(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0091969
(43) 공개일자 2015년08월12일

(51) 국제특허분류(Int. Cl.)
*H04L 9/32* (2006.01) *H04L 9/14* (2006.01)
(21) 출원번호 10-2014-0174273
(22) 출원일자 2014년12월05일
심사청구일자 없음
(30) 우선권주장
377/MUM/2014 2014년02월03일 인도(IN)

(71) 출원인
**타타 컨설턴시 서비시즈 리미티드**
인도 마하라슈트라-400021 뭄바이 나리만 포인트 나인쓰 플로어 니르말 빌딩
(72) 발명자
**바타차리아, 아브히잔**
인도, 웨스트 벵갈, 콜카타-700156, 라자르햇, 뉴타운, 에코스페이스 플롯-아이아이에프/12, 빌딩 1비, 큐비클 4비-34, 타타 컨설턴시 서비시즈
**반디오파디아이, 소마**
인도, 웨스트 벵갈, 콜카타-700156, 라자르햇, 뉴타운, 에코스페이스 플롯-아이아이에프/12, 빌딩 1비, 큐비클 4비-40, 타타 컨설턴시 서비시즈
*(뒷면에 계속)*
(74) 대리인
**특허법인 아이퍼스**

전체 청구항 수 : 총 20 항

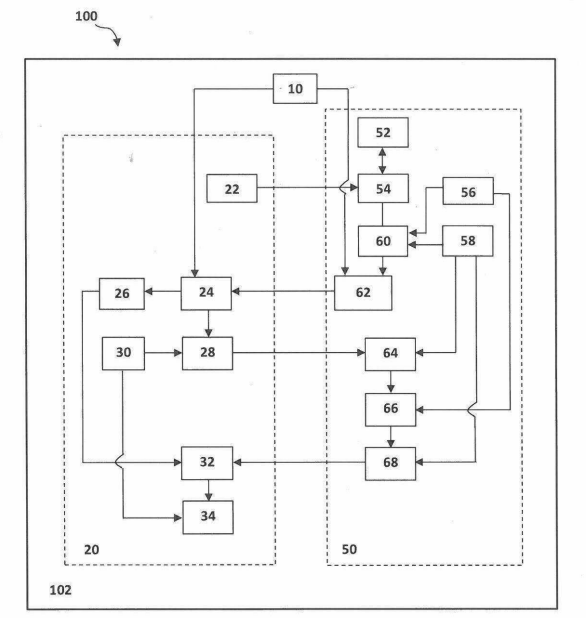
(54) 발명의 명칭 **사물인터넷을 위한 데이터그램 전송에서 경량 인증을 위한 컴퓨터 구현 시스템 및 방법**

## *(57) 요 약*

사물인터넷을 위한 데이터그램 전송에서 경량 인증을 위한 컴퓨터 구현 시스템과 방법은 예비-공유 비밀을 공유하는 두개의 엔드포인트간의 챌린지-응답형 교환에 기반한 견고한 인증 스킴을 제공한다. 본 발명에서는 인증과 키 관리와 통합된 대칭 키-기반 보안 메커니즘이 이용된다. 이는 준비단계과정에서 시스템의 엔드포인트들에 예

(뒷면에 계속)

*대 표 도* - 도1



비-공유 비밀이 제공되고 서버 측에는 클라이언트 식별을 위해 클라이언트 데이터베이스가 제공되는 것을 특징으로 하는 상호 인증을 제공한다. 시스템은 난수 생성을 위한 임의 번호 생성기와 비밀키와 세션 키를 생성하기 위한 키 생성기를 포함한다. 난수와 키는 오직 세션과정에서만 유효하며 따라서 세션들 전체에서 안전한 인증을 제공하는데 도움을 준다.

시스템은 또한 DTLS와 같은 전송 레이어 프로토콜에 맞게 개작될수 있으며 제한 장치들을 위한 CoAP와 같은 애플리케이션 레이어 프로토콜에 통합될수 있다.

(72) 발명자

**유킬, 아리지트**

인도, 웨스트 벵갈, 콜카타-700156, 라자르햇, 뉴 타운, 에코스페이스 플롯-아이아이에프/12, 빌딩 1비, 큐비클 4비-33, 타타 컨설턴시 서비시즈

**팔, 아르판**

인도, 웨스트 벵갈, 콜카타-700156, 라자르햇, 뉴 타운, 에코스페이스 플롯-아이아이에프/12, 빌딩 1비, 4층, 타타 컨설턴시 서비시즈

# *명 세 서*

## *청구범위*

### 청구항 1

데이터그램 전송에서 서버와 클라이언트간의 양방향 인증을 위한 컴퓨터 구현 시스템에 있어서, 상기 시스템은 시스템 프로세서를 포함하고 그리고:

i. 시스템 프로세서와 협력하며 제1 임의 번호를 생성하도록 구성된 제1 임의 번호 생성기;

ii. 시스템 프로세서와 협력하며 제2 임의 번호를 생성하도록 구성된 제2 임의 번호 생성기;

iii. 시스템 프로세서와 협력하며 준비단계 과정에서 양방향 인증 전에 시스템 프로세서의 생성 명령과 전송 명령하에서 비밀키를 서버와 클라이언트에게 생성 및 전송하도록 구성된 비밀키 생성기;

iv. 모든 클라이언트들의 클라이언트 ID를 저장하도록 구성된 제1 저장부;

v. 시스템 프로세서의 전송 명령하에서 클라이언트로부터 클라이언트의 고유 ID를 포함한 제1 메시지를 서버에게 전송하도록 구성된 세션 개시자;

vi. 시스템 프로세서와 협력하여 시스템 프로세서의 수신 명령하에서 제1 메시지를 수신하며 매칭 엔진이 구비되어 있어 수신한 클라이언트 ID를 제1 저장부에 저장된 클라이언트 ID와 매칭하여 클라이언트를 식별하는 수신기;

vii. 고유하게 시간 제한된 세션 키를 생성하고 생성된 세션 키를 시스템 프로세서의 전송 명령하에 전송하도록 구성된 세션 키 생성기;

viii. 시스템 프로세서와 협력하며 세션 키를 수신하며 제1 임의 번호 생성기와 세션 키에 의해 생성된 제1 임의 번호를 포함하는 챌린지 코드를 시스템 프로세서의 생성 명령하에서 생성하며 시스템 프로세서의 전송 명령하에서 전송하도록 구성된 챌린지 코드 생성기;

ix. 챌린지 코드 생성기와 협력하여 생성된 챌린지 코드를 수신하며 시스템 프로세서로부터의 명령에 응답하여 수신한 생성된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀키를 이용하여 암호화하며 또한 시스템 프로세서의 전송 명령하에서 암호화된 챌린지 코드를 식별된 클라이언트에게 전송하도록 구성된 제1 암호기;

x. 시스템 프로세서와 협력하며 암호화된 챌린지코드를 수신하도록 구성되었으며 또한 시스템 프로세서로부터의 명령에 응답하여 암호화된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀키를 이용하여 복호화하여 복호화된 제1 임의 번호와 세션 키를 획득하도록 구성된 제1 복호기;

xi. 제1 복호기로부터 세션 키를 수신하고 저장하도록 구성된 제2 저장부;

xii. 시스템 프로세서와 협력하며 복호화된 제1 임의 번호와 세션 키를 수신하도록 구성되었으며 또한 시스템 프로세서의 전송 명령 하에서 복호화된 제1 임의 번호와 제2 임의 번호 생성기에 의해 생성된 제2 임의 번호를 포함한 세션 키에 의해 암호화된 제2 메시지를 전송하도록 구성된 제2 복호기;

xiii. 시스템 프로세서와 협력하며 제2 메시지를 수신하고 시스템 프로세서로부터의 명령에 응답하여 세션 키 생성기에 의해 생성된 세션 키를 이용하여 제1 임의 번호와 제2 임의 번호를 복호화하도록 구성된 제2 복호기;

xiv. 시스템 프로세서로부터의 명령에 응답하여 제2 메시지로부터 복호화된 제1 임의 번호를 제1 임의 번호 생성기에 의해 생성된 제1 임의 번호와 비교하여 클라이언트를 인증하도록 구성된 제1 비교기와 인증기;

xv. 시스템 프로세서로부터의 명령에 응답하여, 제2 메시지에서 수신한 제2 임의 번호를 세션 키 생성기에 의해 생성된 세션 키를 이용하여 암호화하고 시스템 프로세서의 전송 명령하에서 암호화된 제2 임의 번호를 전송하도록 구성된 제3 암호기;

xvi. 시스템 프로세서로부터의 명령을 받고 암호화된 제2 임의 번호를 수신하고 시스템 프로세서의 명령하에서 제2 저장부로부터 수신한 세션 키를 이용하여 복호화하도록 구성된 제3 복호기; 및

xvii. 시스템 프로세서로부터의 명령에 응답하여 복호화된 제2 임의 번호를 제2 임의 번호 생성기에 의해 생성

된 제2 임의 번호와 비교하여 서버를 인증하고 상호 인증을 달성하도록 구성된 제2 비교기와 인증기로 이루어진 것을 특징으로 하는 시스템.

