ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 23-02-11
78

j.inst.Korean.electr.electron.eng.Vol.27,No.2,204~212,June 2023
https://doi.org/10.7471/ikeee.2023.27.2.204

CAN-FD 프로토콜에서 신뢰성과 보안성이 향상된 인증된 암호화 아키텍쳐 및 하드웨어 엔진의 설계

Design of a Authenticated Encryption Architecure and Hardware Engine with Improved Reliability and Security on CAN-FD Protocol

이 동 현*, 장 가 현*, 이 성 수*★

Donghyeon Lee*, Gahyeon Jang*, and Seongsoo Lee*★

Abstract

In this paper, an authenticated encryption architecture with improved reliability and security is proposed and implemented in hardware. The proposed architecture exploits Encrypt-and-MAC based on AES-128 and HMAC for easy parallelization. It exploits dual modular redundancy to improve reliability. It also exploits channel communication counter as additional encryption key, so security is improved with minimum additional hardware cost. Hardware engine of the proposed architecture was designed in FPGA, and it was verified to work correctly on CAN-FD bus.

요 약

본 논문에서는 CAN-FD 프로토콜에서 인증된 암호화의 신뢰성과 보안성을 향상시키기 위한 아키텍쳐를 제안하고 이를 하드웨어로 구현하였다. 제안된 아키텍쳐는 병렬 처리에 용이하도록 AES-128과 HMAC에 기반한 Encrypt-and-MAC 방식을 사용하였으며 신뢰성을 향상시키기 위해 하드웨어 이중화를 적용했다. 또한 채널 간 전송 횟수를 기억하는 카운터를 도입하여 마치 추가 암호 키처럼 사용하여 하드웨어 부담을 최소화하면서도 보안성을 강화하였다. 제안된 아키텍쳐를 위한 하드웨어 엔진을 FPGA로 설계하고, CAN-FD 버스 상에서 동작하는 것을 확인하였다.

Key words : Automotive, Cyber Physical System, Security, Reliability, Authenticated Encryption, Cryptography, In-Vehicle Network, Controller Area Network

* Soongsil University (Master Student, Undergraduate Student, Professor)
★ Corresponding author
E-mail: sslee@ssu.ac.kr, Tel: +82-2-820-0692
※ Acknowledgment
This work was supported by the R&D Program of the Ministry of Trade, Industry, and Energy (MOTIE) (20023805, RS-2022-00154973, RS-2023-00232192).
Manuscript received Jun. 23, 2023; revised Jun. 27, 2023; accepted Jun. 28, 2023.

Ⅰ. 서론

최근 자동차는 기계적이고 물리적인 시스템과 컴퓨팅 노드가 밀접하게 상호작용하여 제어되는 Cyber Physical System(CPS)이라는 확장된 개념을 바탕으로 설계된다. 기존의 기계식 제어 방식은 전자식 제어 방식으로 대체되고 있으며 이를 위해 다수의 센서 및 액추에이터와 전자제어장치(ECU)를 요구한다. Controller Area Network (CAN)[1]은 ECU 간 통신을 위해 가장 보편적으로 사용되는 프로토콜이다. 하지만 자율주행, Advanced Driver Assistance System (ADAS)와 같은 기술이 고도화됨에

따라 차량 내 네트워크 대역폭에 대한 수요가 증가하였다. CAN을 기반으로 한 CAN Flexible Date Rate (CAN-FD)[2]는 더 높은 전송 속도와 더 긴 페이로드[3]를 제공하여 적은 추가 비용으로 CAN을 대체할 수 있다는 장점이 있다. 하지만 CAN 및 CAN-FD는 보안 취약성이 존재한다. 데이터가 브로드캐스트될 때 데이터 프레임의 기밀성과 무결성 및 인증을 보장하지 않아 공격자가 쉽게 데이터를 도청하거나 추가적인 공격을 할 수 있다. ECU를 보호하기 위한 Hardware Trust Anchor (HTA) 기술은 다양한 보안 기능을 제공하지만 CAN-FD와 같은 특정 프로토콜에 대한 보안을 목적으로 하지 않으며 내결함성을 포함하고 있지 않다. 이를 보완하기 위해 많은 CAN 및 CAN-FD 보안 연구가 진행되었다. [4]에서는 CAN 인증된 암호화 방식에 타임스탬프를 적용한 인증방식을 사용하였다. [5], [6]은 CAN에 대한 MAC과 카운터를 기반으로 재생 공격과 변조 공격을 방지하는 메커니즘을 제안했다. [7]은 CAN-FD를 위한 소프트웨어 기반 실용적인 보안 아키텍쳐를 제안했으며 [8]은 하드웨어 리소스를 무시한 설계에서의 CAN-FD 기반 ECU의 보안 아키텍쳐를 제안하였다.

본 논문에서는 CAN-FD 프로토콜에서 신뢰성과 보안성이 향상된 인증된 암호화(Authenticated Encryption)인 AAE(Advanced Authenticated Encryption) 아키텍쳐를 제안하고 이를 수행할 수 있는 하드웨어를 설계한다. 제안된 아키텍쳐는 병렬 처리를 수행할 수 있도록 AES-128[9]과 HMAC[10]을 적용한 Encrypt-and-MAC 방식을 사용하였다. 기존보다 내결함성 및 신뢰성을 향상시키기 위해 Dual Modular Redundancy(DMR)을 적용했다. 또한 채널 간 전송 횟수를 기억하는 카운터를 도입하여 인증 키를 하나 더 추가한 것과 유사한 효과를 얻도록 함으로서, 하드웨어 부담을 최소로 하면서도 효율적이고 안전하게 인증된 암호화를 수행하도록 하였다. 설계된 하드웨어는 Xilinx Artix-7 FPGA를 통해 구현 및 검증되었다.

