(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0053066
(43) 공개일자 2018년05월21일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01) *H04L 9/08* (2006.01)
H04L 9/30 (2006.01)
(52) CPC특허분류
H04L 9/3263 (2013.01)
H04L 9/0861 (2013.01)
(21) 출원번호 10-2016-0150418
(22) 출원일자 2016년11월11일
심사청구일자 2016년11월11일

(71) 출원인
한밭대학교 산학협력단
대전광역시 유성구 동서대로 125 (덕명동)
(72) 발명자
김은기
대전광역시 유성구 어은로 57, 117동 502호(어은동, 한빛아파트)
선설희
대전광역시 유성구 어은로 57, 118동 1404호(어은동, 한빛아파트)
(뒷면에 계속)
(74) 대리인
특허법인충정

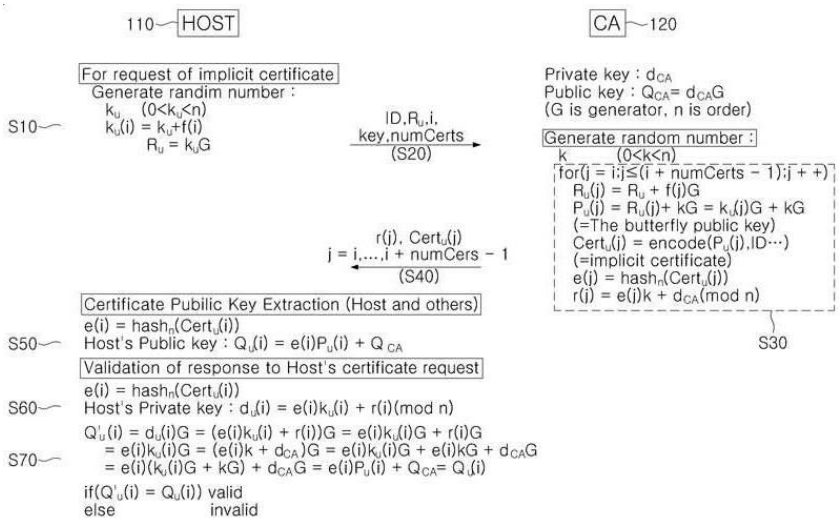
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 **키 확장 방식을 적용한 묵시적 인증서 발급 방법 및 시스템**

(57) 요 약

본 발명은 키 확장 방식을 적용한 묵시적 인증서 발급 방법 및 시스템에 관한 것으로서, 특히, 본 발명의 묵시적 인증서 발급 방법은, 호스트에서 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 인증기관의 서버으로 전송하는 단계; 상기 호스트에서, 상기 인증기관의 서버가 상기 인증서 발급 요청 정보에 따라 발행한 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 포함하는 발급정보를 수신하는 단계; 및 상기 호스트에서, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하는 단계를 포함한다.

대 표 도



(52) CPC특허분류
H04L 9/30 (2013.01)
H04L 9/3236 (2013.01)

(72) 발명자
유권정
대전광역시 유성구 학하중앙로49번길 12 (덕명동)
성정기
역시 유성구 동서대로 158-11, B동 204호(덕명동, HB학사)

서혜인
대전광역시 서구 관저동로99번길 2, 302호 (관저동, 보라빌)

이 발명을 지원한 국가연구개발사업

과제고유번호	R0003847
부처명	산업통상자원부
연구관리전문기관	(재)대전지역사업평가단
연구사업명	지역주력산업육성사업
연구과제명	IEEE p1609. 2 규격의 보안기능 고속처리와 DSRC 연동형 차량 내 V2X 융합장치 개발
기 여 율	1/1
주관기관	(주)에세텔
연구기간	2015.05.01 ~ 2017.04.30

명 세 서

청구범위

청구항 1

호스트에서 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 인증기관의 서버로 전송하는 단계;

상기 호스트에서, 상기 인증기관의 서버가 상기 인증서 발급 요청 정보에 따라 발행한 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 포함하는 발급정보를 수신하는 단계; 및

상기 호스트에서, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하는 단계

를 포함하는 것을 특징으로 하는 묵시적 인증서 발급 방법.