**청구항 2**

제1항에 있어서,

제1 임의 번호 생성기는 시스템 프로세서로부터의 명령에 응답하는 그리고 제1 타이머 값을 생성하도록 구성된 제1 타이머를 포함하는 것을 특징으로 하는 시스템.

**청구항 3**

제1항에 있어서,

제1 임의 번호 생성기에 의해 생성된 제1 임의 번호는 제1 타이머 값이 추가된 제1 의사 임의 번호를 포함하는 것을 특징으로 하는 시스템.

**청구항 4**

제1항에 있어서,

제2 임의 번호 생성기는 시스템 프로세서로부터의 명령에 응답하는 그리고 제2 타이머 값을 생성하도록 구성된 제2 타이머를 포함하는 것을 특징으로 하는 시스템.

**청구항 5**

제1항에 있어서,

제2 임의 번호 생성기에 의해 생성된 제2 임의 번호는 제2 타이머 값이 추가된 제2 의사 임의 번호를 포함하는 것을 특징으로 하는 시스템.

**청구항 6**

제1항에 있어서,

비밀키 생성기에 의해 생성된 비밀키는 세션 시작에서 생성된 고유키이며 오직 진행 세션에서만 유효한 것을 특징으로 하는 시스템.

**청구항 7**

제1항에 있어서,

세션 키 생성기는 시스템 프로세서로부터의 명령에 응답하는 세션 키 타이머를 포함하며 세션 키 타이머 값이 만료되면 생성된 세션 키를 취소하고 새로운 세션의 설립 요구를 나타내도록 구성된 것을 특징으로 하는 시스템.

**청구항 8**

제1항에 있어서,

제1 임의 번호 생성기와 제2 임의 번호 생성기에 의해 생성된 임의 번호들은 복제할수 없으며 서로 다른 세션마다에서 변경되는 것을 특징으로 하는 시스템.

**청구항 9**

제1항에 있어서,

클라이언트는 서버가 클라이언트의 요청 실행 상태에서 응답하지 않는 방식으로 서버와 통신할 수 있는 것을 특징으로 하는 시스템.

**청구항 10**

제1항에 있어서,

시스템은 DTLS를 포함하는 전송 레이어 보안 스킴과 통합될 수 있는 것을 특징으로 하는 시스템.

**청구항 11**

제1항에 있어서,

시스템은 제한 장치들을 위한 CoAP를 포함하는 애플리케이션 레이어 프로토콜과 통합될수 있는 것을 특징으로 하는 시스템.

**청구항 12**

서버와 클라이언트간의 데이터그램 전송에서 양방향 인증을 위한 컴퓨터 구현 방법에 있어서, 상기 방법은 시스템 프로세싱 명령을 포함하며 그리고 하기의 단계들;

• 제1 임의 번호 생성기의 도움으로 제1 임의 번호를 생성하는 단계;

• 제2 임의 번호 생성기의 도움으로 제2 임의 번호를 생성하는 단계;

• 비밀키 생성기의 도움으로 시스템 프로세싱 명령에 응답하여 비밀키를 생성하는 단계;

• 준비 단계과정에서 양방향 인증 전에 시스템 프로세싱 명령에 응답하여 생성된 비밀키를 서버와 클라이언트에게 전송하는 단계;

• 제1저장부에 모든 클라이언트들의 클라이언트 ID를 저장하는 단계;

• 시스템 프로세싱 명령에 응답하여 클라이언트의 고유 ID를 포함한 제1 메시지를 전송하는 단계;

• 시스템 프로세싱 명령에 응답하여 제1 메시지를 수신하고 수신한 클라이언트 ID를 제1 저장부의 저장된 클라이언트 ID와 매칭하는 단계;

• 수신한 클라이언트 ID에 기초하여 클라이언트를 식별하는 단계;

• 세션 키 생성기의 도움으로 고유하게 시간 제한된 세션 키를 생성하는 단계;

• 세션 키를 수신하고 시스템 프로세싱 명령하에서 제1 임의 번호 생성기와 세션 키에 의해 생성된 제1 임의 번호를 포함한 챌린지 코드를 생성하는 단계;

• 챌린지 코드 생성기로부터 생성된 챌린지 코드를 수신하고 시스템 프로세싱 명령에 응답하여 제1 암호기의 도움으로 수신한 생성된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀 키로 암호화하며 암호화된 챌린지 코드를 시스템 프로세싱 명령에 응답하여 전송하는 단계;

• 제1 임의 번호와 세션 키를 획득하기 위해 암호화된 챌린지 코드를 수신하고 시스템 프로세싱 명령에 응답하여 암호화된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀키를 이용하여 제1 복호기의 도움으로 복호화하는 단계;

• 제1 복호기로부터 세션 키를 수신하고 제2 저장부에 저장하는 단계;

• 시스템 프로세싱 명령에 응답하여 복호화된 제1 임의 번호와 세션 키를 수신하고 제2 임의 번호 생성기에 의해 생성된 제1 임의 번호와 제2 임의 번호를 포함한 세션 키를 이용하여 암호화된 제2 메시지를 전송하는 단계;

• 시스템 프로세싱 명령에 응답하여 제2 메시지를 수신하고 세션 키 생성기에 의해 생성된 세션 키를 이용하여 제1 임의 번호와 제2 임의 번호를 복호화하는 단계;

• 시스템 프로세싱 명령에 응답하여 제2 메시지의 복호화된 제1 임의 번호를 제1 임의 번호 생성기에 의해 생성된 제1 임의 번호와 비교하는 단계;

• 복호화된 제1 임의 번호가 생성된 제1 임의 번호와 매칭되는 경우 시스템 프로세싱 명령하에서 클라이언트를

인증하는 단계;

• 시스템 프로세싱 명령에 응답하여 세센키 생성기에 의해 생성된 세션 키로 제2 메시지에서 수신한 제2 임의 번호를 암호화하고 암호화된 제2 임의 번호를 전송하는 단계;

• 시스템 프로세싱 명령에 응답하여 암호화된 제2 임의 번호를 수신하고 제2 저장부로부터 수신한 세션 키로 복호화하는 단계;

• 복호화된 제2 임의 번호를 시스템 프로세싱 명령에 응답하여 제2 임의 번호 생성기에 의해 생성된 제2 임의 번호와 비교하는 단계; 및

• 복호화된 제2 임의 번호가 생성된 제2 임의 번호와 매칭되는 경우 상호 인증을 달성하도록 시스템 프로세싱 명령에 응답하여 서버를 인증하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 13**

제12항에 있어서,

제1 임의 번호를 생성하는 단계는 시스템 프로세싱 명령에 응답하여 제1 타이머 값을 생성하고 그 값을 제1 의사 임의 번호에 첨부하여 제1 임의 번호를 생성하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 14**

제12항에 있어서,

제2 임의 번호를 생성하는 단계는 시스템 프로세싱 명령에 응답하여 제2 타이머 값을 생성하고 그 값을 제2 의사 임의 번호(pseudo random number)에 첨부하여 제2 임의 번호를 생성하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 15**

제12항에 있어서,

비밀키를 생성하는 단계는 세션 시작에서 오직 진행 세션과정에만 유효한 고유키의 생성을 동반하는 것을 특징으로 하는 방법.

**청구항 16**

제12항에 있어서,

세션 키를 생성하는 단계는 세션 키 타이머 값의 만료에 기반하여 세션 키를 취소하고 만료에 따르는 새로운 세션 설립에 대한 요구를 나타내는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 17**

제12항에 있어서,

임의 번호들을 생성하는 단계는 시스템 프로세싱 명령에 응답하여 복제할수 없으며 서로 다른 세션마다에서 변경되는 번호들을 생성하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 18**

제12항에 있어서,

클라이언트는 서버가 클라이언트의 요청 실행 상태에서 응답하지 않는 방식으로 서버와 통신할수 있는 것을 특징으로 하는 방법.

**청구항 19**

제1항에 있어서,

상기 방법은 DTLS를 포함하는 전송 레이어 보안 스킴과 통합될수 있는 것을 특징으로 하는 방법.

**청구항 20**

제1항에 있어서,

상기 방법은 제한 장치들을 위한 CoAP를 포함하는 애플리케이션 레이어 프로토콜과 통합될수 있는 것을 특징으로 하는 방법.

## *발명의 설명*

### *기 술 분 야*

[0001] 본 발명은 사물인터넷을 위한 인증 및 보안에 관한 것이다.

[0002] <정의>

[0003] 본 명세서에서 사용된 표현《*IoT*》는 사물인터넷을 의미하며 여기서 고유하게 식별할수 있는 객체들은 인터넷과 같은 구조로 기술되었다.

[0004] 본 명세서에서 사용된 표현《*M2M*》은  무선 및 유선 시스템들이 이기종 노드를 포함하는 네트워크 상에서 호상 통신하도록 허용하는 사물지능통신기술을 의미한다.

[0005] 본 명세서에서 사용된 표현《난수(*nonce*)》는 오직 한번만 사용되는 임의 번호를 의미한다.

[0006] 본 명세서에서 사용된 표현 《데이터그램 전송》은 비연결형 전송 프로토콜을 의미하며, 이의 예시적이며 및 대중적인 구현은 사용자 데이터그램 프로토콜(UDP)이다.

