(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년11월07일
(11) 등록번호 10-1459802
(24) 등록일자 2014년11월03일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01) *H04L 12/22* (2006.01)
H04W 12/06 (2009.01)
(21) 출원번호 10-2009-7013796
(22) 출원일자(국제) 2007년11월30일
심사청구일자 2012년11월13일
(85) 번역문제출일자 2009년07월01일
(65) 공개번호 10-2009-0095630
(43) 공개일자 2009년09월09일
(86) 국제출원번호 PCT/US2007/086122
(87) 국제공개번호 WO 2008/127447
국제공개일자 2008년10월23일
(30) 우선권주장
11/607,720 2006년12월01일 미국(US)
(56) 선행기술조사문헌
US06367009 B1*
US20060174323 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
메드빈스키, 겐나디
미국 98052-6399 워싱톤주 레드몬드 원 마이크로소프트 웨이
나이스, 니르
미국 98052-6399 워싱톤주 레드몬드 원 마이크로소프트 웨이
(뒷면에 계속)
(74) 대리인
제일특허법인

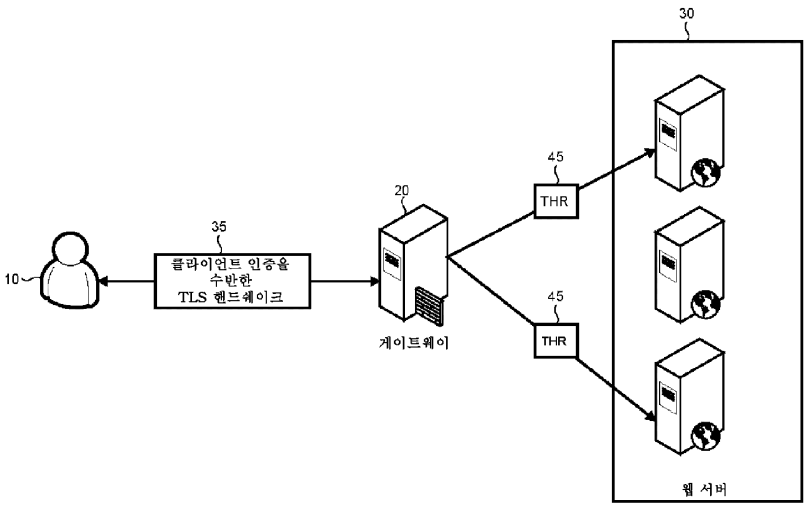
전체 청구항 수 : 총 11 항 심사관 : 양종필

(54) 발명의 명칭 **암호화 증명의 재검증에 기반을 둔 인증 위임**

(57) 요 약

엔터티들의 체인 내에서 인증을 위임하는 방법은, 사용자가 원하는 서버로의 액세스를 필요로 하는 경우에, 게이트웨이 장치와 사용자 간의 TLS 핸드쉐이크의 적어도 일부의 기록에 의존한다. 이 방법은 (1) 액세스가 요구되는 서버(이 경우 서버는 인증을 확인하기 위하여 기록된 부분을 재검증함), 또는 제3자 엔터티(이 경우 제3자 엔터티는 인증을 확인하고, 자격증명을 이용하여 서버에게 사용자로서 인증될 게이트웨이 서버에 그 자격증명을 제공함)에게 전달되는 TLS 핸드쉐이크의 기록된 부분 내의 암호화 증명의 재검증에 의존한다.

대 표 도



(72) 발명자

쉬란, 토머

미국 98052-6399 워싱톤주 레드몬드 원 마이크로소프트 웨이

텝플리트스키, 알렉산더

미국 98052-6399 워싱톤주 레드몬드 원 마이크로소프트 웨이

특허청구의 범위

청구항 1

게이트웨이를 통해 서비스 제공자에 액세스하는 클라이언트에 대한 인증 위임(authentication delegation)의 방법으로서,

클라이언트와 게이트웨이 간에 클라이언트 인증을 수반하는 TLS 핸드쉐이크(a TLS handshake with client authentication)를 수행하는 단계 - 상기 클라이언트 인증을 수반하는 TLS 핸드쉐이크는 복수의 메시지의 교환을 지정(specify)하는 프로토콜에 의해 정의됨 - ;

상기 클라이언트가 상기 게이트웨이에게 인증되었음을 나타내기 위해 상기 TLS 핸드쉐이크의 메시지들의 적어도 일부를 기록하는 단계 - 상기 TLS 핸드쉐이크의 메시지들의 적어도 일부는 상기 TLS 핸드쉐이크에 의해 교환되기로 상기 프로토콜에서 지정된 메시지들을 포함하고 상기 프로토콜에서 지정된 모든 메시지들은 인증서 검증 메시지까지를 포함하며, 상기 TLS 핸드쉐이크의 상기 메시지들의 상기 적어도 일부는 상기 클라이언트와 상기 게이트웨이 간에 교환됨 - ; 및

상기 인증서 검증 메시지까지를 포함하는 상기 모든 메시지들의 상기 기록을 상기 게이트웨이로부터 상기 서비스 제공자에게 제공하는 단계 - 상기 제공되는 모든 메시지들은 디지털 서명됨 -

를 포함하고,

상기 서비스 제공자로의 액세스는 상기 클라이언트와 상기 게이트웨이 간에 교환된 상기 TLS 핸드쉐이크의 메시지들의 상기 적어도 일부에 기초하는, 인증 위임 방법.

청구항 2

제1항에 있어서, 상기 서비스 제공자는 상기 클라이언트와 상기 게이트웨이 간의 인증에는 관여하지 않는, 인증 위임 방법.

청구항 3

제1항에 있어서, 상기 제공하는 단계는 상기 기록을 상기 게이트웨이로부터 상기 서비스 제공자로 직접 제공하는, 인증 위임 방법.

청구항 4

삭제

청구항 5

제1항에 있어서, 상기 TLS 핸드쉐이크를 수행하는 단계는 상기 TLS 핸드쉐이크 내의 메시지들에 타임스탬프(timestamp) 데이터를 삽입하는 단계를 더 포함하는, 인증 위임 방법.

청구항 6

제5항에 있어서, 상기 클라이언트가 상기 타임스탬프 데이터를 삽입하는, 인증 위임 방법.

청구항 7

제5항에 있어서, 상기 게이트웨이가 상기 타임스탬프 데이터를 삽입하는, 인증 위임 방법.

청구항 8

제1항에 있어서, 상기 TLS 핸드쉐이크를 수행하는 단계는 상기 게이트웨이로부터 상기 클라이언트로의 메시지 내에 상기 서비스 제공자에 의해 제공되는 넌스(nonce)를 삽입하는 단계를 더 포함하는, 인증 위임 방법.

청구항 9

제1항에 있어서, 상기 서비스 제공자는 수신된 상기 기록 모두에 대한 메모리를 보유하고, 동일한 기록이 한번

보다 많이 사용되지 않음을 확인하는, 인증 위임 방법.

청구항 10

제1항에 있어서, 상기 TLS 핸드쉐이크를 수행하는 단계는,

클라이언트 인증없이 제1 핸드쉐이크를 수행하는 단계; 및

상기 제1 핸드쉐이크가 성공적으로 완료되면, 클라이언트 인증을 수반한 제2 핸드쉐이크를 수행하는 단계

를 포함하는, 인증 위임 방법.

청구항 11

제10항에 있어서,

상기 클라이언트와 게이트웨이 간의 제2 핸드쉐이크는 상기 제1 핸드쉐이크로부터 도출된 세션 키에 의해 암호화되는, 인증 위임 방법.

청구항 12

제11항에 있어서,

상기 서비스 제공자에게 제공되는 기록은 암호화되지 않는, 인증 위임 방법.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

명 세 서

배 경 기 술

[0001] 한 조직은, 함께 모여 사용자에게 소정의 서비스를 제공하는 엔터티들의 체인을 갖고 있을 수 있다. 데이터, 웹 페이지, 기능적 소프트웨어 동작(functional software operations) 등과 같은 자원들로의 액세스는 알려지고 허가받은 사용자들의 집합으로만 제한될 필요가 있다. 허가받지 않은 사용자와 악의있는 공격자가 컴퓨터 자원

들로의 액세스를 얻는 것을 방지하기 위하여, 웹 자원에 액세스하려고 시도하는 사용자의 신원을 인증하는 데에 사용되는 메커니즘을 포함하는 다양한 액세스 제어 체계가 개발되어 왔다. 특히, 신원 도용 공격(identity theft attacks)(예를 들어, 피싱(phishing), 파밍(pharming))의 증가로 인해, 2중 인증(2-factor authentication, T-FA)이 인기를 얻어가고 있다.

