(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0051220
(43) 공개일자 2020년05월13일

(51) 국제특허분류(Int. Cl.)
H04L 9/08 (2006.01) *H04L 12/40* (2006.01)
H04L 9/32 (2006.01)
(52) CPC특허분류
H04L 9/0869 (2013.01)
H04L 12/40104 (2013.01)
(21) 출원번호 10-2018-0134287
(22) 출원일자 2018년11월05일
심사청구일자 2018년11월05일

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
이상록
서울특별시 마포구 마포대로 115-8, 101동 203호 (공덕동, 공덕삼성아파트)
박수용
서울특별시 마포구 독막로 18길 5, 101동 1001호 (상수동, 두산위브아파트)
(74) 대리인
이지연

전체 청구항 수 : 총 6 항

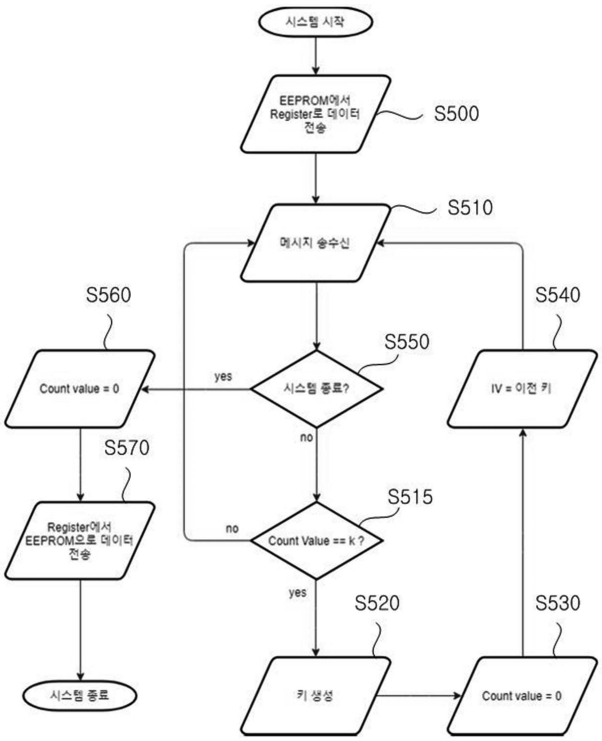
(54) 발명의 명칭 **분산 환경에서의 메시지 순서 기반의 키 생성 방법**

(57) 요 약

본 발명은 다수 개의 처리 유닛들로 구성되는 분산 환경에서의 각 처리 유닛에 의한 키 생성 방법에 관한 것이다. 상기 키 생성 방법은, (a) 목표 카운트 값 및 MAC 사이즈를 설정하는 전처리 단계; (b) EEPROM 에 있는 데이터들을 레지스터로 옮기고 카운트 값을 초기화시키는 단계; (c) 메시지를 분산 환경으로 송신하는 경우, 자

(뒷면에 계속)

대 표 도 - 도5



신의 비밀키를 이용하여 전송할 메시지와 비밀키를 이용하여 메시지 인증 코드(Message Authentication Code; 'MAC')를 생성하고, MAC과 메시지를 합쳐서, 분산 환경의 통신 네트워크에 브로드캐스팅하는 단계; (d) 브로드캐스팅된 메시지를 수신하는 경우, 원본 메시지와 MAC 을 분리하고, 원본 메시지로 MAC을 생성하며, 수신한 MAC과 생성된 MAC이 같은지를 비교하여 인가된 메시지인지 확인하고 검증하는 단계; (e) 수신된 메시지가 검증되면, 카운트 값을 1 증가시키는 단계; (f) 카운트 값이 목표 카운트값에 도달하면, 해당 순서의 메시지를 시드값으로 하여 새로운 키를 생성하는 단계; 를 구비하고, 상기 목표 카운트 값은 시드값으로 사용할 메시지를 결정하는 값으로 사용된다.

(52) CPC특허분류
H04L 9/0816 (2013.01)
H04L 9/3242 (2013.01)
H04L 2012/40215 (2013.01)
H04L 2012/40273 (2013.01)
H04L 2209/84 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711070410
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보통신기술인력양성
연구과제명 적응형 블록체인 플랫폼 기술 개발 및 전문 인력 양성
기 여 율 1/1
주관기관 서강대학교 산학협력단
연구기간 2018.01.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

다수 개의 처리 유닛들로 구성되는 분산 환경에서의 각 처리 유닛에 의한 키 생성 방법에 있어서,

(a) 목표 카운트 값 및 MAC 사이즈를 설정하는 전처리 단계;

(b) EEPROM 에 있는 데이터들을 레지스터로 옮기고 카운트 값을 초기화시키는 단계;

(c) 메시지를 분산 환경으로 송신하는 경우, 자신의 비밀키를 이용하여 전송할 메시지와 비밀키를 이용하여 메시지 인증 코드(Message Authentication Code; 'MAC')를 생성하고, MAC과 메시지를 합쳐서, 분산 환경의 통신 네트워크에 브로드캐스팅하는 단계;

(d) 브로드캐스팅된 메시지를 수신하는 경우, 원본 메시지와 MAC 을 분리하고, 자신의 비밀키를 이용하여 상기 분리된 원본 메시지로 MAC을 생성하며, 상기 분리된 MAC과 상기 생성된 MAC이 같은지를 비교하여 인가된 메시지인지 확인하고 검증하는 단계;

(e) 수신된 메시지가 검증되면, 카운트 값을 1 증가시키는 단계;

(f) 카운트 값이 목표 카운트값에 도달하면, 해당 순서의 메시지를 시드값으로 하여 새로운 키를 생성하는 단계;

를 구비하고, 상기 목표 카운트 값은 시드값으로 사용할 메시지를 결정하는 값으로 사용되는 것을 특징으로 하는 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법.

청구항 2

제1항에 있어서, 상기 키 생성 방법은 (g) 상기 (f) 단계에 의해 새로운 키가 생성되면, 이전 키는 블록 암호화 기반의 키 생성 알고리즘의 초기 벡터(Initial Vector)에 저장하는 단계; 를 더 구비하는 것을 특징으로 하는 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법.

청구항 3

제1항에 있어서, 상기 키 생성 방법은 (h) 종료되면 레지스터에 있는 카운트 값을 0으로 초기화시키고 EEPROM으로 데이터를 옮기고 종료하는 단계; 를 더 구비하는 것을 특징으로 하는 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법.

청구항 4

제1항에 있어서, 상기 키 생성 방법은,

상기 (a) 전처리 단계는 중요 메시지를 설정하는 것을 특징으로 하며,

상기 (c) 및 (d) 단계의 MAC 인증 과정은 상기 설정된 중요 메시지에 대해서만 수행하여 메시지 처리 속도를 향상시키는 것을 특징으로 하는 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서, 상기 키 생성 방법은 복수 개의 ECU들이 CAN 프로토콜에 따라 통신하는 차량 내부 네트워크에 적용되는 것을 특징으로 하는 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법.

