(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0076773
(43) 공개일자 2017년07월04일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 9/08* (2006.01)
H04L 9/14 (2006.01) *H04L 9/30* (2006.01)
H04L 9/32 (2006.01) *H04W 12/04* (2009.01)
H04W 12/06 (2009.01) *H04W 4/00* (2009.01)
(52) CPC특허분류
H04L 63/0884 (2013.01)
H04L 63/0281 (2013.01)
(21) 출원번호 10-2017-7014769
(22) 출원일자(국제) 2015년10월30일
심사청구일자 2017년05월30일
(85) 번역문제출일자 2017년05월30일
(86) 국제출원번호 PCT/US2015/058368
(87) 국제공개번호 WO 2016/114842
국제공개일자 2016년07월21일
(30) 우선권주장
62/073,578 2014년10월31일 미국(US)

(71) 출원인
콘비다 와이어리스, 엘엘씨
미국 19809-3727 델라웨어주 윌밍턴 스위트 300 벨레뷰 파크웨이 200
(72) 발명자
초이, 비노드, 쿠마르
미국 19428 펜실베이니아주 콘쇼호켄 유닛 1233 웨스트 엘름 스트리트 200
시드, 데일, 엔.
미국 18104 펜실베이니아주 앨런타운 노스 36번 스트리트 229
(뒷면에 계속)
(74) 대리인
양영준, 김연송, 백만기

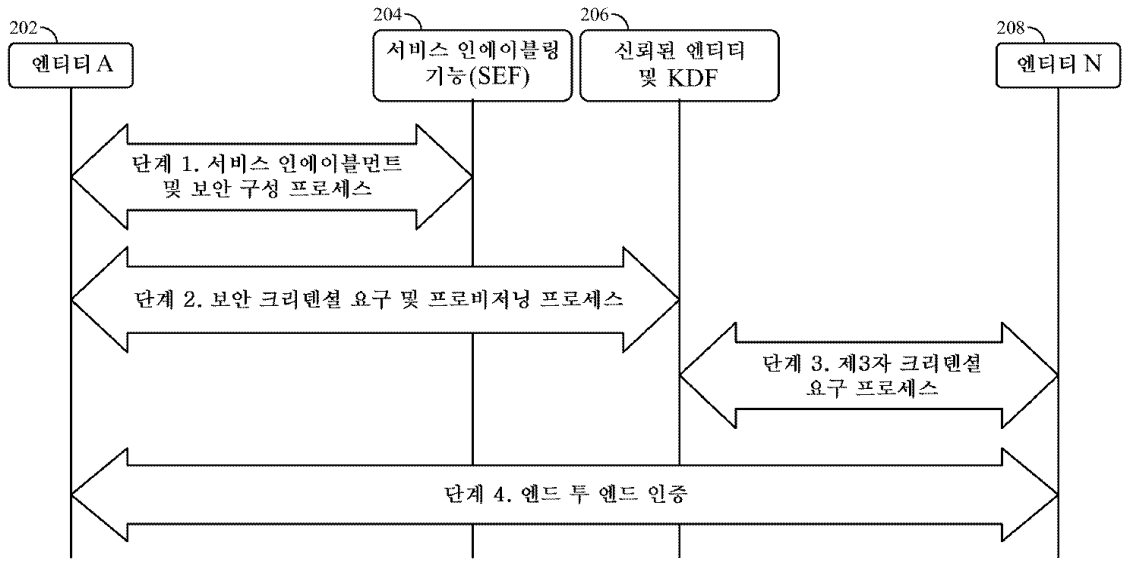
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **엔드 투 엔드 서비스 계층 인증**

(57) 요 약

다양한 능력들(예를 들어, 프로세싱, 메모리 등)을 갖고 이전의 보안 연관들 없이 엔티티들 사이에서 엔드 투 엔드 인증을 수행하기 위한 다양한 메커니즘들이 사용된다. 보안 프로비저닝 및 구성 프로세스는 적절한 보안 크리덴셜들, 기능들, 스코프 및 파라미터들이 엔티티에 프로비저닝될 수 있도록 행해진다. 다른 엔티티들에 보안 크리덴셜들을 배포하는 메커니즘들이 개발되며, 이 엔티티들은 서비스 계층 또는 세션 계층에서 직접 또는 위임 모드들을 사용하여 엔드 투 엔드 인증을 수행하기 위해 크리덴셜들을 사용할 수 있다.

대 표 도



(52) CPC특허분류
H04L 63/061 (2013.01)
H04L 9/0861 (2013.01)
H04L 9/14 (2013.01)
H04L 9/30 (2013.01)
H04L 9/3239 (2013.01)
H04L 9/3242 (2013.01)
H04W 12/04 (2013.01)
H04W 12/06 (2013.01)
H04W 4/005 (2013.01)
(72) 발명자
믈라딘, 카탈리나, 엠.
미국 19040 펜실베이니아주 해트버러 롤러 로드 137
왕, 충강
미국 08540 뉴저지주 프린스턴 칼라일 코트 9

명 세 서

청구범위

청구항 1

네트워크에 접속된 엔티티에 의한 사용을 위한 방법으로서, 상기 엔티티는 프로세서 및 메모리를 포함하고, 상기 엔티티는, 상기 엔티티에 의해 실행될 때, 상기 방법을 수행하는 상기 메모리에 저장된 컴퓨터 실행가능 명령어들을 추가로 포함하고, 상기 방법은,

제2 엔티티에 대한 접속을 위해 신뢰되는 위치로부터 보안 크리덴셜들을 획득하는 단계 - 상기 제2 엔티티는 상기 엔티티로부터 복수의 서비스 계층 홉들만큼 떨어져 있음 -; 및

상기 보안 크리덴셜들을 사용하여 상기 제1 엔티티와 상기 제2 엔티티 사이에 접속을 형성하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서, 상기 엔티티는 상기 제2 엔티티에 대한 통신을 개시하는 방법.

청구항 3

제2항에 있어서, 발신 엔티티가 상기 제2 엔티티와의 통신을 개시하고, 상기 엔티티는 상기 발신 엔티티에 대한 위임된 프록시(delegated proxy)로서 동작하는 방법.

청구항 4

제3항에 있어서, 위임된 프록시로서의 상기 엔티티는 상기 발신 엔티티에 대한 엔드 투 엔드 메시지 인증 데이터 및 상기 위임된 프록시의 엔드 투 엔드 메시지 인증 데이터와 함께 결합된 메시지를 상기 제2 엔티티에 전송하는 방법.

청구항 5

제4항에 있어서, 상기 결합된 메시지는 상기 발신 엔티티로부터의 암호화된 메시지 및 상기 위임된 프록시로부터의 암호화된 메시지를 포함하는 방법.

청구항 6

제1항에 있어서, 상기 보안 크리덴셜들은 엔드 투 엔드 메시지 인증을 생성하는 데 사용되며, 상기 엔드 투 엔드 메시지 인증은 상기 엔티티와 상기 제2 엔티티 사이에서 전송되는 방법.

청구항 7

제1항에 있어서, 상기 접속은 상기 복수의 홉들을 피하는 상기 엔티티와 상기 제2 엔티티 사이의 직접 접속인 방법.

청구항 8

제1항에 있어서, 발신 엔티티가 서비스 인에이블링 기능과 통신하고, 상기 발신 엔티티는 상기 엔티티이거나 또는 상기 엔티티를 위임된 프록시로서 사용하는 다른 엔티티인 방법.

청구항 9

프로세서, 메모리, 및 통신 회로를 포함하는 엔티티로서, 상기 엔티티는 자신의 통신 회로를 통해 통신 네트워크에 접속되고, 상기 엔티티는 상기 엔티티의 상기 메모리에 저장된 컴퓨터 실행가능 명령어들을 추가로 포함하고, 상기 명령어들은, 상기 엔티티의 상기 프로세서에 의해 실행될 때, 상기 엔티티로 하여금,

제2 엔티티에 대한 접속을 위해 신뢰되는 위치로부터 보안 크리덴셜들을 획득하게 하고 - 상기 제2 엔티티는 상

기 엔티티로부터 복수의 서비스 계층 홉들만큼 떨어져 있음 -;

상기 보안 크리덴셜들을 사용하여 상기 제1 엔티티와 상기 제2 엔티티 사이에 접속을 형성하게 하는 엔티티.

청구항 10

네트워크에 접속된 신뢰된 엔티티에 의한 사용을 위한 방법으로서, 상기 신뢰된 엔티티는 프로세서 및 메모리를 포함하고, 상기 신뢰된 엔티티는, 상기 신뢰된 엔티티에 의해 실행될 때, 상기 방법을 수행하는 상기 메모리에 저장된 컴퓨터 실행가능 명령어들을 추가로 포함하고, 상기 방법은,

제1 엔티티에 제1 보안 크리덴셜들을 전송하는 단계; 및

제2 엔티티에 제2 보안 크리덴셜들을 전송하는 단계

를 포함하고,

상기 제2 엔티티는 상기 제1 엔티티로부터 복수의 서비스 계층 홉들만큼 떨어져 있으며, 상기 제1 보안 크리덴셜들 및 상기 제2 보안 크리덴셜들은 상기 제1 엔티티와 상기 제2 엔티티 사이에 접속을 형성하는 데 사용되는 방법.

청구항 11

제10항에 있어서, 상기 제1 엔티티는 상기 제2 엔티티에 대한 통신을 개시하는 방법.

청구항 12

제10항에 있어서, 발신 엔티티가 상기 제2 엔티티와의 통신을 개시하고, 상기 제1 엔티티는 상기 발신 엔티티에 대한 위임된 프록시로서 동작하는 방법.

청구항 13

제12항에 있어서, 위임된 프록시로서의 상기 제1 엔티티는 상기 발신 엔티티에 대한 엔드 투 엔드 메시지 인증 데이터 및 상기 위임된 프록시의 엔드 투 엔드 메시지 인증 데이터와 함께 결합된 메시지를 상기 제2 엔티티에 전송하는 방법.

청구항 14

제10항에 있어서, 상기 제1 보안 크리덴셜들은 엔드 투 엔드 메시지 인증을 생성하는 데 사용되며, 상기 엔드 투 엔드 메시지 인증은 상기 엔티티와 상기 제2 엔티티 사이에서 전송되는 방법.

청구항 15

제10항에 있어서, 상기 접속은 상기 복수의 홉들을 피하는 상기 엔티티와 상기 제2 엔티티 사이의 직접 접속인 방법.

청구항 16

제10항에 있어서, 발신 엔티티가 서비스 인에이블링 기능과 통신하고, 상기 발신 엔티티는 상기 제1 엔티티이거나 또는 상기 제1 엔티티를 위임된 프록시로서 사용하는 다른 엔티티인 방법.

청구항 17

네트워크에 접속된 엔티티에 의한 사용을 위한 방법으로서, 상기 엔티티는 프로세서 및 메모리를 포함하고, 상기 엔티티는, 상기 엔티티에 의해 실행될 때, 상기 방법을 수행하는 상기 메모리에 저장된 컴퓨터 실행가능 명령어들을 추가로 포함하고, 상기 방법은,

보안 프로파일, 디바이스 프로파일 및 엔티티 프로파일 중 적어도 하나의 프로파일을 수신하는 단계; 및

상기 보안 프로파일, 상기 디바이스 프로파일 및 상기 엔티티 프로파일 중 상기 적어도 하나의 프로파일을 사용하여, 멀티-홉(multi-hop) 서비스 계층 접속에 대한 보안 요건들을 선택하는 단계

를 포함하는 방법.

청구항 18

제17항에 있어서, 상기 보안 프로파일, 상기 디바이스 프로파일 및 상기 엔티티 프로파일은 상기 멀티-홉 서비스 계층 접속에 대한 보안 요건들을 선택하는 데 사용되는 방법.

청구항 19

제17항에 있어서, 상기 보안 프로파일, 상기 디바이스 프로파일 및 상기 엔티티 프로파일 중 상기 적어도 하나의 프로파일은 상기 보안 요건들을 충족시키는 보안 프로토콜들, 알고리즘들 및 크리덴셜들을 유도하는 데 사용되는 방법.

청구항 20

제17항에 있어서, 상기 제1 엔티티와 상기 제2 엔티티 사이의 상기 멀티-홉 서비스 계층 접속에 대한 엔드 투 엔드 메시지 인증 및 메시지 기밀성(message confidentiality)을 위한 보안 크리덴셜들은 상기 제1 엔티티와 신뢰된 제3 엔티티 사이의 보안 연관을 사용하는 부트스트래핑 프로세스에 의해 수행되는 방법.

발명의 설명

기 술 분 야

[0001] <관련 출원들에 대한 상호 참조>

[0002] 본 출원은 2014년 10월 31일자로 출원된 미국 가출원 제62/073,578호의 이익을 주장하며, 그 내용은 본 명세서에 참조로 포함된다.

배 경 기 술

[0003] M2M(machine-to-machine) 기술들은 유선 및 무선 통신 시스템들을 사용하여 디바이스들이 서로와 보다 직접적으로 통신할 수 있게 한다. M2M 기술들은 인터넷과 같은 네트워크를 통해 통신하는 IoT(Internet of Things), 고유하게 식별가능한 객체들의 시스템, 및 이러한 객체들의 가상 표현들의 추가적인 실현을 가능하게 한다. IoT는 식료품점에 있는 제품들과 같이 심지어 평범한 일상의 객체들과의 통신을 용이하게 할 수 있으며, 따라서 그러한 객체들에 대한 지식을 향상시킴으로써 비용과 낭비를 줄일 수 있다. 예를 들어, 매장들은, 재고로 있을 수 있거나 판매되었을 수 있는 객체들과 통신하거나 이들로부터 데이터를 획득할 수 있음으로써, 매우 정확한 재고 데이터를 유지 관리할 수 있다. 이해되는 바와 같이, IoT는 수 많은 디바이스들을 포함할 수 있는 잠재력을 가지고 있다.

[0004] 이하, 도 1a는 예시적인 oneM2M 기능 아키텍처(100)를 도시하는 도면이다. 도 1a 내지 도 1b에 도시된 바와 같이, 개발 중인 oneM2M 표준은 "공통 서비스 엔티티(Common Service Entity)(CSE)"로 불리는 서비스 계층을 정의한다. 서비스 계층의 목적은 e-헬스, 차량 관리(fleet management), 스마트 홈들과 같이, 상이한 '수직적' M2M 사일로 시스템들 및 애플리케이션들에 의해 사용될 수 있는 "수평적" 서비스들을 제공하는 것이다. CSE는 4개의 레퍼런스 포인트들을 지원한다. Mca 레퍼런스 포인트는 애플리케이션 엔티티(Application Entity)(AE)와 인터페이싱한다. Mcc 레퍼런스 포인트는 동일한 서비스 제공자 도메인 내의 다른 CSE와 인터페이싱하며, Mcc 레퍼런스 포인트는 상이한 서비스 제공자 도메인의 다른 CSE와 인터페이싱한다. Mcn 레퍼런스 포인트는 기본 네트워크 서비스 엔티티(NSE)와 인터페이싱한다. NSE는 디바이스 관리, 위치 서비스들 및 디바이스 트리거링과 같은 기본 네트워크 서비스들을 CSE들에 제공한다. CSE는 "발견(Discovery)", "데이터 관리 및 레포지토리(Data Management & Repository)"와 같은 "공통 서비스 기능(Common Service Function)(CSF)들"이라고 하는 복수의 로지컬 기능들을 포함한다.

[0005] 도 1b는 oneM2M 아키텍처를 위해 개발 중인 CSF들을 도시하는 도면이다.

[0006] oneM2M은 애플리케이션 서비스 노드(Application Service Node)(ASN)들, 애플리케이션 전용 노드(Application Dedicated Node)(ADN)들, 미들 노드(Middle Node)(MN)들 및 인프라스트럭처 노드(Infrastructure Node)(IN)들과 같은 유형들의 노드들을 인에이블한다.

[0007] 애플리케이션 서비스 노드(ASN)는 하나의 CSE를 포함하고 적어도 하나의 AE를 포함하는 노드이다. 물리적 매핑의 예는 M2M 디바이스에 상주하는 ASN이다.

[0008] 애플리케이션 전용 노드(ADN)는 적어도 하나의 AE를 포함하고 CSE를 포함하지 않는 노드이다. 물리적 매핑의 예는 제한된 M2M 디바이스에 상주하는 ADN이다.

[0009] 미들 노드(MN)는 하나의 CSE를 포함하고 0개 이상의 AE들을 포함하는 노드이다. 물리적 매핑의 예는 M2M 게이트웨이에 상주하는 MN이다.

[0010] 인프라스트럭쳐 노드(IN)는 하나의 CSE를 포함하고 0개 이상의 AE들을 포함하는 노드이다. 물리적 매핑의 예는 M2M 서비스 인프라스트럭쳐에 상주하는 IN이다.

[0011] 현재, oneM2M 엔드 노드들이 보안된 방식으로 서로 통신하기를 원할 때, 노드들과 중간 노드들은 홉 바이 홉(hop-by-hop) 방식으로 서로 보안 연관을 확립한다. 홉 바이 홉 보안 연관들은 인증서들을 사용하여 대칭 키들에 의해, 또는 직접 프로세스 또는 인프라스트럭쳐에 의해 수행될 수 있는 부트스트래핑 프로세스에 의해 확립될 수 있다. 또한, TS-0003-보안 솔루션들 문서는 다음과 같이 기술하고 있다 : "서비스 계층 레벨에서, 보안 연관 구축은 인접한 AE/CSE 사이에서 교환되는 메시지들을 보호하는 TLS 또는 DTLS 세션, 즉 홉 바이 홉을 초래한다. 신뢰되지 않은 중간 노드들로부터 정보 교환의 프라이버시를 보호해야 할 필요가 있는 AE들은 이들 간의 직접적인 보안 연관을 지원하기 위해 프로비저닝될 수 있다.

발명의 내용

[0012] 다양한 능력들(예를 들어, 프로세싱, 메모리 등)을 갖고 이전의 보안 연관들이 없는 엔티티들 사이에서 엔드 투 엔드 인증을 수행하기 위한 다양한 메커니즘들이 사용된다. 보안 프로비저닝 및 구성 프로세스는 적절한 보안 크리덴셜들, 기능들, 스코프(scope) 및 파라미터들이 엔티티에 프로비저닝될 수 있도록 행해진다. 그리고, 다른 엔티티들에 보안 크리덴셜들을 배포하는 메커니즘들이 사용되며, 이 메커니즘들은 직접 또는 위임 모드들을 사용하여 서비스 계층 또는 세션 계층에서 엔드 투 엔드 인증을 수행하기 위해 보안 크리덴셜들을 사용할 수 있다.

[0013] 이 개요는 이하의 상세한 설명에서 더 설명되는 단순화된 형태의 개념들의 선택을 도입하기 위해 제공된다. 이 요약은 청구되는 대상의 주요 특징들 또는 필수 특징들을 식별하도록 의도되지 않으며, 청구되는 대상의 범위를 제한하는 데 사용되는 것으로 의도되지 않는다. 또한, 청구되는 대상은 본 개시내용의 임의의 부분에서 언급된 임의의 또는 모든 단점들을 해결하는 제한들에 한정되지 않는다.

도면의 간단한 설명

[0014] 첨부 도면들과 함께 예로서 주어지는 다음의 설명으로부터 더 상세한 이해가 이루어질 수 있다.

도 1a 및 도 1b는 oneM2M 서비스 계층의 도면들이다.

도 2는 엔드 투 엔드(End-to-End)(E2E) 보안 페이즈들을 도시하는 도면이다.

도 3a 및 도 3b는 엔티티 A와 엔티티 B 사이의 예시적인 E2E 동작들을 도시하는 도면이다.

도 4a 및 도 4b는 oneM2M 실시예를 도시하는 도면들이다.

도 5a 및 도 5b는 보안 크리덴셜 요구 및 프로비저닝(Security Credential Requisition and Provisioning)(SCRP) 페이즈를 도시하는 도면이다.

도 6a 및 도 6b는 제3자 크리덴셜 요청 페이즈를 도시하는 도면이다.

도 7a 및 도 7b는 AE1이 CSE3 상에서 호스팅되는 원격 리소스에 Update 동작을 요청하는 E2E 인증을 도시한다.

도 8a 및 도 8b는 위임 모드(delegate mode) 접근법을 사용하는 서비스 계층에서의 E2E 인증을 도시하는 도면이다.

도 9는 위임 모드를 사용하여 세션 계층(DTLS/TLS)에서 수행되는 E2E 인증을 도시하는 도면이다.

도 10은 직접 모드를 사용하는 세션 계층에서의 E2E 인증을 도시하는 도면이다.

도 11a 및 도 11b는 그룹 인증을 도시하는 도면이다.

도 12는 일 실시예의 인터페이스를 도시하는 도면이다.

도 13은 엔드 투 엔드 메시지 인증을 위한 MAC의 생성을 도시하는 도면이다.

도 14는 일 실시예의 부트스트래핑 프로세스를 도시하는 도면이다.

도 15a 및 도 15b는 AE에서의 리소스 표현 연관, 및 어트리뷰트들, 즉 홉 바이 홉 보안 크리덴셜뿐만 아니라 엔드 투 엔드 크리덴셜을 갖는 <securityParameters> 리소스 구조를 도시하는 도면들이다.

도 16a 내지 도 16c는 엔티티 프로파일, 디바이스 프로파일 및 보안 프로파일들의 리소스 표현들을 도시하는 도면들이다.

도 17은 대칭 키에 의한 엔드 투 엔드 메시지 인증 및 무결성 체크를 도시하는 도면이다.

도 18은 복수의 서비스-계층 홉들만큼 서로 떨어져 있는 두 개의 엔티티들(AE2 및 CSE1) 사이의 대칭 키 메커니즘에 의한 엔드 투 엔드 메시지 인증 및 무결성 체크, 및 또한 메시지 기밀성을 모두 도시하는 도면이다.

도 19는 신뢰되거나 덜 신뢰될 수 있거나 심지어는 신뢰할 수 없는 중간 홉들을 횡단하여, 복수의 서비스-계층 홉들만큼 서로 떨어져 있는 두 개의 엔티티들(AE2 및 CSE1) 사이의 대칭 키 메커니즘에 의한 엔드 투 엔드 메시지 인증 및 무결성 체크, 및 또한 메시지 기밀성을 모두 도시하는 도면이다.

도 20은 엔티티(AE1)가 홉 바이 홉 및/또는 엔드 투 엔드 보안을 위한 적절한 보안 크리덴셜들의 프로비저닝을 포함하여 CSE 또는 서비스 제공자와의 등록 프로세스를 개시하는 것을 도시하는 도면이다.

도 21a는 IoT 이벤트 관리 시스템들 및 방법들의 하나 이상의 개시된 실시예들이 구현될 수 있는 예시적인 머신 대 머신(M2M) 또는 사물 인터넷(IoT) 통신 시스템의 도면이다.

도 21b는 도 21a에 도시된 M2M/IoT 통신 시스템 내에서 사용될 수 있는 예시적인 아키텍처의 시스템도이다.

도 21c는 도 21a에 도시된 통신 시스템 내에서 사용될 수 있는 예시적인 M2M/IoT 단말 또는 게이트웨이 디바이스의 시스템도이다.

도 21d는 도 21a의 통신 시스템의 양태들이 구현될 수 있는 예시적인 컴퓨팅 시스템의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0015] 현재 oneM2M 사양들은 홉 바이 홉 인증만을 제공하므로, 원격 호스팅되는 서비스/리소스에 대해 CRUD(생성(Create), 리트리브(Retrieve), 업데이트(Update), 삭제(Delete)) 동작들을 수행하도록 요청하는 엔티티는 리소스를 호스팅하는 엔티티에 의해 명시적으로 인증되지 않는다. 문제는 다음과 같다.

[0016] ● 타겟 엔티티는 그것으로부터 하나의 홉만큼 떨어져 있는 엔티티만을 인증할 수 있기 때문에, 리소스를 호스팅하는 엔티티가 리소스에 대해 동작들을 수행하려고 시도하는 엔티티들을 완전히 인증할 수는 없으므로, 액세스 제어를 용이하게 시행할 수 없다.

[0017] ● 임의의 중간 엔티티들(예를 들어, MN-CSE, IN-CSE)이 홉 바이 홉 메커니즘들로 인해 다른 중간 엔티티들을 대신하여 메시지들을 흉내낼 수 있다.

[0018] ● 홉 바이 홉 메커니즘들은 (D)TLS를 사용하여 보호되어야 하기 때문에, 각각의 홉의 (D)TLS 세션은 각각의 홉에서 셋업, 무결성 보호 및 인증되어야 할 것이며, 가능하게는 각각의 홉들에서 암호화 및 복호화가 수행되어야 할 것이며, 따라서 추가적인 운영상의 오버헤드가 세션/서비스 계층에서 발생한다. 보안 프로비저닝 및 보안 연관 구축 프로시져들은 각각의 홉들에서 관련된 두 개의 엔티티들에 의해서만 행해진다.

[0019] 도 2는 엔드 투 엔드(E2E) 보안 페이즈들을 도시하는 도면이다. 엔드 투 엔드 인증 프로세스를 수행하는 데는 다음 단계들이 수반될 수 있다.

[0020] 도 2의 단계 1은 서비스 인에이블먼트 및 보안 구성 프로세스를 도시한다. 이 단계에서, 엔티티 A(202)는 서비스 인에이블링 기능(SEF)(204)과의 연관을 확립한다. 확립된 연관은 대역 내 또는 대역 외일 수 있고, 연관이 확립되기 전에 상호 인증 프로세스를 포함할 수도 있다. 연관 확립 프로세스의 일부로서, 엔티티 A(202)(202)에 의해 요청되거나 제공되는 서비스의 성질은 SEF(204)에 의해 식별된다. 또한, 엔티티 A(202)에 의해 요구되거나 요청되는 보안 요건들 및 특징들 또한 SEF(204)에 의해 식별된다. 간단히 말해서, 엔티티 A(202)의 보안 프로파일(Security Profile)(SP) 및 임의로 프라이버시 프로파일(Privacy Profile)(PP)은 엔티티 A(202)로부터 획득되거나, SEF에 의해 결정/추론 및 생성되거나, 제3 엔티티로부터 획득된다. 배치 시나리오들에 기초하여, 각각의 엔티티는 상이한 SP를 가질 수 있으며, 이는 고유한 SP-Id, 및 임의로 PP-Id에 의해 식별된 연관된 PP에 의해 식별될 수 있다.

[0021] 도 2의 단계 2는 보안 크리덴셜 프로비저닝 프로세스(Security Credential Provisioning Process)를 도시한다. 식별되었던 SP 및 대응하는 보안 요건들 및 피쳐들에 기초하여, 엔티티 A(202)는 적절한 보안 크리덴셜들로 프로비저닝된다. 엔티티 A(202)에 발행되었던 보안 크리덴셜들은 자신에 의해 사용되어, 엔티티 A(202)와의 보안 연관을 확립하고자 하는 엔티티들의 인증을 수행한다. 또한, 엔티티의 E2E 크리덴셜과 함께 제공될 수 있는 인가된 엔티티들의 리스트가 생성된다. 특정한 경우들에는, 보안 크리덴셜 프로비저닝 프로세스 동안, 보안 크리덴셜들을 생성하기 위해 요구되는 시드 재료만이 엔티티 A(202)에 제공된다. 적절한 보안 크리덴셜들을 생성하기 위해, 시드 재료가 기존의 보안 크리덴셜과 함께 사용될 수 있다. 적절한 엔드 투 엔드 보안(엔드 투 엔드 메시지 인증, 엔드 투 엔드 메시지 기밀성) 크리덴셜들을 생성하기 위해, 시드 재료가 크리덴셜 부트스트래핑 프로세스에서 함께 사용할 수도 있다. 부트스트래핑 프로세스는 하위 계층(예를 들어, 네트워크 계층/MAC 계층)에 존재하는 보안 연관에 기초하거나, 또는 상위 계층(예를 들어, 애플리케이션 계층 또는 서비스 계층)과의 기존의 보안 연관에 기초할 수 있다. 기존의 보안 연관이 존재하지 않는 일부 경우들에는, 엔드 투 엔드 보안 크리덴셜들이 생성되기 전에, 새로운 부트스트래핑 프로세스(예를 들어, GBA, MEF 기반)가 수행되어야 할 수 있다.

[0022] 도 2의 단계 3은 제3자 크리덴셜 요구 프로세스(Third-party Credential Requisition Process)를 도시한다. 또 다른 엔티티 N도 엔티티 A(202)에 의해 셋업된 보안 크리덴셜들로 프로비저닝될 수 있는데, 이는 엔티티 N(208)과 엔티티 A(202) 사이에 보안 연관이 확립될 수 있어, 엔티티 N(208)이 엔티티 A(202)에 의해 제공된 서비스들/리소스에 액세스하도록 혹은 그 반대가 될 수 있도록 하기 위함이다. 크리덴셜들로 프로비저닝되도록 인가되었던 엔티티들에만 크리덴셜들이 제공된다.

[0023] 도 2의 단계 4는 엔드 투 엔드 인증 프로세스를 도시한다. 이 단계에서, 엔티티 A(202) 및 엔티티 N(208)은 두 엔티티들 간에 직접적으로 엔드 투 엔드 인증 프로세스를 수행할 수 있거나 또는 임의로 다른 것(예를 들어, SEF)에 의해 인에이블될 수 있다.

[0024] 도 2에 도시된 단계들을 수행하는 엔티티들은, 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 2에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 2에 도시된 단계들을 수행한다. 또한, 도 2에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0025] **서비스 인에이블먼트 및 보안 구성(SESC) 프로세스**

[0026] SCSC 단계 동안, SEF(204)는 엔티티 A의 작업에 적합한 적절한 보안 요건들 및 피쳐들을 결정한다. 보안 요건들 및 피쳐들은 엔티티에 의해 제공되는 SP 및 PP에 기초하여 SEF(204)에 의해 결정될 수 있다. SP 및 임의로 PP는 일부 추론 프로세스를 사용하여 생성되거나, 또는 엔티티에 의해 명시적으로 제공되거나, 또는 시스템을 셋업하고 있을 수 있는 개인(예를 들어, 관리자)에 의해 구성될 수 있다.

표 1

[0027]

보안 요건	보안 도메인 내 (HCSE 또는 RCSE 에서)	엔드 투 엔드
메시지 발신자 진정성/무결성	높음	매우 높음
메시지 리플레이 보호	높음	매우 높음
부인 방지 능력	낮음	낮음
메시지 기밀성	중간	중간
이동 중인 데이터의 기밀성	높음	높음
휴지 중인 데이터의 기밀성	중간	높음
이동 중인 데이터의 무결성	높음	매우 높음
휴지 중인 데이터의 무결성	중간	매우 높음
서비스의 가용성	높음	매우 높음
데이터의 가용성	높음	매우 높음

[0028] **표 1 : 엔티티 A의 예시적인 보안 프로파일(SP)**

[0029] 표 1에는 SESC 프로세스의 일부로서 엔티티에 의해 SEF(204)에 제공될 수 있는 예시적인 SP가 도시된다. 대안적으로, SP는 엔티티에 의해 제공되는 서비스의 유형에 기초하여 SEF에 의해 추론될 수 있다. 다른 경우들에는, SP는 특정 엔티티에 대한 관리자에 의해 구성될 수 있으며, 그 후 서비스/네트워크 제공자에서의 서버와 같은 제3 엔티티로부터 SEF에 의해 페치된다.

표 2

[0030]

디바이스 능력	값들
프로세싱 능력	900 MHz
RAM	500 Kb
플래시	1MB
배터리	5.0 Micro-W/MHz
무선 능력	블루투스, WiFi
슬립 모드	슬립 / 딥-슬립
보안환경	예
신뢰된 플랫폼 모듈 (TPM)	아니오
OS / 버전	안드로이드 / 킷캣

[0031] **표 2 : 엔티티 A의 예시적인 디바이스 프로파일(DP)**

[0032] 엔티티(202)는 엔티티가 호스팅되는 디바이스 프로파일(Device Profile)(DP)을 제공할 수 있다. 표 2는, SEF(204)가 엔티티(202)와 동일한 디바이스 상에서 구현되는 경우, 디바이스의 운영 체제들에 쿼리함으로써 엔티티(202)에 의해 SEF(204)에 제공되거나 SEF에 의해 획득되는 예시적인 DP를 도시한다. 대안적으로, SEF(204)는 제3 엔티티로부터 DP를 획득할 수 있다.

표 3

[0033]

엔티티 프로파일	값들
서비스의 클래스	건강 관리
서비스의 유형	실시간
영향	중요 (생명 및 신체)
보안 레벨	높음
프라이버시 레벨	높음

[0034] **표 3 : 엔티티 A의 예시적인 엔티티 프로파일(EP)**

[0035] 게다가, 엔티티(202)는 또한 엔티티 프로파일(Entity Profile)(EP) 또는 애플리케이션 프로파일(Application Profile)(AP)을 SEF(204)에 제공할 수 있다. EP 및 AP란 용어는 나머지 문헌 내에서 상호교환 가능하게 사용될 수 있다. 예시적인 EP/AP가 표 3에 도시되어 있다. 대안적으로, SEF(204)는 EP를 추론하거나 제3 엔티티로부터 EP를 획득할 수 있다. 엔티티는, "실시간" 서비스를 제공하고, "중요" 영향을 미치며, "높은" 보안 및 "높은" 프라이버시를 요구하는 "건강 관리"에 속하는 애플리케이션이다. 특정 경우들에는, SEF는 SP 또는 보안 요건들을 직접 결정하기 위해 EP 및 DP만을 사용할 수 있다.

표 4

엔티티 프로파일	값들
서비스의 클래스	홈 오토메이션
서비스의 유형	거의 실시간
영향	낮음
보안 레벨	중간
프라이버시	낮음

[0036]

[0037] **표 4 : 엔티티 B의 예시적인 EP**

[0038] 표 4는 다른 엔티티인 엔티티 B의 예시적인 AP 또는 EP를 도시한다. 엔티티는 홈 오토메이션에 속하는 애플리케이션이며, 시스템이 고장난 경우에 영향이 "낮은" 것으로 간주되고, "중간"의 보안 프로파일 및 "낮은" 프라이버시 영향을 갖는다.

[0039] SEF(204)는 엔티티(202)에 적절한 보안 요건들을 결정하기 위해 SP, DP 및 EP를 사용할 수 있다. 보안 요건들을 결정할 때의 추론 프로세스는 SP 및/또는 DP 및/또는 EP 내에 제공되는 정보의 조합을 사용하여 수행될 수 있다. 프로파일들 중 어느 것도 존재하지 않으면, SEF(204)는 액세스 권한을 갖는 프로파일에 기초하여 최상의 판단을 사용한다. SEF(204)가 프로파일들에 액세스할 수 없는 경우들에는, 제3 엔티티로부터 프로파일들을 획득할 수 있다. SEF(204)가 적절한 보안 요건들 및 그에 따른 보안 피쳐들을 결정하기 위해서는, 최소한 EP 및 DP에 대한 액세스가 요구될 수 있다. 그러면, SEF(204)는 EP 및 DP를 사용하여 SP를 생성할 수 있다. 엔티티(202)가 SP를 제공할 수 있거나 SEF(204)가 획득할 수 있으면, SEF(204)는 보다 세밀한 보안 요건들 리스트를 생성할 수 있을 것이다. SEF(204)가 매우 상세한 보안 요건들을 결정할 수 있도록 SEF(204)가 엔티티 A(202)의 SP, DP 및 EP에 액세스하는 것이 이상적일 것이다.

[0040] 엔티티에 의해 제공되는 상기 정보에 기초하여, 적절한 보안 요건들이 결정될 수 있다. SEF(204)는 SP에 의해 강조되어 요구되는 보안, DP에 의해 제공되는 디바이스 능력, 및 엔티티에 의해 EP로 제공되는 서비스 유형의 조합에 기초하여 적절한 보안 요건들을 선택할 수 있다.

표 5

보안 요건	보안 도메인 내 (HCSE 또는 RCSE 에서)	엔드 투 엔드
메시지 발신자 진정성 / 무결성	디지털 서명	디지털 서명
메시지 리플레이 보호	타임스탬프 + 난스	타임스탬프 + 난스
부인 방지 능력	없음	없음
메시지 기밀성	보안 프로토콜	보안 프로토콜
이동 중인 데이터의 기밀성	보안 프로토콜	객체 보안
휴지 중인 데이터의 기밀성	객체 보안	객체 보안
이동 중인 데이터의 무결성	보안 프로토콜	객체 보안
휴지 중인 데이터의 무결성	객체 보안	객체 보안
서비스의 가용성	인증: 인증서 멀웨어 서비스	인증: 인증서 멀웨어 서비스
데이터의 가용성	인증: 공개 키	인증: 공개 키

[0041]

[0042] **표 5 : 엔티티 A에 대해 SEF에 의해 추론되는 보안 요건들**

[0043] 마찬가지로, 엔티티 B에 대해 SEF에 의해 추론되는 보안 요건이 표 6에 도시되어 있다.

표 6

보안 요건	보안 도메인 내 (HCSE 또는 RCSE 에서)	엔드 투 엔드
메시지 발신자 진정성 / 무결성	메시지 인증 코드 (MAC)	메시지 인증 코드 (MAC)
메시지 리플레이 보호	난스	난스
부인 방지 능력	없음	없음
메시지 기밀성	없음	없음
이동 중인 데이터의 기밀성	없음	없음
휴지 중인 데이터의 기밀성	없음	없음
이동 중인 데이터의 무결성	데이터 보안: 대칭 키	데이터 보안: 대칭 키
휴지 중인 데이터의 무결성	데이터 보안: 대칭 키	데이터 보안: 대칭 키
서비스의 가용성	인증: PSK	인증: PSK
데이터의 가용성	인증: PSK + ACP	인증: PSK + ACP

[0044]

[0045] **표 6 : 엔티티 B에 대해 SEF에 의해 추론되는 보안 요건들**

[0046] 상세한 보안 피처들은 표 7에 도시되어 있다.

표 7

엔티티 ID	보안 피처들	보안 도메인 내			엔드 투 엔드	
		알고리즘들	사이즈들	프로토콜(들)	알고리즘들	사이즈들
엔티티 A	메시지 발신자 진정성 / 무결성	HMAC-SHA-2	256 / 512	(D)TLS, JWS	HMAC-SHA-2	256 / 512
	메시지 리플레이 보호	난스	256	해당사항 없음	타임스탬프 / 난스 + 시퀀스 번호	256 비트
	부인 방지	없음		해당사항 없음	없음	
	메시지 기밀성	AES	112	(D)TLS	AES	192
	이동 중인 데이터의 기밀성	AES	192	(D)TLS, JWE	AES	192
	휴지 중인 데이터의 기밀성	AES	256	해당사항 없음	AES	256
	이동 중인 데이터의 무결성	HMAC-SHA-2	256	(D)TLS, JWS	HMAC-SHA-2	256
	휴지 중인 데이터의 무결성	HMAC-SHA-512	512	해당사항 없음	HMAC-SHA-512	512
	인증 메커니즘	대칭 키	256	(D)TLS	대칭 키	256
	인증되지 않은 사용	예				
	인증 프로세스	직접				
	보안 엘리먼트의 존재	예				
	데이터의 무결성					

[0047]

[0048] **표 7 : 엔티티 B에 대한 상세한 보안 피처들**

[0049] 따라서, "낮은" 보안, 그에 따른 보안 기능(들), 선택된 알고리즘들 및 키 사이즈들을 요구하는 서비스만을 제공하는 저전력, 저메모리 디바이스가 적절히 선택될 수 있다. 예를 들어, 선택된 메시지 인증 메커니즘은 160 비트 키들을 갖는 HMAC-SHA1일 수 있는 반면, 더 많은 프로세싱 및 메모리를 가지며 더 높은 보안을 요구하는 엔티티는 HMAC-SHA2 메커니즘에서 사용될 수 있는 256비트 키들이 프로비저닝될 것이다. 예를 들어, 순서대로 또는 우선 순위대로 SEF에 의해 추론되거나 엔티티 A(202)에 의해 제공되는 보안 요건들의 리스트는 다음과 같다.

