(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0028088
(43) 공개일자 2019년03월18일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 29/08* (2006.01)
(52) CPC특허분류
H04L 63/065 (2013.01)
H04L 63/062 (2013.01)
(21) 출원번호 10-2017-0115096
(22) 출원일자 2017년09월08일
심사청구일자 2017년09월08일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
성순화
대전광역시 유성구 은구비남로 34, 812동 802호 (노은동, 열매마을아파트 8단지 새미래)
(74) 대리인
김원준

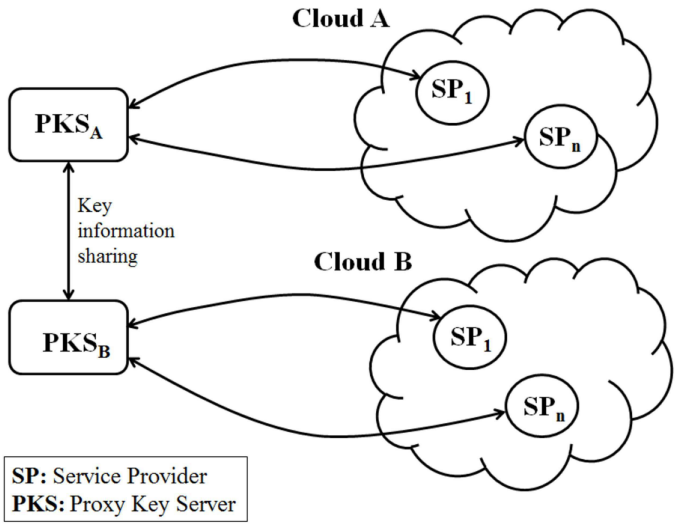
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 **클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법**

(57) 요 약

본 발명은 앞으로 급속 성장할 사물인터넷 데이터를 클라우드 컴퓨팅에서 안전하게 처리할 수 있도록 지원하는 키 관리 방법과 이를 이용한 사물인터넷 서비스에 관한 것으로, 보다 상세하게는 각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 키 정보를 공유하는 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법 및 이를 이용한 사물인터넷 서비스에 관한 것이다.

대 표 도 - 도2



(52) CPC특허분류
H04L 63/0884 (2013.01)
H04L 67/12 (2013.01)

명 세 서

청구범위

청구항 1

각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 키 정보를 공유하는 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법.

청구항 2

제 1 항에 있어서,

상기 연합키는,

(1) 신뢰된 중개인은 다음과 같은 임의의 다항 a(X)를 선택하며,

$a(X) \in Fq[X]$, 차수 $t = a(0) = s$

(2) 파티 $p_1, \ldots, p_n$ 사이에서 비밀키 s를 공유하고.

이때, $t < n/2$인 파티는 s에 대한 어떠한 정보도 가지지 않으며, t+1 파티의 그룹은 비밀키 s를 복구할 수 있다.

(3) pi 에 대하여 $s_i=a(i)$를 공유하며.

(4) 주어진 집합 U의 t+1이 비밀키 $s = a(0) = \sum j \in U \lambda j aj$를 공유 및 복구하며 이때, λi 는 U에 관한 라그랑즈 계수인 과정에 의해 비밀키 s를 공유하는 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법.

청구항 3

제 1 항 또는 제 2 항에 있어서,

상기 연합키는 사물기기와 프락시 키 서버 사이의 새로운 키와 사물기기 들 간의 통신에 사용될 세션키를 포함하는 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법.

청구항 4

제 3 항에 있어서,

사물기기 A에 대한 응용서비스를 위한 상기 사물기기와 프락시 키 서버 사이의 새로운 키의 분산 과정은,

1. $O_A \rightarrow PKS : A_n$;

2. $PKS \rightarrow O_A : K_t, A_k \oplus H(MK, A_n, K_t), H(A_k, K_t), K_n$;

3. $O_A \rightarrow PKS : H(A_k, K_n)$;

4. $PKS \rightarrow O_A : H(A_k, A_n)$

으로 이루어지며, 이때, O_A는 사물기기 A, PKS는 프락시 키 서버, A_n은 사물기기 A에 대한 임시값, K_t는 프락시 키 서버의 새로운 키, A_k는 사물기기 A의 새로운 키, MK는 마스터키, K_n은 프락시 키 서버에 대한 임시값인 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법.

청구항 5

제 4 항에 있어서,

사물기기 A와 B 간의 통신에 사용될 세션키는,

1. $O_A \rightarrow O_B$: A, $H(A_k, B)$, A_n

2. $O_B \rightarrow PKS$: A, B, $H(B_k, A, K, H(A_k, B), A_n)$, A_n, B_n

3. $PKS \rightarrow O_B$: $H(B_k, A, B_n) \oplus SK$, $H(B_k, A, B_n, SK)$, $H(A_k, B, K) \oplus SK$, $H(A_k, B, A_n, SK)$

4. $O_B \rightarrow O_A$: $H(A_k, B, A_n) \oplus SK$, $H(A_k, B, A_n, SK)$

의 과정에 의해 분산되며, 이때, O_A와 O_B 각각 사물기기 A와 B, A_k와 B_k 는 각각 사물기기 A와 B의 새로운 키, A와 B는 각각 사물기기 A와 B의 메시지, A_n과 B_n은 각각 사물기기 A와 B에 대한 임시값, PKS는 프락시 키 서버, K는 프락시 키 서버의 메시지, SK는 세션키인 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법.