청구항 6

제1항 내지 제4항 중 어느 한 항에 따른 키 생성 방법을 구현한 실행 가능한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 동적키 생성 방법에 관한 것으로서, 더욱 구체적으로는 분산 환경에 있어서, 메시지 순서를 기반으로 하여 동적키를 생성함으로써, 분산 환경에서 실시간성과 키의 기밀성을 보장할 수 있는 동적키 생성 방법에 관한 것이다.

배 경 기 술

[0002] 분산 환경은 기존 중앙 집중형 서버의 통제를 받는 것과 달리 피어 간의 직접적인 통신으로 이루어져 있다. 이중 CPU 성능이 낮은 임베디드 환경에서 실시간성을 위해 메시지 인증 매커니즘이 존재하지 않는 경우 해킹에 직접적으로 노출되어 있다. 반면 메시지 인증 매커니즘이 존재하는 경우에도 실시간성이 떨어지거나 암호화의 핵심인 키의 기밀성이 떨어진다.

[0003] 분산 환경 중 하나인 마이크로 컨트롤러가 탑재된 IoT 임베디드 기기들로 구성된 IoT 임베디드 환경에 있어서, IoT 임베디드 기기의 암호화는 데이터 통신 보안에 있어 중요하다. 하지만 키의 탈취가 쉽다면 암호화가 아무리 복잡하다고 해도 무용지물이기 때문에 키의 기밀성이 보장되어야 한다. 또한, 동적으로 키를 생성할 시 전체적인 메시지 전송 속도의 오버헤드를 최소화하는 키 생성 알고리즘이 필요하다.

[0004] 한편, CAN(Controller Area Network)은 자동차에서 수많은 기기들이 통신하기 위해 사용하는 프로토콜로서, 1986년에 보쉬(Bosch)에 의해 개발되었다. CAN이 개발되기 이전에 차량 내부 전자제어장치인 ECU(Electronic Control Unit)가 도입되었는데, 직렬 통신 방식인 UART(Universal Asynchronous Receiver/Transmitter)를 통해 ECU들을 연결했다. 도 1은 종래의 차량 내부 전자제어장치들인 ECU들간의 직렬 통신 방식인 UART 통신을 도시한 모식도이다. 도 1에 도시된 바와 같이, 전술한 UART 통신 방식은 장치가 추가될 때마다 연결선이 필요했기에 자동차의 원가 상승을 초래하여 비효율적이었다.

[0005] 이후, 모든 ECU들을 병렬적으로 연결하며 단일 인터페이스를 사용하는 CAN이 개발되고 차량 내부 통신의 표준 프로토콜로 채택되어 현재까지 사용되고 있다. 도 2는 종래의 차량 내부 전자제어장치들인 ECU들간의 통신 방식인 CAN 통신을 도시한 모식도이다. 도 2에 도시된 바와 같이, 전술한 CAN 버스는 내부의 모든 메시지를 신뢰한다는 가정하에 내부 인증 매커니즘이 전무하여 보안 상의 문제가 발생할 수 있다. 일반적으로 자동차 해킹은 로컬에 접근하는 방식으로 이루어져 있어 보안에 대한 위협이 적을 수 있지만, 예를 들어 주유소, 발렛 파킹, 자동차 렌트, 중고차, 주차장 등 차량이 노출되어 있는 환경이라면 얼마든지 해킹이 가능하다. 이보다 더 주목해야 할 것은 커넥티드 카에 있다. 차량 내부 네트워크가 외부 네트워크와 연결되어 있다면 보다 해킹이 용이해진다. 따라서 근본적으로 CAN의 설계를 바꾸지 않는 이상 메시지 위조 및 변조를 방지할 연구가 필요하다.

[0006] 한편, 블록체인은 블록에 거래 데이터를 저장하고 이를 해시값을 통해 각 블록을 연결한 형태를 가지고 있다. 또한, 단순히 중앙집중형 서버에 저장되는 것이 아니라, 모든 노드들이 동일한 블록체인을 가지게끔 하여 위변조가 불가능하도록 만들었다. 블록체인을 이루고 있는 기반 기술로는 PKI, P2P 네트워크, 합의 알고리즘 등이 존재한다. 처음엔 단순히 비트코인의 거래를 가능하게 해주는 기술이었지만 이더리움의 등장으로 스마트 계약이 가능해지면서 다양한 비즈니스 모델 및 공유 경제 모델에 적용하려는 움직임이 있다.

[0007] 한편, 종래의 암호화 키 생성 및 분배의 문제점을 살펴본다. 비대칭키를 이용한 전자서명은 메모리를 많이 사용하며 속도 또한 느리기에 임베디드 환경에 적합하지 않다. 반대로, 대칭키를 이용한 전자서명은 비대칭 키를 이용한 전자서명에 비해 속도가 빨라 임베디드 환경에 적합하지만, 키를 네트워크 상에 분배해야만 하는 보안의 문제가 있다. 이러한 문제를 해결하기 위해 KDC(Key Distribution Center)기반의 키 분배 프로토콜인 커버로스(kerberos) 프로토콜, Diffie-Hellman 키 분배 프로토콜, RSA를 이용한 키 분배 프로토콜 등 다양한 키 분배 프로토콜이 있지만 결국 단일 지점 장애(SPOF : Single Point of Failure)의 문제에 부딪친다. 또한, 키를 변화없이 사용하는 고정키 방식의 경우 한번 키가 탈취 당했을 시 지속적으로 공격을 받으며 동적키의 경우 지속적으로 키가 내트워크 상에 분배 되기 때문에 해킹의 위협에 자주 노출된다.

[0008] 종래의 동적키 생성 알고리즘으로는 블록키 생성 알고리즘 및 세션키 생성 알고리즘이 있다.

[0009] 먼저, 블록키 생성 알고리즘은 다음과 같다. 전처리 과정으로 히스토리 큐의 사이즈를 선택하고 Critical Cluster 및 Primary Message를 선택한다. 시스템이 시작하면 비휘발성 메모리인 EEPROM에 있는 내역들이 서로

같은지 해시를 통해 확인 후 휘발성 메모리인 레지스터에 전송한다. 송신측 ECU는 레지스터에 있는 키를 통해 MAC(Message Authentication Code)을 생성하여 원본 메시지와 함께 전송하게 된다. 메시지를 받은 다른 ECU들은 각각 이 메시지의 위조 및 변조를 확인 후 MAC을 레지스터에 있는 히스토리 큐에 쌓아놓는다. 이 과정을 반복한 후에 히스토리 큐가 다 쌓이게 되면 히스토리 큐에 있는 메시지들과 가지고 있는 키를 시드 값으로 하여 새로운 키를 생성한다. 시스템이 종료되면 레지스터에 있는 데이터를 해시 후 값이 같다면 EEPROM에 옮기고 종료한다.

[0010] 전술한 종래의 블록키 생성 알고리즘에서의 키 생성시 평균 메시지 전송 시간과 비 생성시 평균 메시지 전송시간의 차이를 비교하여 평균 오버헤드를 살펴보면, 매번 히스토리 큐에 메시지를 저장해야 하기에 사이즈 커지면 커질수록 절대적인 키 생성 시간은 증가하게 되는 문제점이 있으며, 또한, 히스토리 큐 사이즈가 증가함에 따라 메모리의 낭비가 필연적으로 생기는 문제점이 있다.

