CAN을 위한 해시체인 기반 메시지 인증 및 키 분배 기법 설계

*박진혁, *,**김현성
*경일대학교 사이버보안학과
**말라위대학교 수학과
e-mail : *vkqxkr@naver.com, **(교신저자)kim@kiu.ac.kr

Design of Hash Chain based Message Authentication and Key Distribution Scheme for CAN

*Jinhyuck Park, *,**Hyunsung Kim
*Dept. of Cyber Security, Kyungil University
**Dept. of Mathematical Sciences, University of Malawi

Abstract

Modern cars contain a multiplicity of electronic control units (ECUs), which could communicate between ECUs by using controller area network (CAN). The objective of this paper is to provide secure communication over CAN by providing a secure message authentication and key distribution scheme. This paper will propose a hash chain based message authentication key distribution scheme for CAN, which could provide confidentiality and integrity of data packets.

I. 서론

최신 자동차는 다양한 특성을 가진 통신 시스템에 의해 점점 네트워크화 되고 있는 다수의 전자제어장치(Electronic Control Unit, ECU)를 포함한다 [1]. ECU 장치가 있는 자동차는 제어 영역 네트워크(Controller Area Network, CAN), 로컬 상호 연결 네트워크(Local Interconnect Network, LIN), FlexRay, 미디어 지향 시스템 전송(Media Oriented Systems Transport, MOST)등을 사용하여 통신할 수 있다. CAN 통신은 국제표준화기구(International Standard Organization, ISO) 11898와 자동차기술 협회(Society of Automotive Engineer)에 의해 표준화된 차량 기술의 중요한 역할을 담당하고 있다.

일반적으로 가장 큰 규모의 ECU는 엔진 컨트롤 장치이다. 다른 ECU장치들은 변속, 에어백, ABS, 쿠르즈컨트롤, 배터리와 충전시스템 등에 사용된다. 다양한 ECU들은 독립된 서브시스템 형태로 동작한다. 이를 위한 통신 기능은 ECU의 기본적인 기능이다. CAN 통신이 기술적 간섭에 안전성을 보장한다고 하더라도 대부분의 악의적 공격에 취약하다. 그러므로 CAN 통신의 증가는 어플리케이션의 안전성에 지대한 영향을 끼칠 수 있다.

최근 CAN에 대한 다양한 위협이 보고되고 보안 기법을 제공하기 위한 연구가 진행되고 있다 [2-7]. Hoppe등은 CAN공격에 대한 패턴 매칭 기반 탐지 기법을 제안하였다 [2]. 이 기법은 메시지 식별자 필드와 전자적 특징을 이용하지 않는 CAN버스 상의 데이터 프레임을 이용한다. Wolf등은 디지털 서명과 인증서를 사용하는 안전한 ECU 소프트웨어 업데이트 기법을 제안하였다 [3]. 이 기법은 합법적인 소프트웨어 업데이트로 가장하는 공격에 초점을 두었다. 그러나 Wolf등의 기법은 공격이 발생한 후 치료에 초점을 맞추고 있어서 공격으로부터 보호될 수는 없다. CAN 메시지 보

안을 위해, Nilsson등은 4개의 메시지로 구성된 데이터 프레임에서 메시지 인증 코드 (Message Authentication Code, MAC)를 이용하는 기법을 제안하였다 [4]. 이 기법은 메시지를 생성하는데 요구되는 지연 시간 부하로 인해 차량 통신에서 요구되는 실시간 성을 지원하지 못한다. Ravi등은 무결성, 인증, 가용성 및 부인 방지에 초점을 맞춘 CAN 취약성 분석을 제시하였다 [5]. Nilsson등은 read, spoof, drop, modify, flood, steal, replay를 포함한 일반적인 공격이 CAN에 가능함을 논리적 및 물리적으로 제시했다 [6].

최근, Cho는 데이터 패킷의 기밀성과 무결성을 보장하고 재생공격에 CAN 통신을 안전하게 보호하는 차량 네크워크 통신을 위한 인증 기법을 제안하였다 [7]. Cho의 기법은 공개키 인증서를 기반으로 각 네트워크 참가자가 각각 2개의 인증서를 이용해야 하는 문제가 있고, 높은 통신 부하가 제시되는 문제가 있다.

본 논문에서는 CAN의 인증 기법 관련된 다양한 부하를 해결하고 안전성을 해결하기 위한 해시체인 기반 메시지 인증 및 키 분배 기법을 제안한다. 제안된 기법은 참여자를 위한 인증서, 세션 키 분배를 위한 해시체인을 이용하기 때문에 기존의 기법 보다 통신 및 연산 부하들을 줄일 수 있다. 제안된 기법은 CAN통신에 필요한 보안 측면을 제공하고, 다양한 공격에 안전하다.