청구항 2

제1항에 있어서,

상기 전송하는 단계는,

상기 호스트가 랜덤값(k_u)을 생성하는 단계;

상기 호스트가 랜덤값(k_u)에 기초한 제너레이터 관련 정보(R_u)를 생성하는 단계; 및

상기 호스트가 ID, 제너레이터 관련 정보(R_u), 확장함수에 사용될 키(key)와 임의의 초기 입력값(양의 정수 i) 및 요청 인증서 개수(numCerts)를 포함하는 상기 인증서 발급 요청 정보를 생성하는 단계

를 포함하는 것을 특징으로 하는 묵시적 인증서 발급 방법.

청구항 3

제1항에 있어서,

상기 인증서 발급 요청 정보를 수신하는 상기 인증기관의 서버는,

랜덤값(k)를 생성하고, 상기 인증서 발급 요청 정보에 포함된 초기 입력값(양의 정수 i)에 따라, 상기 인증서 발급 요청 정보에 포함된 제너레이터 관련 정보(R_u), 소정의 키(key)와 파라미터(i)를 포함하는 확장함수(f(j)), 및 제너레이터값(G)을 이용하여, 버터플라이 공개키(P_u)를 생성하고 버터플라이 공개키(P_u)와 상기 인증서 발급 요청 정보에 포함된 ID를 인코딩하여 해당 묵시적 인증서를 생성하되,

j=i 부터 i+numCerts-1까지 반복하여 numCerts 개수만큼의 상기 복수의 묵시적 인증서를 생성하여 상기 호스트로 전송하는 것을 특징으로 하는 묵시적 인증서 발급 방법.

청구항 4

제1항에 있어서,

상기 개인키와 공개키를 각각 생성하는 단계는,

각각의 묵시적 인증서에 대하여 생성한 해쉬값(e(i)), 각각의 묵시적 인증서에 포함된 버터플라이 공개키(P_u), 및 상기 인증기관의 공개키를 이용하여 상기 호스트의 공개키를 생성하는 단계; 및

상기 해쉬값(e(i)), 랜덤값(k_u)와 확장함수(f(i))를 기초로 생성한 값(k_u(i)), 및 상기 인증기관의 서버로부터 상기 복수의 묵시적 인증서와 함께 수신한 암호화값(r(i))을 이용하여 상기 호스트의 개인키를 생성하는 단계

를 포함하는 것을 특징으로 하는 묵시적 인증서 발급 방법.

청구항 5

제4항에 있어서,

상기 암호화값(r(i))은, 상기 인증기관의 서버에서 랜덤값(k), 상기 복수의 묵시적 인증서 각각의 해쉬값(e(j)), 및 상기 인증기관의 개인키(d_{CA})를 이용하여 생성된 것을 특징으로 하는 묵시적 인증서 발급 방법.

청구항 6

요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 전송하는 호스트; 및

상기 인증서 발급 요청 정보를 수신하여 상기 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 생성하여 상기 호스트로 전송하는 인증기관의 서버를 포함하고,

상기 호스트는, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하는 것을 특징으로 하는 묵시적 인증서 발급 시스템.

청구항 7

호스트에서 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 인증기관의 서버로 전송하는 기능;

상기 호스트에서, 상기 인증기관의 서버가 상기 인증서 발급 요청 정보에 따라 발행한 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 포함하는 발급정보를 수신하는 기능; 및

상기 호스트에서, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하는 기능

를 실현하기 위한 컴퓨터로 읽을 수 있는 코드가 기록된 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 인증서 발급 방법 및 시스템에 관한 것으로서, 특히, 키 확장 방식을 적용한 묵시적 인증서 발급 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 묵시적 인증서 발급을 위하여 일반적으로 ECQV(Elliptic Curve Qu-Vanstone) 방식이 사용되고 있다. ECQV 인증서 발급 과정은, 호스트가 ID(identification)와 랜덤값(k)에 기초한 정보(Ru)를 전송하면, 인증기관의 CA(Certificate Authority) 서버는 수신한 ID와 수신 정보를 이용하여 생성한 매개값 Pu를 인코딩함으로써 묵시적 인증서를 호스트로 발급한다. 호스트는 인증서를 수신하여 공개키와 개인키를 계산하고, 인증이 필요한 정보의 송수신에 이용할 수 있다.

[0003] 최근 들어, 차량에서는 다른 차량이나 인터넷 상의 서버, 사물 기기 등 다양한 인프라에 포함된 많은 기기들과 통신하여 텔레매틱스 등 편리한 응용 서비스들을 지원하기 위한 V2X(Vehicle to Everything) 통신이 발전하고 있다.