[0011] 다음, 세션키 생성 알고리즘은 다음과 같다. 도 3은 종래의 기술에 따른 세션키 생성 알고리즘을 도시한 흐름도이다. 도 3을 참조하면, 세션키 방식은 비밀키, 랜덤알고리즘, 해시 함수를 활용한다. 미리 공유되어 있는 비밀키와 랜덤알고리즘을 통해 새롭게 생성한 숫자를 해시 함수의 입력 값으로 사용해 키를 생성한다. 이후 AES(Advanced Encryption Standard) 알고리즘을 사용해 암호화된 데이터와 기존에 생성한 숫자를 넘겨준다. 수신자 측에선 받은 숫자와 기존에 가지고 있었던 비밀키를 해시 함수에 넣어 키를 생성한다. 생성한 키와 AES 알고리즘을 통해 암호화된 데이터를 복호화 시킨다. 세션키 방식의 경우 키를 분배하는 과정이 없다.

[0012] 그러나, 전술한 세션키 생성 알고리즘은, 고정된 비밀키를 가지고 있다는 점에서 보안성이 떨어진다고 볼 수 있다. 또한, 실시간성을 필수로 하는 차량 내부 네트워크에서 랜덤알고리즘과 해시 함수는 속도가 느려 매우 치명적이다.

[0013] 따라서, 본 발명에서는 분산 환경에서 실시간성과 키의 기밀성을 보장하는 동적키 생성 방법을 제안하고자 한다.

선행기술문헌

특허문헌

[0014] (특허문헌 0001) 한국등록특허공보 제 10-1704569호

발명의 내용

해결하려는 과제

[0015] 전술한 문제점을 해결하기 위한 본 발명의 목적은 분산 환경에서 실시간성과 키의 기밀성을 보장하는 동적키 생성 방법을 제공하는 것이다.

[0016] 본 발명의 다른 목적은, 컴퓨터로 판독가능한 기록매체로서, 분산 환경에서 실시간성과 키의 기밀성을 보장하는 동적키 생성 방법을 구현하는 프로그램이 기록된 기록매체를 제공하는 것이다.

과제의 해결 수단

[0017] 전술한 목적을 달성하기 위한 본 발명의 특징에 따른 키 생성 방법은, 다수 개의 처리 유닛들로 구성되는 분산 환경에서의 각 처리 유닛에 의한 키 생성 방법에 관한 것으로서, (a) 목표 카운트 값 및 MAC 사이즈를 설정하는 전처리 단계; (b) EEPROM 에 있는 데이터들을 레지스터로 옮기고 카운트 값을 초기화시키는 단계; (c) 메시지를 분산 환경으로 송신하는 경우, 자신의 비밀키를 이용하여 전송할 메시지와 비밀키를 이용하여 메시지 인증 코드(Message Authentication Code; 'MAC')를 생성하고, MAC과 메시지를 합쳐서, 분산 환경의 통신 네트워크에 브로드캐스팅하는 단계; (d) 브로드캐스팅된 메시지를 수신하는 경우, 원본 메시지와 MAC 을 분리하고, 원본 메시지로 MAC을 생성하며, 수신한 MAC과 생성된 MAC이 같은지를 비교하여 인가된 메시지인지 확인하고 검증하는 단계; (e) 수신된 메시지가 검증되면, 카운트 값을 1 증가시키는 단계; (f) 카운트 값이 목표 카운트값에 도달하면, 해당 순서의 메시지를 시드값으로 하여 새로운 키를 생성하는 단계; (g) 상기 새로운 키가 생성되면, 이전 키는 블록 암호화 기반의 키 생성 알고리즘의 초기 벡터(Initial Vector)에 저장하는 단계; (h) 종료되면 레지스터에 있는 카운트 값을 0으로 초기화시키고 EEPROM으로 데이터를 옮기고 종료하는 단계; 를 구비하고, 상기

목표 카운트 값은 시드값으로 사용할 메시지를 결정하는 값으로 사용된다.

[0018] 전술한 특징에 따른 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법에 있어서, 상기 (a) 전처리 단계는 중요 메시지를 설정하는 것을 특징으로 하며, 상기 (c) 및 (d) 단계의 MAC 인증 과정은 상기 설정된 중요 메시지에 대해서만 수행하여 메시지 처리 속도를 향상시키는 것이 바람직하다.

[0019] 전술한 특징에 따른 분산 환경에서의 메시지 순서를 기반으로 한 키 생성 방법에 있어서, 상기 키 생성 방법은 복수 개의 ECU들이 CAN 프로토콜에 따라 통신하는 차량 내부 네트워크에 적용되는 것이 바람직하다.

발명의 효과

[0020] 본 발명에 따른 동적키 생성 방법은 분산 환경에서 실시간성과 키의 기밀성을 보장하는 방법으로서, 랜덤 알고리즘과 해시 함수를 이용하지 않기 때문에 종래의 다른 키 생성 방식에 비해 빠른 키 생성 속도를 보여준다.

[0021] 또한, 본 발명에 따른 동적키 생성 방법은, 종래의 다른 동적키 생성 방법과는 달리, 키 분배가 없고 키 생성의 시드 값으로 메시지를 이용하여 키를 생성하므로, 일반적으로 시드 값을 예측할 수 없도록 하여 키의 기밀성을 높일 수 있다.

[0022] 한편, 차량 내부 네트워크를 구성하는 CAN에서 가장 중요한 두 가지는 실시간성과 보안이다. 본 발명에 따른 동적키 생성 방법은 다른 종래의 방법들보다 우수한 결과를 보여준다. 실시간성의 경우, 본 발명에 따른 키 생성 방법은 기본적으로 속도가 가장 빠를 뿐만 아니라 키 생성 주기의 영향 없이 일정한 성능을 유지하고 있다. 또한, 키의 기밀성을 위해 메시지를 시드 값으로 활용하여 키 분배 없이 동적키를 생성하고 있다. 추가적으로 메모리의 사용량도 종래의 다른 방법들보다 낮아 저성능 마이크로 컨트롤러에도 적합함을 보여준다.

도면의 간단한 설명

[0023] 도 1은 종래의 차량 내부 전자제어장치들인 ECU들간의 직렬 통신 방식인 UART 통신을 도시한 모식도이다.

도 2는 종래의 차량 내부 전자제어장치들인 ECU들간의 통신 방식인 CAN 통신을 도시한 모식도이다.

도 3은 종래의 기술에 따른 세션키 생성 알고리즘을 도시한 흐름도이다.

도 4는, 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 전처리 항목들을 도시한 블록도이다.

도 5는 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법을 전체적으로 도시한 흐름도이다.

도 6은 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, MAC 인증 과정을 설명하기 위하여 도시한 모식도이다.

도 7은 본 발명에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 성능을 실험하기 위한 실험 환경을 도시한 구성도이다.

도 8은 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 실험 결과 중 평균 키 생성 속도를 도시한 비교 그래프이다.

도 9는 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 실험 결과 중 키 생성 주기에 따른 비교 그래프이다.

도 10은 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 실험 결과 중 기밀성을 비교한 도표이다.