[0050] ● 시그널링/제어 메시지들의 메시지 인증 및/또는 무결성

[0051] ○ 지원되는 알고리즘들 : HMAC-SHA2(선호)

[0052] ○ 키 길이들 : 256/512/1024 ...

[0053] ● 데이터 기밀성

[0054] ○ 지원되는 알고리즘들 : AES, DES ...

[0055] ○ 키 길이들 : 128/256/512 ..

[0056] ● 데이터의 무결성 : 필수

[0057] ● 인증 메커니즘들 :

[0058] ○ 대칭 키 및/또는

[0059] ○ 인증서들 및/또는

[0060] ○ 부트스트래핑 프로세스

[0061] ● 인증되지 않은 사용자들을 지원하는 능력

[0062] ● 지원되는 프로토콜들 : EAP/IPSec/(D)TLS/JWT

[0063] ● 인증 : 직접/위임/부분 위임 방식

[0064] SESC 프로세스의 마지막에, SEF(204)는 완전한 프로파일 및 엔티티의 능력들을 갖는다. 엔티티의 능력들에 대한 지식을 가지면, SEF(204)가 엔티티의 작업들, 엔티티에 의해 제공되는 데이터 및 서비스, 및 엔티티와의 통신을 보호하기 위해 구현되어야 하는 적절한 보안 조치들 및 피쳐들을 결정하는 데 도움을 준다. SEF(204)는 엔티티의 능력들에 대한 표를 유지 관리한다. SEF에서 유지 관리되는 예시적인 표는 다음과 같다.

표 8

[0065]

[0066] **표 8 : 각각의 엔티티에 대해 지원되는 보안 피쳐 및 기능들**

[0067] **보안 크리덴셜 프로비저닝(Security Credential Provisioning)(SCP) 프로세스**

[0068] SCP 프로세스는 보안 크리덴셜 요청 프로세스 및 보안 크리덴셜 프로비저닝 프로세스의 단계들을 포함할 수 있다.

[0069] 보안 크리덴셜 요청 프로세스의 프로세스는 엔티티에 의해 또는 엔티티를 대신하여 SEF(204)에 의해 개시될 수 있다. 엔티티에 의해 제공되는 능력 및/또는 서비스의 유형에 기초하여, 적절한 보안 크리덴셜들 및 추가적으로, 다른 구성 파라미터들이 바람직하게는 신뢰된 제3자(Trusted Third-party)(TTP) 상에서 호스팅되는 키 유도 기능(Key Derivation Function)(KDF)(206)에 요청된다. 엔티티와 TTP 간의 인증은 임의적일 수 있다. SEF(204)는 KDF(206)의 역할을 수행할 수 있지만, 확장성 관점에서, TTP/KDF(206) 기능은 상이한 엔티티에 의해 수행될 수 있다. SEF(204)가 엔티티 A(202)를 대신하여 크리덴셜들을 요청하는 경우, SEF(204)는 TTP/KDF(206)와 상호 인증될 수 있다.

[0070] 보안 크리덴셜 프로비저닝 프로세스에서, KDF(206)는 키(들)를 생성하고, 키들이 어떻게 사용될 수 있는지 및 어떤 목적인지(MAC, 암호화, 어떤 계층에서 보호가 적용될 것인지 및 포함될 연관된 파라미터들 등), 키(들)가 사용될 수 있는 방식의 스코프 및 그것이 사용되는 컨텍스트를 기술하며, 임의적으로 새로운 ID가 생성되고 사

용될 알고리즘(들)이 권고될 수 있다. TTP/KDF(206)는 아래와 같이 나타낼 수 있는 표를 유지 관리한다.

표 9

컨텍스트 ID	인증서	키(들)	키 사이즈 (비트)	스코프 / 알고리즘	유효성 (secs)	인증 파라미터들
엔티티 A-컨텍스트 1	해당사항 없음	34B2342E...	256	암호화: AES	259,200	해당사항 없음
		3CC2342AF..	128	메시지 인증: HMAC-SHA1	259,200	Time 및 Nonce
		3BB1234E....	256	마스터 세션 키 / 부트스트래핑 키	604,800	해당사항 없음
엔티티 B-컨텍스트 1	엔티티 B-컨텍스트 1-인증서	52689A2D	128	암호화: AES	259,200	해당사항 없음
		37894621F..	128	메시지 인증: HMAC-SHA1	259,200	Time 및 Nonce
		7028CCE....	256	마스터 세션 키 / 부트스트래핑 키	604,800	해당사항 없음

[0071]

[0072] **표 9 : 각각의 엔티티와 연관된 보안 연관 및 크리덴셜들**

[0073] ContextID, 연관된 키들 및 다른 연관된 파라미터들 및 스코프가 요청 엔티티 또는 SEF(204)에 제공된다. 인증 파라미터들은 보안 프로세스(예를 들어, 인증 프로세스)의 일부로서 포함될 수 있는 보안 정보를 지시할 수 있다. 확립되는 각각의 보안 컨텍스트는 유효한 수명을 가지며, 그 후에는 컨텍스트가 업데이트되거나 새로운 것이 생성될 수 있다. ContextID는 크리덴셜들(키들, 알고리즘들 등) 및 연관된 스코프와 파라미터들을 식별하는 데 사용될 수 있다.

[0074] **제3자 크리덴셜 요구 프로세스**

[0075] 제3자 자격 크리덴셜 요구 단계에서, (엔티티 A(202)와 같은) 다른 엔티티와 엔드 투 엔드 인증을 수행하도록 요구되는 엔티티 N(208)은, 키잉 재료, 키들과 연관된 스코프, E2E 보안 연관이 생성될 수 있도록 메시지 인증 및 기타 정보를 입증하기 위해 사용될 수 있는 파라미터들을 요청한다. 요청 엔티티는 임의로 TTP/KDF(206)에 의해 인증될 수 있고, 또한 엔티티가 E2E 키들로 프로비저닝되도록 인가되었는지를 결정한다. 여기부터, TTP 및/또는 KDF(206)는 TTP로 지칭될 것이다. 엔티티는 컨텍스트 ID, URI, 포트 번호, 연관된 키(들), 스코프 및 연관된 파라미터들로 프로비저닝된다. 생성되는 키들은 두 엔드 엔티티들에 더 적합하도록 조정될 수 있다. 임의로, 다른 레벨의 키 생성 프로세스가 발생할 수 있다. 엔티티 N에서는, 보안 연관들을 생성하고 유지 관리하고자 하는 엔티티들에 의한 다음 파라미터들을 유지 관리할 수 있다.

표 10

리소스 ID	컨텍스트 ID	인증의 유형	포트 번호	인증 프로토콜	크리덴셜 (키/인증서)	보호 계층	유효성	파라미터들
엔티티 A 의 URI	엔티티 A-컨텍스트 1	홉 바이 홉		HMAC-SHA2	2341234E....	서비스 계층: JWS/JWT	3,600	Nonce, Time
엔티티 B 의 URI	엔티티 B-컨텍스트 1	엔드 투 엔드	8443	DTLS	3569424...	세션 계층	7000	

[0076]

[0077] **표 10 : 각각의 엔티티에서 사용되는 인증 메커니즘, 스코프 및 파라미터들**

[0078] 상기 표에서, 엔티티 N(208)이 엔티티 A(202)와의 E2E 인증을 수행하기 위해서는, 임의적인 파라미터일 수 있는 컨텍스트 ID(엔티티A-컨텍스트1)(EnityA-Context1)가 제공될 수 있음을 알 수 있다.

[0079] 컨텍스트 ID : E2E 인증을 확립하기 위해 사용될 보안 피처들/파라미터들을 식별하는 데 사용될 수 있다. ContextID는 E2E 보안 크리덴셜들 및 연관된 스코프 및 파라미터들을 식별하는 데 사용된다. ContextID는 랜덤하게 또는 암호 프로세스를 사용하여 생성될 수 있다. ContextID는 엔티티 또는 트랜잭션의 임시 아이덴티티로서 사용될 수 있다.

[0080] 리소스 ID : 이는 엔티티 N이 E2E 인증 프로세스 및 연관을 생성하고자 하는 엔티티의 아이덴티티(예를 들어, 엔티티의 URI 또는 도메인 이름, IP@ 등)이다.

[0081] 포트 번호 : 세션 계층 E2E 인증의 경우, 포트 번호는 임의로 제공될 수 있다.

[0082] 프로토콜 : 서비스 계층 E2E의 경우, 프로토콜은 단지 사용되는 메시지 인증 알고리즘(예를 들어, HMAC-SHA2)을 지시하지만, 세션 계층의 경우, 프로토콜은 (DTLS 또는 TLS 또는 기타일 수 있는) 프로토콜을 지시한다. 이는 세션 또는 서비스 계층에만 국한되지 않을 수 있고, 애플리케이션 계층들과 연관된 프로토콜들(예를 들어, 보안 RTP) 또는 IPSec, EAP 등과 같은 다른 하위 계층 프로토콜들을 포함할 수 있다.

[0083] 파라미터들 : 키 소유/메시지 인증의 증명을 제공하는 데 사용될 수 있는 값들(예를 들어, Nonce, Time, Random 챌린지 등)을 지시한다.

[0084] 인증의 유형 : 인증들이 수행될 수 있는 계층을 결정한다. 이들은 서비스, 세션, 네트워크, MAC 계층에서 수행될 수 있는 인증들을 포함한다. 본 개시내용에서는 서비스 및 세션 계층들에서의 인증 메커니즘들에 관심이 있다.

[0085] 엔티티 A(202)와 연관된 엔드 투 엔드 크리덴셜들이 TTP에 의해 엔티티 N으로 지칭되는 제3자에게 프로비저닝될 수 있거나 또는 요구된 키잉 재료가 엔티티 N(208)에 프로비저닝되는데, 이는 엔티티 N(208)이 엔드 투 엔드 보안 보호들, 즉 엔티티 A(202)와 엔티티 N(208) 사이의 엔드 투 엔드 메시지 인증, 엔드 투 엔드 메시지 기밀성, 엔드 투 엔드 데이터 기밀성 및 엔드 투 엔드 데이터 무결성을 검증 또는 제공하기 위해 사용되는 적절한 보안 크리덴셜들을 생성할 수 있게 하기 위함이다. 생성될 수 있는 크리덴셜들의 유형들의 리스트는 표 참조에서 제공된다.

표 11

[0086]

보안	생성되고 사용되는 대칭 키들	파라미터들
메시지 발신자 진정성 / 무결성	Ke2e_EntityA_EntityN_msg_auth	없음
메시지 리플레이 보호	Ke2e_EntityA_EntityN_msg_auth	Nonce / Time / Seq#
부인방지 능력	해당사항 없음	해당사항 없음
메시지 기밀성	Ke2e_EntityA_EntityN_msg_conf	IV
이동 중인 데이터의 기밀성	Ke2e_EntityA_EntityN_data_conf 또는 Ke2e_EntityA_EntityN_msg_conf	IV
휴지 중인 데이터의 기밀성	Ke2e_EntityA_EntityN_data _conf 또는	IV
이동 중인 데이터의 무결성	Ke2e_EntityA_EntityN_msg_auth	없음
휴지 중인 데이터의 무결성	Ke2e_EntityA_EntityN_data_auth	Time

[0087] **키잉 재료 생성하기**

[0088] KDF(206)를 채택하는 TTP는 엔티티 N(208)의 인증을 수행할 수 있고, 그 후 엔티티 N이 인가된 경우, 엔티티 N은 적절한 EntityA_EntityN 특정 엔드 투 엔드 키들로 프로비저닝된다. 엔티티 A(202)에 의해 사전 프로비저닝된 사전 구성된 EntityA_EntityN 특정 키들이 엔티티 N(208)에 제공된다. Ke2e_EntityA_master가 프로비저닝되었으면, TTP는 적절한 Ke2e_EntityA_EntityN 특정 키들을 생성하고 이들을 엔티티 N(208)에 프로비저닝한다. 대안적으로, TTP는 엔티티 N(208)에 Ke2e_EntityA_EntityN 키만을 제공하고, 엔티티 N(208)이 엔티티 N(208)에 의한 보안 보호에 필요한 다양한 키들을 생성할 수 있도록 필요한 시드 재료를 엔티티 N(208)에 제공한다. 생성된 다양한 키들은 메시지 인증을 위한 E2E_MAC_Key로서 문헌 내에서 참조되는 Ke2e_EntityA_EntityN_msg_auth, 메시지 기밀성을 위한 Ke2e_EntityA_EntityN_msg_conf, 데이터 기밀성을 제공하기 위한 Ke2e_EntityA_EntityN_data_conf 및 엔드 투 엔드 데이터 무결성을 제공하기 위한 Ke2e_EntityA_EntityN_data_auth일 수 있다.

[0089] 참고 : 특정 도면들에서, 엔드 투 엔드 Ke2e_EntityA_EntityN_msg_auth 및 Ke2e_EntityA_EntityN_msg_auth는 일반적으로 KpsaE2E라고 지칭될 수 있다.

[0090] Ke2e_EntityA_master는 엔티티 A(202) 및 TTP에 의해 수행되는 인증 프로세스에 기초하여 엔티티 A(202) 및 TTP에 의해 생성될 수 있다. Ke2e_EntityA_master는 엔티티 A(202)와 TTP 사이에서 수행되는 부트스트래핑 프로세스의 결과일 수 있다. 또한, Ke2e_EntityA_master는 엔티티 A(202)와 TTP 사이의 인증 및 인증(예를 들어, TLS 또는 DTLS 또는 GBA)을 수행하는 데 사용되는 인증 채널에 채널 바인딩될 수 있다. 부트스트래핑된 프로세스 : GBA와 같은 부트스트래핑 메커니즘들은 각각의 엔티티 쌍과 연관될 수 있는 Ke2e 키들을 유도하기 위해 사용될 수 있다. E2E 관점에서 엔티티들을 인증하고자 하는 엔티티(예를 들어, EntityA)는 GBA를 사용하여 TTP에 의해 인증될 수 있다. GBA 프로세스를 사용하여 EntityA를 인증한 결과로서 생성되는 마스터 E2E 키는 다음의 형태와 같을 수 있다.

[0091] Ke2e_EntityA_master: 148735880652C65238B432A….(256 비트)

[0092] Ke2e_EntityA_master는 엔티티 A(202)뿐만 아니라, 엔티티 A(202)와 TTP 간의 성공적인 상호 인증에 기초한 TTP 부트스트래핑뿐에 의해 생성될 수 있다.

[0093] 엔티티-특정 키들이 TTP에 의해 생성되고, 프로비저닝되거나, 또는 엔티티-특정 엔드 투 엔드 키들이 생성될 수 있도록 시드 재료가 각각의 엔드 엔티티들에 제공된다. 엔드 투 엔드 키들을 생성하는 예시적인 메커니즘들이 아래에 나와 있다.

[0094] Ke2e_EntityA_EntityB = HMAC-SHA256 (Ke2e_EntityA_master , "Bootstrap Process"

[0095] || Entity_B-ID || Random1)

[0096] Ke2e_EntityA_EntityC= HMAC-SHA256 (Ke2e_EntityA_master, "Bootstrap Process" || Entity_C-ID || Random2)

[0097] Ke2e_EntityA_EntityN = HMAC-SHA256(Ke2e_EntityA_master, "Bootstrap

[0098] Process" || Entity_N-ID || Random3)

[0099] 엔티티 A와 엔티티 N 사이의 메시지들에 대해 각각 엔드 투 엔드 메시지 진정성뿐만 아니라 엔드 투 엔드 메시지 기밀성을 제공하기 위해 사용되는 연관된 Ke2e_EntityA_EntityN_msg_auth 및 Ke2e_EntityA_EntityN_msg_conf 키들을 생성하기 위해, 키 확장 메커니즘들이 엔티티 A 및 엔티티 N에 의해 사용될 수 있다. 엔드 투 엔드 키들에 대한 키 확장의 예가 제공된다.

[0100] Ke2e_EntityA_EntityN_msg_auth = HMAC-Hash

[0101] (Ke2e_EntityA_EntityN_master, T(0) | “E2E Message Authentication Key”| 0x01)

[0102] Ke2e_EntityA_EntityN_msg_conf = HMAC-Hash (

[0103] Ke2e_EntityA_EntityN_master, T(1)| “E2E Message Confidentiality Key”|0x02)

[0104] 단일 키에 기초한 AEAD 암호 프로세스가 사용되면, 상기 키들 중 하나의 키만이 생성된다.

[0105] 서비스 계층(Service Layer) : 서비스 계층에서의 E2E 인증이 사용되며, 여기서 홉 바이 홉 보호 메커니즘들이 여전히 사용될 수 있지만, 이에 더하여 E2E 메시지 원래의 인증이 사용된다. 또한, 보안 중요사항인 것으로 간주되는 정보 및 파라미터들이 서비스 계층에서 보호될 수 있다. 보호는 JSON 웹 서명(JSON Web Signature)(JWS)을 통해 제공될 수 있다. 메타 데이터만이 중간 노드들에 의해 프로세싱될 수 있다. 메타 데이터는 메시지 인증 코드(Message Authentication Code)(MAC) 키로서 역할을 하는 E2E 키에 기초하여 E2E JSON 웹 서명에 의해 무결성 보호될 수 있고, JSON 웹 서명과 같은 JSON 포맷을 사용하여 표현될 수 있다. AES-CCM 및 AES-GCM과 같은 연관된 데이터를 이용한 인증 암호화(Authentication Encryption with Associated Data)(AEAD)-클래스와 같은 암호 알고리즘들을 사용하면 엔드 투 엔드 메시지 진정성뿐만 아니라 메시지 기밀성을 모두 제공할 수 있다. 메시지 진정성을 제공하고 체크하는 데 사용되는 연관된 데이터를 식별한다. 연관된 데이터(Associated Data)는 메시지 헤더로 구성될 수 있으며, 이는 메시지 기밀성이 요구되는 경우들에는 암호화되지 않는다. 대안적으로, 임의의 중간 노드들에 의해 수정되지 않은 전체 메시지가 메시지 인증 코드를 생성하는 데 사용될 수 있다. 앞서 언급된 바와 같이, 메시지의 메타 데이터라고 불리는 메시지 헤더의 서브세트는 AEAD 알고리즘 내에서 연관된 데이터로서 사용될 수 있으며, 이는 MAC의 계산에 사용된다. 또한, MAC이 다른 수단을 사용하여 생성되고 독점 수단을 사용하여 표현되는 것도 가능할 수 있다. 메시징 내에서 MAC 및 MAC의 표현을 생성하는 데 사용되는 메커니즘들과 관계없이, 중간 노드들에 의해 수정 또는 제거되지 않은 전체 메시지는 시간 컴포넌트와 연관되는 Nonce 또는 메시지가 생성된 Time 및 Nonce(시간 의존적일 수 있는 매우 큰 랜덤 값) 둘 다의 조합을 사용함으로써 리플레이 공격들로부터 보호될 수 있다. 대안적으로, 메시지가 전송될 때마다 증분되는 각각의 메시지에 대한 시퀀스 번호가 서명 생성 프로세스 동안에 사용될 수 있거나 또는 Nonce와 함께 Time 대신에 사용될 수 있다. 대안적으로, 리플레이 보호를 위해 메시지의 시퀀스 번호가 Time 및 Nonce와 함께 포함된다. 예를 들어, 서명 또는 MAC 또는 인증 태그(Auth_Tag)는 다음과 같이 유도될 수 있다.

[0106] MAC = HMAC-SHA-256 (Ke2e_EntityA_EntityN_msg_auth, “E2E_ServiceLayerMAC” || OriginData || Time || Nonce)

[0107] 또는

[0108] MAC = HMAC-SHA-256 (Ke2e_EntityA_EntityN_msg_auth, “E2E_ServiceLayerMAC” || OriginData || Message Sequence Number || Nonce)

[0109] "OriginData" 대신에, 메시지와 연관된 완전한 메시지 또는 메타 데이터가 사용될 수 있다.

[0110] Ke2e_EntityA_EntityN_msg_auth : E2E 인증을 요청하는 엔티티에 프로비저닝된 키이다. 여기서는, 엔티티 A와 엔티티 N 사이의 엔드 투 엔드 메시지 인증 키를 암시한다. 일반적으로, 두 엔티티들(예를 들어, 엔티티 A(202) 및 엔티티 N(208))에 의해 공유되는 대칭 키이다. 공개 키잉 메커니즘의 경우,

Ke2e_EntityA_EntityN_msg_auth는 메시지를 서명하는 데 사용된 개인 키(E2E_DS_Key : 엔드 투 엔드 디지털 서명 키라고도 함)이며(서명하는 엔티티에만 알려짐), 공개 키를 포함하는 인증서를 사용하여 다른 엔티티에 의해 검증될 수 있다. 인증서가 없는 공개 키 메커니즘의 경우, 엔드 엔티티는 E2E 인증이 수행되는 엔티티의 공개 키로 프로비저닝되어야 한다. 대안적인 실시예에서, 공개 키 메커니즘은 본질적으로 대칭적이고 엔티티들에 의해 공유되는 Ke2e_EntityA_EntityN_msg_auth를 유도하는 데 사용될 수 있다.

[0111] OriginData : 원래 요청에 관한 정보를 포함하는 데이터로서, 이 데이터는 실제 메시지의 메타 데이터로 간주될 수 있지만, 실제 메시지의 발신자에 관한 정보를 또한 포함한다. "OriginData"가 중간 노드들에 의해 수정되지 않았다고 가정한다. OriginData는 메시지 헤더 내에 포함되는 정보, 즉 Originator-Id, Destination-Id, Resource-Id, Type-of-Operation뿐만 아니라 Session-Id의 서브세트를 포함할 수 있다.

[0112] Time : 임의적일 수 있고, 원래 메시지가 생성되었을 때의 타임 스탬프를 제공한다.

[0113] Nonce : 시간 컴포넌트와 연관되고 세션과 연관된 랜덤 값으로서, 리플레이 공격으로부터 보호한다.

[0114] Sequence Number (Seq#) : 이것은 메시지를 식별하는 고유 번호이다. 일부 경우들에는, Seq#은 Session-Id와 동일할 수 있다.

[0115] 세션 계층(Session Layer) : DTLS 또는 TLS에 의한 E2E 인증이 사용된다. 이것은 홉 바이 홉 보안 메커니즘들을 우회한다. 엔드 엔티티들은 상호 인증되며, 보안 연관이 확립된다. 이는 진정한 E2E 방식(직접) 또는 위임 모드들에서 엔티티들 간에 수행될 수 있다.

[0116] **엔드 투 엔드 인증 프로세스**

[0117] E2E 인증 프로세스는 진정한 E2E 방식으로, 또는 위임되거나 부분적으로 위임된 방식으로 수행될 수 있다. 엔티티에 의해 제공되거나 선택된 스코프에 기초하여, E2E 인증 프로세스는 다음을 사용하여 수행될 수 있다.

[0118] 대칭 키 : 이전에 설명한 바와 같이, E2E 인증 크리덴셜들을 요청한 엔티티는 E2E 인증을 수행하는 데 사용되는 대칭 키들, 스코프 및 파라미터로 프로비저닝될 수 있다. 직접 또는 위임 시나리오들에서, 서비스 계층 E2E 또는 세션 계층 E2E 인증에 대칭 키가 사용될 수 있다. 스코프 및 연관된 파라미터들이 제공되는 한, 엔티티는 그에 따라 키들을 사용할 수 있다. E2E 인증 키들(Ke2e_EntityA_EntityN_msg_auth)는 주기적으로 재생성(re-generate)될 수 있다. 유사하게, Ke2e_EntityA_master는 각각의 크리덴셜과 연관된 수명에 기초하여 주기적으로 생성될 수 있다.

[0119] 인증서 기반/공개 키 : 프로비저닝되는 크리덴셜들은 인증서들의 형태로 표현된 공개 키들 또는 단순한 공개/개인 키들, 아이덴티티 기반 암호화 또는 공개 키잉 메커니즘들에 기초한 다른 메커니즘들에 기초할 수 있다. 세션 계층 인증을 위한 E2E 인증 키들(ke2e)은 인증을 위해 인증서들을 사용하여 인증된 디피-헬먼(Authenticated Diffie-Hellman) 프로세스를 사용하는 엔티티들 간에 생성될 수 있다.

[0120] **위임 대 직접 보안 메커니즘들:**

[0121] 엔티티가 인증을 위해 "높은 무결성(High Integrity)" 또는 "상위 보증 등급(Higher degree of Assurance)"을 요구하는 경우, 프로세싱 요건들은 비례하여 더 높을 수 있고, 엔티티의 능력들(예를 들어, 메모리/프로세싱)이 제한적이라면, 엔티티는 위임 방식으로 보안 기능들을 수행하는 것으로 선택할 수 있다. 보다 복잡한 보안 기능들(예를 들어, E2E 인증, 보안 스토리지, 순방향 비밀성)을 수행하기 위해, 엔티티는 인증 및 다른 보안 기능들을 신뢰된 제3 엔티티(예를 들어, SEF(204))에 위임할 수 있다. 위임된 인증을 수행하는 다른 이점은 위임된 에이전트(예를 들어, SEF)가 다수의 E2E 인증들을 함께 결합할 수 있다는 것이다.

[0122] 엔티티가 자체적으로 E2E 인증 및 다른 보안 동작들을 수행할 수 있는 경우, 엔티티는 위임의 필요없이 자체적으로 직접 인증을 수행하도록 선택할 수 있다. SEF(204)는 디바이스 능력들 또는 서비스 요건들에 기초하여 그 자체로 위임을 위한 옵션을 선택할 수 있다(예를 들어, 시그널링 또는 다른 동작 오버헤드를 줄임). 보안 기능들의 일부는 위임되는 반면, 다른 보안 기능들은 직접 수행되는 경우에는 하이브리드 접근법이 사용된다.

[0123] 도 3a 및 도 3b는 엔티티 A(202)와 엔티티 B(302) 사이의 예시적인 E2E 동작들을 도시하는 도면이다.

[0124] 도 3a 및 도 3b의 단계 1에서, 엔티티 A(202) 및 SEF1(204)(예를 들어, 사전에 상호 신뢰 프로비저닝된 제1 홉 엔티티)은 인에이블된 보안 통신에 의해 인증되는 (D)TLS 터널을 확립한다. 보안 터널을 사용하면, 서비스 인에이블링 보안 구성(Service Enabling Security Configuration)(SESC) 프로세스가 발생하며, 여기에서 엔티티 A의 프로파일이 생성되고, 보안 요건들이 결정된다.

[0125] 도 3a 및 도 3b의 단계 2에서, 엔티티 A(202)는 그 자신과 엔티티들의 인가된 리스트(예를 들어, 엔티티 B(302), 엔티티 C ... 엔티티 N) 사이에서 E2E 키들의 확립을 임의로 요청할 수 있다. 요청은 엔티티 A(202)에 의해 SEF1(204)에 전송되고, SEF1(204)는 요청을 TTP/KDF(206)에 전송할 수 있다. 대안적으로, 엔티티 A로부터의 명시적인 메시지가 필요 없이, SEF1(204)는 TTP(206)에 의한 E2E 키들의 생성을 요청할 수 있다. 이 시나리오에서, SEF1(204)는 E2E 키들이 제공되는 엔티티들의 인가된 리스트를 결정할 것이다. 대안적으로, 엔티티 A(202)와 TTP(206) 사이에 신뢰 관계가 존재하는 경우, 엔티티 A(202)는 TTP/KDF(206)에 직접적으로 키 요구 및 엔티티들의 인가된 리스트를 전송할 수 있다. 엔티티 A(202)가 TTP의 인증서로 프로비저닝되거나 또는 TTP와 엔티티 A(202) 사이의 공유 기밀이 사전 프로비저닝되는 것이 가능할 수 있다. 그 시나리오가 작동하기 위해서는 SEF1(204)에 의존할 필요없이 직접적으로 엔티티 A(202)를 인증하도록 TTP(206)가 크리덴셜들을 가져야 한다는 것에 주의하여야 한다.

[0126] 도 3a 및 도 3b의 단계 3에서, 엔티티 A(202)의 능력들, 스코프에 기초하여, TTP는 엔티티 A(202)와 연관된 Ke2e_EntityA_master를 생성하고, 크리덴셜 요구가 SEF1으로부터 시작된 경우에는, 생성된 마스터 키는 SEF1(204)과 연관된 Ke2e_SEF1_master일 수 있다. Key가 어떻게 사용될 수 있는지에 대한 추가적인 파라미터들, 및 Key 및 Key 용도(usage)를 식별하는 ContextID가 또한 생성된다. 임의로, TTP는 다음과 같은 방식으로 MasterKey를 사용하여 E2E 엔티티 특정 키인 E2E 대칭 키들을 생성할 수 있다.

[0127] a. 예를 들어, $\mathrm{Ke2e\_EntitiyA\_EntityB\_master} = (\mathrm{Ke2e\_EntityA\_master}, \text{“Entity B}$

[0128] $\mathrm{ID} \parallel \mathrm{Parameters}\text{”})$

[0129] 여기서, 엔티티 B ID는 엔티티 B의 아이덴티티(예를 들어, 엔티티 B의 URI)를 나타내며, 엔티티 A(202) 또는 SEF1에 의해 제공된다.

[0130] Ke2e_EntitiyA_EntityB-master : 이것은 엔티티 B에 대해 엔티티 A(202)를 인증하는 데 사용되는 E2E 대칭 키이며, 반대의 경우도 마찬가지이다.

[0131] 도 3a 및 도 3b의 단계 4에서, TTP는 EntityA의 E2E 마스터 키, 및 임의로 E2E 엔티티-특정 대칭 키들의 리스트를 포함하는 키들을 SEF1(204)에 제공한다. SEF1(204)는 키들을 엔티티 A(202)에 포워딩할 수 있다. 대안적으로, SEF1(204)이 요구를 한 경우, 키들은 SEF1(204)에 저장되고 엔티티 A(202)에 포워딩되지 않는다. 이것은 SEF1(204)이 엔티티 A(202) 대신에 위임된 인증(Delegated Authentication)을 수행할 때 적용 가능하다.

[0132] 도 3a 및 도 3b의 단계 5에서, 엔티티(예를 들어, 엔티티 B(302))는 SEF2(304)와 SESC 프로세스를 수행한다. SEF1(204) 및 SEF2(304)가 동일할 수 있는 일부 시나리오들에서, 만약 동일한 경우, E2E 인증 프로세스는 생략될 수 있고, 또는 TTP를 포함할 필요 없이, 키 요구가 단순화되는 것이 가능하다.

[0133] 도 3a 및 도 3b의 단계 6에서, 엔티티 B(302)는 엔티티 A(202)와의 통신에 사용될 E2E 대칭 키를 요청하기 위해 TTP에 요청한다. 엔티티 B(302)는 임의로 TTP에 의해 직접적으로 인증될 수 있고, 또는 대안적으로 TTP는 (D)TLS 접속에 기초하여 SEF2(304)를 신뢰한다. 대안적인 실시예에서, SEF2(304)는 엔티티 B를 대신하여 TTP에 대해 요청을 수행할 수 있다. 다른 실시예에서, SEF2(304)는 자신에 대한 E2E 엔티티-특정을 요청할 수 있으며, 이 경우, 보다 동적인 키 생성 메커니즘이 TTP에 의해 사용될 수 있다.

[0134] 도 3a 및 도 3b의 단계 7에서, TTP는 엔티티 B(302)가 엔티티 A의 E2E 키로 프로비저닝되도록 엔티티 A(202)에 의해 인가되었다고 결정한다. 그 다음, TTP(206)는 E2E 엔티티 특정 키(Ke2e_EntitiyA_EntityB_master)를 SEF2(304)에 포워딩하며, SEF2(304)는 이를 엔티티 B(302)에 포워딩한다. 대안적으로, SEF2(304)가 위임된 인증을 제공하면, SEF2(304)는 키를 저장할 수 있다. 위임된 인증의 경우, TTP에 의해 프로비저닝되는 키는 Ke2e_EntityA_SEF2-master일 수 있다. 엔티티 A(202)는 SEF2(304)를 인가하지 않았지만, TTP는 SEF2 특정 키를 생성하고, 위임된 인증을 사용하고 있음을 지시하기 위해 파라미터들 내에 적절한 정보를 제공할 수 있다. 그러한 시나리오에서, 엔티티 A(202)는 제공된 파라미터들과 함께 프로비저닝되는 Master Key를 사용하여 SEF2-특정 키를 유도할 것이다.

[0135] 도 3a 및 도 3b의 단계 8에서, 세션 계층을 통해 발생할 수 있는 임의의 메시징 및 수행되는 대응하는 동작들(예를 들어, Create, Retrieve, Update 또는 Delete)은 MAC 또는 JSON 웹 서명(JWS), 또는 키 프로비저닝 프로세스 동안에 제공된 E2E 엔티티-특정 키 및 파라미터들에 기초하여 메시지 발신자 인증을 증명할 수 있는 임의의 다른 수단을 사용하여 보호될 수 있다.

[0136] 도 3a 및 도 3b에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 3a 및 도 3b에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 3a 및 도 3b에 도시된 단계들을 수행한다. 또한, 도 3a 및 도 3b에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0137] **실시예들**

[0138] 본 개시내용에서 설명되는 메커니즘들은 인증을 수반하는 환경들에 적용될 수 있고, 보다 구체적으로는, 제약되는 것으로 간주되는 엔티티들(예를 들어, IoT/M2M 디바이스들)의 E2E 인증에 적용 가능하다. 그러나, 이는 IoT 디바이스들에만 한정되지 않고, 제약된 디바이스들이 복잡한 보안 기능들을 수행하는 것을 완화시켜주는 것에 더하여, 시스템에 전체적으로 수반되는 메시징 오버헤드를 완화하기 위해 신뢰된 엔티티가 적절한 보안 피쳐들, 기능들 및 크리덴셜들을 결정할 수 있는 곳에서 사용할 수 있다. 이하의 하위 섹션들에서 기술되는 실시예는 oneM2M 사양들과 관련된다. 여기서는, 호스팅 CSE에서 SEF(204)를 호스팅하는 것으로 제안한다. 일부 경우들에서, CSE는 TTP/KDF(206) 지원을 또한 제공할 수 있지만, 확장성의 관점에서, TTP/KDF(206)는 M2M 서비스 제공자 CSE에서 또는 인증서 인가 기관(Certificate Authority)으로서 본 개시내용에서 설명된 바와 같은 추가된 기능에 의해 호스팅될 수 있다.

[0139] 도 4a 및 도 4b는 oneM2M 실시예를 도시하는 도면들이다. oneM2M은 능력 서비스 기능(Capability Service Function)들(CSF(404))이라고 지칭되는 oneM2M 서비스 계층에 의해 지원되는 능력들을 정의한다. oneM2M 서비스 계층은 능력 서비스 엔티티(Capability Services Entity)(CSE(402))라고 지칭된다. 일 실시예에서, 도 4a에 도시된 바와 같이, 제안된 서비스 인에이블링 기능(204)은 oneM2M CSF로서의 CSF(408)에서 호스팅될 수 있다. 도 18b에 도시된 바와 같이, 키 전달 기능(Key Delivery Function)(206)은 oneM2M CSF로서의 CSF(412)에서 호스팅될 수 있다.

[0140] **서비스 인에이블먼트 및 보안 구성(SESC)**

[0141] SESC는 도 5a 및 도 5b에 도시된 보안 크리덴셜 요구 및 프로비저닝(Security Credential Requisition and Provisioning)(SCRP) 페이즈를 포함할 수 있으며, 여기서 엔티티 CSE3(502)는 E2E 인증 크리덴셜들의 셋업을 요청한다. E2E 크리덴셜들은 CSE3(502)에서 E2E 인증이 수행되게 하기 위해 다른 엔티티들에 의해 사용될 수 있다. 메시징 세부 사항은 다음과 같다.

[0142] 도 5a 및 도 5b의 단계 0은 홉 바이 홉 인증 크리덴셜들을 셋업하기 위한 키 프로비저닝 스텝(Key Provisioning Step)이다. 이 단계는 현재 oneM2M 사양들에 기초하여 수행될 수 있다. 이것은 오프라인으로 수행될 수 있다. 키 프로비저닝 단계의 결과로서, CSE3(502) 및 호스팅 CSE(HCSE)(504)가 대칭 키(Kpsa1)로 프로비저닝된다.

[0143] 도 5a 및 도 5b의 단계 1에서, CSE3(502) 및 HCSE(504)는 인증을 위한 기초로서 Kpsa1을 사용하여 DTLS 접속을 셋업한다.

[0144] 도 5a 및 도 5b의 단계 2에서, DTLS 인증의 일부로서, 세션 키들이 확립된다.

[0145] 도 5a 및 도 5b의 단계 3에서, CSE3(502)는 oneM2M 리소스의 생성에 대한 필요성 및 E2E 크리덴셜들의 생성에 대한 요청을 지시하는 "Create Request" 메시지를 전송한다. CREATE Request 메시지는 DTLS 세션 키들에 의해 보호된다. CSE3(502)는 E2E 크리덴셜들을 사용할 수 있는 인가된 엔티티들의 리스트를 제공한다.

[0146] 도 5a 및 도 5b의 단계 4에서, HCSE(504)는 메시지의 출처가 실제로 DTLS 세션 키들을 사용하는 것에 의해 AE1로부터 왔는지를 검증한다.

[0147] 도 5a 및 도 5b의 단계 5에서, CSE3(502)에 대한 호스팅 CSE인 HCSE(504)는 oneM2M 사양들에 명시된 메커니즘들에 기초하여 CSE3(502)에 대한 리소스를 생성한다. 또한, 전술한 바와 같이 서비스 인에이블먼트 프로세스 동안 추론되거나 획득될 수 있는 CSE3(502)의 능력들에 기초하여, HCSE(504)는 디바이스의 능력들에 기초하여 적절한 E2E 크리덴셜들에 대한 요청을 생성한다. 또한, 사용될 수 있는 보안 크리덴셜들 및 파라미터들의 용도에 대한 스코프를 제공한다. 스코프는 서비스 계층/세션 계층 E2E 인증일 수 있고, 파라미터들은 리플레이 보호를 위해 사용될 수 있는 정보, 메시지 인증에 사용되는 정보(예를 들어, 이는 메시지 또는 메타 데이터 등의

발신자의 진정한 아이덴티티를 식별)를 포함한다.

[0148] 도 5a 및 도 5b의 단계 6에서, 사전 확립된 보안 크리덴셜들(PSK)을 사용하여 HCSE(504)와 TTP/KDF(206) 사이에서 TLS 세션이 셋업된다.

[0149] 도 5a 및 도 5b의 단계 7에서, 크리덴셜들, 스코프, 용도 및 파라미터들에 대한 Request가 보안 TLS 터널을 사용하여 HCSE(504)로부터 TTP로 전송된다.

