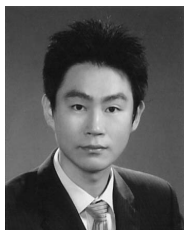
〔20〕 J. Portilla, A. Otero, E. de la Torre, T. Riesgo, O. Stecklina, S. Peter, and P. Langendorfer, "Adaptable security in wireless sensor networks by using

reconfigurable ecc hardware coprocessors," International Journal of Distributed Sensor Networks, Vol. 2010, Oct 2010.
[21] Grembowski. T, Lien. R, Gaj. K, Nguyen. N, Bellows, P, Flidr. J, Lehman. T, and Schott. B, "Comparative analysis of the hardware implementations of hash functions SHA-1 and SHA-512," ISC 2002, LNCS 2433, pp. 75-89, 2002.

〈著者紹介〉

조 아 람 (A-Ram Cho) 학생회원
2011년 2월: 고려대학교 산업시스템 정보공학과 학사
2011년 3월~현재: 고려대학교 정보보호대학원 석사
〈관심분야〉 In-vehicle Security, 스마트 그리드 보안

조 효 진 (Hyo Jin Jo) 학생회원
2009년 2월: 고려대학교 산업시스템 정보공학과 학사
2009년 3월~현재: 고려대학교 정보보호대학원 석·박사 통합과정
〈관심분야〉 VANET, In-vehicle Security, Secure Roaming



우 사 무 엘 (Samuel Woo) 학생회원
2006년: 단국대학교 컴퓨터과학과 졸업 학사
2006년: (주)EOTECHNICS 근무
2010년: 단국대학교 전자계산학 석사
2011년~현재: 고려대학교 정보보호대학원 박사과정
관심분야: 무선 네트워크 보안, IN-Vehicle Security

손 영 동 (Young Dong Son) 정회원
1986년: 한국경제신문 정보통신전문기자
2003년: KTH 상무이사/상임감사
2008년: 국가보안기술연구소 소장
2011년: 숭실대학교 IT정책경영학 박사
2011년~현재: 고려대학교 정보보호대학원 초빙교수
〈관심분야〉 사이버테러, 사이버전, 사이버심리전

이 동 훈 (Dong Hoon Lee) 종신회원
1983년 8월: 고려대학교 경제학사 졸업
1987년 12월: Oklahoma University 전산학과 석사 졸업
1992년 5월: Oklahoma University 전산학과 박사 졸업
1993년 3월~1997년 2월: 고려대학교 전산학과 조교수
1997년 3월~2001년 2월: 고려대학교 전산학과 부교수
2001년 3월~현재: 교려대학교 정보보호대학원 교수
〈관심분야〉 암호프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET기술