(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.[7]
G06F 15/00

(11) 공개번호 특2001-0042902
(43) 공개일자 2001년05월25일

(21) 출원번호 10-2000-7011688
(22) 출원일자 2000년10월20일
번역문제출일자 2000년10월20일
(86) 국제출원번호 PCT/US1999/08665 (87) 국제공개번호 WO 1999/54803
(86) 국제출원출원일자 1999년04월20일 (87) 국제공개일자 1999년10월28일
(81) 지정국 AP ARIPO특허 : 케냐 레소토 말라위 수단 스와질랜드 우간다 시에라리온 가나 감비아 짐바브웨

EA 유라시아특허 : 아르메니아 아제르바이잔 벨라루스 키르기즈 카자흐스탄 몰도바 러시아 타지키스탄 투르크메니스탄

EP 유럽특허 : 오스트리아 벨기에 스위스 독일 덴마크 스페인 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투칼 스웨덴 핀랜드 사이프러스

OA OAPI특허 : 부르키나파소 베넹 중앙아프리카 콩고 코트디브와르 카메룬 가봉 기네 말리 모리타니 니제르 세네갈 차드 토고 기네비쏘

국내특허 : 알바니아 아르메니아 오스트리아 오스트레일리아 아제르바이잔 보스니아-헤르체고비나 바베이도스 불가리아 브라질 벨라루스 캐나다 스위스 중국 쿠바 체코 독일 덴마크 에스토니아 스페인 핀랜드 영국 그루지야 헝가리 이스라엘 아이슬란드 일본 케냐 키르기즈 북한 대한민국 카자흐스탄 세인트루시아 스리랑카 라이베리아 레소토 리투아니아 룩셈부르크 라트비아 몰도바 마다가스카르 마케도니아 몽고 말라위 멕시코 노르웨이 뉴질랜드 슬로베니아 슬로바키아 타지키스탄 투르크메니스탄 터어키 트리니다드토바고 우크라이나 우간다 우즈베키스탄 베트남 폴란드 포르투칼 루마니아 러시아 수단 스웨덴 싱가포르 남아프리카 아랍에미리트 모잠비크 그레나다 가나 감비아 크로아티아 인도네시아 인도 시에라리온 유고슬라비아 짐바브웨

(30) 우선권주장 09/063,339 1998년04월20일 미국(US)
(71) 출원인 썬 마이크로시스템즈, 인코포레이티드

미국 94303 캘리포니아주 팔로 알토 산안토니오 로드 901
(72) 발명자 루벅알랜티

미합중국캘리포니아94404포스터시트에머랄드베이레인605

한코제임스지

미합중국캘리포니아94061레드우드시티오하이오애비뉴2746

노스컷저이듀에인

미합중국캘리포니아94025먼로파크세미너리드라이브184

부셔로렌스엘

미합중국캘리포니아94043마운틴뷰콜렌스코트#84315

월게라드에이

미합중국캘리포니아95156산조세크로쿠스드라이브4515
(74) 대리인 윤동열, 이선희

심사청구 : 없음

(54) 세션 관리와 사용자 인증을 위한 장치 및 방법

요약

인간 인터페이스 장치(HID; Human Interface Device)와 전산 서비스 제공자(예컨대, 서버) 사이에 기능을 구분하는 시스템 아키텍처로 인증과 세션 관리가 사용될 수 있다. 서버 상에서 실행되는 인증 관리자는 HID와 대화하여 HID를 통해 사용자가 시스템에 접속했을 때 사용자를 확인한다. 서버 상에서 실행되는 세션 관리자는 사용자를 위해 전산 서비스를 제공하는 컴퓨터 상에서 실행되는 서비스를 관리한다. 세션 관리자는 세션에 있는 서비스 각각에게 사용자가 주어진 HID를 사용하여 시스템에 접속되었음을 통지한다. 서버는 사용자가 시스템에 연결되어 있는 동안 디스플레이 출력을 HID에게 보낼 수 있다. 사용자가 시스템에서 분리되면, 사용자를 위해 실행되고 있는 각각의 서비스는 인증 관리자와 세션 관리자를 통해 이를

통지받는다. 사용자가 시스템에서 분리되었음을 통지받으면, 서비스는 HID에 대한 디스플레이를 중지하지만 실행을 계속할 수는 있다.

명세서

기술분야

본 발명은 컴퓨터 시스템에 관한 것으로, 좀 더 구체적으로는 사용자 인증 및 사용자 세션의 위치 관리에 관한 것이다.

배경기술

컴퓨터 시스템을 구성하는 패러다임은 시대에 따라 변해왔다. 초기 컴퓨터는 여러 대의 더미 터미널(dumb terminal)이 접속된 소위, 메인 프레임(main frame)으로 구성되었다. 메인 프레임은 컴퓨팅 능력과 데이터 저장을 가능하게 하는 중앙 단말이었다. 더미 터미널은 메인프레임에서 제공하는 데이터에 대한 디스플레이 장치임과 동시에 데이터를 메인 프레임과 통신하는 기능을 제공하였다. 그 뒤에 나타난 시스템 패러다임으로는 데스크탑(desktop) 컴퓨터, 클라이언트/서버 아키텍처 (client/server architecture)가 있으며 최근에는 소위, 네트워크 컴퓨터(network computer)가 등장하였다.

데스크탑 컴퓨터는 컴퓨팅 시스템을 내장하고 있으며 모든 애플리케이션과 데이터가 데스크탑 컴퓨터 시스템 자체에 상주한다. 데스크탑 컴퓨터는 개인용 컴퓨터로 구현되어 가정이나 사무실에서 컴퓨터가 사용되는 데에 자극이 되었다. 데스크탑 컴퓨터의 단점은 이 시스템에 사용되는 하드웨어 수명이 짧다는 것이다. 데스크탑 컴퓨터는 마이크로프로세서에 의해 구동되는데, 속도가 더 빠르고 강력한 마이크로프로세서가 등장함에 따라 현재 사용하고 있는 데스크탑 시스템을 업그레이드하거나 새로운 데스크탑 시스템을 구입해야 할 필요가 있다. 많은 사무실에는 개인용 데스크탑 컴퓨터가 보급되어 있는데, 그 수는 수천에서 수십만대에 이른다. 이러한 대규모 시스템에서는 개인 시스템에 대한 애플리케이션이나 데이터의 호환성이 부족하다는 것이 단점이다. 어떤 사용자는 좀 더 최신 버전의 소프트웨어 애플리케이션을 가지고 있는데, 이것은 이전 버전의 소프트웨어와 호환되지 않는다. 이러한 문제를 해결하기 위해 모든 시스템에 대해 소프트웨어를 일관되게 유지해야 한다. 그러나, 모든 시스템을 업그레이드하고 정품 소프트웨어 및 이 소프트웨어의 업그레이드판을 공급하는 데에는 많은 비용이 든다.

클라이언트 서버 시스템은 중앙 데이터 저장부 및/또는 애플리케이션을 개인용 컴퓨터 클라이언트가 네트워크를 통해 접속하는 시스템을 말한다. 이 시스템은 공유 데이터를 유지하는 관리 효율을 어느 정도 제공할 수 있다. 그러나, 클라이언트는 여전히 로컬 애플리케이션과 데이터를 가지고 있기 때문에, 데스크탑 시스템과 관련하여 설명했던 문제점과 같은 종류의 문제를 나타낼 수 있다.

최근에, 인터넷의 성장은 소위, 네트워크 컴퓨터의 사용을 자극하는 결과를 낳았다. 네트워크 컴퓨터는 저장 공간의 축소, 메모리 축소, 더 낮은 계산 능력을 가지므로, 개인용 컴퓨터의 축소 버전(stripped down version)이라고 할 수 있다. 기본적인 사상은 인터넷을 통해 네트워크 컴퓨터가 데이터에 접근하고 특정 작업에 필요한 애플리케이션만 네트워크 컴퓨터에 제공되도록 하는 것이다. 제공된 애플리케이션이 더 이상 사용되지 않으면, 이것은 네트워크 컴퓨터에도 저장되지 않을 것이다. 이러한 시스템은 데스크탑 시스템의 모든 기능을 제대로 갖지 못하면서도 이것을 대체할만큼 비용도 싸지 않다는 점에서 비판을 받아 왔다. 또한, 네트워크 컴퓨터는 데스크탑 컴퓨터의 서브세트임에도 불구하고 적당한 수준의 성능을 유지하기 위해서는 하드웨어와 스프트웨어를 업그레이드해야 할 필요성이 여전히 존재한다.

동적 호스트 구성 프로토콜(dynamic host configuration protocol)의 예가 RFC 2131에 제공되어 있다. RFC 1321 및 2104는 MD5 (message digesting)의 예를 포함한다. 2 지점간 챌린지 호스트 인증 프로토콜(point to point challenge host authentication protocol)이 RFC 1994에 포함되어 있다.

발명의 상세한 설명

발명의 요약

인증과 세션 관리는 인간 인터페이스 장치(HID; Human Interface Device)와 컴퓨팅 서비스 제공자(예컨대, 서버) 사이에 기능을 분할하는 시스템 아키텍처로 사용될 수 있다. 서버 상에서 실행되는 인증 관리자는 HID와 대화하여 사용자가 HID를 통해 시스템에 접속했을 때 사용자를 확인(validate)한다. 서버 상에서 실행되는 세션 관리자는 사용자에게 전산 서비스(예컨대, 프로그램)를 제공하는 컴퓨터 상에서 구동되는 서비스를 관리한다. 세션 관리자는 어느 세션 내의 각각의 서비스에게 사용자가 주어진 데스크탑 기계를 사용하여 시스템에 연결되어 있음을 통지한다. 사용자가 시스템에 연결되어 있는 동안에는, 서비스가 디스플레이 출력을 HID에게 직접 보낼 수 있다. 사용자가 시스템에서 연결 해제되면, 사용자를 위해 실행되고 있던 각각의 서비스는 인증 관리자와 세션 관리자로부터 그 사실을 통지받는다. 사용자가 시스템에서 연결 해제되었음을 통지받은 서비스는 데스크탑 기계에 디스플레이하는 것을 중지하면서 실행을 계속한다.

도면의 간단한 설명

도 1은 본 발명의 여러 실시예에 사용되는 시스템 아키텍처의 예이다.

도 2는 본 발명의 일실시예에 따른 인증 관리 부분과 세션 관리 부분 및 그들 사이의 상호 작용을 설명하는 도면.

도 3은 본 발명의 일실시예에 따른, 전원 공급 동작에 응답하여 네트워크 단말을 개시하는 프로세스 흐름을 설명하는 도면.

도 4A-4C는 기상 프로세스(awaken process)에 응답하여 네트워크 단말(202)을 개시하는 본 발명의 일실시

예에 따른 프로세스 흐름도.

도 5A-5B는 본 발명의 일실시예에 따른 인증 프로세스 흐름도.

도 6은 본 발명의 일실시예에 따른 챌린지 프로세스(challenge process) 흐름도.

도 7 및 도 8은 본 발명의 여러 실시예에 사용되는 시스템 아키텍처의 예를 보여준다.

실시예

세션 관리와 사용자 인증을 위한 장치 및 방법에 대해 설명한다. 이하의 설명에서, 여러 가지 구체적인 사항들은 본 발명을 좀 더 자세하고 완전하게 설명하기 위한 것이다. 그러나, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 이러한 구체적인 사항들이 없더라도 본 발명을 구현할 수 있다는 것을 쉽게 알 수 있을 것이다. 다른 예에서는, 이미 알려진 특징들에 대해서는 본 발명을 불명확하게 하지 않는한 설명을 생략한다.

개요

시스템 사용자를 인증하고 사용자를 위해 이 시스템에서 사용되는 서비스를 관리하는 본 발명의 여러 실시예에 따른 방법과 장치에 대해 설명한다. 본 발명의 일실시예에서, 인증과 세션 관리는 컴퓨팅 기능을 사용자의 HID와 전산 서비스 제공자(예컨대, 서버) 사이에서 분할하는 시스템 아키텍처 내에서 수행된다.

도 1, 7, 8은 본 발명의 여러 실시예에서 사용되는 시스템 아키텍처의 예를 보여준다. 본 발명은 도 1에 나타낸 표준 데스크탑 컴퓨터 시스템에서 구현될 수도 있고, 도 7과 도 8에 설명되어 있는 클라언트-서버 시스템, 네트워크 컴퓨터, HID 시스템과 같은 다른 컴퓨터 시스템에서 구현될 수도 있다.

컴퓨터 실행 환경(하드웨어)에 대한 실시예

본 발명의 실시예는, 도 1에 설명된 컴퓨터(100)와 같은 범용 컴퓨터 상에서 실행되는 컴퓨터 판독 가능 코드 형태로 된 컴퓨터 소프트웨어, 또는 범용 컴퓨터 상에서 실행되는 자바 실행시간(Java runtime) 환경에서 실행될 수 있는 바이트코드 클래스 파일(bytecode class file) 형태로 된 컴퓨터 소프트웨어로 구현될 수 있다. 키보드(110)와 마우스(111)은 쌍방향 시스템 버스(118)에 연결되어 있다. 키보드와 마우스는 컴퓨터 시스템에 사용자 입력을 넣고 사용자 입력을 프로세서(113)와 통신시키기 위한 것이다. 마우스(111)와 키보드(110)에 추가하여 또는 그 대신에 다른 입력 장치를 사용할 수도 있다. 쌍방향 시스템 버스(118)에 연결되어 있는 I/O (입출력) 유니트(119)는 프린터, 오디오/비디오 (A/V) I/O 등과 같은 I/O 장치들을 나타낸다.

컴퓨터(100)는 비디오 메모리(114), 메인 메모리(115) 및 대용량 기억장치(112)를 포함하는데, 이들은 모두 키보드(110), 마우스(111) 및 프로세스(113)과 함께 쌍방향 시스템 버스(118)에 연결되어 있다. 대용량 기억장치(112)는 자기 기억 시스템, 광 기억 시스템 또는 광자기 기억 시스템이나 기타 사용가능한 대용량 기억 장치와 같은 매체로서 고정식이나 분리식 매체를 포함한다. 시스템 버스(118)는 예컨대, 비디오 메모리(114), 메인 메모리(115)를 어드레싱하는 예컨대, 32개의 어드레스 라인을 포함한다. 시스템 버스(118)는 또한 프로세서(113), 메인 메모리(115), 비디오 메모리(114) 및 대용량 기억장치(112)와 같은 부품들 사이에 데이터를 전달하기 위한 예컨대 32-비트 데이터 버스를 포함한다. 또는, 별도의 데이터 라인과 어드레스 라인을 사용하는 대신, 다중 데이터/어드레스 라인을 사용할 수도 있다.