[0002] 2중 인증(T-FA)은 신원과 특권을 확립하기 위해 2가지의 다른 방식을 필요로 하는 임의의 인증 프로토콜이다. 2중 인증의 일반적인 구현은 "대상자가 알고 있는 것(something you know)"(예를 들어, 패스워드 또는 개인 식별 번호)을 하나의 인자로서 사용하고, "대상자가 가지고 있는 것(something you have)"(예를 들어, 신용카드 또는 하드웨어 토큰) 또는 "대상자 자체(something you are)"(예를 들어, 지문 또는 망막 패턴)를 다른 인자로서 사용하는 것이다. 예를 들어, 스마트 카드는 2중 인증을 제공하는 방법들 중 하나이다. 스마트 카드는 하드웨어 토큰의 일례로서, 제공된 데이터에 암호화 기능을 수행하는 것과 같은 다양한 보안 동작을 실행할 수 있는 마이크로프로세서를 전형적으로 포함한다. 통상적으로, 스마트 카드는 인증서-기반 인증을 필요로 하는 프로토콜들에 대해 사용될 수 있는 하나 이상의 ITU-T(International Telecommunications Union) X.509 인증서(및 그 관련 개인 키)를 보유한다. SSL(Secure Socket Layer), TLS(Transport Layer Security), 및 (PKINIT(Public Key Crypography for Initial Authentication in Kerberos)를 수반하는) 커베로스가 그러한 프로토콜들의 예이다.

[0003] 인증서를 사용하기 위하여 스마트 카드가 필수적인 것은 아니다. 많은 장치들(예를 들어, 컴퓨터, 모바일 폰)이 인증서(및 그 관련 개인 키)를 저장하고 사용할 수 있다. 예를 들어, Windows Mobile 5.0은 (SSL 또는 TLS 상에서 실행되는 Exchange Active Sync 프로토콜을 통해) 이메일과 캘린더 정보를 동기화하기 위하여, Exchange 2003 SP2 서버에 인증하기 위하여 인증서를 사용할 수 있다.

[0004] 예를 들어 조직의 내부 네트워크(인트라넷)에 위치하는 웹 서버들로의 웹 기반 액세스를 제공하는 네트워크의 에지에 게이트웨이 장치를 포함하는, 조직의 네트워크의 엔터티들의 체인에 대한 신원 도용 공격을 방지하는 것은 매우 중요하다.

[0005] 인증 위임(authentication delegation)은, 클라이언트가 서버에 대한 인증을 위임하는 것, 또는 더 구체적으로는 자원(또는 서버)에 액세스할 수 있는 제3자 인증 서비스(또는 게이트웨이)가 (본질적으로 사용자를 가장함으로써) 사용자를 대신하여 인증할 수 있게 하는 것으로서 넓게 정의된다. 액세스된 서버는 인증 서비스의 계정보다는, 사용자의 신원에 기초하여 인증 결정을 내릴 것이다.

[0006] 이러한 배경기술은 이하의 발명의 요약 및 상세한 설명에 대한 간략한 배경을 소개하기 위해 제공된다. 이러한 배경기술은 청구되는 발명의 주제를 결정하는 데에 도움을 주도록 의도된 것이 아니며, 또한 청구되는 발명의 주제를 위에서 제시된 단점이나 문제점 중 일부 또는 전부를 해결할 수 있는 구현들로만 제한하는 것으로서 보여지도록 의도된 것도 아니다.

[0007] <발명의 요약>

[0008] 재검증(re-verification) 또는 암호화 증명(cryptographic evidence)에 기반을 둔 인증 위임은, 사용자가 엔터티들의 체인 내에서 특정 서버로의 액세스를 원할 때, 게이트웨이 장치와 사용자 간의 TLS 핸드쉐이크의 적어도 일부의 기록(recording)을 이용한다. 암호화 증명은 (1) 원하는 서버에(이 경우, 서버는 인증을 확인(confirm)하기 위해 기록된 부분을 재검증함), 또는 (2) 제3자 엔터티에(이 경우, 제3자 엔터티는 기록된 부분을 재검증하고, 사용자 자격증명을 그 자격증명을 사용하여 서버에 사용자로서 인증하는 게이트웨이에 제공함으로써 인증을 확인함), TLS 핸드쉐이크의 기록된 부분을 전달함으로써 제공된다. 각각의 경우에서, 서버 및 제3자 엔터티는 사용자와 게이트웨이 장치 간의 인증에는 관여하지 않고서, TLS 핸드쉐이크의 기록된 부분을 이용하여 사용자에게 액세스를 허가할지를 결정한다.

[0009] 다양한 예에서, 암호화 증명은 TLS 핸드쉐이크가 시기적절한 것(즉, 프레쉬(fresh)함)을 보장함으로써 추가의 보안 조치를 제공하기 위해 타임스탬프를 포함한다. 또한, 제3자 엔터티는, 유효한 TLS 핸드쉐이크를 확인하면, 게이크웨이가 예를 들어 PKINIT를 수반하는 커베로스를 이용하여 사용자를 대신하여 원하는 서버에 인증할 수 있게 하는 임시의(즉, 시간 제한적인) 사용자 자격증명을 발행하도록 구성될 수 있다.

[0010] 본 요약은 선택된 개념들을 단순화된 형태로 소개하기 위해 제공된 것이다. 개념들은 이하의 상세한 설명에서 더 설명된다. 본 요약에 설명된 것 이외의 구성요소 및 단계들도 가능하며, 어떠한 구성요소 또는 단계도 필수적으로 요구되지 않는다. 본 요약은 청구되는 발명의 주제의 핵심적인 특징 또는 본질적인 특징을 확인하도록 의도된 것이 아니며, 청구되는 발명의 주제의 범위를 결정하는 데에 도움을 주는 데 사용되도록 의도된 것도 아

니다. 청구되는 발명의 주제는 본 명세서의 임의의 부분에 나타난 단점들의 일부 또는 전부를 해결하는 구현으로만 제한되지 않는다.

실 시 예

[0017] 재검증 또는 암호화 증명에 기초하는 본 인증 위임에 대한 예시적인 문맥은 클라이언트/사용자가 게이트웨이를 통해 하나 이상의 서비스 제공자를 액세스하는 것이다. 그러나, 이 문맥은 단순히 예시적인 것이며, 다른 문맥 및 환경도 적합할 수 있음을 강조한다. 예를 들어, 본 인증 위임은 웹 서버가 백엔드 애플리케이션 또는 데이터베이스에게 사용자로서 인증할 필요가 있을 때, 또는 다르게는 엔터티들의 체인이 존재하고 체인 내의 엔터티들 간에 인증이 필요한 임의의 세팅에서 사용될 수 있다.

[0018] 액세스 문맥에서, 게이트웨이 장치는 웹 서버로의 액세스를 제공하고, 여기에서 사용자는 게이트웨이에, 결국에는 내부 웹 서버에 도달하는 요청을 제출한다. 그러나, 게이트웨이와 웹 서버 둘 다는, 접속하고 있는 사용자가 원하는 자원에 액세스하도록 허용되는지 여부를 판정하기 위하여 전형적으로 소정 형태의 인증을 필요로 한다.

[0019] 게이트웨이가 폼-기반 인증(forms-based authentication, FBA)을 사용하도록 구성된 경우, 그 게이트웨이는 사용자에게 로그온 폼에 사용자명 및 패스워드를 입력할 것을 요구할 것이다. 그러면, 사용자는 폼을 제출하고, 게이트웨이는 사용자의 사용자명 및 패스워드를 수신한다. 그러면, 게이트웨이는 그러한 자격증명을 이용하여 사용자를 대신하여 내부 웹 서버에 인증할 수 있다. 게이트웨이가 패스워드를 수신하여 그것을 자신이 원하는 대로 사용할 수 있기 때문에, 이것은 매우 단순하고 가능하다. 그러나, 일부 인증 체계에서는 이것이 가능하지 않다. 예를 들어, 사용자가 패스워드를 제공하지 않는 인증 체계를 이용하여 게이트웨이에 인증하는 경우, 게이트웨이는 사용자를 대신하여 내부 웹 서버에 인증하기 위하여 재사용할 수 있는 어떠한 자격증명도 갖지 않는다.