청구항 6

제 1 항의 키 관리 방법을 이용한 클라우드 IoT 서비스 방법에 관한 것으로서,

(A) 사물인터넷 환경에 포함되는 사물기기들이 클라우딩 컴퓨터 환경에서 각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 센서 데이터의 파티 인증을 수행하는 인증 단계;

(B) 상기 인증이 완료되면, 클라우드 프락시 키 서버는 다른 프락시 키 서버와 연합키 정보를 공유하여 사물기기들 간 인증된 센서 데이터를 전송하는 단계; 및

(C) 인증된 센서 데이터를 사용하여 코어 서비스를 진행하는 단계;

를 포함하는 클라우드 사물인터넷 서비스의 제공 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 앞으로 급속 성장할 사물인터넷 데이터를 클라우드 컴퓨팅에서 안전하게 처리할 수 있도록 지원하는 키 관리 방법과 이를 이용한 사물인터넷 서비스에 관한 것이다.

배 경 기 술

[0002] 사물인터넷(IoT, Internet of Things)은 사물에 센서를 부착하여 실시간으로 데이터를 인터넷으로 주고 받는 것에 의해 모든 사물을 연결하여 사람과 사물, 사물과 사물 간의 정보를 상호 소통하는 지능형 기술 및 서비스를 말한다. IoT는 기존의 유선통신을 기반으로 한 인터넷이나 모바일 인터넷보다 더욱 진화된 단계로 인터넷에 연결된 기기가 사람의 개입없이 상호간에 알아서 정보를 주고 받아 처리한다. 사물이 인간에 의존하지 않고 통신을 주고받는 점에서 기존의 유비쿼터스나 M2M(Machine to Machine: 사물지능통신)과 비슷하기도 하지만, 통신장비와 사람과의 통신을 주목적으로 하는 M2M의 개념을 인터넷으로 확장하여 사물은 물론이고 현실과 가상세계의 모든 정보와 상호작용하는 개념으로 진화한 단계라고 할 수 있다. IoT 기술에 의하면 다양한 물리 공간의 사물들과 가상 공간의 프로세스 및 데이터 콘텐츠들을 인터넷으로 상호 연결시키는 것에 의해 초연결사회(hyper-connected society)를 구축하도록 한다.

[0003] 인터넷에 연결된 장치 수의 증가는 공격할 수 있는 대상의 증가와 위협 요소의 확장으로 IoT 보안 기술을 요구한다. IoT 기술 특성상 주변의 일상 사물들이 연결되므로 개인정보 유출이나 프라이버시 침해가 우려되는 범위가 증가하며, 침해 정도 역시 현재와 비교할 수 없을 정도로 증폭될 것이다. 따라서 IoT에 대한 인증과

기밀성, 데이터 무결성에 대해 많은 관심이 집중되고 있다. 또한 IoT 서비스는 센싱 정보를 토대로 사람, 사물, 서비스의 통합 및 데이터 공유를 지향하므로 프라이버시 침해 발생이 필연적이다. 이러한 프라이버시 침해 문제를 해결하지 못할 경우, 전자주민증, 인체내장형 Radio Frequency Identification(RFID) 같은 IoT 서비스 활성화는 불가능하다. 따라서 IoT 서비스의 확장여부는 보안에 의해 결정된다고 하여도 과언이 아니다.

[0004] IoT의 구조는 인식, 네트워크, 응용 단계로 나눌 수 있다. 인식 단계는 RFID, 센서, 2차원 코드에 의해 사물과 주위환경으로부터 정보를 얻는 것으로 주요 기능은 RFID 기술, 센싱, 제어 기술로 칩 개발, 통신 프로토콜 연구, 스마트 노드 에너지 공급과 세그먼트를 포함한다. 네트워크 단계의 주요 기능은 사물이 인터넷에 연결되도록 지원하는 것으로, 네트워크 통합을 통하여 데이터 인식과 정보 제어 사이 실시간 정보를 양방향으로 전송하는 것이다. 센서가 정보를 감지한 후, 사용할 네트워크를 통하여 그 배경을 이동한다. 응용 단계는 각종 서비스 분야와 형태에 적합하게 정보를 가공하고 처리하거나 각종 기술을 융합하는 것으로, 응용 단계는 사물, 소프트웨어의 개발이 목적이며, 클라우드 컴퓨팅 기술로 사용자에게 IoT의 다양한 응용을 제공한다.