[0007] 본 명세서에서 사용된 표현 《준비단계》는 통신하기 전에 클라이언트 측과 서버 측을 준비 및 구비시키는 공정을 의미한다. 그 공정은 예비-공유-비밀을 내장시키는 것과 같은 단계들을 포함한다.

[0008] 본 명세서에서 사용된 표현 《*세션 개시자*》는 서버에게 초기 메시지《HELLO》를 보내어 세션을 개시하는 장치를 나타낸다.

[0009] 이들 정의는 현존기술에서 표현되는 용어들에 첨부된다.

### *배 경 기 술*

[0010] IoT/M2M 은 물리적인 엔터티들을 포함하며; 이들의 식별 정보 또는 상태는 인터넷 인프라스트럭쳐 상에서 교환될수 있다. M2M 은 IoT 의 서브세트로 간주될수 있을 것이다. M2M 구동 IoT에 자료가 전송되는 패턴은 데이터 트래픽 모델과 참여 노드의 개수에 있어서 종래의 인터넷과 다르다. M2M은 종래의 사람 대 사람(H2H) 유형의 인터넷 상에서의 대화에 비해 훨씬 많은 노드를 취급한다.

[0011] IoT/M2M 시스템은 주로 무선 및/또는 유선 네트워크 상에서의 통신을 허용하는 센서와 같은 제한 장치들로 이루어진다. 일반적으로 이 무선 통신 네트워크는 또한 대역폭에 있어서도 제한된다. 이와 같은 제한 도메인에서 인증을 가지는 견고하면서도 낮은 오버헤드 보안형 통신 수단을 배치하는 것은 하나의 챌린지이다. 공개키 암호체계를 이용하는 종래의 견고한 인증 기반 구조들은 여기에 관계되는 프로세싱과 에너지, 대역폭의 요구 조건으로 인해 지내 비용이 많이 들수 있을 것이다. 또한 IP 레이어에서의 보안, 예를 들어, IPSec가 고려되는 경우, 리소스 사용 및 관리에 있어서 이는 부분 최적으로 된다. 또한 TLS와 같은 전송 레이어 보안 스킴(scheme)은 비록 매우 견고하기는 하지만 그의 리소스 요구 조건으로 인해 제한 장치들에 대해 비용이 지내 많이 들수 있기 때문에 적용할수 없다.

[0012] 제한 애플리케이션 프로토콜(CoAP)은 REST형 방식으로 인터넷상에서 제한 장치들사이의 대화형 통신을 허용하는 예시적인 네트워크 애플리케이션 레이어 프로토콜이다. 인터넷 엔진니어링 특별위원회(IETF)의  CoAP는 원래 경량 해결책을 창조하기 위해 사용자 데이터그램 프로토콜(UDP)상에서 동작하도록 설계되었으며 IoT/M2M을 위한 보안 레이어 해결책으로써 데이터그램 전송 레이어 보안(DTLS)을 제안한다. 그러나 완벽한 인증 기반의 공개키 인프라스트럭쳐(PKI)를 가지는 DTLS는 제한 장치들에 대하여 최적으로 되지 않는다. 따라서, DTLS의 예비-공유키(PSK) 방식은 제한된 장치들에 대한 경량 대안으로 정의된다. 이러한 스킴은 비록 경량이지만 견고성을 희생시킨다. 또한 엔드포인트의 인증이 부족하다.

[0013] CoAP에서, DTLS는 서비스 거부 (DoS) 공격을 완화시키기 위해 쿠키 교환 기술을 이용하며 여기서는 공격자는 증폭 공격을 시작하기 위해 클라이언트헬로우 (ClientHello) 메시지를 전송한다. 특히, PSK 방식에서, 클라이언트

는 예비-공유키로부터 예비-마스터 비밀과 마스터 비밀을 계산한 다음 서버가 요구되는 예비-공유키를 검색하기 위해 사용하는 psk_식별 정보를 포함한 클라이언트키교환 (ClientKeyExchange) 메시지를 서버에게 전송한다. 그러나 일반 본문 형식의 쿠키 교환은 견고하지 않다. 또한 쿠키 교환 메커니즘은 제한 환경에 대해 비용이 지내 많이 드는 연결 설립 오버헤드에 첨가되게 된다.

[0014] 따라서, IoT/M2M의 제한 공간에서 이용할수 있는 인증이 견고하고 보안이 경량인 시스템에서 명백히 공백이 존재한다는 것은 자명하다. 또한 일반 네트워킹/통신 시스템의 인증을 위한 요구조건을 포괄적으로 충족시키는 시스템 및 방법에 대한 필요성이 제기되게 된다.

## *발명의 내용*

### *해결하려는 과제*

[0015] 본 발명의 일 목적은 전형적인 제한된 IoT/M2M 환경에서 엔드포인트의 호상 인증을 위한 견고하면서도 경량인 시스템을 제공하는 것이다.

[0016] 본 발명의 다른 목적은 보다 핸드세이크 메시지를 적게 이용하는데 있어서 경량의 예비-공유 비밀방식의 보안스킴을 이용하는 시스템을 제공하는 것이다.

[0017] 본 발명의 또 다른 목적은 일반 네트워킹/통신 시스템의 인증을 위한 요구 조건을 포괄적으로 충족시키는 시스템을 제공하는 것이다.

[0018] 본 발명의 또 다른 목적은 대칭 암호화를 이용하여 클라이언트에 의해 개시된 페이로드-내장 인증 및 키 관리를 허용하는 시스템을 제공하는 것이다.

[0019] 본 발명의 시스템의 또 다른 목적은 DTLS와 같은 전송 레이어 보안 스킴과 통합될수 있는 시스템을 제공함으로써 현존의DTLS  스킴을 보강할뿐 아니라 교환 횟수를 줄여 이를 보다 경량화할수 있도록 하는 것이다.

[0020] 본 발명의 시스템의 또 다른 목적은 새로운 헤더 옵션을 결합하여 제한 장치들을 위한 CoAP와 같은 애플리케이션 레이어와 통합될수 있는 시스템을 제공하는 것이다.

[0021] 본 발명의 다른 목적들과 장점들은 첨부도면과 연결하여 읽을 때 하기의 설명으로부터 보다 자명해질 것이며, 이는 본 발명의 범위를 제한하고자 하는 것이 아니다.

### *과제의 해결 수단*

[0022] 본 발명은 서버와 클라이언트사이의 데이터그램 전송에서의 양방향 인증을 위한 컴퓨터 구현 시스템을 예상한다.

[0023] 전형적으로, 본 발명에 따라, 서버와 클라이언트사이의 데이터그램 전송에서의 양방향 인증을 위한 컴퓨터 구현 시스템은 시스템 프로세서를 포함하며 시스템 프로세서와 협력하여 제1 임의 번호를 생성하는 제1 임의 번호 생성기를 포함한다. 상기 시스템에 포함된 제2 임의 번호 생성기는 또한 시스템 프로세서와 협력하여 제2 임의 번호를 생성한다. 상기 시스템은 시스템 프로세서와 협력하는 비밀키 생성기를 포함하며 준비단계에서 양방향 인증 전에 시스템 프로세서의 생성 명령과 전송 명령하에서 비밀키를 서버와 클라이언트에게 생성 및 전송하도록 구성되었다. 상기 시스템에 존재하는 제1 저장부는 모든 클라이언트들의 클라이언트 ID를 저장하도록 구성되었다. 상기 시스템은 또한 시스템 프로세서의 전송 명령하에서 클라이언트로부터 클라이언트의 고유 ID를 포함한 제1 메시지를 서버에게 전송하도록 구성된 세션 개시자를 포함한다. 수신 장치는 시스템 프로세서와 협력하여 시스템 프로세서의 수신 명령하에서 제1 메시지를 수신하며 매칭 엔진이 구비되어 있어 수신한 클라이언트 ID를 제1 저장부에 저장된 클라이언트 ID와 비교하여 클라이언트를 식별한다. 세션 키(session key) 생성기는 고유하게 시간 제한된 그리고 제한적으로 유효한 세션 키를 생성하고 생성된 세션 키를 시스템 프로세서의 전송 명령하에 전송하도록 구성되어있다. 시스템은 또한 시스템 프로세서와 협력하는 챌린지 코드 생성기를 포함하며 세션 키를 수신하며 제1 임의 번호 생성기와 세션 키에 의해 생성된 제1 임의 번호를 포함하는 챌린지 코드를 시스템 프로세서의 생성 명령하에서 생성하며 시스템 프로세서의 전송 명령하에서 전송한다. 시스템에 존재하는 제1 암호기는 챌린지 코드 생성기와 협력하여 챌린지 코드 생성기로부터 생성된 챌린지 코드를 수신하며 시스템 프로세서로부터의 명령에 응답하여 수신한 생성된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀키를 이용하여 암호화하며 또한 시스템 프로세서의 전송 명령하에서 암호화된 챌린지 코드를 식별된 클라이언트에게 전송하도록 구성되었다. 제1 복호기는 시스템 프로세서와 협력하며 암호화된 챌린지코드를 수신하도록 구성되었으며