본 발명의 일실시예에 따르면, 프로세서(13)는 모토롤라(Motorola)에서 제조되는 680X0 프로세서와 같은 마이크로프로세서, 또는 인텔(Intel)에서 제조되는 80X86이나 펜티엄(Pentium) 프로세서와 같은 마이크로프로세서이거나, 선 마이크로시스템즈(Sun Microsystems)에서 제공되는 스팍(SPARC) 마이크로프로세서를 사용할 수 있다. 하지만, 적당한 다른 마이크로프로세서나 마이크로 컴퓨터를 사용하는 것도 가능하다. 메인 메모리(115)는 동적 임의 접근 메모리(DRAM; Dynamic Random Access Memory)로 구성된다. 비디오 메모리(114)는 이중 포트 비디오 임의 접근 메모리(RAM)이다. 비디오 메모리(114)의 하나의 포트는 비디오 증폭기(116)에 연결된다. 비디오 증폭기(116)는 음극선관(CRT) 래스터 모니터(raster monitor)를 구동하는 데에 사용된다. 이와 달리, 평판 디스플레이나 액정 디스플레이(LCD), 또는 기타 적당한 데이터 표시 장치를 구동하는 데에 비디오 메모리(114)를 사용할 수도 있다. 비디오 메모리는 당해 기술 분야에서 널리 알려진 것이고 절절한 장치로 구현될 수 있다. 이 회로는 비디오 메모리(114)에 저장되어 있는 화소 데이터를 모니터(117)에서 사용되는 적절한 래스터 신호로 변환한다. 모니터(117)는 그래픽 이미지를 디스플레이하는 데에 적당한 형태의 모니터이다.

컴퓨터(100)는 버스(118)에 연결된 통신 인터페이스(120)를 포함한다. 통신 인터페이스(120)는 네트워크 링크(122)를 통하여 로컬 네트워크(122)에 대한 쌍방향 데이터 통신 커플링(coupling)을 제공한다. 예를 들어서, 통신 인터페이스(120)가 통합 서비스 디지털 통신망(ISDN; Integrated Servcie Digital Network) 카드 또는 모뎀이나 케이블 모뎀인 경우, 통신 인터페이스(120)는 해당 유형의 전화선(이것은 네트워크 링크(121)의 일부를 구성함)에 대한 데이터 통신 접속을 제공한다. 만약, 통신 인터페이스(120)가 근거리 통신망(LAN; Local Area Network) 카드인 경우에는, 통신 인터페이스(120)는 네트워크 링크(121)를 통해 호환성 LAN에 대한 데이터 통신 접속을 제공한다. 무선 링크도 가능하다. 이러한 구현예 중 어느 경우에도, 통신 인터페이스(120)는 여러 유형의 정보를 나타내는 디지털 데이터 스트림을 운반하는 전기적, 전자기적 또는 광적 신호를 주고 받는다.

네트워크 링크(121)는 하나 이상의 네트워크를 통해 다른 데이터 장치에 대한 데이터 통신을 제공하는 것이 보통이다. 예를 들어서, 네트워크 링크(121)는 근거리 통신망(122)을 통해, 인터넷 서비스 제공자(123; ISP)에 의해 운영되는 데이터 장비 또는 지역 서버 컴퓨터(123)에 대한 접속을 제공한다. ISP(124)는 그 다음에 월드 와이드 패킷 데이터 통신 네트워크[지금은 보통 인터넷(125)이라 함]를 통해 데이터 통신을 제공한다. 근거리 통신망(122)과 인터넷(125)은 모두 디지털 데이터 스트림을 운반하는 전기적, 전자기적, 또는 광적 신호를 사용한다. 여러 네트워크를 통과하는 이러한 신호들과 네트워크 링크(121) 상에 있는 신호들 및 컴퓨터(100)에 대해 디지털 데이터를 주고 받는 통신 인터페이스(120)를 통과하는

신호들은 정보를 전달하는 반송파의 예시적 형태이다.

컴퓨터(10)는 프로그램 코드를 포함하여, 네트워크(들), 네트워크 링크(121), 통신 인터페이스(120)를 통해 메시지를 보내고 데이터를 주고 받을 수 있다. 인터넷을 예로 들면, 원격 서버 컴퓨터(126)는 애플리케이션 프로그램에 대한 요청 코드를 인터넷(125), ISP(124), 근거리 통신망(122) 및 통신 인터페이스(120)를 통해 전송할 것이다.

수신된 코드는 그것이 수신된 대로 프로세서(113)에 의해 실행되고/되거나 대용량 저장 장치(112) 또는 기타 비휘발성 저장 매체에 저장되어 나중에 실행된다. 이렇게 함으로써, 컴퓨터(100)는 애플리케이션 코드를 반송파의 형태로 얻을 수 있다.

애플리케이션 코드는 어떤 형태의 컴퓨터 프로그램 제품으로도 구현될 수 있다. 컴퓨터 프로그램 제품은 컴퓨터로 읽을 수 있는 코드를 저장하거나 전송하도록 구성된 매체 또는 컴퓨터로 읽을 수 있는 코드가 내장된(embedded) 매체를 포함한다. 컴퓨터 프로그램 제품의 예로는 CD-ROM 디스크, ROM 카드, 플로피 디스크, 자기 테이프, 컴퓨터 하드 드라이브, 네트워크상의 서버 및 반송파가 있다.

인간 인터페이스 장치 컴퓨터 시스템

본 발명은 디스플레이될 데이터가 네트워크를 통해 제공되는 컴퓨터 시스템에도 적용될 수 있다. 이 네트워크는 근거리 통신망일 수도 있고, 광역 네트워크, 인터넷, 월드 와이드 웹 또는 다른 적당한 네트워크 구조일 수 있다. 본 발명의 일실시예는 이하 인간 인터페이스 장치 컴퓨터 시스템이라고 하는 컴퓨터 시스템 구조로 사용된다.

이 시스템에서, 시스템의 기능은 디스플레이 및 입력 장치와 데이터 소스 또는 서비스 사이에서 분할된다. 디스플레이와 입력 장치는 인간 인터페이스 장치(HID)이다. 이 시스템의 분할(partitioning)은 다음과 같다. 상태 및 계산 기능은 HID로부터 제거되고, 데이터 소스 또는 서비스에 상주한다. 본 발명의 일실시예에서, 하나 이상의 서비스가 네트워크와 같은 어떤 연결 조직(interconnect fabric)을 통해 하나 이상의 HID와 통신한다. 이러한 시스템의 예는 도 7에 나타나 있다. 도 7을 참조하면, 시스템은 HID(702)와 연결 조직(701)을 통해 데이터를 통신하는 전산 서비스 제공자(700)로 구성된다.

전산 서비스 제공자

HID 시스템에서, 계산 능력과 상태 유지는 서비스 제공자 또는 서비스에서 발견된다. 이 서비스는 특정 컴퓨터에 묶여 있지 않고, 도 1을 참조로 설명한 바와 같은 하나 이상의 통상적인 데스크탑 시스템 또는 통상적인 서버를 포함하는 시스템에 대해 분산되어 있다. 하나의 컴퓨터는 하나 이상의 서비스를 가질 수 있고, 하나의 서비스는 하나 이상의 컴퓨터에 의해 구현될 수 있다. 이러한 서비스는 계산, 상태 및 데이터를 HID에게 제공하고, 공통 관리자의 통제 아래에 있다. 도 7에서 컴퓨터(710, 711, 712, 713, 714) 상에서 서비스를 볼 수 있다.

서비스의 예로는 X11/유닉스 서비스, 아카이브(archived) 비디오 서비스, 윈도우 NT 서비스, 자바 프로그램 실행 서비스 등이 있다. 여기서 서비스란 사용자 요청과 입력에 대해 데이터를 출력하고 응답하는 프로세스를 말한다.

연결 조직

본 발명에서, 연결 조직은 서비스와 HID 사이에서 데이터를 전달하는 적절한 다중 통신 경로이다. 본 발명의 일실시예에서, 연결 조직은 이서네트 네트워크(Ethernet network)로 구현되는 근거리 통신망이다. 다른 형태의 근거리 통신망을 사용하는 것도 가능하다. 본 발명은 광역 네트워크, 인터넷, 월드 와이드 웹 등을 사용하는 데에도 적용된다. 연결 조직은 와이어나 광섬유와 같은 물리적인 매체로 구현될 수도 있고, 무선 환경에서 실현될 수도 있다.

HID

HID는 서비스가 제공하는 전산 서비스에 사용자가 접근할 수 있도록 하는 수단이다. 도 7에는 HID(721, 722, 723)가 도시되어 있다. HID는 디스플레이(726), 키보드(724), 마우스(725) 및 오디오 스피커(727)로 구성된다. HID는 이 장치를 연결 조직과 인터페이스하고, 서비스로부터 데이터를 주고 받는 데에 필요한 전자 회로들을 포함한다.

HID의 블록도가 도 8에 나타나 있다. HID의 구성 부품들은 내부적으로 PCI 버스(812)로 연결되어 있다. 네트워크 제어 블록(802)은 라인(814)을 통해 이서네트와 같은 연결 조직과 통신한다. 오디오 코덱(803; audio codec)은 인터페이스(816) 상에서 오디오 데이터를 수신하며, 블록(802)에 연결되어 있다. USB 데이터 통신은 라인(813)을 통해 USB 제어기(801)에 제공된다.

내장형 프로세서(804; embedded processor)는 예를 들어서, Sparc2ep로서 플래시 메모리(805)와 DRAM(806)과 연결되어 있다. USB 제어기(801), 네트워크 제어기(802) 및 내장형 프로세서(804)는 모두 PCI 버스(812)에 연결되어 있다. PCI 버스(812)에는 비디오 제어기(809)도 연결되어 있다. 비디오 제어기(809)는 예를 들어서 ATI RagePro+ 프레임 버퍼 제어기인데, 이것은 라인(815) 상에 SVGA 출력을 제공한다. NTSC 데이터(817)는 비디오 디코더(810)를 통해 비디오 제어기에 입력되고 비디오 인코더(811)를 통해 비디오 제어기로 출력된다. 스마트카드 인터페이스(808) 역시 비디오 제어기(809)에 연결되어 있다.

앞에서 설명한 컴퓨터 시스템은 예시적인 목적으로 설명된 것이다. 본 발명의 실시예는 어떠한 유형의 컴퓨터 시스템이나 프로그래밍 또는 프로세싱 환경에서도 구현될 수 있다.

본 발명의 하나 이상의 실시예에서, 인증 관리 부품과 세션 관리 부품은 사용자를 인증하고 세션을 위치 찾기(locate) 및 관리한다. 세션은 사용자를 위해 실행되는 하나 이상의 서비스로 이루어진 관련 세트의 지속적인 표현(persistent representation)이다. 본 발명의 실시예들은 어떤 세션 내의 서비스가 사용자 확인과 재배치(relocation)를 수행하도록 구성되어야 할 필요없이 사용자의 현재 위치에 기반하여 사용자 세션을 재배치하고 사용자를 인증한다. 본 발명의 실시예들은 모든 사용자 서비스에 대해 사용자를 한번만 인증한다. 본 발명의 실시예를 사용하면, 사용자가 현재 사용하고 있는 HID (또는 다른 단말 장치)에

서비스가 바로 보내진다. 사용자는 각각의 서비스에 대해 일일이 로그인(login)할 필요가 없고 특정 HID에 대해 새로운 연결을 구성할 필요도 없다.

본 발명의 실시예들에 따르면, 인증은 단방향 인증이기 때문에, 인증의 관리성과 기준성(scalability)이 개선된다. 키(key)를 교환할 필요가 없으므로, 중앙 데이터베이스에서 키 조사(key lookup)를 할 필요가 없다.

도 2는 본 발명의 일실시예에 따른 인증 부품과 세션 관리 부품 및 이들의 상호 작용을 보여준다. 네트워크 단말(202)은 인간 인터페이스 장치(HID) [예컨대, HID (821, 822, 823)]이다. HID는 예를 들어서, 서비스의 출력을 사용자에게 디스플레이하고 사용자로부터 서비스에게 제공되는 입력을 구하는 작업을 그 기능으로 가진다. 네트워크 단말(202)은 전산 서비스 제공자[예를 들어서, 컴퓨터(710, 711, 712, 713, 714)]에서 실행되는 소프트웨어 프로그램[예컨대, 서비스(230-238), 인증 관리자(204), 세션 관리자(206)]으로부터 받은 명령(command)(예컨대, 디스플레이 명령)에 응답하는 능력을 가지고 있다. 사용자로부터 받은 입력은 예를 들어서, 사용자 요청을 수행하는 서비스로 전달된다.

세션을 포함하는 서비스는 하나 이상의 서버에 의해 실행된다. 예를 들어서, 세션(208)에서, 서비스(230)는 서버(210)에서 실행되고, 서비스(232, 234)는 서버(212)에서 실행되며, 서비스(236, 238)는 서버(214)에서 실행된다.

사용자는 로그인을 개시함으로써 시스템(예컨대, 서버, 세션, 서비스, 네트워크 단말)에 접근한다. 로그인 동안에, 인증 관리자(204)가 사용자를 확인한다. 사용자가 로그인을 개시하는 데에는 여러가지 다양한 기술들이 사용될 수 있다. 예를 들어서, 사용자는 네트워크 단말(202) 상의 키를 누름으로써 로그인을 시작할 수 있다.

본 발명의 일실시예에서, 사용자는 네트워크 단말(202)에 부착된 카드 판독기[예컨대, 카드 판독기(216)]에 스마트 카드를 삽입함으로써 시스템에 접근한다. 스마트 카드는 스마트 카드의 메모리나 자기 스트립과 같은 곳에 정보를 저장할 수 있다. 스마트 카드는 사용자 고유번호(즉, 64-비트 숫자의 사용자 ID)와 같은 사용자 정보와 네트워크 단말(202)에 전송되는 비밀 코드(예컨대, 128-비트의 난수)를 저장할 수 있다. 비밀 코드는 인증 과정에서 사용된다.