[0005] 클라우드 컴퓨팅은 이용자의 모든 정보를 인터넷 상의 서버에 저장하고, 이 정보를 각종 Information Technology(IT) 기기를 통하여 언제 어디서든 이용할 수 있는 것으로 인터넷을 이용한 IT 자원의 주문형 아웃소싱 서비스라고 할 수 있다. 그러나 이는 서버가 해킹당할 경우 개인정보가 유출될 수 있고, 서버 장애가 발생하면 자료 이용이 불가능하다는 단점이 있다. 이러한 보안 문제로 인하여 클라우드 컴퓨팅은 아직 활성화되고 있지 않다.

[0006] 일반적으로 데이터 위치를 알고 있을 때 데이터를 보호하기 위하여 암호화를 한다. 클라우드 환경에서의 암호화는 사용자가 원하는 데이터가 어디에 있는지 알지 못하므로 암호화할 키 관리가 어렵다.

[0007] 암호 기술로 데이터를 보호하기 위한 전통적이고 중앙집중적인 방법은 그룹키를 생성한 후, 각 그룹 멤버에게 이 키를 분산하는 것이다. 이러한 그룹키는 중앙집권적인 제어기인 키 서버에 문제점이 있다면 전체 그룹이 영향을 받게 되고, 비용이 많이 드는 공개키의 작동을 필요로 한다. 김 등(7th ACM Conference on Computer and Communications Security (CCS'00), pp. 235-24, Nov. 2004)은 트리 기반 그룹 디피 헬먼(tree-based group Diffie-Hellman)으로 가장 효율적인 그룹키 사용을 제안하였으나, 이를 클라우드 환경에 적용하기는 어렵다. Arijit Ukil 등(IEEE International Conference on Emerging Trends and Applications in Computer Science (NCETACS), pp 1-6, Apr. 2011)은 IoT 장치의 낮은 보안 서비스 문제와 위협을 해결하기 위하여 전통적인 방법을 사용한 도구를 제시하였으나, 이는 카메라와 같은 모니터링에 사용하는 장치이므로 IoT의 특성에 적합하지 않다.

[0008] IoT는 클라우드 컴퓨팅 환경에서 어디서나 민감한 데이터를 가지고 이러한 데이터 접근이 필요한 응용 능력을 요구한다. 이를 위하여, 클라우드 환경에서 데이터를 안전하게 보호·유지하기 위한 암호화키와 사용 가능한 데이터를 얻기 위한 복호화키를 얻을 수 있는 응용 방법에 대한 필요성이 지속적으로 제기되고 있다.

선행기술문헌

특허문헌

[0009] (특허문헌 0001) 등록특허 10-1701052

(특허문헌 0002) 등록특허 10-1688118

비특허문헌

[0010] (비특허문헌 0001) 7th ACM Conference on Computer and Communications Security (CCS'00), pp. 235-24, Nov. 2004.

(비특허문헌 0002) IEEE International Conference on Emerging Trends and Applications in Computer Science (NCETACS), pp 1-6, Apr. 2011.

발명의 내용

해결하려는 과제

[0011] 본 발명은 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법을 제공하는 것을 목적으로 한다.

[0012] 본 발명은 또한 상기 키 관리 방법을 이용한 사물인터넷 서비스의 제공 방법을 제공하는 것을 다른 목적으로 한다.

과제의 해결 수단

[0013] 전술한 목적을 달성하기 위한 본 발명은 각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 키 정보를 공유하는 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법에 관한 것이다.

[0014] 또한 본 발명은 상기 키 관리 방법을 이용한 클라우드 IoT 서비스 방법에 관한 것으로서, (A) 사물인터넷 환경에 포함되는 사물기기들이 클라우딩 컴퓨터 환경에서 각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 센서 데이터의 파티 인증을 수행하는 인증 단계; (B) 상기 인증이 완료되면, 클라우드 프락시 키 서버는 다른 프락시 키 서버와 연합키 정보를 공유하여 사물기기들 간 인증된 센서 데이터를 전송하는 단계; 및 (C) 인증된 센서 데이터를 사용하여 코어 서비스를 진행하는 단계;를 포함하는 클라우드 IoT 서비스 방법에 관한 것이다.

발명의 효과

[0015] 이상과 같이 본 발명의 키 관리 방법은 많은 센서 데이터의 추가 및 제거가 용이할 뿐만 아니라 키 생성을 줄이고 통신 오버헤드를 줄일 수 있도록 지원하는 프락시 키 서버를 기반으로 하므로, 다중 보안 시스템을 통한 중앙 집중식 모니터링에 필요한 서버와 관리를 줄일 수 있다. 또한, 다른 추가적인 키 관련 정보 기술 없이도 보안을 위한 가장 최근 키의 복구와 통합을 업데이트할 수 있다.