발명을 실시하기 위한 구체적인 내용

[0024] 본 발명에 따른 키 생성 방법은 메시지 순서를 기반으로 하여 키를 생성함으로써, 분산 환경에서 실시간성과 키의 기밀성을 보장하게 된다. 특히, 본 발명에 따른 생성 방법은 랜덤 알고리즘과 해시 함수를 이용하지 않아 다른 키 생성 방식에 비해 빠른 키 생성 속도를 보여주며, 키 분배가 없고 키 생성의 시드 값으로 메시지를 이용하여 키를 생성해 일반적으로 시드 값을 예측할 수 없도록 하여 키의 기밀성을 높여 주게 된다.

[0025] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성

방법에 대하여 구체적으로 설명한다.

[0026] 도 4는, 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 전처리 항목들을 도시한 블록도이다. 도 4를 참조하면, 먼저 목표 카운트값을 설정하고, 메시지 인증 코드(Message Authentication Code;'MAC')의 크기를 설정하고, 중요 메시지를 선정하고, 암호화 알고리즘을 선정한다.

[0027] 목표 카운트 값은 시드값으로 사용할 메시지를 결정하는 값으로 사용되는 것으로서, 이전 키 생성이후 몇 번째 메시지를 시드값으로 해서 키를 생성할 것인지를 말한다. 만약 목표 카운트 값이 낮게 설정되면 키를 자주 생성하게 되므로 키의 기밀성은 높아지나 메시지 전송에 있어 오버헤드가 더 증가하게 된다. 반대로 목표 카운트 값을 높게 설정하면, 메시지 처리 속도는 증가되나 키의 기밀성이 떨어지게 된다.

[0028] 다음, 메시지를 전송할 때 데이터 필드의 일정 부분을 MAC으로 채워야 하며, 이때 필요한 MAC 크기를 전처리에서 설정하게 된다. MAC(Message Authentication Code; 메시지 인증 코드)은 메시지의 인증을 위해 메시지에 부가되어 전송되는 작은 크기의 정보로서, 비밀키를 사용하여 메시지의 무결성 및 인증 서비스를 제공하게 된다. 메시지 인증 코드는 임의의 길이의 메시지와 송신자 및 수신자가 공유하는 비밀키를 기초로 하여 MAC 알고리즘으로 처리하여 생성된 코드로서, 메시지와 함께 전송한다. 차량 내부 네트워크 환경에서 사용되는 CAN 프로토콜에 있어서, 데이터 프레임에서 메시지로 활용 가능한 데이터 필드의 크기는 8바이트 밖에 되지 않는다. 따라서, 보안과 메시지의 용량의 트레이드 오프가 발생하게 되는데, 이러한 점을 고려하여 MAC 크기를 설정하여야 한다.

[0029] 한편, 차량 내부 네트워크 환경에서는 무엇보다 실시간성이 중요하며 신속한 처리가 요구된다. 따라서, 본 발명에서는 모든 메시지 송수신 과정에서 MAC 인증을 거치게 하는 것이 아니라, 중요 메시지를 사전 선정하고, 이렇게 선정된 중요 메시지에 대해서만 MAC 인증 과정을 거치도록 함으로써, 메시지 처리 속도를 향상시키게 된다.

[0030] 또한, 본 발명에 따른 키 생성 방법의 전처리 단계는, 컨트롤러의 성능에 맞게 MAC 알고리즘 및 키 생성 알고리즘을 선택해야 된다. 고성능 컴퓨팅 환경에서 개발되거나 경량화된 알고리즘들은 저성능 컴퓨팅 환경에서 급격한 연산 속도 저하가 발생할 수 있다. 예를 들면, AES 알고리즘은 저성능 컴퓨팅 환경에서 개발되어 8 비트 연산을 하므로, 저성능 마이크로 컨트롤러인 아두이노 우노에서 더 적합한 성능을 보여 주게 된다.

[0031] 이하, 도 5를 참조하여 전술한 전처리 단계를 거친 이후의 본 발명에 따른 키 생성 방법에 대하여 구체적으로 설명한다.

[0032] 도 5는 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법을 전체적으로 도시한 흐름도이다. 도 5를 참조하면, 본 발명에 따른 키 생성 방법은, 다수 개의 처리 유닛들로 구성되는 분산 환경에서의 각 처리 유닛에 의해 수행되며, 특히, 복수 개의 ECU들이 CAN 프로토콜에 따라 통신하는 차량 내부 네트워크과 같은 분산 환경 네트워트에 적용될 수 있다.

[0033] 본 발명에 따른 키 생성 방법은, 앞서 설명한 바와 같이, 사전에 목표 카운트 값, MAC 사이즈를 설정하고, 중요 메시지를 선정하고, 암호화 알고리즘을 선정하는 전처리 단계를 수행한다.

[0034] 다음, 도 5를 참조하면, 시스템이 시작되면, EEPROM 에 있는 데이터들을 레지스터로 옮기고 카운트 값을 초기화시킨다(단계 500).

[0035] 다음, MAC 인증 과정을 통해 메시지를 송수신하게 된다(단계 510). 이하, 도 6을 참조하여 본 발명에 따른 키 생성 방법에 사용되는 MAC 인증 과정을 구체적으로 설명한다.

[0036] 도 6은 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, MAC 인증 과정을 설명하기 위하여 도시한 모식도이다. 도 6을 참조하면, 메시지를 분산 환경으로 송신하는 Sender는, 자신의 비밀키를 이용하여 전송할 메시지와 비밀키를 이용하여 메시지 인증 코드(Message Authentication Code;'MAC')를 생성하고, 상기 생성된 MAC과 메시지를 합쳐서, 분산 환경의 통신 네트워크에 브로드캐스팅한다.

[0037] 한편, 브로드캐스팅된 메시지를 수신한 Receiver는, 원본 메시지와 MAC 을 분리하고, 자신이 갖고 있는 비밀키를 이용하여 원본 메시지로 MAC을 생성하며, 상기 수신한 MAC과 상기 생성한 MAC이 같은지를 비교하고, 이들이 서로 동일하면 인가된 메시지인지 확인하고 검증한다. 수신된 메시지가 검증되면, 카운트 값을 1 증가시킨다.

[0038] 다음, 카운트 값이 목표 카운트값(k)에 도달하면(단계 515), 해당 순서의 메시지를 시드값으로 하여 새로운 키를 생성하고(단계 520), 카운트 값은 다시 0으로 초기화시키고(단계 530), 이전 키는 블록 암호화 기반의 키 생성 알고리즘의 초기 벡터(Initial Vector;IV)에 저장한다(단계 540).

[0039] 상기 시스템이 종료되면(단계 550), 레지스터에 있는 카운트 값을 0으로 초기화시키고(단계 560), EEPROM으로 데이터를 옮기고 종료한다(단계 570).

[0040] 한편, 본 발명에 따른 키 생성 방법은 상기 전처리 단계에서 중요 메시지를 선정하고, 상기 MAC 인증 과정(단계 510)은 상기 설정된 중요 메시지에 대해서만 수행하도록 함으로써, 메시지 처리 속도를 향상시키는 것이 바람직하다.