또한 시스템 프로세서로부터의 명령에 응답하여 암호화된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀키를 이용하여 복호화하여 복호화된 제1 임의 번호와 세션 키를 획득하도록 구성되었다. 시스템은 또한 제1 복호기로부터 세션 키를 수신하고 저장하도록 구성된 제2 저장부를 포함한다. 시스템에 존재하는 제2 암호기는 시스템 프로세서와 협력하며 복호화된 제1 임의 번호와 세션 키를 수신하도록 구성되었으며 또한 시스템 프로세서의 전송 명령 하에서 복호화된 제1 임의 번호와 제2 임의 번호 생성기에 의해 생성된 세션 키로 암호화된 제2 임의 번호를 포함한 제2 메시지를 전송하도록 구성되었다. 시스템은 또한 시스템 프로세서와 협력하는 제2 복호기를 포함하며 제2 메시지를 수신하고 시스템 프로세서로부터의 명령에 응답하여 세션 키 생성기에 의해 생성된 세션 키를 이용하여 제1 임의 번호와 제2 임의 번호를 복호화하도록 구성되었다. 상기 시스템에 존재하는 제1 비교기와 인증기는 시스템 프로세서로부터의 명령에 응답하여 제2 메시지로부터 복호화된 제1 임의 번호를 제1 임의 번호 생성기에 의해 생성된 제1 임의 번호와 비교하여 클라이언트를 인증하도록 구성되었다. 상기 시스템은, 시스템 프로세서로부터의 명령에 응답하여, 제2 메시지에서 수신한 제2 임의 번호를 세션 키 생성기에 의해 생성된 세션 키를 이용하여 암호화하고 시스템 프로세서의 전송 명령하에서 암호화된 제2 임의 번호를 전송하도록 구성된 제3 복호기를 포함한다. 제3 복호기는 암호화된 제2 임의 번호를 시스템 프로세서의 수신 명령하에서 수신하고 또한 시스템 프로세서로부터의 명령에 응답하여 제2 저장부로부터 수신한 세션 키를 이용하여 복호화하도록 구성되었다. 제2 비교기와 인증기는 시스템 프로세서로부터의 명령에 응답하여 복호화된 제2 임의 번호를 제2 임의 번호 생성기에 의해 생성된 제2 임의 번호와 비교하여 서버를 인증하고 상호 인증을 달성하도록 구성되었다.

[0024] 본 발명에 따라, 서버와 클라이언트사이의 데이터그램 전송에서의 양방향 인증을 위한 컴퓨터 구현 방법이 제공되며, 상기 방법은 시스템 프로세싱 명령과 하기의 단계를 포함한다:

[0025] • 제1 임의 번호 생성기의 도움으로 제1 임의 번호(난수_1)를 생성하는 단계;

[0026] • 제2 임의 번호 생성기의 도움으로 제2 임의 번호(난수_2)를 생성하는 단계;

[0027] • 비밀키 생성기의 도움으로 시스템 프로세싱 명령에 응답하여 비밀키를 생성하는 단계;

[0028] • 준비 단계과정에서 양방향 인증 전에 시스템 프로세싱 명령에 응답하여 생성된 비밀키를 서버와 클라이언트에게 전송하는 단계;

[0029] • 제1저장부에 모든 클라이언트들의 클라이언트 ID를 저장하는 단계;

[0030] • 시스템 프로세싱 명령에 응답하여 클라이언트의 고유 ID를 포함한 제1 메시지를 전송하는 단계;

[0031] • 시스템 프로세싱 명령에 응답하여 제1 메시지를 수신하고 수신한 클라이언트 ID를 제1 저장부의 저장된 클라이언트 ID와 매칭하는 단계;

[0032] • 수신한 클라이언트 ID에 기초하여 클라이언트를 식별하는 단계;

[0033] • 세션 키 생성기의 도움으로 고유하게 시간 제한된 그리고 제한적으로 유효한 세션 키를 생성하는 단계;

[0034] • 세션 키를 수신하고 시스템 프로세싱 명령하에서 제1 임의 번호 생성기와 세션 키에 의해 생성된 제1 임의 번호를 포함한 챌린지 코드를 생성하는 단계;

[0035] • 챌린지 코드 생성기로부터 생성된 챌린지 코드를 수신하고 시스템 프로세싱 명령에 응답하여 제1 암호기의 도움으로 수신한 생성된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀 키로 암호화하며 암호화된 챌린지 코드를 시스템 프로세싱 명령에 응답하여 전송하는 단계;

[0036] • 제1 임의 번호와 세션 키를 획득하기 위해 암호화된 챌린지 코드를 수신하고 시스템 프로세싱 명령에 응답하여 암호화된 챌린지 코드를 비밀키 생성기에 의해 생성된 비밀키를 이용하여 제1 복호기의 도움으로 복호화하는 단계;

[0037] • 제1 복호기로부터 세션 키를 수신하고 제2 저장부에 저장하는 단계;

[0038] • 시스템 프로세싱 명령에 응답하여 복호화된 제1 임의 번호와 세션 키를 수신하고 제1 임의 번호와 제2 임의

번호 생성기에 의해 생성된 제2 임의 번호를 포함한 세션 키를 이용하여 암호화된 제2 메시지를 전송하는 단계;

[0039] • 시스템 프로세싱 명령에 응답하여 제2 메시지를 수신하고 세션 키 생성기에 의해 생성된 세션 키를 이용하여 제1 임의 번호와 제2 임의 번호를 복호화하는 단계;

[0040] • 시스템 프로세싱 명령에 응답하여 제2 메시지의 복호화된 제1 임의 번호를 제1 임의 번호 생성기에 의해 생성된 제1 임의 번호와 비교하는 단계;

[0041] • 복호화된 제1 임의 번호가 생성된 제1 임의 번호와 매칭되는 경우 시스템 프로세싱 명령하에서 클라이언트를 인증하는 단계;

[0042] • 시스템 프로세싱 명령에 응답하여 세션 키 생성기에 의해 생성된 세션 키로 제2 메시지에서 수신한 제2 임의 번호를 암호화하고 암호화된 제2 임의 번호를 전송하는 단계;

[0043] • 시스템 프로세싱 명령에 응답하여 암호화된 제2 임의 번호를 수신하고 제2 저장부로부터 수신한 세션 키로 복호화하는 단계;

[0044] • 복호화된 제2 임의 번호를 시스템 프로세싱 명령에 응답하여 제2 임의 번호 생성기에 의해 생성된 제2 임의 번호와 비교하는 단계; 및

[0045] • 복호화된 제2 임의 번호가 생성된 제2 임의 번호와 매칭되는 경우 상호 인증을 달성하도록 시스템 프로세싱 명령에 응답하여 서버를 인증하는 단계.

## *발명의 효과*

[0046] 상기 설명된 본 발명에 따르는 데이터그램 전송에서의 경량 인증을 위한 콤퓨터 구현 시스템 및 방법은 하기의 내용의 실현을 포함한 그러나 이에 제한되지는 않는 여러 기술적우점을 가진다.

[0047] • 페이로드 내장된 대칭 키 기반의 인증과 세션 키 재생 타이머를 가지는 통합 키 관리로 하여 오버헤드를 감소시키는 시스템;

[0048] • 리소스 제한 센서 장치들을 보호하는데 바람직한 시스템;

[0049] • DTLS와 같은 전송 레이어 보안 스킴과 통합되어 현존의 DTLS 스킴을 보강할뿐 아니라 교환 횟수를 줄여 보다 경량화할수 있도록 하는 시스템;

[0050] • 페이로드 내장된 인증 스킴으로 애플리케이션 프로토콜을 개작할수 있는 시스템;

[0051] • 제한 장치들을 위한 CoAP와 같은 애플리케이션 레이어와 통합될수 있는 시스템;

[0052] • 인증이 달성된 이후 리소스 사용을 최적화하기 위해 애플리케이션 레이어 프로토콜을 이용한 새로운 헤더 옵션을 도입하여 CoAP의 개방-루프통신을 허용하는 시스템; 및

[0053] • 일반적인 네트워킹/통신 시스템의 인증 요구조건을 충족시키기 위한 포괄적인 시스템.

## *도면의 간단한 설명*

[0054] 본 발명의 시스템은 이제 첨부도면을 참조로 설명되며, 여기서:

도 1은 서버와 클라이언트사이에 상호 인증을 제공하는 시스템의 개략도를 도시한다.

도 2는 상호 인증과 안전한 통신을 달성하기 위한 시스템 흐름도를 도시한다.

도 3은 서버와 클라이언트사이의 핸드세이크과정에 포함된 단계들을 도시한다.

도 4는 예시적인 경우로써 현존하는 애플리케이션 레이어 프로토콜에 내장하기 위해 CoAP 메시지 포맷으로 도입된 헤더 옵션들을 도시한다.

도 5는 센서 장치(클라이언트)와 서버사이의 예시적인 인증 핸드세이킹을 도시한다.

도 6은 DTLS형의 보안 레이어를 이용한 인증을 위한 추가적인 레이어로써의 본 발명의 시스템 통합을 도시한다.

도 7은 예비-공유키 방식의 안전한 세션 개시를 위한 종래의 DTLS 핸드세이크의 타이밍-도표를 도시한다.

도 8은 본 발명에 따라 예비-공유 비밀을 가지는 수정된 DTLS 핸드세이크를 도시한다.