[0004] 이와 같은 V2X 통신에서는 다양한 주위의 기기들과 통신하기 위하여 다수의 인증서가 요구된다. 그러나, 위와 같은 종래의 ECQV 인증서 발급 방법과 같이, 각각의 인증서 발급 요청 시마다 하나씩 인증서를 발급해 주고 각각의 인증서를 별도로 관리해야 하는 방식은, 대역폭이 작으며 메모리가 충분하지 않고 빠른 처리가 요구되는 V2X 통신 환경에 적합하지 않다는 문제점이 있다.

[0005] 인증서 발급 관련 종래 기술의 문헌으로서 한국특허공개번호 제10-2016-0038091호 (2016.04.07) 등이 참조될 수 있다.

발명의 내용

해결하려는 과제

[0006] 따라서, 본 발명은 상술한 문제점을 해결하기 위하여 안출된 것으로, 본 발명의 목적은, ECQV와 같은 묵시적 인증서 발급 방식에 Butterfly Key Expansion 알고리즘과 같은 키 확장 방식을 적용한 묵시적 인증서 발급 방법

및 시스템을 제공하는 데 있다.

과제의 해결 수단

[0007] 먼저, 본 발명의 특징을 요약하면, 상기의 목적을 달성하기 위한 본 발명의일면에 따른 묵시적 인증서 발급 방법은, 호스트에서 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 인증기관의 서버로 전송하는 단계; 상기 호스트에서, 상기 인증기관의 서버가 상기 인증서 발급 요청 정보에 따라 발행한 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 포함하는 발급정보를 수신하는 단계; 및 상기 호스트에서, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하는 단계를 포함한다.

[0008] 상기 전송하는 단계는, 상기 호스트가 랜덤값(k_u)을 생성하는 단계; 상기 호스트가 랜덤값(k_u)에 기초한 제너레이터 관련 정보(R_u)를 생성하는 단계; 및 상기 호스트가 ID, 제너레이터 관련 정보(R_u), 확장함수에 사용될 키(key)와 임의의 초기 입력값(양의 정수 i) 및 요청 인증서 개수(numCerts)를 포함하는 상기 인증서 발급 요청 정보를 생성하는 단계를 포함한다.

[0009] 상기 인증서 발급 요청 정보를 수신하는 상기 인증기관의 서버는, 랜덤값(k)를 생성하고, 상기 인증서 발급 요청 정보에 포함된 초기 입력값(양의 정수 i)에 따라, 상기 인증서 발급 요청 정보에 포함된 제너레이터 관련 정보(R_u), 소정의 키(key)와 파라미터(i)를 포함하는 확장함수(f(j)), 및 제너레이터값(G)을 이용하여, 버터플라이 공개키(P_u)를 생성하고 버터플라이 공개키(P_u)와 상기 인증서 발급 요청 정보에 포함된 ID를 인코딩하여 해당 묵시적 인증서를 생성하되, j=i 부터 i+numCerts-1까지 반복하여 numCerts 개수만큼의 상기 복수의 묵시적 인증서를 생성하여 상기 호스트로 전송한다.

[0010] 상기 개인키와 공개키를 각각 생성하는 단계는, 각각의 묵시적 인증서에 대하여 생성한 해쉬값(e(i)), 각각의 묵시적 인증서에 포함된 버터플라이 공개키(P_u), 및 상기 인증기관의 공개키를 이용하여 상기 호스트의 공개키를 생성하는 단계; 및 상기 해쉬값(e(i)), 랜덤값(k_u)와 확장함수(f(i))를 기초로 생성한 값(k_u(i)), 및 상기 인증기관의 서버로부터 상기 복수의 묵시적 인증서와 함께 수신한 암호화값(r(i))을 이용하여 상기 호스트의 개인키를 생성하는 단계를 포함한다.

[0011] 상기 암호화값(r(i))은, 상기 인증기관의 서버에서 랜덤값(k), 상기 복수의 묵시적 인증서 각각의 해쉬값(e(j)), 및 상기 인증기관의 개인키(d_{CA})를 이용하여 생성된다.

[0012] 또한, 본 발명의 다른 일면에 따른 묵시적 인증서 발급 시스템은, 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 전송하는 호스트; 및 상기 인증서 발급 요청 정보를 수신하여 상기 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 생성하여 상기 호스트로 전송하는 인증기관의 서버를 포함하고, 상기 호스트는, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성한다.