[0020] 이 문제에 대한 몇가지 솔루션이 제안되어 있다. 예를 들어, 한가지 솔루션은 "신뢰된 제3자(a trusted third party)"를 포함한다. 여기에서, 신뢰된 제3자는 규정된 집합의 웹 서버들(또는 일반적으로 서비스들)에 대해 모든 사용자를 대신하여 인증하도록 게이트웨이를 "신뢰"하도록 미리 구성된다. 이러한 기술은 게이트웨이(또는 프론트엔드 서버)가 다른 서버들에서 사용하기 위해 클라이언트를 대신하여 티켓을 요청하는 것을 허용하는 프로토콜로서 구현될 수 있다. 그러면, 신뢰된 제3자는 기꺼이 임의의 사용자를 대신하여 게이트웨이에 서비스 티켓을 제공하여, 게이트웨이가 임의의 사용자를 가장(impersonating)할 수 있게 한다.

[0021] 또한, 신뢰된 제3자는 특정한 조건 하에서 서비스 티켓을 제공하도록 구성될 수 있다. 예를 들어, 커베로스 프로토콜에서, 클라이언트는 서비스 티켓을 통하여 게이트웨이에게 인증하고, 커베로스 제약된 위임(Kerberos constrained delegation)은 신뢰된 제3자(키 배포 센터)가 그러한 조건을 부여하도록 구성될 수 있게 하는 방법을 제공한다. 이 경우, 게이트웨이는 (서비스 티켓을 통하여) 요구한 사용자가 정말로 게이트웨이에 인증되었다는 증명을 반드시 제공해야 하고, 이는 시스템의 전체적인 보안을 증가시키는 데에 중요하다. 예를 들어, 그러한 증명을 요구하는 것의 이점은, 사용자가 먼저 게이트웨이에게 적절하게 인증하지 않고서는, 손상된(compromised) 게이트웨이가 그 사용자를 대신하여 서버로의 액세스를 얻지 못할 것이라는 점이다.

[0022] 이러한 제안은 패스워드없이 인증을 해결하는 방식을 제공하긴 하지만, 몇몇 경우에서는 키 배포 센터(Key Distribution Center, KDC) 또는 다른 신뢰된 제3자 엔터티를 관여시키지 않는 인증 위임 모델을 구현하는 것이 바람직할 수 있다. 그러한 모델에서, 게이트웨이는 KDC와의 어떠한 통신도 없이 사용자를 대신하여 내부 웹 서버에게 인증할 것이다. 이러한 종류의 기능성을 제공하는 많은 솔루션들이 존재한다. 예를 들어, 게이트웨이 및 임의의 수의 내부 웹 서버 상에 몇몇 제품이 설치/구성되어, 사용자가 게이트웨이에게 인증하고 나면, 게이트웨이가 내부 웹 서버에 의해 신뢰되는 토큰(HTTP -Hypertext Transfer Protocol, 일부 경우에는 쿠키)을 반환하게 된다. 다른 제안들과 마찬가지로, 이러한 모델의 한가지 문제점은 게이트웨이가 완전히 신뢰되는 엔터티이고, 따라서 시스템의 전체적인 보안을 감소시킨다는 것이다.

[0023] 본 구성은 암호화 증명의 재검증에 기초하는 인증 위임을 제공한다. 게이트웨이(또는 프론트엔드 서버)는 웹 서버(또는 백엔드 서버)로의 액세스를 제공한다. 클라이언트/사용자는 클라이언트 인증을 수반한 TLS 핸드쉐이크(a TLS handshake with client authentication)를 이용하여 게이트웨이에게 인증한다. 그 다음, TLS 핸드쉐이크의 기록(recording), 또는 적어도 사용자가 게이트웨이에게 인증되어 있음을 증명하는 데에 충분한 만큼의 TLS 핸드쉐이크는, (핸드쉐이크의 유효성을 재검증하는) 웹 서버에, 또는 (기록을 검증하고 나면, 웹 서버에게 인증하는 게이트웨이에 사용자 자격증명을 제공하는) 제3자 엔터티에 제공된다.

[0024] 이제, 유사한 참조번호가 유사한 구성요소를 나타내고 있는 도면들을 보면, 도 1은 본 인증 위임이 이용되는 예시적인 네트워크 아키텍처를 도시한 것이다. 클라이언트/사용자 컴퓨터 시스템(10)은 게이트웨이(20)(인증 서버로도 칭해짐)에 동작적으로 결합되어, 클라이언트/사용자(10)와 웹 서버 네트워크(30)(네트워크 서버라고도 칭함) 간의 통신을 허용한다. 게이트웨이(20)는 사용자들을 인증하는 데에 필요한 정보를 포함하는 데이터베이스/디렉토리(도시되지 않음)를 포함한다 (다르게는, 게이트웨이는 네트워크를 통해 외부 사용자 데이터베이스/디렉토리와 통신할 수 있다). 로그온하는 것에 응답하여, 사용자/클라이언트(10)는 우선 참조 번호 35로 표시된 것과 같은 클라이언트 인증을 수반하는 TLS 핸드쉐이크를 통해 게이트웨이(20)에게 인증한다. 클라이언트 인증은 TLS 핸드쉐이크 프로토콜에서 선택사항이기 때문에, 여기에서는 클라이언트 인증이 의도적으로 언급되었음에 유의한다.

[0025] TLS 프로토콜은 인터넷을 통한 통신 프라이버시를 제공하고, 클라이언트/서버 애플리케이션이 도청, 탬퍼링 또는 메시지 위조를 방지하도록 설계된 방식으로 통신하는 것을 허용한다. TLS 핸드쉐이크 프로토콜은 서버와 클라이언트가 서로 인증하고 애플리케이션 프로토콜이 데이터의 첫번째 바이트를 송신 또는 수신하기 전에 암호화 알고리즘 및 암호화 키를 협의할 수 있게 해 준다.

[0026] TLS의 한가지 이점은 이것이 애플리케이션 프로토콜에 독립적이라는 것이다. 따라서, TLS 프로토콜의 상부에, 보다 높은 레벨의 프로토콜이 투명하게(transparently) 계층화될 수 있다. TLS 핸드쉐이크 프로토콜은 다음과 같이 요약될 수 있다: 사용자/클라이언트는 클라이언트 헬로우 메시지를 송신하고, 서버(도 1에서 게이트웨이(20))는 그 클라이언트 헬로우 메시지에 서버 헬로우 메시지로 응답해야 하며, 그렇지 않으면 치명적 오류가 발생할 것이고 접속은 실패할 것이다. 클라이언트 헬로우 및 서버 헬로우는 클라이언트와 서버 간의 보안 강화 능력을 확립하는 데에 사용된다.

[0027] 도 6은 전형적인 TLS 핸드쉐이크 위상 동안의 클라이언트와 서버 간의 메시지 교환을 도시하는 개략적인 메시지 흐름도이다. TLS 프로토콜은 RFC2246의 "TLS protocol, Version 1.0"에 상세하게 기술되어 있으며, 그 개시 내용은 본 명세서에 참조로서 포함된다. 클라이언트/사용자는 클라이언트-서버 관계로 게이트웨이 장치와 통신한다.

[0028] 더 상세하게는, 도 6에 도시된 바와 같이, 실제의 키 교환은 4개까지의 메시지, 즉 서버 인증서, 서버 키 교환, 클라이언트 인증서, 및 클라이언트 키 교환을 이용한다. 새로운 키 교환 방법은 클라이언트와 서버가 공유 비밀(a shared secret)에 동의할 수 있게 하기 위해 이러한 메시지들을 위한 포맷을 지정하고 메시지들의 사용을 규정함으로써 생성될 수 있다. 서버가 인증되어야 하는 경우에, 서버는 헬로우 메시지에 후속하여 자신의 인증서를 송신할 것이다. 또한, 필요하다면 (예를 들어 서버가 인증서를 가지고 있지 않다면, 또는 그 인증서가 서명 전용이라면), 서버 키 교환 메시지가 송신될 수 있다.

[0029] 서버가 인증되는 경우, 서버는 선택된 암호 모음(cipher suite)에 적합하다면, 클라이언트로부터 인증서를 요청할 수 있다. 그 다음, 서버는 핸드쉐이크의 헬로우 메시지 위상이 완료되었음을 나타내는 서버 헬로우 완료(server hello done) 메시지를 송신할 것이다. 그 다음, 서버는 클라이언트 응답을 기다릴 것이다. 서버가 인증서 요청 메시지를 송신한 경우, 클라이언트는 인증서 메시지를 반드시 보내야 한다. 클라이언트 키 교환 메시지가 송신되고, 그 메시지의 내용은 클라이언트 헬로우와 서버 헬로우 간에 선택된 공개 키 알고리즘에 의존할 것이다. 클라이언트가 서명 능력을 갖는 인증서를 보냈다면, 인증서를 명시적으로 검증하기 위하여 디지털-서명된 인증서 검증 메시지(a digitally-signed certificate verify message)가 송신된다.