네트워크 단말(202)은 그것의 접속 네트워크 어드레스 및 인증 관리자(204)의 어드레스를 알고 있다 (또는 얻을 수 있다). 사용자가 로그인을 시작하면, 네트워크 단말(202)은 인증 관리자(204)와 통신을 개시하여 인증을 시작한다. 인증 관리자(204)는 예컨대 근거리 통신망(LAN)과 같은 통신 네트워크를 통해 네트워크 단말(202)과 연결된 전산 서비스 제공자 상에서 활성 상태인(예컨대, 실행되는) 프로그램이다. 그러나, 네트워크 단말(202)이 패브릭 채널 루프(fabric channel loop) 또는 2 지점간(point-to-point) 케이블과 같은 다른 접속 네트워크 기술을 사용하여 인증 관리자(204)에 접속될 수도 있다는 것은 명백하다. 네트워크 단말(202)은 사용자 고유 번호(사용자ID)를 포함하는 인증 관리자(204)에게 시작 요청(startup request)을 보낸다.

본 발명의 일실시예에서, 인증 관리자(204)는 시작 요청에 응답하여 사용자를 확인하는 인증을 개시한다. 인증은 시스템에 대한 사용자의 신분(identify)을 확인하는 어떠한 메카니즘도 포함할 수 있다. 사용자만 알고 있는 키 또는 암호 또는 생체 정보(biometrics information)를 사용하여 사용자를 인증하는 것도 가능하다.

본 발명의 일실시예에서, 인증은 네트워크 단말(202)에서 사용자가 입력한 개인 식별 번호(PIN; Personal Identification Number)를 확인함으로써 수행된다. 인증 관리자(214)는 사용자의 PIN 입력을 시작할 것을 지시하는 명령(즉, 챌린지 명령)을 네트워크 단말(202)에게 보낸다. 사용자의 입력은 네트워크 단말(202)에 의해 패키지되어 인증 관리자(204)에게 전송된다(즉, 챌린지 응답).

인증 관리자(204)는 인증 데이터베이스(218)에 저장되어 있는 사용자 정보와 사용자에게 제공한 정보 및 인증 과정에서 생성된 정보를 가지고 챌린지 응답을 검증한다. 사용자가 인증되면, 사용자에게 세션[예컨대, 세션(208)]에 접근한 권한을 준다.

사용자로부터 기대했던 결과를 받으면, 인증 관리자(204)는 (접속 메시지를 통해) 세션 관리자(206)에게 사용자가 네트워크 단말(202) 상에서 시스템에 로그인하였음을 통지한다. 인증 데이터베이스(218)에 들어있는 세션 정보는 세션 관리자(206)에 대한 세션 고유번호(ID), 포트, 서버를 식별하는 데에 사용된다. 세션 관리자(206)는 전산 서비스 제공자 상에서 실행되며 예컨대 연결 네트워크를 통해 인증 관리자(204) 및 네트워크 단말(202)에 연결된 프로그램이다. 인증 관리자(204)는 세션 관리자(206)의 서버와 인증 데이터베이스(218)에 들어 있는 포트 정보를 사용하여 세션 관리자(206)에게 메시지를 보낸다.

인증 관리자(204)가 보낸 연결 메시지에 응답하여, 세션 관리자(206)는 사용자의 현재 세션에 있는 서비스[즉, 세션(208)에 있는 서비스]에게 사용자가 네트워크 단말(202)에 접속되었음을 통지한다. 다시 말하면, 세션 관리자(206)는 연결 메시지를 서비스(230-238)에게 보내 출력을 네트워크 단말(202)로 보낸다. 세션 관리자(206)는 세션의 필요한 서비스라고 판단되는 서비스들이 실행되도록 보장한다. 만약 그렇지 않으면, 세션 관리자(206)는 서비스가 개시되도록 한다. 사용자는 세션[예컨대, 세션(208)] 내의 서비스들(230-238)과 대화할 수 있다. 네트워크 단말(202)은 근거리 통신망와 같은 연결 네트워크 또는 다른 연결 기술을 사용하여 서버(210, 212, 214) [및 서버(230-238)]에 연결된다. 사용자는 새로운 서비스를 시작하거나 현재 서비스를 종료할 수 있다.

사용자는 카드 판독기(216)에서 카드를 제거함으로써 시스템에서 분리될 수 있다. 분리를 표현하는 다른 메카니즘들[예를 들어서, 네트워크 단말(202) 상의 sign-off 버튼]을 본 발명에 사용하는 것도 가능하다. 서비스(230-238)는 사용자가 카드 판독기(216)에서 카드를 제거한 후에도 실행을 계속할 수 있다. 즉, 사용자의 관련 세션(들)과 세션을 포함하는 서비스는 사용자가 시스템으로부터 분리(예를 들면, 로그 오프)된 기간 동안에도 계속 존재할 수 있다. 사용자가 카드 판독기(216)에서 카드를 제거하면, 네트워크 단말(202)은 인증 관리자(204)에게 (예컨대, 분리 메시지를 통해) 통지하고, 인증 관리자(204)는 세션 관리자(206)에게 (예컨대, 분리 메시지를 통해) 통지한다. 세션 관리자(206)는 네트워크 단말(202)에 대한

디스플레이 명령 전송을 종료한 서비스(230-238)에게 (예컨대, 분리 메시지를 통해) 통지한다. 그러나, 서비스(230-238)는 사용자가 네트워크 단말에 로그온되어 있지 않는 시간 동안에도 실행을 계속한다. 사용자는 네트워크 단말(202)과 같은 네트워크 단말을 사용하여 다시 로그인할 수 있고, 세션(208)에 접속하여 서비스(230-238)과 대화할 수 있다.

도 2는 각각 하나의 인스턴스(instance)를 나타내고 있지만, 네트워크 단말(202), 인증 관리자(204), 세션(208)의 다중 인스턴스가 가능하다는 것은 명백하다. 예를 들어서, 네트워크 단말(202)에 서비스하는 인증 관리자(204)의 하나 이상의 인스턴스가 가능하고 네트워크 단말(202)의 다중 인스턴스도 가능하다. 인증 관리자(204) 인스턴스는 예컨대, 네트워크의 토폴로지(topology)에 따라 계층적으로 조직화될 수도 있고, 어디서든 활용가능하게 할 수도 있다.

인증 관리자의 인스턴스를 하나 이상으로 하면, 현재의 부하(예컨대, 사용자의 수)에 기초하여 인증 관리자(204)의 인스턴스의 추가 (또는 삭제)가 가능하기 때문에 시스템의 기준성이 향상된다. 또한, 인증 관리자(204)의 여분의 인스턴스(redundant instace)를 배치(deploy)할 수 있기 때문에 신뢰성이 개선된다.

마찬가지로, 세션 관리자(206) 인스턴스의 다중화도 가능하다. 인증 관리자(204)의 경우와 같이, 세션 관리자(206)의 다중 인스턴스는 시스템의 기준성과 신뢰도를 높인다.

세션 관리자

세션 관리자(206)는 사용자, 세션, 서비스 사이의 매핑(mapping)을 보유하는 세션 데이터베이스(220)를 유지한다. 세션 관리자(206)는 세션 관리자(206)에 의해 관리되는 각각의 세션을 포함하는 서비스를 관리한다. 예를 들어서, 세션 관리자(206)는 세션(208)을 유지하고 세션(208) 내의 서비스(230-238)를 유지한다.

전산 서비스 제공자에게 접근하기 위하여, 어카운트(account)를 먼저 셋업하거나 사용자에게 인에이블된다. 예를 들어서, 본 발명의 일실시예에 따라 사용자를 인에이블하기 위해서는, 사용자에게 사용자 ID와 비밀 코드를 저장하고 있는 스마트 카드, 사용자 ID 및 PIN을 부여한다. 또한, 사용자에 대해 세션을 생성한다. 이하에 설명되는 바와 같이, 세션은 요청된 서비스를 가지지 않을 수도 있고 더 많은 서비스를 가질 수도 있다. 세션이 생성되면, 요청된 서비스 중 몇몇 서비스를 개시할 필요가 있다. 일단 서비스가 개시되면, 사용자가 시스템에 연결되어 있는지에 상관없이 서비스는 계속 활성 상태를 유지한다. 요청된 서비스의 균형(balance)은 사용자가 맨 처음 로그인할 때 개시될 수 있다.

사용자는 하나의 세션에 제한되지 않는다. 어느 시점에서 사용자와 관련된 세션은 여러 개일 수 있다. 세션 데이터베이스(220)는 사용자와 관련된 세션 내의 서비스(들)과 세션(들)을 식별하는 기록를 가지고 있다. 인에이블된 사용자는 시스템으로부터 제거될 수 있다. 사용자가 시스템으로부터 제거되면, 모든 사용자 관련 세션들이 시스템 및 세션 데이터베이스(220)로부터 제거된다. 사용자 세션과 관련된 서비스들도 역시 중지된다.

일단 사용자가 인에이블되어 시스템을 사용하면, 사용자는 네트워크 단말(202)을 통해 시스템에 로그온할 수 있다. 세션 관리자(206)가 인증 관리자(204)로부터 사용자가 네트워크 단말(202)에 접속되었음을 통지받으면, 세션 관리자(206)는 사용자의 세션(즉, 세션을 포함하는 서비스들)에게 통지한다. 세션 관리자(206)는 세션 데이터베이스(220)에 문의하여 세션의 서비스를 식별하고 통지한다. 예를 들어서, 세션 데이터베이스(220)는 세션(208)에 포함된 서비스(230-238)와 세션(208)을 식별하는 정보를 포함한다.

세션 데이터베이스(220)는 세션과 관련된 서비스 및 세션을 식별하는 동적 세션 기록과 영구 세션 기록을 포함한다. 세션 데이터베이스(220)는 하나 이상의 데이터베이스이거나 하나 이상의 데이터 저장소이다. 예를 들어서, 영구 세션 기록은 구성 파일(configuration file)에 저장되고 동적 세션 기록은 데이터베이스 시스템 내의 메모리에 저장될 수 있다. 영구 세션 기록은 사용자에 대한 구성 정보를 포함하며, 사용자가 인에이블되어 예컨대 시스템을 사용하는 시점에서 생성되는 것이 보통이다. 동적 세션 기록은 사용자와 관련된 서비스를 식별한다. 동적 세션 기록은 현재 활성 상태에 있는 서비스 뿐만 아니라 영구 세션 기록에 있는 사용자 세션과 관련된 요청 서비스를 식별한다. 본 발명의 일실시예에 따른 영구 세션 기록의 포맷은 다음과 같다.

sessionID serviceID serviceHost servicePort isLazy

sessionID 필드는 요청된 서비스(들)을 포함하는 세션을 유일하게 식별한다. serviceID 필드는 sessionID에 의해 식별된 세션과 관련된 서비스를 식별한다. serviceHost와 servicePort 필드들은 서비스가 실행되고 있는 서버와 서비스가 통신을 받을 수 있는 서버 상의 포트를 식별한다. isLazy 필드는 서비스가 개시되는 방식을 식별한다. 예를 들어서, isLazy는 세션의 생성과 동시에 즉시 시작되어야 하는 서비스 또는 사용자가 시스템에 맨 처음 접속했을 때 시작되어야 하는 서비스를 지정한다. serviceID, serviceHost, servicePort, isLazy 필드는 다중 발생(multiple occurence)이 존재할 수 있는데, 각각의 발생는 sessionID에 의해 식별된 세션과 관련된 요청 서비스를 식별한다.

동적 세션 기록은 세션 내에서 현재 실행되고 있는 서비스들과 세션에 대한 요청 서비스를 식별한다. 세션의 요청된 서비스는 예컨대, 영구 세션 기록에서 가져온다. 동적 세션 기록은 사용자를 위해 현재 실행되고 있는 영 또는 그 이상의 서비스 (요청된 서비스나 그렇지 않은 서비스)를 식별할 수 있다.

동적 세션 기록에서 서비스에 대한 정보를 저장하는 데에 사용되는 필드는 서비스가 요청된 서비스인가 아닌가에 따라 정해진다. 현재 활성 상태에 있는 요청된 서비스는 현재 서비스이다. 세션의 요청된 서비스를 식별하는 동적 세션 기록의 포맷은 영구 세션 기록 포맷과 동일하다. 본 발명의 일실시예에 따른, 현재 실행되고 있는 서비스와 관련된 기록에 대한 포맷은 다음과 같이 식별된다.

sessionLink TCPSocketfd requiredServiceLink serviceID

sessionLink 필드는 서비스의 세션을 식별한다. 개방 접속부, 즉, 파이프(pipe)는 세션 관리자(206)와 세션에서 현재 실행되고 있는 서비스 사이에 만들어진다. 개방 접속부를 사용하면, 세션 관리자(206)나 서비스에게 나머지는 비정상적으로 또는 다른 이유로 종료되었음을 통지할 수 있다. 본 발명의 일실시예에

서, 개방 접속부는 TCPSocketfd 필드에 의해 식별되는 TCP 소켓 접속부이다. 그러나, 접속이 디스에이블 또는 사라졌음을 통지할 수 있는 어떠한 형태의 신뢰성있는 접속 기술도 본 발명의 실시예에 사용될 수 있다는 사실은 명백하다.

서비스는 serviceID 필드에 저장되는 식별자(identifier)를 가진다. 현재 실행되고 있는 서비스는 요청된 서비스에 링크될 수 있다. 요청된 서비스에 대한 링크는 requiredServiceLink에 의해 식별된다. 만약 요청된 서비스에 대한 링크가 없으면, requiredServiceLink는 널(null)이다.