[0013] 그리고, 본 발명의 또 다른 일면에 따른 컴퓨터로 읽을 수 있는 코드가 기록된 기록 매체는, 호스트에서 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 인증기관의 서버로 전송하는 기능; 상기 호스트에서, 상기 인증기관의 서버가 상기 인증서 발급 요청 정보에 따라 발행한 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 포함하는 발급정보를 수신하는 기능; 및 상기 호스트에서, 상기 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 상기 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하는 기능을 실현할 수 있다.

발명의 효과

[0014] 본 발명에 따른 키 확장 방식을 적용한 묵시적 인증서 발급 방법 및 시스템에 따르면, 호스트는 ID, 확장함수에 사용될 키(key)/초기 입력값(i), 요청 인증서 개수(numCerts)와 랜덤값(k_u)에 기초한 제너레이터 관련 정보(R_u)를 전송하여, 한쌍의 R_u와 k_u에 대응된 다수의 묵시적 인증서를 빠르게 발급받을 수 있으며, 차량의 V2X 통신과 같이 대역폭이 작고 다수의 인증서를 필요로 하는 차량 등의 통신환경에서 기존의 ECQV 방식보다 메모리의 효율성을 더욱 높일 수 있다.

도면의 간단한 설명

[0015] 도 1은 호스트와 인증 기관 CA 서버 간의 상호 작용을 통해 묵시적 인증서를 발급하는 본 발명의 일 실시예에 따른 인증서 발급 시스템을 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 인증서 발급 시스템에서 인증 기관 CA 서버의 확장된 복수의 인증서 발급과 호스트의 확장된 공개키와 개인키의 생성에 대한 개념을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0016] 이하에서는 첨부된 도면들을 참조하여 본 발명에 대해서 자세히 설명한다. 이때, 각각의 도면에서 동일한 구성요소는 가능한 동일한 부호로 나타낸다. 또한, 이미 공지된 기능 및/또는 구성에 대한 상세한 설명은 생략한다. 이하에 개시된 내용은, 다양한 실시 예에 따른 동작을 이해하는데 필요한 부분을 중점적으로 설명하며, 그 설명의 요지를 흐릴 수 있는 요소들에 대한 설명은 생략한다. 또한 도면의 일부 구성요소는 과장되거나 생략되거나 또는 개략적으로 도시될 수 있다. 각 구성요소의 크기는 실제 크기를 전적으로 반영하는 것이 아니며, 따라서 각각의 도면에 그려진 구성요소들의 상대적인 크기나 간격에 의해 여기에 기재되는 내용들이 제한되는 것은 아니다.

[0017] 도 1은 호스트(110)와 인증 기관 CA 서버(120) 간의 상호 작용을 통해 묵시적 인증서를 발급하는 본 발명의 일 실시예에 따른 인증서 발급 시스템을 설명하기 위한 도면이다.

[0018] 도 1을 참조하면, 본 발명의 일 실시예에 따른 인증서 발급 시스템은, 호스트(110)와 인증기관의 CA(Certificate Authority) 서버(120)를 포함한다.

[0019] 호스트(110)는 CA 서버(120)로 인증서 발급을 요청하여 발급받은 인증서를 이용하여 개인키와 공개키를 생성하고 보유한다. 호스트(110)는 개인키와 공개키를 이용하여 상대 기기와 송수신 데이터를 암호화/복호화하여 신뢰성있는 통신이 이루어지도록 한다.

[0020] 특히, 본 발명에서는 ECQV와 같은 기존의 묵시적 인증서 발급 방식을 개선하여 이에 Butterfly Key Expansion 알고리즘과 같은 키 확장 방식을 적용한다. 즉, 호스트(110)가 요청 인증서 개수(numCerts)를 포함한 인증서 발급 요청 정보를 전송하고, 이를 수신하는 CA 서버(120)는 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 생성하여 호스트(110)로 전송한다. 호스트(110)는 복수의 묵시적 인증서를 이용하여 각각의 묵시적 인증서에 대응되는 요청 인증서 개수(numCerts)만큼의 개인키와 공개키를 각각 생성하고 보유한다. 이에 따라 차량의 V2X 통신과 같이 대역폭이 작고 다수의 인증서를 필요로 하는 통신환경에서 기존의 ECQV 방식보다 메모리의 효율성을 더욱 높일 수 있게 된다.