[0150] 도 5a 및 도 5b의 단계 8에서, TTP는 HCSE(504)에 의해 제공된 디바이스 능력 정보에 기초하여 HCSE에 의해 요청된 바와 같은 적절한 크리덴셜들을 생성한다. 디바이스 능력이 낮다면, 적절한 알고리즘(예를 들어, HMAC-SHA1 또는 3DES 또는 다른 저 리소스 요구 알고리즘)이 올바른 키 사이즈와 함께 선택된다. 스코프, 파라미터들과 함께 크리덴셜들이 데이터베이스에 저장된다. 생성된 크리덴셜들은 "Ke2e_CSE3_master" 키라고 불릴 수 있으며, 그것과 연관된 적절한 키 핸들/컨텍스트 ID를 가질 수 있다. CSE3(502)가 TTP와 직접 접속하는 경우들에서는, Ke2e_CSE3_master 키가 TTP에 의해 CSE3(502)에 직접 포워딩될 수 있다. 키들은 CSE3(502)와 TTP 간에 확립된 (D)TLS 접속을 사용하여 수송될 수 있다.

[0151] 도 5a 및 도 5b의 단계 9에서, 크리덴셜들은 그 후 필요한 스코프 및 파라미터들과 함께 HCSE(504)에 포워딩된다.

[0152] 도 5a 및 도 5b의 단계 10에서, HCSE(504)는 다른 관련 정보와 함께 크리덴셜들을 CSE3(502)에 포워딩한다.

[0153] 도 5a 및 도 5b의 단계 11에서, 메시지는 HCSE(504)로부터 수신된 것으로 검증된다.

[0154] 도 5a 및 도 5b의 단계 12에서, 스코프 및 파라미터들과 함께 크리덴셜들을 키 스토어(Keystore)에 저장한다.

[0155] 도 5a 및 도 5b에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 5a 및 도 5b에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 5a 및 도 5b에 도시된 단계들을 수행한다. 또한, 도 5a 및 도 5b에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0156] 도 20은 엔티티(AE1(602))가 홉 바이 홉 및/또는 엔드 투 엔드 보안을 위한 적절한 보안 크리덴셜들의 프로비저닝을 포함하여 CSE 또는 서비스 제공자와의 등록 프로세스를 개시하는 것을 도시한다. 적절한 크리덴셜들이 AE1(602)과 연관된 DP, SP 및/또는 EP에 기초하여 결정될 수 있다.

[0157] 도 20의 단계 1에서, AE1(602)은 CSE1(604)과의 접속 요청을 개시한다. 접속 요청은 등록 요청일 수 있다.

[0158] 도 20의 단계 2에서, CSE1(604)은 AE1(602)과 연관된 프로파일, 파라미터들을 가지지 않기 때문에, 단계 3에서 IN-CSE(2002)로부터 가입 프로파일을 요청한다.

[0159] 도 20의 단계 4에서, IN-CSE(2002)는 AE1(602)과 연관된 M2M-가입 프로파일(M2M-Subscription Profile)을 CSE1(604)에 전송한다.

[0160] 도 20의 단계 5에서, CSE1(604)은 서비스 또는 네트워크 제공자의 네트워크 외부에 위치할 수 있는 SP 레포지토리(2004)에 SP를 요청할 수 있다. 도 20의 단계 6에서, AE1(602)과 연관된 AE1_SP를 포함하는 응답이 CSE1(604)에 전송된다.

[0161] 도 20의 단계 7에서, CSE1(604)은 AE1_DP, 즉, AE1(602)과 연관된 DP, 및/또는 AE1_EP, 즉, AE1(602)과 연관된 EP 또는 AP를 DP/EP 레포지토리(2006)로부터 요청할 수 있다. 단계 8에서, AE1_DP 및/또는 AE1_EP를 포함하는 응답이 CSE1(604)에 전송된다.

[0162] 도 20의 단계 9에서, SP, DP 및/또는 EP에 기초하여, CSE1(604)은 AE1(602)과의 통신을 안전하게 하기 위하여 보안 요건들의 올바른 세트, 및 그에 따른 연관된 보안 피처들 및 파라미터들을 결정한다.

[0163] 도 20의 단계 10에서, CSE1(604)은 CSE1(604)에 의해 수행되는 평가에 기초하여 M2M 등록 기능(M2M Enrollment Function)(TTP/KDF)에 적절한 보안 크리덴셜들을 요청한다. 크리덴셜 요청은 명시적이거나 암시적일 수 있으며, 보다 세분화된 보안 요건 또는 보다 덜 세분화된 요건을 제공할 수 있다.

[0164] 도 20의 단계 11에서, M2M 등록 기능(MEF)(2008)은 AE1(602)과의 부트스트래핑 프로세스를 개시하고, 적절한 부트스트래핑된 세션 크리덴셜들을 생성한다.

[0165] 도 20의 단계 12에서, MEF(2008)는 AE1(602)과 연관된 CSE1-특정 엔드 투 엔드 크리덴셜들(Ke2e_AE1_CSE1_master)을 생성하고, 그것을 CSE1(604)에 프로비저닝한다. 대안적으로, MEF(2008)는 Kpsa_AE1_CSE1을 생성할 수 있고, 그것을 CSE1(604)에 프로비저닝한다. 또한, MEF(2008)는 크리덴셜들과 연관된 UsageInfo 및 ContextInfo를 프로비저닝할 수도 있다.

[0166] 도 20의 단계 13에서, AE1(602)은 CSE1-특정 엔드 투 엔드 크리덴셜들을 생성한다. Ke2e_AE1_CSE1_master 및 연관된 Ke2e_AE1_CSE1_msg_auth 및/또는 Ke2e_AE1_CSE1_msg_conf 크리덴셜들이 정책들 및 UsageInfo 및 ContextInfo에 따라 또한 생성될 수 있다. 대안적으로, AE1(602)은 홉 바이 홉 보안을 위해 사용되는 Kpsa_AE1_CSE1을 생성할 수 있다.

[0167] 도 20의 단계 14에서, CSE1(604)은, MEF(2008)에 의해 Ke2e 크리덴셜들로 프로비저닝되지 않았고 Ke2e_AE1_CSE1_master뿐만 아니라 엔드 투 엔드 크리덴셜들의 생성을 위해 요구되는 시드 재료로만 프로비저닝된 경우, Ke2e_AE1_CSE1_msg_auth 및/또는 Ke2e_AE1_CSE1_msg_conf를 생성한다.

[0168] 도 20에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 20에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 20에 도시된 단계들을 수행한다. 또한, 도 20에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0169] **제3자 크리덴셜 요구 페이즈**

[0170] 다른 엔티티(예를 들어, CSE3(502))에 의해 호스팅되는 리소스를 리트리브하고 싶은 엔티티(예를 들어, AE1(602))가 다른 엔티티의 E2E 크리덴셜들을 요청하고자 하는 실시예가 다음의 도면에 도시되어 있다. 도 6a 및 도 6b는 제3자 크리덴셜 요청 페이즈를 도시하는 도면이다.

[0171] AE1(602) 및 CSE1(604) 및 TTP(206)는 각각의 엔티티들에서 oneM2M 사양에 의해 특정된 바와 같이 키 스토어들 내에 저장된 대칭 키들로 모두 사전 프로비저닝된다고 가정한다. AE1(602)이, AE와 호스팅 CSE 사이의 홉 바이 홉 연관을 셋업하기 위한 크리덴셜들을 획득하는 데에 사용되는 TTP(206)의 E2E 크리덴셜들로만 사전 프로비저닝되는 것을 상상하는 것도 가능할 수 있다. 메시징 세부 사항은 다음과 같다.

[0172] 도 6a 및 도 6b의 단계 1에서, AE1(602)은 CSE1(604)과 Kpsa1을 사용하여 DTLS 보안 연관을 셋업한다.

[0173] 도 6a 및 도 6b의 단계 2에서, 각각의 엔티티는 서로를 인증하고 세션 키들을 셋업한다.

[0174] 도 6a 및 도 6b의 단계 3에서, AE1(602)은 CSE3(502)에 의해 호스팅되는 리소스를 타겟으로 하는 "RETRIEVE Request" 메시지를 임의적인 E2E 크리덴셜 요청 메시지와 함께 전송한다. CSE1(604)가 E2E 인증 크리덴셜들이 요구되는지 결정할 수 있기 때문에, E2E 크리덴셜 요청은 임의적일 수 있다.

[0175] 도 6a 및 도 6b의 단계 4에서, RETRIEVE Request 메시지는 DTLS 터널 내에서 포워딩되고, 메시지의 출처는 CSE1(604)에 의해 검증된다.

[0176] 도 6a 및 도 6b의 단계 5에서, AE1(602)의 능력들에 기초하여 CSE1(604)은 CSE3(502)에 대한 크리덴셜들, 스코프 및 파라미터들에 대한 요청을 생성한다.

[0177] 도 6a 및 도 6b의 단계 6에서, CSE1(604)은 PSK를 사용하여 TTP와 TLS 접속을 셋업한다.

[0178] 도 6a 및 도 6b의 단계 7에서, CSE3의 크리덴셜들, 스코프, 파라미터들 및 임의로 AE1의 바람직한 보안 능력들에 대한 요청이 또한 제공될 수 있다.

[0179] 도 6a 및 도 6b의 단계 8에서, AE1(602)이 SCRP 페이즈 동안 엔티티 CSE3(502)에 의해 인가되었고, 인가된 엔티티들의 리스트에 있다면, CSE3 크리덴셜들에 대한 요청에 기초하여, TTP는 CSE3(604)와 연관된 크리덴셜들을 리트리브한다.

[0180] 도 6a 및 도 6b의 단계 9에서, 스코프, 파라미터들과 같은 다른 관련 정보와 함께 CSE3(604)의 크리덴셜들이 TLS 터널을 사용하여 CSE1에 전송된다. 위임된 인증이 수행되고 있는 경우, CSE1은 크리덴셜들을 임의로 저장할 수 있다.

[0181] 도 6a 및 도 6b의 단계 10에서, CSE1은 CSE3의 크리덴셜들 및 연관된 정보와 함께 AE1에 RETRIVE Response 메시지를 전송한다.

[0182] 도 6a 및 도 6b의 단계 11에서, 메시지는 AE1에 의해 검증된다.

[0183] 도 6a 및 도 6b의 단계 12에서, AE1(602)은 키 스토어 내에 CSE3의 크리덴셜들 및 연관된 파라미터들을 저장한다.

[0184] 도 6a 및 도 6b에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 6a 및 도 6b에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 6a 및 도 6b에 도시된 단계들을 수행한다. 또한, 도 6a 및 도 6b에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0185] 부트스트래핑 프로세스에 기초한 실시예가 도 14에 도시되고 여기에서 설명된다. 마스터 크리덴셜(Master Credential) 및 마스터 크리덴셜 식별자(Master Credential Identifier) 또는 프로비저닝된 보안 접속 키(Provisioned Secure Connection Key)(Kpsa) 및 프로비저닝된 보안 접속 키 식별자(Provisioned Secure Connection Key Identifier)(KpsaId)의 원격 프로비저닝을 요청하는 AE 또는 CSE를 등록자(Enrollee)라고 한다. 등록자와 보안 연관을 확립하는 AE 또는 CSE를 등록자 B라고 한다. 등록자와 공유 키를 확립하는 AE 또는 CSE 또는 M2M 인증 기능(M2M Authentication Function)(MAF)을 등록 타겟(Enrolment Target)이라고 한다. oneM2M 시스템은 사전 프로비저닝된 대칭 등록자 키를 지원하며, 이 키는 이들 엔티티들의 상호 인증을 위해 등록자 및 M2M 등록 기능(M2M Enrollment Function)(MEF)에 사전 프로비저닝된 대칭 키이다. 마찬가지로, 인증서 기반 메커니즘 또는 로우(raw) 공개 키가 등록자 및 MEF에서 프로비저닝될 수 있다. 등록자 및 MEF는 인증서의 공개 검증 키들을 신뢰하기 전에 서로의 인증서를 확인해야 한다. 보안 핸드셰이크(Security Handshake) 내에서, M2M 등록 기능은 자체 개인 서명 키를 사용하여 세션 파라미터들의 디지털 서명을 생성하고, 등록자는 M2M 등록 기능의 공개 검증 키를 사용하여 디지털 서명을 검증한다. 그런 다음, 역할들이 뒤바뀐다. 등록자가 디지털 서명을 생성하고, M2M 등록 기능이 이를 검증한다. 대안적으로, GBA 기반 프로비저닝 메커니즘이 사용된다. 이 경우, MEF의 역할은 GBA 부트스트랩 서버 기능(Bootstrap Server Function)(BSF)에 의해 수행된다. 이 프레임 워크는 3GPP 또는 3GPP2 대칭 키들을 사용하여, 등록자 및 MEF(GBA BSF이기도 함)를 인증한다. 세부사항들은 3GPP TS 33.220 및 3GPP2 S.S0109-A에 의해 특정된다.

[0186] 등록자 및 M2M 등록 기능은 엔티티가 다른 엔티티에 대해 자신을 인증하는 데 사용할 부트스트랩 크리덴셜(Bootstrap Credential)로 사전 프로비저닝된다. 이 사전 프로비저닝을 위한 메커니즘들은 디바이스 관리자 기능을 사용하거나 글로벌 플랫폼(Global Platform)에 의해 특정된 신뢰된 서비스 관리자(Trusted Service Manager)(TSM)와 같은 메커니즘을 사용하여, 공장에서 자동화된 관리자에 의해 수행될 수 있다. Kpsa로 지칭되는 "M2M 보안 확립을 위해 프로비저닝된 크리덴셜들" 및 그 연관된 식별자인 KpsaId, 및 Km으로 지칭되는 "마스터 크리덴셜" 및 그 연관된 "마스터 크리덴셜 식별자"인 KmId를 확립하는 프로세스는 oneM2M에 대한 TS-0003 사양 내의 8.3.1.2 절에서 설명된 것과 같은 메커니즘들을 따른다. Km 및/또는 Kpsa가 생성되고 나면, 이들은 E2E 크리덴셜들을 생성하기 위해 "마스터 크리덴셜"로 사용될 수 있다. 사양들은 부트스트랩 크리덴셜 구성, 부트스트랩 명령어 구성, 부트스트랩 등록 핸드쉐이크, 등록 키 생성 및 연관 보안 핸드쉐이크 프로시져들에 대한 통합(Integration to Association Security Handshake procedures)을 수행하는 메커니즘들을 설명한다. 이 개시내용에서는, "엔드 투 엔드 크리덴셜들 생성"이라는 추가적인 프로세스를 추가할 것을 제안한다.

[0187] 우리는 적어도 다음의 파라미터들, 즉 컨텐츠 정보(Content Info), 레이블(Label) 및 솔트(Salt)를 제공함으로써 등록 타겟 또는 MAF에게 엔드 투 엔드 크리덴셜들을 생성하는 능력을 제공하는 메커니즘에 의해 "등록 페이즈(Enrollment Phase)"를 향상시킬 것을 제안하고 있다. 컨텐츠 정보는 생성되는 크리덴셜들의 유형에 대한 충분한 정보, 엔드 투 엔드 크리덴셜들을 생성할 수 있기 위해 따라야 하는 메커니즘들 또는 표준들 등을 등록 타겟에게 제공한다. 크리덴셜들의 예시적인 유형은, 엔드 투 엔드 메시지 인증 크리덴셜들, 엔드 투 엔드 데이터

보안 크리덴셜들, 크리덴셜들이 공개 키인지 또는 대칭 키인지 여부에 대한 정보, 키들의 길이, 따라야 할 알고리즘들/프로토콜들 등일 수 있다. 레이블은 RFC 5809 또는 RFC 5246 또는 RFC 5705 또는 임의의 기타 표준화된 키 유도 기능들 및 키 확장에 의해 설명된 용도에 기초하여 그 크리덴셜들을 생성하는 데 사용되는 필요한 정보를 제공한다. 컨텍스트 정보 및 레이블은 등록자에 의해 직접 제공되거나, MEF에 의해 등록 타겟에게 제공될 수 있다. 솔트는 키 생성 메커니즘의 일부로서 사용되는 랜덤 값이다. 바람직한 접근법은 등록자가 등록 페이즈의 일부로서 초기 메시지 동안에 등록 타겟에게 솔트를 제공하는 것이다. 솔트는 등록자와 등록 타겟 간의 초기 통신에 기초하여 계산되는 해시 값일 수도 있다.

[0188] "엔드 투 엔드 크리덴셜들의 생성" 프로세스의 일부로서, 등록자 및 등록 타겟은 엔드 투 엔드 마스터 키, 즉 Ke2e_AE_CSE_master를 생성하기 위해 Kpsa_AE_CSE를 마스터 키로서 사용하여 엔드 투 엔드 크리덴셜들을 생성한다. 대안적으로, 타겟이 MAF인 경우, Km은 엔드 투 엔드 마스터 키를 생성하기 위한 마스터 키로서 사용될 것이다. RFC 5809를 사용하는 엔드 투 엔드 키 생성의 예가 아래에 제공된다.

[0189] Ke2e_AE_CSE_master = HMAC-Hash (Salt, Kpsa_AE_CSE)

[0190] T(0) = empty string (zero length)

[0191] Ke2e_AE_CSE_msg_auth = T(1) = HMAC-Hash (Ke2e_AE_CSE_master, T(0)

[0192] | “E2E Message Authentication Key”| 0x01)

[0193] Ke2e_AE_CSE_message_confidentialtiy = T(2) = HMAC-Hash

[0194] (Ke2e_AE_CSE_master, T(1)| “E2E Message Confidentiality Key”|0x02)

[0195] 유사하게, 데이터 기밀성 및 데이터 무결성 키들이 등록 타겟 및 등록자에 의해 생성된다. 이 프로세스는 등록자와 등록 타겟 간에 공유되는 고유한 Enrollee-EnrolmentTarget_Ke2e_master(예를 들어, AE 및 CSE 특정 엔드 투 엔드 키들)에 기초하여 각각의 등록자 및 연관된 등록 타겟에 의해 반복된다. 일부 경우들에서는, 복수의 등록 타겟들에 의해 공유되고 MEF에 의해 등록 타겟들에 프로비저닝될 수 있는 Ke2e_master만이 등록자에 대해 생성되며, 등록자는 각각의 엔드 엔티티들에 대해 고유한 엔드 투 엔드 키들을 생성할 수 있다.

[0196] 특정한 경우들에는, Kpsa/Km이 Ke2e_master로서 사용될 수 있고, 상술한 프로세스는 각각의 엔드 투 엔드 보안 보호, 즉 메시지 인증, 메시지 무결성, 데이터 무결성 및 데이터 기밀성을 위한 고유 키들을 생성하는 데 사용된다.

[0197] 특정한 다른 경우들에는, 메시지 인증, 메시지, 메시지 기밀성, 데이터 무결성, 데이터 기밀성, 키 생성 키 등에 대해 오직 하나의 키인 Kpsa 또는 Km만이 사용된다.

[0198] 특정한 다른 경우들에는, 세션 키가 Kpsa 또는 Km으로부터 생성되고, 이는 그 후 엔드 투 엔드 보안 보호 메커니즘들, 즉 메시지 인증, 메시지 기밀성, 데이터 무결성 및 데이터 기밀성 각각에 대한 고유한 키들을 생성하는 데 사용된다.

[0199] 특정한 다른 경우들에는, Kpsa 또는 Kpm으로부터 생성되는 단일 세션 키만이 엔드 투 엔드 메시지 인증, 기밀성, 데이터 무결성 및 데이터 기밀성을 제공하는 데 사용된다.

[0200] 특정한 다른 경우들에는, MEF는 Ke2e_master 또는 등록 타겟 또는 MAF에 대해 다음의 키들, 즉,

[0201] Ke2e_AE_CSE_msg_auth, Ke2e_AE_CSE_msg_conf, Ke2e_AE_CSE_data_auth,

[0202] Ke2e_AE_CSE_data_conf 및 Ke2e_key_generation

[0203] 의 세트 또는 서브세트를 프로비저닝할 수 있다.

[0204] 도 15a 및 도 15b는 AE에서의 리소스 표현 연관, 및 어트리뷰트들, 즉 홉 바이 홉 보안 크리덴셜뿐만 아니라 엔드 투 엔드 크리덴셜을 각각 갖는 <securityParameters> 리소스 구조를 제공한다. 도 16a 내지 도 16c는 앞서 설명한 엔티티 프로파일, 디바이스 프로파일 및 보안 프로파일들의 리소스 표현들을 나타낸다.

[0205] **E2E 인증 페이즈**

[0206] E2E 인증 페이즈 동안에, 키 생성 페이즈 동안보다 이전에 결정된 스코프에 기초하여, 애플리케이션, 서비스, 세션 또는 다른 계층들에서 인증이 수행될 수 있다. 또한, 인증은 직접 모드에서 또는 위임 모드를 사용하여 수행할 수 있다.

[0207] 직접 모드를 사용하는 서비스 계층에서의 E2E 인증

[0208] 도 7a 및 도 7b는 AE1(602)이 CSE3(502) 상에서 호스팅되는 원격 리소스에 Update 동작을 요청하는 E2E 인증을 도시한다. 이 도면은 직접 모드를 사용하는 서비스 계층 E2E 인증을 도시한다. 도시된 메커니즘은 oneM2M 사양들과 매우 흡사하다. 메시징 세부사항들은 다음과 같다.

[0209] 도 7a 및 도 7b의 단계 1에서, AE1(602)은 Kpsa1을 사용하여 CSE1(604)과 DTLS 접속을 셋업한다.

[0210] 도 7a 및 도 7b의 단계 2에서, AE1(602)은 CSE3(502) 상에서 호스팅되는 리소스에 대해 UPDATE 동작을 수행하라는 Request를 전송한다. 위에서 설명한 제3자 크리덴셜 요구 페이즈 동안에, AE1(602)은 이전에 획득된 E2E 인증 키(Ke2e_CSE3_AE1_msg_auth)를 이용하여 메시지 인증 코드(Message Authentication Code)(MAC)를 생성한다. MAC은 사용될 알고리즘, 발신 인증을 제공하기 위해 사용되는 파라미터들, 리플레이 보호 등을 포함하는 제공된 스코프에 기초하여 생성된다. MAC은 Request 메시지의 일부로서 제공되며 DTLS 터널을 사용하여 보호된다.

[0211] 도 7a 및 도 7b의 단계 3에서, oneM2M 사양에 의해 특정된 메커니즘들을 사용하여 요청을 프로세싱한다.

[0212] 도 7a 및 도 7b의 단계 4에서, 일단 요청이 프로세싱되고 나면, 응답이 AE1에 전송된다.

[0213] 도 7a 및 도 7b의 단계 5에서, CSE1(604)은 Kpsa2를 사용하여 다음 홉인 CSE2(702)와 DTLS 접속을 생성한다.

[0214] 도 7a 및 도 7b의 단계 6에서, CSE1(604)은 전달 리소스 요청 메시지를 생성하고, 그것을 AE1(602)에 의해 포함된 MAC과 함께 다음 홉인 CSE2(702)에게 포워딩한다.

[0215] 도 7a 및 도 7b의 단계 7에서, CSE2에서 요청을 프로세싱한다. CSE2는 CSE3의 URI와 기타 관련 정보를 파일하기 위해 전달 Req 데이터를 프로세싱한다.

[0216] 도 7a 및 도 7b의 단계 8에서, CSE1(604)에 응답을 전송한다.

[0217] 도 7a 및 도 7b의 단계 9에서, CSE2(702)는 Kpsa3을 사용하여 CSE3(502)과 DTLS 접속을 셋업한다.

[0218] 도 7a 및 도 7b의 단계 10에서, CSE2(702)는 전달 리소스 요청 메시지를 생성하고, 그것을 AE1에 의해 포함된 MAC과 함께 다음 홉인 CSE3에 포워딩한다.

[0219] 도 7a 및 도 7b의 단계 11에서, CSE3(502)는 메시지 발신을 검증한다.

[0220] 도 7a 및 도 7b의 단계 12에서, CSE3(502)는 AE1(602)과 연관된 메시지에 포함된 MAC을 검증한다. CSE3(502)가 E2E 크리덴셜들(KpsaE2E)을 가지지 않으면, CSE3(502)는 마스터 키들을 TTP로부터 획득한 다음, AE1의 아이덴티티에 기초하여 E2E 키를 생성할 수 있다. CSE3(502)는 또한, 메시지가 파라미터들(예를 들어, Nonce/Time 스탬프)을 사용하여 리플레이되지 않았으며, AE1(602)이 원래 메시지의 발신자로서 검증되었고, MAC이 실제로 AE1에 의해 계산되고 삽입되었다는 것을 검증한다.

[0221] 도 7a 및 도 7b의 단계 13에서, Request에 대한 Response는 CSE3(502)에 의해 CSE2(702)로 다시 제공된다.

[0222] 대안적으로, 단계들 4 및 8의 메시지들은 단계 13이 수행될 때까지의 단계들 후에 전송될 수 있다. CSE2(702)가 CSE3(502)으로부터 응답을 수신하면(메시지 13), CSE2는 CSE1(604)에 응답을 전송한 후(단계 8의 메시지), CSE1은 엔티티에 응답을 전송한다(단계 4의 메시지).

[0223] 도 7a 및 도 7b에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 7a 및 도 7b에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 7a 및 도 7b에 도시된 단계들을 수행한다. 또한, 도 7a 및 도 7b에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0224] **위임 모드를 사용하는 서비스 계층에서의 E2E 인증**

[0225] 도 8a 및 도 8b는 위임 모드 접근 방식을 사용하는 서비스 계층에서의 E2E 인증을 도시한다. 도 7a 및 도 7b와 관련하여 설명된 직접 모드와 여기서 설명되는 접근 방식 간의 주요 차이점은 CSE1(604)(호스팅 CSE)이 AE1(602)을 대신하여 E2E 인증을 수행한다는 것이다. CSE1(604)은 AE1(602)을 대신하여, 위에서 전술한 제3자 크리덴셜 요구 프로세스를 수행한다. 또한, 약간의 수정된 실시예는 스코프 정보가 MAC 대신에 JSON 웹 서명(JSON Web Signing)(JWS)/JSON 웹 토큰 표현의 사용을 제안한다는 것이다. 사용된 파라미터들은 MAC 계산에 사용된 파라미터들과 유사할 수 있으며, 표현은 JWT에 기초하며 위에 설명된 보안 프로비저닝 프로세스 동안에 합의된다. 메시징 세부 사항들은 다음 메시지들을 제외하고는 도 7a 및 도 7b와 관련하여 설명된 내용과 매우 유사하다.

[0226] 도 8a 및 도 8b의 단계 1에서, Request 메시지는 MAC을 포함하지 않으며, 따라서 AE1(602)은 E2E 방식으로 인증될 수 없다.

[0227] 도 8a 및 도 8b의 단계 3 내지 단계 5는 이전에 설명된 시나리오들과 유사하다.

[0228] 도 8a 및 도 8b의 단계 6에서, CSE1(604)은 엔드 엔티티 CSE3(502)가 CSE1(604)을 인증할 수 있도록 MAC과 유사한 JWS를 생성한다. 여기서, CSE1(604)은 AE1(602)을 대신하여 인증을 수행하는 데에 위임된다. JWS는 Request 메시지 내에 포함된다. JWS는 TTP로부터 획득된 Ke2e_AE1_CSE1_msg-auth를 사용하여 계산될 수 있다.

[0229] 도 8a 및 도 8b의 단계 7 내지 단계 9는 이전에 설명된 시나리오들과 유사하다.

[0230] 도 8a 및 도 8b의 단계 10에서, JWS를 포함하는 Request 메시지는 홉 바이 홉 방식으로 CSE3(502)에 포워딩된다.

[0231] 도 8a 및 도 8b의 단계 11에서, CSE3(502)은 그것이 메시지의 타겟임을 검증한다.

[0232] 도 8a 및 도 8b의 단계 12에서, CSE3(502)는 원래의 Request가 AE1(602)을 대신하여 CSE1(604)에 의해 전송되었음을 검증한다. JWS를 검증함으로써 발신자가 실제로 CSE1(604)이었으며, 그것이 리플레이되지 않았음을 검증한다.

[0233] 도 8a 및 도 8b에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 8a 및 도 8b에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 8a 및 도 8b에 도시된 단계들을 수행한다. 또한, 도 8a 및 도 8b에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0234] 도 17은 신뢰되거나 또는 신뢰가 떨어지거나 또는 심지어 신뢰할 수 없는 중간 홉들을 횡단하여, 복수의 서비스-계층 홉들만큼 서로 떨어져 있는 2개의 엔티티들(AE2 및 CSE1(604)) 사이의 대칭 키 메커니즘에 의한 엔드 투 엔드 메시지 인증 및 무결성 체크를 나타내는 실시예를 기술하는 도면이다. 클라이언트 애플리케이션(AE2)은 다른 애플리케이션 엔티티(AE1(602))의 리소스에 대해 업데이트 동작을 수행하고자 한다. 리소스는 호스팅 CSE(CSE1(604) 상)에서 호스팅되기 때문에, AE2는 리소스의 위치로 사전 프로비저닝되었거나 또는 리소스 위치(/CSE1/R-ID)를 발견하기 위해 발견 서비스를 사용한다. CSE1(604)은 인가된 엔티티들만이 AE1 리소스에 대해 Create, Retrieve, Update, Delete 또는 Notify 동작들 중 임의의 동작을 수행할 수 있도록 보장하고자 한다. CSE1(604)가 인가된 엔티티들만이 CRUD 동작들을 수행할 수 있도록 보장할 수 있게 하기 위해, CSE1(604)은 메시지들의 소스가 인증되고, 메시지들이 메시지의 발신자에 의한 키의 소유를 검증함으로써 무결성 보호될 것을 요청할 수 있다. 메시지의 생성을 준비함에 있어서, AE2는 적절한 메시지 인증 키인 Ke2e_AE2_CSE1_msg_auth를 TTP로부터 획득하거나, 또는 그것에 프로비저닝되었거나 또는 전술한 "제3자 크리덴셜 요구 프로세스"를 사용하는 부트스트래핑 프로세스를 사용하여 생성된 엔드 투 엔드 마스터 키인 Ke2e_AE2_CSE1_master로부터 키들을 생성해야 한다. 키 외에도, AE2의 메시지의 진정성 및 무결성을 검증하기 위해 CSE1(604)에 의해 사용될 수 있는 올바른 인증 태그를 AE2가 생성할 수 있도록 하기 위해 요구되는 컨텍스트 정보, 용도 정보, 및 레이블이 AE2에 대해 획득, 생성 또는 프로비저닝된다. AE2는 엔드 투 엔드 메시지 인증을 수행하기 위해 키 스토어로부터 적절한 크리덴셜들을 선택한다.

[0235] 도 17의 단계 1에서, AE2는 자신과 CSE2(702) 간의 (D)TLS 접속을 셋업하기 위해 Kpsa_AE2_CSE2를 사용한다.

접속 확립 프로세스는 oneM2M 사양들의 TS-0003 릴리스 1에 기술된 메커니즘들을 따른다.

[0236] 도 17의 단계 2에서, AE1(602)은 CSE1(604) 상에서 호스팅되는 AE1의 리소스(/CSE/R-ID로 식별됨)에 대해 "Update" 동작을 수행하는 데 사용되는 oneM2M "Request" 메시지를 생성한다. Request 메시지는 M2M-Request-ID1에 의해 고유하게 식별된다. AE2는 OriginData라고도 하는 메시지 헤더 정보를 사용하여 인증 태그(Auth_Tag) 또는 메시지 인증 코드(MAC)를 생성한다. 대안적으로, Auth_Tag를 생성하기 위해 전체 메시지가 입력으로서 사용된다. 다음 정보는 Auth_Tag 생성의 일부로서 사용될 수 있다.

[0237] Auth_Tag = HMAC-SHA-256 (Ke2e_AE2_CSE1_msg_auth, “Message

[0238] Header” | Nonce | Time)

[0239] 대안적으로, Auth_Tag = HMAC-SHA-256 (Ke2e_AE2_CSE1_msg_auth,

[0240] “Entire Message” | Nonce | Time)

[0241] Nonce, Time 또는 둘 모두가 Auth_Tag의 생성에 포함될 수 있다. 특정 경우들에는, 각각의 세션에 대해 고유한 것으로 간주되는 M2M-Request-ID가 각각의 메시지에 포함되므로 둘 다 제외될 수 있다. Auth_Tag를 계산하기 위해 사용되는 전체 메시지를 사용하는 것이 바람직하며, 대안적으로, Auth_Tag를 생성하기 위해 메시지 헤더가 사용될 수 있다. 그러나, 메시지 내의 특정 컴포넌트들이 CSE2(702)와 같은 중간 엔티티들에 의해 변경될 수 있는 경우, 메시지의 진정성 및 의도를 보증하기 위해 사용될 수 있는 메시지의 컴포넌트들만이 사용될 수 있다. to 필드(to field)인 "to" 필드 및 세션 식별자 "M2M-Request-ID"와 상이한 경우, 무결성 보호되어야 하는 메시지의 절대적인 필수 컴포넌트들은, from 필드(from field)인 "fr", to 필드(to field)인 "to", 동작 필드(operation field)인 "op", 리소스 ID(resource id)인 "res-id"이다. 메시지에 "데이터"가 포함되어 있으면, 또한 무결성 보호될 수도 있다. 전술한 바와 같이, 바람직한 접근 방식은 전체 메시지를 무결성 보호하는 것이지만, 특정 구현들에서, 컴포넌트들 중 일부는 라우팅 목적들을 위해 중간 엔티티들에 의해 합법적으로 변경될 수 있으며, 그러한 경우들에서는, 컴포넌트들이 중간 엔티티들에 의해 변경되지 않는 동시에 AE2의 요청의 진정성뿐만 아니라 무결성을 제공할 수 있다는 것이 보장되어야 한다.

[0242] 도 17의 단계 3에서, AE2는 Auth_Tag를 생성하는 데 사용된 보안 어트리뷰트들과 함께 Auth_Tag를 운반하기 위해, oneM2M 메시징을 위해 수정될 수 있는 JSON-기반 표현인 JSON 웹 서명을 생성한다. JWS에는 다음과 같은 보안 어트리뷰트들이 포함된다: credential-ID로서, 크리덴셜 또는 키를 식별하는 데 사용되는 "cred-id"(이 경우, Ke2e_AE2_CSE1_msg_auth-ID임), Auth_Tag를 계산하는 데 사용되는 알고리즘 "alg"("HMAC-SHA-256"), 데이터와 함께 메시지 또는 메시지 헤더를 포함하는 페이로드인 "페이로드", 및 Auth_Tag/MAC인 서명 "sig". 대안적으로, Base64 대신에 CBOR(Concise Binary Object Representation) 기반 표현이 CBOR 객체 서명 및 암호화 표준들에서 설명된 메커니즘에 의해 사용될 수 있다. oneM2M 메시지는 "Request" 메시지이다.

[0243] 도 17의 단계 4에서, 기존의 (D)TLS 접속이 CSE1(604)과 CSE2(702) 사이에 존재하지 않으면, Kpsa_CSE2_CSE1을 대칭 키로서 사용하는 oneM2M 사양들에 따라, CSE1(604)과 CSE2(702) 사이에 (D)TLS 접속이 확립된다.

[0244] 도 17의 단계 5에서, AE2에 의해 생성된 메시지는 CSE1(604)에 포워딩된다. 서명에 사용된 알고리즘이 공개 키 기반 메커니즘이었으면, CSE2(702)는 메시지를 CSE1(604)에 포워딩하기 전에 메시지를 인증할 수 있지만, 여기서는 대칭 키들이 사용되기 때문에, 메시지의 인증은 AE2와 CSE2(702) 사이에 존재하는 신뢰에 기초하여 암시되고, (D)TLS 접속에 기초하여 확립된 보안 연관에 기초하여, 메시지는 신뢰할 수 있는 AE2로부터 도달되는 것으로 예상된다. CSE2(702)는 메인 메시지 헤더를 수정하지 않고 메시지를 CSE1(604)에 포워딩한다. 메시지 헤더가 CSE2(702)에 의해 변경되는 경우들에서는, CSE2(702)가 메시지 헤더의 복사본을 만들고, 메시지 헤더는 Sec-Attributes과 함께 데이터의 일부로서 포함된다. AE2가 Sec-Attributes(JWS)을 생성하기 위해 전체 메시지를 사용한 경우, CSE2(702)는 전체 메시지를 CSE1(604)에 포워딩하기 전에 헤더 및 Sec-Attributes(JWS)과 함께 전체 메시지를 데이터 페이로드 부분에 복사한다. 이 메시지는 단계 4에서 셋업된 보안 (D)TLS 접속을 통해 CSE2(702)에 의해 CSE1(604)에 전송된다.

[0245] 도 17의 단계 6에서, CSE1(604)은 그것이 메시지의 타겟인지를 검증한다. Sec-Attributes(JWS)을 사용하여, 올바른 크리덴셜을 식별하고 보안 키 스토어(예를 들어, SIM 카드와 같은 보안 엘리먼트)로부터 이것을 페치하고 적절한 컨텍스트 정보 및 용도 파라미터들을 결정하기 위해 Credential-ID(cred-id)를 사용한다. 보안의 유형(서명)을 결정하는 컨텍스트 정보, 관련된 엔티티들 등 및 용도(알고리즘들, 난스의 가용성 등)에 기초하여

메시지가 올바른 특성 세트를 갖고 있는지를 검증하고, AE2(1102)에 의해 원래 전송된 전체 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터일 수 있는 메시지와 함께 Ke2e_AE2_CSE1_msg_auth 키를 사용하고, 이 경우 HMAC-SHA-256일 수 있는 JWS 내에서 식별된 "alg"에 대한 입력으로서 파라미터들을 제공하는 컨텍스트 정보와 함께 제시될 수 있는 난스를 사용하고, Generated_Auth_Tag를 생성한다. CSE1(604)은 Generated_Auth_Tag가 JWS 내에 포함된 Auth_Tag와 동일한지를 검증하고, 동일한 경우, AE2의 메시지가 인증되었다. 그 후, CSE1(604)은 AE2(1102)가 AE1 리소스에 대해 "Update" 동작을 수행하도록 인가되었는지를 보기 위한 체크를 한다.

[0246] 도 17의 단계 7에서, AE2(1102)가 "Update" 동작을 수행하도록 인가되면, CSE1(604)은 R-ID에 의해 식별된 AE1 리소스를 업데이트한다. CSE1(604)은 응답 메시지를 생성하고, 상이한 Auth_Tag2를 생성하기 위해 도 17의 단계 2에서 AE2에 의해 사용된 프로시져와 유사한 프로세스를 사용한다. 매번 새로운 Nonce를 사용하고, JWS의 일부로서 이것을 포함시키고, 기존의 Nonce를 다시 사용하지 않는 것이 권고된다. JWS2를 생성하기 위해 모든 Sec-Attributes(예를 들어, Nonce, Auth-Tag2, Credential-ID, 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터)이 포함된다.

[0247] 도 17의 단계 8에서, AE1(602)과 CSE1(604) 사이에 기존의 (D)TLS 접속이 존재하지 않으면, oneM2M 기술 사양들의 TS-0003 릴리스 1에 기초하여 공유된 대칭 키(Kpsa_AE1_CSE1)를 사용함으로써 새로운 것이 생성된다.

[0248] 도 17의 단계 9에서, CSE1(604)은 AE1의 리소스 "R-ID"에 대한 "Update"를 지시하는 "Notify" 메시지를 AE1(602)에 전송한다. 이 메시지는 도 17의 단계 8에서 셋업된 보안 (D)TLS 접속을 통해 전송된다.