[0016] 따라서 본 발명의 키 관리 방법은 앞으로 급속 성장할 IoT 데이터를 클라우드 컴퓨팅에서 안전하게 처리할 수 있도록 지원하여, 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안에 효율적으로 이용될 수 있다.

도면의 간단한 설명

[0017] 도 1은 데이터 통신 방식인 점대점(point-to-point) 방식과 다대다 방식을 나타내는 개념도.

도 2는 클라우드에서의 본 발명에 의한 키 연합에 대한 모식도.

도 3은 웹 서비스에 의한 장치에서 장치로 센서 데이터 이동을 보여주는 개념도.

발명을 실시하기 위한 구체적인 내용

[0018] 이하 첨부된 도면을 들어 본 발명을 보다 상세히 설명한다. 그러나 이러한 설명은 본 발명의 기술적 사상의 내용과 범위를 쉽게 설명하기 위한 예시일 뿐, 이에 의해 본 발명의 기술적 범위가 한정되거나 변경되는 것은 아니다. 이러한 예시에 기초하여 본 발명의 기술적 사상의 범위 안에서 다양한 변형과 변경이 가능함은 당업자에게는 당연할 것이다.

[0019] 전술한 바와 같이 본 발명은 각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 키 정보를 공유하는 것을 특징으로 하는 클라우드 컴퓨팅에서 사물인터넷 데이터의 보안을 위한 키 관리 방법에 관한 것이다.

[0020] **클라우드 IoT 키 관리**

[0021] 연합키 관리는 다른 컴퓨터 시스템들이 함께 작동할 때 언급이 되며, 이는 다른 응용이 같은 키 서버로부터 키들을 어떻게 얻을 수 있는지를 포함한다. 이는 클라우드의 민감한 데이터를 보호하기 위해 암호화 되기 전에 필요로 하는 키 관리의 중요한 측면이며, 키 연합 능력 부족은 많은 암호와 키 관리의 유용성을 제한한다.

[0022] 암호화 키는 항상 유일한 식별을 위한 키와 관련된 추가적인 정보를 가진다. 예를 들면, 테이프 구동 장치는 저장된 데이터를 암호화하기 위한 키를 얻을 때, 암호화된 데이터와 함께 저장된 유일한 키 식별자를 얻는다. 암호화된 데이터를 복호화하기 위하여 테이프 구동 장치는 암호화된 데이터와 함께 발견되는 키 식별자와 일치하는 키를 요구한다. 그러나 클라우드 환경에서의 암호화는 사용자가 원하는 데이터가 어디에 있는지 알지 못

하므로 암호화할 키 관리가 어려우며, 따라서 현재 시스템으로는 연합키 관리를 실행할 수 없다. 이에 본 발명은 정확한 키를 얻을 수 있는 키 서버에 관한 정보를 포함하는 키 식별자 대신 클라우드 컴퓨팅에서 프락시 키 서버 기반의 연합키를 사용한다.

[0023] 도 1은 데이터 통신 방식인 점대점(point-to-point) 방식과 다대다 방식을 나타내는 개념도이다. 클라우드 환경에서 점대점 방식은 다양한 곳에서 요구하는 데이터를 누가 어떤 데이터를 원하는지, 원하는 데이터를 원하는 곳으로 안전하게 전송할 수 있는 인증 절차가 복잡하며 이에 사용되는 키가 많아 키 관리가 비효율적이다. 다대다(many-to-many) 방식은 기존의 인터넷과 차세대 인터넷을 하나의 네트워크로 묶어, 마치 하나의 신경조직처럼 작동할 수 있게 제어하는 가상 슈퍼컴퓨터인 그리드(grid)로서 클라우드 컴퓨팅을 지원하는 모델이다. 본 발명은 다대다 방식을 바탕으로 한 IoT 응용 단계의 서비스 활성화를 위한 클라우드 컴퓨팅의 안전한 센서 데이터 지원을 위한 키 설계에 관한 것이다. 센서 데이터 파티별 인증을 위한 다대다 방식은 클라우드 환경의 프락시 키 서버(Proxy Key Server:PKS)를 도입하여 이 서버들의 연합키로서 파티별 센서 데이터 인증 절차를 시작한다.

[0024] 본 발명에서 IoT 특성을 고려한 연합키는 많은 센서 데이터의 키 추가 및 제거가 용이할 뿐만 아니라 키 생성을 줄이고 통신 오버헤드를 줄일 수 있도록 지원하는 프락시 키 서버에서 진행된다. 클라우드 컴퓨팅 환경에서 적응할 수 있는 프락시 키 서버는 클라우드마다 하나의 서버를 두어 연합키로서 키 정보를 공유한다. 연합키는 다음과 같은 과정으로 비밀키 s를 공유한다.

[0025] (1) 신뢰된 중개인은 다음과 같은 임의의 다항 a(X)를 선택한다.