[0021] 여기서, 호스트(110) 및 호스트(110)와 통신하는 대상 기기는, WAVE(Wireless Access in Vehicular Environment) 무선 통신과 같은 V2X 통신을 지원하는 차량 이외에도, 경우에 따라 데스크탑 PC, 스마트폰, 음성/영상 전화 통화가능한 웨어러블 디바이스, 테블릿 PC, 노트북 PC 등 유선 인터넷 통신 또는 무선 통신(예, WiFi, WiBro, WCDMA, LTE 등)이 가능한 모든 전자 기기를 포함할 수 있다.

[0022] 또한, 인증 기관은 인증서 생성, 발급, 등록, 관리 등을 중계하기 위한 기관으로서, 예를 들어, 금융사(예, 은행, 증권사 등) 등 등록기관, 또는 코스콤(주), 한국정보인증(주) 등 전문 인증기관 등을 포함할 수 있다.

[0023] 이하에서 설명하는 호스트(110)와 인증기관의 CA 서버(120) 등 본 발명의 일 실시예에 따른 인증서 발급 시스템의 동작을 위한 기능을 실현하기 위하여, 반도체 프로세서와 같은 하드웨어, 응용 프로그램과 같은 소프트웨어, 또는 이들의 결합으로 구현될 수 있음을 미리 밝혀 둔다.

[0024] 이하 도 1의 각부 세부 항목을 참조하여, CA 서버(120)에서의 복수의 묵시적 인증서의 발행 및 호스트(110)에서의 각각의 묵시적 인증서에 대응되는 요청 인증서 개수(numCerts)만큼의 개인키와 공개키의 생성 과정에 대하여 자세히 설명한다.

[0025] 먼저, 인증서 발급 요청을 위하여, 호스트(110)는 랜덤값(k_u)을 생성하며, 랜덤값(k_u)에 기초한 제너레이터(generator) 관련 정보($R_u=k_uG$)를 생성한다(S10). G 제너레이터(generator)가 생성한 임의의 값(제너레이터값)이다.

[0026] 여기서, $0 < k_u < n$이고, n은 자연수로서 mod(modular) 오더(order)를 나타낸다. 또한, $k_u(i)$는 확장함수 f(i)

와 연관되어 있으며, "$k_u(i) = k_u + f(i)$"인 관계를 이용한다. 확장함수 f(j)는 AES(Advanced Encryption Standard)와 같은 암호화 함수로서, 소정의 키(key)와 파라미터(i)에 따라 해당 확장 함수값이 산출되도록 한다. 하기하는 바와 같이 확장함수 f(i)는 요청 인증서 개수(numCerts)만큼 반복하여 키(key) (정보)를 확장하기 위한 함수로 이용된다.

[0027] 다음에, 호스트(110)는 인증기관에 인증서 발급을 요청하기 위하여, ID(Identification), 제너레이터 관련 정보(R_u), 확장함수에 사용될 키(key)와 임의의 초기 입력값(양의 정수 i) 및 요청 인증서 개수(numCerts)를 포함하는 인증서 발급 요청 정보를 생성하여 CA 서버(120)로 전송한다(S20).

[0028] 이에 따라, CA 서버(120)는, 위와 같은 인증서 발급 요청 정보를 수신하면, 랜덤값(k)을 생성하고 수신 정보를 함께 이용하여 요청 인증서 개수(numCerts)만큼의 서로 구분되는 복수의 묵시적 인증서를 생성하여(S30) 호스트(110)로 전송한다(S40). 여기서, $0 < k < n$이고, n은 자연수로서 mod(modular) 오더(order)를 나타낸다. CA 서버(120)는 미리 자신의 개인키(d_{CA})와 공개키($Q_{CA}= d_{CA}G$)를 가지고 있으며, G는 제너레이터(generator)가 생성한 임의의 값(제너레이터값)이다.

[0029] 복수의 묵시적 인증서를 생성하기 위하여(S30), CA 서버(120)는, 상기 인증서 발급 요청 정보에 포함된 초기 입력값(양의 정수 i)에 따라, 다음과 같이 j=i 부터 i+numCerts-1까지 반복하여 numCerts 개수만큼의 복수의 묵시적 인증서를 생성한다.