[0030] 도 1을 다시 보면, 본 설명적인 예시에서, 게이트웨이(20)는 이러한 핸드쉐이크의 일부로서 교환된 데이터의 기록(도 1에서는 참조 번호 45와 함께 THR로서 표시됨)을 생성한다. 더 상세하게는, 기록은 적어도, TLS 핸드쉐이크의 이전 메시지들의 전부에 대한 서명으로 이루어진 인증서 검증 메시지까지의 데이터를 포함하며, 사용자/클라이언트(10)가 인증서에 일치하는 개인 키를 정말로 소유하고 있음을 증명한다.

[0031] 그 다음, TLS 핸드쉐이크 기록(THR)은 사용자/클라이언트(10)이 게이트웨이(20)에게 인증되었다는 인증 증명(즉, 증거)으로서 내부 웹 서버(30)에 직접 제공된다.

[0032] 내부 웹 서버(30)는 클라이언트/사용자(10)와 게이트웨이(20) 간의 인증에 관여하지 않고, 오히려 단순히 클라이언트/사용자(10)와 게이트웨이(20) 간의 인증의 인증 증명(즉, THR)을 제공받아서, 원하는 자원으로의 액세스를 제공할지를 판정한다.

[0033] 제안된 체계는 TLS 핸드쉐이크가 인증서 검증 메시지를 포함하는 경우에만 사용될 수 있음에 유의해야 한다. 이 메시지는 다음 조건들 중 어느 것이라도 참이면 사용되지 않는다.

[0034] 1) TLS 핸드쉐이크가 클라이언트 인증을 포함하지 않는다.

[0035] 2) 클라이언트와 게이트웨이가 (새로운 보안 파라미터를 협의하는 대신에) 이전 TLS 세션을 재개하기로, 또는 현존 세션을 복제하기로 정한다. 이 경우, TLS 핸드쉐이크는 인증서 검증 메시지를 포함하지 않는다 (RFC 2246의 30-31쪽 참조).

[0036] 3) 클라이언트 인증서가 서명 기능을 갖는다 (즉, 고정된 Diffie-Hellman 파라미터를 포함하는 것들을 제외한 모든 인증서). 예를 들어, 암호 그룹 ECDH_ECDSA 및 ECDH_RSA(RFC 4492 참조)은 클라이언트 인증을 지원하지만, 인증서 검증 메시지는 이용하지 않는다.

[0037] 도 2는 도 1에 도시된 예시적인 아키텍처에서 클라이언트/사용자가 웹 서버로의 액세스를 구할 때 수행되는 인증 프로세스에 대한 예시적인 흐름도이다. 클라이언트/사용자가 웹 서버 자원을 원하고 게이트웨이(20)에 액세스하면 프로세스가 시작된다 (단계(200)). 클라이언트/사용자가 웹 서버에 로그인하지 않은 경우, 클라이언트/사용자는 웹 서버가 액세스를 허용하기 전에 반드시 인증되어야 한다.

[0038] 그 다음, 클라이언트/사용자는 원하는 웹 서버 자원을 요청한다 (단계(210)). 그 다음, 클라이언트/사용자를 인증하기 위하여, 게이트웨이(20) 및 클라이언트/사용자(10)는 위에서 상세하게 설명된 방식으로 클라이언트 인증을 수반한 TLS 핸드쉐이크를 수행한다 (단계(220)) (당업자라면, 클라이언트/사용자가 원하는 자원을 요청하기 직전에, 또는 클라이언트/사용자가 자원을 요청한 직후에, 게이트웨이(20)가 클라이언트/사용자(10)를 인증할 수 있음을 알 것이다). TLS 핸드쉐이크의 적어도 일부의 기록이 생성되어 요청된 웹 서버에 제공된다 (단계(230)).

[0039] 웹 서버(30)는 THR을 수신하면, (THR의 인증서 검증 메시지 내의 클라이언트/사용자의 서명의 유효성을 검증함으로써, 그리고 일부 실시예들에서는 THR의 타임스탬프(이하에 설명됨)의 유효성도 검증함으로써) 클라이언트/사용자가 게이트웨이에게 인증되었음을 검증한다 (단계(240)). 클라이언트/사용자가 웹 서버에 액세스하도록 허가받은 것으로 가정하면, THR이 검증된 경우, 요청된 웹 서버로의 액세스가 허여되고(단계(250)), THR이 검증되지 못하면 액세스가 거부된다 (단계(260)).

[0040] 공격자가 THR을 얻을 수 있고 그것을 재사용하여 클라이언트/사용자를 가장하려고 시도하는 경우, 그러한 "리플레이 공격(replay attack)"을 방지하거나 적어도 완화하기 위하여 몇가지 기술 및 메커니즘이 예상된다. 첫번째로, 서버 및/또는 클라이언트/사용자가 그들의 핸드쉐이크 메시지에 시간-관련 데이터(예를 들어, 타임스탬프)를 삽입했다고 가정하면, 서비스 제공자(예를 들어, 내부 웹 서버)는 그것이 "프레쉬(fresh)"한지를 확인하기 위하여 수신된 THR을 검사할 수 있다. 이러한 솔루션은 전형적으로 게이트웨이(20)와 웹 서버(30)(또는 사용자/클라이언트(10)와 웹 서버(30))가 동기화된 클럭을 가지고 있을 것을 필요로 하겠지만, 당업자라면 가능한 우회수단들이 많이 있음을 알 것이다.

[0041] 다르게는, 게이트웨이(20)는 서비스 제공자(웹 서버(30))에게 넌스(nonce)를 요청하고, 그 넌스를 자신이 TLS 핸드쉐이크의 일부로서 사용자/클라이언트(10)에게 송신하는 메시지들 중 하나에 삽입할 수 있다. 그러면, 서비스 제공자는 수신된 THR을 검사하여, 자신이 이전에 생성하여 게이트웨이(20)에 전달한 넌스를 포함하고 있는지를 확인할 수 있다.

[0042] 이러한 두가지 가능성들 각각에서, 게이트웨이(20)(또는 사용자/클라이언트(10))는 TLS 핸드쉐이크 메시지에 소정의 데이터를 삽입한다는 점에 유의한다 (다시 한번, 핸드쉐이크 프로토콜은 근본적으로 데이터 전달 세션의 보안 파라미터를 협의하는 일련의 순서화된 메시지이다). 이러한 데이터의 삽입은 전형적으로 다음과 같은 방식들 중 하나로 행해진다.

[0043] (1) 서버는 서버 헬로우 메시지 내에, 이 메시지의 랜덤 필드의 일부로서 타임스탬프 또는 넌스를 넣는다 (핸드쉐이크 프로토콜의 이러한 양태의 세부사항은 RFC 2246의 섹션 7.4.1.3에서 찾을 수 있다).

[0044] (2) 서버는 서버 헬로우 익스텐션에 타임스탬프 또는 넌스를 넣는다 (세부사항은 RFC 3546의 섹션 2.2에서 찾을 수 있다).

[0045] (3) 클라이언트/사용자는 클라이언트 헬로우 메시지에 그 메시지의 랜덤 필드의 일부로서 타임스탬프를 넣는다 (핸드쉐이크 프로토콜의 이러한 양태의 세부사항은 RFC 2246의 섹션 7.4.1.2에서 찾을 수 있다).

[0046] (4) 클라이언트/사용자는 클라이언트 헬로우 익스텐션에 타임스탬프를 넣는다 (세부사항은 RFC 3546의 섹션 2.1에서 찾을 수 있다).

[0047] 마지막으로, 위에서 설명한 대안들 각각에 더하여, 서비스 제공자(웹 서버(30))는 동일한 THR이 2회 이상 사용되지 않을 것을 보장하기 위하여, 자신이 수신한 모든 THR을 기억할 수 있다. 이러한 메모리는 소정의 공유 저장 또는 통신 메커니즘을 통하여 서비스 제공자들 간에 공유될 수 있다.

[0048] 다른 예시적인 구현에서, 클라이언트와 게이트웨이 간의 통신 채널, 또는 게이트웨이를 손상시키는 공격자에 대항한 추가의 보호를 위해, "이중(dual)" TLS 핸드쉐이크가 사용될 수 있다. 이 경우, 클라이언트/사용자(10) 및 게이트웨이(20)는 클라이언트 인증 없이 제1 TLS 핸드쉐이크를 수행한다. 이러한 제1 TLS 핸드쉐이크가 성공적으로 완료되면, 클라이언트/사용자(10) 및 게이트웨이(20)는 클라이언트 인증을 수반한 제2 TLS 핸드쉐이크를 수행한다. 나중에 증명(THR)으로서 사용될 제2 핸드쉐이크는 클라이언트/사용자(10)와 게이트웨이(20)가 제1 핸드쉐이크로부터 도출한 세션 키에 의해 전송을 위해 암호화된다. 이에 의해, THR은 암호화되지 않고서(즉, 평문으로) 송신되지 않기 때문에 보호되며, 공격자가 게이트웨이(20)를 손상시킬 수 있더라도, THR을 얻기가 더 어려울 것이다.