[0249] 도 17의 단계 10에서는, 도 17의 단계 7에서 기술된 바와 같이 생성된 응답 메시지를 생성한 후에, CSE1(604)이 단계 4에서 확립된 보안 (D)TLS 접속을 통해 메시지를 CSE2(702)에 전송한다. 그러한 접속이 존재하지 않으면, 도 17의 단계 4에서 생성된 것과 유사하게, 새로운 (D)TLS 접속이 생성되어야 할 수 있다. 메시지 10은 도 17의 단계 8과 병렬로 전송될 수 있지만, 특정 중요한 경우들에서는, 단계 8이 도 17의 단계 10보다 먼저 수행된다.

[0250] 도 17의 단계 11에서, 공개 키 메커니즘들이 JWS를 생성하는 데 사용된 경우, JWS 내의 디지털 서명을 확인함으로써, CSE2(702)는 CSE1(604)으로부터 수신된 메시지를 진정성/무결성에 대해 검증할 수 있다. 대칭 키잉이 사용되었기 때문에, CSE2(702)는 메시지가 보안 (D)TLS 접속을 통해 수신된 것으로 인해 암시되는 신뢰를 사용하고, 도 17의 단계 1에서 셋업되었던 보안 (D)TLS 접속을 통해 메시지를 AE2(1102)에 포워딩한다. 위에서 언급된 바와 같이, 유효한 (D)TLS 접속이 존재하지 않는 경우, Kpsa_AE2_CSE2 대칭 키를 사용하여, 또한 oneM2M 기술 사양의 TS-0003 릴리스 1에 기술된 메커니즘들을 사용하여 CSE2(702)와 AE2(1102) 사이에 새로운 (D)TLS 접속이 확립되어야 한다.

[0251] 도 17의 단계 12에서, AE2(1102)는 JWS 내의 Auth_Tag2를 검증하고, 도 17의 단계 6에서 설명된 것과 유사한 메커니즘들을 사용하여 메시지를 인증한다. 사용되는 보안 어트리뷰트들은 단계 6에서의 보안 어트리뷰트들과 상이할 수 있지만, 프로세스는 동일하다.

[0252] 도 17에 도시된 단계들을 수행하는 엔티티들은 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 17에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 17에 도시된 단계들을 수행한다. 또한, 도 17에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0253] 도 18은 신뢰되거나 또는 신뢰가 떨어지거나 또는 심지어 신뢰할 수 없는 중간 홉들을 횡단하여, 복수의 서비스-계층 홉들만큼 서로 떨어져 있는 2개의 엔티티들(AE2(1102) 및 CSE1(604)) 사이의 대칭 키 메커니즘에 의한 엔드 투 엔드 메시지 인증 및 무결성 체크, 및 또한 메시지 기밀성을 나타내는 실시예를 도시한다. 클라이언트 애플리케이션(AE2(1102))은 다른 애플리케이션 엔티티(AE1(602))의 리소스에 대한 업데이트 동작을 수행하고자 한다. 리소스는 호스팅 CSE(CSE1(604)) 상에서 호스팅되기 때문에, AE2(1102)는 리소스의 위치로 사전에 프로비저닝되었거나 또는 리소스 위치(/CSE1/R-ID)를 발견하기 위해 발견 서비스를 사용한다. CSE1(604)은 인가된 엔티티들만이 AE1 리소스에 대해 Create, Retrieve, Update, Delete 또는 Notify 동작들 중 임의의 동작을 수

행할 수 있도록 보장하고자 한다. CSE1(604)이 인가된 엔티티들만이 CRUD 동작들을 수행할 수 있도록 보장할 수 있게 하기 위해, CSE1(604)은 메시지들의 소스가 인증되고 메시지들이 메시지의 발신자에 의한 키의 소유를 검증함으로써 무결성 보호되도록 요구할 수 있다. 또한, 데이터 및 메시징에는 기밀성 보호될 것이 요구된다. 메시지의 생성을 준비함에 있어서, AE2(1102)는 적절한 메시지 인증 및 메시지 기밀성 키인 Ke2e_AE2_CSE1_msg_auth 및 Ke2e_AE2_CSE1_msg_conf를 각각 TTP로부터 획득하거나, 또는 그것에 프로비저닝되었거나 또는 전술한 "제3자 크리덴셜 요구 프로세스"를 사용하는 부트스트래핑 프로세스를 사용하여 생성된 엔드 투 엔드 마스터 키인 Ke2e_AE2_CSE1_master로부터 키들을 생성해야 한다. 대안적으로, 하나의 Ke2e_AE2_CSE1_msg_auth_conf가 AEAD(Authenticated Encryption and Associated Data) 기반 암호화 메커니즘(예를 들어, AES-CCM, AES-GCM)을 사용하여, 메시지 인증뿐만 아니라 메시지 기밀성에 사용될 수 있다. 키 외에도, AE2의 메시지의 진정성 및 무결성을 검증하기 위해 AE2(1102)가 CSE1(604)에 의해 사용될 수 있는 올바른 인증 태그를 생성할 수 있도록 하기 위해 요구되는 컨텍스트 정보, 용도 정보 및 레이블이 AE2(1102)에 대해 획득, 생성 또는 프로비저닝된다. 기밀성에 대해 적절한 알고리즘을 결정하기 위해, 동작되는 모드뿐만 아니라 초기화 벡터(Intilization Vector)(IV)의 요건 등도 포함한다. AE2(1102)는 엔드 투 엔드 메시지 인증 및 메시지 기밀성을 수행하기 위해 키 스토어로부터 적절한 크리덴셜들을 선택하고, 그에 따라 Ke2e_AE2_CSE1_msg_auth_conf 키가 선택된다.

[0254] 도 18의 단계 1에서, AE2(1102)는 자신과 CSE2(702) 간의 (D)TLS 접속을 셋업하기 위해 Kpsa_AE2_CSE2를 사용한다. 접속 확립 프로세스는 oneM2M 사양들의 TS-0003 릴리스 1에 기술된 메커니즘들을 따른다.

[0255] 도 18의 단계 2에서, AE1(602)은 CSE1(604) 상에서 호스팅되는 AE1의 리소스(/CSE/R-ID로서 식별됨)에 대해 "Update" 동작을 수행하는 데 사용되는 oneM2M "Request" 메시지를 생성한다. Request 메시지는 M2M-Request-ID1에 의해 고유하게 식별된다. AE2(1102)는 OriginData라고도 하는 메시지 헤더 정보를 사용하여 인증 태그(Auth_Tag) 또는 메시지 인증 코드(MAC)를 생성한다. 대안적으로, Auth_Tag를 생성하기 위해 전체 메시지가 입력으로서 사용된다. 전술한 바와 같이, 선호되는 접근 방식은 전체 메시지를 무결성 보호하는 것이지만, 특정 구현들에서, 컴포넌트들 중 일부의 컴포넌트들은 라우팅 목적들을 위해 중간 엔티티들에 의해 합법적으로 변경될 수 있으며, 그러한 경우들에서는, 컴포넌트들이 중간 엔티티들에 의해 변경되지 않고 동시에 AE2의 요청의 진정성뿐만 아니라 무결성을 제공할 수도 있다는 것이 보장되어야 한다. oneM2M 계층 라우팅에 사용되어야 하는 메시지 또는 메시지 헤더 또는 메타 데이터는 암호화되지 않고, 어소시에이트 데이터(Associate Data)(AAD)로서 분류된다. AAD는 무결성 보호될 수 있다. 메시지 헤더 또는 메타 데이터는 "AAD" 값이 할당되기에 적절한 후보들이다.

[0256] Auth_Tag = HMAC-SHA-256 (Ke2e_AE2_CSE1_msg_auth_conf, AAD |

[0257] Nonce | Time)

[0258] AAD는 전체 메시지 헤더에 할당될 수 있거나, 또는 대안적으로, AAD는 메시지 헤더의 서브 세트 또는 메시지의 메타 데이터로서 할당될 수 있다.

[0259] Nonce, Time 또는 둘 모두가 Auth_Tag의 생성에 포함될 수 있다. 특정 경우들에서는, 각각의 세션마다 고유한 것으로 간주되는 M2M-Request-ID가 각각의 메시지에 포함되므로 둘 다 제외될 수 있다. Auth_Tag를 계산하기 위해 사용되는 전체 메시지를 사용하는 것이 바람직하며, 대안적으로, Auth_Tag를 생성하기 위해 메시지 헤더가 사용될 수 있다. 그러나, 메시지 내의 특정 컴포넌트들이 CSE2(702)와 같은 중간 엔티티들에 의해 변경될 수 있는 경우, 메시지의 진정성 및 의도를 보증하기 위해 사용될 수 있는 메시지의 컴포넌트들만이 사용될 수 있다. to 필드인 "to" 필드 및 세션 식별자 "M2M-Request-ID"와 상이한 경우, 무결성 보호되어야 하는 메시지의 절대적인 필수 컴포넌트들은, from 필드인 "fr", to 필드인 "to", 동작 필드인 "op", 리소스 ID인 "res-id"이다. 데이터 페이로드를 포함하는 나머지 메시지는 ContextInfo 및 Usage 파라미터들(예를 들어, 암호화 알고리즘, 암호화 모드 및 IV ..)에 따라 암호화될 수 있다.

[0260] 단계 3에서, AE2(1102)는 Auth_Tag를 생성하는 데 사용된 보안 어트리뷰트들뿐만 아니라 암호화된 메시지 및 데이터와 함께 Auth_Tag를 운반하기 위해, oneM2M 메시징을 위해 수정되고 조정될 수 있는 JSON-기반 표현인 JSON 웹 암호화 표현(JWE)을 생성한다. JWE에는 다음과 같은 보안 어트리뷰트들이 포함된다: Credential-ID로서, 크리덴셜 또는 키를 식별하는 데 사용되는 "cred-id"(이 경우, Ke2e_AE2_CSE1_msg_auth-ID임). 대안적으로, 별도의 메시지 인증 키뿐만 아니라 별도의 메시지 기밀성 키들이 사용되면, 연관된 Credential-ID들이 모두 전송되어야 한다. 사용된 알고리즘 "alg"인 "AES-CCM"(예시), 데이터와 함께 메시지 또는 메시지 헤더를 포함하는

페이로드인 "payload", 및 Auth_Tag/MAC인 서명 "sig"가 포함된다. 또한, 사용된 초기화 벡터인 "iv", 및 암호화 프로세스에 기초하여 생성된 암호문인 "ciphertext"도 JWE의 일부로서 포함된다. 대안적으로, Base64 대신에 CBOR(Concise Binary Object Representation)-기반 표현이 CBOR 객체 서명 및 암호화 표준들에 설명된 메커니즘에 의해 사용될 수 있다. 메시지 헤더뿐만 아니라 JWE로서 표현되는 Sec-Attributes을 포함하는 oneM2M 메시지 "Request"가 생성된다.

[0261] 도 18의 단계 4에서, CSE1(604)과 CSE2(702) 사이에 기존의 (D)TLS 접속이 존재하지 않으면, Kpsa_CSE2_CSE1을 대칭 키로서 사용하는 oneM2M 사양들에 따라 CSE1(604)과 CSE2(702) 사이에 (D)TLS 접속이 확립된다.

[0262] 도 18의 단계 5에서, AE2(1102)에 의해 생성된 메시지는 CSE1(604)에 포워딩된다. 서명에 사용된 알고리즘이 공개 키 기반 메커니즘이면, CSE2(702)는 CSE1(604)에 메시지를 포워딩하기 전에 메시지를 인증할 수 있지만, 여기서는 대칭 키들이 사용되기 때문에, 메시지의 인증은 AE2(1102)와 CSE2(702) 사이에 존재하는 신뢰에 기초하여 암시되고, (D)TLS 접속에 기초하여 확립된 보안 연관에 기초하여, 메시지는 신뢰할 수 있는 AE2(1102)로부터 도달되는 것으로 예상된다. CSE2(702)는 메인 메시지 헤더를 수정하지 않고 메시지를 CSE1(604)에 포워딩한다. 메시지 헤더가 CSE2(702)에 의해 변경되는 경우들에는, CSE2(702)는 메시지 헤더의 복사본을 만들고, 메시지 헤더가 Sec-Attributes과 함께 데이터의 일부로서 포함된다. AE2(1102)가 Sec-Attributes(JWE)을 생성하기 위해 전체 메시지를 사용한 경우, CSE2(702)는 전체 메시지를 CSE1(604)에 포워딩하기 전에 헤더 및 보안-어트리뷰트들(JWE)과 함께 전체 메시지를 데이터 페이로드 부분에 복사하는데, 이는 Auth_Tag1이 수신자(예를 들어, CSE1)에 의해 적절히 구성될 수 있도록 모든 필요한 메시지 헤더 정보가 보존되도록 하기 위함이다. 이 메시지는 도 18의 단계 4에서 셋업된 보안 (D)TLS 접속을 통해 CSE2(702)에 의해 CSE1에 전송된다.

[0263] 도 18의 단계 6에서, CSE1(604)은 그것이 메시지의 타겟인지를 검증한다. Sec-Attributes(JWE)를 사용하여, 올바른 크리덴셜(들)을 식별하고 보안 키 스토어(예를 들어, SIM 카드와 같은 보안 엘리먼트)로부터 이들을 페치하고, 적절한 컨텍스트 정보 및 용도 파라미터들을 결정하기 위해, CSE1(604)은 Credential-ID(들)를 사용한다. 메시지 인증뿐만 아니라 메시지 기밀성을 위해 별도의 키들이 사용되는 경우, 양 키들은 키 스토어로부터 페치되어야 할 것이다. JWE 정보 "alg"뿐만 아니라 "cred-id"를 사용하여, CSE1(604)은 AEAD가 보안 보호에 사용되는지를 결정할 수 있고, 그렇다면, cred-id에 의해 식별된 하나의 연관된 크리덴셜만이 리트리브될 수 있다. 보안의 유형(서명, 암호화)을 결정하는 컨텍스트 정보, 관련 엔티티들 등 및 용도(알고리즘들, 난스의 가용성 등)에 기초하여 메시지가 올바른 특성들의 세트를 갖고 있는지를 검증하고, AAD, IV, 난스들 및 기타 파라미터들을 식별하고, "ciphertext"을 복호하고 메시지뿐만 아니라 데이터 페이로드를 포함할 수 있는 "plaintext"을 추출하기 위해 Ke2e_AE2_CSE1_msg_auth_conf 키를 사용한다. CSE1(604)은 Generated_Auth_Tag를 계산하기 위해 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터를 사용하거나, 또는 AAD로서 식별된 정보를 사용한다. 일부 경우들에서는, AE2(1102)에 의해 원래 전송된 전체 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터를 사용하고, 이 경우 AES-CCM일 수 있는 JWE 내에서 식별된 "alg"에 대한 입력으로서 파라미터들을 제공하는 컨텍스트 정보와 함께 제시될 수 있는 난스를 사용하고, Generated_Auth_Tag를 생성한다. CSE1(604)은 Generated_Auth_Tag가 JWE 내에 포함된 Auth_Tag와 동일한지를 검증하고, 동일한 경우, AE2의 메시지가 인증되었다. 그 후, CSE1(604)은 AE2(1102)가 AE1(602)의 리소스에 대해 "Update" 동작을 수행하도록 인가되었는지를 보기 위한 체크를 한다.

[0264] 도 18의 단계 7에서, AE2(1102)가 "Update" 동작을 수행하도록 인가되면, CSE1(604)은 R-ID에 의해 식별된 AE1 리소스를 업데이트한다.

[0265] 단계 8v에서, AE1(602)과 CSE1(604) 사이에 기존의 (D)TLS 접속이 존재하지 않으면, oneM2M 기술 사양 TS-0003의 릴리스 1에 기초하여 공유된 대칭 키 Kpsa_AE1_CSE1을 사용함으로써 새로운 것이 생성된다.

[0266] 도 18의 단계 9에서, CSE1(604)은 AE1의 리소스 "R-ID"에 대한 "Update"를 지시하는 "Notify" 메시지를 AE1(602)에 전송한다. 이 메시지는 단계 8에서 셋업된 보안 (D)TLS 접속을 통해 전송된다.

[0267] 도 18의 단계 10에서, CSE1(604)은 응답 메시지를 생성하고, 상이한 Auth_Tag2, 암호화된 메시지 및 JWE를 생성하기 위해 도 18의 단계 2에서 AE2(1102)에 의해 사용된 프로시져와 유사한 프로세스를 사용한다. 매번 새로운 Nonce와 IV를 사용하고, 이것을 JWE의 일부로서 포함시키고, 기존의 Nonce를 재사용하지 않는 것이 권고된다. JWE2를 생성하기 위해 모든 Sec-Attributes(예를 들어, Nonce, Auth-Tag2, Credential-ID, AAD로서 식별되는 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터, IV 및 암호문)이 포함될 수 있다.

[0268] 도 18의 단계 11에서, CSE1(604)은 단계 4에서 확립된 보안 (D)TLS 접속을 통해 메시지를 CSE2(702)에

전송한다. 그러한 접속이 존재하지 않으면, 도 18의 단계 4에서 생성된 것과 유사한 새로운 (D)TLS 접속이 생성되어야 할 수 있다. 메시지 10은 단계 8과 병렬로 전송될 수 있지만, 특정 중요한 경우들에서는, 도 18의 단계 8은 도 18의 단계 10보다 먼저 수행된다.

[0269] 도 18의 단계 12에서, 공개 키 메커니즘들이 JWE를 생성하기 위해 사용된 경우, CSE2(702)는 JWS 내의 디지털 서명을 확인함으로써, CSE1(604)으로부터 수신된 메시지를 진위성/무결성에 대해 검증할 수 있다. 여기서는 대칭 키잉이 사용되었기 때문에, CSE2(702)는 메시지가 보안 (D)TLS 접속을 통해 수신된 것으로 인해 암시되는 신뢰를 사용하고, 단계 1에서 셋업된 보안 (D)TLS 접속을 통해 메시지를 AE2(1102)에 포워딩한다. 전술한 바와 같이, 유효한 (D)TLS 접속이 존재하지 않으면, Kpsa_AE2_CSE2(702) 대칭 키를 사용하여, 또한 oneM2M 기술 사양 TS-0003 릴리스 1에 기술된 메커니즘들을 사용하여 CSE2(702)와 AE2(1102) 사이에 새로운 (D)TLS 접속이 확립되어야 한다.

[0270] 도 18의 단계 13에서, AE2(1102)는 도 18의 단계 6에서 설명된 것과 유사한 메커니즘들을 사용하여 메시지를 복호한 후에 JWE 내의 Auth_Tag2를 검증한다. 사용된 보안 어트리뷰트들은 도 18의 단계 6의 것들과 상이할 것이만, 프로세스는 동일하다.

[0271] 도 19는 신뢰되거나 또는 신뢰가 떨어지거나 또는 심지어 신뢰할 수 없는 중간 홉들을 횡단하여, 복수의 서비스 계층 홉들만큼 서로 떨어져 있는 2개의 엔티티들(AE2(1102) 및 CSE1(604)) 사이의 대칭 키 메커니즘에 의한 엔드 투 엔드 메시지 인증 및 무결성 체크 및 또한 메시지 기밀성을 나타내는 실시예를 도시한다. 클라이언트 애플리케이션(AE2(1102))은 다른 애플리케이션 엔티티(AE1(602))의 리소스에 대한 업데이트 동작을 수행하고자 한다. 리소스는 호스팅 CSE(CSE1(604)) 상에서 호스팅되기 때문에, AE2(1102)는 리소스의 위치로 사전에 프로비저닝되었거나, 또는 리소스의 위치(/CSE1/R-ID)를 발견하기 위해 발견 서비스를 사용한다. CSE1(604)은 인가된 엔티티들만이 AE1(602)의 리소스에 대해 Create, Retrieve, Update, Delete 또는 Notify 동작들 중 임의의 것을 수행할 수 있음을 보장하고자 한다. CSE1(604)가 인가된 엔티티들만이 CRUD 동작들을 수행할 수 있도록 보장할 수 있도록 하기 위해, CSE1(604)은 메시지의 소스가 인증되고 메시지들이 메시지의 발신자에 의한 키의 소유를 검증함으로써 무결성 보호되도록 요구할 수 있다. 또한, 데이터 및 메시징은 기밀성 보호될 것이 요구된다. 메시지의 생성을 준비함에 있어서, AE2(1102)는 적절한 메시지 인증 및 메시지 기밀성 키인 Ke2e_AE2_CSE1_msg_auth 및 Ke2e_AE2_CSE1_msg_conf를 각각 TTP로부터 획득하거나, 또는 그것에 프로비저닝되었거나 또는 전술한 "제3자 크리덴셜 요구 프로세스"를 사용하는 부트스트래핑 프로세스를 사용하여 생성된 엔드 투 엔드 마스터 키인 Ke2e_AE2_CSE1_master로부터 키들을 생성해야 한다. 대안적으로, 하나의 Ke2e_AE2_CSE1_msg_auth_conf가 AEAD(Authenticated Encryption and Associated Data) 기반 암호화 메커니즘(예를 들어, AES-CCM, AES-GCM)을 사용하여 메시지 인증뿐만 아니라 메시지 기밀성을 위해 사용될 수 있다. 키 외에도, AE2의 메시지의 진정성 및 무결성을 검증하기 위해 AE2(1102)가 CSE1(604)에 의해 사용될 수 있는 올바른 인증 태그를 생성할 수 있도록 하기 위해 요구되는 컨텍스트 정보, 용도 정보 및 레이블이 AE2(1102)에 대해 획득, 생성 또는 프로비저닝된다. 기밀성에 대해 적절한 알고리즘을 결정하기 위해, 동작되는 모드뿐만 아니라 초기화 벡터(Intilization Vector)(IV)의 요건 등도 포함한다. AE2(1102)는 엔드 투 엔드 메시지 인증 및 메시지 기밀성을 수행하기 위해 키 스토어로부터 적절한 크리덴셜들을 선택하고, 그에 따라 Ke2e_AE2_CSE1_msg_auth_conf 키가 선택된다.

[0272] 전술한 시나리오와는 달리, AE2(1102)와 CSE2(702) 사이에 (D)TLS 기반 보안 접속 확립을 수행하기 위한 키가 존재하지 않고, 대신에 객체 기반 보안 모델을 사용하여 서비스 계층에서 AE2(1102)와 CSE2(702) 사이에서 메시지 인증을 제공하는 데에 이용 가능한 크리덴셜이 사용된다. Ke2e_AE2_CSE2_msg_auth는 엔드 투 엔드 키 또는 홉 바이 홉 크리덴셜로 지칭될 수 있으며(어느 쪽이든 문제가 되지 않음), 크리덴셜의 용도와 컨텍스트가 중요하다. 용도 및 컨텍스트 정보는 크리덴셜이 사용되는 방법에 대한 지침을 제공한다. 용도 및 컨텍스트 정보는 제3자 크리덴셜 요구 프로세스 동안에 TTP로부터 획득되거나 프로비저닝될 수 있다. TTP는 엔티티 등록 프로세스 동안에 서비스 제공자 또는 엔티티로부터 차례로 획득된 SP, DP 및/또는 EP에 기초하여, 적절한 용도 및 컨텍스트 정보 및 연관된 보안 요건들 및 피쳐들을 획득하거나 추론할 수 있다. 메커니즘은 M2M-Subscription-Profile 내에 포함된 참조 링크들을 사용함으로써 IN-CSE(2002)로부터 SP, DP 및/또는 EP를 획득하는 것이다.

[0273] 도 18에 도시된 단계들을 수행하는 엔티티들은, 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 18에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해

실행될 때, 도 18에 도시된 단계들을 수행한다. 또한, 도 18에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0274] 도 19의 단계 1에서, AE2(1102)는 홉 바이 홉 인증 및 보안 통신 확립 메커니즘들을 사용하지 않는다. AE1(602)은 CSE1(604) 상에서 호스팅되는 AE1의 리소스(/CSE/R-ID로서 식별됨)에 대해 "Update" 동작을 수행하는 데 사용되는 oneM2M "Request" 메시지를 생성한다. Request 메시지는 M2M-Request-ID1에 의해 고유하게 식별된다. AE2(1102)는 OriginData라고도 하는 메시지 헤더 정보를 사용하여 인증 태그(Auth_Tag) 또는 메시지 인증 코드(MAC)를 생성한다. 대안적으로, Auth_Tag를 생성하기 위해 전체 메시지가 입력으로서 사용된다. 전술한 바와 같이, 선호되는 접근 방식은 전체 메시지를 무결성 보호하는 것이지만, 특정 구현들에서, 컴포넌트들 중 일부의 컴포넌트들은 라우팅 목적들을 위해 중간 엔티티들에 의해 합법적으로 변경될 수 있으며, 그러한 경우들에서는, 컴포넌트들이 중간 엔티티들에 의해 변경되지 않고 동시에 AE2의 요청의 진정성 및 무결성을 제공할 수 있다는 것이 보장되어야 한다. oneM2M 계층 라우팅에 사용되어야 하는 메시지 또는 메시지 헤더 또는 메타 데이터는 암호화되지 않고 어소시에이트 데이터(AAD)로서 분류된다. AAD는 무결성 보호될 수 있다. 메시지 헤더 또는 메타 데이터는 "AAD" 값이 할당되기에 적절한 후보들이다.

[0275] Auth_Tag = HMAC-SHA-256 (Ke2e_AE2_CSE1_msg_auth_conf, AAD |

[0276] Nonce | Time)

[0277] AAD는 전체 메시지 헤더에 할당될 수 있거나, 또는 대안적으로, AAD는 메시지 헤더의 서브 세트 또는 메시지의 메타 데이터로서 할당될 수 있다.

[0278] Nonce, Time 또는 둘 모두가 Auth_Tag의 생성에 포함될 수 있다. 특정 경우들에는, 각각의 세션마다 고유한 것으로 간주되는 M2M-Request-ID가 각각의 메시지에 포함되므로 둘 다 제외될 수 있다. Auth_Tag를 계산하기 위해 사용되는 전체 메시지를 사용하는 것이 바람직하며, 대안적으로, Auth_Tag를 생성하기 위해 메시지 헤더가 사용될 수 있다. 그러나, 메시지 내의 특정 컴포넌트들이 CSE2(702)와 같은 중간 엔티티들에 의해 변경될 수 있는 경우, 메시지의 진정성 및 의도를 보증하기 위해 사용될 수 있는 메시지의 컴포넌트들만이 사용될 수 있다. to 필드인 "to" 필드 및 세션 식별자 "M2M-Request-ID"와 상이한 경우, 무결성 보호되어야 하는 메시지의 절대적인 필수 컴포넌트들은, from 필드인 "fr", to 필드인 "to", 동작 필드인 "op", 리소스 ID인 "res-id"이다. 데이터 페이로드를 포함하는 나머지 메시지는 ContextInfo 및 Usage 파라미터들(예를 들어, 암호화 알고리즘, 암호화 모드 및 IV ..)에 따라 암호화될 수 있다.

[0279] 도 19의 단계 2에서, AE2(1102)는 Auth_Tag를 생성하는 데 사용된 보안 어트리뷰트들뿐만 아니라 암호화된 메시지 및 데이터와 함께 Auth_Tag를 운반하기 위해, oneM2M 메시징을 위해 수정 및 조정될 수 있는 JSON-기반 표현인 JSON 웹 암호화 표현(JWE)을 생성한다. JWE에는 다음과 같은 보안 어트리뷰트들이 포함된다: Credential-ID로서, 크리덴셜 또는 키를 식별하는 데 사용되는 "cred-id"(이 경우, Ke2e_AE2_CSE1_msg_auth-ID임). 대안적으로, 별도의 메시지 인증 키뿐만 아니라 별도의 메시지 기밀성 키들이 사용되면, 연관된 Credential-ID들이 모두 전송되어야 한다. 사용된 알고리즘 "alg"인 "AES-CCM"(예시), 데이터와 함께 메시지 또는 메시지 헤더를 포함하는 페이로드인 "payload", 및 Auth_Tag/MAC인 서명 "sig"가 포함된다. 또한, 사용된 초기화 벡터인 "iv", 및 암호화 프로세스에 기초하여 생성된 암호문인 "ciphertext"도 JWE의 일부로서 포함된다. 대안적으로, Base64 대신에 CBOR(Concise Binary Object Representation)-기반 표현이 CBOR 객체 서명 및 암호화 표준들에 설명된 메커니즘에 의해 사용될 수 있다. 메시지 헤더뿐만 아니라 JWE1으로서 표현되는 Sec-Attributes을 포함하는 oneM2M 메시지 "Request"가 생성된다.

[0280] 또한, AE1(602)은 Ke2e_AE2_CSE2_msg_auth를 사용하고, 새로운 Nonce를 생성하고, 내부 Sec-Attributes/JWE1 파라미터들을 포함하는 Request 메시지에 대해 Auth_Tag2를 생성한다. 외부 Auth_Tag2는 CSE2(702)와의 인증에 사용된다. AE2(1102)는 그 크리덴셜-ID인 Ke2e_AE2_CSE2_msg_auth-ID에 의해 식별가능한 연관된 크리덴셜인 Ke2e_AE2_CSE2_msg_auth가 제공된 ContextInfo 및 UsageInfo에 제공된 지침에 기초하여, Auth_Tag2(MAC)를 포함하는 JWS1을 생성한다. AE2(1102)에 의해 생성된 메시지는 CSE1(604)에 포워딩된다.

[0281] 도 19의 단계 3에서, CSE2(702)는 수신된 메시지 내에 포함된 JWS1 정보를 사용하여, UsageInfo 및 ContextInfo와 함께 키 스토어로부터 Credential-ID에 기초한 연관된 크리덴셜을 획득한다. CSE2(702)는 Nonce, Ke2e_AE2_CSE2_msg_auth 및 메시지/메시지 헤더를 사용하여 Auth_Tag를 생성하고, 이를 JWS1 내에 포함된 Auth_Tag와 비교하고, 이들이 매칭되면, AE2의 메시지가 인증되었다는 것을 암시하고, AE2(1102)가 이러한

메시지를 전송하도록 인가되었으면, CSE2(702)는 Request 메시지를 프로세싱한다. CSE2(702)는 메시지로부터 외부 JWS1/MAC을 제거한다.

[0282] 도 19의 단계 4에서, CSE2(702)는 JWS2 또는 MAC을 생성하고, 그것을 Request 메시지에 첨부한다. JWS2 내의 Auth_Tag는, 새로 생성된 난스, 메시지 또는 메시지 헤더 또는 메타 데이터와 함께 Ke2e_CSE2_CSE1_msg_auth 키를 사용하여, 크리덴셜/키와 연관된 ContextInfo 및 UsageInfo에 기초하여 생성된다. CSE2(702)는 Request 메시지에 JWS2/MAC를 첨부하고, 이를 CSE1(604)에 전송한다. 여전히 활성화되어 있는 (D)TLS 접속에 의해 생성된 홉 바이 홉 보안 연관이 존재하는 경우, Request 메시지는 JWS2/MAC을 생성하는 대신에 그 보안 접속을 통해 전송될 수 있다. (예를 들어, JWS를 사용하는) 객체 보안(Object Security) 대신에 (D)TLS를 사용하는 것은 서비스 제공자 정책들, 디바이스 능력들 등에 기초하여 결정될 수 있다. 메시지 기밀성이 요구되지 않은 경우들에서는, 객체 보안 사용이 바람직할 수 있다. 특정 경우들에서는, 메시지 및 데이터 기밀성이 요구될 수 있더라도, 하위 계층 보안에 의존하는 대신에 서비스 계층이 보안 서비스들을 제공할 수 있기 때문에, 또는 (D)TLS가 계산적으로 및/또는 공간적으로 보다 집중적일 수 있다는 성능상의 이유들로, 정책들이 (D)TLS 대신 JWE의 사용을 지시할 수 있다.

[0283] 도 19의 단계 5에서, CSE1(604)은 그것이 메시지의 타겟인지를 검증한다. 외부의 Sec-Attributes(JWS2/MAC)을 사용하여, 올바른 크리덴셜(들)을 식별하고, 보안 키 스토어(예를 들어, SIM 카드와 같은 보안 엘리먼트)로부터 이들을 페치하고, 적절한 컨텍스트 정보 및 용도 파라미터들을 결정하기 위해, CSE1(604)은 연관된 Credential-ID(들)를 사용한다. 이 경우, Ke2e_CSE2_CSE1_msg_auth 키가 JWS2의 Nonce와 함께 리트리브되고, 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터를 사용하여, CSE1(604)이 Generated_Auth_Tag를 생성하고, 이를 JWS2/MAC 내의 Auth_Tag와 비교하고, 이들이 매칭되면, CSE1(604)은 메시지가 신뢰된 CSE1(604)을 통해 전송되었다는 것을 인증한다.

[0284] CSE1(604)은 외부 JWS2/MAC을 폐기하고, 내부 Sec-Attributes/JWE1을 프로세싱한다. JWE1 내에서, CSE1(604)은 크리덴셜-ID(들)를 획득하고, 메시지 인증뿐만 아니라 메시지 기밀성 둘 다를 위해 별도의 키들이 사용되는 경우, Credential-ID(들)에 기초하여 키 스토어로부터 양 키들이 페치되어야 할 것이다. JWE 정보 "alg"뿐만 아니라 "cred-id"를 사용하여, CSE1(604)은 AEAD가 보안 보호에 사용되는지를 결정할 수 있고, 그렇다면, cred-id에 의해 식별된 하나의 연관된 크리덴셜만이 리트리브될 수 있다. 보안의 유형(서명, 암호화)을 결정하는 컨텍스트 정보, 관련 엔티티들 등 및 용도(알고리즘, 난스의 가용성 등)에 기초하여 메시지가 올바른 특성들의 세트를 갖고 있는지 검증하고, AAD, IV, 난스들 및 기타 파라미터들을 식별하고, "ciphertext"를 복호하고 메시지뿐만 아니라 데이터 페이로드를 포함할 수 있는 "plaintext"를 추출하기 위해 Ke2e_AE2_CSE1_msg_auth_conf 키를 사용한다. CSE1(604)은 Generated_Auth_Tag를 계산하기 위해 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터를 사용하거나, 또는 AAD로서 식별된 정보를 사용한다. 일부 경우들에서는, AE2(1102)에 의해 원래 전송된 전체 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터를 사용하고, 이 경우 AES-CCM일 수 있는 JWE 내에서 식별된 "alg"에 대한 입력으로서 파라미터들을 제공하는 컨텍스트 정보와 함께 제시될 수 있는 난스를 사용하고, Generated_Auth_Tag를 생성한다. CSE1(604)은 Generated_Auth_Tag가 JWE 내에 포함된 Auth_Tag와 동일한지를 검증하고, 동일한 경우, AE2의 메시지가 인증되었다.

[0285] 도 19의 단계 6에서, CSE1(604)은 AE2(1102)가 AE1(602)의 리소스에 대해 "Update" 동작을 수행하도록 인가되었는지를 보기 위한 체크를 한다. AE2(1102)가 "Update" 동작을 수행하도록 인가되면, CSE1(604)은 R-ID에 의해 식별된 AE1(602)의 리소스를 업데이트한다.

[0286] 도 19의 단계 7에서, CSE1(604)은 AE1의 리소스에 대해 수행된 업데이트 동작을 지시하는 "Notify" 메시지를 AE1(602)에 전송할 준비를 한다. AE1(602)과 CSE1(604) 사이에 기존의 (D)TLS 접속이 존재하지 않는 경우, 또는 홉 바이 홉 보안 연관을 수행하기 위해 사용되는 크리덴셜이 CSE1(604)과 AE1(602) 사이에 존재하지 않는 경우, 또는 정책들이 (D)TLS를 통한 홉 바이 홉 보안 연관들이 사용되지 않을 것을 지시하는 경우, JWS에 의한 객체 보안 메커니즘들이 메시지 인증을 제공하는 데 사용된다. CSE1(604)은, 새로 생성된 Nonce를 메시지, 메시지 헤더 또는 메시지의 메타 데이터와 함께 사용하여, Ke2e_CSE1_AE1_msg_auth 키와 연관된 ContextInfo 및 UsageInfo에 기초하여 JWS3/MAC을 생성한다. JWS3/MAC은 CSE1(604)에 의해 생성되고 AE1의 리소스 "R-ID"에 대한 "Update"를 지시하는 "Notify" Request 메시지에 첨부되어, AE1(602)에 전송된다. 만약 정책들이 CSE1(604)과 AE1(602) 사이의 통신을 안전하게 하기 위해 (D)TLS에 의한 홉 바이 홉 보안이 사용될 것을 지시하면, 공유된 대칭 키 Kpsa_AE1_CSE1을 사용함으로써 (D)TLS 접속들이 생성될 수 있으며, 이는 oneM2M 기술 사양들 TS-0003 릴리스 1에 기초하여 프로비저닝되거나 생성되어야 할 수 있다. "Notify" 메시지는 객체 보안 메커

니즘을 사용하는 대신 보안 접속을 통해 전송될 수 있다.

[0287] 도 19의 단계 8에서, AE1(602)은 JWS3/MAC을 검증하고, "notify" 메시지를 인증한다.

[0288] 도 19의 단계 9에서, CSE1(604)은 AE2(1102)에 의해 전송된 요청 메시지에 대한 응답인 응답 메시지를 생성한다. CSE1(604)은 AE2(1102)와 CSE1(604) 사이에 연관되는 Ke2e_AE2_CSE1_msg_auth_conf를 사용하여 Auth_Tag2, 암호화된 메시지 및 JWE2를 생성하기 위해, 단계 2에서 AE2(1102)에 의해 사용된 프로시져와 유사한 프로세스를 사용한다. Sec-Attributes/JWE2는 메시지 헤더 또는 AAD에 첨부된다. 매번 새로운 Nonce와 IV를 사용하고, 이를 JWE의 일부로서 포함시키고, 기존의 Nonce를 재사용하지 않는 것이 권고된다. JWE2를 생성하기 위해 모든 Sec-Attributes(예를 들어, Nonce, Auth-Tag2, Credential-ID, AAD로서 식별되는 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터, IV 및 암호문)이 포함될 수 있다. 임의로, CSE1(604)은 또한 CSE2(702)에 메시지 인증을 제공하기 위해 사용되는 외부 JWS4/MAC(Auth_Tag)를 생성한다. JWS4는 이전에 기술된 적절한 파라미터들과 함께 Ke2e_CSE2_CSE1_msg_auth를 사용함으로써 생성된다.

[0289] 도 19의 단계 10에서, CSE1(604)은 JWS4/MAC과 함께 응답 메시지를 CSE2(702)에 전송한다. 정책들이 CSE1(604)과 CSE2(702) 사이에 (D)TLS 접속의 셋업을 요구한다면, 응답 메시지는 보안 접속을 통해 전송될 수 있고, JWS4의 생성을 건너 뛸 수 있다.

[0290] 도 19의 단계 11에서, CSE2(702)는 JWS4를 검증함으로써 CSE1(604)으로부터 수신된 메시지를 진정성/무결성에 대해 검증할 수 있고, 메시지로부터 JWS4/MAC을 제거할 수 있다. 그 후, CSE2(702)는 상술한 바와 같이 Ke2e_AE2_CSE2_msg_auth 및 다른 파라미터들(예를 들어, 새로운 Nonce, 메시지 헤더 또는 메시지 또는 메시지의 메타 데이터, ContextInfo 및 다른 파라미터들)를 사용하여 Auth_Tag를 생성하고, Auth_Tag를 JWS5에 통합한다.

[0291] 도 19의 단계 12에서, AE2(1102)는 JWS5를 검증하고, 도 19의 단계 5에서 설명된 것과 유사한 메커니즘들을 사용하여 Response 메시지 상의 Ke2e_AE2_CSE2_msg_auth 키를 사용함으로써 Sec-Attributes/JWE2를 포함하는 Response 메시지를 인증한다. 따라서, AE2(1102)는 Response 메시지가 신뢰성있는 CSE2(702)에 의해 포워딩되었다고 결정한다.