[0041] 전술한 본 발명에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법은 실행 가능한 프로그램으로 구현되어, 컴퓨터로 읽을 수 있는 기록 매체에 기록될 수 있다.

[0042] 이하, 전술한 본 발명에 따른 키 생성 방법의 성능에 대하여 설명한다. 먼저, 본 발명에 따른 키 생성 방법의 성능을 실험하기 위한 실험 환경은, 도 7에 도시된 바와 같이, 아두이노 우노를 ECU로 두고 CAN Bus Shield를 사용하였다. 도 7은 본 발명에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 성능을 실험하기 위한 실험 환경을 도시한 구성도이다.

[0043] 실험 및 검증을 위해 본 발명에 따른 키 생성 방법에 대한 프로토타입을 개발하여, 한 개의 ECU가 지속적으로 메시지를 송신하도록 하였으며, 아두이노 우노의 낮은 성능을 고려하여 CBC(Cipher-Block Chaining) 모드를 활용한 AES128 알고리즘을 사용했다.

[0044] 실험 내용으로는, 차량 내부 네트워크는 실시간성과 보안이 중요하므로, 실시간성과 키의 기밀성을 분석하는 것으로 나누었다. 실험 내용은 아래의 세 가지이다.

[0045] 첫째, 동일한 실험 환경에서 순수한 키 생성 속도를 비교한다. 일반적인 동적키 방식의 경우 키 생성 주기에 따라 메시지 처리 속도에 대한 오버 헤드가 달라지지 않기에 생성 주기가 같다고 가정하고 실험하였다. 특히, 메시지를 시드 값으로 키를 동적으로 생성하는 블록키 생성 알고리즘과 본 발명에 따른 키 생성 방법의 성능을 추가적으로 비교한다.

[0046] 둘째, 본 발명에 따른 키 생성 방법과 관련 연구의 키 기밀성에 대해 비교한다. 종래의 동적키 방식은 키의 분배가 필수적이기에 KDC나 CA기관에 의존하여 키 분배 프로토콜이 진행된다. 따라서 현실적으로 차량 내부 네트워크에 적용이 불가능하다고 보고 본 발명에 따른 방법, 블록키, 세션키 방식을 비교한다.

[0047] 마지막으로는, 셋째, 본 발명에 따른 키 생성 방법과 블록키 방식의 메모리 사용량을 비교한다. 일반적으로 임베디드 환경은 PC보다 한정된 자원을 가지고 있어 메모리의 사용량이 더욱 중요하다고 볼 수 있기 때문이다.

[0048] 위의 실험 결과는 아래와 같다.

[0049] 도 8은 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 실험 결과 중 평균 키 생성 속도를 도시한 비교 그래프이다. 도 8에서, 각각 3만번의 키 생성 속도의 평균을 마이크로 세컨드로 표기하였다. 도 8을 참조하면, 본 발명에 따른 키 생성 방법은 카운트 값을 16으로 설정하였으며, 블록키 방식은 히스토리 큐 사이즈를 16으로 설정하였다. 각 키의 평균 생성 속도를 비교해 보았을 때, 제안된 방법의 속도가 가장 빠른 것으로 나타났다. 블록키의 경우 메시지가 올 때마다 히스토리 큐에 메시지를 쌓아야 하기 때문에, 해당 Count Value의 메시지를 시드 값으로 사용하는 제안된 방법보다 140.45us가 높게 나왔다. 랜덤키 방식의 경우 랜덤 알고리즘의 처리 속도가 다소 느리기에 제안된 방법의 약 4.74배의 차이를 보여주고 있다. 세션키 방식의 경우 랜덤 알고리즘과 해시 함수를 사용하여 다른 키들보다 압도적으로 느린 결과를 보여주었다.

[0050] 도 9는 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 실험 결과 중 키 생성 주기에 따른 비교 그래프이다. 도 9에서, 괄호 안에 나타나 있는 숫자는 Count Value와 히스토리 큐 사이즈를 나타내어 키의 생성 주기를 보여주고 있다. 도 8에서 블록키의 경우 본 발명에 따른 키 생성 방법과 차이가 적어 보이지만 히스토리 큐 사이즈를 증가시켰을 때 키 생성 속도가 선형적으로 증가한다는 것을 알 수 있다. 블록키의 경우 히스토리 큐 사이즈가 커지는 만큼 큐에 메시지를 쌓는 작업도 증가하지만, 본 발명에 따른 키 생성 방법의 경우 Count Value에 상관 없이 일정한 키 생성 속도를 보여준다. 따라서 본 발명에 따른 키 생성 방법이 키 생성 속도에 있어 가장 좋은 결과를 보여줌을 알 수 있다.

[0051] 도 10은 본 발명의 바람직한 실시예에 따른 분산 환경에서의 메시지 순서 기반의 키 생성 방법에 있어서, 실험 결과 중 기밀성을 비교한 도표이다. 도 10에 따르면, 본 발명에 따른 키 생성 방법(Count Key 방식)과 블록키, 세션키 방식은 모두 키를 네트워크상에 전송하진 않으며 시드 값을 송수신한다는 점에서 같다. 세션키의 경우 시드 값인 난수를 직접적으로 네트워크에 전달하기 때문에 해커가 네트워크에 침입했을 때 시드 값을 인지하기

쉽다.

[0052] 반면, 본 발명에 따른 키 생성 방법과 블록키는 메시지를 시드 값으로 활용하여 ECU 내부에 존재하는 Count Value 혹은 히스토리 큐 사이즈를 알아야 시드 값을 키 생성에 활용할 수 있다. 추가적으로 키의 기밀성을 위해 Count Value나 히스토리 큐 사이즈를 동적으로 변경할 수도 있다. 따라서, 본 발명에 따른 키 생성 방법과 블록키 방식은 해커가 네트워크에 침입한 상황에서도 시드 값을 인지하고 활용하기에 어려워 키의 기밀성이 높다고 볼 수 있다.

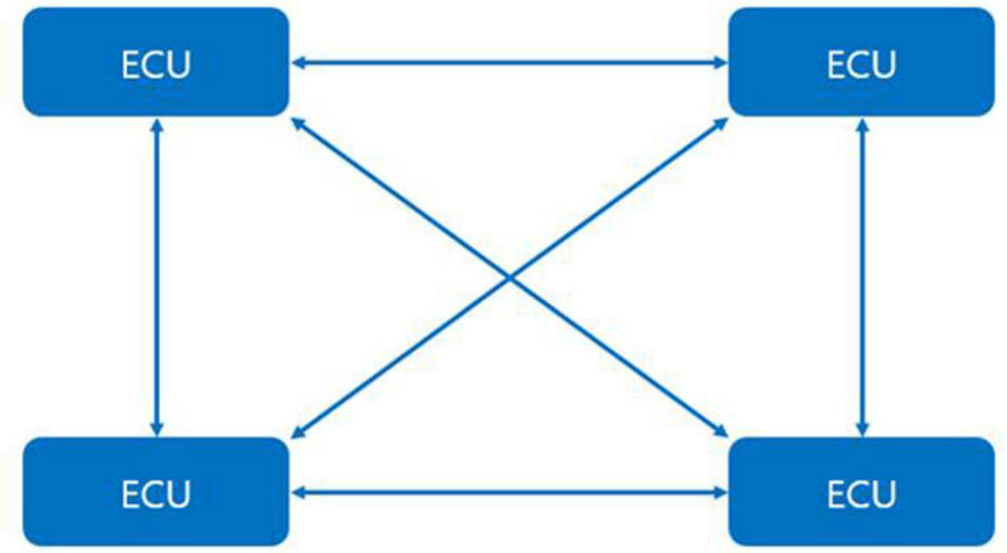
[0053] 마지막으로, 메모리 사용량에 있어서, 본 발명에 따른 키 생성 방법은 IV로 사용되는 128비트와 메시지 64비트를 사용하여 총 192bit가 키 생성 알고리즘에 소요된다. 반면 블록키의 경우 k가 히스토리 큐 사이즈라고 가정했을 때, 64×k 비트만큼 메모리가 소요된다. 히스토리 사이즈가 커지면 커질수록 메모리도 64비트씩 증가하게 된다. 따라서 $k \geq 4$ 이면 제안된 방법이 블록키에 비해 메모리 사용량이 적다는 것을 보여준다.