[0049] 지금까지 논의되고 도 1에 도시된 실시예는 사용자/클라이언트(10), 게이트웨이(20) 및 서비스 제공자(웹 서버(30))를 포함한다. 도 3에 도시되고 이하에 더 상세하게 논의되는 대안적인 실시예는 제3자 엔터티(40)를 이용한다. 이 경우, 서비스 제공자(웹 서버(30))는 제3자 엔터티(40)가 사용자의 실제 신원을 제공할 것으로 "신뢰"한다. 그러한 제3자 엔터티(40)는 커베로스 KDC(S4U2Self + S4U2Proxy에서와 같은) 또는 인증 기관(Certificate Authority, CA)을 포함한다. 도 3에서는 제3자 엔터티(40)가 별개의 엔터티로서 도시되었지만, 일부 구성들에서, 제3자 엔터티(KDC 또는 CA)는 게이트웨이(20)와 동일한 기계 상에 존재할 수 있음에 유의한다.

[0050] 도 3에 도시된 실시예를 더 상세하게 논의하기 전에, 클라이언트와 서비스 간에 공유되는 세션 키를 협의하고 그들 간의 상호 인증을 제공하기 위하여 KDC로서 알려진 신뢰된 제3자의 사용을 포함하는 커베로스 프로토콜을 논의할 것이다.

[0051] 커베로스의 코너-스톤(corner-stone)은 티켓(Ticket) 및 인증자(Authenticator)이다. 티켓은 대칭 키(티켓 세션 키 - 2개의 엔드포인트 간에 공유되는 단 하나의 키가 존재함)를 특정 서비스를 위해 의도된 엔벨로프(공개 메시지) 내에 캡슐화한다. 티켓의 내용은 서비스 주체(service principal)와 발행 KDC 간에 공유되는 대칭 키로 암호화된다. 티켓의 암호화된 부분은 다른 항목들 중에서도 클라이언트 주체 이름을 포함한다. 인증자는 관련된 티켓의 티켓 세션 키를 이용하여 최근에 생성된 것으로 보여질 수 있는 기록이다. 티켓 세션 키는 티켓을 요청한 클라이언트에 의해 알려진다. 인증자의 내용은 관련된 티켓 세션 키로 암호화된다. 인증자의 암호화된 부분은 다른 항목들 중에서도 타임스탬프 및 클라이언트 주체 이름을 포함한다.

[0052] 도 4에 도시된 바와 같이, 커베로스 (V5) 프로토콜은 클라이언트(405)와 KDC(410) 간의, 그리고 클라이언트(405)와 애플리케이션 서버(415) 간의 다음과 같은 메시지 교환으로 이루어진다.

[0053] 인증 서비스(AS) 교환

[0054] 클라이언트는 커베로스 인증 서버(AS)로부터 "초기" 티켓을 얻는데, 이것은 전형적으로 티켓 발급 티켓(Ticket Granting Ticket, TGT)이다. AS-REQ 메시지(420) 및 AS-REP(425) 메시지는 각각 클라이언트와 AS 간의 요청 및 응답 메시지이다.

[0055] 티켓 발급 서비스(TGS) 교환

[0056] 클라이언트는 후속하여 TGT를 이용하여, 커베로스 티켓-발급 서버(TGS)로부터의 특정 서비스를 위한 서비스 티켓을 인증하고 요청한다. TGS-REQ 메시지(430) 및 TGS-REP 메시지(435)는 각각 클라이언트와 TGS 간의 요청 및 응답 메시지이다.

[0057] 클라이언트/서버 인증 프로토콜(AP) 교환

[0058] 그리고, 클라이언트는 서비스 티켓 및 클라이언트의 티켓 세션 키 소유를 증명하는 인증자로 이루어진 AP-REQ 메시지(440)로 요청을 한다. 서버는 선택적으로 AP-REP 메시지(445)로 응답할 수 있다. AP 교환은 전형적으로 세션-특정의 대칭 키를 협의한다.

[0059] 전형적으로, AS 및 TGS는 KDC로도 알려진 단일 장치로 통합된다.

[0060] AS 교환에서, KDC 응답은 다른 항목들 중에서도, 클라이언트와 KDC 간에 공유되는 키(AS 응답 키)를 이용하여 암호화된 티켓 세션 키를 포함한다. AS 응답 키는 전형적으로 인간 사용자를 위한 클라이언트의 패스워드로부

터 도출된다. 그러므로, 인간 사용자에 대하여, 커베로스 프로토콜의 공격 저항 강도는 인간 사용자의 패스워드의 강도보다 강하지 않다.

[0061] 데이터 원본 인증(data origin authentication) 및 완벽한 비밀(perfect secrecy)를 위하여, X.509 인증서의 형태로 된 비대칭 암호화의 사용("ISOC(the internet society)"에 의해 관리되는 "Request for Comments" 문서 시리즈 하의 RFC 3280 참조)이 인기가 있다. 확립된 공개 키 기반구조(Public Key Infrastructure, PKI)는 인증 및 보안 통신을 확립하는 데에 사용될 수 있는 키 관리 및 키 배포 메커니즘을 제공한다. 커베로스에 공개 키 암호화를 더하는 것은, 공개 키 프로토콜에 양호한 조화를 제공하고, 강력한 패스워드를 관리하기 위한 인간 사용자의 부담을 제거하며, 커베로스화된(Kerbertized) 애플리케이션들이 현존하는 키 서비스 및 신원 관리를 활용할 수 있게 해 준다.

[0062] 커베로스 TGT에 의해 제공되는 이점은 클라이언트가 자신의 장기적인 비밀(long-term secrets)을 단 한번만 노출시킨다는 것이다. TGT 및 그 관련 세션 키는 임의의 후속 서비스 티켓 요청에 대하여 사용될 수 있다. 한가지 결과는 모든 추가의 인증이 초기 인증을 수행한 방법에 독립적이라는 것이다. 결과적으로, 초기 인증은 공개 키 암호화를 커베로스 인증에 통합할 수 있는 편리한 환경을 제공한다. 또한, 초기 교환 후에 대칭 암호화를 사용하는 것이 성능면에서 선호된다.

[0063] RFC 4556은 클라이언트 및 KDC가 공개 및 개인 키 쌍을 이용하여 AS 교환에서 상호 인증하고, KDC에 의해 보내지는 AS-REP를 암호화하기 위하여 클라이언트 및 KDC에게만 알려진 AS 응답 키를 협의할 수 있게 하는 방법 및 데이터 포맷을 기술한다.

[0064] 다시 도 3을 보면, TLS 핸드쉐이크(도 1을 참조하여 논의된 바와 같이 "이중" 핸드쉐이크일 수 있음)의 완료 후에, 게이트웨이(20)는 THR(45)을 제3자 엔터티(40)에 제공한다. 도 1에서와 마찬가지로, 신뢰하는(trusting) 엔터티(이 경우에서는 제3자 엔터티(40))는 클라이언트/사용자(10)와 게이트(20) 간의 인증에 참여하지 않고서 결정을 내린다. 오히려, 신뢰하는 엔터티는 사용자 인증서를 게이트웨이에 제공할 것인지를 결정하기 위하여 THR에 의존한다.

[0065] 더 상세하게는, 유효한 TLS 핸드쉐이크(즉, THR)에 대한 교환에서, 제3자 엔터티(40)는 참조 번호 55에 의해 표시된 바와 같이 소정 형태의 사용자 인증서를 게이트웨이(20)에 반환한다. 그러면, 게이트웨이(20)는 참조 번호 65에 의해 표시된 바와 같이 웹 서버(30)에게 인증하기 위하여 UC를 사용한다. KDC의 경우에, 사용자 자격증명은 사용자(10)의 이름 내의 TGT 또는 커베로스 서비스 티켓일 것이다. CA의 경우, 사용자 자격증명은 사용자의 이름 내의 인증서(통상적으로 짧은 수명을 가짐)일 것이다.