[0292] AE2(1102)는 외부 JWS5/MAC를 폐기하고, 내부 Sec-Attributes/JWE2를 프로세싱한다. JWE2 내에서, AE2(1102)는 크리덴셜-ID(들)를 획득하는데, 메시지 인증뿐만 아니라 메시지 기밀성 둘 다를 위해 별도의 키들이 사용되는 경우, Credential-ID(들)에 기초하여 키 스토어로부터 양 키들이 모두 페치되어야 할 것이다. JWE 정보 "alg"뿐만 아니라 "cred-id"를 사용하여, AE2(1102)는 AEAD가 보안 보호에 사용되는지를 결정할 수 있고, 그렇다면, cred-id에 의해 식별된 하나의 연관된 크리덴셜만이 리트리브될 수 있다. 보안의 유형(서명, 암호화)을 결정하는 컨텍스트 정보, 관련 엔티티들 등 및 용도(알고리즘, 난스의 가용성 등)에 기초하여 메시지가 올바른 특성들의 세트를 갖고 있는지 검증하고, AAD, IV, 난스들 및 기타 파라미터들을 식별하고, "ciphertext"를 복호하고 메시지뿐만 아니라 데이터 페이로드를 포함할 수 있는 "plaintext"를 추출하기 위해 Ke2e_AE2_CSE1_msg_auth_conf 키를 사용한다. AE2(1102)는 Generated_Auth_Tag를 계산하기 위해 메시지 또는 메시지 헤더 또는 메시지의 메타 데이터를 사용하거나 또는 AAD로서 식별된 정보를 사용한다. 일부 경우들에는, CSE1(604)에 의해 원래 전송된 전체 메시지, 또는 메시지 헤더 또는 메시지의 메타 데이터가 사용되며, 이 경우 AES-CCM일 수 있는 JWE 내에서 식별된 "alg"에 대한 입력으로서 파라미터들을 제공하는 컨텍스트 정보와 함께 제시될 수 있는 난스를 사용하고, Generated_Auth_Tag를 생성한다. AE2(1102)는 Generated_Auth_Tag가 JWE 내에 포함된 Auth_Tag와 동일한지를 검증하고, 그렇다면, CSE1의 메시지가 인증되었다.

[0293] 도 19에 도시된 단계들을 수행하는 엔티티들은, 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 19에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 19에 도시된 단계들을 수행한다. 도 19에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0294] **위임 모드를 사용하는 세션 계층에서의 E2E 인증**

[0295] 도 9는 위임 모드를 사용하여 세션 계층(DTLS/TLS)에서 수행되는 E2E 인증을 도시한다. 여기서는 인증이 두 개의 E2E 엔티티들(CSE1(604) 및 CSE3(502)) 간에 별도의 세션 계층 접속을 설정함으로써 수행된다는 것을 제외하고는, 위에서 사용된 접근 방식과 유사하다. CSE1(604) 및 CSE3(502)는 홉 바이 홉 인증을 수행하는 대신에 DTLS 또는 TLS 기반 인증을 수행하고, Request 메시지들 내에서 E2E 인증 MAC이 운반된다. 메시지 세부 정보는 다음과 같다.

[0296] 도 9의 단계 1 내지 단계 4는 도 8a 및 도 8b와 관련한 메시징 메커니즘들과 유사하다.

[0297] 도 9의 단계 5에서, CSE1(604)은 KpsaE2E를 사용하여 CSE3(502)과 DTLS 접속을 확립한다. 파라미터 프로비저닝 프로세스의 일부로서, CSE1(604)은 CSE3(502)의 URI 및 CSE1(604)과 CSE3(502) 사이의 E2E DTLS 접속을 셋업하는 데 사용되어야 하는 포트 번호를 획득할 수 있다.

[0298] 도 9의 단계 6에서, CSE1(604)으로부터의 Request 메시지는 DTLS 터널 내에서 CSE3(502)에 포워딩된다. 여기서, CSE3은 CSE1(604)으로부터 DTLS 터널을 통한 또 다른 다음 홉으로 가정된다.

[0299] 도 9의 단계 7에서, CSE3(502)은 여기에서는 CSE1(604)일 수 있는 메시지 발신자 정보를 검증한다.

[0300] 도 9에 도시된 단계들을 수행하는 엔티티들은, 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 9에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 9에 도시된 단계들을 수행한다. 또한, 도 9에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0301] **직접 모드를 사용하는 세션 계층에서의 E2E 인증**

[0302] 도 10은 직접 모드를 사용하는 세션 계층에서의 E2E 인증을 도시하고, 전술한 메커니즘들과는 달리, CSE3(502)과 연관되는 TTP로부터 획득된 크리덴셜들에 기초하여 AE1(602)이 CSE3(502)와의 직접적인 DTLS 접속을 셋업한다. URI, 포트 번호 및 AE ID를 사용하여, 리소스가 적절하게 구성된다. CSE3(502)는 메시지의 발신자를 검증한다.

[0303] 도 10에 도시된 단계들을 수행하는 엔티티들은, 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 10에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 10에 도시된 단계들을 수행한다. 또한, 도 10에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어 하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터 실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0304] **그룹 인증**

[0305] 엔티티들은 그 엔티티들에 의해 제공되는 능력들 및 기능들에 기초하여 그룹화될 수 있다. 동일한 유형의 서비스를 제공하는 각각의 엔티티들은 "서비스 아이덴티티"에 의해 식별될 수 있고, 또는 oneM2M의 경우, M2M 서비스 제공자 도메인 내에서 고유하거나 또는 심지어 전역적으로 고유할 수 있는 "애플리케이션 아이덴티티"로서 식별될 수 있다는 것이 상상될 것이다.

[0306] 케이스 1 : 그룹 인증은 몇 가지 방식들로 수행될 수 있다.

[0307] ○ 각각의 그룹과 연관된 고유한 그룹 키.

[0308] ○ 서비스 인에이블먼트 및 보안 구성 프로세스 동안의 프로비저닝.

[0309] ○ 동일한 그룹에 속한 모든 엔티티들이 동일한 E2E 그룹 키를 공유한다.

[0310] ○ 프로비저닝 페이즈 동안에 사전 프로비저닝될 수 있는 그룹 인증 키를 사용하여 서로를 E2E 인증한다.

[0311] ○ 인증 후에 그룹 세션 키들이 유도될 수 있고, 그룹 관리자(예를 들어, CSE)에 의해 프로비저닝되는 그룹 멤

버들 간에 공유될 수 있다.

[0312] 케이스 2 : 고유한 그룹 키는 없지만, 고유한 E2E 인증 키가 있는 경우(E2E 메시징을 감소시킴) - 위임 인증의 특별한 경우

[0313] ○ 그룹 관리자(예를 들어, 호스팅 CSE)에 의해 그룹이 관리될 수 있다.

[0314] ○ 모든 그룹 멤버들이 호스팅 CSE에 등록되어 있다.

[0315] ○ 그룹의 각각의 멤버가 고유한 E2E 인증 키를 가지고 있다.

[0316] ○ 이전 섹션들에서 이미 설명한 바와 같이, 그룹 멤버들은 고유한 E2E 인증 키를 사용하여 원격 CSE 또는 임의의 기타 엔티티에 의해 인증된다.

[0317] 케이스 3 : 하이브리드 모드:

[0318] ○ 그룹은 그룹 관리자에 의해 관리되며, 그룹 관리자는 자체 그룹 관리자 키(GM 키)로 사전에 프로비저닝된다.

[0319] ○ 모든 그룹 멤버들은 그룹 관리자에 의해 등록된다.

[0320] ○ 그룹의 각각의 멤버는 고유한 E2E 인증 키를 가지고 있다.

[0321] ○ 그룹 관리자는 GM 키를 초기 그룹 인증을 위한 E2E 그룹 키로서 사용한다.

[0322] ○ 각각의 엔티티에 고유한 E2E 인증 키들이 추가적인 멀티-팩터/멀티-계층 인증들에 사용된다.

[0323] ○ 그룹의 새로운 멤버들은 E2E 그룹 키를 사용하여 자신의 고유한 E2E 키들을 획득할 수 있고, 또는 서비스 인에이블먼트 및 보안 구성 프로세스 동안에 프로비저닝될 수 있다.

[0324] 그룹 인증 프로시져들은 다음을 포함할 수 있다.

[0325] ○ 그룹에 사용되는 공통 보안 파라미터들에 대한 그룹 관리자와 그룹 멤버들 간의 협의.

[0326] ○ 그룹의 새로운/장래의 멤버들에게 사용될 크리덴셜들, 및 그룹 멤버 정보 및 연관된 크리덴셜들의 파기에 대한 그룹 관리자와 TTP 간의 협의.

[0327] 위임 인증 모드 시나리오에서 그룹 키가 프로비저닝되지 않고 엔티티들 간에 고유한 E2E 키를 사용하는 그룹 인증의 실시예가 도 11a 및 도 11b에 도시되어 있다. 그룹은 그룹 관리자(예를 들어, CSE1(604))에 의해 관리된다. 관련 단계들이 다음과 같이 설명된다.

[0328] 도 11a 및 도 11b의 단계 1 내지 단계 6은 이전 섹션들에서 설명된 메커니즘들을 따른다.

[0329] 도 11a 및 도 11b의 단계 7에서, 메시징의 타겟에 기초하여 추론된 정보에 기초하여, CSE1은 MAC1(엔티티 A와 연관된 Kpsa_E2E1을 사용하여 생성됨)과 함께 메시지 2의 관련 부분들 및 MAC2(엔티티 B와 연관된 Kpsa_E2E2를 사용하여 생성됨)와 함께 메시지 6의 관련 부분들을 포함하는 통합된 메시지를 생성한다.

[0330] 도 11a 및 도 11b의 단계 8은 이전 섹션들에서 설명된 것과 동일하다.

[0331] 도 11a 및 도 11b의 단계 9에서, CSE1(604)은 KpsaE2E를 사용하여 CSE3(502)(타겟)과 (D)TLS 접속을 생성한다.

[0332] 도 11a 및 도 11b의 단계 10에서, 단계 7에서 생성된 통합된 메시지는 CSE1(604)에 의해 (D)TLS 접속을 통해 CSE3(502)에 보안 전송된다.

[0333] 도 11a 및 도 11b의 단계 11에서, CSE3(502)은 2개의 엔티티들(AE1(602) 및 cse1(102))로부터 발생하는 2개의 서비스 계층 메시지들이 존재하는 것을 검증하고, 발신자들 또는 메시지를 검증함으로써 각각의 MAC들을 검증하며, 또한 메시지들이 리플레이되지 않았다는 것을 보장한다.

[0334] 도 11a 및 도 11b에 도시된 단계들을 수행하는 엔티티들은, 도 21c 또는 도 21d에 도시된 것과 같은 네트워크 노드 또는 컴퓨터 시스템의 메모리에 저장되고 또한 그 프로세서 상에서 실행되는 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있는 로지컬 엔티티들이라는 것이 이해된다. 즉, 도 11a 및 도 11b에 도시된 방법(들)은 도 21c 또는 도 21d에 도시된 노드 또는 컴퓨터 시스템과 같은 네트워크 노드의 메모리에 저장된 소프트웨어(즉, 컴퓨터 실행가능 명령어들)의 형태로 구현될 수 있으며, 컴퓨터 실행가능 명령어들은, 노드의 프로세서에 의해 실행될 때, 도 11a 및 도 11b에 도시된 단계들을 수행한다. 또한, 도 11a 및 도 11b에 도시된 임의의 송신 및 수신 단계들은 노드의 프로세서의 제어하에 있는 노드의 통신 회로 및 그것이 실행하는 컴퓨터

실행가능 명령어들(예를 들어, 소프트웨어)에 의해 수행될 수 있다는 것이 이해된다.

[0335] **인터페이스들**

[0336] GUI(Graphical User Interface)들과 같은 인터페이스들은 엔드 투 엔드 인가와 관련된 기능들을 사용자가 제어 및/또는 구성하는 것을 돕기 위해 사용될 수 있다. 도 12는 사용자가 서비스 인에이블링 기능 및 키 전달 기능을 구성하는 것을 포함하여 엔드 투 엔드 인증을 선택 및 구성할 수 있게 하는 인터페이스(1202)를 도시하는 도면이다. 사용자 인터페이스(1202)는 M2M 디바이스/게이트웨이/서버에서 엔드 투 엔드 보안 정책들 및 연관된 보안 파라미터들을 구성/디스플레이하는 데 사용될 수 있다. 인터페이스(2102)는 이하에 기술되는 도 21c 및 도 21d에 도시된 것과 같은 디스플레이들을 사용하여 생성될 수 있다는 것이 이해될 것이다.

[0337] **예시적인 M2M/IoT/WoT 통신 시스템**

[0338] 도 21a는 하나 이상의 개시된 실시예들이 구현될 수 있는 예시적인 머신 대 머신(M2M), 사물 인터넷(IoT) 또는 사물 웹(Web of Things)(WoT) 통신 시스템(10)의 도면이다. 일반적으로, M2M 기술들은 IoT/WoT를 위한 빌딩 블록들을 제공하며, 임의의 M2M 디바이스, M2M 게이트웨이, M2M 서버 또는 M2M 서비스 플랫폼은 IoT/WoT의 컴포넌트 또는 노드뿐만 아니라, IoT/WoT 서비스 계층 등일 수 있다. 통신 시스템(10)은 개시된 실시예들의 기능을 구현하는 데 사용될 수 있고, 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604, 704 및 2002), CSF(408 및 412), AE1(602), AE2(1102), SP 레포지토리(2004), DP/EP 레포지토리(2006), MEF(2008) 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 기능 및 로지컬 엔티티들을 포함할 수 있다.

[0339] 도 21a에 도시된 바와 같이, M2M/IoT/WoT 통신 시스템(10)은 통신 네트워크(12)를 포함한다. 통신 네트워크(12)는 고정 네트워크(예를 들어, 이더넷, 파이버, ISDN, PLC 등) 또는 무선 네트워크(예를 들어, WLAN, 셀룰러 등) 또는 이종 네트워크들의 네트워크일 수 있다. 예를 들어, 통신 네트워크(12)는 음성, 데이터, 비디오, 메시징, 브로드캐스트 등과 같은 컨텐츠를 복수의 사용자들에게 제공하는 복수의 액세스 네트워크들로 구성될 수 있다. 예를 들어, 통신 네트워크(12)는 코드 분할 다중 액세스(CDMA), 시분할 다중 액세스(TDMA), 주파수 분할 다중 액세스(FDMA), 직교 FDMA(OFDMA), 싱글-캐리어 FDMA(SC-FDMA) 등과 같은 하나 이상의 채널 액세스 방법들을 채택할 수 있다. 또한, 통신 네트워크(12)는 예를 들어, 코어 네트워크, 인터넷, 센서 네트워크, 산업 제어 네트워크, 개인 영역 네트워크, 융합된 개인 네트워크, 위성 네트워크, 홈 네트워크 또는 엔터프라이즈 네트워크와 같은 다른 네트워크들을 포함할 수 있다.

[0340] 도 21a에 도시된 바와 같이, M2M/IoT/WoT 통신 시스템(10)은 인프라스트럭처 도메인 및 필드 도메인을 포함할 수 있다. 인프라스트럭처 도메인은 엔드 투 엔드 M2M 배치의 네트워크 측을 지칭하며, 필드 도메인은 일반적으로 M2M 게이트웨이 뒤에 있는 영역 네트워크들을 지칭한다. 필드 도메인 및 인프라스트럭처 도메인은 모두 다양한 상이한 네트워크 노드들(예를 들어, 서버들, 게이트웨이들, 디바이스 등)을 포함할 수 있다. 예를 들어, 필드 도메인은 M2M 게이트웨이들(14) 및 단말 디바이스들(18)을 포함할 수 있다. 임의의 수의 M2M 게이트웨이 디바이스들(14) 및 M2M 단말 디바이스들(18)이 원하는 대로 M2M/IoT/WoT 통신 시스템(10)에 포함될 수 있다는 것이 이해될 것이다. M2M 게이트웨이 디바이스들(14) 및 M2M 단말 디바이스들(18) 각각은 통신 회로를 이용하여 통신 네트워크(12) 또는 직접 무선 링크를 통해 신호들을 송신 및 수신하도록 구성된다. M2M 게이트웨이(14)는 무선 M2M 디바이스들(예를 들어, 셀룰러 및 비-셀룰러)뿐만 아니라 고정 네트워크 M2M 디바이스들(예를 들어, PLC)이 통신 네트워크(12) 또는 직접 무선 링크와 같은 운영자 네트워크들을 통해 통신하게 할 수 있다. 예를 들어, M2M 단말 디바이스(18)는 데이터를 수집하고, 통신 네트워크(12) 또는 직접 무선 링크를 통해 M2M 애플리케이션(20) 또는 다른 M2M 디바이스들(18)에 데이터를 전송할 수 있다. M2M 단말 디바이스들(18)은 또한 M2M 애플리케이션(20) 또는 M2M 단말 디바이스(18)로부터 데이터를 수신할 수 있다. 또한, 데이터 및 신호들은 이하에 설명되는 바와 같이 M2M 애플리케이션(20)으로부터 M2M 서비스 계층(22)을 통해 전송 및 수신될 수 있다. M2M 단말 디바이스들(18) 및 게이트웨이들(14)은 예를 들어, 셀룰러, WLAN, WPAN(예를 들어, 지그비, 6LoWPAN, 블루투스), 직접 무선 링크 및 유선을 포함하는 다양한 네트워크들을 통해 통신할 수 있다.

[0341] 예시적인 M2M 단말 디바이스들(18)은 태블릿들, 스마트폰들, 의료 디바이스들, 온도 및 날씨 모니터들, 커넥티드 카들, 스마트 미터들, 게임 콘솔들, 개인용 정보 단말기들, 건강 및 피트니스 모니터들, 조명들, 서모스탯들, 가전 제품들, 차고 문들 및 기타 액추에이터 기반 디바이스들, 보안 디바이스들 및 스마트 콘센트들(smart outlets)을 포함하지만, 이에 제한되지 않는다.

[0342] 도 21b를 참조하면, 필드 도메인에 도시된 M2M 서비스 계층(22)은 M2M 애플리케이션(20), M2M 게이트웨이 디바

이스들(14) 및 M2M 단말 디바이스들(18) 및 통신 네트워크(12)에 대한 서비스들을 제공한다. 통신 네트워크(12)는 개시된 실시예들의 기능을 구현하는 데 사용될 수 있으며, 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604, 704 및 2002), CSF(408 및 412), AE1(602), AE2(1102), SP 레포지토리(2004), DP/EP 레포지토리(2006), MEF(2008) 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 기능 및 로지컬 엔티티들을 포함할 수 있다. M2M 서비스 계층(22)은 예를 들어, 아래에 기술된 도 21c 및 도 21d에 도시된 디바이스들을 포함하여, 하나 이상의 서버들, 컴퓨터들, 디바이스들, 가상 머신들(예를 들어, 클라우드/스토리지 팜들 등) 등에 의해 구현될 수 있다. M2M 서비스 계층(22)은 임의의 수의 M2M 애플리케이션들, M2M 게이트웨이들(14), M2M 단말 디바이스들(18) 및 통신 네트워크들(12)과 원하는 대로 통신할 수 있다는 것이 이해될 것이다. M2M 서비스 계층(22)은 서버들, 컴퓨터들, 디바이스들 등을 포함할 수 있는 네트워크의 하나 이상의 노드들에 의해 구현될 수 있다. M2M 서비스 계층(22)은 M2M 단말 디바이스들(18), M2M 게이트웨이들(14) 및 M2M 애플리케이션들(20)에 적용되는 서비스 능력들을 제공한다. M2M 서비스 계층(22)의 기능들은 다양한 방식들로, 예를 들어, 웹 서버로서, 셀룰러 코어 네트워크에서, 클라우드에서 등과 같이 구현될 수 있다.

[0343] 도시된 M2M 서비스 계층(22)과 유사하게, 인프라스트럭쳐 도메인 내에도 M2M 서비스 계층(22')이 있다. M2M 서비스 계층(22')은 인프라스트럭쳐 도메인 내의 M2M 애플리케이션(20') 및 기본 통신 네트워크(12')에 대한 서비스들을 제공한다. M2M 서비스 계층(22')은 또한 필드 도메인의 M2M 게이트웨이들(14) 및 M2M 단말 디바이스들(18)에 대한 서비스들을 제공한다. M2M 서비스 계층(22')은 임의의 수의 M2M 애플리케이션들, M2M 게이트웨이들 및 M2M 디바이스들과 통신할 수 있다는 것이 이해될 것이다. M2M 서비스 계층(22')은 상이한 서비스 제공자에 의한 서비스 계층과 상호 작용할 수 있다. M2M 서비스 계층(22')은 서버들, 컴퓨터들, 디바이스들, 가상 머신들(예를 들어, 클라우드 컴퓨팅/스토리지 팜들 등) 등을 포함할 수 있는 네트워크의 하나 이상의 노드들에 의해 구현될 수 있다.

[0344] 도 21b를 또한 참조하면, M2M 서비스 계층들(22 및 22')은 다양한 애플리케이션들 및 버티컬들이 레버리지될 수 있는 서비스 전달 능력들의 코어 세트를 제공한다. 이러한 서비스 능력들은 M2M 애플리케이션들(20 및 20')이 디바이스들과 상호 작용하고, 데이터 수집, 데이터 분석, 디바이스 관리, 보안, 과금, 서비스/디바이스 발견 등과 같은 기능들을 수행하게 할 수 있다. 본질적으로, 이러한 서비스 능력들은 애플리케이션들이 이러한 기능들을 구현할 부담을 제거함으로써, 애플리케이션 개발을 간소화하고, 비용 및 출시 시간을 줄일 수 있다. 서비스 계층들(22 및 22')은 또한 M2M 애플리케이션들(20 및 20')이 서비스 계층들(22 및 22')이 제공하는 서비스들과 관련하여 다양한 네트워크들(12 및 12')을 통해 통신할 수 있게 한다.

[0345] 본 출원의 방법들은 서비스 계층(22 및 22')의 일부로서 구현될 수 있다. 서비스 계층(22 및 22')은 애플리케이션 프로그래밍 인터페이스(Application Programming Interface)(API)들 및 기본 네트워킹 인터페이스들의 세트를 통해 부가 가치 서비스 능력들을 지원하는 소프트웨어 미들웨어 계층이다. ETSI M2M 및 oneM2M은 모두 본 출원의 접속 방법들을 포함할 수 있는 서비스 계층을 사용한다. ETSI M2M의 서비스 계층은 서비스 능력 계층(Service Capability Layer)(SCL)이라고 지칭한다. SCL은 M2M 디바이스(디바이스 SCL(DSCL)로 지칭됨), 게이트웨이(게이트웨이 SCL(GSCL)로 지칭됨) 및/또는 네트워크 노드(네트워크 SCL(NSCL)로 지칭됨) 내에서 구현될 수 있다. oneM2M 서비스 계층은 공통 서비스 기능(CSF)들(즉, 서비스 능력들)의 세트를 지원한다. 하나 이상의 특정한 유형들의 CSF들의 세트의 인스턴스화는 상이한 유형들의 네트워크 노드들(예를 들어, 인프라스트럭쳐 노드, 미들 노드, 애플리케이션-특정 노드) 상에서 호스팅될 수 있는 공통 서비스 엔티티(Common Services Entity)(CSE)로 지칭된다. 또한, 본 출원의 접속 방법들은 서비스 지향 아키텍처(Service Oriented Architecture)(SOA) 및/또는 리소스 지향 아키텍처(resource-oriented architecture)(ROA)를 사용하여 본 출원의 접속 방법들과 같은 서비스들에 액세스하는 M2M 네트워크의 일부로서 구현될 수 있다.

[0346] 일부 실시예들에서, M2M 애플리케이션들(20 및 20')은 개시된 시스템들 및 방법들과 함께 사용될 수 있다. M2M 애플리케이션들(20 및 20')은 UE 또는 게이트웨이와 상호 작용하는 애플리케이션들을 포함할 수 있고, 다른 개시된 시스템들 및 방법들과 함께 사용될 수도 있다.

[0347] 일 실시예에서, 도 21b에 도시된 바와 같이, 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604, 704 및 2002), CSF(408 및 412), AE1(602), AE2(1102), SP 레포지토리(2004), DP/EP 레포지토리(2006), MEF(2008) 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 로지컬 엔티티들은 M2M 서버, M2M 게이트웨이 또는 M2M 디바이스와 같은 M2M 노드에 의해 호스팅되는 M2M 서비스 계층 인스턴스 내에서 호스팅될 수 있다. 예를 들어, 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604 및 704), CSF(408 및 412), AE1(602), AE2(1102) 및 사용자

인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 로지컬 엔티티들은 M2M 서비스 계층 인스턴스 내에 또는 기존의 서비스 능력 내의 서브 기능으로서 개별적인 서비스 능력을 포함할 수 있다.

[0348] M2M 애플리케이션들(20 및 20')은 운송, 건강 및 건강 관리, 커넥티드 홈, 에너지 관리, 자산 추적, 및 보안 및 감시와 같은 다양한 산업 분야들의 애플리케이션들을 포함할 수 있다. 위에서 언급한 바와 같이, 시스템의 디바이스들, 게이트웨이들, 서버들 및 기타 노드들을 통해 실행되는 M2M 서비스 계층은 예를 들어, 데이터 수집, 디바이스 관리, 보안, 과금, 위치 추적/지오펜싱, 디바이스/서비스 발견 및 레거시 시스템 통합과 같은 기능들을 지원하고, 이러한 기능들을 M2M 애플리케이션들(20 및 20')에 서비스들로서 제공한다.

[0349] 일반적으로, 서비스 계층들(22, 22')은 애플리케이션 프로그래밍 인터페이스(API)들 및 기본 네트워킹 인터페이스들의 세트를 통해 부가 가치 서비스 능력들을 지원하는 소프트웨어 미들웨어 계층을 정의한다. ETSI M2M 및 oneM2M 아키텍처들은 모두 서비스 계층을 정의한다. ETSI M2M의 서비스 계층은 서비스 능력 계층(SCL)이라고 지칭된다. SCL은 ETSI M2M 아키텍처의 다양한 상이한 노드들에서 구현될 수 있다. 예를 들어, 서비스 계층의 인스턴스는 M2M 디바이스(디바이스 SCL(DSCL)이라고 지칭됨), 게이트웨이(게이트웨이 SCL(GSCL)이라고 지칭됨) 및/또는 네트워크 노드(네트워크 SCL(NSCL)이라고 지칭됨) 내에서 구현될 수 있다. oneM2M 서비스 계층은 공통 서비스 기능(CSF)들(즉, 서비스 능력들)의 세트를 지원한다. 하나 이상의 특정 유형들의 CSF들의 세트의 인스턴스화는 상이한 유형들의 네트워크 노드들(예를 들어, 인프라스트럭쳐 노드, 미들 노드, 애플리케이션-특정 노드) 상에서 호스팅될 수 있는 공통 서비스 엔티티(CSE)로 지칭된다. 3세대 파트너십 프로젝트(Third Generation Partnership Project)(3GPP)는 또한 머신 유형 통신(machine-type communications)(MTC)을 위한 아키텍처를 정의했다. 그 아키텍처에서, 그것이 제공하는 서비스 계층과 서비스 능력들은 서비스 능력 서버(Service Capability Server)(SCS)의 일부로서 구현된다. ETSI M2M 아키텍처의 DSCL, GSCL 또는 NSCL에서, 3GPP MTC 아키텍처의 서비스 능력 서버(SCS)에서, oneM2M 아키텍처의 CSF 또는 CSE에서, 또는 네트워크의 일부 다른 노드에서 구현되든 아니든 간에, 서비스 계층의 인스턴스는 서버들, 컴퓨터들 및 다른 컴퓨팅 디바이스들 또는 노드들을 포함하는 네트워크 내의 하나 이상의 독립형 노드들 상에서, 또는 하나 이상의 기존의 노드들의 일부로서 실행되는 로지컬 엔티티(예를 들어, 소프트웨어, 컴퓨터 실행가능 명령어 등)로서 구현될 수 있다. 예로서, 서비스 계층 또는 그 컴포넌트의 인스턴스는 이하에서 설명되는 도 21c 또는 도 21d에 도시된 일반적인 아키텍처를 갖는 네트워크 노드(예를 들어, 서버, 컴퓨터, 게이트웨이, 디바이스 등) 상에서 실행되는 소프트웨어의 형태로 구현될 수 있다.

[0350] 또한, 서비스 인에이블링 기능(204, 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604 및 704), CSF(408 및 412), AE1(602), AE2(1102) 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 로지컬 엔티티들이 본 출원의 서비스들에 액세스하기 위해 서비스 지향 아키텍처(SOA) 및/또는 리소스 지향 아키텍처(ROA)를 사용하는 M2M 네트워크의 일부로서 구현될 수 있다.

[0351] 도 21c는 M2M 디바이스(18), M2M 게이트웨이(14), M2M 서버 등과 같은 M2M 네트워크 노드(30)의 예시적인 하드웨어/소프트웨어 아키텍처의 블록도이다. 노드(30)는 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604, 704 및 2002), CSF(408 및 412), AE1(602), AE2(1102), SP 레포지토리(2004), DP/EP 레포지토리(2006), MEF(2008) 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 로지컬 엔티티들을 실행하거나 포함할 수 있다. 디바이스(30)는 도 21a 및 도 21b에 도시된 바와 같은 M2M 네트워크의 일부 또는 비 M2M 네트워크의 일부일 수 있다. 도 21c에 도시된 바와 같이, M2M 노드(30)는 프로세서(32), 비이동식 메모리(44), 이동식 메모리(46), 스피커/마이크로폰(38), 키패드(40), 디스플레이, 터치 패드 및/또는 인디케이터들(42), 전원(48), 글로벌 포지셔닝 시스템(global positioning system)(GPS) 칩셋(50) 및 다른 주변 디바이스들(52)을 포함할 수 있다. 노드(30)는 또한 송수신기(34) 및 송신/수신 엘리먼트(36)와 같은 통신 회로를 포함할 수 있다. M2M 노드(30)는 전술한 엘리먼트들의 임의의 서브-조합(sub-combination)을 포함할 수 있으며, 나머지도 실시예와 일맥상통한다는 것이 이해될 것이다. 이 노드는 여기에 설명된 SMSF 기능을 구현하는 노드일 수 있다.

[0352] 프로세서(32)는 범용 프로세서, 전용 프로세서, 종래 프로세서, 디지털 신호 프로세서(DSP), 복수의 마이크로프로세서들, DSP 코어와 연관된 하나 이상의 마이크로프로세서들, 제어기, 마이크로 제어기, ASIC(Application Specific Integrated Circuit)들, FPGA(Field Programmable Gate Array) 회로들, 임의의 다른 유형의 집적 회로(IC), 상태 머신(state machine) 등일 수 있다. 일반적으로, 프로세서(32)는 노드의 다양한 요구된 기능들을 수행하기 위해 노드의 메모리(예를 들어, 메모리(44) 및/또는 메모리(46))에 저장된 컴퓨터 실행가능 명령어들을 실행할 수 있다. 예를 들어, 프로세서(32)는 M2M 노드(30)가 무선 또는 유선 환경에서 동작 할 수 있게 하는 신호 코딩, 데이터 프로세싱, 전력 제어, 입력/출력 프로세싱 및/또는 임의의 다른 기능을 수행할 수 있다.

프로세서(32)는 애플리케이션 계층 프로그램들(예를 들어, 브라우저들) 및/또는 무선 액세스 계층(radio access-layer)(RAN) 프로그램들 및/또는 다른 통신 프로그램들을 실행할 수 있다. 프로세서(32)는 또한 예를 들어, 액세스 계층 및/또는 애플리케이션 계층에서와 같이 인증, 보안 키 합의 및/또는 암호 동작들과 같은 보안 동작들을 수행할 수 있다.

[0353] 도 21c에 도시된 바와 같이, 프로세서(32)는 자신의 통신 회로(예를 들어, 송수신기(34) 및 송신/수신 엘리먼트(36))에 연결된다. 프로세서(32)는 컴퓨터 실행가능 명령어들의 실행을 통해 노드(30)가 자신이 접속된 네트워크를 통해 다른 노드들과 통신하게 하기 위해 통신 회로를 제어할 수 있다. 특히, 프로세서(32)는 본 명세서 및 청구 범위에서 설명된 송신 및 수신 단계들을 수행하기 위해 통신 회로를 제어할 수 있다. 도 21c는 프로세서(32) 및 송수신기(34)를 별개의 컴포넌트들로서 도시하지만, 프로세서(32) 및 송수신기(34)는 전자 패키지 또는 칩 내에 함께 통합될 수 있다는 것이 이해될 것이다.

[0354] 송신/수신 엘리먼트(36)는 M2M 서버들, 게이트웨이들, 디바이스 등을 포함하여 다른 M2M 노드들에 신호들을 송신하거나 그로부터 신호들을 수신하도록 구성될 수 있다. 예를 들어, 실시예에서, 송신/수신 엘리먼트(36)는 RF 신호들을 송신 및/또는 수신하도록 구성된 안테나일 수 있다. 송신/수신 엘리먼트(36)는 WLAN, WPAN, 셀룰러 등과 같은 다양한 네트워크들 및 무선 인터페이스들을 지원할 수 있다. 실시예에서, 송신/수신 엘리먼트(36)는 예를 들어, IR, UV 또는 가시광 신호들을 송신 및/또는 수신하도록 구성된 이미터/검출기일 수 있다. 또 다른 실시예에서, 송신/수신 엘리먼트(36)는 RF 및 광 신호들 모두를 송신 및 수신하도록 구성될 수 있다. 송신/수신 엘리먼트(36)는 무선 또는 유선 신호들의 임의의 조합을 송신 및/또는 수신하도록 구성될 수 있다는 것이 이해될 것이다.

[0355] 또한, 송신/수신 엘리먼트(36)가 단일 엘리먼트로서 도 21c에 도시되어 있지만, M2M 노드(30)는 임의의 수의 송신/수신 엘리먼트들(36)을 포함할 수 있다. 보다 구체적으로, M2M 노드(30)는 MIMO 기술을 채택할 수 있다. 따라서, 실시예에서, M2M 노드(30)는 무선 신호들을 송신 및 수신하기 위한 2개 이상의 송신/수신 엘리먼트들(36)(예를 들어, 복수의 안테나들)을 포함할 수 있다.

[0356] 송수신기(34)는 송신/수신 엘리먼트(36)에 의해 송신되는 신호들을 변조하고 송신/수신 엘리먼트(36)에 의해 수신되는 신호들을 복조하도록 구성될 수 있다. 전술한 바와 같이, M2M 노드(30)는 멀티-모드 능력들을 가질 수 있다. 따라서, 송수신기(34)는 예를 들어, UTRA 및 IEEE 802.11과 같은 복수의 RAT들을 통해 M2M 노드(30)가 통신할 수 있게 하는 복수의 송수신기들을 포함할 수 있다.

[0357] 프로세서(32)는 비이동식 메모리(44) 및/또는 이동식 메모리(46)와 같은 임의의 유형의 적절한 메모리로부터 정보를 액세스할 수 있고, 그 안에 데이터를 저장할 수 있다. 예를 들어, 전술한 바와 같이, 프로세서(32)는 그 메모리 내에 세션 컨텍스트를 저장할 수 있다. 비이동식 메모리(44)는 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM), 하드 디스크 또는 임의의 다른 유형의 메모리 스토리지 디바이스를 포함할 수 있다. 이동식 메모리(46)는 가입자 식별 모듈(subscriber identity module)(SIM) 카드, 메모리 스틱, 보안 디지털(secure digital)(SD) 메모리 카드 등을 포함할 수 있다. 다른 실시예들에서, 프로세서(32)는 서버 또는 가정용 컴퓨터 상과 같이 M2M 노드(30) 상에 물리적으로 위치하지 않는 메모리로부터 정보를 액세스하고, 그 메모리에 데이터를 저장할 수 있다. 프로세서(32)는, M2M 서비스 계층 세션 마이그레이션 또는 공유의 상태를 반영하거나 또는 사용자로부터 입력을 획득하거나 또는 노드의 세션 마이그레이션 또는 공유 능력들 또는 설정들에 관한 정보를 사용자에게 디스플레이하기 위해, 디스플레이 또는 인디케이터(들)(42) 상의 조명 패턴들, 이미지들 또는 컬러들을 제어하도록 구성될 수 있다. 다른 예에서, 디스플레이는 세션 상태에 관한 정보를 나타낼 수 있다. 본 개시내용은 oneM2M 실시예에서 RESTful 사용자/애플리케이션 API를 정의한다. 디스플레이 상에 보일 수 있는 그래픽 사용자 인터페이스는, 여기에 기술된 기본 서비스 계층 세션 기능을 통해 사용자가 E2E 세션 또는 그것의 마이그레이션 또는 공유를 상호작용식으로 확립하고 관리할 수 있게 하기 위해, API 상부에 계층화될 수 있다.

[0358] 프로세서(32)는 전원(48)으로부터 전력을 수신할 수 있고, M2M 노드(30) 내의 다른 컴포넌트들에 전력을 분배 및/또는 제어하도록 구성될 수 있다. 전원(48)은 M2M 노드(30)에 전력을 공급하기 위한 임의의 적절한 디바이스일 수 있다. 예를 들어, 전원(48)은 하나 이상의 건전지 배터리들(예를 들어, 니켈-카드뮴(NiCd), 니켈-아연(NiZn), 니켈 금속 수소화물(NiMH), 리튬-이온(Li-이온) 등), 태양 전지, 연료 전지 등을 포함할 수 있다.

[0359] 프로세서(32)는 또한 M2M 노드(30)의 현재 위치에 관한 위치 정보(예를 들어, 경도 및 위도)를 제공하도록 구성된 GPS 칩셋(50)에 연결될 수 있다. M2M 노드(30)는 임의의 적절한 위치 결정 방법을 통해 위치 정보를 취득할 수 있으며, 나머지도 실시예와 일맥상통한다는 것이 이해될 것이다.

[0360] 프로세서(32)는 추가 피쳐들, 기능 및/또는 유선 또는 무선 접속성을 제공하는 하나 이상의 소프트웨어 및/또는 하드웨어 모듈들을 포함할 수 있는 다른 주변 디바이스들(52)에 추가로 연결될 수 있다. 예를 들어, 주변 디바이스들(52)은 가속도계, 전자 나침반, 위성 송수신기, 센서, 디지털 카메라(사진 또는 비디오 용), 범용 직렬 버스(USB) 포트, 진동 디바이스, 텔레비전 송수신기, 핸즈프리 헤드셋, 블루투스® 모듈, 주파수 변조(FM) 라디오 유닛, 디지털 음악 플레이어, 미디어 플레이어, 비디오 게임 플레이어 모듈, 인터넷 브라우저 등을 포함할 수 있다.