## *발명을 실시하기 위한 구체적인 내용*

[0055] 본 발명의 시스템은 이제 첨부도면에서 보여주는 실시례들을 참조로 설명될 것이다. 실시례들은 발명의 범위와 영역을 제한하지 않는다. 설명은 순수 본 발명의 실례들과 바람직한 실시례들 그리고 이의 제안된 애플리케이션에 관한 것이다.

[0056] 여기서 시스템과 이의 다양한 특징들과 상세한 우점들은 하기의 설명에서 비-제한 실시례들을 참조로 설명된다. 여기서 공지된 파라미터와 공정 기술들에 대한 설명은 실시례들을 불필요하게 불명료하게 하지 않도록 하기 위해 생략되었다. 여기서 이용된 실시례들은 오직 실시례들이 실행되는 방식에 대한 이해를 용이하게 하고 차후 당업자들이 실시례들을 실시할수 있게 하기 위한 것이다. 따라서, 실시례들은 구현범위를 제한하는 것으로 해석되지 말아야 한다.

[0057] 본 발명에 따라, 시스템은 예비-공유 비밀을 공유하는 두개의 엔드포인트사이의 챌린지-응답 형 교환 기반의 견고한 경량 인증스킴을 제공한다. 제안된 보안 해결책은 대칭 키 기반 보안 메커니즘이며 여기서 키 관리는 인증과 통합되어 있다. 시스템은 서버와 클라이언트사이의 데이터그램 전송에서의 양방향 인증을 제공하며 IoT/M2M에 적합하다.

[0058] 본 발명의 시스템은 낮은 오버헤드를 가지는 상호 인증을 제공한다. 상호 인증을 달성하기 위해, 시스템의 엔드포인트들에는 준비단계과정에서 예비-공유 비밀이 제공되며 클라이언트 데이터베이스가 클라이언트 식별을 위해 서버측에 제공된다. 시스템은 또한 의사 임의 번호(PRN) 모듈과 난수 생성을 위한 타이머(시스템 시간), 그리고 서버 키 생성 모듈을 포함한다. 난수와 키는 안전한 인증을 제공하는데 도움을 준다. 챌린지 메시지는 인증과정에서 서버와 클라이언트 측으로부터 생성된다. AES암호화와 복호화는 클라이언트와 서버측에서 사용된다.

[0059] 본 발명의 시스템은 또한 PSK 방식을 이용하는 DTLS와 같은 전송 레이어 보안 프로토콜에 맞게 개작될수 있다. 개작 단계는 DTLS의 상부에서의 암호화된 난수 기반의 챌린지 응답을 이용한 인증 세션의 설립단계와 사용을 위한 안전한 채널의 설립단계를 포함한다.

[0060] 첨부도면에 관하여, 도 1은 시스템 프로세서(102)의 명령에 기반한 서버(50)와 클라이언트(20)사이에 상호 인증을 제공하는 시스템(100)의 개략도를 도시한다. 본 발명의 시스템(100)은 예비-공유 비밀을 공유하는 두개의 엔드포인트사이의 챌린지-응답 형 교환 기반의 인증 스킴을 제안한다. 이 예비-공유 비밀은 시스템 프로세서(102)로부터의 생성 명령 기반의 비밀키 생성기(10)에 의해 생성된다. 본 발명의 보안 해결책은 대칭키 기반의 보안 메커니즘이며 여기서 키 관리는 인증과 통합되어 있다. 준비 단계과정에서 엔드포인트들은 예비-공유 비밀과 함께 구성되어 있다. 클라이언트(20) 측의 세션 개시자(22)는 클라이언트의 고유 식별 부호(ID)를 이용하여 서버(50)에게 HELLO 메시지를 보내는 것으로 세션을 개시한다. 서버(50) 측의 수신기(54)는 시스템 프로세서(102)로부터의 명령에 기반한 메시지를 수신하며 모든 클라이언트의 ID들을 저장하는 예비구성된 저장부(52)에서 ID를 먼저 검색한다. 그러나, 비법적인 클라이언트에 의한 스푸핑(spoofing)을 방지하기 위해 서버(50)는 챌린지 코드 생성기(60)의 도움으로 챌린지 코드를 생성한다. 이 챌린지 코드는 세션 키 생성기(58)에 의해 생성된 고유 세션 키《k》와 제1 임의 번호생성기(56)에 의해 생성된 임의 번호 《난수 1》을 포함한다. 세션 키 생성기(58)는 생성된 세션 키의 유효기간을 결정하기 위해 세션 키 타이머 값을 생성하는 세션 키 타이머(미도시)를 포함한다. 세션 키 《k》는 따라서 세션 키 타이머 값에 기반하여 취소되게 된다. 세션 키 타이머 값의 만료는 키의 취소를 나타내며 새로운 세션 키를 이용한 새로운 세션 설립의 필요성을 나타낸다. 제1 임의 번호 생성기(56)는 제1 타이머 값을 생성하는 제1 타이머(미도시)를 포함한다. 제1 임의 번호 생성기(56)에 의해 생성된 《난수1》은 바로 이 제1 타이머 값이 추가된 의사 임의 번호(PRN)이다. 다음 챌린지 코드는 비밀키 생성기(10)에 의해 생성되고 공유된 예비-공유 비밀을 이용하여 제1 암호기(62)에 의해 암호화된다. 이 챌린지 코드는 클라이언트에게 전송된다. 합법적인 클라이언트(20)는 비밀키 생성기(10)에 의해 공유된 비밀키의 도움으로 제1 복호기(24)를 통하여 챌린지코드를 복호화할수 있으며 또한 서버(50)에 의해 제공된 《난수1》과 세션 키《k》를 획득할수 있다. 다음 복호화된 세션 키 《k》는 제2 저장부(26)에 저장된다. 또한 챌린지 코드에 응답하여 클라이언트(20)는 서버(50)로부터 수신한 《난수1》과 클라이언트(20) 측에서 제2 임의 번호 생성기(30)에 의해 생성

된 《난수2》를 포함한 응답메시지를 작성한다. 제2 임의 번호 생성기(30)는 제2 타이머 값을 생성하는 제2 타이머(미도시)를 포함한다. 제2 타이머 값은 또 다른 의사 임의 번호(PRN)에 추가되어 《난수2》를 형성한다. 응답 메시지는 제2 암호기(28)에 의해 제1 복호기(24)를 이용하여 앞서 복호화된 세션 키 《k》로 암호화된다. 응답 메시지를 수신하면, 서버(50) 측에서의 제2 복호기(64)는 클라이언트(20)로부터의 응답을 복호화하고 제1 비교기와 인증기(66)를 이용하여 《난수1》을 제1 임의 번호 생성기(56)로부터 《난수1》의 자체 사본과 매칭한다. 두개가 매칭되면 서버(50)는 클라이언트(20)를 제1 비교기와 인증기(66)를 이용하여 인증하고 다음 시스템 프로세서(102)로부터의 명령에 기반하여 클라이언트(20)에게 제3 암호기의 도움으로 세션 키 《k》에 의해 암호화된 수신한 《난수2》를 포함한 메시지를 보낸다. 클라이언트는 제2 저장부(26)에 저장된 세션 키 《k》를 이용하여 제3 복호기(32)의 도움으로 《난수2》를 복호화한다. 복호화된 《난수2》는 다음 제2 비교기와 인증기(34)의 도움으로 제2 임의 번호 생성기(30)에 의해 생성된 《난수2》와 매칭된다. 복호화된 《난수2》가 클라이언트(20)의 《난수2》와 매칭되는 경우, 서버(50)는 제2 비교기와 인증기(66)에 의해 인증되고, 따라서 상호 인증을 달성한다.

[0061] 사용되는 난수와 키는 서로 다른 세션마다 변경되게 된다. 생성된 난수들은 타이머(카운터)로부터의 타이머 값이 추가된 의사 임의 번호(PRN)들을 포함하기 때문에 복제될수 없다. 이는 반복공격에 대한 저항을 제공한다.

[0062] 첨부도면에 관하여, 도 2는 상호 인증과 안전한 통신을 달성하기 위한 시스템 흐름도를 도시한다. 서버와 클라이언트는 준비 단계에서 하나의 예비-공유 비밀(y)로 구성된 두개의 엔드포인트들이다. 준비 단계의 완료이후, 클라이언트는 서버(200)에게 인증 요청을 보낸다. 따라서 세션은 클라이언트가 서버에게 자기의 고유 클라이언트 ID와 함께 《HELLO》 메시지를 보내는 것으로 개시된다. 메시지를 수신한 다음, 서버는 예비 구성된 데이터베이스에서 클라이언트의 ID를 검색한다. 그러나 비법 클라이언트에 의한 스푸핑을 방지하기 위해 서버는 고유키(k)와 임의 난수(난수1)를 포함한 챌린지 코드(202)를 생성한다. 챌린지 코드를 예비-공유 비밀(y)로 암호화하여 클라이언트에게 보낸다. 합법적인 클라이언트는 챌린지코드를 예비-공유 비밀(y)(204)을 이용하여 복호화할수 있으며 서버에 의해 제공된 난수와 키를 획득하게 된다. 이에 응답하여 클라이언트는 수신한 고유 키(k)로 암호화되고 서버(난수1)로부터 수신한 난수와 클라이언트 측에서 생성된 난수(난수2)를 포함한 응답 메시지를 작성한다. 서버는 클라이언트로부터의 응답을 복호화하고 난수1을 자체 사본(208)과 매칭한다. 두개의 난수가 매칭되지 않는 경우 클라이언트는 인증되지 않는다(212). 두개가 매칭된다면, 클라이언트 인증과 키 공유가 완료된다(210). 클라이언트를 인증한 다음, 서버는 k와 연결되고 y에 의해 암호화된 난수2로 클라이언트 챌린지(214)에 응답한다. 클라이언트 측에서, 서버로부터 수신한 난수2는 클라이언트의 난수2의 사본과 매칭되어 서버의 응답이 클라이언트 챌린지를 충족시키는가를 검열한다(216). 클라이언트가 자체 사본과 난수2를 매칭할수 있게 되면 그후 클라이언트는 서버를 인증한다(218). 만일 난수가 매칭되지 않으면, 서버는 인증되지 않는다(220). 일단 클라이언트와 서버가 호상 모두 인증되면 그들사이에 안전한 채널이 설립되게 된다(222).