II. 차량 전자 시스템의 보안 취약성

1. CAN-FD의 보안 취약성

CAN은 Bosch 사에서 개발되었으며 차량 내 센서, 액추에이터 및 제어 장치 간의 통신을 지원하는 차량용 네트워크이다. CAN의 작은 대역폭이 오늘날 복잡한 전장 시스템의 요구사항을 충족시키지 못해 CAN의 확장성에 대한 요구가 증가했다. CAN-FD는 기존의 CAN 2.0 표준에 일부 개선 사항을 도입한 통신표준이다. 데이터 필드에서 높은 전송 속도를 지원하며, 최대 64바이트의 데이터를 한 번에 전송할 수 있다.

CAN 프로토콜은 중재, 메시지 필터링, 오류 감지와 같은 기능으로 높은 신뢰성과 안정성을 지원한다. 하지만 [11]-[13]에서 언급한 대로 CAN은 보안 기능을 제공하지 않으며 이는 CAN 및 CAN-FD가 다음과 같이 기밀성, 무결성, 인증 능력이 부족하여 도청 및 재생 공격이 가능함을 의미한다[7].

(1) 기밀성

CAN-FD의 데이터 프레임은 모든 노드로 브로드캐스팅되어 공격자가 쉽게 프레임을 도청할 수 있다. 전송된 메시지가 암호화되지 않은 경우, 공격자는 도청된 메시지에서 쉽게 정보를 추출하여 추가 공격을 할 수 있다[8]. 예로 엔진 데이터에 대한 도청을 통해 공격이 이뤄진다면 차량 주행에 심각한 문제가 발생할 수 있다. 따라서 차량용 네트워크상에서 전송되는 데이터는 암호화 돼야 하며 ECU 내의 HTA에서 해당 암호문을 처리하는 기능을 지원해야 한다.

(2) 무결성 및 인증

CAN-FD에서 사용되는 메시지 ID는 프레임의 기능 및 목적을 식별하는 데 사용되나 메시지의 출처 또는 인증 능력을 제공하지 않는다. 공격자는 ID를 도용함으로써 악의적인 ECU를 정상 ECU로 위장하여 CAN-FD 네트워크에 접속하고 통신에 참여할 수 있다. 이후에 공격자가 네트워크에 악의적인 메시지를 주입하거나 기존 메시지를 변조 및 재전송하는 것에 대한 억제 능력이 존재하지 않는다. 따라서 이러한 유형의 공격을 막기 위해서는 데이터 프레임의 인증이 보장되어야 한다.

2. Hardware Trust Anchor 기술의 한계

ECU 내의 보안에 민감한 데이터를 외부로부터 보호하거나 보안 사양을 충족시키기 위해 다양한 HTA 기술이 ECU에 통합되고 있다[14]. ECU의 온칩 확장으로 구현될 수 있는 Secure Hardware Extension(SHE)[15]와 EVITA[16] Hardware Security Module(HSM)은 현재 자동차 영역에서 최신 기술로 사용되고 있으며 보안 전용 칩인 Trusted Platform Module(TPM)[17]은 원래 Intel프로세서 등에 적용되었으나 점차 자동차 어플리케이션에도 적용되고 있다[18].

HTA 기술이 보안 키 생성, 암호화, 해싱 등의 보안 기

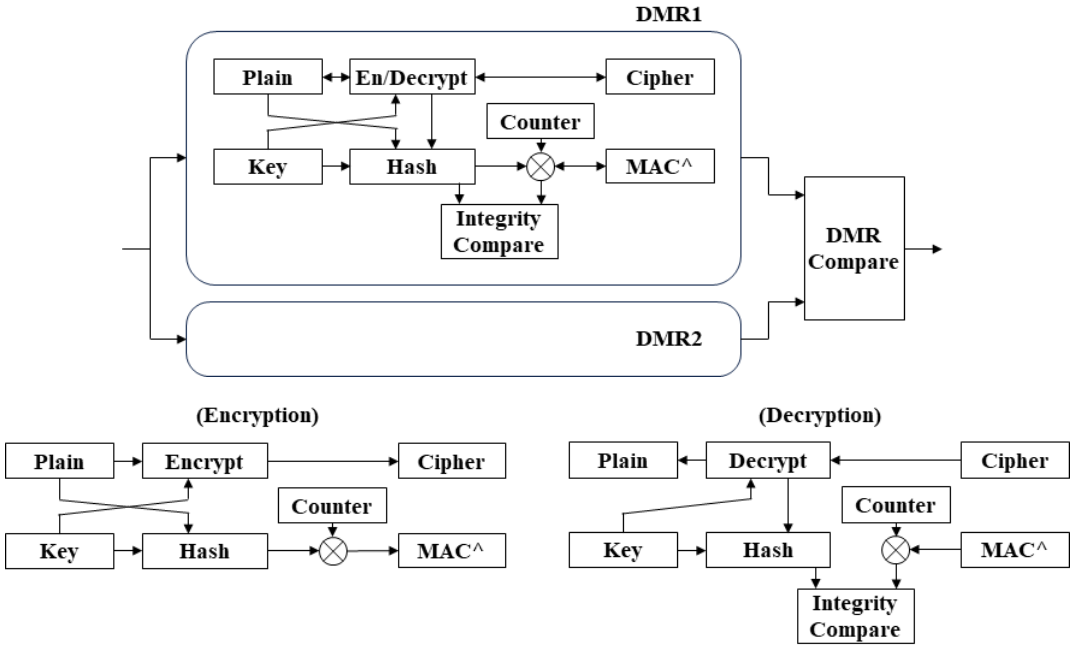


Fig. 1. Architecture of the proposed authenticated encryption.
그림 1. 제안하는 인증된 암호화의 아키텍쳐