II. CAN 개요

CAN은 ECU를 연결하는 다중 마스터 직렬 버스 규격이며, 그림 1과 같은 네트워크를 구성한다.

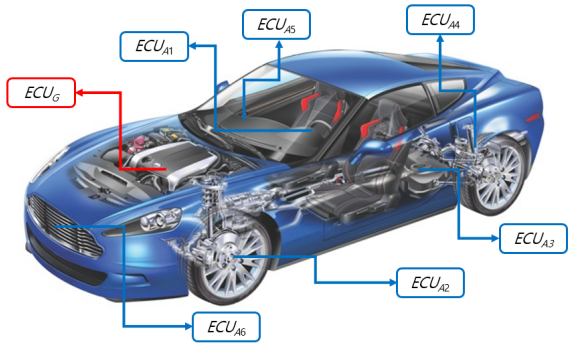

그림 1. CAN 구성도

CAN에서 통신하려면 2개 또는 그 이상의 노드가 필요하다. 노드는 간단한 I/O장치에서 CAN인터페이스를 가진 임베디드 컴퓨터와 정교한 소프트웨어에 이르기까지 다양하다. ECU는 표준컴퓨터가 이더넷 포트의 USB를 통해 CAN 상의 장치와 통신할 수 있도록 연결한다.

Koscer등은 CAN의 약점을 세 가지 측면에서 분석했다. CAN 메시지는 브로드 캐스트 기법을 이용하기 때문에 도청공격에 취약하다. 본질적으로 데이터 프레임은 ECU 인증에 어떠한 필드도 제공하지 않는다.

III. 제안된 기법

본 장은 CAN을 위한 해시체인 기반메시지 인증 및 키 분배 기법을 제안한다. Table 1은 기법에서 이용하는 기호의 정의를 보여준다.

Table 1. 기호 정의

기 호	정 의
ECU_{Ai}	i번째 ECU
ECU_G	게이트웨이 ECU
PK_{Ai}, SK_{Ai}	ECU_{Ai}의 공개키와 개인키
PK_G, SK_G	ECU_G의 공개키와 개인키
$Cert_{Ai}$	ECU_{Ai}의 인증서
$Cert_G$	ECU_G의 인증서
$Cert_{CA}$	인증서버의 인증서
Sig_{Ai}	ECU_{Ai} 개인키를 사용한 서명 값
Sig_G	ECU_G 개인키를 사용한 서명 값
S_i	i번째 세션의 비밀키
SK_{Si}	i번째 세션의 세션키
AK_{Si}	i번째 세션의 인증키
KEK_{Si}	i번째 세션키의 업데이트키
CTR_{Ai}	ECU_{Ai}의 메시지 카운터
$E_K()$	키 K를 이용한 대칭키 암호
$h()$	일방향 해시함수
$\|\|$	문자열 결합 연산자

제안된 기법은 각 개체의 인증서와 해시체인을 활용하여 통신 오버헤드를 줄인다. 본 논문에서 제안한 기법은 인증서 탑업, 인증서 등록, 키분배, 키 업데이트, 메시지 인증의 4개의 단계로 구성된다.

3.1 인증서 탑업

ECU 제조회사는 인증기관을 통해 각 장치의 인증서 $Cert_i$를 ECU_i와 $Cert_G$를 게이트웨이 ECU (ECU_G)에 각각 탑업한다. ECU_i를 위한 인증서 탑업은 다음 과정을 따른다. ECU_G를 위한 인증서 탑업도 이와 비슷한 과정을 수행한다.

단계 1. 제조회사는 공개키와 개인키 쌍인 (PK_{A1}, SK_{A1}), (PK_{A2}, SK_{A2}), ..., (PK_{An}, SK_{An})를 각 ECU를 위해 생성한다. 인증서 발급을 위해 ECU의 시리얼 번호 SN_{Ai}와 ECU의 공개키

PK_{Ai}를 인증기관으로 보낸다.

단계 2 인증기관은 각 ECU의 공개키와 시리얼 번호를 이용하여 인증서($Cert_{A1}$, $Cert_{A2}$, ..., $Cert_{An}$)를 생성하고 이들 인증서를 제조회사에 보낸다.

단계 3. 제조회사는 시리얼 번호 SN_{Ai}와 인증서 $Cert_{Ai}$ 그리고 인증기관의 인증서 $Cert_{CA}$를 각 ECU_{Ai}에 탑업한다.

3.2 인증서 등록

ECU는 차 제조회사가 새로운 자동차 출고시 인증서에 대한 업데이트가 필요할 때, 또는 ECU를 다시 등록할 때 이 과정을 수행한다.