[0026] $a(X) \in Fq[X]$, 차수 $t = a(0) = s$

[0027] (2) 파티 $p_1, \ldots, p_n$ 사이에서 비밀키 s를 공유한다.

[0028] 이때, $t < n/2$인 파티는 s에 대한 어떠한 정보도 가지지 않으며, t+1 파티의 그룹은 비밀키 s를 복구할 수 있다.

[0029] (3) pi 에 대하여 $s_i=a(i)$를 공유한다.

[0030] (4) 주어진 집합 U의 t+1이 비밀키 $s = a(0) = \sum j \in U \lambda jaj$를 공유 및 복구한다.

[0031] 이때, λi 는 U에 관한 라그랑즈 계수이다.

[0032] 도 2는 클라우드에서의 본 발명에 의한 키 연합에 대한 모식도이다. 도 2에서와 같이 클라우드 A의 클라우드 서비스 제공자(Cloud Service Provider:CSP)는 프락시 키 서버 A에게 센서 데이터 파티 인증을 요청하면, 프락시 키 서버 A는 클라우드 B의 프락시 키 서버 B와 상호 통신하여 연합키 결과로 센서 데이터 파티를 인증한다.

[0033] 인증 과정을 마치면 도 3에서와 같이 웹 서비스는 장치에서 장치로 센서 데이터 이동을 시작한다. 센서 노드들은 랜덤하게 배치되기 때문에 네트워크 토폴로지 정보를 사전에 획득하기 어렵다. 따라서 서비스 중개인이 필요한 서비스의 센서 데이터를 모아 연합키 생성을 프락시 키 서버에게 요청한다. 연합키 생성 요청을 받은 프락시 키 서버 A는 연합키를 생성한 후, 다른 클라우드 프락시 키 서버 B와 연합키 정보를 공유한다. 클라우드 서비스 제공자의 센서 데이터 요청에 따라 프락시 키 서버 A와 프락시 키 서버 B의 상호 통신으로 연합키 결과를 서비스 인터페이스를 통하여 클라우드 서비스 제공자에게 보낸다. 클라우드 서비스 제공자는 인증된 센서 데이터를 확인 후, 코어 서비스를 진행한다.

[0034] **클라우드 컴퓨팅의 연합키 관리**

[0035] 프락시 키 서버는 두 개체(사물기기) A와 B 사이 안전한 데이터 통신을 위하여 두 가지 중요한 키 분산을 한다. 하나는 개체 A와 프락시 키 서버에게만 알려진 개체 A에 의해 사용될 새로운 키의 분산이다. 다른 하나는 프락시 키 서버가 개체 A와 B에 의해 사용될 세션키의 분산이다. 이러한 키 분산은 배타적 논리합(exclusive OR)과 해시 함수 사용을 바탕으로 한다. 개체 A에 대한 새로운 키의 분산은 이러한 목적에만 사용되는 마스터 키를 기반으로 한다. 이하에서 사물기기에 대한 키 분산과, 세션키의 분산에 대해 상세히 설명한다. 하기 표 1은 키 분산에 대해 사용되는 약어를 나타낸다.

표 1

Symbol	Description
O_A, O_B	개체(사물기기) A, B
A, B	개체 A, B의 메시지
K	프락시 키 서버(PKS)의 메시지
PKS_A, PKS_B	클라우드 A, B의 프락시 키 서버(PKS)
MK	마스터키
SK	세션키
A_k, B_k	개체 A, B의 새로운 키
K_t	프락시 키 서버(PKS)의 새로운 키
K_n	프락시 키 서버(PKS)에 대한 임시값
A_n, B_n	개체 A, B에 대한 임시값

[0036]

[0037] **1) 사물기기에 대한 키 분산**

[0038] 다음은 개체 A에 대한 응용 서비스를 위한 새로운 키 분산 과정이다.

[0039] 1. $O_A \rightarrow PKS : A_n$

[0040] 2. $PKS \rightarrow O_A : K_t, A_k \oplus H(MK, A_n, K_t), H(A_k, K_t), K_n$