[0063] 인증 공정에서 사용된 난수와 키는 서로 다른 세션들마다에서 변경되게 된다. 세션은 타이머를 이용하여 새롭게 된다. 본 발명의 시스템은 매 세션과정에서 고유 키와 고유 128비트 난수를 제공하기 위해 타이머(카운터)가 추가된 의사 임의 번호 생성기(PRNG)를 포함한다. 난수는 따라서 $R_j$(PRN)의 임의성과 $T_j$(타이머)의 단조로운 증가성으로 인해 복제할수 없다. $R_j$는 의사-임의 방식으로 생성되며 $T_j$과 함께 포함되어 반복공격이 일어날수 없도록 담보한다.

[0064] $$\left\{\Pr\left(R_j|_{t=T} = R_j|_{t=T'}\right) = 1\right\} < \epsilon', \epsilon' \to 0.$$

[0065] 공격의 충돌 가능성은 약 $2^{-56}$이다.

[0066] 난수의 예상할수 있는 비복제성은 16 비트 $T_j$에 의해 지배되고 예측불가능 부분은 $R_j$에 의해 지배된다.

[0067] 첨부도면에 관하여, 도 3은 서버와 클라이언트사이의 핸드세이크과정에 포함된 단계들을 도시한다. 도 3은 경량 상호 인증 및 키-관리 알고리즘을 도시하며 여기서 $D_i$는 클라이언트를 $\mathcal{G}$는 서버를 나타낸다. 인증 공정이 시작되기전에, 비밀 $y_i=\{0,1\}^{128}$은 준비단계에서 $D_i$와 $\mathcal{G}$사이에 오프라인으로 공유된다. 인증 공정은 클라이언트 $D_i$가 《HELLO, #$D_i$'》를 서버 $\mathcal{G}$에게 전송하는 세션 개시와 함께 시작된다(300). 일단 세션이 개시되면, 서버 $\mathcal{G}$는 챌린지 코드 《$AES\{y_i,(y_i \oplus \kappa_i \mid nonce_1)\}$ 난수 $AES\{y_i,(y_i \oplus \kappa_i \mid nonce_1)\}$》를 클라이언트에게 보내는 것

으로 응답하는데(302), 여기서 $\kappa_i$,$nonce_1$={0,1}$^{128}$ $\kappa_i$,$nonce_1$={0,1}$^{128}$, 메시지 크기는 256비트이다. 클라이언트는 챌린지 코드를 복호화하고 서버에게 *난수* $nonce_1$과 추가적인 *난수* $nonce_2$를 포함한 또다른 챌린지 코드를 전송하는 것으로 응답한다. 《AES{$\kappa_i$,($nonce_1 \oplus \gamma_i$ | $nonce_2$)} *난수* AES{$\kappa_i$,($nonce_1 \oplus \gamma_i$ | $nonce_2$)} *난수* AES{$\kappa_i$,($nonce_1 \oplus \gamma_i$ | $nonce_2$)}》(304)는 클라이언트 응답 및 챌린지이다. 서버 측에서, 서버는 *난수* $nonce_1$을 검증하고 클라이언트에게 《AES{$\gamma_i$,($nonce_2$ | $\kappa_i$)} *난수* AES{$\gamma_i$,($nonce_2$ | $\kappa_i$)}》의 형식으로 $\kappa_i$ 로 암호화된 *난수*$nonce_2$를 보내는 것으로 응답한다. 일단 클라이언트와 서버 측이 난수를 검증하고 인증이 완료되면, 클라이언트는 《AES{$\kappa_i$,($\rho_i$)}》의 형식으로 데이터 $\rho_i$를 서버에게 전송한다(308).

[0068] 첨부도면에 관하여, 도4와 도5는 각각 본 발명의 시스템을 현존하는 애플리케이션 레이어 프로토콜(400)에 내장한 후에 수정된 CoAP 메시지 포맷을 제안된 일반 스킴의 예시적인 애플리케이션으로써 도시하며 센서 장치(클라이언트)와 서버사이의 예시적인 인증 핸드세이킹을 도시한다.

[0069] CoAP의 전형적인 대화 모델은 HTTP의 클라이언트/서버 모델과 유사하며 REST형이다. 그러나 HTTP와는 달리, CoAP는 UDP와 같은 데이터그램 중심형 트랜스포트를 통한 교환을 비동기식으로 취급한다. 전형적으로 CoAP는 네 가지 형태의 메시지를 포함한다: 확인할수 있음, 확인할수 없음, 긍정 확인, 리셋. 이들 메시지는 방법 또는 응답 코드에 따라 요청 또는 응답을 전한다.

[0070] 본 발명에서 개시된 인증 스킴은 CoAP가 내장된 REST형 페이로드로 통합될수 있다. 도5에서 관찰할수 있는바와 같이, 센서 장치(클라이언트)와 서버사이의 상호 인증을 달성하기 위해 확인할수 있는(CON) 데이터 전송 방식을 이용한 POST 방법이 적용된다. 새로운 필드 《AUTH》는 안전(인증) 방식을 실행하기 위해 CoAP 헤더에 도입된다(400). 이 필드는 림계 옵션 클래스를 나타내는 미사용 옵션을 이용한다.《AUTH_MSG_TYPE》으로 지정된 또다른 옵션이 《AUTH》와 함께 도입되어 인증 세션을 설립하기 위한 다른 메시지를 나타낸다.

[0071] CoAP 헤더에서 옵션 필드들은 CoAP 메시지에서 선택적인 요청/응답 기능을 수행한다. 본 발명을 위해 지정된 필드는 하기와 같다:

[0072] • AUTH: 인증의 실행/인증방식 비실행을 나타낸다.

[0073] 이 필드에 대한 참값 또는 거짓값이 설정될수 있다.

[0074] • AUTH_MSG_TYPE: 이 필드는 《0》 또는 《1》으로 될수 있으며, 여기서,

[0075] 0= auth_init (인증개시)이며 1 =《챌린지에 대한 응답》이다.

[0076] • 《AUTH = true(참)》로 설정하여 실행된 인증 세션은 인증 단계과정에서 교환된 모든 관련 메시지들에 대한 헤더에서의 고정 《토큰》값을 이용하여 유지관리된다.

[0077] 도 4와 도5에 관하여, CoAP에 인증을 내장하기 위해 하기의 단계들이 실행된다:

[0078] • 개시에서, 센서-게이트웨이는 《auth_init》 즉 《0》의 AUTH_MSG_TYPE값을 가지는 것으로 하여 AUTH옵션 필드 값이 참인 CON방식의 POST 메시지와 페이로드의 《장치 식별 부호》를 서버 권한 부여URI에게 보낸다(600).

[0079] • 서버는《AUTH》옵션과 AUTH_MSG_TYPE의 《auth_init》값을 수신한 다음 페이로드로부터 장치 식별부호를 도출하고 이 장치 식별 부호와 관련된 예비-공유 비밀을 결정한다. 다음 난수(난수_1)과 키(K)를 생성한다. 서버는 공유 비밀을 이용하여 암호화된 페이로드를 생성한다.

[0080] • 서버는 새로운 리소스가 창조되었다는 것을 나타내는 응답 코드로 클라이언트에게 응답한다. 응답에서 URI는 인증을 위한 전체 핸드세이킹에 대한 림시 세션 ID를 나타낸다. 무효 장치 식별 부호인 경우, 서버는 《권한이 없음》이라는 응답 코드를 보낸다. 암호화된 페이로드는 피기백 처리(piggyback)되거나 또는 따로따로 클라이언트에게 전송된다(602).

[0081] • 클라이언트는 서버로부터 수신한 응답을 복호화하고 난수_1과 《K》를 획득한다. 난수(난수_2)를 생성한 다음 키《K》를 이용하여 암호화된 페이로드를 생성한다. 옵션 필드가 《AUTH》이고 AUTH_MSG_TYPE값이 《챌린지에 대한 응답》인, 그리고 마지막 POST 메시지에서와 같이 토큰 값이 동일한 POST 메시지를 이용하여 이 페이로드

를 전송한다.