동적 세션 기록은 네트워크 단말[예컨대, 네트워크 단말(202)]에 대한 접속과 관련된 정보를 저장하는 데에 사용될 수 있다. 다음은 본 발명의 일실시예에 따라 접속을 시결하는 필드를 포함한다.

sessionLink Status IPAddress

다중 세션이 사용자와 관련될 수 있다. sessionLink 필드는 네트워크 단말(202)에 연결된 사용자가 현재 링크되어 있는 세션을 식별한다. sessionLink는 자신의 값으로서 예컨대, sessionID 값을 가질 수 있다. Status 필드는 세션에 대한 네트워크 단말(202)의 접속 상태 (즉, 접속되어 있는지 분리되어 있는지의 상태)를 식별한다. IPAddress 필드는 네트워크 단말(202)의 접속 네트워크 어드레스를 포함한다. IP 어드레스는 본 발명의 하나 이상의 실시예에서 사용된다. 그러나, 다른 어드레싱 기법을 사용하는 다른 접속 기술을 사용할 수 있음은 명백하다. 예를 들어서, 비동기식 전속 모드(ATM; Asynchronous Transfer Mode) 네트워크는 13-숫자 스위치 프리픽스/엔드 포인트(thirteen digit switch prefix/end point) 식별자를 사용할 수도 있다.

이러한 정보는 세션 관리자(206)가 상태 메시지를 네트워크 단말(202)에 보내는 데에 사용될 수 있다. 만약 네트워크 단말(202)이 특정 기간 내에 응답하지 않으면, 세션 관리자(206)는 사용자가 더 이상 네트워크 단말(202)을 사용하지 않는다고 가정하고 세션에 있는 각각의 서비스에게 분리 메시지를 보낸다.

세션 관리자(206)가 알고 있는 다른 정보에는 서비스에 대한 개방 접속부의 리스트(예컨대, 개방형 TCPsocketfd를 가지는 서비스)와 개방 접속부와 세션 사이의 매핑 및 세션 내의 서비스를 포함한다. 이러한 정보는 예컨대, 세션 기록으로부터 컴파일될 수 있다.

세션 관리자(206)가 사용할 수 있는 정보는 세션을 위치지정하는 데에 사용될 수 있다. 예를 들어서, 어떤 서비스가 주어졌을 때 이 서비스를 포함하는 세션 및/또는 어떤 세션 내에 포함되어 있는 서비스를 찾는 것이 가능하다. 또한, 현재 실행되고 있는지 여부에 관계없이 어떤 사용자 또는 네트워크 단말(202)의 인스턴스와 관련된 세션을 위치지정하는 것이 가능하다.

서비스 개시

세션 관리자(206)가 인증 관리자(204)로부터 사용자가 네트워크 단말(202)에 접속되어 있다는 메시지를 수신하면, 세션 관리자(206)는 현재 활성 상태에 있지 않은 요청 서비스를 개시한다. 세션 관리자(206)는 또한 현재 활성 상태에 있는 서비스에게 통지하여 입/출력(I/O)을 네트워크 단말(202)에 보내도록 한다. I/O는 네트워크 단말(202)과 그것의 주변 장치와 통신하는 데에 사용되는 명령 프로토콜을 사용하여 표현될 수 있다 (부록 A는 본 발명의 일실시예에 따른 명령 프로토콜의 일례를 포함한다).

서비스를 개시하기 위해, 세션 관리자(206)는 서비스가 실행될 서버에 접근하여 서비스를 시작한다. 예를 들어서, 세션 관리자(206)는 서버 상의 알려진 포트에 요청을 보내고 세션 관리자(206)에 대한 sessionHost, sessionPort 및 sessionID를 패스(pass)한다. 서버는 서비스에 접속된 네트워크 단말(202)에 접속하고, 서버의 순수(naive) 인증과 허가(permission)를 사용하여 사용자가 서버에 접근하도록 허용한다. 예를 들어서, UNIX 운영 환경인 경우, UNIX 서비스는 네트워크 단말(202)에 디스플레이되는 "CDE Login" 스크린으로 시작하여 사용자를 인증하고 사용자가 접속하기를 원하는 서비스를 보증한다.

세션 관리자(206)가 서버 상의 서비스를 시작하기 위하여, 서비스를 시작하는 데에 필요한 특권(privilege)이 부여된다. 세션 관리자(206)에게 이러한 특권을 부여하는 것이 바람직하지 않을 수도 있다. 또한, 현재 네트워크 환경에서, 서버는 서로 다른 운영 환경에서 동작하고 있을 수도 있다. 이런 경우, 세션 관리자(206)는 서비스를 시작하기 위해 각각의 운영 환경의 절차(procedure)에 대해 알고 있어야 한다.

또는, 서버 상에서 실행되는 세션-인식 애플리케이션(session-aware application)이 상기 시작 동작을 수행하고 세션 관리자(206)로 서비스를 등록한다. 이 경우, 세션 관리자(206)는 상기 필요한 특권을 가질 필요가 없다. 또한, 세션 관리자(206)는 다중 운영 환경에서 서비스를 개시하기 위해 중앙집중형 모델을 구현하지 않아도 된다. 서비스를 개시할 책임은 다른 운영 환경에서 구동되고 있는 세션-인식 애플리케이션에게 있다. 세션-인식 애플리케이션은 세션 관리자(206)에 대한 지식[예컨대, 세션 관리자(206)의 sessionID, sessionHost 및 sessionPort] 및 그것의 인터페이스(예컨대, 메시지 포맷)을 가지고 있다.

세션-인식 서버 애플리케이션은 세션 관리자(206)로부터 접수한 요청에 응답하여 서비스를 개시한다. 세션 관리자(206)는 서버 운영 한경에서 서비스를 시작할 허용권을 가지고 있는 서버 애플리케이션에게 개시 메시지를 보낸다. 서버 애플리케이션은 세션 관리자(206)에 대한 서비스를 개시하고 유효 sessionID를 가진 세션 관리자(206)에 응답한다. 예를 들어, UNIX 시스템과 NT 시스템에서, sessionID는 운영 환경에서 사용가능하게 될 수 있다. 비디오 윈도우와 같은 서비스는 예컨대, 이런 방식으로 개시될 수 있다.

또는, 세션-인식 애플리케이션은 서비스에 접촉하여 암호화 서명된 인증(cryptographically signed authorization) 형태로 허용을 받을 수 있다. 서버 애플리케이션은 sessionID와 상기 서명된 인증을 세션 관리자(206)에게 보낼 수 있다. 만약, 세션-인식 애플리케이션이 서비스에 대한 설명만 있고 인증은 없는 상태로 세션 관리자(206)에게 접촉하면, 세션 관리자(206)는 네트워크 단말(202)로부터의 승인(approval)을 요청하여 사용자가 서비스를 인증하였음을 보증한다. 사용자가 긍정적으로 응답하여 서비스는 세션에 추가된다.

세션 관리자 메시지

세션 관리자(206)는 메시지를 수신하고 생성하여 세션 내의 서비스를 관리한다. 서비스를 개시하는 데에는 본 명세서에 설명된 것과 다른 기술을 사용할 수 있다. 세션 관리자(206)가 서비스를 개시하면 개시 메시지를 서버(또는 세션-인식 서버 애플리케이션)에게 보낸다. 세션 관리자(206)는 개시 메시지를 생성하여 예컨대, 세션 데이터베이스(220)에서 식별된 요청 서비스를 시작할 수 있다. 다른 예로서, 세션 관리자(206)는 개시 메시지를 보내 종료된 것으로 판단된 요청 서비스를 다시 활성 상태로 만들 수 있다[예컨대, 세션 관리자(206)와 서비스 사이의 개방 TCP 접속부를 통해].

세션 관리자(206)는 네트워크 단말(202)이 시스템에 성공적으로 접속하면 접속 메시지를 수신한다. 접속 메시지에 응답하여, 세션 관리자(206)는 요청된 서비스가 모두 시작되었음을 검증하고 실행되고 있지 않은 서비스를 개시한다. 세션 관리자(206)는 세션 내에 있는 서비스에게 메시지(예컨대, 접속 메시지)를 보내서, I/O가 네트워크 단말(206)에 향하도록 한다.

분리 메시지가 수신되면 세션 관리자(206)는 분리 메시지를 세션 내에 있는 각각의 서비스에게 보내서 I/O를 네트워크 단말(202)에 보내는 것을 종료하도록 한다.

세션 관리자(206)는 상태 메시지를 네트워크 단말(202)에 주기적으로 송부하여 네트워크 단말(202)이 아직 연결되어 있는지를 확인한다. 예를 들어서, 세션 관리자(206)는 세션 데이터베이스(220)의 동적 세션 기록을 조사하여 현재 네트워크 단말과 연결되어 있는 각각의 세션을 식별한다. 즉, 세션 관리자(206)는 세션 데이터베이스(220)의 동적 세션 기록에 있는 네트워크 단말 관련 상태 필드를 조사할 수 있다. 세션 관리자(206)는 세션과 연결된 각각의 네트워크 단말에게 상태 요청 (예컨대, "핑(ping)")을 보낸다. 네트워크 단말(202)로부터 일정한 시간(예컨대, 20초) 동안 특정 세션에 대한 응답을 받지 못하면, 세션 관리자(206)는 이 세션은 디스에이블되었다고 가정하고 분리 메시지를 세션 내의 각각의 서비스에 보내 디스플레이 기능을 종료하도록 지시한다.

네트워크 단말(202)는 세션 관리자(206)의 상태 요청(예컨대, 핑)에 대해 "Card In" 또는 "Card Out" 상태로 응답한다. 만약, "Card Out" 상태를 네트워크 단말(202)로부터 수신한 경우에는, 세션 관리자(206)는 분리 메시지를 각각의 세션의 서비스에게 보낸다.

상태 요청에 응답하여 "Card In" 상태가 보내진 경우, 네트워크 단말(202)은 카드 판독기(216)에 카드가 삽입된 횟수, 카드 삽입 후 경과한 시간(초) 및 cardID를 나타낸다. cardID는 예컨대, 사용자의 세션에 대한 sessionID이다. 세션 관리자(206)는 최소한 네트워크 단말(202)로부터 수신된 상태 정보를 보유하여 새로운 상태 정보를 과거 상태 정보와 비교한다. 만약에, 예를 들어서, 카드의 삽입 회수 또는 경과 시간(초)이 마지막 상태 정보와 다르다면, 세션 관리자(206)는 세션이 디스에이블되어야 하는지를 검토한다. 이 경우, 세션 관리자(206)는 분리 메시지를 세션의 서비스에게 보낸다.

서비스가 예컨대, 세션-인식 서버 애플리케이션에 의해 시작된 경우에는 서비스 연결 메시지가 세션 관리자(206)에게 송부된다. 서비스가 적절한 권한을 가지면, 세션 관리자(206)는 이 서비스를 세션에 대한 서비스 리스트에 추가하고 메시지를 서비스에게 보내서 I/O가 네트워크 단말(202)로 향하게 한다.

인증 관리자

인증 관리자는 사용자의 합법성을 보증하는 것과 세션과 사용자를 연관시키는 책임을 진다. 본 발명의 일실시예에 따르면, 개시 과정 동안 (이것은 이하에 좀 더 자세하게 설명함) 인증 교환이 일어나서 사용자를 인증한다. 인증은 시스템에 대한 사용자의 신원을 검증한다. 예를 들어서, 키 암호를 입력하거나 생체 정보를 수집하여 사용자를 인증한다.

인증 데이터베이스(218)는 인증 관리자(204)가 접근할 수 있는 사용자 정보와 세션 정보를 가진다. 본 발명의 일실시예에서, 인증 데이터베이스(218)에 포함된 기록 포맷은 다음과 같다.

userID secret PIN sessionHost sessionPort sessionID

userID 필드와 secret 필드는 사용자의 스마트 카드에 저장된 것과 동일한 값을 가진다. userID 값과 secret 값은 예컨대 사용자가 시스템을 사용할 수 있도록 인에이블된 경우에 만들어지는 것이 보통이다. 본 발명의 일실시예에서, secret 필드는 128-비트 값을 가진다. PIN 필드는 사용자가 알고 있는 개인 고유번호(PIN)이고, 인증 과정 동안 인증 관리자(204)에 의해 요청된다. userID 값, secret 값 및 PIN 값들은 사용자를 인증하는 데에 사용된다. 인증 데이터베이스(218)는 암호 데이터나 생체 데이터(이것이 사용자를 인증하는 데에 사용되었다면)와 같은 다른 정보를 포함할 수도 있다.

sessionHost 필드는 사용자의 현재 세션을 관리하고 있는 세션 관리자(206)를 실행하고 있는 전산 서비스 제공자(예컨대, 서버)를 식별한다. sessionPort 필드는 세션 관리자(206)와 통신하는 포트를 식별한다. sessionID 필드는 세션 관리자(206)에 대한 고유 식별자를 포함한다. 만약, 인증이 성공적이면, sessionHost 필드, sessionPort 필드 및 sessionID 필드를 사용하여 세션 관리자(206)에게 네트워크 단말(202)에 있는 사용자의 위치를 통지한다.

본 발명의 일실시예에서는 챌린지 메카니즘을 사용하여 사용자를 인증한다. (도 6은 본 발명의 일실시예에 따른 챌린지 프로세스 흐름을 나타낸다.) 인증 관리자(204)는 네트워크 단말(202)에 챌린지를 보내서 사용자의 인증을 검증한다. 네트워크 단말(202)은 챌린지 응답을 준비하고 이것을 인증 관리자(204)에게 돌려보낸다. 만약 챌린지에 대한 응답이 기대했던 것과 동일한 경우에는 인증 관리자(204)에게 사용자를 검증한다.

도 5A 및 도 5B는 본 발명의 일실시예에 따른 인증 프로세스 흐름을 나타낸다. 인증 프로세스는 인증이 성공할 때까지 또는 인증 과정의 반복 횟수(즉, 라운드)가 일정 횟수를 초과할 때까지 반복된다. 단계(502)에서, 인증 라운드의 횟수를 나타내는 식별자는 영으로 초기화된다. 단계(504)에서, 챌린지 번호로 사용될 난수(random number)가 생성된다. 단계(506)에서, 인증 관리자(204)는 N_AUTHENTICATE 명령을 인증 프로세스에 대한 정보의 패키지와 함께 네트워크 단말(202)에게 보낸다.

본 발명의 일실시예에서, N_AUTHENTICATE 명령과 함께 다음의 정보가 송부된다.

code identifier length valueSize value

code 필드는 정보 패킷에 들어 있는 정보의 유형을 식별한다. 예를 들어서, 값 "1"은 정보 패킷에 챌린지가 들어 있음을 나타낸다. identifier 필드는 단계(502)에서 생성된 값(즉, 라운드 지시자)를 포함한다. length 필드는 정보 패킷의 길이를 식별한다. value 필드에는 단계(504)에서 생성된 난수 즉, 챌린지의 값이 들어 있다. valueSize는 value 필드의 크기(예컨대, 128 비트)를 식별한다.