능을 제공하지만 특정한 통신 프로토콜에 대한 보안 아키텍쳐를 제공하지 않는다[16]. 또한 HTA 모듈 내부 구조는 내결함성을 통한 신뢰성을 보장해주지 않기에 HTA 기반 모듈의 작동에 오류가 발생한다면 ECU 및 물리적 시스템의 정상적인 동작이 불가하게 된다.

Ⅲ. 신뢰성과 보안성이 향상된 인증된 암호화 아키텍쳐

1. 아키텍쳐

본 논문에서는 차량용 ECU의 네트워크로 사용되는 CAN-FD 프로토콜의 보안성과 신뢰성을 향상상시킨 인증된 암호화 아키텍쳐인 AAE를 그림 1과 같이 제안한다. [8]에서 제안된 자동차 CPS를 위한 신뢰할 수 있는 접근 방식은 성능, 에너지 및 리소스 오버헤드를 무시했으며 CAN-FD 공격에 대한 대처방안이 부족하다. 또한 FPGA의 기본 특성을 활용한 신뢰성 요소는 기존 HTA 기술과 호환되기 어렵다.

제안하는 아키텍쳐는 인증된 암호화 방식 중 Encrypt and MAC 방식을 사용하며 기밀성과 무결성을 위한 알고리즘으로 각각 AES-128[9]과 SHA-256[19] 기반 HMAC을 사용하여 기밀성, 무결성 및 인증을 제공한다. 특히 소프트 에러, 가혹 환경 등으로 인해 차량용 어플리케이션에서 발생할 수 있는 신뢰성 문제를 해결하기 위해 주요 하드웨어의 이중화 기술인 DMR을 적용하여 비교적 적은 하드웨어 리소스로도 차량용 ECU 내의 HTA에서 지원하지 않는 물리적 신뢰성을 제공한다.

CAN-FD 프로토콜에서 고려되는 주요 공격 유형과 인증된 암호화 방식의 단점을 보완하기 위해 채널 별로 데이터를 주고받은 횟수를 기억하는 카운터를 도입하고 이 카운터 값으로 MAC을 XOR한 MAC^을 Encrypt-and-MAC에서 MAC 대신에 사용하여 인증 키를 하나 더 추가한 것과 유사한 효과를 얻도록 함으로서, 최소한의 리소스를 사용하면서도 CAN-FD 공격에 대한 향상된 대처를 수행할 수 있다.

제안하는 AAE 아키텍쳐는 단일 하드웨어로 암호화 모드, 복호화 모드를 모두 수행할 수 있도록 설계하였다. 그림 2 (a)와 (b)는 각각 AAE 아키텍쳐에서 송신 노드와 수신 노드가 암복호화를 처리하는 과정이다. 그림 2 (a)에서 송신 노드는 128비트 평문 메시지를 입력받아 병렬적으로 128비트 암호문과 인증을 위한 256비트 MAC을 생성한다. MAC은 데이터 처리를 시작할 때마다 증가하는 256비트 카운터와 XOR 연산을 거쳐 MAC^으로 변경된다. 암호문과 MAC^은 연결되어 총 384비트가 CAN-FD 페이로드를 통해 다른 ECU로 전송된다. 그림 2 (b)에서 수신 노드는 384비트의 페이로드를 분리하고 암호문을 해독해 평문 메시지를 얻는다. 평문 메시지를 통해 MAC을 생성하고 MAC^을 카운터와 XOR한 값과 비교한다. 값이 일치하면 인증에 성공한 것으로, 불일치하면 계산에 오류가 있거나 다른 카운터를 사용한 것으

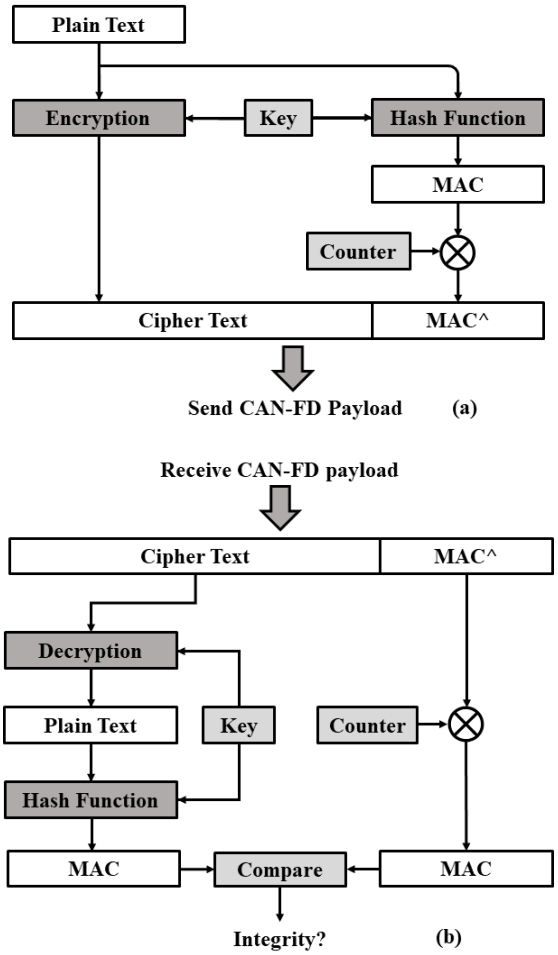


Fig. 2. Encryption/decryption process of the proposed architecture (a) Sender node (b) Receiver node.
그림 2. 제안하는 아키텍쳐의 암복호화 과정 (a) 송신 노드 (b) 수신 노드