단계 1. ECU_{Ai}가 인증을 위한 난수 N_{Ai}를 생성하고 개인키 SK_{Ai}를 이용하여 이 값을 서명한다. 그리고 N_{Ai}, $Cert_{Ai}$, $Sig_{Ai}(N_{Ai})$를 ECU_G에게 전송한다.

단계 2. ECU_G는 복호한 N_{Ai}가 수신 받은 값과 같은지 확인하기 위해서 인증서의 공개키 PK_{Ai}를 이용하여 복호한다. ECU_G는 난수인 N_G를 생성하고 N_g, $Cert_G$, $Sig_G(N_{Ai}, N_G)$를 ECU_{Ai}로 전송한다.

단계 3. ECU_{Ai}는 $Sig_G(N_{Ai}, N_G)$를 검증하여 ECU_G를 검증한다. 검증이 성공하면 등록 단계를 마친다.

3.3 키 분배

이 단계는 다양한 세션의 초기화가 필요한 경우에 수행한다. 하나의 ECU_G가 n개의 ECU로 구성된 CAN에 존재한다는 가정 하에 본 단계를 수행한다.

단계 1. ECU_G는 n개의 각 ECU의 키 전송을 위해 난수 N_1, N_2, ..., N_n을 생성한다. ECU_G는 $S_{i1}=h(S_0||N_i)$, $S_{i2}=h(S_{i1})$, ..., $S_{in}=h(S_{in-1})$을 계산하고 N_i와 S_{in}를 ECU_{Ai}의 공개키로 암호하여 ECU_{Ai}에게 보낸다.

단계 2. ECU_{Ai}는 자신의 비밀키를 이용하여 메시지를 복호하고 S_{in}과 N_i를 확인한 후 S_{in}를 이용하여 SHA-1과 같은 해시함수를 통해 세션키 SK_{Sin}, 메시지 인증키 AK_{Sin}, 세션키 업데이트키 KEK_{Sin}를 생성한다. 세션키 증명을 위해 AK_{Sin}을 이용해서 N_i를 암호하고 ECU_G에게 보낸다.

단계 3. ECU_G는 ECU_{Ai}와 동일한 해시함수 및 해시체인 비밀키 S_{in}을 이용하여 다양한키들을 생성한 후, AK_{Sin}을 이용하여 키분배가 성공하였는지 확인한다. ECU_G는 ECU_{Ai}는 각각 자신의 카운터 값 CTR_{Ai}을 0으로 설정한다. CAN의 각 개체는 본 단계에서 생성한 세션관련 키들을 이용하여 안전한 통신을 수행할 수 있다.

3.4 키 업데이트

세션키의 안전성을 위해 특정 시점 i 후에 본 단계를 주기적으로 수행한다.

단계 1. ECU_G는 S_{in-i}를 ECU_{Ai}의 세션키 업데이트키인 $KEK_{Sin-i+1}$을 이용하여 암호화한 후 ECU_{Ai}에게 보낸다.

단계 2. ECU_{Ai}는 자신의 $KEK_{Sin-i+1}$을 이용하여 수신한 값을 복호하고 수신된 새로운 키의 유효성을 $h(S_{in-i+1})$을 계산하여 검증한다. 검증이 성공하면, ECU_{Ai}는 수신 된 해시체인 키 S_{in-i}와 KDF를 이용하여 세션키 SK_{Sin-i}, 메시지 인증키 AK_{Sin-i}, 세션키 업데이트키 KEK_{Sin-i}를 생성한다. ECU_G와 ECU_{Ai}는 각각 자신의 카운터 값 CTR_{Ai}를 0으로 설정한다.

3.5 CAN 메시지 인증

세션키는 다양한 통신 세션의 CAN 메시지 인증을 위해 이용한다. 다음은 i번째 CAN 메시지 인증을 위한 절차이다.

단계 1. 메시지 전송을 위해 ECU_{Ai}는 세션키 SK_{Sin-i}를 이용하여 카운터 값 CTR_{Ai}를 암호하고 $C=E_{SK_{Sin-i}}(CTR_{Ai})\oplus Message$와 메시지 인증 코드 $MAC=AES_{AK_{Sin-i}\oplus CTR_{Ai}}(C)$를 계산한 후 상위 5바이트를 활용한다. ECU_{Ai}는 생성된 메시지에 메시지 인증 코드를 추가하여 브로드캐스트하고 CTR_{Ai}를 1 증가시킨다.

단계 2. ECU_G는 ECU_{Ai}로부터 메시지를 받은 후, 카운터 값 CTR_{Ai}를 이용하여 메시지를 복호하고, 메시지 인증 코드를 검증한다. 검증이 성공하면 메시지 내용을 성공적으로 확인하고 CTR_{Ai}를 1 증가시킨다.