단계(508)에서 인증 관리자(204)는 네트워크 단말(202)에게 랜더링 명령을 보내서 사용자에게 사용자의 PIN을 입력하도록 안내한다. 단계(510)에서, 인증 관리자(204)는 네트워크 단말(202)에서 응답이나 타임아웃을 기다린다.

타임 아웃이 단계(510)에서 발견되면, 단계(514)로 진행하여 라운드의 최대값이 초과되었는지를 판단한다. 초과하지 않았으면, 단계(518)로 진행하여 식별자를 증가시키고 단계(504)로 진행하여 새로운 인증 라운드를 시작한다. 한편, 단계(514)에서 라운드의 최대값이 나타나면, 단계(516)로 진행하는데, 여기서는 인증 관리자(204)가 네트워크 단말(202)에게 랜더링 명령을 보내 실패를 표시하고 인증 프로세서는 종료한다. 랜더링 명령은 예컨대, 네트워크 단말(202)과 그 주변 장치 사이의 통신에 사용되는 명령 프로토콜의 일부이다.

챌린지 루틴은 사용자의 PIN 입력을 받고 응답을 받으라는 인증 관리자(204)가 네트워크 단말(202)에게 보낸 명령을 포함한다. 네트워크 단말(202)은 사용자의 PIN, 식별자의 값, 사용자의 스마트 카드에 저장된 암호 및 챌린지의 값[예컨대, 단계(504)에서 생성된 난수]를 포함하는 입력으로부터 나온 해쉬 함수(hash function)의 출력인 응답 값(즉, 해쉬 값 또는 챌린지 응답)을 생성한다.

해쉬 함수는 가변 길이 입력을 가질 수 있고 이것을 고정 길이 출력(해쉬 값)으로 변환한다. 해쉬 함수의 일례에서는 해쉬 함수가 입력을 취하여 모든 입력 바이트의 XOR(exclusive-OR)로 구성된 바이트를 되돌려 보낸다. 본 발명의 실시예에 사용될 수 있는 해쉬 함수는 여러 가지가 될 수 있다. hmac_md5 함수(RFC2104)는 응답을 생성하기 위한 본 발명의 실시예에 사용될 수 있는 해쉬 함수의 한 예이다.

다음의 패킷 포맷은 본 발명의 일실시예에 따라 인증 관리자(204)에게 응답을 보내는 네트워크 단말(202)에 의해 사용되는 포맷이다.

code identifier length valueSize value userID

code 필드는 값 "2"로 설정되어 있는데, 이것은 정보 패킷이 챌린지 응답을 포함하고 있음을 나타낸다. value 필드는 챌린지 응답 (예컨대, 해싱 기능(hashing function)의 결과)을 포함한다. userID 필드에는 사용자의 userID가 포함된다.

만약 인증 관리자(204)가 네트워크 단말(202)로부터 응답을 받았다고 판단한 경우에는, 단계(512)로 진행하여 네트워크 단말(202)에 의해 되돌아온 식별자가 인증 관리자(204)가 생성한 식별자와 일치하는지를 판단한다. 일치한다면, 단계(520)로 진행하여 네트워크 단말(202)에 의해 되돌아온 응답을 조사한다.

단계(520)에서, 인증 관리자(204)는 챌린지 응답이 인증 관리자(204)가 기대했던 응답과 일치하는지 판단한다. 예를 들어서, 인증 관리자(204)는 자신의 식별자, PIN 값, secret 값 및 챌린지 값을 사용하여 해쉬 값을 생성한다. 인증 관리자(204)가 생성한 해쉬 값이 네트워크 단말(202)에서 생성된 챌린지 응답과 일치하는 경우에는, 인증이 부분적으로 성공한 것이다. 인증 관리자는 또한 네트워크 단말(202)의 접속 네트워크 어드레스와 사용자의 userID가 유효한지 검증한다. 챌린지 응답, 접속 네트워크 어드레스 및 userID가 유효한 것으로 검증되면, 인증은 성공한 것이다. 그렇지 않으면, 인증은 실패한 것이다.

인증이 성공하면, 단계(528)로 진행하여 N_AUTHENTICATE 명령을 보낸다. 본 발명의 일실시예에 따른 명령의 포맷은 다음과 같다.

code identifier length

code 필드는 값 "3"을 포함하는데, 이것은 사용자가 성공적으로 인증되었음을 나타낸다. 단계(530)로 진행하여 랜더링 명령을 네트워크 단말(202)로 보내는데, 이것은 세션 관리자(206)가 사용자를 사용자의 세션 중 하나로 연결시키고 있음을 나타낸다. 단계(532)에서, 인증 관리자(204)는 세션 관리자(206)에게 사용자가 네트워크 단말(202)을 통해 시스템에 연결되어 있음을 나타낸다. 인증 관리자(204)는 단계(532)에서 네트워크 단말(202)의 접속 네트워크 어드레스 및 세션 관리자(206)의 sessionID를 세션 관리자(206)를 실행하고 있는 서버(즉, 사용자의 인증 데이터베이스 기록의 sessionHost 필드에서 식별된 서버)로 보낸다.

만약 인증이 실패하면, 단계(522)로 진행하여 N_AUTHENTICATE 명령을 보낸다. 인증이 성공한 경우와 마찬가지로, N_AUTHENTICATE 명령은 인증 프로세스의 상태를 나타내는 코드 필드를 포함한다. 인증이 실패했음을 나타내기 위해서 예컨대 코드 값 "4"를 사용할 수 있다. 단계(524)로 진행하여 랜더링 명령을 네트워크 단말(202)에게 보내는데, 이것은 인증이 실패했음을 나타내고 사용자에게 카드 판독기(216)에서 카드를 제거하라고 지시한다.

인증 프로세스는 단계(526)에서 종료한다.

도 5A, 5B를 참조로 지금까지 설명한 프로세스는 인증 프로세스의 일례이다. 본 발명의 실시예에서 다른 인증 기술을 사용할 수 있음은 물론이다. 다른 실시예에서, 사용자는 PIN의 입력을 요청받지 않는다. 카드 판독기(216)에 있는 사용자의 카드만으로 사용자를 충분히 인증할 수 있다. userID와 secret 값을 인증 관리자(204)로부터 받은 챌린지와 식별자로 해쉬하여 인증 관리자(204)에 의한 챌린지에 대한 응답을 생성한다. 이렇게 하면, 사용자는 유효 정보가 들어있는 카드를 카드 판독기(202)에 집어 넣기만 함으로써 사용자의 서비스에 접속할 수 있다.

또한, 본 발명의 실시예는 사용자의 인증이 수행되지 않는 경우에도 사용될 수 있다. 예를 들어서, 신뢰된 또는 보안화된 환경에서는 사용자의 인증을 검증할 필요가 없다. 따라서, 본 발명의 일실시예에서 사

용자는 인증 관리자(204)에 의해 가장 먼저 인증되지 않고서도 세션에 접속된다. 사용자는 예컨대, 고유 번호(예컨대, userID)를 제공하기만 하면 된다. 사용자가 유효한 userID를 제공한다면 그 사용자는 userID와 관련된 세션에 접속된다.

사용자가 네트워크 단말(202)에서 분리되면, 인증 관리자(204)는 통지를 받고 세션 관리자(206)에게 분리를 통지한다. 예를 들어서, 사용자가 카드 판독기(216)에서 스마트 카드를 제거하면 카드 판독기(216)는 네트워크 단말(202)에게 통지한다. 네트워크 단말(202)은 인증 관리자(206)에게 분리를 통지한다. 인증 관리자(204)가 세션 관리자(206)에거 사용자가 네트워크 단말(202)로부터 분리되었음을 통지한다. 세션 관리자(206)는 사용자 세션에 있는 각각의 서비스에게 통지한다.

챌린지 루틴(challenge routine)

인증 프로세스는 인증 관리자(204)에 의해 개시되는 챌린지를 포함한다. 도 6은 본 발명의 일실시예에 따른 챌린지를 다루는 챌린지 루틴 프로세스 흐름을 나타낸다. 챌린지 루틴은 인증 관리자(204)로부터 수신한 챌린지 명령에 응답하여 네트워크 단말(202) 상에서 실행된다.

단계(602)에서 사용자로부터 받은 키 입력을 리턴 키 또는 엔터 키가 눌러질때까지 읽는다. 키 입력은 단계(604)에서 아스키(ASCII) 문자로 변환된다. 단계(606)에서, identifier 값, PIN 값, secret 값 및 챌린지 값이 연결된 것으로부터 해쉬 함수를 사용하여 해쉬 값 또는 챌린지 응답을 생성한다. 챌린지 응답은 단계(608)에서 인증 관리자(204)로 전송된다. 단계(610)에서, 네트워크 단말(202)은 인증 관리자(204)로부터의 응답 또는 타임 아웃을 기다린다. 응답이나 타임 아웃이 발생하면, 챌린지 루틴은 단계(614)에서 종료한다.

네트워크 단말 초기화

네트워크 단말(202)은 맨 처음 턴온되었을 때 초기화를 수행한다. 사용자가 네트워크 단말(202)을 사용하지 않은 동안, 네트워크 단말(202)은 전원이 온 상태인 경우에는 휴지(dormant) 상태에 있을 수 있다. 사용자는 예를 들어서 본 명세서에 설명되어 있는 기술 중 하나를 사용하여 휴지 상태에 있는 네트워크 단말(202)을 깨울 수 있다. 네트워크 단말을 깨우는 데에는 다른 기술도 물론 사용할 수 있다.

도 3은 본 발명의 일실시예에 따라 전원 연결 동작에 응답하여 네트워크 단말(202)을 초기화하는 프로세스 흐름을 나타낸다. 단계(302)에서는 전원 연결 동작이 일어났는지를 판단한다. 전원이 연결되지 않으면 전원이 연결될 때까지 기다린다. 단계(304)에서, 네트워크 단말(202)은 요청을 생성하여 네트워크에 보내 네트워크 접속을 검사한다. 단계(306)에서는 응답이 수신되었는지를 판단한다. 응답이 수신되지 않으면, 단계(310)로 진행하여 에러를 발생하고 단계(302)로 진행하여 전원 연결 동작을 기다린다.

단계(306)에서 응답이 수신된 것으로 판단되면, 단계(308)로 진행하여 수신확인(ACK) 메시지를 보내고 네트워크 단말(202)의 초기화가 도 4A의 단계(402)에서 계속된다.

도 4A-4C는 본 발명의 일실시예에 따라 기상 동작(awaken operation)에 응답하여 네트워크 단말(202)을 초기화하는 프로세스 흐름을 나타낸다. 도 4A를 참조하면, 네트워크 단말은 기상 동작의 통지를 기다린다. 본 발명의 일실시예에서, 기상 동작은 카드 판독기(216)에 사용자의 스마트 카드를 삽입하는 동작에 해당한다.

스마트 카드가 카드 판독기(216)에 삽입되었다고 판단되면, 단계(404)에서 인증 관리자(204) 및 네트워크 단말(202)의 접속 네트워크 어드레스를 구하는 요청을 보낸다. 또는, 사용자의 스마트 카드를 접속 네트워크 어드레스로 프로그램해 두는 것도 가능하다. 네트워크 단말(202)은 예를 들어서, 카드 판독기(216)를 통해 스마트 카드로부터 접속 네트워크 어드레스를 읽을 수 있다.

단계(406)에서, 네트워크 단말(202)은 응답 또는 타임 아웃을 기다린다. 타임 아웃이 발생하면, 단계(412)로 진행하여 최대 시도 횟수가 초과되었는지를 판단한다. 최대 시도 횟수가 초과되었으면, 단계(410)로 진행하여 에러를 발생한다. 최대 시도 횟수가 초과되지 않았으면, 단계(414)로 진행하여 시도 횟수를 증가시키고 단계(404)로 진행하여 접속 네트워크 어드레스 요청을 다시 보낸다.

요청에 대한 응답이 수신되면 단계(408)로 진행하여 ACK를 보낸다. 프로세스는 도 4B의 단계(416)로 진행한다. 이 단계(416)에서 네트워크 단말(202)은 시작 요청을 인증 관리자(204)에게 보낸다. 단계(418)에서는 네트워크 단말(202)이 시작 요청에 대한 응답을 기다리는 재시도 시간이 세트된다. 단계(420)에서 변수를 세트하여 네트워크 단말(202)이 시작 요청에 대한 응답을 기다리고 있음을 나타낸다. 단계(422)에서 네트워크 단말(202)은 시작 요청에 대한 응답을 기다린다.

응답이 수신되지 않았다고 판단되면, 단계(424)로 진행하여 재시도 시간이 초과하였는지 판단한다. 초과하지 않았으면, 단계(422)로 진행하여 응답을 기다린다. 재시도 시간이 초과하였으면, 단계(426)로 진행하여 최대 시도 횟수가 초과하였는지 판단한다. 초과하지 않았으면, 단계(428)로 진행하여 에러를 발생하고 단계(416)로 되돌아가서 시작 요청을 다시 보낸다. 초과하지 않았으면 단계(430)로 진행하여 시도 횟수를 증가시키고 재시도 시간을 리세트한다. 단계(432)에서 시작 요청을 재전송되고 단계(444)로 진행하여 카드가 카드 판독기(216)에서 제거되었는지 판단한다.

단계(422)에서 응답이 수신되었다고 판단되면, 도 4C의 단계(434)로 진행한다. 단계(434)에서 네트워크 단말(202)은 단계(420)에서 초기에 세트된 변수를 조사하여 시작 요청에 대한 응답을 기다리고 있는지 판단한다. 기다리고 있다면, 단계(436)로 진행하여 응답이 챌린지 응답인지 판단한다. 챌린지 응답이 아니면 단계(424)로 진행하여 최대 시도 횟수가 초과하지 않는 경우 시작 요청을 반복한다. 단계(436)에서 챌린지 응답이 수신되었다고 판단되면, 단계(438)로 진행하여 waiting_for_startup 변수를 무(즉, "N")로 세트한다. 단계(440)로 진행하여 단계(440, 442)의 챌린지 요청을 처리한다. 챌린지 요청은 예컨대, 도 5A-5B 및 도 6을 참조로 앞에서 설명했던 것처럼 처리될 수 있다.