[0054] 차량 내부 네트워크를 구성하는 CAN에서 가장 중요한 두 가지는 실시간성과 보안이다. 전술한 바와 같이, 본 발명에 따른 동적키 생성 방법은 다른 종래의 방법들보다 우수한 결과를 보여준다. 실시간성의 경우, 본 발명에 따른 키 생성 방법은 기본적으로 속도가 가장 빠를 뿐만 아니라 키 생성 주기의 영향 없이 일정한 성능을 유지하고 있다. 또한, 키의 기밀성을 위해 메시지를 시드 값으로 활용하여 키 분배 없이 동적키를 생성하고 있다. 추가적으로 메모리의 사용량도 종래의 다른 방법들보다 낮아 저성능 마이크로 컨트롤러에도 적합함을 보여준다. 단, 사용하는 컨트롤러의 성능에 맞게 암호화 알고리즘을 사용해야 하고 실시간성과 보안의 트레이드 오프 관계를 고려해서 키 생성 주기를 설정해야 한다.

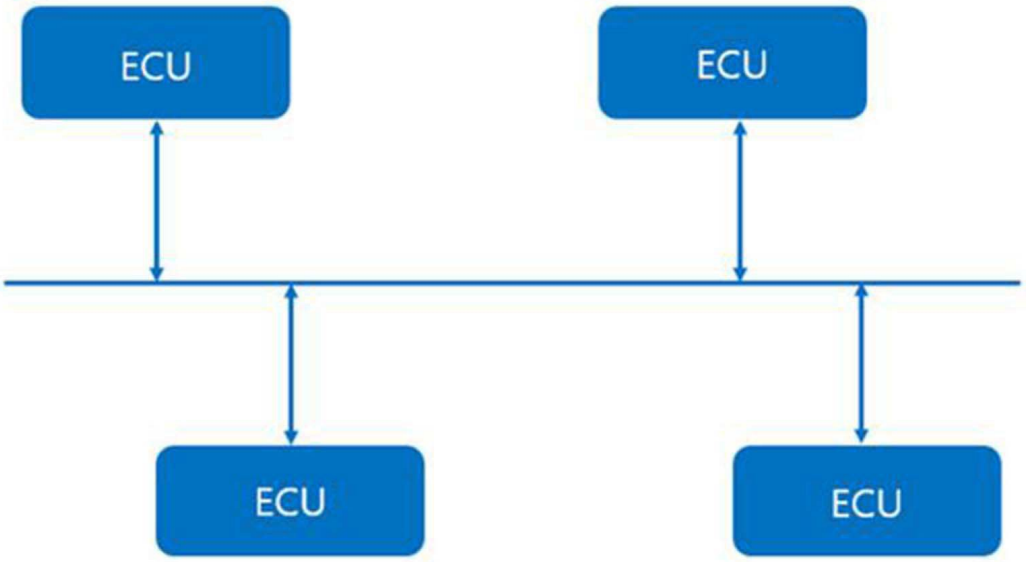
[0055] 이상에서 본 발명에 대하여 그 바람직한 실시예를 중심으로 설명하였으나, 이는 단지 예시일 뿐 본 발명을 한정하는 것이 아니며, 본 발명이 속하는 분야의 통상의 지식을 가진 자라면 본 발명의 본질적인 특성을 벗어나지 않는 범위에서 이상에 예시되지 않은 여러 가지의 변형과 응용이 가능함을 알 수 있을 것이다. 그리고, 이러한 변형과 응용에 관계된 차이점들은 첨부된 청구 범위에서 규정하는 본 발명의 범위에 포함되는 것으로 해석되어야 할 것이다.