로 간주하여 메시지를 폐기한다. 제안하는 아키텍쳐는 초기 키가 외부로부터 안전한 ECU 내 메모리에 저장되었으며 CAN-FD를 통해 모든 노드로 배포된 것을 가정한다.

2. 인증된 암호화

인증된 암호화는 데이터의 기밀성, 무결성을 동시에 제공하는 암호화 체계이다. 기밀성은 암호화를 통해, 무결성은 주로 MAC을 통해 제공된다. 데이터가 전송 도중 조작되거나 변조되지 않았음을 확인할 수 있어 통신 체계의 보안성을 높여준다. 인증된 암호화 방식에는 Encrypt-then-MAC(EtM), Encrypt-and-MAC(E&M), MAC-then-Encrypt(MtE) 방식이 있다. EtM 방식과 MtE 방식은 평문 데이터에 대한 정보가 노출되지 않아 상대적으로 안전하다고 평가받는다. 하지만 기밀성과 무결성을 순차적으로 확인해야 하기에 계산 오버헤드가 증

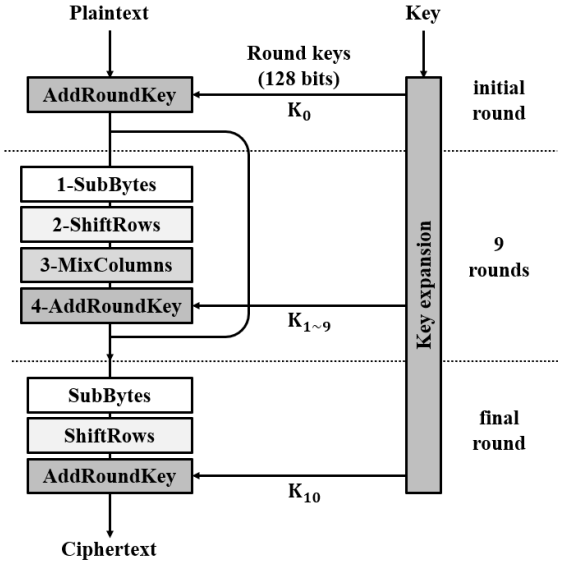


Fig. 3. Encryption process of AES-128.
그림 3. AES-128의 암호화 과정

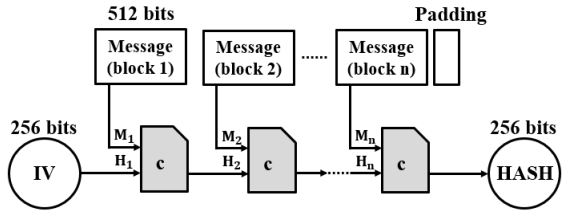


Fig. 4. Hashing process of SHA-256.
그림 4. SHA-256의 해시 계산 과정

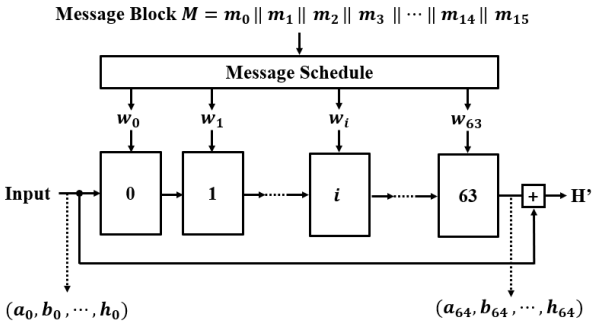


Fig. 5. Compression function C.
그림 5. 압축함수 C

가한다. E&M 방식은 기밀성과 무결성 검증을 병렬적으로 수행해 계산 오버헤드가 적다는 장점이 있다. 차량용 ECU에 사용되는 암호화는 빠른 처리 시간을 요구하기 때문에 본 논문에서는 E&M 방식을 사용하였다.

가. AES-128

AES-128은 NIST에서 표준으로 제정한 대칭키 암호화 알고리즘이며 [20]에서 보안성이 입증되었다. 128비

트의 입력과 128, 192, 256비트의 가변 키 길이를 지원한다. 암호화 키의 길이에 따라 수행하는 라운드 수가 달라지며 128비트 키를 사용할 경우 10 라운드를 수행한다. AES-128 알고리즘은 크게 KeyExpansion, 초기 라운드, 1~9라운드, 마지막(10) 라운드의 네 단계로 이루어지며 그림 3과 같이 구성되어 있다. KeyExpansion 단계에서는 128비트 라운드 키를 생성하며 각 라운드에서는 라운드 키를 사용하여 SubBytes, ShiftRows, MixColumns, AddRoundKey 연산을 수행하는데, 단계별 연산의 순서와 개수는 서로 다르다.