단계(434)에서 네트워크 단말(202)이 시작 요청에 대한 응답을 기다리고 있지 않다고 판단되면, 단계(440, 442)로 진행하여 메시지[예컨대, 서비스(234)에 의해 생성된 출력을 디스플레이하는 랜더링 명령]

를 처리한다.

단계(444)에서, 사용자가 카드 판독기(216)에서 카드를 제거하였는지 판단한다. 사용자가 카드 판독기(216)에서 카드를 제거하였으면, 네트워크 단말(202)은 분리 메시지를 단계(448)에서 인증 관리자(204)에게 보낸다. 네트워크 단말(202)은 인증 관리자(204)로부터의 수신 확인 (ACK) 메시지를 기다린다. ACK 메시지가 수신되면 네트워크 단말(202)은 단계(450)에서 스크린을 클리어(clear)하고 단계(402)로 되돌아가 다른 사용자가 카드 판독기(216)에 스마트 카드를 삽입하는 것을 기다린다.

단계(444)에서 사용자가 카드 판독기(216)로부터 카드를 제거하지 않았다고 판단되면, 단계(446)로 진행하여 네트워크 단말이 그것의 시작 요청을 기다리고 있는지 판단한다. 기다리고 있다면, 단계(422)로 진행하여 응답이 수신되었는지 판단한다. 네트워크 단말이 시작 요청에 대한 응답을 기다리고 있지 않다면, 단계(440, 442)로 진행하여 네트워크 단말(202)로 전송된 모든 메시지를 처리한다.

메시지 포맷

본 발명의 일실시예에서, 네트워크 단말에 대한접속은 사용자 데이터그램 프로토콜(UDP; User Datagram Protocol) 포트를 통해 이루어진다. 즉, UDP 접속부를 통해 패킷이 전송되고 목적(destination) UDP 포트에서 패킷이 수신된다. 목적 UDP 포트는 접속부를 유일하게 식별한다. 패킷 길이와 체크섬(checksum) 정보는 UDP 헤더에 의해 제공된다. 버퍼 크기는 IP/UDP 헤더를 갖는 이서네트 최대 전송 유니트(Ethernet Maximum Transfer Unit) (MTU)에 적합하다. 데이크는 네트워크 바이트 순서(big-endian)로 네트워크 상에서 전송된다.

UDP 대신에 다른 프로토콜을 사용할 수도 있다. 예를 들어서, ATM AAL5 [AAL 또는 ATM 적응 계층(Adaptation Layer)]를 사용하는 것도 가능하다.

지금까지 세션 관리와 사용자 인증에 대해 설명하였다. 본 명세서에 설명된 특정 실시예는 단지 설명을 위한 것이고 발명을 제한하려는 것이 아니다. 본 발명은 청구범위와 이것의 균등범위에 의해 제한된다.

부록 A

렌더링 명령

유선 프로토콜 명령 포맷

모든 데이터는 네트워크 바이트 오더(비그엔디언(big-endian))로 네트워크상에 보내지고 비트-필드(bit-fields)가 MSB에서 LSB로 패킷된다.

다음은 기본적인 렌더링 명령 포맷이다.
<COMMAND:8> <SEQUENCE:24> <X:16> <Y:16> <WIDTH:16> <HEIGHT:16> <Info>

COMMAND	Code	<Info> 설명
Set	0xA1	WIDTH*HEIGHT of 32-bit values <X,B,G,R> [WIDTH*HEIGHT <= 512 pixels]
Fill	0xA2	one 32-bit value <X,B,G,R>
Glyph	0xA3	one 32-bit value <X,B,G,R> (HEIGHT*CEILING(WIDTH/8)) bytes of bitmap [i.e. each line padded to 8 bits] [WIDTH*HEIGHT <= 2048 pixels]; the entire command is padded to the next 32-bit boundary
Copy	0xA4	<FROM_X:16> <FROM_Y:16>
Bilevel	0xA5	two 32-bit values c0, and c1, <X,B,G,R>, followed by (HEIGHT*ceiling(WIDTH/8)) bytes of bitmap [i.e. each line padded to 8 bits] [WIDTH*HEIGHT <= 2048 pixels]; the entire command is padded to the next 32-bit boundary
Set24	0xA6	WIDTH*HEIGHT of packed 24-bit values <B,G,R> [WIDTH*HEIGHT <= 512 pixels] padded to the next 32-bit boundary
Set YUV Image	0xA7	<SOURCE_W:16> <SOURCE_H:16> <RFU:8> <LUMA_ENCODING:2> <CHROMA_SUB_X:3> <CHROMA_SUB_Y:3> followed by (SOURCE_W*SOURCE_H) pixels Y [luma] with each line padded to a byte boundary, and

		(ceiling(SOURCE_W / x_subsample) * ceiling (SOURCE_H / y_subsample)) bytes each of 8-bit signed U and V [chroma] in CCIR-601 value encodings; the entire command is padded to the next 32-bit boundary; [SOURCE_W * SOURCE_H <= 1024 pixels]; [SOURCE_W <= WIDTH]; [SOURCE_H <= HEIGHT];
Set Cursor	0xA9	two 32-bit values c0, and c1, <X,B,G,R>, followed by two sets of (HEIGHT * ceiling(WIDTH/8)) bytes of bitmap [i.e. each line padded to 8 bits] [WIDTH & HEIGHT <= 64 pixels each]. The first bitmap is the pixel values, the second is the per-pixel mask. The entire command is padded to the next 32-bit boundary.
Set Pointer	0xAA	<INDEX:8> <DIM:2> <PAD:6> { <Z:16> { <P:16> <R:16> <E:16> <PAD:16>}} | <PAD:16> 모든 값은 2의 보수로 할당된다. 각도는 -180에서 +180-(1 lsb)=+179.9945까지 범위의 값을 갖는다(모든 범위를 포함하는 각도). WIDTH, HEIGHT는 무시된다.
Set Key Locks	0xAB	X, Y, WIDTH, HEIGHT는 무시된다. <INDEX:8> <LOCKS: 8> <PAD:16>
Damage Repair	0xAC	<EPOCH:32> <PAD:8> <SEQ:24>
Play Audio	0xB1	X,Y, WIDTH, HEIGHT는 다음과 같이 부호화된다: X:4 audio sequence number X:12 interleave offset Y total sequence length-1 WIDTH:4 mixer mode specifies the # of channels to include in the standard mix. Channel numbers above this

number are sent raw and not
combined with any other channel
if the terminal has insufficent
channels to cover the request.

WIDTH:12 packet len in samples
max 2000 bytes

HEIGHT:4 number of channels-1
HEIGHT:12 interleave size-1

The header is followed by the specified
number of samples x number of
channels x 16 bits.

전체 명령어는 32비트로 형성된다.

상술한 일련 번호는 각 명령에 대하여 증가된다.
일련 번호들은 아래에 설명된 플러시 명령을 변화시키는 에픽(epoch) 동안을 제외하고 모두 제로가 되지 않는다. 사각형은 겹쳐지지 않는다.
즉, x + width < 0x10000이고 y + height < 0x1000이다.

하나의 추가적인 정보 명령은 다른 포맷으로 정의되어 있다:

<COMMAND:8> <SEQUENCE:24> <EPOCH:32> <FILL:16 * 8>

COMMAND	Code
Flush	0xAF

플러시 명령의 일련 번호는 에픽 변화를 제외하고 이전 명령의 일련 번호와 동일하다(아래 설명 참조). 즉, 일련 번호들은 픽셀들이 변화하거나 에픽이 변화할 때에만 증가된다.

명령 설명

명령	설명
Set	<x, y> <width, height>로 정의된 사각형을 아래의 픽셀 값으로 설정한다. 그 영역 내의 각 픽셀마다 하나의 픽셀값이 있다. 레이아웃은 열들에 의한 것이다; 즉, <x+width-1, y+1> 등을 통해 <x, y+1>에서 픽셀들에 의해 이어지는 <x+width-1, y>를 통해 <x, y>에서의 픽셀에 대한 "폭" 픽셀값들이다. <0, 0>은 상부 왼쪽 코너를 나타낸다.
Fill	<x, y> <width, height>에 의해 정의된 사각형 내의 모든 픽셀들을 단일 32-비트 값으로 설정.
Glyph	32-비트 값은 비트맵에서 각각의 비트와 대응하는 픽셀 위치에 놓여지고, 제로 비트와 결합된 위치들은 변하지 않는다. 비트맵은 매바이트마다 MSB에서 LSB를 이용하여 열들(y, y+1, ...)에 의해 레이아웃된다.
Copy	<from_x, from_y> <width, height>에 의해 한정된 사각형을 <x, y> <width, height>에 의해 한정된 사각형으로 복사. 클라이언트는 오버랩핑 영역을 올바르게 복사하는 것을 확보해야한다(예를 들면, Solaris bstring(3)).
Bilevel	두 개의 32비트 값들 c0, c1은 비트맵내에서 각 제로 및 하나의 비트와 대응하는 픽셀 위치에 각각 놓여진다. 비트맵은 매바이트마다 MSB에서 LSB를 이용하여 열들(y, y+1, ...)에 의해 레이아웃된다.
Set24	<x, y> <width, height>에 의해 정의된 사각형을 아래의 픽셀 값들로 설정. 그 픽셀 값들은 세개의 32비트 값들 <bgrb, grbg, rbgr>로 정의된 네개의 픽셀들이 존재하도록 패킷된다. 폭이 4의 배수가 아니면 끝부분이 위와 같이 남은 값들로 패킷되고 가장 근접한 32비트 값이 된다. 그 영역 내에 각 픽셀마다 하나의 픽셀 값이 존재한다. 레이아웃은 열들에 의한 것이다; 즉, <x+width-1, y+1> 등을 통해 <x, y+1>에서 픽셀들에 의해 이어지는 ((3 * width + 3) / 4) 32-bit words) 내의 <x+width-1, y>를 통해 <x, y>에서의 픽셀들에 대한 "폭" 픽셀 값들이다. <0, 0>은 상부 왼쪽 코너를 나타낸다.

Set YUV Image

<x, y> <width, height>로 정의된 사각형을 다음과 같이 제공된 픽셀 값으로 설정. source_w × source_h 픽셀들의 CCIR/ITU.BT-601 Y'CbCr (or YUV) 포맷 내의 이미지는 RGB로 디코드된다. 채도 엘리먼트는 수평 및/또는 수직 치수들로 서브샘플되고, 변환전에 업-샘플되어야 한다.

CHROMA_SUB_X 및 CHROMA_SUB_Y(각각의 x_subsample 및 y_subsample)의 값들은 다음과 같이 부호화된다:

0 - No chroma values; monochrome image
1 - Subsample by 1 (i.e. no subsample)
2 - Subsample by 2
3 - Subsample by 4
4-7 - Undefined/reserved

다음은 LUMA_ENCODING 값들이다:

0 - Y(luma) is specified by 8-bit unsigned data
1 - Y(luma) consists of 4-bit quantized DPCM values(see below)
2,3 - Undefined/reserved

RFU는 장래 사용을 위해 보존되고 0이 되어야 한다.

해독 후에, RGB 이미지는 폭 대 높이 픽셀들에 대해 필요한 만큼 일정 비율로 확대된다. 얻어진 이미지는 위치 <x, y>에서 화면상에 표시된다.

주: CHROMA_SUB_X 및 CHROMA_SUB_Y 모두가 제로이면, 이미지는 단색(luma만)이고 U 또는 V 데이터는 존재하지 않는다. 하나가 제로로 설정되고 나머지가 제로가 아닌 것은 무효하다.

성분 순서는 Y (또는 CCIR-601 Y'), U (CCIR-601 Cb), 다음에 V (CCUR-601 Cr)이다.

Set Cursor

이 명령은 로컬 표시 커서(Pointer[0]에 의해 움직여지고 기록되는)의 형태를 설정한다. 커서는 최대 64×64 블록이지만, 그것 이외의 다른 크기가 될 수 있다. 특정 픽셀에 대한 마스크 값이 '1'이면, 대응 커서 픽셀이 표시된다; 이 마스크가 '0'이면, 커서는 그 위치에서 투명하다. 마스크가 '1'이면 픽셀 값은 'c0'이고, 그 값이 '0'이면 'c1'이다. 마스크가 제로이면 픽셀 값도 제로가 되어야 한다. 제로 마스크 및 하나의 픽셀 값은 장래의 확장을 위해 저장된다.

WIDTH 및 HEIGHT 는 제로가 되며 커서를 당기지 않음을 나타낸다(모든 제로의 마스크와 동일하다). 포인터 트랙킹은 정상적으로 작동한다.

X 및 Y는 커서의 '핫 스팟(hot spot)'을 의미한다; 즉, 커서 이미지 이벤트의 픽셀이 무엇인가에 따라 결과가 알려진다. 이는 주로 화면의 가장자리상에서 커서를 정지시킬 때 사용된다. X[0, WIDTH], Y[0, HEIGHT].

Set Pointer

포인터의 위치를 정한다. Pointer[0]은 대개 고정(마우스 또는 터치 스크린)될 수 있으며 2-D 스크린 커서이다. 이 명령은 그 포인터를 세팅하려는 어플리케이션용, 또는 대응 포인터가 필요한 어플리케이션용으로 제공된다(예를 들어, 커서를 이전 위치에 리셋한다). 이와 같이, 몇가지 제한을 둔다:

. 포인터를 세팅작업이 전혀 되지 않는다 (예를 들어, 조이스틱).

. 그 포인터 값이 포인터 장치 또는 스크린에 매치되도록 임의대로 절단될 수 있다.

. 사용자는 포인터가 설정된 다음에 포인터를 계속 움직일 수 있지만, 'Pointer State' 상태 메시지를 이용하여 기록된다.

. 허위(pseudo)-대응 모드에 대한 포인터를 재설정하는 것에 대한 비헤이비어는 다른 장치를 갖는 다른 비헤이비어를 발생시킨다; 예를 들면, 사용자가 '드래깅'하지 않을 때에 터치 스크린은 설정가능하다.

포인터는 6개의 디멘전까지를 포함하도록 허용된다. 명령에 대한 영역 및 크기의 수는 DIM 비트를 이용하여 설정된다. 모든 포인터 값은 2의 보수로 표시된다.