Ⅳ. 분석

본 장에서는 제안한 보안 기법에 대한 기밀성, 무결성 그리고 메시지의 실시간 처리 관점의 보안 분석과 통신 오버헤드 관점에서 성능 분석을 실시한다.

4.1 보안 분석

- 메시지 기밀성 : 제안된 기법은 AES-CTR을 이용하는 세션키 SK_i를 이용하여 메시지를 암호한다. 또한 세션키는 해시체인에 의존적인 세션키 S_{in-i}를 이용하여 주기적으로 업데이트된다. 즉, 공격자가 메시지의 내용에 대한 공격을 성공하기 위해서는 주어진 세션 내에 세션에 의존적인 키 S_{in-i}를 획득하는 것이다. 하지만 공격자가 세션키를 알기 위해서는 해시함수의 일방향성 문제를 해결할 수 있어야 한다. 또한 본 논문에서 제안한 기법의 세션키는 KDF를 이용하여 생성하기 때문에 공격자는 KDF 함수의 문제 또한 해결할 수 있어야 한다.
- 메시지 무결성 : 제안된 기법은 AES-128을 이용한 40 비트의 MAC을 이용한다. 공격자는 1 / 229.5의 확률로 올바른 MAC을 획득할 수 있다. 즉, ECU의 인증키 AK_i를 모르는 공격자는 메시지의 적법한 MAC을 생성 할 수 없다. 특히, 인증키는 세션마다 주기적으로 업데이트되기 때문에 공격자의 성공 가능성은 일반 MAC보다 더 낮다.
- 재전송 공격 대응 : 제안된 기법은 해시체인을 이용하여 세션키, 메시지 인증 코드, 세션키 업데이트키를 생성한다. 즉 세션별 해시체인에서 하나의 비밀값을 이용하고 ECU마다 별도의 N_i를 이용하여 해시체인의 값을 생성하기 때문에 ECU 마다 별도의 해시체인 값을 생성한다. 또한 각 해시 값은 하나의 세션에서만 유효하므로 제안된 기법은 재전송 공격에 안전하다.

4.2 성능 분석

본 절에서는 성능 분석을 제시하기 위해 통신 오버헤드에 초점을 맞춘 분석을 기존 기법과 비교하여 제시한다.

Table 2. 통신 부하 비교

기법 / 단계	Cho [7]	제안한기법
인증서 탑업	3	3
인증서 등록	6	2
키 분배	4	2
키 업데이트	2	1
CAN 메시지 인증	1	1

본 논문에서 제안한 기법은 키 분배 및 업데이트를 위해 해시체인을 사용한다. 제안한 기법은 메시지 암호화를 위해 오직 한 개의 인증서와 서명을 사용하기 때문에 두 개의 인증서를 필요로 하는 Cho의 기법보다 통신 부하가 Table 2에서 보여주는 바와 같이 명확히 효율적임을 확인할 수 있다.

V. 결론 및 향후 연구 방향

본 논문에서는 해시체인에 기반 한 CAN의 메시지 인증 및 키분배 기법을 제안하였다. 본 논문에서 제안한 기법은 데이터 패킷의 기밀성과 무결성을 제공하고, 또한 재생 공격으로부터 CAN 통신을 보호할 수 있다. 제안된 기법은 AES-CTR를 사용하여 재생 공격에 대응 할 수 있다. CAN 통신에 초점을 맞춘 기타 보안 관련 연구에서는 침입 탐지 기법에 초점을 수동 보안을 제공하였지만, 제안된 기법은 스니핑 공격이나 재생 공격 등에 대한 가용성을 지원하는 액티브 보안 제공한다.

참고문헌

[1] M. Wolf, A. Weimerskirch, and C. Paar, Embedded Security in Cars - Securing Current and Future Automotive IT Applications, Springer, 2006.

[2] T. Hoppe, S. Kiltz, and J. Dittmann, “Security threats to automotive CAN networks - practical examples and selected short-term countermeasures,” Proceedings of the 27th International Conference on Computer Safety, Reliability, and Security, pp. 235-248, 2008.

[3] M. Wolf, A. Weimerskirch, and T. Wollinger, “State of the art : embedding security in vehicles,” EURASIP Journal on Embedded Systems, vol. 2007, pp. 16, 2007.

[4] D. K. Nisson and U. E. Larson, “A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure,” in Journal of Networks, vol. 4, no. 7, pp. 552-564, 2009.

[5] 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, 2004.

[6] D. K. Nisson and U. E. Larson, “Simulated Attacks on CAN Buses:Vehicle virus,” in Proceedings of the Fifth LASTED Asian Conference on Communication Systems and Networks, pp. 66-72, 2008.

[7] A. Cho, A Message Authentication and Key

Distribution Mechanism Secure Against CAN Bus Attack, Master Thesis, Korea University, 2013.