나. SHA-256

SHA-256은 NIST에서 표준으로 제정한 암호학적 해시 함수이며 SHA-2 패밀리에 속한다. 임의의 길이를 갖는 메시지를 받아 256비트 고정 길이의 고유한 해시값을 생성한다. 그림 4는 SHA-256 알고리즘의 수행 과정을 나타낸 것이다. 메시지를 512비트의 배수에 맞춰 패딩하고 초기 해시값 IV를 사용하여 압축함수 C를 512비트의 배수만큼 수행한다. 압축함수 C의 수행 과정은 그림 5와 같다. 512비트 메시지와 이전 해시값을 사용해 64회 라운드하여 고유한 256비트 해시값을 얻을 수 있다.

다. HMAC

HMAC은 해시 함수와 대칭 암호화 키를 사용해 메시지 무결성과 인증을 보장하는 방식이다. 메시지와 함께 HMAC을 전송해야 하며 수신자는 수신된 메시지에 대해 대칭키를 사용하여 HMAC을 계산한다. 계산된 결과와 수신한 HMAC의 일치 여부를 통해 무결성을 판단한다. 그림 6은 HMAC 알고리즘의 수행 과정을 나타낸 것이다. Ipad와 대칭키를 XOR 연산한 값 S를 메시지와 함께 해시 함수의 입력으로 사용하여 HMAC'을 계산한다. Opad와 대칭키를 XOR 연산한 E와 HMAC'을 사용하여 해시 함수를 반복하면 최종 HMAC이 계산된다.

3. 카운터 기반 공격 방지

제안한 아키텍쳐에 사용된 E&M 방식은 계산 시간에 대한 장점이 있지만 암호화되지 않은 평문 데이터에 대한 MAC이 노출된다는 문제가 존재한다. [8]에서 제안한 접근 방식은 64비트 카운터를 64비트 평문 메시지와 결합하여 암호화하는 방식을 사용했다. 하지만 이는 메시지의 가용 비트 수를 제한하고 평문에 대한 MAC이 노출된다는 문제가 존재한다. 또한, 카운터를 통한 재생 공격의 방지 방법을 구체적으로 제시하지 않았다.

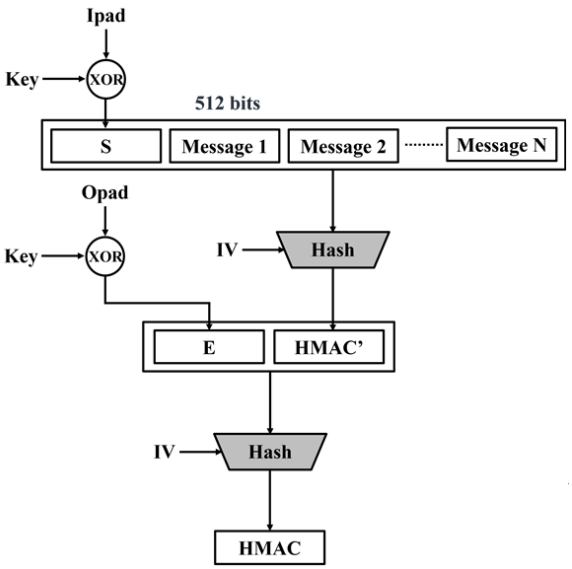


Fig. 6. Computation process of HMAC.
그림 6. HMAC의 계산 과정

따라서 본 논문에서는 외부에 노출되는 MAC을 카운터 값과 XOR하여 MAC을 보호하고 동시에 재생 공격에 대응하는 방법을 사용한다. 송신 노드는 데이터를 프로세서로부터 입력받아 처리할 때마다 카운터를 증가시킨다. MAC을 생성한 후 해당 MAC에 XOR 연산함으로써 해당 MAC의 생성 정보를 내장한다. 수신 노드 또한 데이터를 복호화할 때마다 카운터를 증가시켜 수신한 MAC에 XOR 연산하여 평문에 대한 MAC을 얻는다. 공격자의 재생 공격으로 인해 이전과 동일한 데이터 프레임이 전송되었더라도 수신 노드는 해당 카운터와 다른 카운터를 사용하기 때문에 MAC에 대한 무결성 검증에 실패하게 된다. 하지만 CAN-FD에서는 메시지 ID 기반 전송 방식을 사용하기 때문에 개별 ID에 대한 카운터 관리 연구가 추가적으로 필요하다.

4. 내결함성

내결함성은 시스템이 하나 이상의 결함이 발생했을 때 시스템이 정상적으로 작동하는 능력이다. 내결함성이 있는 시스템은 결함을 감지하고 복구할 수 있어야 하며 이를 위해 해당 요소는 중복된 구조를 가져야 한다. [21], [22]에서는 자동차 임베디드 시스템에서 사용할 수 있는 내결함성 구조를 연구했다. [8]에서는 DMR 구조에 1개의 예비 모듈을 사용해 오류를 감지하는 일종의 Triple Modular Redundancy(TMR)를 사용하였으며 FPGA의 특성을 활용한 알고리즘을 통해 오류를 복구한다. 그러나 이 방식은 동일한 연산을 위해 세 개의 모듈을 사용하여 하드웨어 리소스가 다소 과도하게 필요하기 때문에