Set Key Locks

이 명령은 <INDEX> 키보드에 대한 락(lock) 값을 설정한다. 락은 일반적으로 소프트웨어적으로 제어될 수 있는 키모드상의 라이트에 대응한다. 락 상태가 표시되면, 비트는 마스트 내에서 설정되어야 하고, 다른 한편으로는 비트가 제거되어야 한다. 일부 키보드가 국부적으로(예를 들면, 기계적으로) 락을 실행하기 때문에, 락 설정이 영향을 받지 않는다. 키보드로부터의 키들은 키보드에 의해 입력되는 상태로부터 항상 인터프릿트되어야 한다. 한편, 인터페이스가 잠겨진 키코드를 감지하면, 정상 키보드 및 터미널 모두가 국부적으로 락킹(locking)을 다루지 않기 때문에, 호스트는 잠겨진 키코드의 수신시에 Set Key Lock 명령이 나오도록 요구한다.

키 락 비트맵은 부트 키보드용의 USB 클래스 정의로부터 나온다:

0x01	Num Lock
0x02	Caps Lock
0x04	Scroll Lock
0x08	Compose
0x10	Kana

모든 다른 비트들은 유지된다.
-- 읽기 무시, 설정 제로

Damage Repair

이것은 에펙 EPOCH에서 및 앞서 진행되고 보수된 데이터를 보낸 이전 상태에서 일련 번호 SEQ에 대한 모든 손상 메시지를 클라이언트에게 알린다
(Damage back-channel command 참조).
PAD는 0이 되어야 하고, X, Y, WIDTH 및 HEIGHT는 0이 되어야 한다.

Play Audio

이것은 48㎑ 오디오 샘플을 연주하고, 그래픽 명령 스트림 내에 내장된다.

first-come-first-served 원리하에서 터미널에 의해 비한정 스트림이 수신된다. 스트림은 as-needed 원리하에서 할당되고 버퍼 결핍이 발생하면 정지된다(시간이 지나면 진행할 데이타가 없다. - 부분적으로 수신된 버퍼는 감추어진 에러이다). 터미널은 타임베이스 드리프트를 정정한다.

데이타는 네트워크 에러 감추기에 도움이 되도록 끼워넣는 방법으로 보내진다. 샘플 시퀀스는 인터리브 사이즈로 쪼개지고 많아야 1+(시퀀스 사이즈) / (인터리브 사이즈) 샘플이 패킷당 방출된다. 샘플은 다음과 같이 선택된다:

```
Sample sequence[sample_size];
int seq_number = 0;

while (1) {
  get_samples(sequence, sample_size);

  for (i = 0; i < interleave_size; i++) {
    interleave_offset
      = random_select(0..interleave_size);

    packet=new_packet(seq_number, sample_size,
                      num_chan, num_chan,
                      interleave_size,
                      interleave_offset);
    for (j = interleave_offset; j < sample_size;
         j += interleave_size)
       emit(packet, sequence[j]);
    send_packet(packet);
  }
  seq_number = (seq_number+1)%16;
}
```

패킷들이 보내지는 순서가 임의로 될 수 있다는 것에 주목하라

예를 들면, 3의 인터리브 및 8의 스퀀스 사이즈에 대하여 다음의 세개의 패킷이 보내질 수 있다:

```
 (samples)          (0 1 2 3 4 5 6 7)
pkt 1,  off 1:        1     4     7
pkt 2,  off 0:       0     3     6
pkt 3,  off 1:           2     5
```

이 시퀀스는 터미널이 언제 에러를 감추고 샘플 시퀀스가 방출하는지를 알도록 번호매겨진다.

샘플은 48㎑, 16비트 리니어(linear)이고 16 채널까지 포함할 수 있다. 예를 들면, 5-채널 샘플이 10개의 연속 바이트들을 취한다.

터미널에 의해 지원되는 오디오 채널의 수에 대한 제한은 없지만, 현재까지는 16 채널까지 한번에 보내질 수 있다. 터미널이 지원하는 채널수와는 다른 채널수가 보내질 수 있기 때문에, 첫번째 8 채널에 대한 기준 믹스의 개념이 도입된다. 이것은 함께 혼합되지 않는 소정의 인덱스된 채널을 격리하는 "MIX"를 설정하여 사용할 수 없다. 마지막 8 채널이 소리가 들리도록 처음 8 채널과 동일한 방법으로 혼합된다. 충분한 채널이 있으면, 결과적으로 터미널 설정상태에 의존된다.

표준 할당된 채널은 다음과 같다:

```
            channel ->
  # chan     0   1   2   3   4   5   6   7
       1     mono
       2     l   r
       3     l   r   sw
       4     l   r   rl  rr
       5     l   r   rl  rr  sw
       6     l   r   rl  rr  sw  cf
       7     l   r   rl  rr  sw  cf  top
       8     l   r   rl  rr  sw  cf  cl  cr
(l=left, r=right, r[lr]=rear{left,right}
sw=subwoofer, cf=center fill,
c[lr]=center{left,right}, top=center-center
```

예를 들면, 두 개의 스피커가 있고 하나의 채널이 이용가능한 표준 믹스(mix)로 보내진다면, 하나의 채널은 왼쪽 스피커와 오른쪽 스피커 모두에 보내질 수 있다. 반대로, 동일한 터미널이 6 채널을 보내었다면, 채널 0,2,4,5가 혼합되어 왼쪽 스피커에 보내질 것이며 채널 1,3,4,5가 혼합되어 오른쪽 스피커에 보내질 것이다.

터미널 스피커는 동일한 방법으로 조정된다.

전체가 혼합된 매트릭스가 전체 사양에서 이용될 수 있다.

Flush

현재 명령을 따르는 시간 주기 동안에 디스플레이 스트림 내에 명령이 없는 경우이다; 따라서, 이것은 클라이언트가 화면에 모든 진행중인 렌더링을 중지시키기에 좋다. 에픽 범위는 시퀀스 번호마다 32 추가 하이 오더 비트(32 additional high order order bits)를 제공한다. FILL은 모드 0xFF로 설정된 16바이트로 구성된다. 이 명령은 시스템이 정지된 이후에 데이타 스트림을 재동기화시키는 기회를 제공한다.

플러시 명령의 시퀀스 번호는 일반적으로 마지막 넌(non)-플러시 명령과 동일하다. 하지만, 에픽이 소모되면, (즉, 마지막 명령의 시퀀스 번호가 0xFFFFFF), 제로의 시퀀스 번호와 (1씩 증가되는) 새로운 에픽 번호를 포함한 플러시 명령이 보내진다.

백-채널 명령(Back-channel Commands)

유선 프로토콜 상태 메시지 포맷

다음은 기본적인 상태 명령 포맷이다:
<COMMAND:8> <TIME:24> <Info>

COMMAND	Code	<Info> 설명
Keyboard State	0xc1	<INDEX:8> <COUNTRY CODE:8> <LOCKS:8> <MODIFIERS:8> <KEYCODE:8>[8]
Pointer State	0xc2	<INDEX:8> <DIM:2> <BUTTONS:6> <X:16> {<Y:16> {<Z:16> {<P:16> <R:16> <E:16>}}} 모든 값은 2의 보수로 할당된다. 각도는 -180에서 +180-(1 lsb)=+179.9945까지 범위의 값을 갖는다 (모든 범위를 포함하는 각도).
Active Region	0xc3	<X:16> <Y:16> <WIDTH:16> <HEIGHT:16>
Damage	0xc4	<EPOCH:32> <PAD0:8> <SEQ_L:24> <PAD1:8> <SEQ_H:24>

DIM	Dimensions
0	X
1	X, Y
2	X, Y, Z
3	X, Y, Z, P, R, H (yaw)

주: 시간은 마이크로초 단위이고; 그것은 2**24 다음에 겹친다(대략 16초).

Pointer State

<INDEX>된 포인터의 상태를 기록한다. DIM은 기록된 디멘젼(dimension)의 수를 나타낸다: 1, 2, 3, 또는 6. 버튼(button)은 부트 키보드용 USB 클래스 데피니션을 따르며, 비트 제로는 '프라이머리' 버튼(좌측에서)이고, 좌측에서 우측으로 그 수가 증가한다. 기록된 값은 모두 절대값이며 2의 보수로 표시된다.

Active Region

뉴트(newt) 상에서 저장된 논리 프레임 버퍼의 영역을 나타낸다. 특히, 이것은 카피 렌더링 명령의 "프럼(from)"영역이 성공적으로 특정 지워질 수 있는 영역이다.

예를 들어, 핸드-헬드 장치(hand-held device)내의 인터페이스의 팬-앤-스캔 스타일 때문에 주어진 클라이던트에 대해 시간에 따라 변할 수 있다. 또한, 다른 클라이언트 장치는 다른 활성 영역을 리포트 할 수 있다.

Damage

클라이언트가 에펙 EPOCH내의 일련번호 SEQ_E를 포함하여 이를 통해 일련번호 SEQ_L 통하여 서버로부터 다운스트림(렌더) 명령을 받지 못함을 나타낸다. PAD0 및 PAD1은 반드시 0이다.

클라이언트는 데미지 리페어 메시지가 손상된 일련번호를 받을 때까지 데미지를 기록한다.

SEQ_L이 0이면, 현재 화면의 이미지 전체를 보내야 한다.

일단 주어진 일련번호에 대한 데미지 메시지를 보내면, 초기 일련 번호에 대하여 새롭게 나중에 발생한 데미지는 보낼 수 있다. 그러나, 최근 상태 패킷(packets)에 공간을 확보하기 위하여 두 개 이상의 영역을 하나로 만들 수 있다.

DPCM YUV 설명:

LUMA_ENCODING으로 YUV 데이터의 추가적인 압축이 가능하다.

LUMA 데이터는 다음과 같이 부호화된다:

```
for each line
        last_value = 0x80
        foreach  luma-value l in line
                diff = l - last_value
                q_value = quant[diff]
                last_value = clamp[last_value + dquant[q_value]]
                emit q_value
        end
end
```

Luma 데이터는 다음과 같이 디코드된다:

```
for each line
        last_value = 0x80
        foreach  quantization-value q_value in line
                last_value = clamp[last_value + dquant[q_value]]
                emit last_value
        end
end
```

클램프(clamp)는 클램핑 테이블이다; clamp[i] is:

```
0       if i < 0
255     if i > 255
i       otherwise.
```

다음은 사용된 퀀티자이저(quantizer)이다:

Difference	code	rquant
-255 to -91	0	-100
-90 to -71	1	-80
-70 to -51	2	-60
-50 to -31	3	-40
-30 to -16	4	-20
-15 to -8	5	-10
-7 to -3	6	-4
-2 to 0	7	-1
1 to 2	8	1
3 to 7	9	4
8 to 15	10	10
16 to 30	11	20
31 to 50	12	40
51 to 70	13	60
71 to 90	14	80
91 to 255	15	100

(57) 청구의 범위

청구항 1

컴퓨터 시스템에서 세션들을 관리하는 방법으로서,

사용자와 관련된 세션에 대한 정보를 유지하는 단계,

상기 세션에서 최소한 하나의 서비스(이 서비스는 사용자가 시스템으로부터 분리되어 있는 동안에도 실행될 수 있음)를 개시하는 단계,

상기 컴퓨터 시스템의 인간 인터페이스 장치에 사용자가 연결될 때 및 사용자가 인간 인터페이스 장치로부터 분리될 때, 이것을 상기 최소한 하나의 서비스에에 통지하는 단계를 포함하며,

상기 최소한 하나의 서비스는 사용자가 인간 인터페이스 장치에 연결되어 있는 동안에 인간 인터페이스 장치에게 그 출력을 보내는 것인 세션 관리 방법.

청구항 2

제1항에서, 상기 정보는 상기 최소한 하나의 서비스의 고유 번호(identification)을 포함하는 것인 세션 관리 방법.

청구항 3

제2항에서, 상기 정보는 상기 최소한 하나의 서비스가 활성 상태인지 및 상기 최소한 하나의 서비스가 상기 세션의 요청된 서비스인지를 식별하는 것인 세션 관리 방법.

청구항 4

제3항에서, 상기 최소한 하나의 서비스는 이것이 요청된 서비스일 때 사용자가 시스템에 접속함에 따라 개시되는 것인 세션 관리 방법.

청구항 5

제1항에서, 상기 정보는 사용자 고유 번호 및 인증 정보를 포함하는 것인 세션 관리 방법.

청구항 6

제1항에서, 상기 인간 인터페이스 장치로부터 사용자가 분리되었을 때 인간 인터페이스 장치에 대한 출력의 전송을 차단하는 상기 최소한 하나의 서비스를 더 포함하는 세션 관리 방법.

청구항 7

제1항에서, 상기 정보를 사용하여 사용자를 인증하는 단계를 더 포함하는 세션 관리 방법.

청구항 8

시스템으로서,

컴퓨터 시스템에서 실행될 수 있는 서비스,

상기 서비스에게 입력을 전송하고 서비스의 출력을 수신할 수 있는 네트워크 단말,

상기 사용자가 네트워크 단말에 접속했을 때 및 상기 사용자가 네트워크 단말에서 분리되었을 때 이를 통지하도록 구성된 세션 관리자를 포함하며,

상기 서비스는 사용자가 네트워크 단말에 접속했을 때 및 사용자가 네트워크 단말에서 분리되었을 때 네트워크 단말에게 출력을 보내고, 네트워크 단말에게 출력을 보내는 것을 단절함과 동시에 실행을 하도록 구성된 것인 시스템.

청구항 9

제8항에서, 상기 네트워크 단말의 사용자를 확인하도록 구성된 인증 관리자를 더 포함하는 시스템.

청구항 10

제9항에서, 상기 인증 관리자는 정당한 사용자가 네트워크 단말에 접속했을 때 이를 세션 관리자에게 통지하도록 구성된 것인 시스템.

청구항 11

제8항에서, 상기 세션 관리자는 네트워크 단말에 질의(inquiry)를 보내서 사용자가 네트워크 단말에 접속되어 있는지를 판단하도록 구성된 것인 시스템.

청구항 12

제8항에서, 상기 서비스를 포함하며 상기 사용자와 관련된 세션을 더 포함하는 시스템.

청구항 13

제8항에서, 복수의 서비스를 포함하며 사용자와 관련된 최소한 하나의 세션을 더 포함하는 시스템.