도면

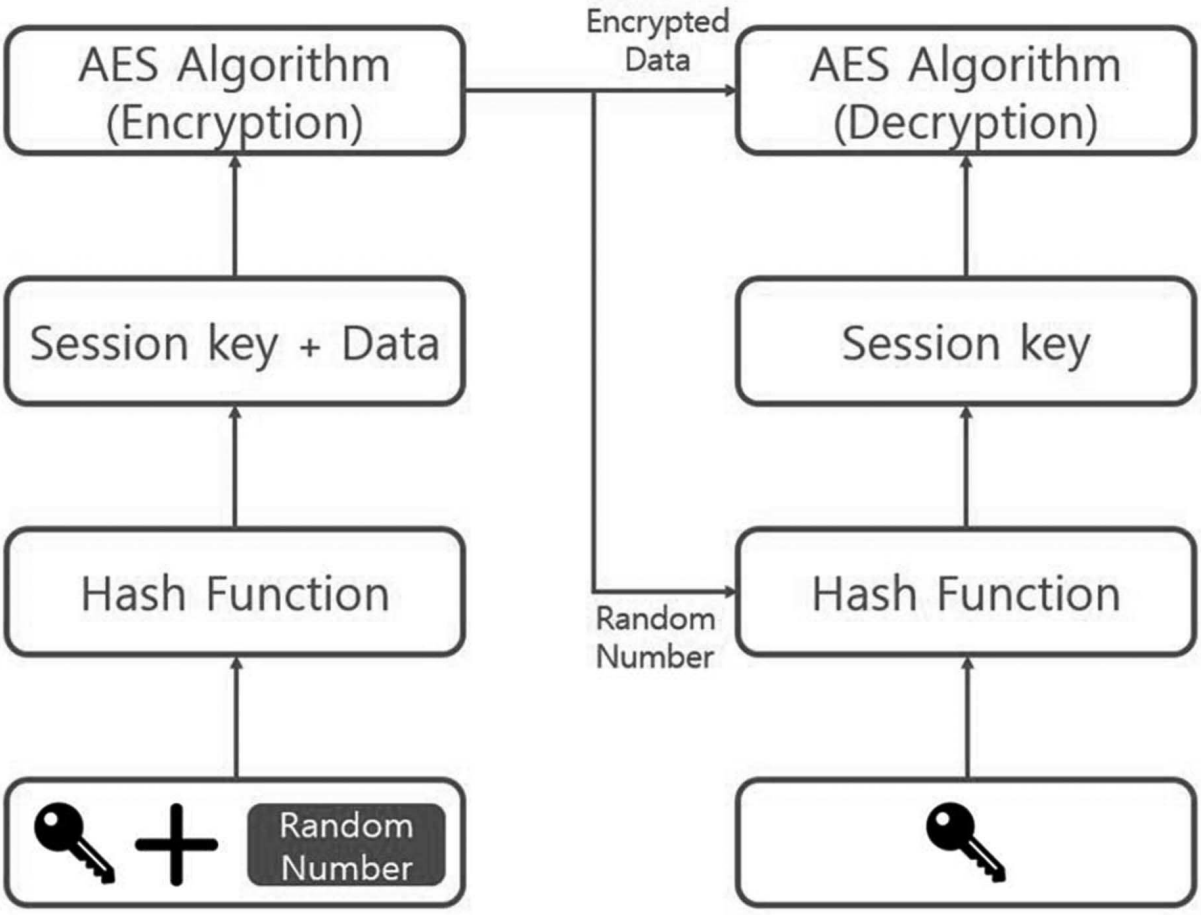
도면1



도면2

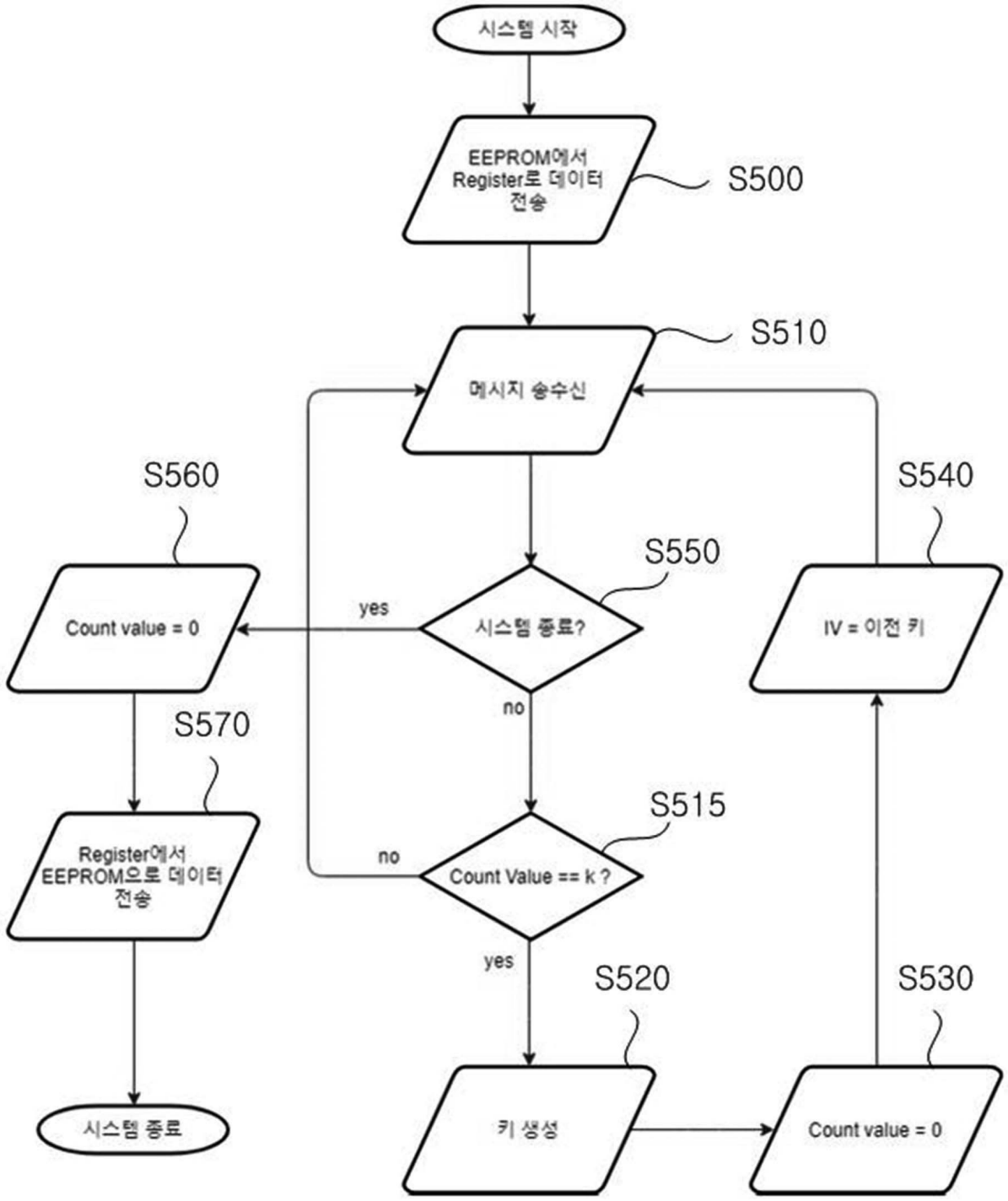

ECU
ECU
ECU
ECU

도면3

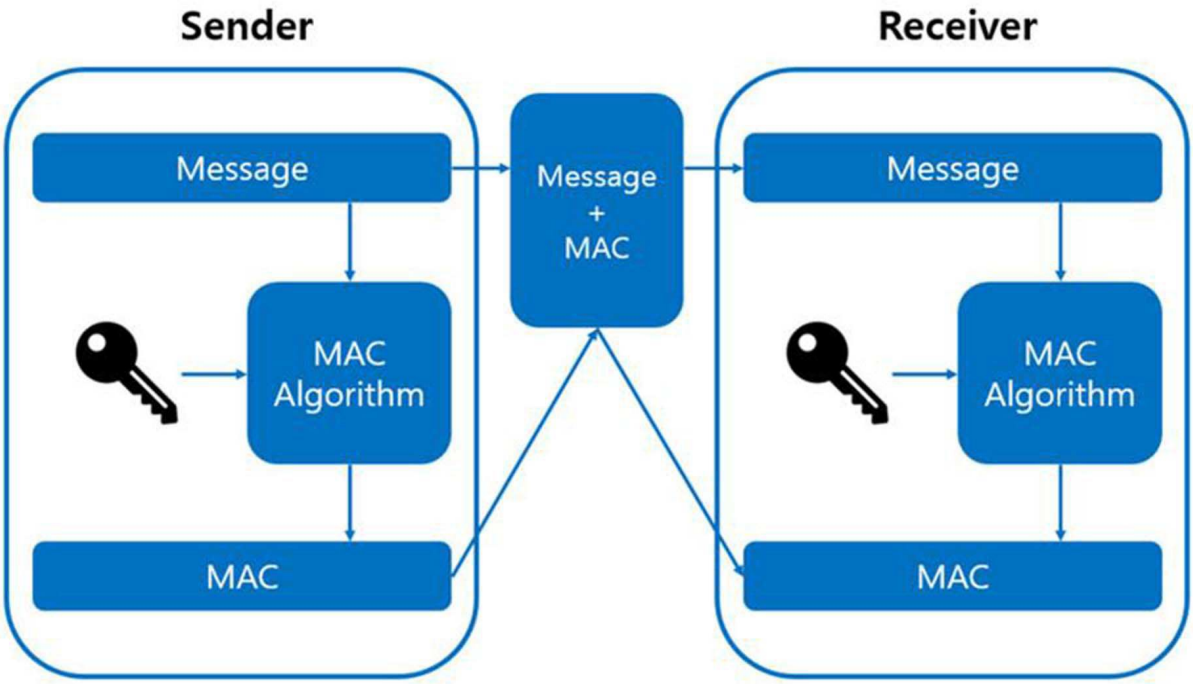

AES Algorithm (Encryption)
Encrypted Data
AES Algorithm (Decryption)
Session key + Data
Session key
Hash Function
Random Number
Hash Function
Random Number