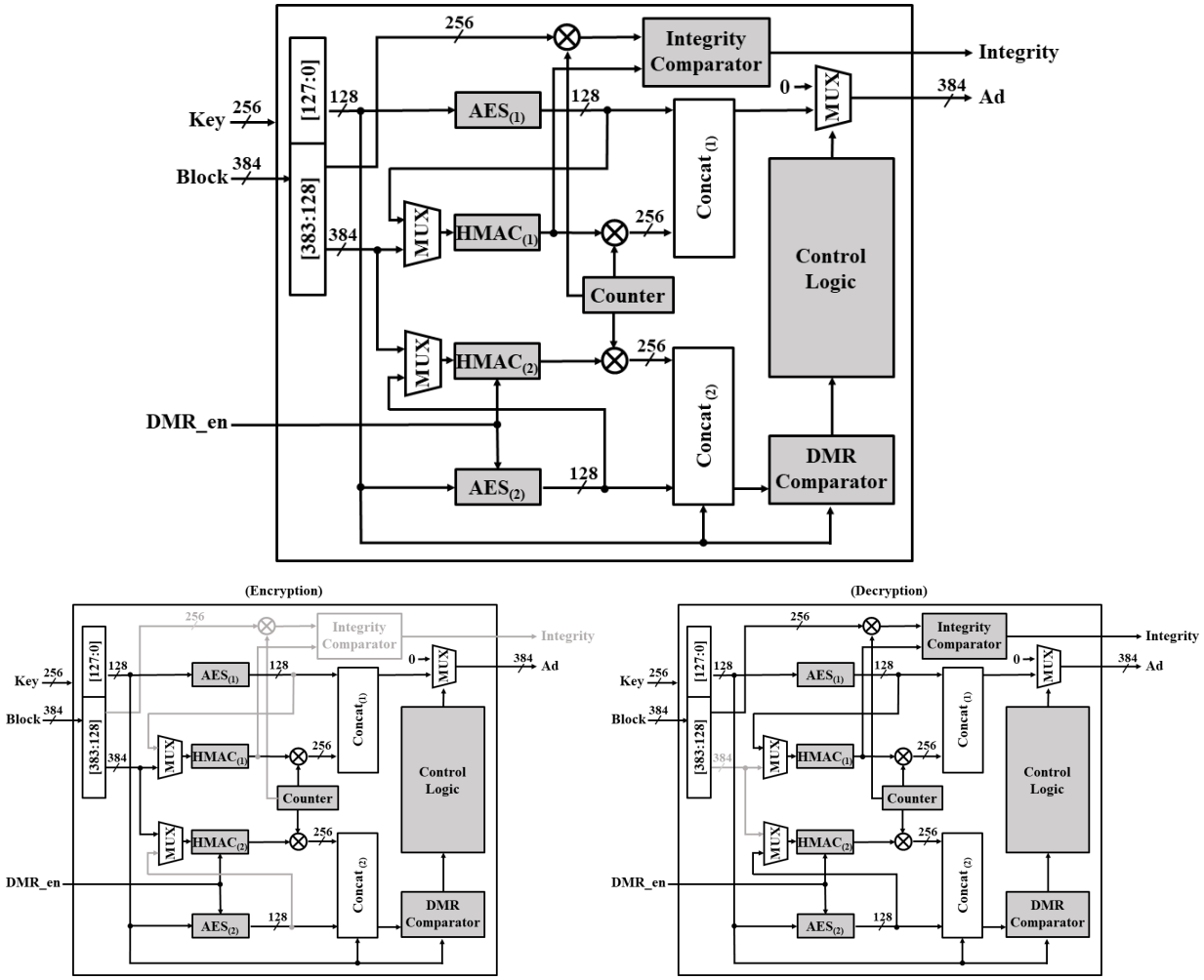


Fig. 7. Block diagram of AxE, i.e. hardware engine of the proposed authenticated encrytion.
그림 7. 제안하는 인증된 암호화를 수행하는 하드웨어 엔진인 AxE의 블록도

온 칩 기반 HTA에 사용하기 어려운 방식이다. 제안하는 아키텍쳐에서는 두 개의 모듈을 사용하는 DMR을 적용하였으며 오류를 감지하면 Soft Reset을 이용해 모듈을 초기화하고 재계산하기 때문에 구조가 단순하고 비용 측면에서 효율적이다.

Ⅳ. 제안하는 아키텍쳐의 하드웨어 엔진 구현

1. 엔진 설계

본 논문에서 제안하는 AAE 아키텍쳐에 기반한 하드웨어 엔진인 Advanced eXcellent Encryption(A×E) Engine의 구조는 그림 7과 같다. A×E Engine은 제어 신호를 생성하는 Control Logic, 각각 두 개의 AES 및 HMAC 코어, 256비트 카운터, 그리고 비교기로 이루어진다. 384비트 입력인 Block 신호 전체를 HMAC 코어에 입력하고, 이 중 평문 메시지에 해당하는 상위 128비트를 AES 코어로 입력한다. 256비트의 Key가 각 코어로 입력되며 AES 코어는 128/256비트의 키 길이를 선택할 수 있다. 카운터와 XOR 연산된 HMAC의 결과와 AES 코어의 출력은 384비트의 Concat 블록에 정렬된다. Control Logic의 제어 신호에 따라 Concat 블록이 인증된 암호문 결과인 Ad로 출력된다. DMR_en이 1이면 하단에 (2)로 표기된 두 번째 DMR 중복 하드웨어가 동작하고 Concat(1), Concat(2)를 비교한다. 비교 결과가 다른 경우 AES 또는 HMAC 코어에 오류가 발생한 상황이므로 이들 코어를 Soft Reset 시킨 후 재계산 과정이 이뤄진다. 수신 시에는 수신된 HMAC의 무결성을 확인하기 위해 Block 신호의 하위 256비트를 카운터와 XOR 연산하여 새롭게 계산된 HMAC과 비교하여 Integrity 신호를 생성한다.