[0066] 도 5는 클라이언트/사용자가 제3자 엔터티를 포함하는 시스템에서 웹 서버에 액세스하려고 할 때에 수행되는 인증 프로세스의 단계들을 도시한 예시적인 흐름도이다. 이러한 시스템에서, 프로세스는 클라이언트/사용자(10)가 게이트웨이(20)에 액세스할 때 시작된다(단계(500)). 그 다음, 클라이언트/사용자(10)는 원하는 웹 서버(30) 자원을 요청한다 (단계(510)). 클라이언트/사용자(10)가 웹 서버에 로그인하지 않은 경우, 웹 서버(30)가 액세스를 허용하기 전에 클라이언트/사용자(10)가 반드시 인증되어야 한다.

[0067] 그 다음, 클라이언트/사용자(10) 및 게이트웨이(20)는 위에서 상세하게 설명된 방식으로 클라이언트 인증을 수반하는 TLS 핸드쉐이크를 수행한다 (단계(520)). TLS 핸드쉐이크의 적어도 일부의 기록(THR)이 생성되어 제3자 엔터티("신뢰하는 엔터티")(40)에 제공된다 (단계(530)).

[0068] THR을 수신하면, 제3자 엔터티(40)는 (THR 내의 인증서 검증 메시지의 유효성을 검증함으로써) 사용자가 게이트웨이에게 인증되었음을 검증한다 (단계(540)). THR이 유효하고 프레쉬한 것으로 검증되는 경우(단계(550)), 사용자 자격증명(예를 들어, 인증 기관의 경우에는 임시 인증서, 또는 KDC의 경우에는 커베로스 서비스 티켓)이 게이트웨이(20)에 제공된다 (단계(560)). 그 다음, 게이트웨이(20)는 실제의 클라이언트/사용자로서 웹 서버(30)에게 인증하기 위하여 사용자 자격증명을 사용한다 (단계(570)). 그러나, (단계(550)에서) THR이 유효하고 프레쉬한 것으로서 검증될 수 없으면, 사용자 자격증명은 게이트웨이(20)에 제공되지 않고, 액세스는 거부된다 (단계(555)).

[0069] 클라이언트/사용자가 웹 서버에 액세스하도록 허가받았다고 가정할 때, 사용자 자격증명(예를 들어, 클라이언트 인증서)이 웹 서버(30)에 의해 인증되면, 요청된 웹 서버로의 액세스가 클라이언트/사용자에게 허여되고 (단계(580)), 클라이언트 인증서가 검증될 수 없으면, 전형적으로 액세스는 거부된다 (단계(590)).

[0070] 게이트웨이(20)에 제공된 사용자 자격증명은 (KDC-기반 배포에서) 서비스 티켓 또는 (CA-기반 배포에서) 임시 인증서로 이루어진다. KDC-기반 배포(deployment)는 위에서 논의된 커베로스 제약된 위임(S4U2Self +

S4U2Proxy)와 매우 유사하므로, 더 논의되지 않을 것이다. 대신에, CA(Certificate Authority)-기반 배포가 논의될 것이다.

[0071] CA-기반 배포에서, 유효하고 "프레쉬"한 THR이 제공될 때 클라이언트 인증서를 발행하기 위하여 하나 이상의 CA가 사용된다. 이러한 시나리오에서, 서비스 제공자(웹 서버(30))는 사용자의 실제 신원을 제공하기 위하여 CA를 신뢰하도록 구성되어야만 한다 (전형적으로, 이것은 CA 자체의 인증서가 서비스 제공자의 운영 체제의 어느 특정 장소에 설치되어 있어야 한다는 것을 의미한다).

[0072] 사용자의 이름에 클라이언트 인증서(및 관련 개인 키)가 제공되고 나면, 게이트웨이(20)는 실제 사용자로서 서비스 제공자(30)에게 인증하기 위해 그러한 자격증명을 이용할 것이다. 서비스 제공자(예를 들어, 웹 서버(30))에게 인증하기 위하여 인증서 및 개인 키를 사용하기 위해, 게이트웨이(20) 및 서비스 제공자는 클라이언트 인증서를 지원하는 인증 프로토콜을 사용해야만 한다. 클라이언트 인증서를 지원하는 2가지의 공지된 인증 프로토콜은 위에서 상세하게 논의된 것과 같은 TLS(또는 SSL) 또는 (PKINIT를 수반하는) 커베로스인데, TLS(또는 SSL) 핸드쉐이크는 (인증서에 기초한) 클라이언트 인증을 포함할 수 있고, PKINIT(Public Key Crytography for Initial Authentication in Kerberos) 메커니즘(RFC 4556)은 커베로스 가능형 클라이언트가 공개 키 암호화를 통해 (즉, 인증서 및 관련 개인 키를 통해) TGT를 얻을 수 있게 하는 커베로스 프로토콜에 대한 프로토콜 확장의 메커니즘이다. 더 상세하게는, 이러한 확장은 사전 인증 데이터 필드에서 비대칭 키 서명 및/또는 암호화 알고리즘을 이용함으로써, 공개 키 암호화를 초기 인증 교환에 통합시키기 위한 방법을 제공한다.

[0073] 클라이언트/사용자가 웹 서버를 액세스하도록 허가받았다고 가정할 때, 자격증명(클라이언트 인증서)가 웹 서버(30)에 의해 인증되고 나면, 요청된 웹 서버로의 액세스가 클라이언트/사용자에게 허여된다. 물론, 클라이언트 인증서가 검증될 수 없으면(또는 웹 서버가 CA를 신뢰하도록 구성되지 않았으면), 사용자에 의한 웹 서버로의 액세스는 거부된다.

[0074] 도 1의 실시예에서와 같이, "이중" TLS 핸드쉐이크는 게이트웨이(20)를 손상시키는 공격자에 대항한 추가의 보호를 위해서 도 3의 실시예에서 구현될 수 있다.

[0075] 본 명세서의 발명의 주제는 구조적 특징 및/또는 방법론적 액트들에 특정한 언어로 설명되었지만, 특허청구범위에 정의된 발명의 주제는 위에서 설명된 특정한 특징 또는 액트에만 한정되는 것은 아님을 알 것이다. 오히려, 위에서 설명된 특정한 특징 또는 액트는 특허청구범위를 구현하는 예시적인 형태로서 개시된 것이다.

[0076] 예를 들어, 본 명세서 전반에서, 클라이언트/사용자로 이루어진 엔터티들의 체인이 게이트웨이를 통해 서비스 제공자에 액세스하는 것이 언급되었다. 그러나, 이것은 단순히 가능한 한가지 시나리오에 지나지 않으며, 본 명세서 전반에서 편의상 사용된 것이다. 다른 가능한 시나리오로는, 백-엔드 애플리케이션 또는 데이터베이스에게 사용자로서 인증할 필요가 있는 웹 서버가 포함된다. 본 명세서의 신규한 양태들은 체인 내의 엔터티들 간에 인증을 필요로 하는 어떠한 엔터티들의 체인에도 적용될 수 있다. 체인 내에는 어떠한 수의 엔터티라도 존재할 수 있으며, 각각의 엔터티는 체인 내의 후속 엔터티에 대하여 원래의 클라이언트로서 인증해야 한다.

[0077] 또한, 하나의 구성요소가 다른 구성요소에 응답하는 것으로서 나타난 때, 구성요소들은 직접 또는 간접적으로 결합될 수 있음을 이해할 것이다. 여기에 나타낸 접속들은 실제에서 구성요소들 간의 결합 또는 통신상의 인터페이스를 달성하기 위해 논리적 또는 물리적일 수 있다. 접속들은 다른 방식들 중에서도 소프트웨어 프로세스들 간의 프로세스간 통신, 또는 네트워크화된 컴퓨터들 간의 머신간 통신으로서 구현될 수 있다.

[0078] "예시적인" 및 "설명적인"이라는 용어는 여기에서 예시, 경우, 또는 설명으로서 기능함을 의미하도록 사용되었다. "예시적인" 또는 "설명적인" 것으로서 여기에 논의된 임의의 구현 또는 양태는 다른 구현 또는 양태들에 비하여 선호되거나 유리한 것으로서 해석되어서는 안 된다.

[0079] 첨부된 특허청구범위의 취지 및 범위를 벗어나지 않고서도, 여기에 설명된 특정한 실시예들 이외의 실시예들이 예상될 수 있을 것으로 이해되므로, 본 발명의 주제의 범위는 이하의 특허청구범위에 의해 결정될 것이다.

도면의 간단한 설명

[0011] 도 1은 TLS 핸드쉐이크 메시지의 재검증에 기초하는 인증 위임의 예시적인 아키텍처의 간략화된 기능 블럭도이다.