[0041] 3. $O_A \rightarrow PKS : H(A_k, K_n)$

[0042] 4. $PKS \rightarrow O_A : H(A_k, A_n)$

[0043] 먼저 개체 A가 임시값 A_n을 전달하면, 프락시 키 서버는 개체 A에게 전달하기 위해, 마스터키 MK로 두 개 장치의 응용 서비스를 위한 메시지 암호화에 필요한 새로운 키 A_k와 K_t를 생성하고, 임시값 K_n을 생성한다. 개체 A는 마스터키와 임시값 A_n를 알고 새로운 키 K_t가 주어지기 때문에, $x=H(MK, A_n, K_t)$를 생성할 수 있으며, $x \oplus (A_k \oplus H(MK, A_n, K_t))$로부터 A_k를 복구하기 위해 이를 사용한다. 개체 A는 $A_k \oplus H(MK, A_n, K_t)$를 위해 대체된 무효한 영역이 메시지가 가로채지 않았는지 체크하기 위해 $H(A_k, K_t)$를 사용한다. 즉 A_k가 프락시 키 서버가 보내려는 키인지를 체크하기 위해서이다. 두 개 장치의 응용 서비스를 위한 메시지 암호화에 필요한 새로운 키 K_t는 두 가지 목적을 제공한다. 첫째는 응용 서비스를 위한 개체 A의 메시지는 K_t없이 $A_k \oplus H(MK)$를 읽을 수 있고, H(MK)은 나중에 키들을 복구하기 위해 사용되므로 H(MK)을 발견하는 것은 마스터 키를 발견하는 만큼 유용하다. 서로 다른 K_t는 $H(MK, K_t)$를 생성할 때마다 하나의 값을 가지기 때문에 그가 $H(MK, K_t)$를 발견하는 것은 유용하지 않다. 둘째, K_t는 데이터 전송 동안 데이터가 분리되거나 다른 데이터로 대체되지 않게 보호하기 위한 메시지 요소들을 결합하는데 그 목표가 있다.

[0044] **2) 세션키의 분산**

[0045] 개체 A와 프락시 키 서버는 새로운 키를 사용하기 전에 응용 서비스를 위한 메시지 응답을 요구하고, $H(A_k, K_t)$는 제외될 수 있다. A_k에 대한 확인 기능은 응용 서비스를 위한 그 다음 메시지에 내재될 수 있다.

[0046] IoT 인증은 프락시 키 서버가 두 개체 사이 통신에 사용하는 세션키를 어떻게 분산하는가를 확인해 준다. IoT의 두 개체 사이 기밀성을 제공하기 위해, 프락시 키 서버는 각 개체에 대한 임시값과 세션키를 생성하며, 그 이동은 다음과 같이 진행한다.

[0047] 1. $O_A \rightarrow O_B : A, H(A_k, B), A_n$

[0048] 2. O_B → PKS : A, B, H(B_k, A, K, H(A_k, B), A_n), A_n, B_n

[0049] 3. PKS → O_B : H(B_k, A, B_n)$\oplus$SK, H(B_k, A, B_n, SK), H(A_k, B, K)$\oplus$SK, H(A_k, B, A_n, SK)

[0050] 4. O_B → O_A : H(A_k, B, A_n)$\oplus$SK, H(A_k, B, A_n, SK)

[0051] 두 장치 사이의 데이터 기밀성을 위하여, 키 이동 순서는 개체 A에서 개체 B, 개체 B에서 프락시 키 서버로 진행된다. 프락시 키 서버는 개체 A와 B에게 세션키 SK를 보내야 하며, 응용 서비스를 위한 새로운 키 A_k, B_k는 프락시 키 서버로부터 생성된다. 개체 A와 B는 세션키를 찾기 위해 H(A_k, B, A_n)와 H(B_k, A, B_n)을 부호화할 수 있다. H(A_k, B, A_n, SK)와 H(B_k, A, B_n, SK)는 메시지가 방해받지 않았고, 유효한 세션키가 복구되었는지 확인하기 위하여 개체 A와 B에 의해 확인될 수 있다. 상기 과정에 의해 개체 A와 B는 같은 세션키를 복구할 수 있으므로 일반적인 세션키를 갖는다.

[0052] **분석**

[0053] 클라우드 프락시 키 서버의 연합키 역할 가능성을 표 2와 같이 분석하였다. 표 2는 클라우드 파운드리 [https://www.cloudfoundry.org/future-network-security-cloud-foundry/]에서 본 연구(실시예)와 J. Park 등 (The Journal of Korean Institute of Communications and Information Sciences, 38(9), pp. 707-714, Sept. 2013; 비교예 1)와 R. Hummen 등(WISEC'13 Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 37-42, Apr. 2013; 비교예 2)의 키 관리를 공간 복잡도, 키 관리의 확장성, 스케줄링, 키 철회 저항성, 네트워크 안전성의 측면에서 비교 분석하였다.

표 2

[0054]

	비교예 1	비교예 2	실시예
공간복잡도	$O(r^2 \times t^2)$	O(r×t+3t)	O(r)
키 관리 확장성	낮음	낮음	높음
스케줄링	낮음	낮음	높음
키 철회 저항성	낮음	중간	높음
네트워크 안정성	중간	중간	높음

[0055] 공간 복잡도에서 r은 이웃 센서 노드 수, t는 요구된 센서 노드 철회 수행에 필요한 시간으로써 비교예 1은 사물인터넷 장치 간 상호 인증을 위한 대칭키 인증과 세션키 합의 동의 시스템이 필요하므로 공간복잡도는 이웃 센서 노드 수와 센서 노드 철회에 필요한 시간에 비례하고 $O(r^2 \times t^2)$, 비교예 2는 인증을 위한 게이트웨이의 선 인증, 세션 회수, 핸드셰이크 위임이 필요하므로 O(r×t+3t), 실시예는 클라우드 프락시 키 서버의 연합키에 의한 O(r)로서 공간복잡도가 낮음을 알 수 있다.