2. FPGA 구현

표 1은 그림 7을 바탕으로 설계한 AxE Engine의 FPGA 합성 결과이다. Xilinx Artix-7 FPGA에서 Verilog

HDL로 설계를 구현하고 Vivado 2022.1을 통해 합성되었다. 전체 AxE Engine의 면적 대부분을 AES 및 HMAC 코어가 차지하고 있으며 이외에 부가적인 기능을 위한 회로의 면적은 상대적으로 매우 적음을 알 수 있다.

Table 1. FPGA implementation result of AxE hardware Engine.
표 1. AxE 하드웨어 엔진의 FPGA 구현 결과

	LUT	Slice	Power (W)	Operation time(us)
AxE w/ahb (s)	11761	3624	0.027	9
AxE w/ahb (r)				13.76
AES	2913	1258	0.006	1.5
HMAC	2084	857	0.005	2.66

실행 시간은 IDEC(IC Design Education Center)에서 제공한 Siemens Questasim에서 측정되었으며 50MHz 주파수의 ARM Cortex M3 기반 MCU System에서 제안한 아키텍쳐를 위한 펌웨어 수행 시간이다. DMR을 통한 내결함성을 포함한 동작은 병렬처리를 통해 기본동작과 수 클럭 차이에 지나지 않기 때문에 측정하지 않았다. 전체 송신 및 수신 동작에서의 AxE Engine의 수행 시간은 각각 9us, 13.76us로 [8]에서 분석한 차량 내 애플리케이션의 실시간 제약 조건을 여전히 충족한다.

3. FPGA 검증

성능 평가에서 사용한 MCU System을 Xilinx Artix-7 FPGA에 합성하고 CAN-FD 트랜시버를 사용해 검증 환경을 송신 노드와 수신 노드 양쪽에 그림 8과 같이 구성했다. 송신 노드에서 평문 메시지를 AxE Engine을 통해 암호화해 Ad를 생성한다. 그림 9 (a)는 송신 노드의 암호화 과정이 UART를 통해 터미널로 출력된 결과이다. Ad는 MCU 코어를 통해 CAN-FD 페이로드에 내장되어 수신 노드로 전송된다. 수신 노드는 Ad를 복호화하여 MAC 계산 결과와 수신된 MAC이 일치하는지 확인한다. 그림 9 (b)는 수신 노드의 복호화 결과 및 무결성 확인 과정이 출력된 결과이다. 이와 같은 FPGA 검증을 통해 AxE Engine의 평문 메시지가 암호화 및 복호화 동작과 무결성 인증이 정상적으로 이루어지는지 확인하였다.

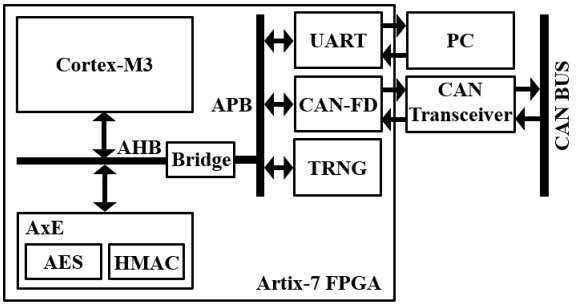
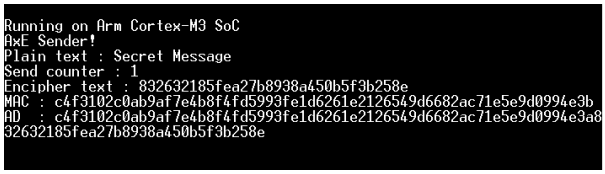


Fig. 8. FPGA verification environment.
그림 8. FPGA 검증 환경



(a)

(b)

Fig. 9. Output result of AxE Engine (a) Sender node (b) Receiver node.
그림 9. AxE Engine의 출력 결과 (a) 송신 노드 (b) 수신 노드

V. 결론

본 논문에서는 CAN-FD 프로토콜의 보안 취약점과 HTA 기술을 보완하기 위한 아키텍쳐인 AAE를 제안한다. AAE는 AES-128과 HMAC을 사용하는 Encrypt-and-MAC 방식에 카운터 XOR을 통한 향상된 인증된 암호화 아키텍쳐이다. AAE를 실행할 수 있는 하드웨어 AxE Engine을 설계하고 FPGA를 통해 동작을 검증하였다. 해당 아키텍쳐를 통해 CAN-FD 프로토콜에 기밀성, 무결성 및 인증 능력을 제공할 수 있음을 보였다. CAN-FD 프로토콜의 메시지 ID 기반 전송 방식을 고려했을 때, ID 별 카운터 관리에 대한 추가적인 연구가 필요하다.

References

[1] Bosch, “CAN Specification,” 1991.
[2] Bosch, “CAN with Flexible Data Rate Specification,” 2012.
[3] B. Cheon and J. Jeon, “The CAN FD Network performance analysis using the CANoe,” *Proceedings of IEEE International Symposium on Robotics,* pp.1-4, 2013.