[0012] 도 2는 도 1의 예시적인 아키텍처를 이용하는 인증 프로세서의 단계들을 보여주는 예시적인 흐름도이다.

[0013] 도 3은 TLS 핸드쉐이크의 적어도 일부의 기록을 수신하는 제3자 엔터티에 의해 사용자 자격증명이 제공되는 인

증 위임을 위한 예시적인 아키텍처의 간략화된 기능 블럭도이다.

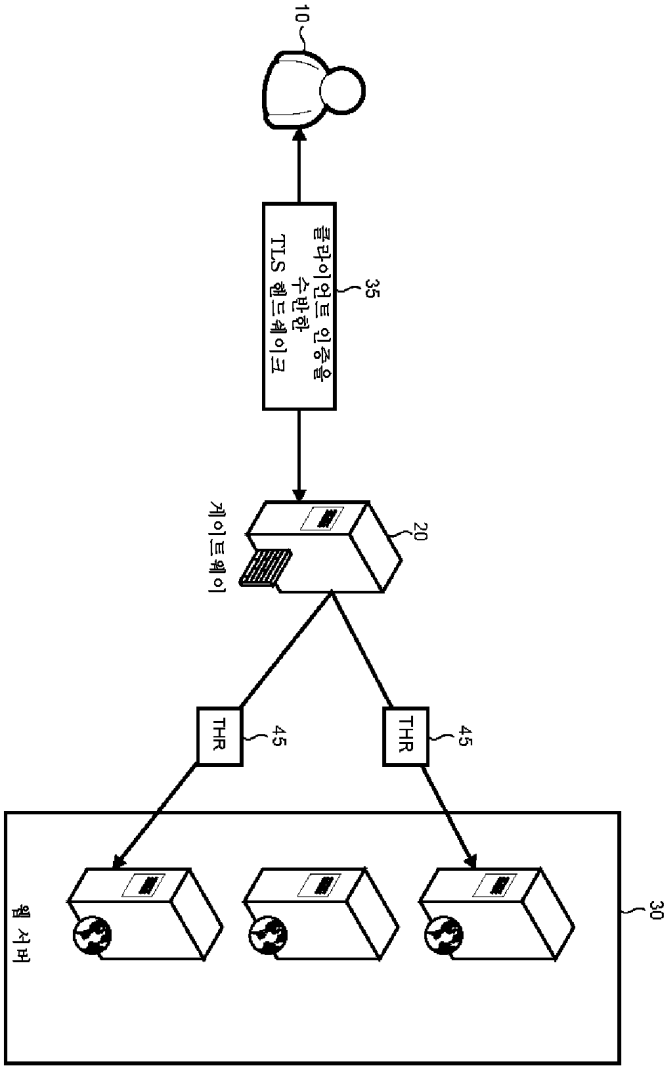
[0014] 도 4는 커베로스(V5) 프로토콜에서 키 분배 센터와 클라이언트 간의 예시적인 메시지 교환을 보여주는 도면이다.

[0015] 도 5는 도 3의 예시적인 아키텍처를 이용하는 인증 프로세스의 단계들을 보여주는 예시적인 흐름도이다.

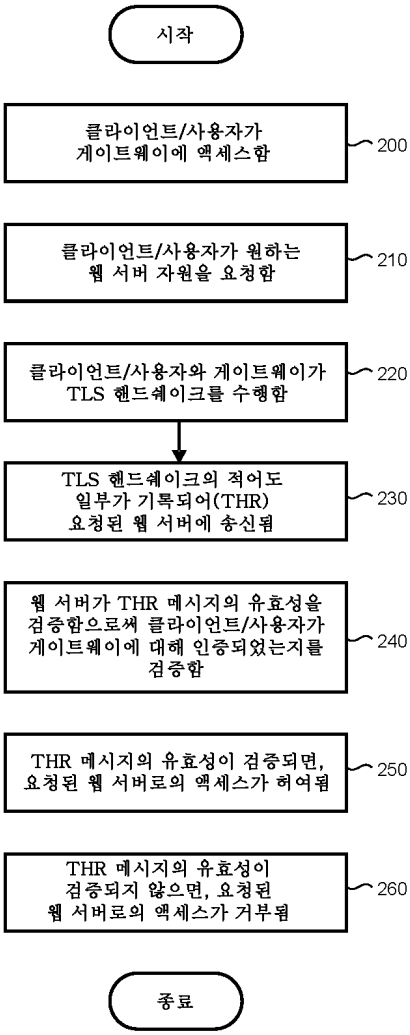
[0016] 도 6은 전형적인 TLS 핸드쉐이크 위상 동안 클라이언트와 서버 간의 메시지 교환을 도시하는 개략적인 메시지 흐름도이다.