[0361] 도 21d는 M2M 서버, 게이트웨이, 디바이스 또는 다른 노드와 같은 M2M 네트워크의 하나 이상의 노드들을 구현하는 데에도 사용될 수 있는 예시적인 컴퓨팅 시스템(90)의 블록도이다. 컴퓨팅 시스템(90)은 컴퓨터 또는 서버를 포함할 수 있으며, 소프트웨어의 형태일 수 있는 컴퓨터 판독가능 명령어들에 의해 주로, 어디에서든, 또는 이러한 소프트웨어가 저장되거나 액세스되는 수단이 무엇이든 간에 제어될 수 있다. 컴퓨팅 시스템(90)은 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604, 704 및 2002), CSF(408 및 412), AE1(602), AE2(1102), SP 레포지토리(2004), DP/EP 레포지토리(2006), MEF(2008) 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 로지컬 엔티티들을 실행하거나 포함할 수 있다. 컴퓨팅 시스템(90)은 예를 들어, M2M 디바이스, 사용자 장비, 게이트웨이, UE/GW, 또는 모바일 케어 네트워크, 서비스 계층 네트워크 애플리케이션 제공자, 단말 디바이스(18) 또는 M2M 게이트웨이 디바이스(14)의 노드들을 포함하는 임의의 다른 노드들일 수 있다. 이러한 컴퓨터 판독가능 명령어들은 중앙 처리 장치(CPU)(91)와 같은 프로세서 내에서 실행되어 컴퓨팅 시스템(90)이 작업을 수행하게 할 수 있다. 많은 공지된 워크 스테이션들, 서버들 및 퍼스널 컴퓨터들에서, 중앙 처리 장치(91)는 마이크로프로세서라고 불리는 단일 칩 CPU에 의해 구현된다. 다른 머신들에서, 중앙 처리 장치(91)는 복수의 프로세서들을 포함할 수 있다. 코-프로세서(81)는 추가적인 기능들을 수행하거나 CPU(91)를 보조하는, 메인 CPU(91)와 구별되는 임의의 프로세서이다. CPU(91) 및/또는 코-프로세서(81)는 세션 크리덴셜들의 수신 또는 세션 크리덴셜들에 기초한 인증과 같이 E2E M2M 서비스 계층 세션들에 대한 개시된 시스템들 및 방법들에 관련된 데이터를 수신, 생성 및 프로세싱할 수 있다.

[0362] 동작시, CPU(91)는 명령어들을 페치, 디코딩 및 실행하고, 컴퓨터의 메인 데이터-전달 경로인 시스템 버스(80)를 통해 다른 리소스들에/로부터 정보를 전송한다. 이러한 시스템 버스는 컴퓨팅 시스템(90) 내의 컴포넌트들을 접속하고, 데이터 교환을 위한 매체를 정의한다. 시스템 버스(80)는 전형적으로 데이터를 전송하기 위한 데이터 라인, 어드레스들을 전송하기 위한 어드레스 라인들, 및 인터럽트들을 전송하고 시스템 버스를 동작시키기 위한 제어 라인들을 포함한다. 이러한 시스템 버스(80)의 예는 PCI(Peripheral Component Interconnect) 버스이다.

[0363] 시스템 버스(80)에 연결된 메모리들은 랜덤 액세스 메모리(RAM)(82) 및 판독 전용 메모리(ROM)(93)를 포함한다. 이러한 메모리들은 정보가 저장되고 리트리브되도록 하는 회로를 포함한다. ROM들(93)은 일반적으로 쉽게 수정될 수 없는 저장된 데이터를 포함한다. RAM(82)에 저장된 데이터는 CPU(91) 또는 다른 하드웨어 디바이스들에 의해 판독되거나 변경될 수 있다. RAM(82) 및/또는 ROM(93)에 대한 액세스는 메모리 제어기(92)에 의해 제어될 수 있다. 메모리 제어기(92)는, 명령어들이 실행될 때, 가상 어드레스들을 물리적 어드레스들로 변환하는 어드레스 변환 기능을 제공할 수 있다. 메모리 제어기(92)는 또한 시스템 내의 프로세스들을 분리시키고 사용자 프로세스들로부터 시스템 프로세스들을 분리시키는 메모리 보호 기능을 제공할 수 있다. 따라서, 제1 모드로 실행되는 프로그램은 자신의 프로세스의 가상 어드레스 공간에 의해 매핑된 메모리에만 액세스할 수 있고, 프로세스들 간의 메모리 공유가 셋업되어 있지 않으면, 다른 프로세스의 가상 어드레스 공간 내의 메모리에 액세스 할 수 없다.

[0364] 또한, 컴퓨팅 시스템(90)은 CPU(91)로부터 프린터(94), 키보드(84), 마우스(95) 및 디스크 드라이브(85)와 같은 주변 디바이스들로 명령어들을 전달할 책임이 있는 주변 디바이스 제어기(83)를 포함할 수 있다.

[0365] 디스플레이 제어기(96)에 의해 제어되는 디스플레이(86)는 컴퓨팅 시스템(90)에 의해 생성된 시각적 출력을 디스플레이하는 데 사용된다. 이러한 시각적 출력은 텍스트, 그래픽, 애니메이션 그래픽 및 비디오를 포함할 수 있다. 디스플레이(86)는 CRT-기반 비디오 디스플레이, LCD-기반 평면 패널 디스플레이, 가스 플라즈마-기반 평면 패널 디스플레이, 또는 터치 패널로 구현될 수 있다. 디스플레이 제어기(96)는 디스플레이(86)로 전송되는 비디오 신호를 생성하는 데 요구되는 전자 컴포넌트들을 포함한다.

[0366] 또한, 컴퓨팅 시스템(90)은 컴퓨팅 시스템(90)을 도 21a 및 도 21b의 네트워크(12)와 같은 외부 통신 네트워크에 접속시키는 데 사용될 수 있는, 예를 들어, 네트워크 어댑터(97)와 같은 통신 회로를 포함할 수 있어, 컴퓨

팅 시스템(90)이 네트워크의 다른 노드들과 통신할 수 있게 한다.

[0367] 사용자 장비(UE)는 통신하기 위해 최종 사용자에 의해 사용되는 임의의 디바이스일 수 있다. 이것은 핸드헬드 전화기, 모바일 광대역 어댑터가 장착된 랩탑 컴퓨터 또는 임의의 기타 디바이스일 수 있다. 예를 들어, UE는 도 21a 및 도 21b의 M2M 단말 디바이스(18) 또는 도 21c의 디바이스(30)로서 구현될 수 있다.

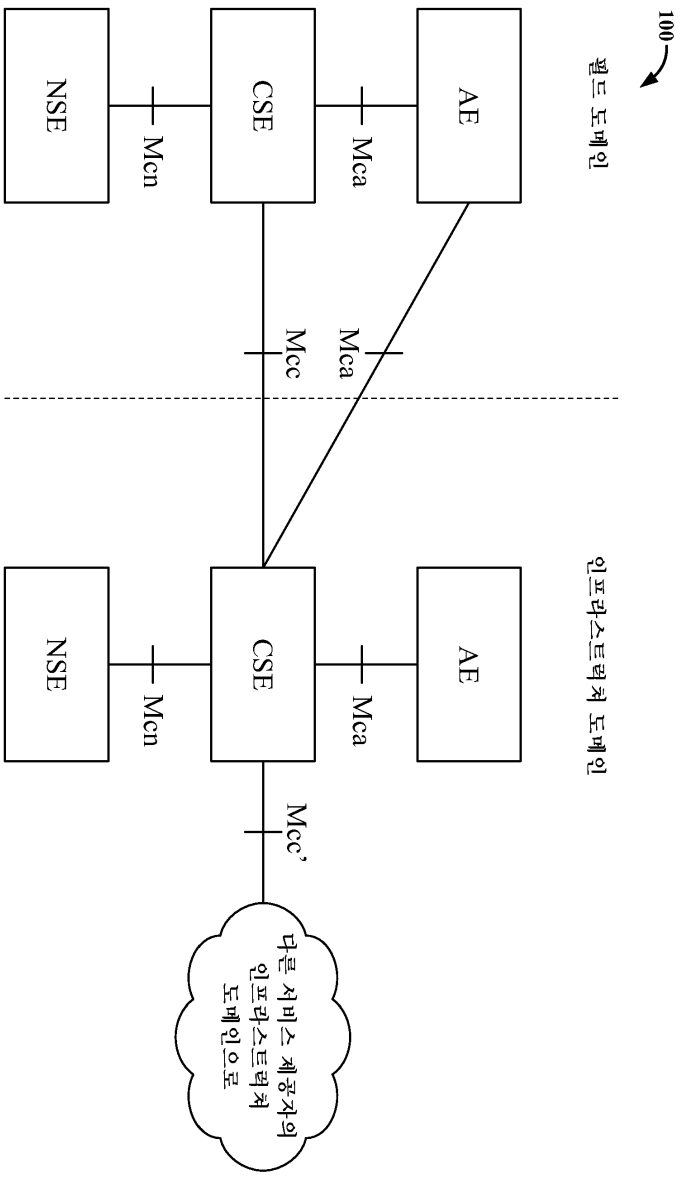
[0368] 본 명세서에 기재된 시스템들, 방법들 및 프로세스들 중 임의의 것 또는 모든 것이 컴퓨터 판독가능 저장 매체 상에 저장된 컴퓨터 실행가능 명령어들(즉, 프로그램 코드)의 형태로 구현될 수 있으며, 명령어들은, 예를 들어, M2M 서버, 게이트웨이, 디바이스 등을 포함하는 M2M 네트워크의 노드와 같은 머신에 의해 실행될 때, 본 명세서에 기재된 시스템들, 방법들 및 프로세스들을 수행 및/또는 구현한다. 구체적으로, 게이트웨이, UE, UE/GW, 또는 모바일 코어 네트워크, 서비스 계층 또는 네트워크 애플리케이션 제공자의 노드들 중 임의의 노드의 동작들을 포함하여, 전술한 단계들, 동작들 또는 기능들 중 임의의 것이 그러한 컴퓨터 실행가능 명령어들의 형태로 구현될 수 있다. 서비스 인에이블링 기능(204 및 304), 키 전달 기능(206), 신뢰된 제3자들, CSE(402, 502, 504, 604, 704 및 2002), CSF(408 및 412), AE1(602), AE2(1102), SP 레포지토리(2004), DP/EP 레포지토리(2006), MEF(2008), 및 사용자 인터페이스(1202)를 생성하기 위한 로지컬 엔티티들과 같은 로지컬 엔티티들은 컴퓨터 판독가능 스토리지 매체 상에 저장된 컴퓨터 실행가능 명령어들의 형태로 구현될 수 있다. 컴퓨터 판독가능 스토리지 매체는 정보 스토리지를 위한 임의의 비일시적(즉, 유형 또는 물리적) 방법 또는 기술로 구현된 휘발성 및 비휘발성, 이동식 및 비이동식 매체 모두를 포함하지만, 그러한 컴퓨터 판독가능 스토리지 매체는 신호들을 포함하지 않는다. 컴퓨터 판독가능 스토리지 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 다른 광학 디스크 스토리지, 자기 카세트, 자기 테이프, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 유형 또는 물리적 매체를 포함하지만, 이에 제한되지 않는다.

[0369] 도면들에 도시된 바와 같이, 본 개시내용의 대상의 바람직한 실시예들을 설명함에 있어서, 명확한 설명을 위해 특정 용어가 채택되었다. 그러나, 청구되는 대상은 그렇게 선택된 특정 용어에 한정되도록 의도되지 않으며, 각각의 특정 엘리먼트가 유사한 목적을 달성하기 위해 유사한 방식으로 동작하는 모든 기술적 등가물들을 포함한다는 것이 이해될 것이다.

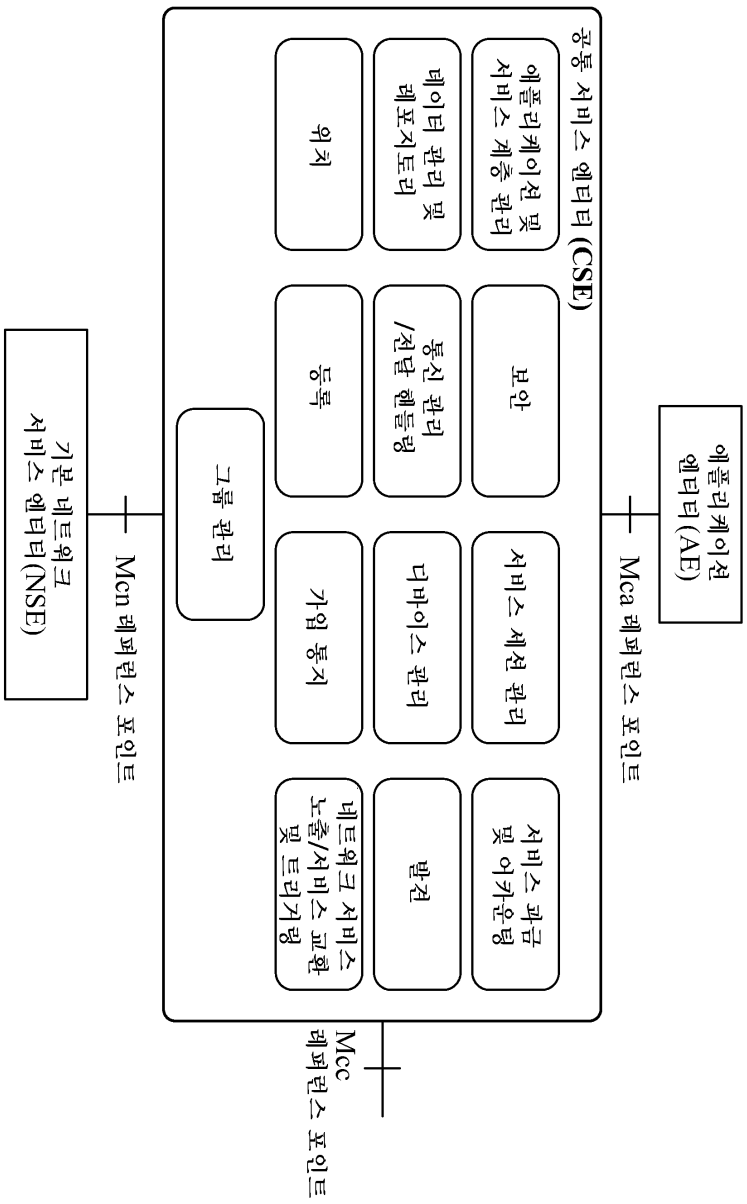
[0370] 이 서술된 설명은 최상의 모드를 포함하여 본 발명을 개시하기 위해 예들을 사용하고, 또한 본 기술분야의 통상의 기술자가 임의의 디바이스들 또는 시스템들을 제작 및 사용하고 임의의 통합된 방법들을 수행하는 것을 포함하여 본 발명을 실시할 수 있게 한다. 본 발명의 특허 가능한 범위는 청구 범위에 의해 정의되며, 본 기술분야의 통상의 기술자에게 발생하는 다른 예들을 포함할 수 있다. 이러한 다른 예들은, 청구 범위의 문자 언어와 상이하지 않은 엘리먼트들을 갖는 경우, 또는 청구 범위의 문자 언어와 실질적인 차이가 없는 등가의 엘리먼트들을 포함하는 경우, 청구항의 범위 내에 있는 것으로 의도된다.

도면

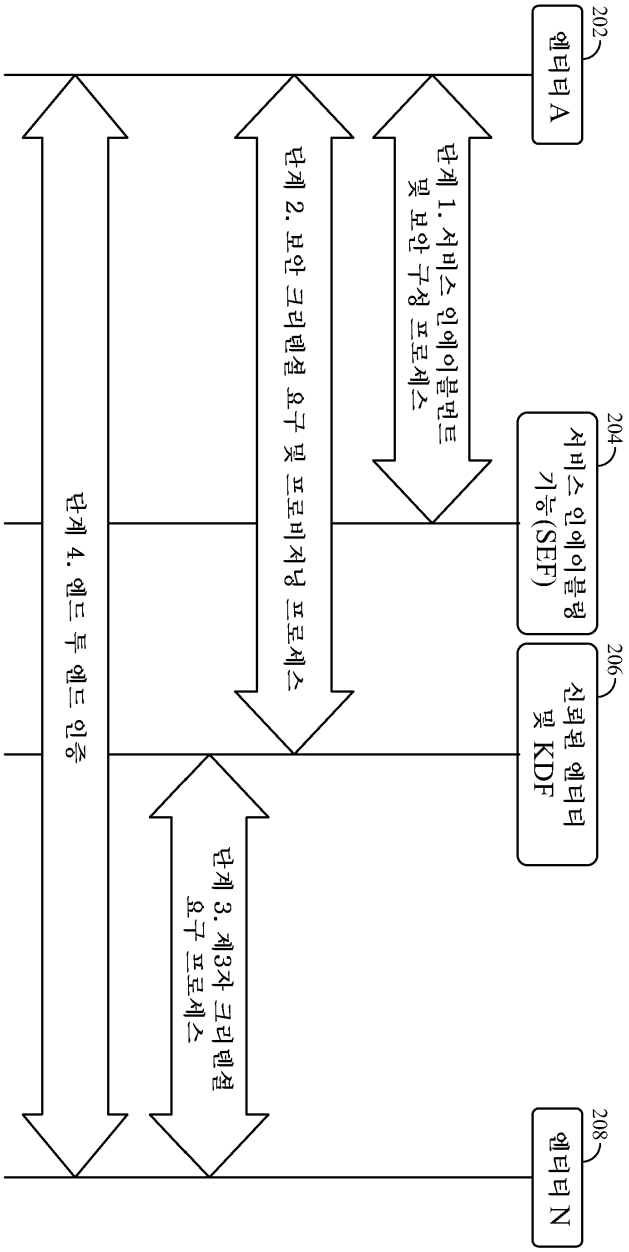
도면1a


100
필드 도메인
NSE
Mcn
CSE
Mca
AE
Mcc
Mca
인프라스트럭처 도메인
NSE
Mcn
CSE
Mca
AE
Mcc'
다른 서비스 제공자의
인프라스트럭처
도메인으로

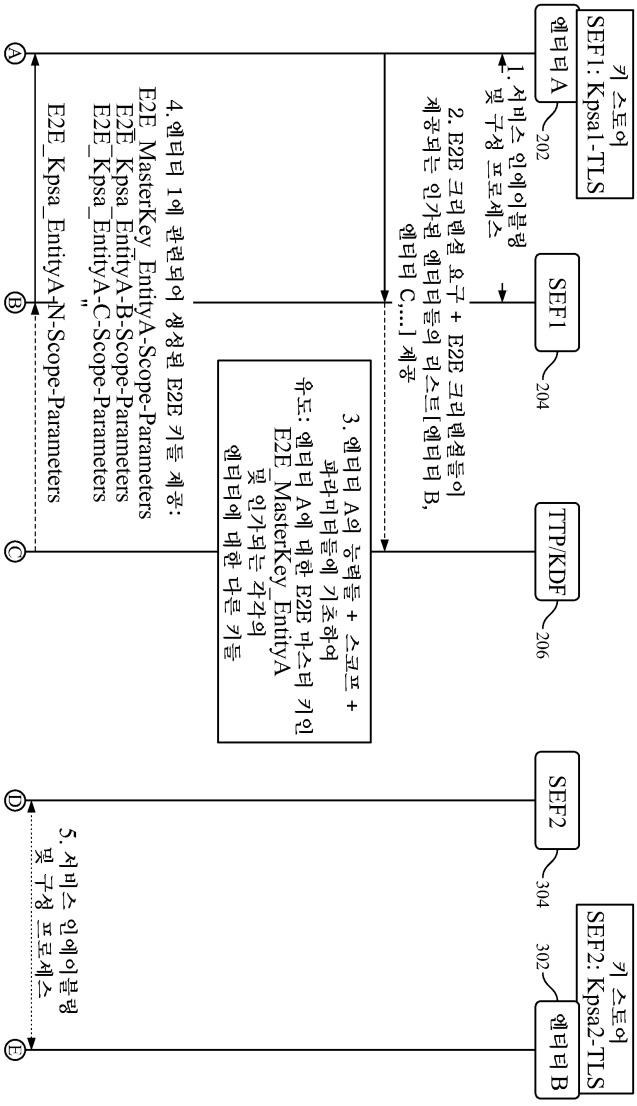
도면1b


공통 서비스 엔티티 (CSE)
에플리케이션 및 서비스 계층 관리
데이터 관리 및 레포지토리
위치
보안
통신 관리 /전달 핸들링
등록
그룹 관리
서비스 세션 관리
디바이스 관리
가입 통지
서비스 과금 및 어카운팅
발견
네트워크 서비스 노출/서비스 교환 및 트리거링
애플리케이션 엔티티 (AE)
Mca 레퍼런스 포인트
기본 네트워크 서비스 엔티티 (NSE)
Mcn 레퍼런스 포인트
Mcc 레퍼런스 포인트

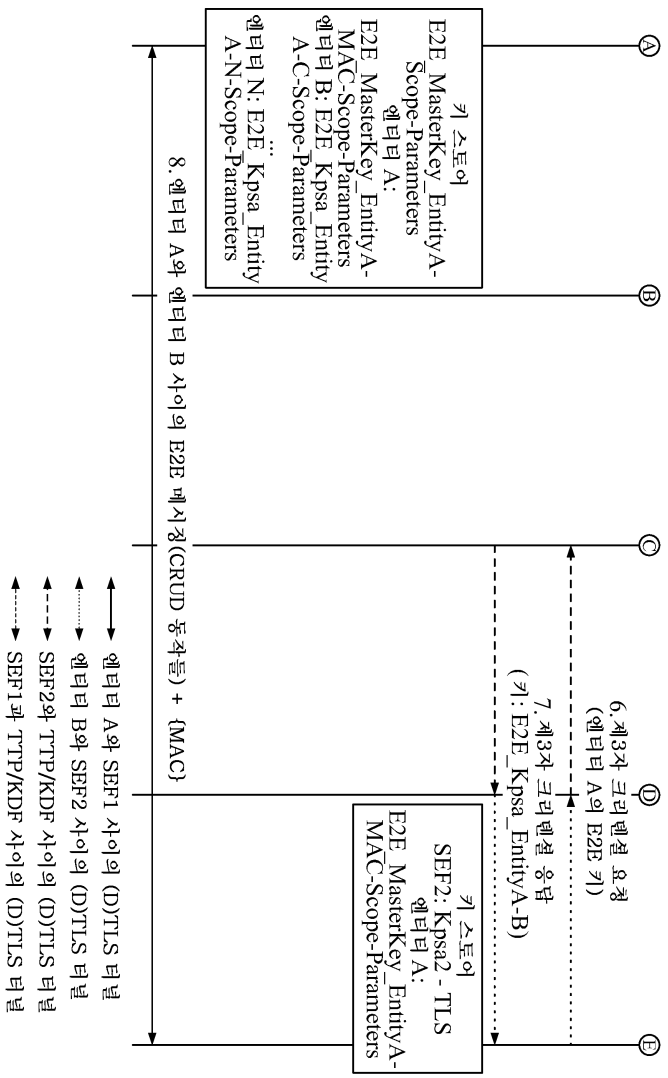
도면2


202
엔터티 A
204
서비스 인에이블링 기능(SEF)
206
신뢰된 엔터티 및 KDF
208
엔터티 N
단계 1. 서비스 인에이블먼트 및 보안 구성 프로세스
단계 2. 보안 크리덴셜 요구 및 프로비저닝 프로세스
단계 3. 제3자 크리덴셜 요구 프로세스
단계 4. 엔드 투 엔드 인증

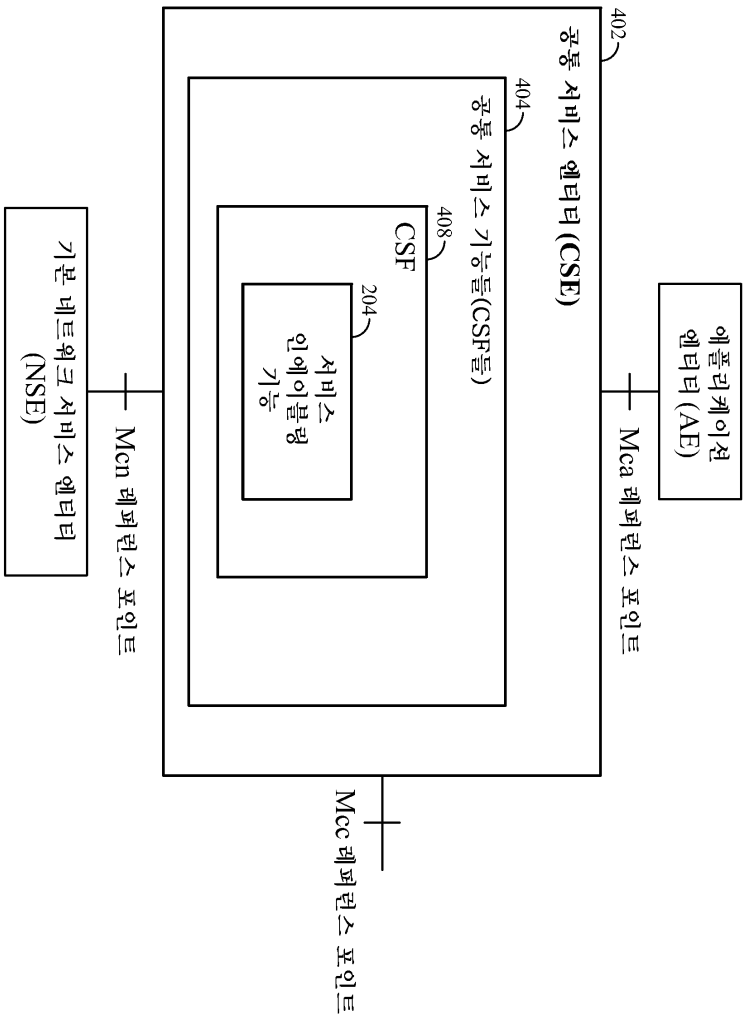
도면3a


키 스토어
SEF1: Kpsa1-TLS
엔터티 A
202
1. 서비스 인에이블링
및 구성 프로세스
SEF1
204
2. E2E 크리덴셜 요구 + E2E 크리덴셜들이
제공되는 인가된 엔터티들의 리스트[엔터티 B,
엔터티 C,...] 제공
TTP/KDF
206
3. 엔터티 A의 능력들 + 스코프 +
파라미터들에 기초하여
유도: 엔터티 A에 대한 E2E 마스터 키인
E2E_MasterKey_EntityA
및 인가되는 각각의
엔터티에 대한 다른 키들
SEF2
304
키 스토어
SEF2: Kpsa2-TLS
302
엔터티 B
4. 엔터티 1에 관련되어 생성된 E2E 키들 제공:
E2E_MasterKey_EntityA-Scope-Parameters
E2E_Kpsa_EntityA-B-Scope-Parameters
E2E_Kpsa_EntityA-C-Scope-Parameters
"
E2E_Kpsa_EntityA-N-Scope-Parameters
5. 서비스 인에이블링
및 구성 프로세스
A
B
C
D
E

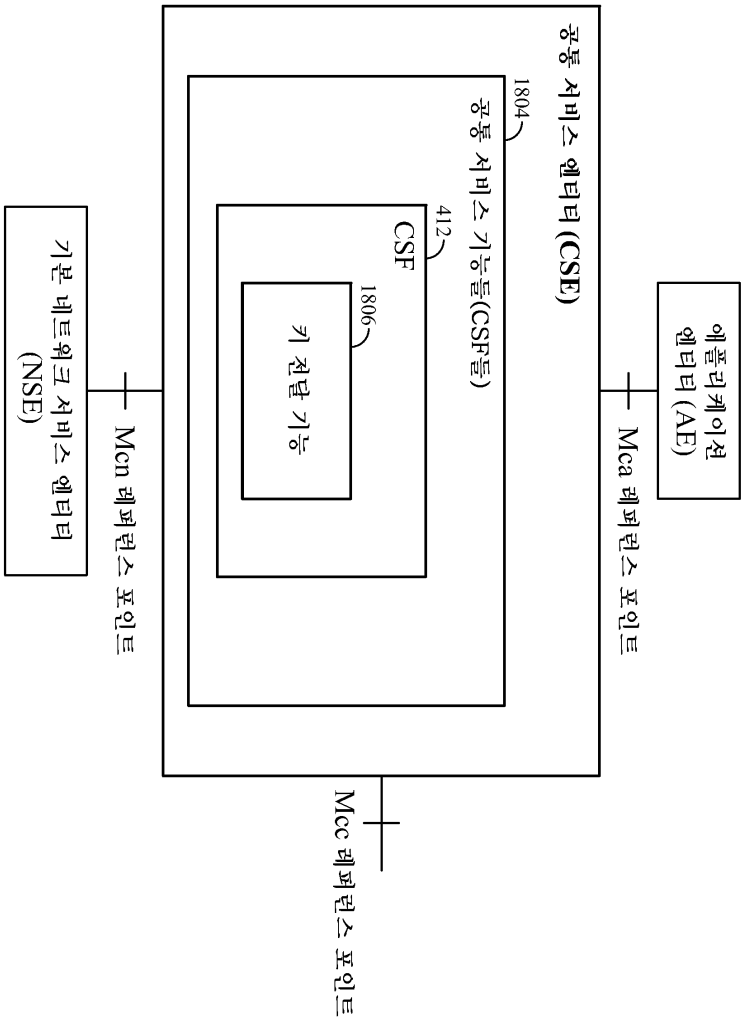
도면3b


Ⓐ
Ⓑ
Ⓒ
Ⓓ
Ⓔ
키 스토어
E2E_MasterKey_EntityA-Scope-Parameters
엔터티 A:
E2E_MasterKey_EntityA-MAC-Scope-Parameters
엔터티 B: E2E_Kpsa_Entity A-C-Scope-Parameters
엔터티 N: E2E_Kpsa_Entity A-N-Scope-Parameters
8. 엔터티 A와 엔터티 B 사이의 E2E 메시징(CRUD 동작들) + {MAC}
6. 제3자 크리덴셜 요청 (엔터티 A의 E2E 키)
7. 제3자 크리덴셜 응답 (키: E2E_Kpsa_EntityA-B)
키 스토어
SEF2: Kpsa2 - TLS
엔터티 A:
E2E_MasterKey_EntityA-MAC-Scope-Parameters
엔터티 A와 SEF1 사이의 (D)TLS 터널
엔터티 B와 SEF2 사이의 (D)TLS 터널
SEF2와 TTP/KDF 사이의 (D)TLS 터널
SEF1과 TTP/KDF 사이의 (D)TLS 터널

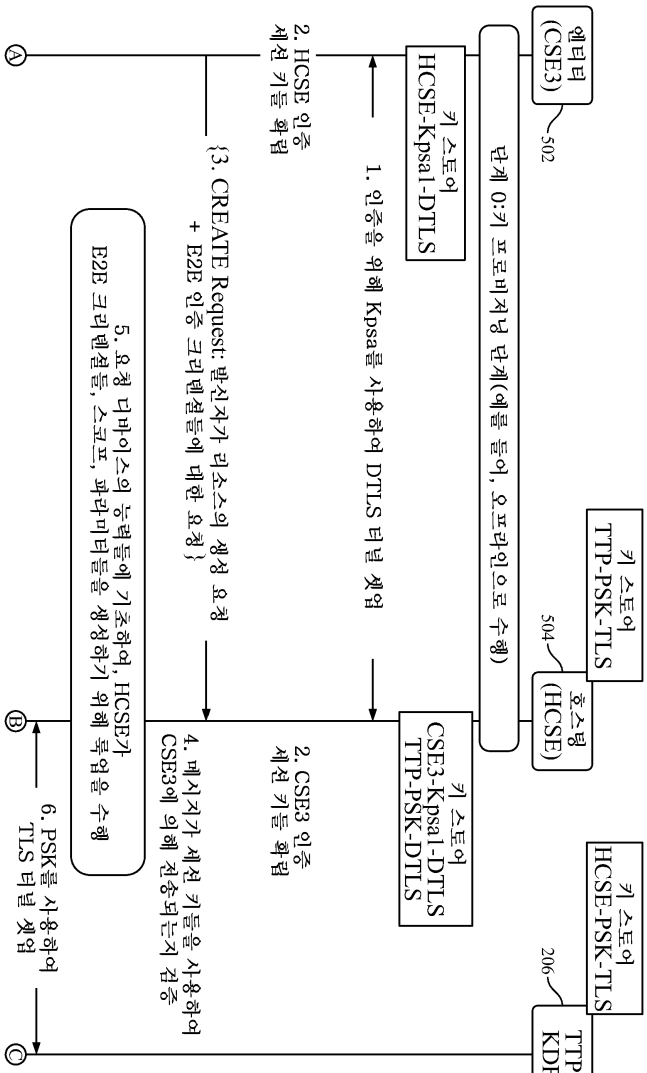
도면4a


402
공통 서비스 엔터티 (CSE)
404
공통 서비스 기능들(CSF들)
408
CSF
204
서비스
인에이블링
기능
에플리케이션
엔터티 (AE)
Mca 레퍼런스 포인트
Mcn 레퍼런스 포인트
기본 네트워크 서비스 엔터티
(NSE)
Mcc 레퍼런스 포인트

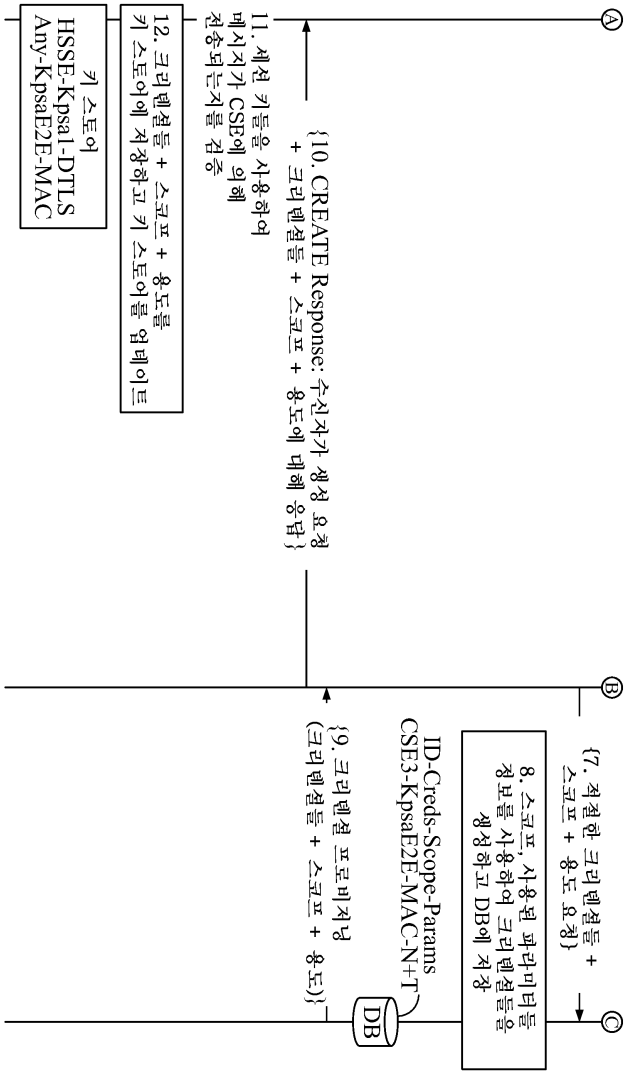
도면4b


공통 서비스 엔티티 (CSE)
1804
공통 서비스 기능들(CSF들)
412
CSF
1806
키 전달 기능
애플리케이션 엔티티 (AE)
Mca 레퍼런스 포인트
Mcn 레퍼런스 포인트
기본 네트워크 서비스 엔티티 (NSE)
Mcc 레퍼런스 포인트

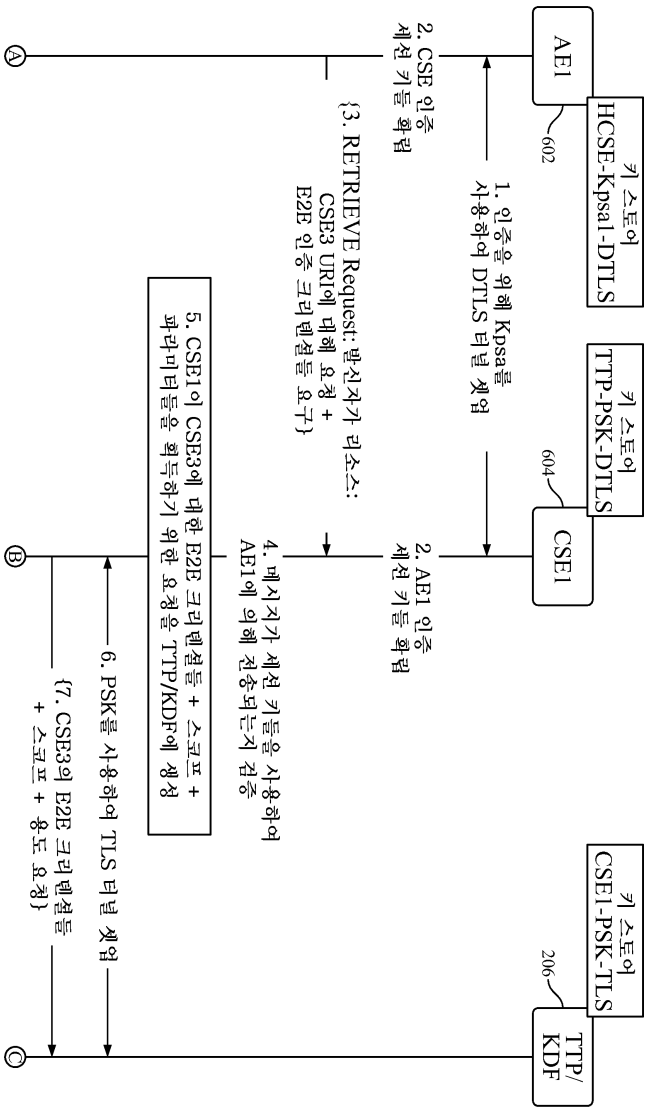
도면5a


엔터티
(CSE3)
502
키 스토어
HCSE-Kpsa1-DTLS
키 스토어
TTP-PSK-TLS
504
호스텡
(HCSE)
키 스토어
HCSE-PSK-TLS
206
TTP/
KDF
단계 0:키 프로비저닝 단계(예를 들어, 오프라인으로 수행)
키 스토어
CSE3-Kpsa1-DTLS
TTP-PSK-DTLS
1. 인증을 위해 Kpsa를 사용하여 DTLS 터널 셋업
2. HCSE 인증
세션 키들 확립
2. CSE3 인증
세션 키들 확립
{3. CREATE Request: 발신자가 리소스의 생성 요청
+ E2E 인증 크리덴셜들에 대한 요청}
4. 메시지가 세션 키들을 사용하여
CSE3에 의해 전송되는지 검증
5. 요청 디바이스의 능력들에 기초하여, HCSE가
E2E 크리덴셜들, 스코프, 파라미터들을 생성하기 위해 룩업을 수행
6. PSK를 사용하여
TLS 터널 셋업
Ⓐ
Ⓑ
Ⓒ