DOI: 10.1109/ISR.2013.6695598
[4] S. Chandrasekaran, K. Ramachandran, S. Adarsh, and A. Puranik, "Avoidance of Replay Attack in CAN Protocol Using Authenticated Encryption," *Proceedings of International Conference on Computing, Communication and Networking Technologies,* pp.1-6, 2020.
DOI: 10.1109/ICCCNT49239.2020.9225529
[5] B. Groza, S. Murvay, A. Herrewege, and I. Verbauwhede, "LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks," *Proceedings of International Conference on Cryptology and Network Security*, pp.185-200, 2012. DOI: 10.1007/978-3-642-35404-5_15
[6] C. Lin and A. Sangiovanni-Vincentelli, "Cyber-Security for the Controller Area Network (CAN) Communication Protocol," *Proceedings of International Conference on Cyber Security*, pp.1-7, 2012.
[7] S. Woo, H. Jo, I. Kim, and D. Lee, "A Practical Security Architecture for In-Vehicle CAN-FD," *IEEE Transactions on Intelligent Transportation Systems,* vol.17, no.8, pp.2248-2261, 2016.
DOI: 10.1109/TITS.2016.2519464
[8] B. Poudel and A. Munir, "Design and Evaluation of a Reconfigurable ECU Architecture for Secure and Dependable Automotive CPS," *IEEE Transactions on Dependable and Secure Computing,* vol.18, no.1, pp.235-252, 2021.
DOI: 10.1109/TDSC.2018.2883057
[9] FIPS 197, "Advanced Encryption Standard," https://csrc.nist.gov/publications/detail/fips/197/final
[10] FIPS 198-1, "The Keyed-Hash Message Authentication Code," https://csrc.nist.gov/publications/detail/fips/198/1/final
[11] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the art: Embedding security in vehicles," *EURASIP Journal of Embedded Systems*, article 074706, vol.2007, 2007.
[12] K. Koscher, A. Czeskis, F. Roesner, S. Patrel, T. Kohno, S. Checkoway, D. McKoy, B. Kantor, and D. Anderson, "Experimental Security Analysis of a Modern Automobile," *Proceedings of IEEE Symposium on Security and Privacy,* pp.447-462, 2010. DOI: 10.1109/SP.2010.34
[13] S. Woo, H. Jo, and D. Lee, "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems,* vol.16, no.2, pp.993-1006, 2015.
DOI: 10.1109/TITS.2014.2351612
[14] C. Plappert, A. Fuchs, and R. Heddergott, "Analysis and Evaluation of Hardware Trust Anchors in the Automotive Domain," *Proceedings of International Conference on Availability, Reliability and Security,* pp.1-11, 2022.
DOI: 10.1145/3538969.3538995
[15] AUTOSAR, "Secure Hardware Extension," https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_TR_SecureHardwareExtensions.pdf
[16] Evita Consortium, "EVITA E-safety Vehicle Intrusion Protected Applications," https://evita-project.org.
[17] Trusted Computing Group, "TPM 2.0 Library Specification," https://trustedcomputinggroup.org/resource/tpm-library-specification/
[18] Infineon Technologies, "A Safe for Sensitive Data in the Car: Volkswagen Relies on TPM from Infineon," https://www.infineon.com/cms/en/about-infineon/press/market-news/2019/INFATV201901-030.html
[19] FIPS 180-4, "Secure Hash Standard," https://csrc.nist.gov/publications/detail/fips/180/4/final
[20] A. Hodjat and I. Verbauwhede, "Minimum Area Cost for a 30 to 70 Gbits/s AES Processor," *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pp.83-88, 2004.
DOI: 10.1109/ISVLSI.2004.1339512
[21] E. Beckschulze, F. Salewski, T. Siegbert, and S. Kowalewski, "Fault Handling Approaches on Dual-Core Microcontrollers in Safety-Critical Automotive Applications," *Proceedings of International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation,* pp.82-92,

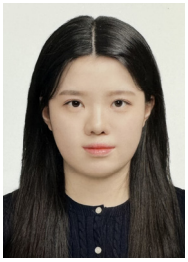
2008. DOI: 10.1007/978-3-540-88479-8_7
[22] M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, M. Peri, and S. Pezzini, "Fault-Tolerant Platforms for Automotive Safety-Critical Applications," *Proceedings of International Conference on Compilers, Architecture, and Synthesis for Embedded Systems,* pp.170-177, 2003. DOI: 10.1145/951710.951734

BIOGRAPHY

Donghyeon Lee (Member)

2022 : BS degree in Electronic Engineering, Soongsil University.
2022~ : Candidate for MS degree in Electronic Engineering, Soongsil University.
〈Main Interest〉 Security SoC, Automotive SoC, Cryptography

Gahyeon Jang (Member)



2020~ : Candidate for BS degree in Electronic Engineering, Soongsil University.
〈Main Interest〉 Security SoC, Automotive SoC, Cryptography

Seongsoo Lee (Life Member)

1991 : BS degree in Electronic Engineering, Seoul National University.
1993 : MS degree in Electronic Engineering, Seoul National University.
1998 : PhD degree in Electrical Engineering, Seoul National University.
1998~2000 : Research Associate, University of Tokyo
2000~2002 : Research Professor, Ewha Womans University
2002~Now : Professor in School of Electronic Engineering, Soongsil University
〈Main Interest〉 AI SoC, Automotive SoC, Security SoC, Processor SoC, Power Management SoC, Battery Management SoC, Reliability and Safety