[0082] • 키 재생(refreshment) 타이머는 세션을 재생하기 위하여 유지된다. (CoAP의 경우 이 값은 최대_재전송_카운트 (MAX_RETRANSMIT_COUNT) * 최대_재전송_ 타임아웃 (MAX_RETRANSMISSION_TIMEOUT) 값보다 커야 한다.)

[0083] • 서버는 《K》를 이용하여 헤더에서 상기 언급된 선택적인 값으로 상기 POST의 페이로드를 복호화하고 수신한 난수를 검열한다. 서버는 난수가 그의 이전 값(단계2에서 생성된)과 동일한 경우 응답 코드 《변경됨》으로 응답을 보내어 리소스에서의 변경이 인증되었다는 것을 나타내며, 그렇지 않은 경우에는 《허가되지 않음》을 보낸다(606).

[0084] 도5을 참조로 사용된 구문들은 하기와 같다:

[0085] $\Psi_n$:센서 게이트웨이 $\delta_n$와 $\delta$서버 사이의 공유 비밀

[0086] $\kappa_{n|\tau}$: $\tau^{th}$세션에서 센서 게이트웨이$\delta_n$와 서버사이에 교환된 키

[0087] $<\delta_n>$: 고유 센서 장치/$\delta_n$ 의 게이트웨이 ID

[0088] $AES(.)_\kappa$: 키 $\kappa$를 이용한 일반 본문형식의 AES 작동

[0089] $난수 nonce_{i=s,gw}$: 난수 $nonce_s$= 서버가 개시한 난수

[0090] 난수 $nonce_{gw}$=센서 게이트웨이/클라이언트가 개시한 난수

[0091] $\omega_n$:센서 게이트웨이 $\delta_n$의 센서 데이터

[0092] 인증 단계가 완료되고 안전한 채널이 설립되면, 클라이언트는 완전한 개방-루프(open-loop)방식으로 선택적으로 통신할수도 있고 한편 서버의 일부 리소스들을 업데이트하여 서버의 응답에 무관심을 표현할수도 있다.

[0093] 본 발명의 일 실시례에서, CoAP 는 NON(비신뢰) 방식으로 사용되며 옵션 필드(실례로 미-응답)가 도입되어 서버가 리소스 실행상태에 응답할 필요가 없다는 것을 나타낸다. 따라서 네트워크에서 부하가 감소되게 된다. 《미응답》필드 값은 《0》 또는 《1》이며, 여기서, 0은 서버가 상태에 대해 응답해야 한다는 것을 나타내며 1은 서버가 응답할 필요가 없다는 것을 나타낸다.

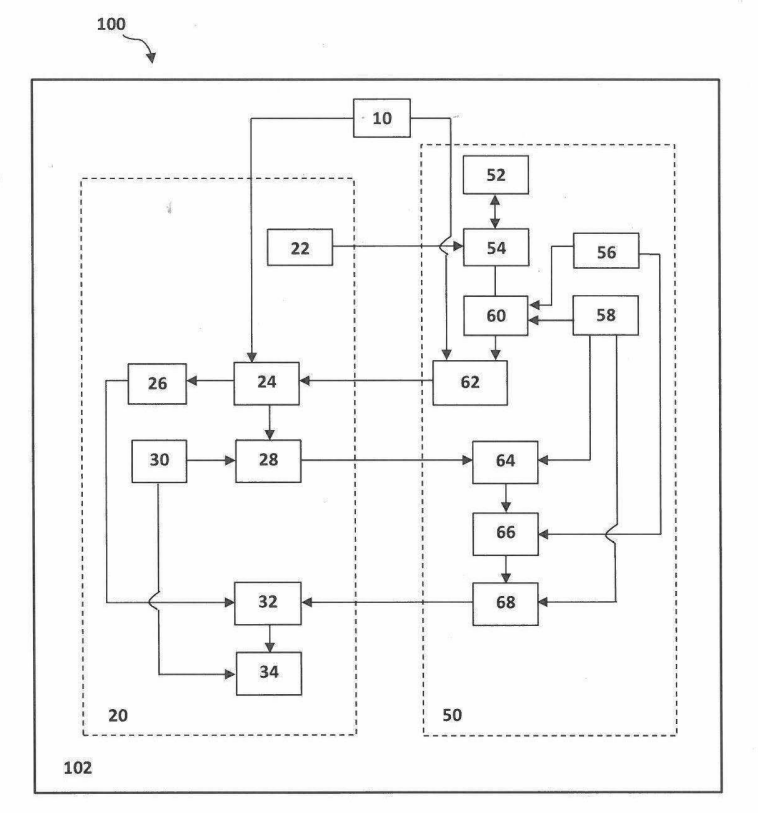
[0094] 첨부도면에 관하여, 도 6은 인증을 위한 추가적인 레이어로써의 본 발명의 시스템과 DTLS 형 보안 레이어(500)와의 통합을 도시한다.

[0095] 첨부도면에 관하여, 도 7은 예비-공유키 방식(PSK)를 가지는 안전한 세션 개시를 위한 DTLS 핸드세이크의 타이밍-도표를 도시하며 도 8은 안전한 연결을 설립하기 전에, 도출된 키와 함께 예비-공유 비밀을 가지는 변경된 DTLS 핸드세이크를 가져오는 본 발명의 방법과 DTLS 구조와의 통합을 도시한다. 매개의 핸드세이크 메시지는 예비-공유 비밀 또는 도출된 키 K에 의해 암호화된다. 도 7에서 '*'을 가지는 요소들은 상황 의존 메시지들을 나타낸다. 도 8은 메시지 교환사이의 맵핑과 도3에서 나타낸 서버와 클라이언트사이의 핸드세이크과정에 포함된 단계들을 도시한다. 도 7과 도 8을 참조로, 본 발명의 시스템은 핸드세이크의 횟수를 기존의 6개의 핸드세이크로부터 4개의 핸드세이크로 감소시킨다는 결론을 내릴수 있다.

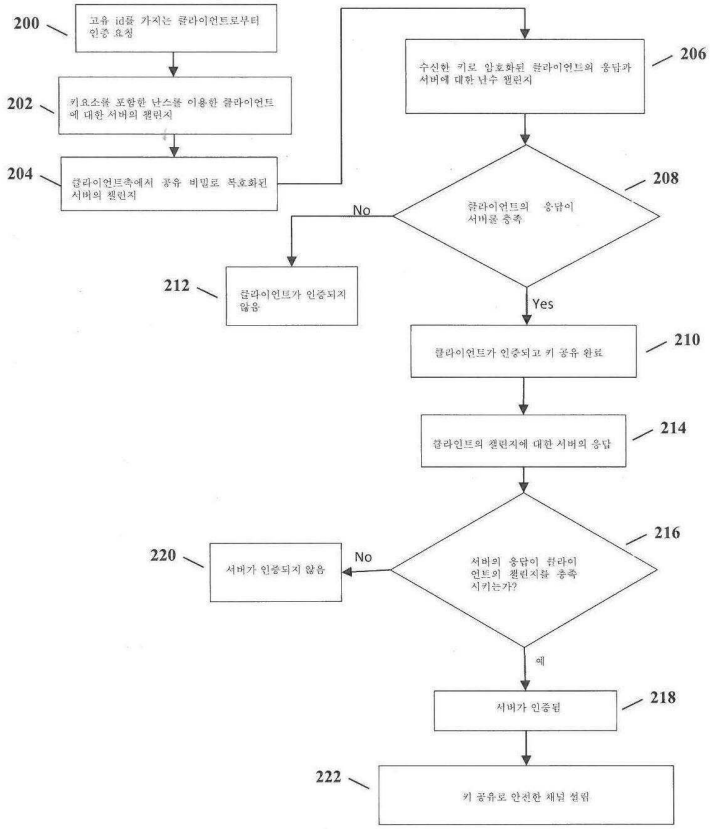
[0096] 특정 실시례들에 대한 상기 설명은 실시례의 일반 특성을 완전히 나타낼 것이며 따라서 당업자들은 현존지식을 적용하여 일반개념에서 이탈하지 않고 이러한 특정 실시례들을 다양한 애플리케이션을 위해 쉽게 변경 및/또는 개작할수 있으며, 따라서, 이러한 변경 및 개작은 개시된 실시례들의 등가물의 의미와 범위내에서 인식되어야 할 것이다. 본 명세서에서 채용된 술어 또는 용어들은 설명을 목적으로 하며 제한하고자 하는 것이 아니다. 따라서, 본 명세서의 실시례들은 보다 적합한 실시례의 견지에서 설명되었지만, 당업자들은 본 명세서의 실시례들이 여기서 설명된 실시례들의 정신과 범위내에서 수정하여 실행될수 있다는 것을 인식할 것이다.