[0030] 이를 위하여, 먼저, CA 서버(120)는, 상기 인증서 발급 요청 정보에 포함된 제너레이터 관련 정보(R_u), 소정의 키(key)와 파라미터(i)를 포함하는 확장함수(f(j)), 및 제너레이터값(G)을 이용하여, 식 "$R_u(j) = R_u + f(j)G$"과 "$P_u(j) = R_u(j) + kG = k_u(j)G + kG$"에 따라, 버터플라이 공개키($P_u$)를 생성한다. 또한, CA 서버(120)는, 버터플라이 공개키(P_u)와 상기 인증서 발급 요청 정보에 포함된 ID를 인코딩하여 해당 j번째 묵시적 인증서 $Cert_u(j)$를 생성한다. 여기서, 위의 식 "$k_u(i) = k_u + f(i)$"에 따라(S10 참조), "$R_u(i) = k_u(i)G = (k_u + f(i))G = k_uG + f(i)G = R_u + f(i)G$" 가 되므로, "$R_u(j) = k_u(j)G$" 가 되는 과정을 이용하였다.

[0031] CA 서버(120)는 위와 같은 과정을 j=i 부터 i+numCerts-1까지 반복하여 numCerts개의 묵시적 인증서 $Cert_u(j)$ (j=i ~ i+numCerts-1)을 생성하면, 소정의 암호화값(r(j))과 함께, 호스트(110)로 전송한다(S40). 암호화값(r(j))은 위와 같은 랜덤값(k), 상기 복수의 묵시적 인증서 $Cert_u(j)$ 각각의 해쉬값($e(j) = hash_nCert_u(j)$)), 및 인증기관의 개인키(d_{CA})를 이용하여, 식 "$r(j) = e(j)k + d_{CA}(\mathrm{mod}\ n)$"에 따라 산출될 수 있다.

[0032] 도 2와 같이, CA 서버(120)는 호스트(110)로부터의 요청 인증서 개수(numCerts)와 랜덤값(k_u)에 기초한 제너레이터 관련 정보(R_u)에 따라, 한쌍의 R_u와 k_u에 대응하여 복수개로 확장된 다수의 묵시적 인증서 $Cert_u(j)$를 빠르게 발급하게 된다.

[0033] 한편, 호스트(110)는 CA 서버(120)로부터 위와 같은 numCerts개의 묵시적 인증서 $Cert_u(j)$ (j=i ~ i+numCerts-1), 암호화값(r(j))을 수신하면, numCerts개로 확 장된 자신의 공개키($Q_u(i)$)와 개인키($d_u(i)$)를 각각 생성한다.

[0034] 즉, 호스트(110)는, 예를 들어, 각각의 묵시적 인증서 $Cert_u(i)$에 대하여 생성한 해쉬값($e(i) = hash_nCert_u(i)$)), 각각의 묵시적 인증서 $Cert_u(i)$에 포함된 버터플라이 공개키($P_u(i)$), 및 인증기관의 공개키 Q_{CA}를 이용하여, 식 "$Q_u(i) = e(i)\ P_u(i) + Q_{CA}$"에 따라 자신의 공개키 $Q_u(i)$를 생성할 수 있다(S50).

[0035] 도 2와 같이, 호스트(110)는 발급받은 다수의 묵시적 인증서 $Cert_u(i)$에 따라 인증기관의 공개키 Q_{CA}를 N=numCerts개 확장한 공개키 $Q_u(i)$를 생성하고 보유할 수 있게 된다.

[0036] 또한, 호스트(110)는, 예를 들어, 각각의 묵시적 인증서 $Cert_u(i)$에 대하여 생성한 해쉬값($e(i) = hash_nCert_u(i)$)), 랜덤값(k_u)와 확장함수(f(i))를 기초로 생성한 값($k_u(i)$) (S10 참조, $k_u(i) = k_u + f(i)$), 및 CA 서버(120)로부터 복수의 묵시적 인증서와 함께 수신한 암호화값(r(i))을 이용하여, 식 "$d_u(i) = e(i)\ k_u(i) + r(i)(\mathrm{mod}\ n)$"에 따라 자신의 개인키 $d_u(i)$를 생성할 수 있다(S60).

[0037] 도 2와 같이, 호스트(110)는 발급받은 다수의 묵시적 인증서 $Cert_u(i)$에 따라 인증기관의 개인키 d_{CA}를 N=numCerts개 확장한 공개키 $d_u(i)$를 생성하고 보유할 수 있게 된다.