도면4

도면5


시스템 시작
EEPROM에서 Register로 데이터 전송
S500
S510
메시지 송수신
S560
S540
S550
Count value = 0
yes
시스템 종료?
IV = 이전 키
no
S570
S515
Register에서 EEPROM으로 데이터 전송
no
Count Value == k ?
yes
S520
S530
키 생성
Count value = 0
시스템 종료

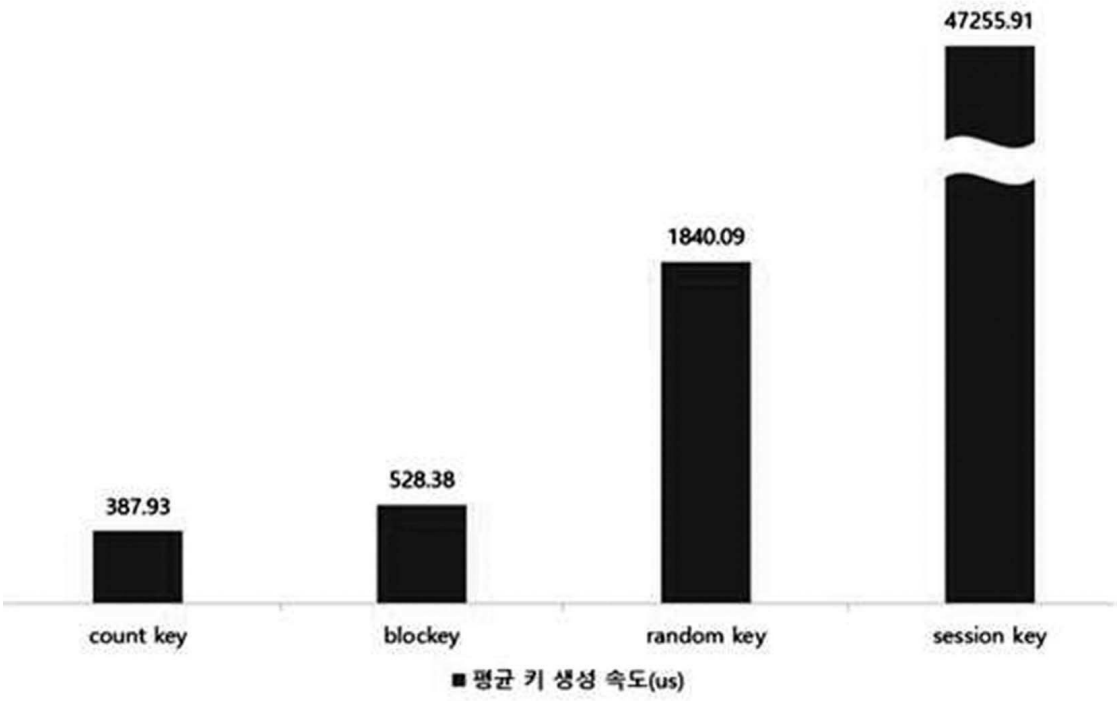
도면6


Sender
Receiver
Message
MAC Algorithm
MAC
Message + MAC
Message
MAC Algorithm
MAC

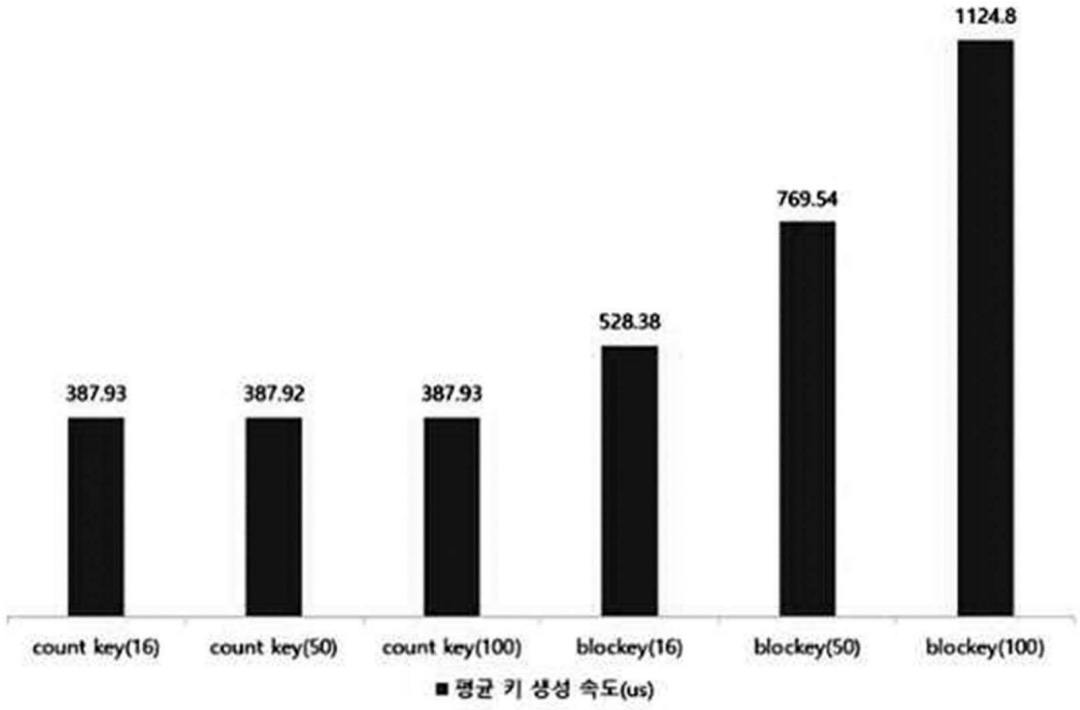
도면7


BUS
CAN BUS Shield
ARDUINO UNO
CAN BUS Shield
ARDUINO UNO
CAN BUS Shield
ARDUINO UNO

도면8


47255.91
1840.09
528.38
387.93
count key
blockey
random key
session key
■ 평균 키 생성 속도(us)

도면9


1124.8
769.54
528.38
387.93
387.92
387.93
count key(16)
count key(50)
count key(100)
blockey(16)
blockey(50)
blockey(100)
■ 평균 키 생성 속도(us)

도면10

	키 분배 여부	시드 값 분배 여부	시드
Count Key	X	O	메시지
Blockey	X	O	메시지
Session Key	X	O	난수