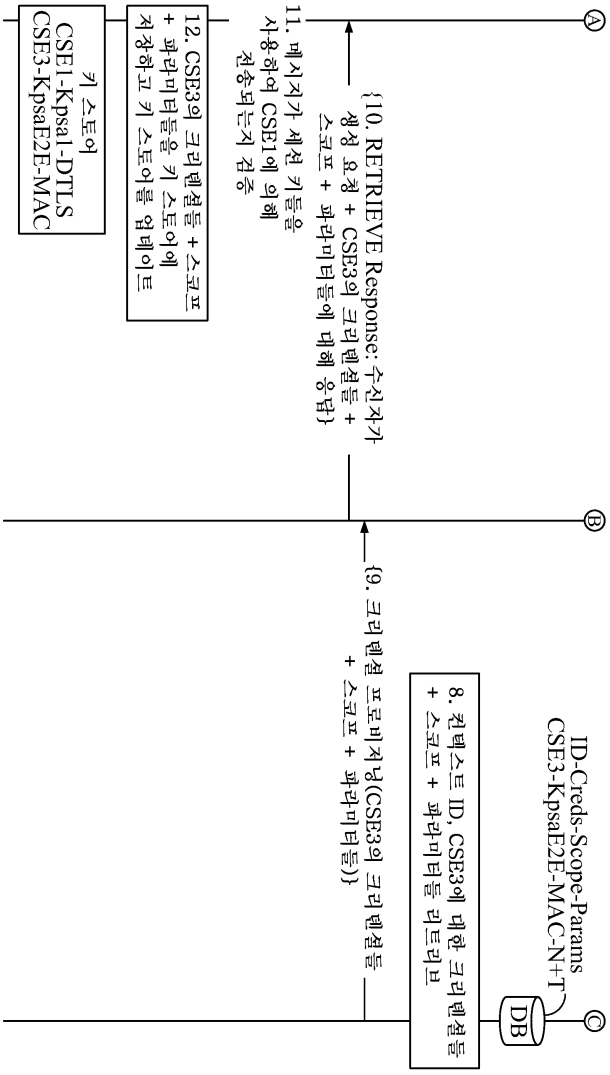
도면5b


Ⓐ
{10. CREATE Response: 수신자가 생성 요청
+ 크리덴셜들 + 스코프 + 용도에 대해 응답}
11. 세션 키들을 사용하여
메시지가 CSE에 의해
전송되는지를 검증
12. 크리덴셜들 + 스코프 + 용도를
키 스토어에 저장하고 키 스토어를 업데이트
키 스토어
HSSE-Kpsa1-DTLS
Any-KpsaE2E-MAC
Ⓑ
{9. 크리덴셜 프로비저닝
(크리덴셜들 + 스코프 + 용도)}
ID-Creds-Scope-Params
CSE3-KpsaE2E-MAC-N+T
DB
8. 스코프, 사용된 파라미터들
정보를 사용하여 크리덴셜들을
생성하고 DB에 저장
{7. 적절한 크리덴셜들 +
스코프 + 용도 요청}
Ⓒ

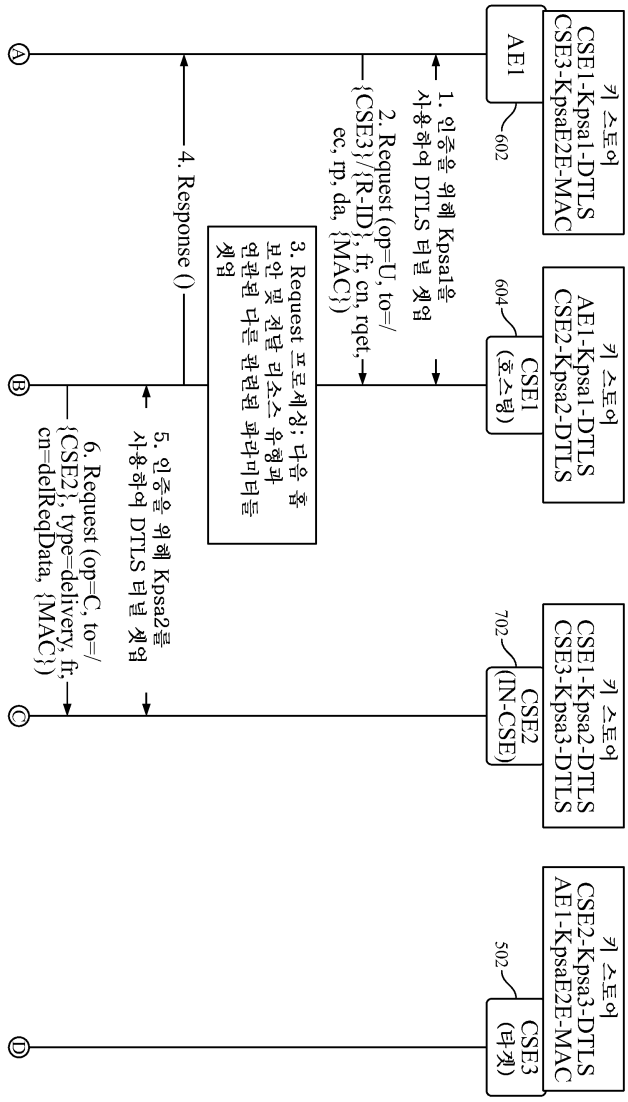
도면6a


AE1
602
키 스토어
HCSE-Kpsa1-DTLS
키 스토어
TTP-PSK-DTLS
604
CSE1
키 스토어
CSE1-PSK-TLS
206
TTP/
KDF
1. 인증을 위해 Kpsa를
사용하여 DTLS 터널 셋업
2. AE1 인증
세션 키들 확립
2. CSE 인증
세션 키들 확립
{3. RETRIEVE Request: 발신자가 리소스:
CSE3 URI에 대해 요청 +
E2E 인증 크리덴셜들 요구}
4. 메시지가 세션 키들을 사용하여
AE1에 의해 전송되는지 검증
5. CSE1이 CSE3에 대한 E2E 크리덴셜들 + 스코프 +
파라미터들을 획득하기 위한 요청을 TTP/KDF에 생성
6. PSK를 사용하여 TLS 터널 셋업
{7. CSE3의 E2E 크리덴셜들
+ 스코프 + 용도 요청}
Ⓐ
Ⓑ
Ⓒ

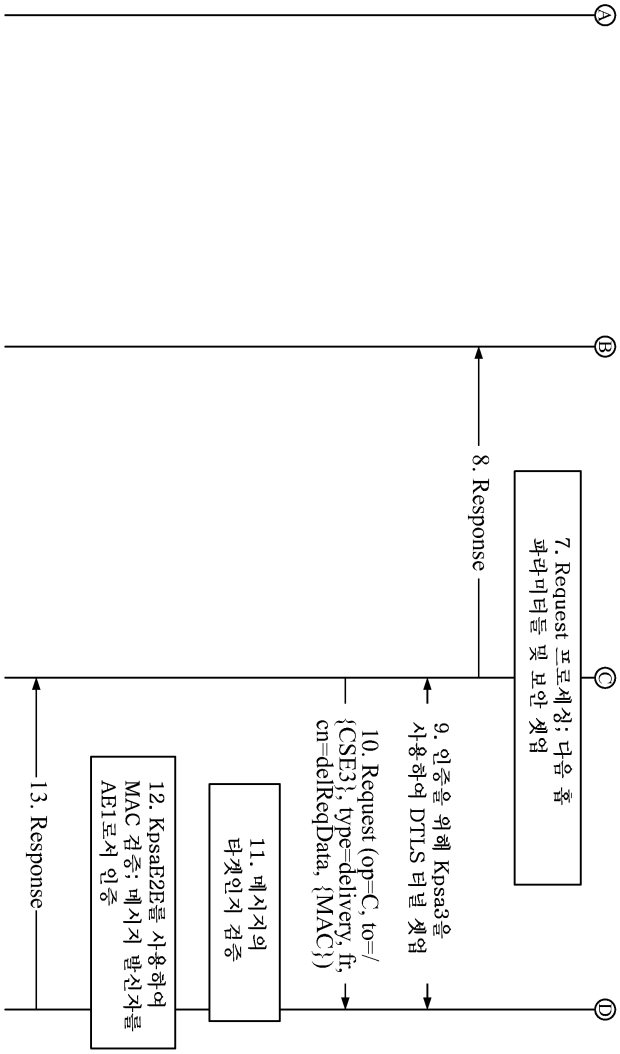
도면6b


Ⓐ
Ⓑ
Ⓒ
ID-Creds-Scope-Params
CSE3-KpsaE2E-MAC-N+T
DB
8. 컨텍스트 ID, CSE3에 대한 크리덴셜들
+ 스코프 + 파라미터들 리트리브
{9. 크리덴셜 프로비저닝(CSE3의 크리덴셜들
+ 스코프 + 파라미터들)}
{10. RETRIEVE Response: 수신자가
생성 요청 + CSE3의 크리덴셜들 +
스코프 + 파라미터들에 대해 응답}
11. 메시지가 세션 키들을
사용하여 CSE1에 의해
전송되는지 검증
12. CSE3의 크리덴셜들 + 스코프
+ 파라미터들을 키 스토어에
저장하고 키 스토어를 업데이트
키 스토어
CSE1-Kpsa1-DTLS
CSE3-KpsaE2E-MAC

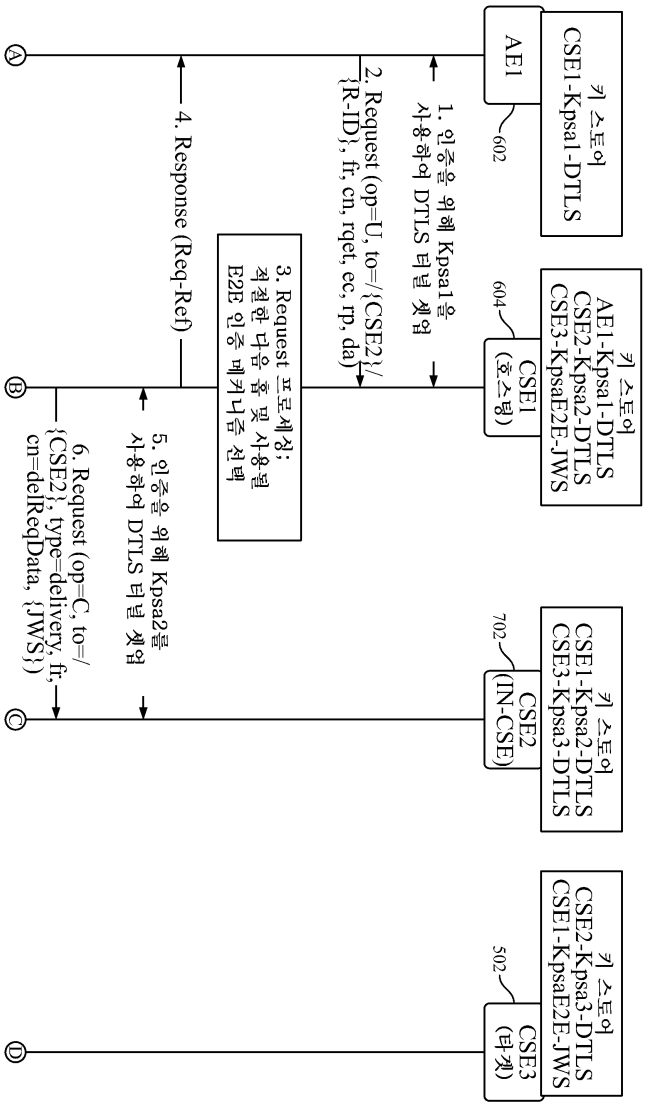
도면7a


키 스토어
CSE1-Kpsa1-DTLS
CSE3-KpsaE2E-MAC
AE1
602
키 스토어
AE1-Kpsa1-DTLS
CSE2-Kpsa2-DTLS
604
CSE1
(호스팅)
키 스토어
CSE1-Kpsa2-DTLS
CSE3-Kpsa3-DTLS
702
CSE2
(IN-CSE)
키 스토어
CSE2-Kpsa3-DTLS
AE1-KpsaE2E-MAC
502
CSE3
(타겟)
Ⓐ
Ⓑ
Ⓒ
Ⓓ
1. 인증을 위해 Kpsa1을
사용하여 DTLS 터널 셋업
2. Request (op=U, to=/
{CSE3}/{R-ID}, fr, cn, rqet,
ec, rp, da, {MAC})
3. Request 프로세싱; 다음 홉
보안 및 전달 리소스 유형과
연관된 다른 관련된 파라미터들
셋업
4. Response ()
5. 인증을 위해 Kpsa2를
사용하여 DTLS 터널 셋업
6. Request (op=C, to=/
{CSE2}, type=delivery, fr,
cn=delReqData, {MAC})

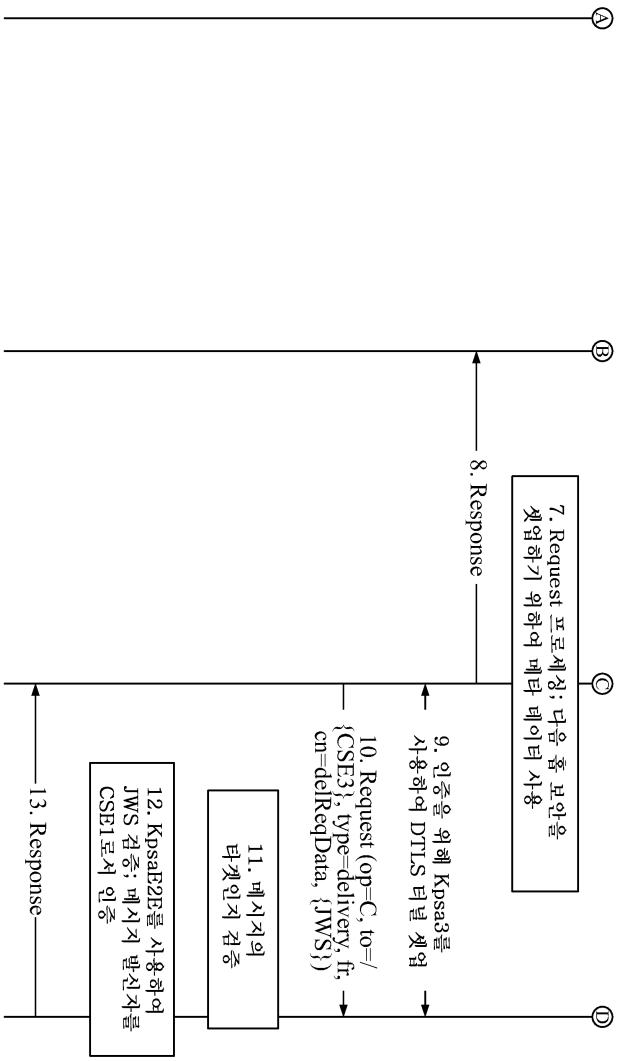
도면7b


Ⓐ
Ⓑ
Ⓒ
Ⓓ
7. Request 프로세싱; 다음 홉 파라미터들 및 보안 셋업
8. Response
9. 인증을 위해 Kpsa3을 사용하여 DTLS 터널 셋업
10. Request (op=C, to=/{CSE3}, type=delivery, fr, cn=delReqData, {MAC})
11. 메시지의 타겟인지 검증
12. KpsaE2E를 사용하여 MAC 검증; 메시지 발신자를 AE1로서 인증
13. Response

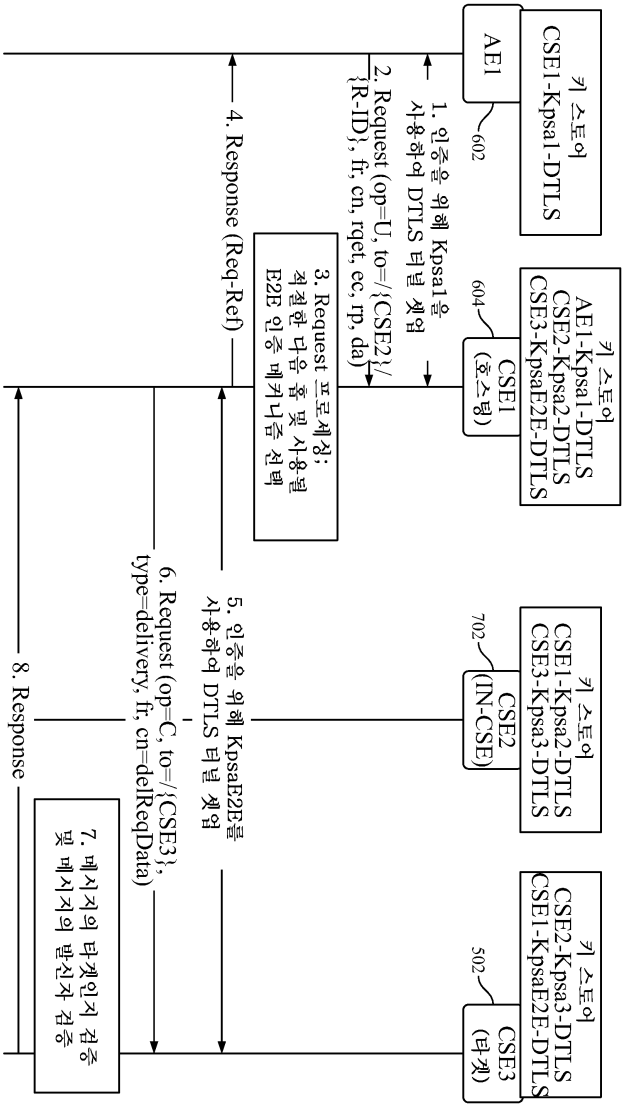
도면8a


키 스토어
CSE1-Kpsa1-DTLS
AE1
602
키 스토어
AE1-Kpsa1-DTLS
CSE2-Kpsa2-DTLS
CSE3-KpsaE2E-JWS
604
CSE1
(호스팅)
1. 인증을 위해 Kpsa1을
사용하여 DTLS 터널 셋업
2. Request (op=U, to=/{CSE2}/
{R-ID}, fr, cn, rqet, ec, rp, da)
3. Request 프로세싱;
적절한 다음 홉 및 사용될
E2E 인증 메커니즘 선택
4. Response (Req-Ref)
5. 인증을 위해 Kpsa2를
사용하여 DTLS 터널 셋업
6. Request (op=C, to=/
{CSE2}, type=delivery, fr,
cn=delReqData, {JWS})
키 스토어
CSE1-Kpsa2-DTLS
CSE3-Kpsa3-DTLS
702
CSE2
(IN-CSE)
키 스토어
CSE2-Kpsa3-DTLS
CSE1-KpsaE2E-JWS
502
CSE3
(타겟)
Ⓐ
Ⓑ
Ⓒ
Ⓓ

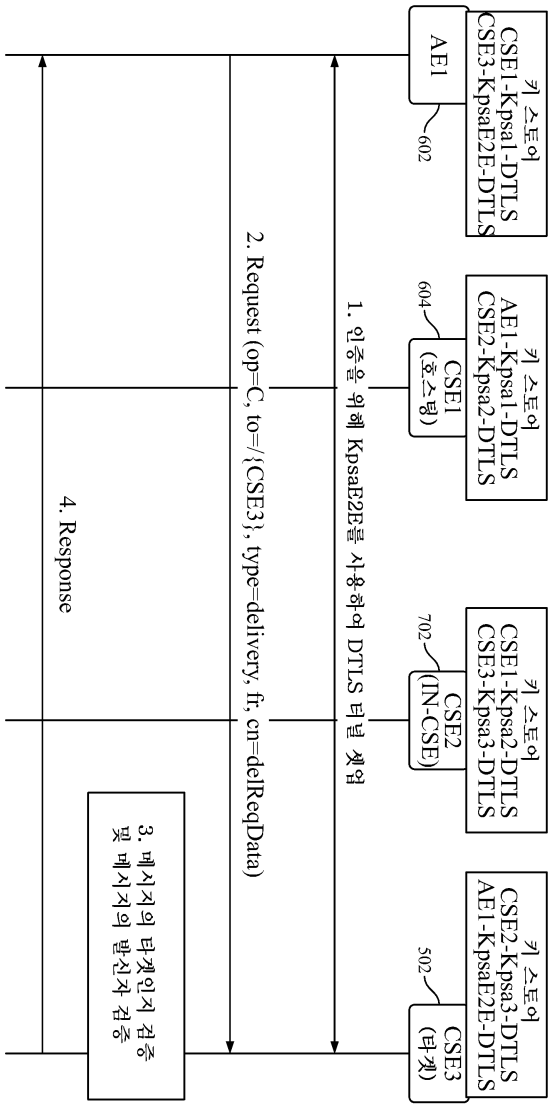
도면8b


Ⓐ
Ⓑ
Ⓒ
Ⓓ
7. Request 프로세싱; 다음 홉 보안을 셋업하기 위하여 메타 데이터 사용
8. Response
9. 인증을 위해 Kpsa3를 사용하여 DTLS 터널 셋업
10. Request (op=C, to=/{CSE3}, type=delivery, fr, cn=delReqData, {JWS})
11. 메시지의 타겟인지 검증
12. KpsaE2E를 사용하여 JWS 검증; 메시지 발신자를 CSE1로서 인증
13. Response

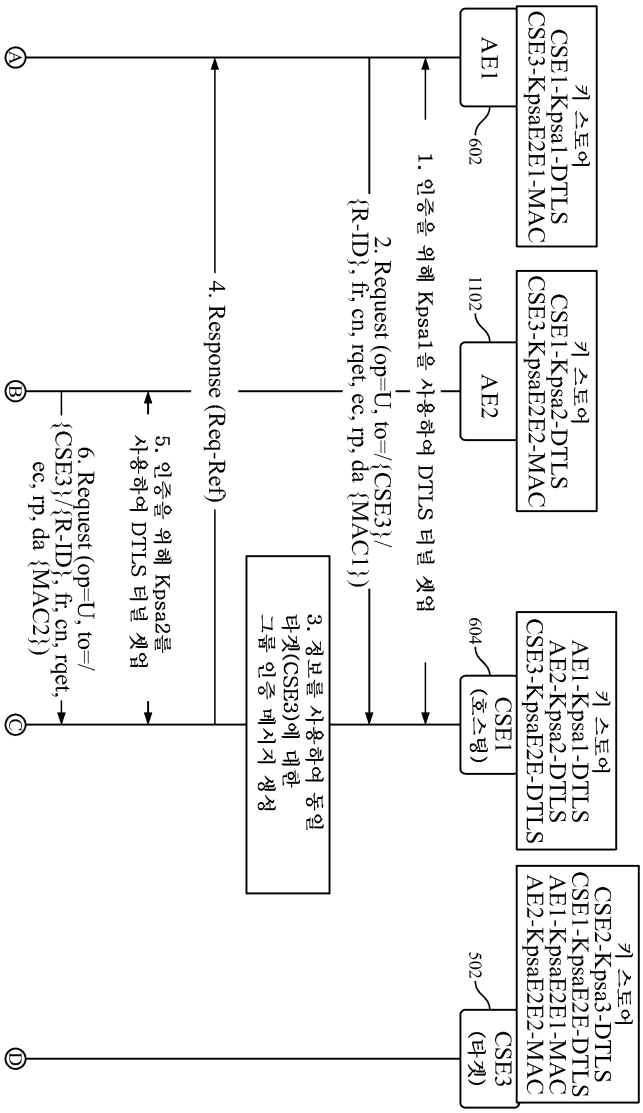
도면9


키 스토어
CSE1-Kpsa1-DTLS
AE1
602
키 스토어
AE1-Kpsa1-DTLS
CSE2-Kpsa2-DTLS
CSE3-KpsaE2E-DTLS
604
CSE1
(호스팅)
1. 인증을 위해 Kpsa1을
사용하여 DTLS 터널 셋업
2. Request (op=U, to=/{CSE2}/
{R-ID}, ft, cn, rqet, ec, rp, da)
3. Request 프로세싱;
적절한 다음 홉 및 사용될
E2E 인증 메커니즘 선택
4. Response (Req-Ref)
키 스토어
CSE1-Kpsa2-DTLS
CSE3-Kpsa3-DTLS
702
CSE2
(IN-CSE)
5. 인증을 위해 KpsaE2E를
사용하여 DTLS 터널 셋업
키 스토어
CSE2-Kpsa3-DTLS
CSE1-KpsaE2E-DTLS
502
CSE3
(타겟)
6. Request (op=C, to=/{CSE3},
type=delivery, ft, cn=delReqData)
7. 메시지의 타겟인지 검증
및 메시지의 발신자 검증
8. Response

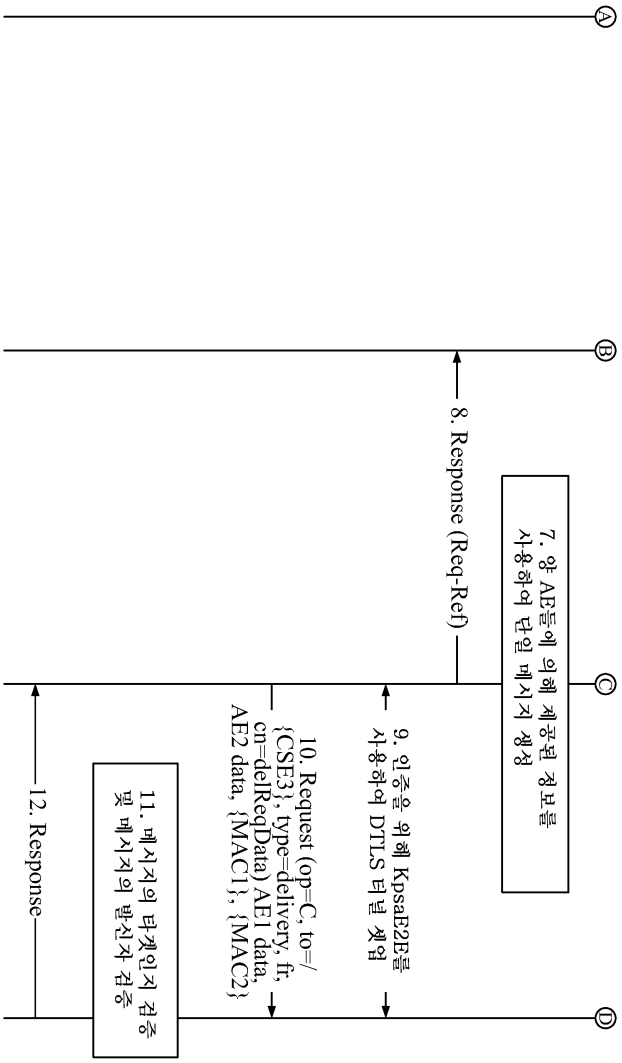
도면10


키 스토어
CSE1-Kpsa1-DTLS
CSE3-KpsaE2E-DTLS
AE1
602
키 스토어
AE1-Kpsa1-DTLS
CSE2-Kpsa2-DTLS
604
CSE1
(호스팅)
키 스토어
CSE1-Kpsa2-DTLS
CSE3-Kpsa3-DTLS
702
CSE2
(IN-CSE)
키 스토어
CSE2-Kpsa3-DTLS
AE1-KpsaE2E-DTLS
502
CSE3
(타겟)
1. 인증을 위해 KpsaE2E를 사용하여 DTLS 터널 셋업
2. Request (op=C, to=/{CSE3}, type=delivery, fr, cn=delReqData)
3. 메시지의 타겟인지 검증
및 메시지의 발신자 검증
4. Response

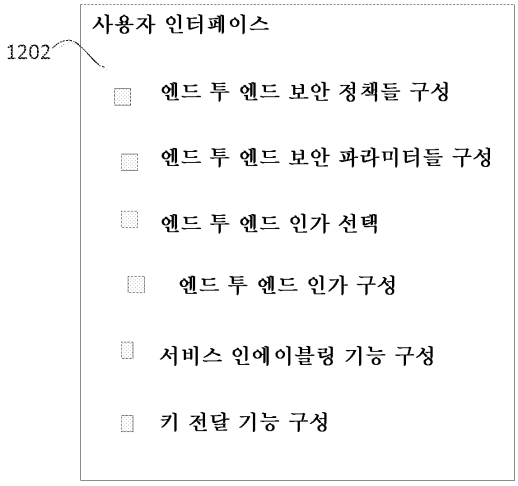
도면11a


키 스토어
CSE1-Kpsa1-DTLS
CSE3-KpsaE2E1-MAC
AE1
602
키 스토어
CSE1-Kpsa2-DTLS
CSE3-KpsaE2E2-MAC
1102
AE2
키 스토어
AE1-Kpsa1-DTLS
AE2-Kpsa2-DTLS
CSE3-KpsaE2E-DTLS
604
CSE1
(호스팅)
키 스토어
CSE2-Kpsa3-DTLS
CSE1-KpsaE2E-DTLS
AE1-KpsaE2E1-MAC
AE2-KpsaE2E2-MAC
502
CSE3
(타겟)
1. 인증을 위해 Kpsa1을 사용하여 DTLS 터널 셋업
2. Request (op=U, to=/{CSE3}/{R-ID}, fr, cn, rqet, ec, rp, da {MAC1})
3. 정보를 사용하여 동일 타겟(CSE3)에 대한 그룹 인증 메시지 생성
4. Response (Req-Ref)
5. 인증을 위해 Kpsa2를 사용하여 DTLS 터널 셋업
6. Request (op=U, to=/{CSE3}/{R-ID}, fr, cn, rqet, ec, rp, da {MAC2})
Ⓐ
Ⓑ
Ⓒ
Ⓓ

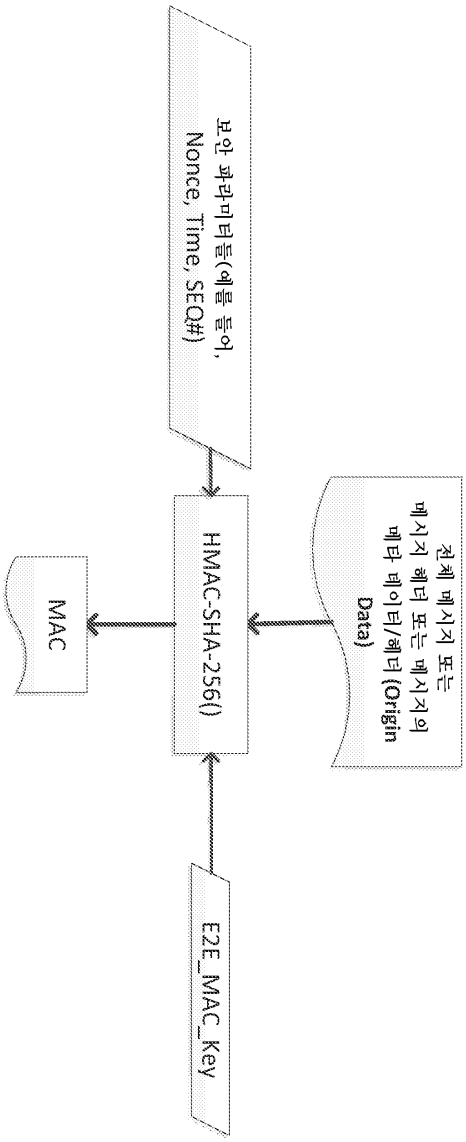
도면11b


Ⓐ
Ⓑ
8. Response (Req-Ref)
7. 양 AE들에 의해 제공된 정보를
사용하여 단일 메시지 생성
Ⓒ
9. 인증을 위해 KpsaE2E를
사용하여 DTLS 터널 셋업
10. Request (op=C, to=/
{CSE3}, type=delivery, fr,
cn=delReqData) AE1 data,
AE2 data, {MAC1}, {MAC2}
11. 메시지의 타겟인지 검증
및 메시지의 발신자 검증
12. Response
Ⓓ

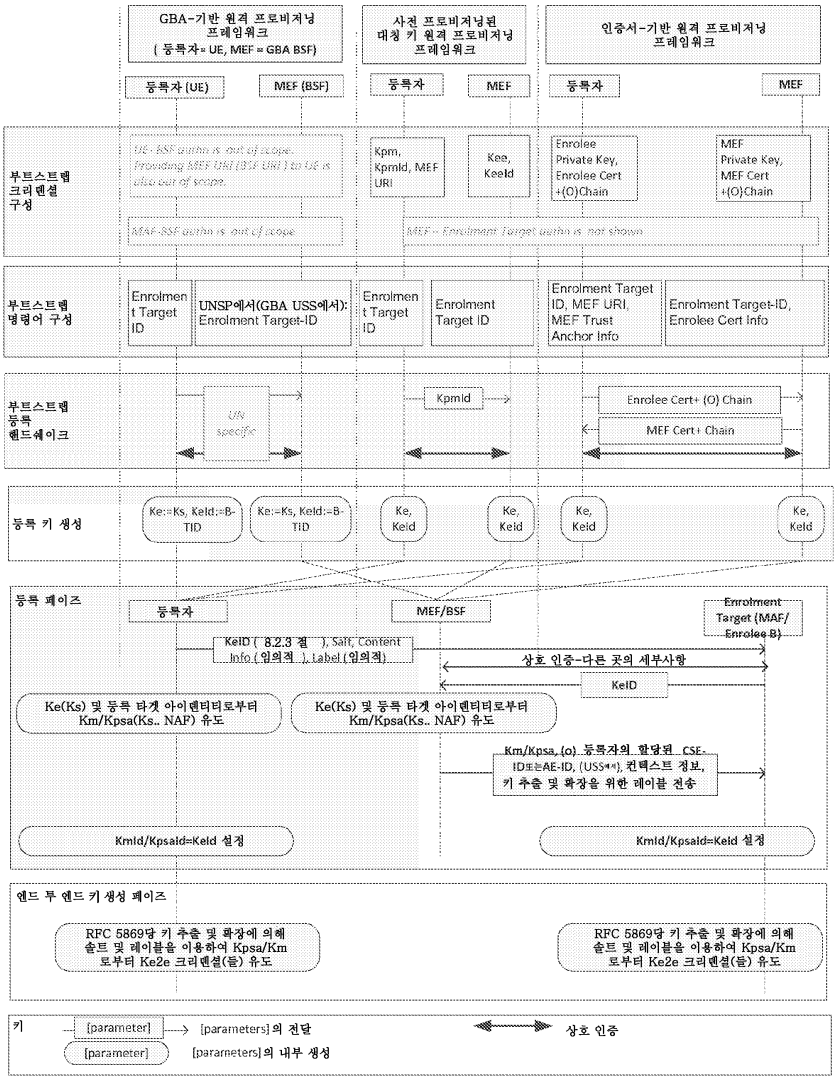
도면12


사용자 인터페이스
1202
엔드 투 엔드 보안 정책들 구성
엔드 투 엔드 보안 파라미터들 구성
엔드 투 엔드 인가 선택
엔드 투 엔드 인가 구성
서비스 인에이블링 기능 구성
키 전달 기능 구성

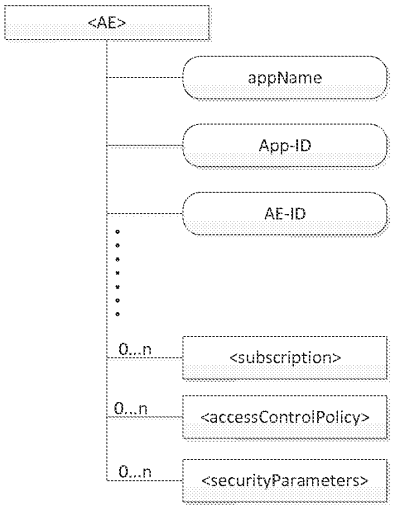
도면13


보안 파라미터들(예를 들어,
Nonce, Time, SEQ#)
전체 메시지 또는
메시지 헤더 또는 메시지의
메타 데이터/헤더 (Origin
Data)
HMAC-SHA-256()
MAC
E2E_MAC_Key

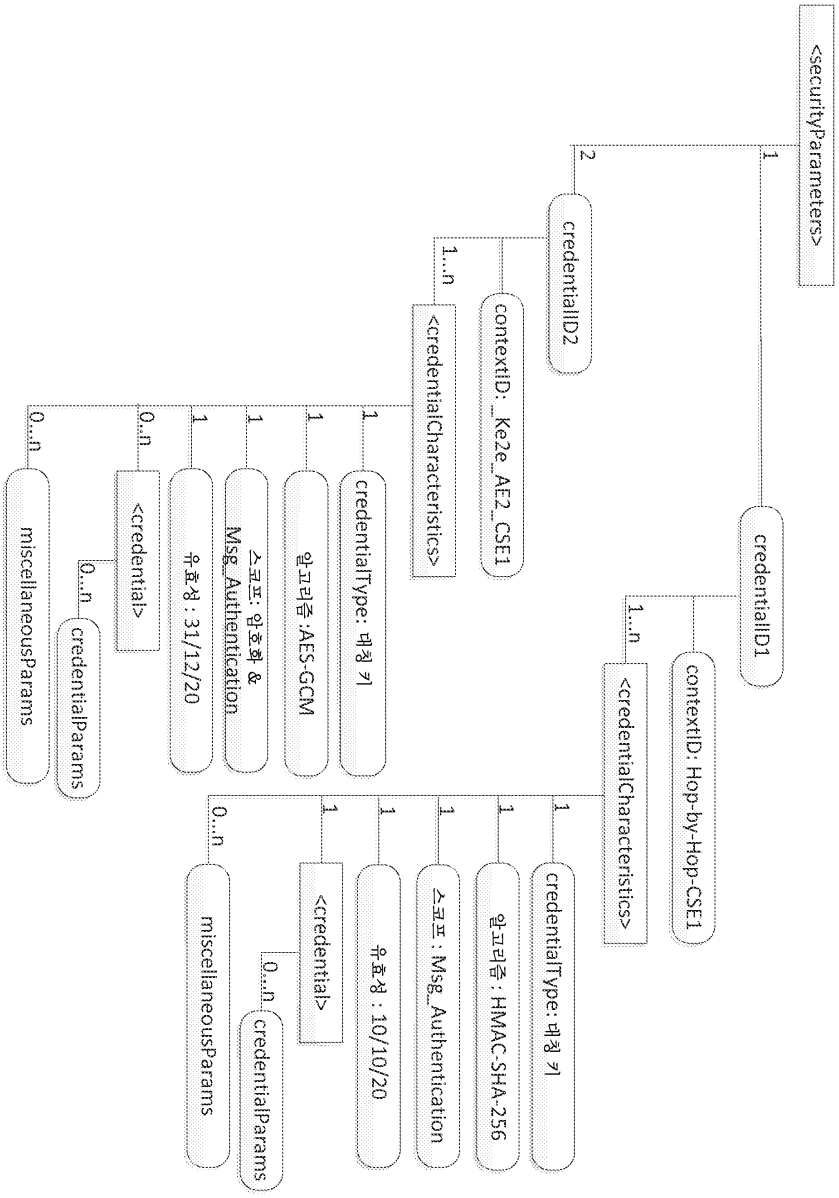
도면14


GBA-기반 원격 프로비저닝 프레임워크 (등록자= UE, MEF = GBA BSF)
사전 프로비저닝된 대칭 키 원격 프로비저닝 프레임워크
인증서-기반 원격 프로비저닝 프레임워크
등록자 (UE)
MEF (BSF)
등록자
MEF
등록자
MEF
부트스트랩 크리덴셜 구성
Kpm, Kpmld, MEF URI
Kee, KeeId
Enrolee Private Key, Enrolee Cert +(O)Chain
MEF Private Key, MEF Cert +(O)Chain
부트스트랩 명령어 구성
Enrolment Target ID
UNSP에서(GBA USS에서): Enrolment Target-ID
Enrolment Target ID
Enrolment Target ID, MEF URI, MEF Trust Anchor Info
Enrolment Target-ID, Enrolee Cert Info
부트스트랩 등록 핸드쉐이크
Kpmld
Enrolee Cert+ (O) Chain
MEF Cert + Chain
등록 키 생성
Ke, KeId
등록 페이즈
등록자
MEF/BSF
Enrolment Target (MAF/ Enrolee B)
KeID (8.2.3 절), Salt, Content Info (임의적), Label (임의적)
상호 인증-다른 곳의 세부사항
KeID
Ke(Ks) 및 등록 타겟 아이덴티티로부터 Km/Kpsa(Ks.. NAF) 유도
Km/Kpsa, (o) 등록자의 할당된 CSE-ID또는AE-ID, (USS에), 컨텍스트 정보, 키 추출 및 확장을 위한 레이블 전송
Kmld/Kpsaid=KeId 설정
엔드 투 엔드 키 생성 페이즈
RFC 5869당 키 추출 및 확장에 의해 솔트 및 레이블을 이용하여 Kpsa/Km 로부터 Ke2e 크리덴셜(들) 유도
키
[parameter]
[parameters]의 전달
[parameter]
[parameters]의 내부 생성
상호 인증