[0056] 프락시 키 서버를 사용하는 본 발명에서의 스케줄링은 클라우드 환경이 아닌 비교예 1, 2보다 우수하였다. 비교예 1은 성능을 개선하기 위해 인증 완료된 장치가 세션키 선 계산을 해야 하므로 키 관리 확장성이 낮으며, 비교예 2는 장치의 인증 오버헤드를 줄이기 위한 게이트웨이에서의 선 인증과 세션 회수로 키 관리 확장성이 낮은 반면, 실시예는 클라우드 컴퓨팅 프락시 키 서버의 연합키 관리로 키 관리 확장성이 높다는 것을 알 수 있다.

[0057] 키 철회 저항성에서 실시예는 자율적인 프락시 키 서버의 키 관리로 저항성이 높으며, 비교예 1은 인증을 시도할 때마다 새로운 난수를 생성하여 인증에 사용하므로 저항성이 낮고, 비교예 2는 장치의 핸드셰이크 위임으로 적은 자원으로 인증을 하지만 세션 회수로 부담이 있다.

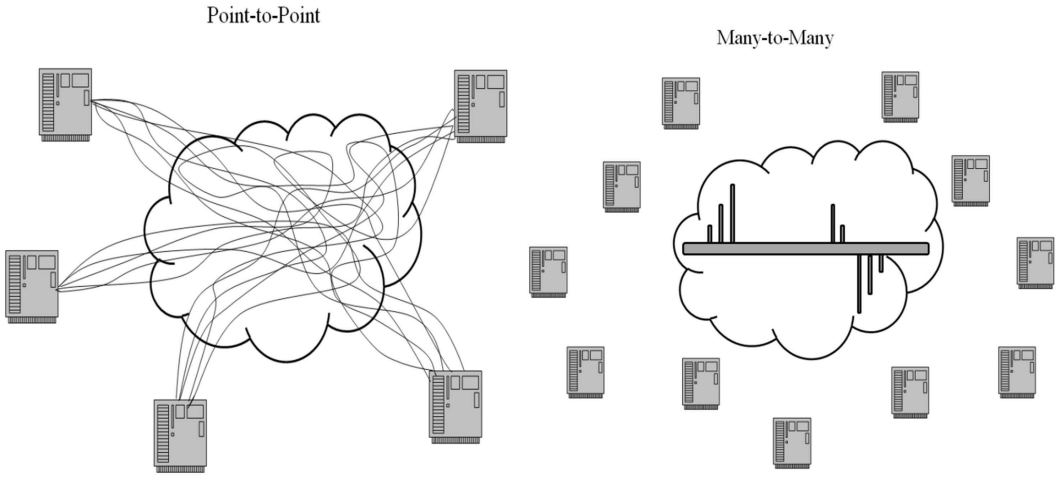
[0058] 네트워크 안전성에서 비교예 1은 장치 간 상호 인증과 세션키 동의를 제공하지만, 인증의 참여 장치들이 비밀키를 안전하게 공유하고 있다는 가정이 필요하며, 비교예 2는 인증을 위한 게이트웨이에서의 선 인증, 세션 회수를 하지만 핸드셰이크 위임의 안정성 보장이 미비하다. 그러나 실시예는 클라우드 컴퓨팅의 네트워크 연결에서

프락시 키 서버의 능동적인 키 복구와 업데이트로 그 안전성이 보장된다.