도면

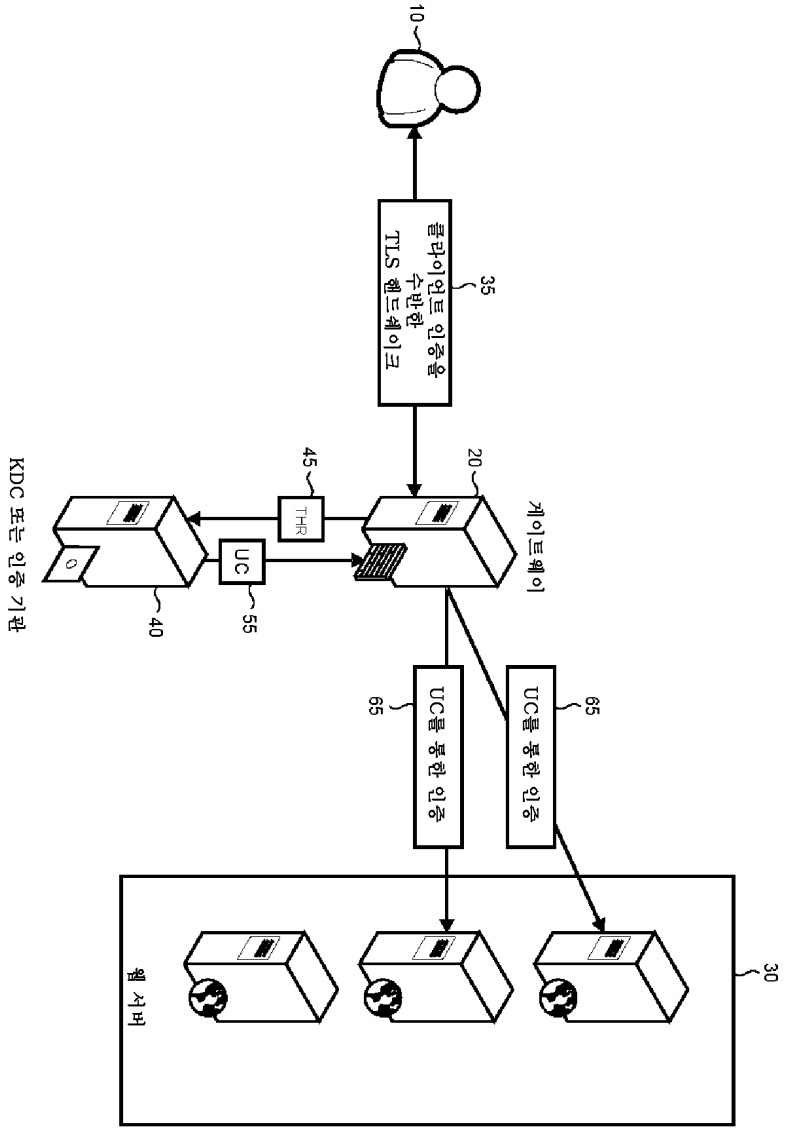
도면1



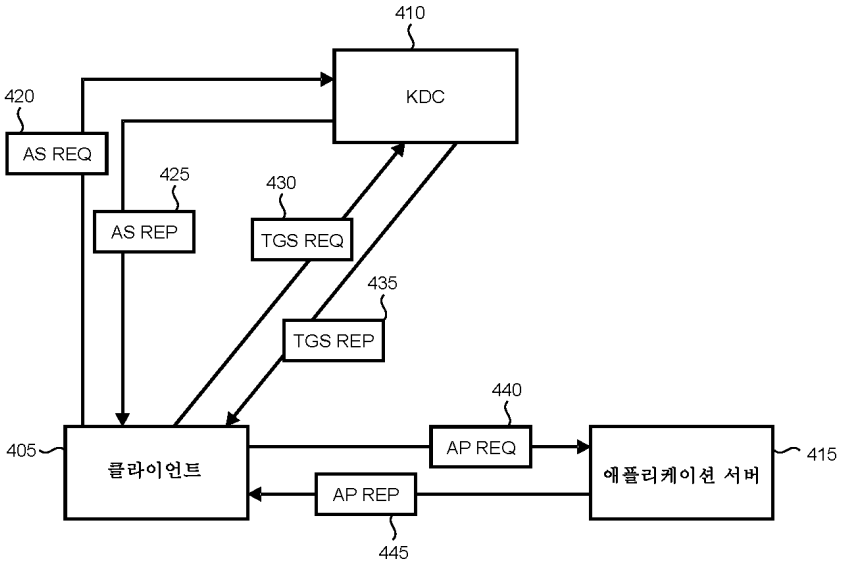
도면2


시작
클라이언트/사용자가 게이트웨이에 액세스함
200
클라이언트/사용자가 원하는 웹 서버 자원을 요청함
210
클라이언트/사용자와 게이트웨이가 TLS 핸드쉐이크를 수행함
220
TLS 핸드쉐이크의 적어도 일부가 기록되어(THR) 요청된 웹 서버에 송신됨
230
웹 서버가 THR 메시지의 유효성을 검증함으로써 클라이언트/사용자가 게이트웨이에 대해 인증되었는지를 검증함
240
THR 메시지의 유효성이 검증되면, 요청된 웹 서버로의 액세스가 허여됨
250
THR 메시지의 유효성이 검증되지 않으면, 요청된 웹 서버로의 액세스가 거부됨
260
종료

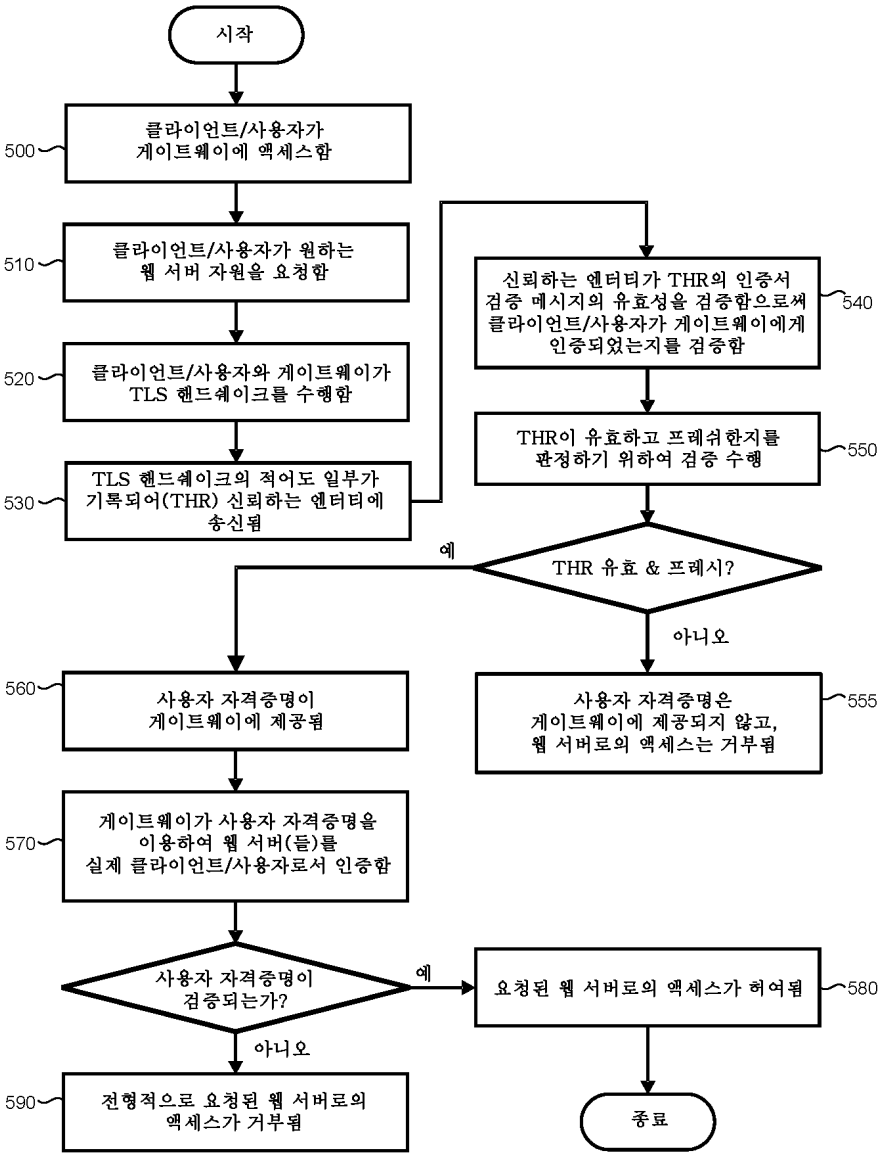
도면3


10
클라이언트 인증을 수반한 TLS 핸드쉐이크
35
45
20
THR
게이트웨이
KDC 또는 인증 기관
UC
40
55
65
UC를 통한 인증
65
UC를 통한 인증
웹 서버
30

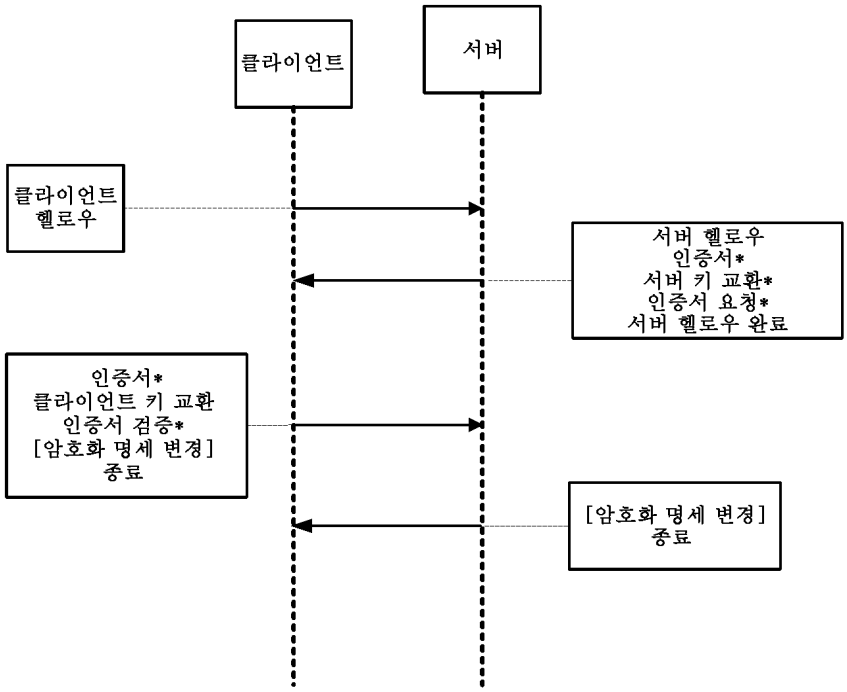
도면4


410
KDC
420
AS REQ
425
AS REP
430
TGS REQ
435
TGS REP
440
AP REQ
405
클라이언트
415
애플리케이션 서버
AP REP
445

도면5


시작
500
클라이언트/사용자가 게이트웨이에 액세스함
510
클라이언트/사용자가 원하는 웹 서버 자원을 요청함
520
클라이언트/사용자와 게이트웨이가 TLS 핸드쉐이크를 수행함
530
TLS 핸드쉐이크의 적어도 일부가 기록되어(THR) 신뢰하는 엔터티에 송신됨
540
신뢰하는 엔터티가 THR의 인증서 검증 메시지의 유효성을 검증함으로써 클라이언트/사용자가 게이트웨이에게 인증되었는지를 검증함
550
THR이 유효하고 프레쉬한지를 판정하기 위하여 검증 수행
THR 유효 & 프레시?
예
아니오
555
사용자 자격증명은 게이트웨이에 제공되지 않고, 웹 서버로의 액세스는 거부됨
560
사용자 자격증명이 게이트웨이에 제공됨
570
게이트웨이가 사용자 자격증명을 이용하여 웹 서버(들)를 실제 클라이언트/사용자로서 인증함
사용자 자격증명이 검증되는가?
예
580
요청된 웹 서버로의 액세스가 허여됨
아니오
590
전형적으로 요청된 웹 서버로의 액세스가 거부됨
종료

도면6


클라이언트
서버
클라이언트
헬로우
서버 헬로우
인증서*
서버 키 교환*
인증서 요청*
서버 헬로우 완료
인증서*
클라이언트 키 교환
인증서 검증*
[암호화 명세 변경]
종료
[암호화 명세 변경]
종료