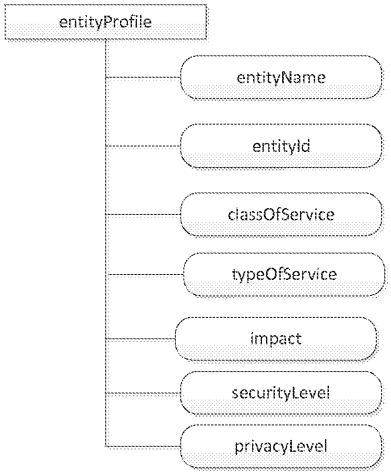
도면15a


<AE>
appName
App-ID
AE-ID
0...n
<subscription>
0...n
<accessControlPolicy>
0...n
<securityParameters>

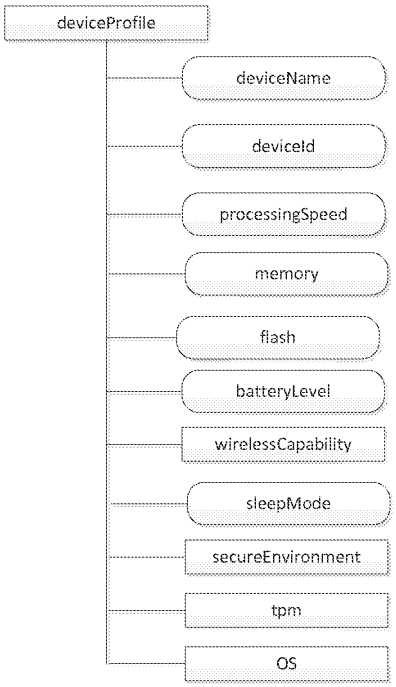
도면15b


<securityParameters>
1
credentialID1
contextID: Hop-by-Hop-CSE1
1...n
<credentialCharacteristics>
1
credentialType: 메칭 키
1
알고리즘 : HMAC-SHA-256
1
스코프 : Msg_Authentication
1
유효성 : 10/10/20
1
<credential>
0...n
credentialParams
0...n
miscellaneousParams
2
credentialID2
contextID: _Ke2e_AE2_CSE1
1...n
<credentialCharacteristics>
1
credentialType: 메칭 키
1
알고리즘 :AES-GCM
1
스코프: 암호화 & Msg_Authentication
1
유효성 : 31/12/20
0..n
<credential>
0...n
credentialParams
0...n
miscellaneousParams

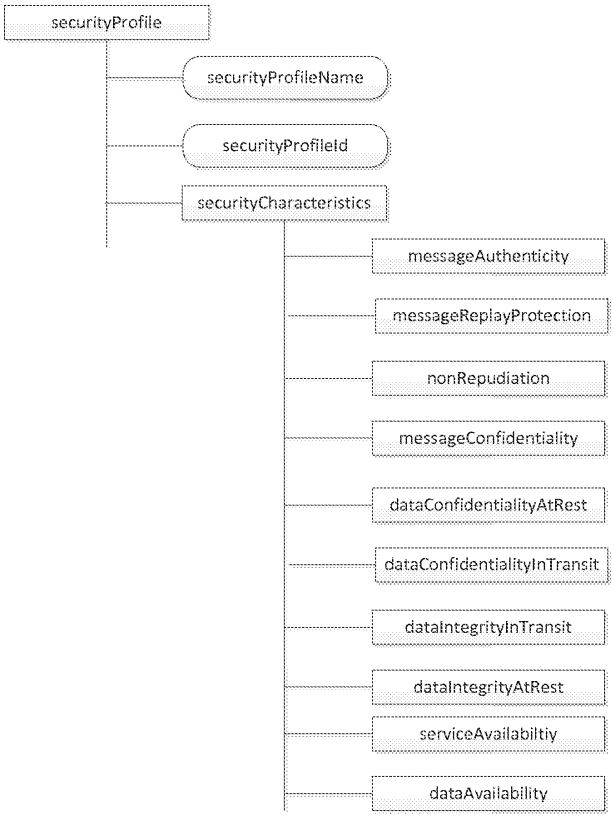
도면16a


entityProfile
entityName
entityId
classOfService
typeOfService
impact
securityLevel
privacyLevel

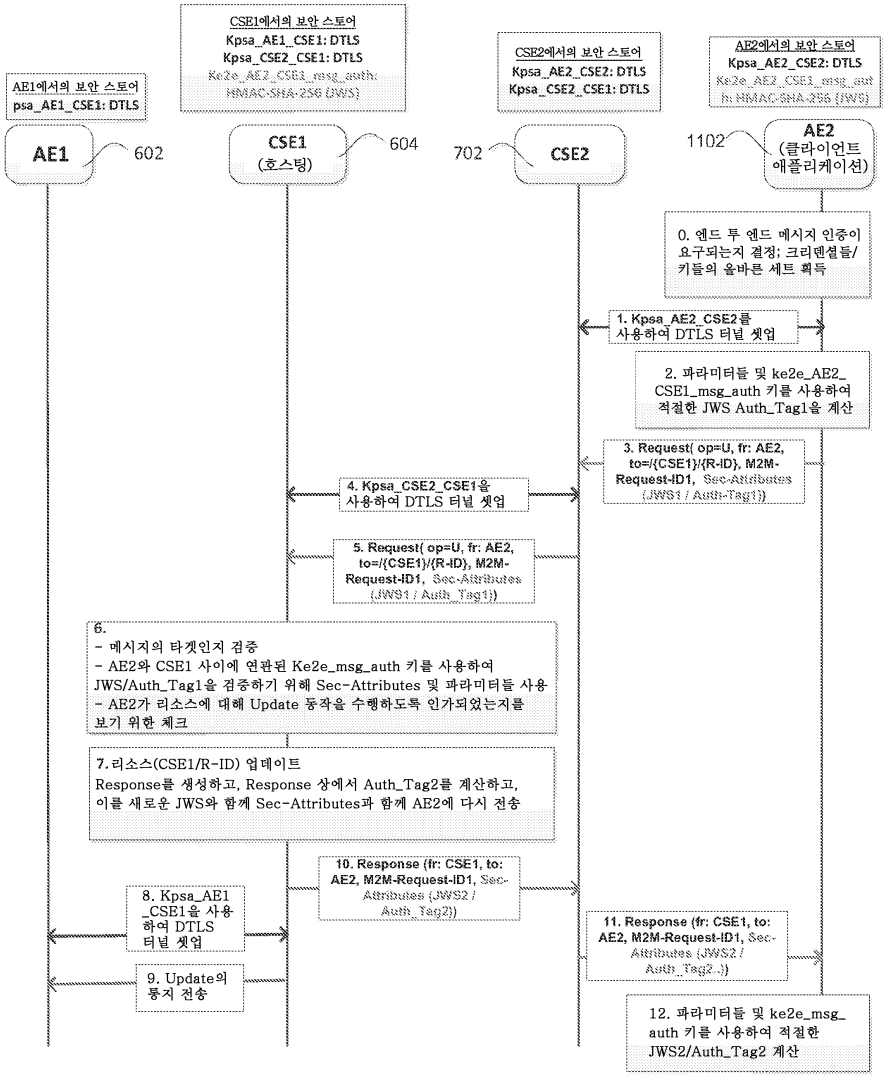
도면16b


deviceProfile
deviceName
deviceId
processingSpeed
memory
flash
batteryLevel
wirelessCapability
sleepMode
secureEnvironment
tpm
OS

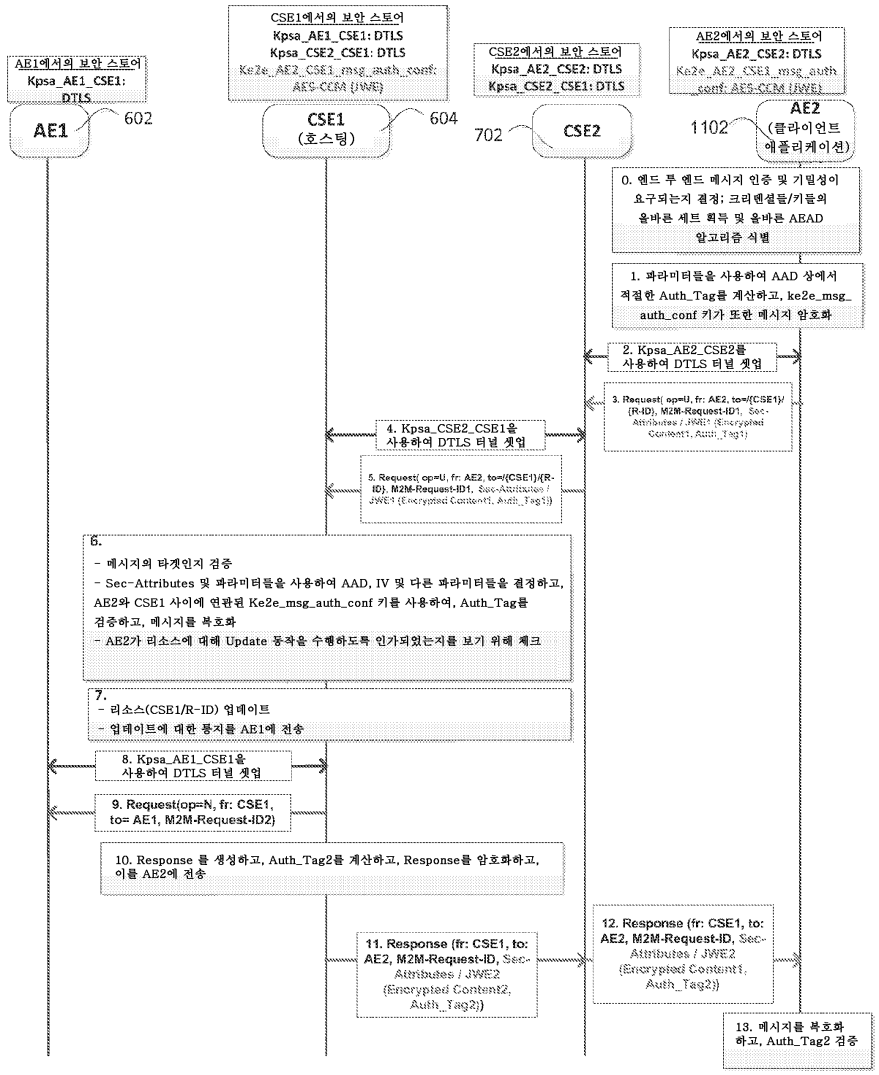
도면16c


securityProfile
securityProfileName
securityProfileId
securityCharacteristics
messageAuthenticity
messageReplayProtection
nonRepudiation
messageConfidentiality
dataConfidentialityAtRest
dataConfidentialityInTransit
dataIntegrityInTransit
dataIntegrityAtRest
serviceAvailabiltiy
dataAvailability

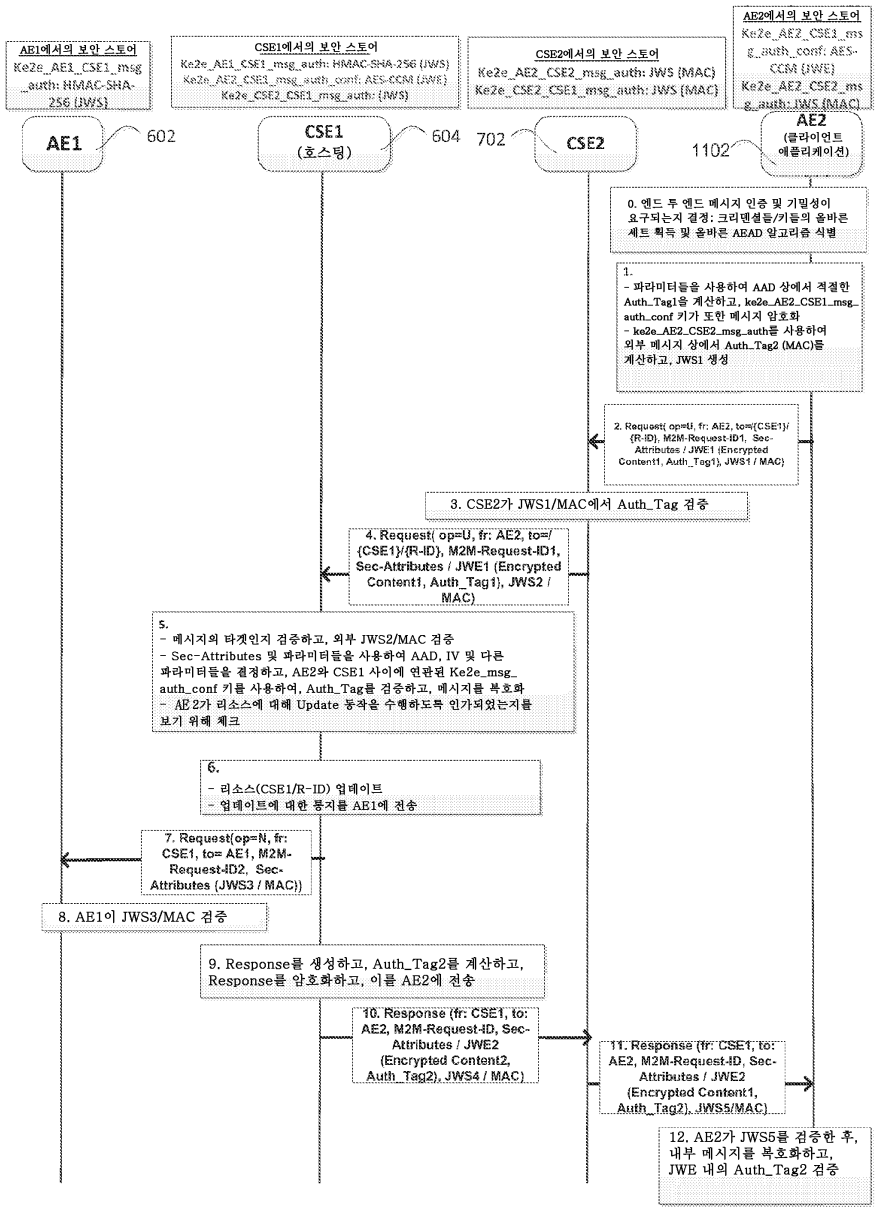
도면17


AE1에서의 보안 스토어
psa_AE1_CSE1: DTLS
CSE1에서의 보안 스토어
Kpsa_AE1_CSE1: DTLS
Kpsa_CSE2_CSE1: DTLS
Ke2e_AE2_CSE1_msg_auth: HMAC-SHA-256 (JWS)
CSE2에서의 보안 스토어
Kpsa_AE2_CSE2: DTLS
Kpsa_CSE2_CSE1: DTLS
AE2에서의 보안 스토어
Kpsa_AE2_CSE2: DTLS
Ke2e_AE2_CSE1_msg_auth: HMAC-SHA-256 (JWS)
AE1
602
CSE1
(호스팅)
604
702
CSE2
1102
AE2
(클라이언트 애플리케이션)
0. 엔드 투 엔드 메시지 인증이 요구되는지 결정; 크리덴셜들/키들의 올바른 세트 획득
1. Kpsa_AE2_CSE2를 사용하여 DTLS 터널 셋업
2. 파라미터들 및 ke2e_AE2_CSE1_msg_auth 키를 사용하여 적절한 JWS Auth_Tag1을 계산
3. Request(op=U, fr: AE2, to=/{CSE1}/{R-ID}, M2M-Request-ID1, Sec-Attributes (JWS1 / Auth-Tag1))
4. Kpsa_CSE2_CSE1을 사용하여 DTLS 터널 셋업
5. Request(op=U, fr: AE2, to=/{CSE1}/{R-ID}, M2M-Request-ID1, Sec-Attributes (JWS1 / Auth_Tag1))
6.
- 메시지의 타겟인지 검증
- AE2와 CSE1 사이에 연관된 Ke2e_msg_auth 키를 사용하여 JWS/Auth_Tag1을 검증하기 위해 Sec-Attributes 및 파라미터들 사용
- AE2가 리소스에 대해 Update 동작을 수행하도록 인가되었는지를 보기 위한 체크
7. 리소스(CSE1/R-ID) 업데이트
Response를 생성하고, Response 상에서 Auth_Tag2를 계산하고, 이를 새로운 JWS와 함께 Sec-Attributes과 함께 AE2에 다시 전송
10. Response (fr: CSE1, to: AE2, M2M-Request-ID1, Sec-Attributes (JWS2 / Auth_Tag2))
8. Kpsa_AE1_CSE1을 사용하여 DTLS 터널 셋업
9. Update의 통지 전송
11. Response (fr: CSE1, to: AE2, M2M-Request-ID1, Sec-Attributes (JWS2 / Auth_Tag2))
12. 파라미터들 및 ke2e_msg_auth 키를 사용하여 적절한 JWS2/Auth_Tag2 계산

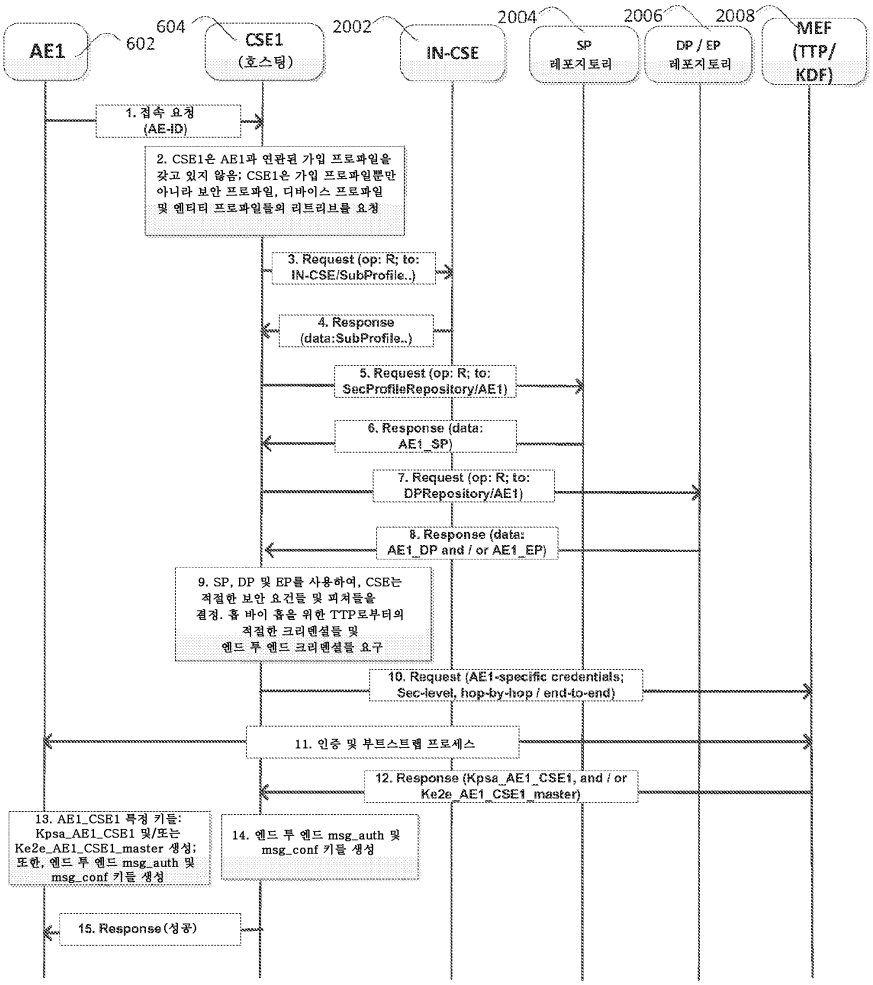
도면18


AE1에서의 보안 스토어
Kpsa_AE1_CSE1: DTLS
AE1
602
CSE1에서의 보안 스토어
Kpsa_AE1_CSE1: DTLS
Kpsa_CSE2_CSE1: DTLS
Ke2e_AE2_CSE1_msg_auth_conf: AES-CCM (JWE)
CSE1
(호스팅)
604
CSE2에서의 보안 스토어
Kpsa_AE2_CSE2: DTLS
Kpsa_CSE2_CSE1: DTLS
702
CSE2
AE2에서의 보안 스토어
Kpsa_AE2_CSE2: DTLS
Ke2e_AE2_CSE1_msg_auth_conf: AES-CCM (JWE)
1102
AE2
(클라이언트 애플리케이션)
0. 엔드 투 엔드 메시지 인증 및 기밀성이 요구되는지 결정; 크리덴셜들/키들의 올바른 세트 획득 및 올바른 AEAD 알고리즘 식별
1. 파라미터들을 사용하여 AAD 상에서 적절한 Auth_Tag를 계산하고, ke2e_msg_auth_conf 키가 또한 메시지 암호화
2. Kpsa_AE2_CSE2를 사용하여 DTLS 터널 셋업
3. Request(op=U, fr: AE2, to=/{CSE1}/{R-ID}, M2M-Request-ID1, Sec-Attributes / JWE1 (Encrypted Content1, Auth_Tag1))
4. Kpsa_CSE2_CSE1을 사용하여 DTLS 터널 셋업
5. Request(op=U, fr: AE2, to=/{CSE1}/{R-ID}, M2M-Request-ID1, Sec-Attributes / JWE1 (Encrypted Content1, Auth_Tag1))
6.
- 메시지의 타겟인지 검증
- Sec-Attributes 및 파라미터들을 사용하여 AAD, IV 및 다른 파라미터들을 결정하고, AE2와 CSE1 사이에 연관된 Ke2e_msg_auth_conf 키를 사용하여, Auth_Tag를 검증하고, 메시지를 복호화
- AE2가 리소스에 대해 Update 동작을 수행하도록 인가되었는지를 보기 위해 체크
7.
- 리소스(CSE1/R-ID) 업데이트
- 업데이트에 대한 통지를 AE1에 전송
8. Kpsa_AE1_CSE1을 사용하여 DTLS 터널 셋업
9. Request(op=N, fr: CSE1, to= AE1, M2M-Request-ID2)
10. Response 를 생성하고, Auth_Tag2를 계산하고, Response를 암호화하고, 이를 AE2에 전송
11. Response (fr: CSE1, to: AE2, M2M-Request-ID, Sec-Attributes / JWE2 (Encrypted Content2, Auth_Tag2))
12. Response (fr: CSE1, to: AE2, M2M-Request-ID, Sec-Attributes / JWE2 (Encrypted Content1, Auth_Tag2))
13. 메시지를 복호화하고, Auth_Tag2 검증

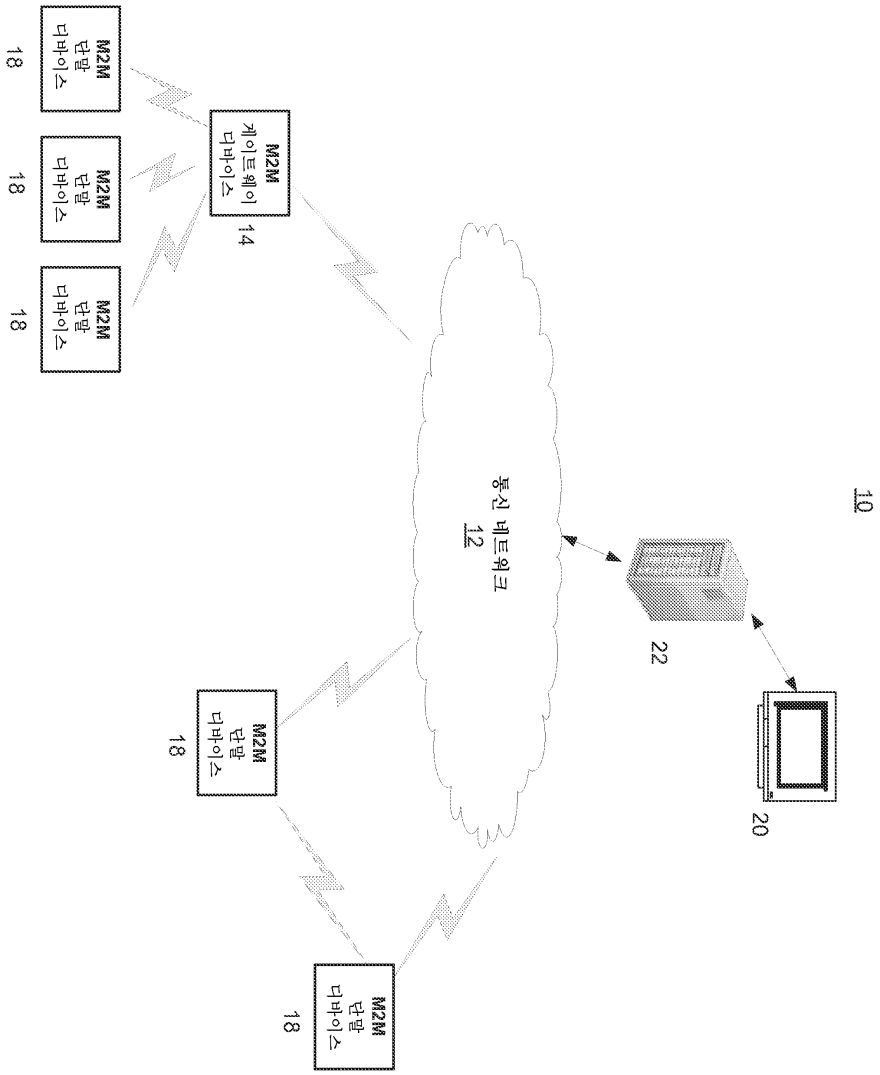
도면19


AE1에서의 보안 스토어
Ke2e_AE1_CSE1_msg_auth: HMAC-SHA-256 (JWS)
CSE1에서의 보안 스토어
Ke2e_AE1_CSE1_msg_auth: HMAC-SHA-256 (JWS)
Ke2e_AE2_CSE1_msg_auth_conf: AES-CCM (JWE)
Ke2e_CSE2_CSE1_msg_auth: (JWS)
CSE2에서의 보안 스토어
Ke2e_AE2_CSE2_msg_auth: JWS (MAC)
Ke2e_CSE2_CSE1_msg_auth: JWS (MAC)
AE2에서의 보안 스토어
Ke2e_AE2_CSE1_msg_auth_conf: AES-CCM (JWE)
Ke2e_AE2_CSE2_msg_auth: JWS (MAC)
AE1
602
CSE1
(호스팅)
604
702
CSE2
1102
AE2
(클라이언트 에플리케이션)
0. 엔드 투 엔드 메시지 인증 및 기밀성이 요구되는지 결정; 크리덴셜들/키들의 올바른 세트 획득 및 올바른 AEAD 알고리즘 식별
1.
- 파라미터들을 사용하여 AAD 상에서 적절한 Auth_Tag1을 계산하고, ke2e_AE2_CSE1_msg_auth_conf 키가 또한 메시지 암호화
- ke2e_AE2_CSE2_msg_auth를 사용하여 외부 메시지 상에서 Auth_Tag2 (MAC)를 계산하고, JWS1 생성
2. Request(op=U, fr: AE2, to=/{CSE1}/{R-ID}, M2M-Request-ID1, Sec-Attributes / JWE1 (Encrypted Content1, Auth_Tag1), JWS1 / MAC)
3. CSE2가 JWS1/MAC에서 Auth_Tag 검증
4. Request(op=U, fr: AE2, to=/{CSE1}/{R-ID}, M2M-Request-ID1, Sec-Attributes / JWE1 (Encrypted Content1, Auth_Tag1), JWS2 / MAC)
5.
- 메시지의 타겟인지 검증하고, 외부 JWS2/MAC 검증
- Sec-Attributes 및 파라미터들을 사용하여 AAD, IV 및 다른 파라미터들을 결정하고, AE2와 CSE1 사이에 연관된 Ke2e_msg_auth_conf 키를 사용하여, Auth_Tag를 검증하고, 메시지를 복호화
- AE 2가 리소스에 대해 Update 동작을 수행하도록 인가되었는지를 보기 위해 체크
6.
- 리소스(CSE1/R-ID) 업데이트
- 업데이트에 대한 통지를 AE1에 전송
7. Request(op=N, fr: CSE1, to= AE1, M2M-Request-ID2, Sec-Attributes (JWS3 / MAC))
8. AE1이 JWS3/MAC 검증
9. Response를 생성하고, Auth_Tag2를 계산하고, Response를 암호화하고, 이를 AE2에 전송
10. Response (fr: CSE1, to: AE2, M2M-Request-ID, Sec-Attributes / JWE2 (Encrypted Content2, Auth_Tag2), JWS4 / MAC)
11. Response (fr: CSE1, to: AE2, M2M-Request-ID, Sec-Attributes / JWE2 (Encrypted Content1, Auth_Tag2), JWS5/MAC)
12. AE2가 JWS5를 검증한 후, 내부 메시지를 복호화하고, JWE 내의 Auth_Tag2 검증

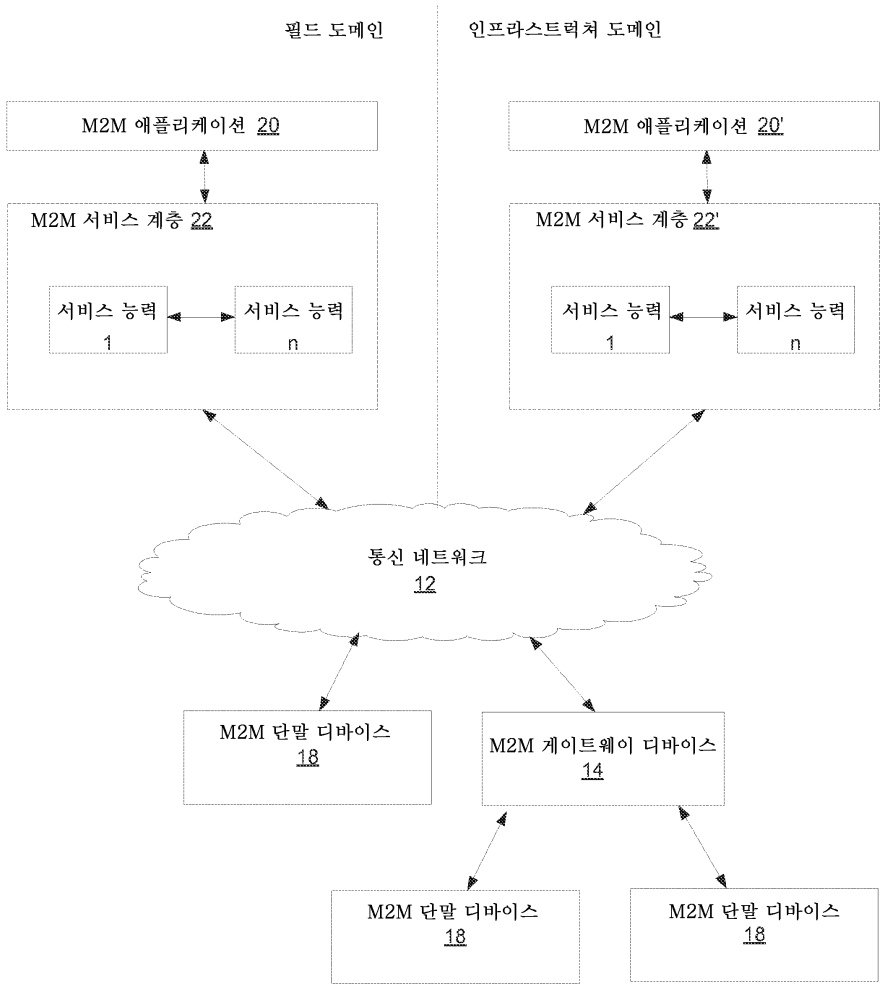
도면20


AE1
602
604
CSE1
(호스팅)
2002
IN-CSE
2004
SP
레포지토리
2006
DP / EP
레포지토리
2008
MEF
(TTP/
KDF)
1. 접속 요청
(AE-ID)
2. CSE1은 AE1과 연관된 가입 프로파일을
갖고 있지 않음; CSE1은 가입 프로파일뿐만
아니라 보안 프로파일, 디바이스 프로파일
및 엔티티 프로파일들의 리트리브를 요청
3. Request (op: R; to:
IN-CSE/SubProfile..)
4. Response
(data:SubProfile..)
5. Request (op: R; to:
SecProfileRepository/AE1)
6. Response (data:
AE1_SP)
7. Request (op: R; to:
DPRepository/AE1)
8. Response (data:
AE1_DP and / or AE1_EP)
9. SP, DP 및 EP를 사용하여, CSE는
적절한 보안 요건들 및 피처들을
결정. 홉 바이 홉을 위한 TTP로부터의
적절한 크리덴셜들 및
엔드 투 엔드 크리덴셜들 요구
10. Request (AE1-specific credentials;
Sec-level, hop-by-hop / end-to-end)
11. 인증 및 부트스트랩 프로세스
12. Response (Kpsa_AE1_CSE1, and / or
Ke2e_AE1_CSE1_master)
13. AE1_CSE1 특정 키들:
Kpsa_AE1_CSE1 및/또는
Ke2e_AE1_CSE1_master 생성;
또한, 엔드 투 엔드 msg_auth 및
msg_conf 키들 생성
14. 엔드 투 엔드 msg_auth 및
msg_conf 키들 생성
15. Response (성공)

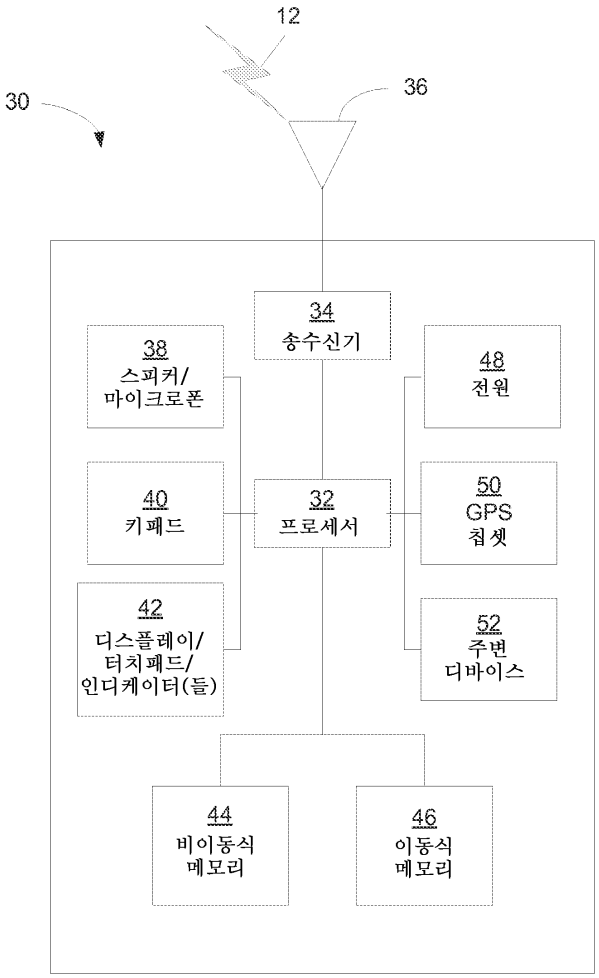
도면21a


M2M 단말 디바이스
18
M2M 단말 디바이스
18
M2M 단말 디바이스
18
M2M 게이트웨이 디바이스
14
통신 네트워크
12
10
22
20
M2M 단말 디바이스
18
M2M 단말 디바이스
18

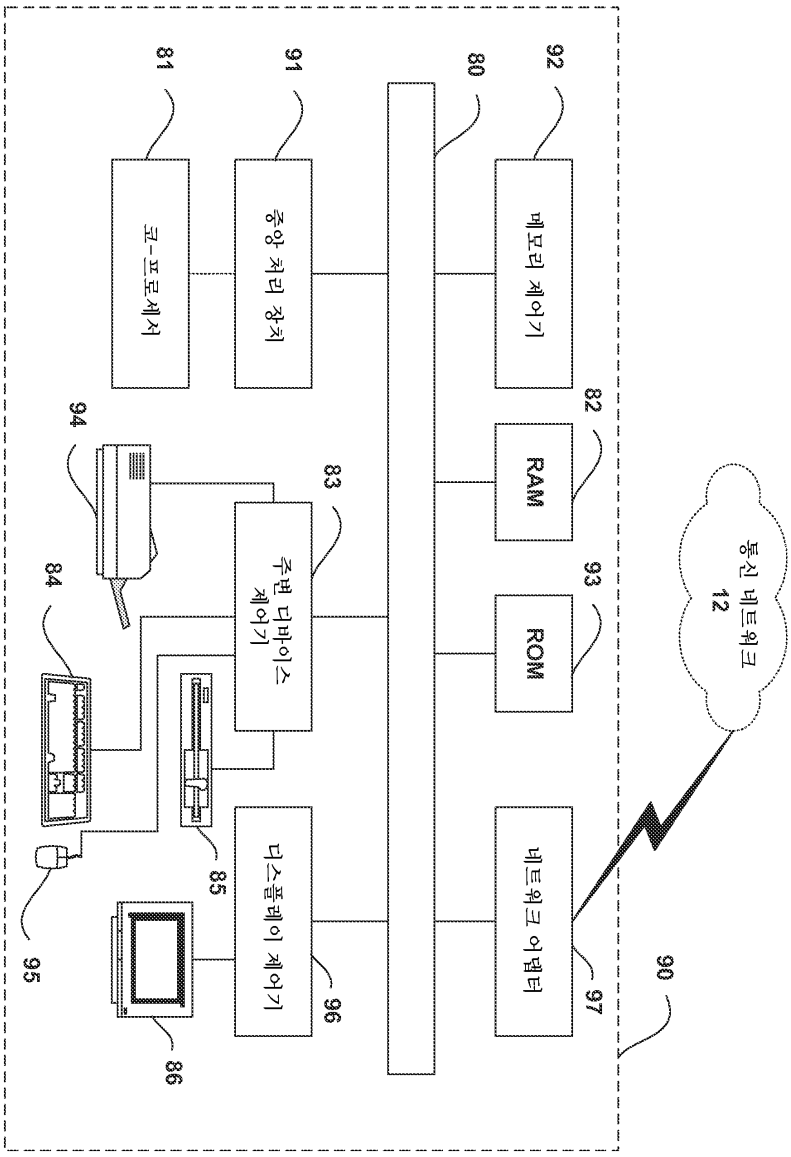
도면21b


필드 도메인
인프라스트럭처 도메인
M2M 애플리케이션 20
M2M 애플리케이션 20'
M2M 서비스 계층 22
M2M 서비스 계층 22'
서비스 능력 1
서비스 능력 n
서비스 능력 1
서비스 능력 n
통신 네트워크 12
M2M 단말 디바이스 18
M2M 게이트웨이 디바이스 14
M2M 단말 디바이스 18
M2M 단말 디바이스 18

도면21c


12
30
36
34
송수신기
38
스피커/
마이크로폰
48
전원
40
키패드
32
프로세서
50
GPS
칩셋
42
디스플레이/
터치패드/
인디케이터(들)
52
주변
디바이스
44
비이동식
메모리
46
이동식
메모리

도면21d


81
91
80
92
코-프로세서
중앙 처리 장치
메모리 제어기
82
RAM
94
83
주변 디바이스 제어기
93
ROM
84
통신 네트워크
12
85
디스플레이 제어기
네트워크 어댑터
95
96
97
90
86