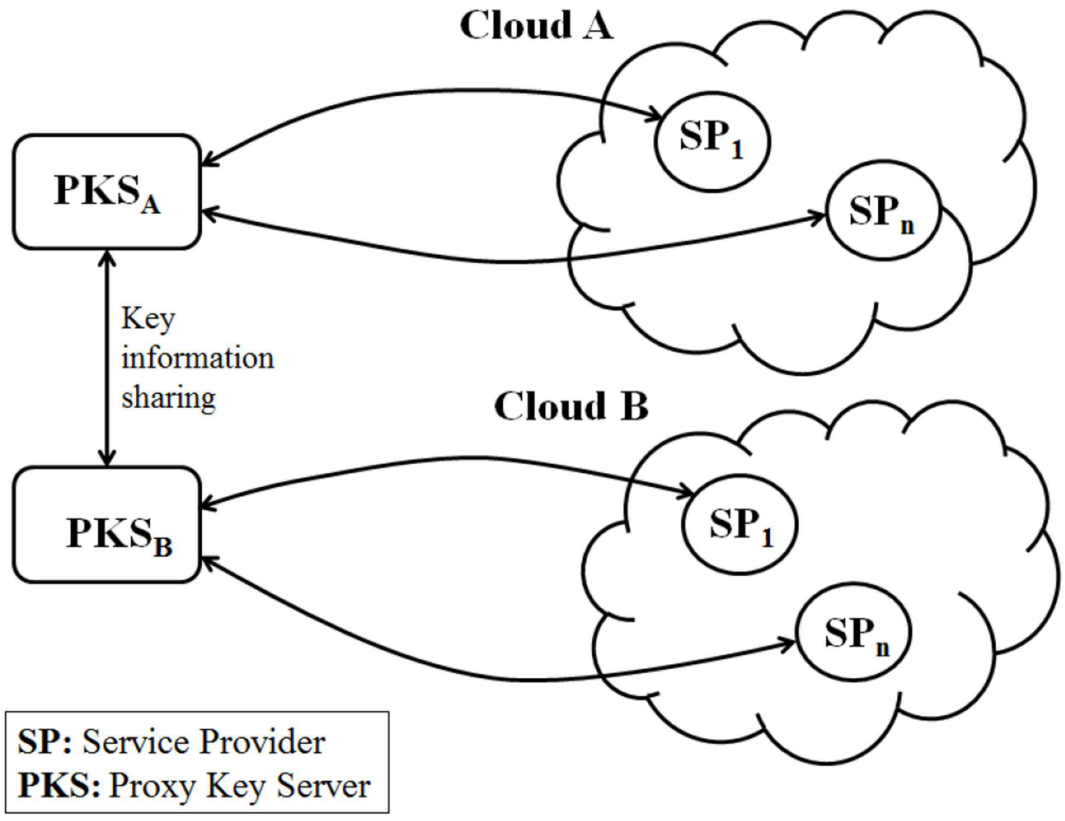
[0059] 또한 본 발명은 전술한 키 관리 방법을 이용한 클라우드 IoT 서비스 방법에 관한 것으로서, (A) 사물인터넷 환경에 포함되는 사물기기들이 클라우딩 컴퓨터 환경에서 각 클라우드마다 설정된 프락시 키 서버 기반의 연합키를 사용하여 센서 데이터의 파티 인증을 수행하는 인증 단계; (B) 상기 인증이 완료되면, 클라우드 프락시 키 서버는 다른 프락시 키 서버와 연합키 정보를 공유하여 사물기기들 간 인증된 센서 데이터를 전송하는 단계; 및 (C) 인증된 센서 데이터를 사용하여 코어 서비스를 진행하는 단계;를 포함하는 클라우드 IoT 서비스의 제공 방법에 관한 것이다.

도면

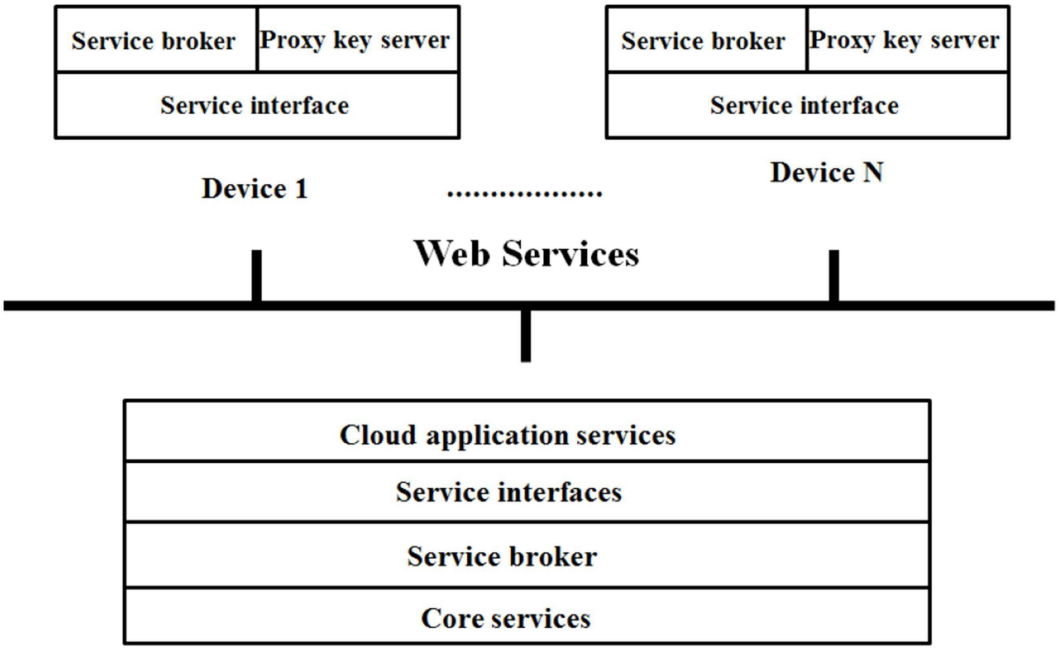
도면1



도면2



도면3


Service broker
Proxy key server
Service interface
Device 1
.....
Service broker
Proxy key server
Service interface
Device N
Web Services
Cloud application services
Service interfaces
Service broker
Core services