도면

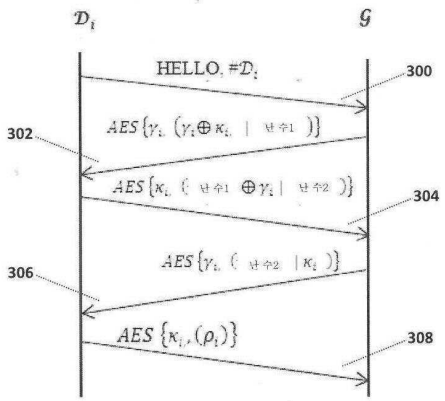
도면1



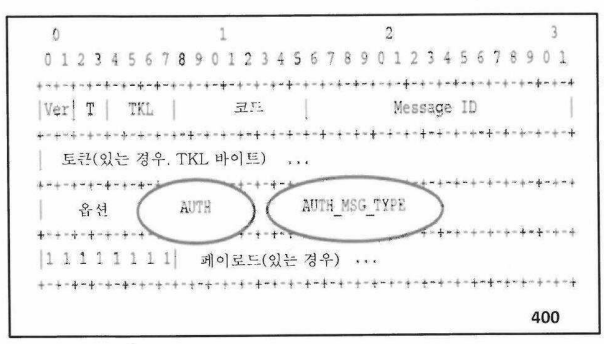
도면2


200
고유 id를 가지는 클라이언트로부터 인증 요청
202
키요소와 포함한 난스를 이용한 클라이언트에 대한 서버의 챌린지
204
클라이언트측에서 공유 비밀로 복호화된 서버의 챌린지
206
수신한 키로 암호화된 클라이언트의 응답과 서버에 대한 난수 챌린지
208
클라이언트의 응답이 서버를 충족
No
Yes
212
클라이언트가 인증되지 않음
210
클라이언트가 인증되고 키 공유 완료
214
클라이언트의 챌린지에 대한 서버의 응답
216
서버의 응답이 클라이언트의 챌린지를 충족시키는가?
220
서버가 인증되지 않음
No
예
218
서버가 인증됨
222
키 공유로 안전한 채널 설립

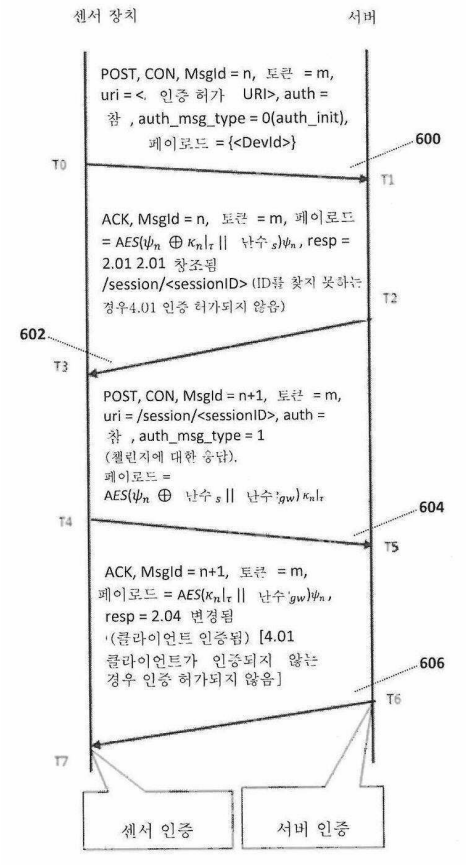
도면3


$\mathcal{D}_i$
$\mathcal{G}$
HELLO, $\#\mathcal{D}_i$
300
$AES\{\gamma_i, (\gamma_i \oplus \kappa_i \mid$ 난수1 $)\}$
302
$AES\{\kappa_i, ($ 난수1 $\oplus \gamma_i \mid$ 난수2 $)\}$
304
$AES\{\gamma_i, ($ 난수2 $\mid \kappa_i)\}$
306
$AES\{\kappa_i, (\rho_i)\}$
308

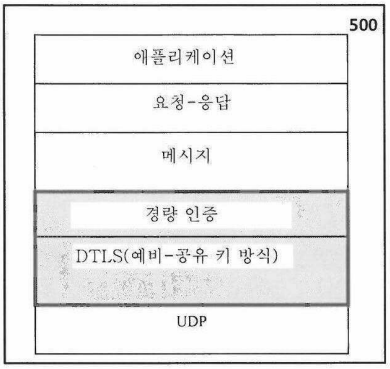
도면4


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Ver
T
TKL
코드
Message ID
토큰(있는 경우. TKL 바이트) ...
옵션
AUTH
AUTH_MSG_TYPE
1 1 1 1 1 1 1 1
페이로드(있는 경우) ...
400

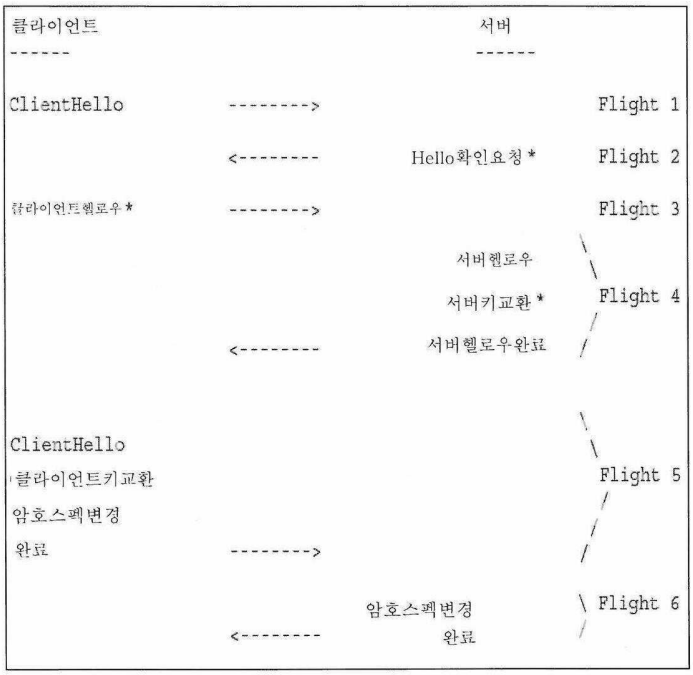
도면5


센서 장치
서버
POST, CON, MsgId = n, 토큰 = m, uri = <. 인증 허가 URI>, auth = 참 , auth_msg_type = 0(auth_init), 페이로드 = {<DevId>}
600
T0
T1
ACK, MsgId = n, 토큰 = m, 페이로드 = AES(ψn ⊕ κn|τ || 난수 s)ψn , resp = 2.01 2.01 창조됨 /session/<sessionID> (ID를 찾지 못하는 경우4.01 인증 허가되지 않음)
T2
602
T3
POST, CON, MsgId = n+1, 토큰 = m, uri = /session/<sessionID>, auth = 참 , auth_msg_type = 1 (챌린지에 대한 응답). 페이로드 = AES(ψn ⊕ 난수 s || 난수 'gw) κn|τ
604
T4
T5
ACK, MsgId = n+1, 토큰 = m, 페이로드 = AES(κn|τ || 난수 'gw)ψn , resp = 2.04 변경됨 (클라이언트 인증됨) [4.01 클라이언트가 인증되지 않는 경우 인증 허가되지 않음]
606
T6
T7
센서 인증
서버 인증

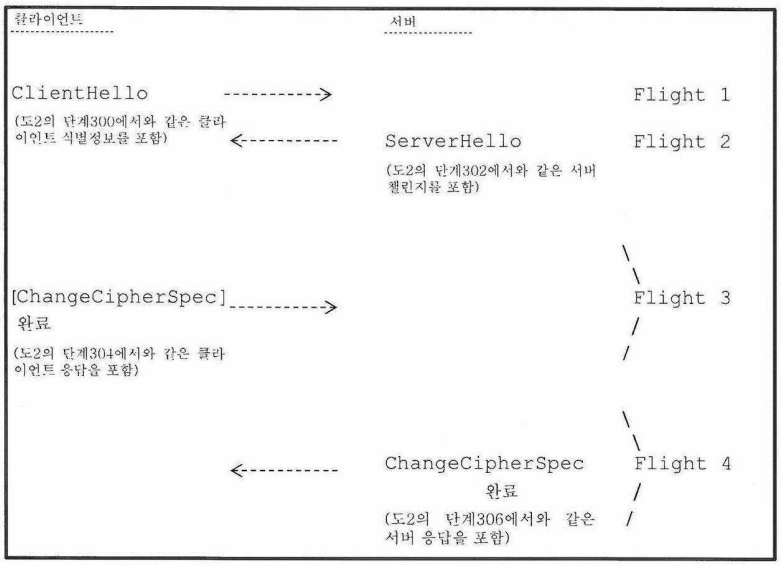
도면6


500
애플리케이션
요청-응답
메시지
경량 인증
DTLS(예비-공유 키 방식)
UDP

도면7


클라이언트
서버
ClientHello
Flight 1
Hello확인요청*
Flight 2
클라이언트헬로우*
Flight 3
서버헬로우
서버키교환*
Flight 4
서버헬로우완료
ClientHello
클라이언트키교환
암호스펙변경
완료
Flight 5
암호스펙변경
완료
Flight 6

도면8


클라이언트
서버
ClientHello
(도2의 단계300에서와 같은 클라이언트 식별정보를 포함)
ServerHello
(도2의 단계302에서와 같은 서버 챌린지를 포함)
Flight 1
Flight 2
[ChangeCipherSpec]
완료
(도2의 단계304에서와 같은 클라이언트 응답을 포함)
Flight 3
ChangeCipherSpec
완료
(도2의 단계306에서와 같은 서버 응답을 포함)
Flight 4