[0038] 하기의 식과 같이 개인키 $d_u(i)$와 G를 곱한값 $Q'_u(i) = d_u(i)G$가 위와 같은 관계를 이용하면 공개키 $Q_u(i)$가 되므로(S70), 이러한 유효한 개인키 $d_u(i)$와 공개키 $Q_u(i)$를 이용하여, 호스트(110)는 통신 상대 기기와 송신 데이터를 암호화하여 전송할 수 있고, 수신 데이터를 복호화하여 신뢰성있는 통신을 수행할 수 있게 된다.

[0039]
$$\begin{aligned} \boldsymbol{Q'_u(i)} &= d_u(i)G = (e(i)k_u(i) + r(i))\mathrm{G} = e(i)k_u(i)G + r(i)G \\ &= e(i)k_u(i)G + (e(i)k + d_{CA})G = e(i)k_u(i)G + e(i)kG + d_{CA}G \\ &= e(i)(k_u(i)G + kG) + d_{CA}G = \mathrm{e}(i)P_u(i) + Q_{CA} = \boldsymbol{Q_u(i)} \end{aligned}$$

[0040] 이와 같은 본 발명의 일 실시예에 따른 호스트(110)와 CA 서버(120) 등의 입출력 데이터 처리에 사용되는 기능은 컴퓨터 등 장치로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하며, 이와 같은 기록 매체와 컴퓨터 등 장치의 결합으로 기능 수행에 필요한 데이터나 정보를 입력하거나 출력하고 디스플레이하도록 구현할 수 있다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치, 하드 디스크, 이동형 저장장치 등이 있으며 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크(예, 인터넷, 이동통신 네트워크 등)로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장된 형태가 가능하며 네트워크를 통해 실행될 수도 있다.

[0041] 상술한 바와 같이, 본 발명에 따른 키 확장 방식을 적용한 묵시적 인증서 발급 시스템은, ECQV와 같은 묵시적 인증서 발급 방식에 Butterfly Key Expansion 알고리즘과 같은 키 확장 방식을 적용함으로써, 호스트는 ID, 확장함수에 사용될 키(key)/초기 입력값(i), 요청 인증서 개수(numCerts)와 랜덤값(k_u)에 기초한 제너레이터 관련 정보(R_u)를 전송하여, 한쌍의 R_u와 k_u에 대응된 다수의 묵시적 인증서를 빠르게 발급받을 수 있으며, V2X 통신과 같이 다수의 인증서를 필요로 하는 차량 등의 통신환경에서 기존의 ECQV 방식보다 메모리의 효율성을 더욱 높일 수 있다.

[0042] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

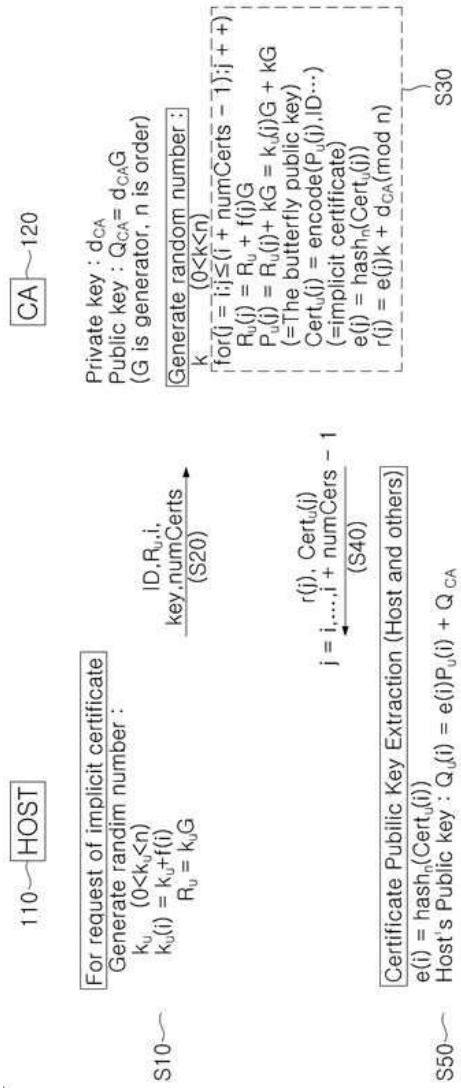
부호의 설명

[0043] 호스트(110)

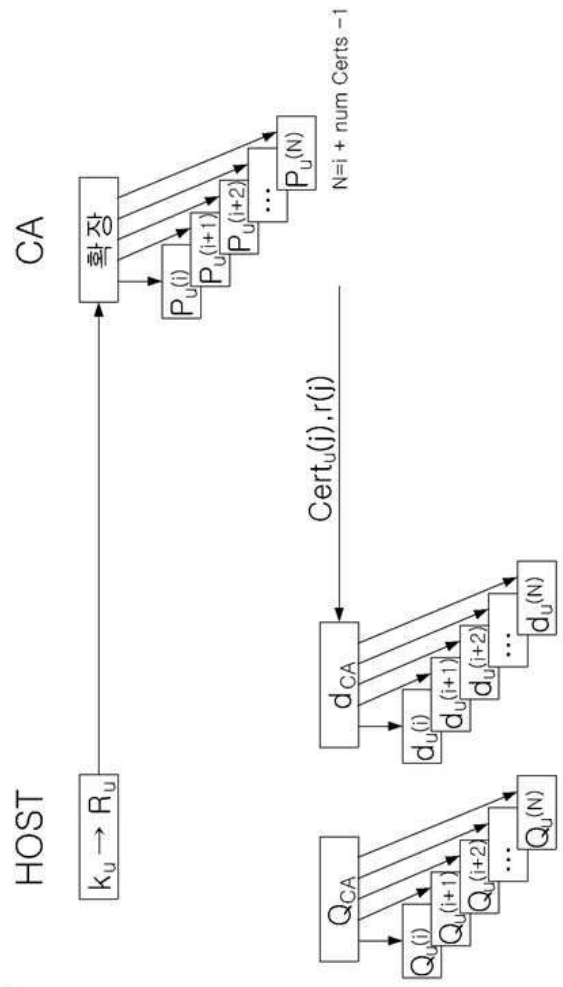
CA 서버(120)

도면

도면1


110
HOST
CA
120
Private key : d_{CA}
Public key : $Q_{CA} = d_{CA}G$
(G is generator, n is order)
For request of implicit certificate
Generate randim number :
k_u $(0<k_u<n)$
$k_u(i) = k_u+f(i)$
$R_u = k_uG$
S10
ID,R_u,i,
key,numCerts
(S20)
Generate random number :
k $(0<k<n)$
for(j = i;j≤(i + numCerts − 1);j++)
$R_u(j) = R_u + f(j)G$
$P_u(j) = R_u(j)+ kG = k_u(j)G + kG$
(=The butterfly public key)
$Cert_u(j)$ = encode($P_u(j)$,ID…)
(=implicit certificate)
e(j) = $hash_n(Cert_u(j))$
r(j) = e(j)k + d_{CA}(mod n)
S30
r(j), $Cert_u(j)$
j = i,…,i + numCers − 1
(S40)
Certificate Pubilic Key Extraction (Host and others)
e(i) = $hash_n(Cert_u(i))$
Host's Public key : $Q_u(i) = e(i)P_u(i) + Q_{CA}$
S50
Validation of response to Host's certificate request
e(i) = $hash_n(Cert_u(i))$
Host's Private key : $d_u(i) = e(i)k_u(i) + r(i)$(mod n)
S60
$Q'_u(i) = d_u(i)G = (e(i)k_u(i) + r(i))G = e(i)k_u(i)G + r(i)G$
$= e(i)k_u(i)G = (e(i)k + d_{CA})G = e(i)k_u(i)G + e(i)kG + d_{CA}G$
$= e(i)(k_u(i)G + kG) + d_{CA}G = e(i)P_u(i) + Q_{CA} = Q_u(i)$
S70
if($Q'_u(i) = Q_u(i)$) valid
else invalid

도면2


HOST
CA
$k_U \rightarrow R_U$
확장
$P_U(i)$
$P_U(i+1)$
$P_U(i+2)$
...
$P_U(N)$
N=i + num Certs -1
$Cert_U(i), r(j)$
d_{CA}
$d_U(i)$
$d_U(i+1)$
$d_U(i+2)$
...
$d_U(N)$
Q_{CA}
$Q_U(i)$
$Q_U(i+1)$
$Q_U(i+2)$
...
$Q_U(N)$