청구항 14

컴퓨터 프로그램 제품으로서,

세션 관리와 인증을 위한 컴퓨터 판독가능한 프로그램 코드가 내장된 컴퓨터 사용가능한 매체를 포함하며, 상기 매체는,

사용자와 관련된 세션에 대한 정보를 컴퓨터가 유지하도록 구성된 컴퓨터 판독가능한 프로그램 코드와,

상기 세션에 있는 최소한 하나의 서비스(이 서비스는 사용자가 시스템에서 분리되어 있는 동안에도 실행을 할 수 있음)를 컴퓨터가 개시하도록 구성된 컴퓨터 판독가능한 프로그램 코드와,

상기 사용자가 컴퓨터 시스템의 인간 인터페이스 장치에 연결되었을 때 및 사용자가 인간 인터페이스 장치로부터 분리되었을 때, 이를 상기 최소한 하나의 서비스에게 컴퓨터가 통지하도록 구성된 컴퓨터 판독가능한 프로그램 코드와,

상기 사용자가 인간 인터페이스 장치에 연결되어 있는 동안 인간 인터페이스 장치에게 서비스의 출력을 컴퓨터가 보내도록 구성된 컴퓨터 판독가능한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

청구항 15

제14항에서, 상기 정보는 상기 최소한 하나의 서비스의 고유 번호(identification)을 포함하는 것인 컴퓨터 프로그램 제품.

청구항 16

제15항에서, 상기 정보는 상기 최소한 하나의 서비스가 활성 상태인지 및 상기 최소한 하나의 서비스가 세션의 요청된 서비스인지를 식별하는 것인 컴퓨터 프로그램 제품.

청구항 17

제16항에서, 상기 최소한 하나의 서비스는 상기 최소한 하나의 서비스가 요청된 서비스인 경우 사용자가 시스템에 접속함으로써 개시되는 것인 컴퓨터 프로그램 제품.

청구항 18

제14항에서, 상기 정보는 사용자 고유번호 및 인증 정보를 포함하는 것인 컴퓨터 프로그램 제품.

청구항 19

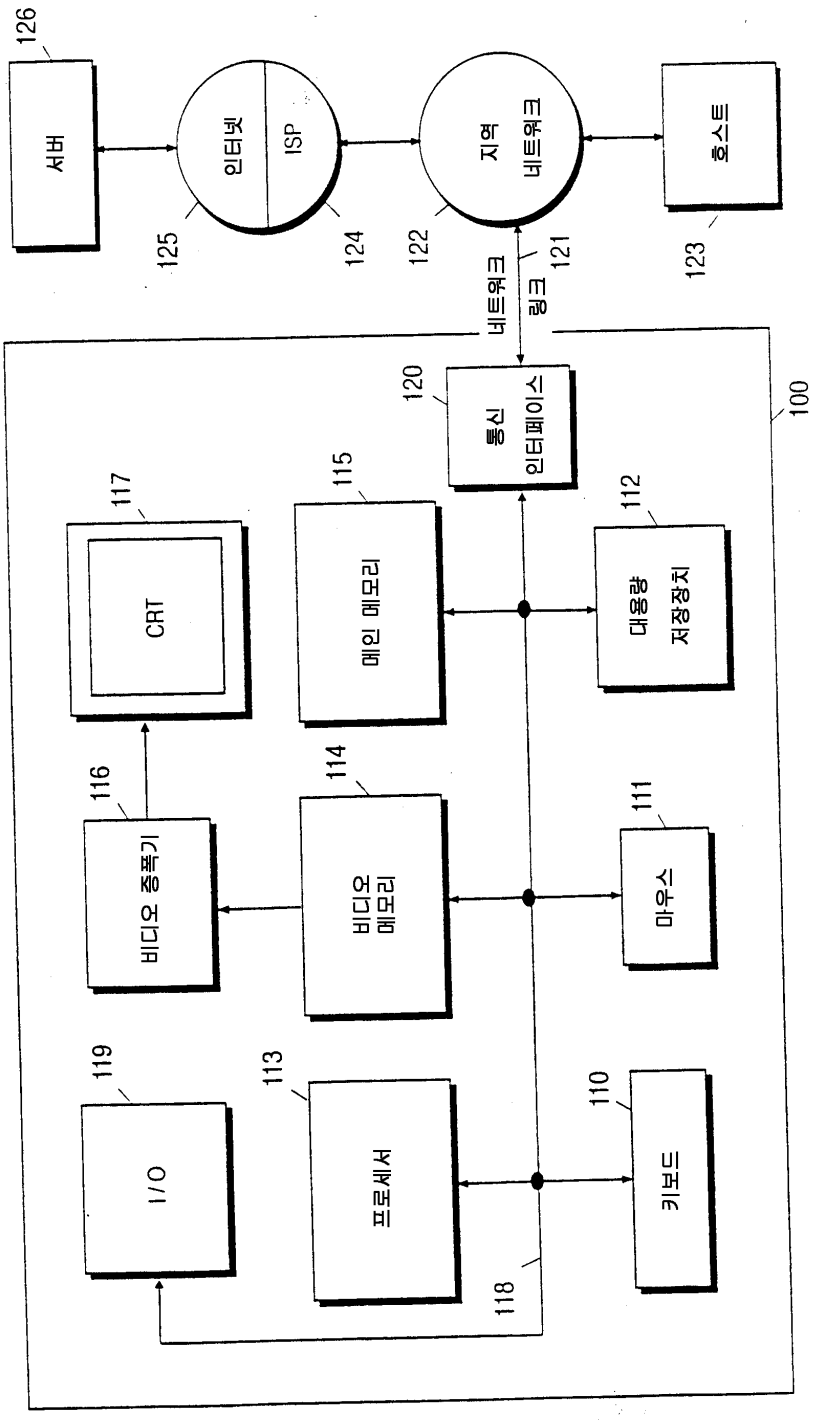
제14항에서, 상기 사용자가 인간 인터페이스 장치로부터 분리되었을 때 상기 최소한 하나의 서비스가 인간 인터페이스 장치에 출력을 전송하는 것을 분리시키도록 구성된 컴퓨터 판독가능한 프로그램 코드를 더 포함하는 컴퓨터 프로그램 제품.

청구항 20

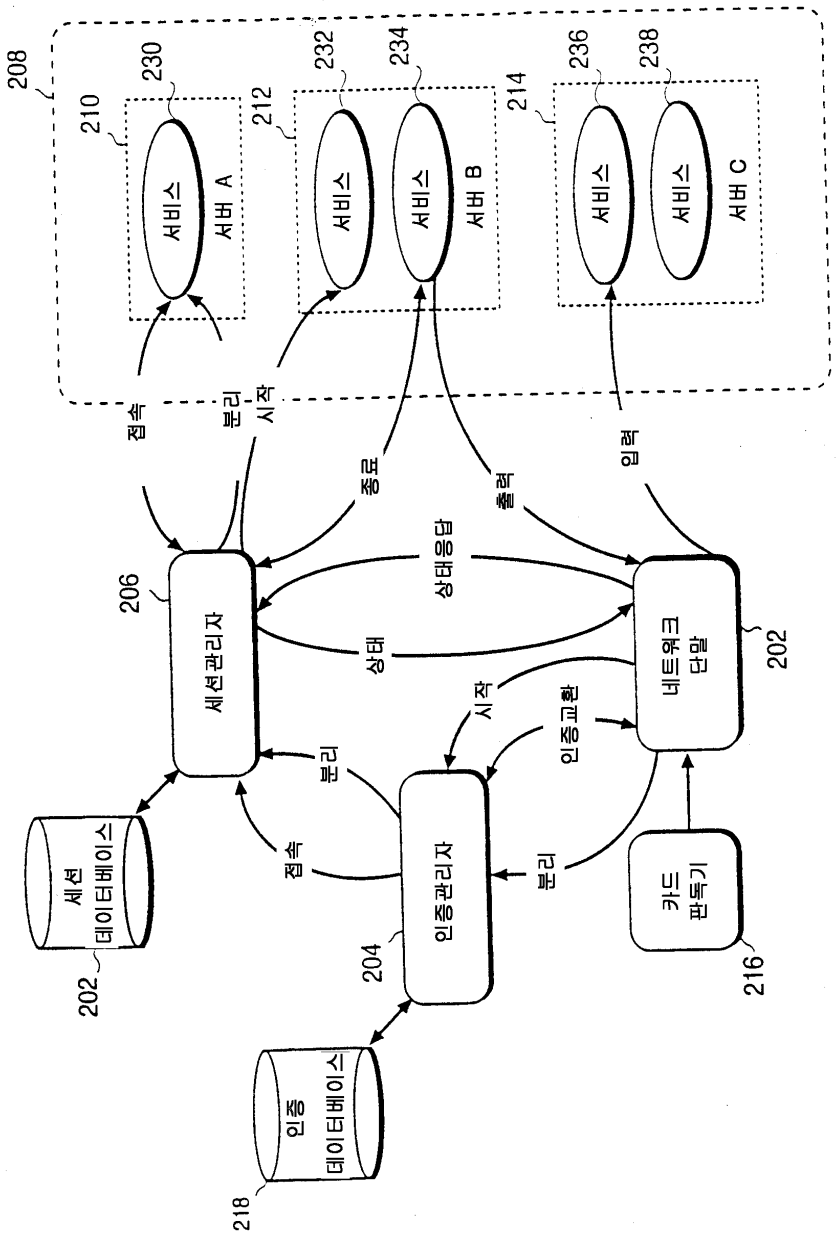
제14항에서, 상기 정보를 사용하여 컴퓨터가 사용자를 인증하도록 구성된 컴퓨터 판독가능한 프로그램 코드를 더 포함하는 컴퓨터 프로그램 제품.

도면

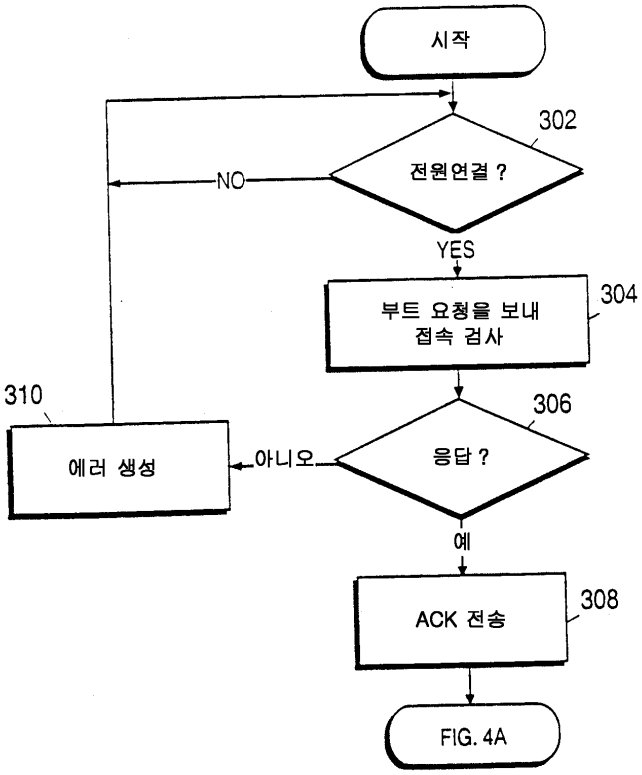
도면1


126
서버
인터넷
ISP
125
124
지역
네트워크
122
호스트
123
121
링크
네트워크
100
120
통신
인터페이스
115
메인 메모리
112
대용량
저장장치
117
CRT
116
비디오 증폭기
114
비디오
메모리
111
마우스
119
I/O
113
프로세서
110
키보드
118

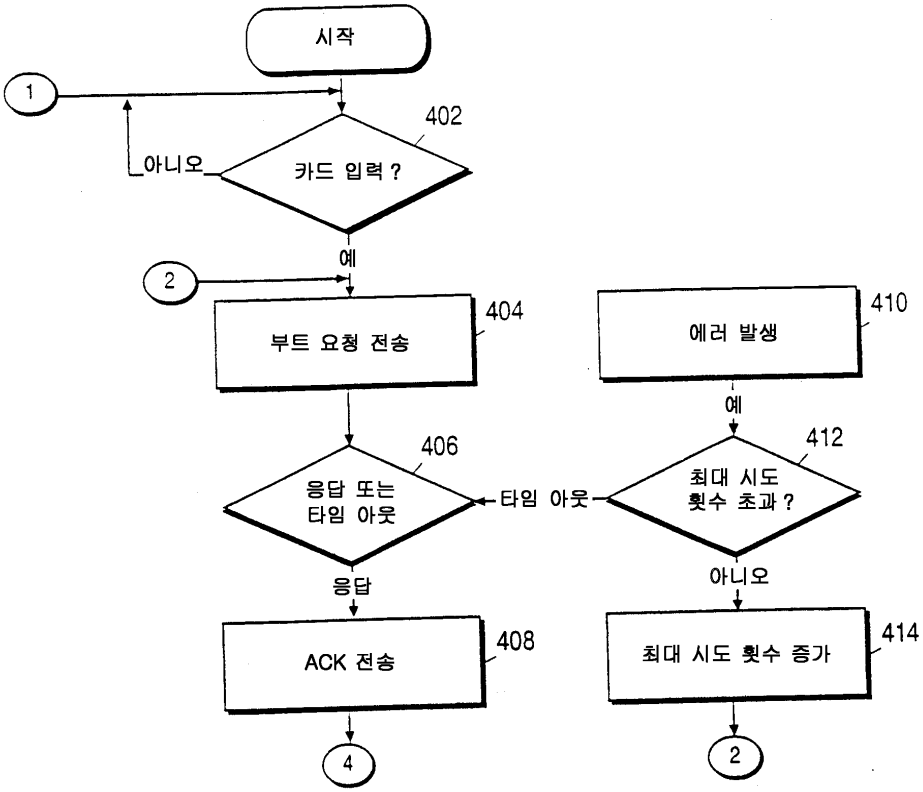
도면2


208
210
230
서버 A
서비스
212
232
234
서버 B
서비스
서비스
214
236
238
서버 C
서비스
서비스
접속
분리
시작
종료
출력
입력
상태응답
상태
206
세션관리자
202
네트워크
단말
시작
인증교환
분리
접속
분리
세션
데이터베이스
202
204
인증관리자
카드
판독기
216
인증
데이터베이스
218

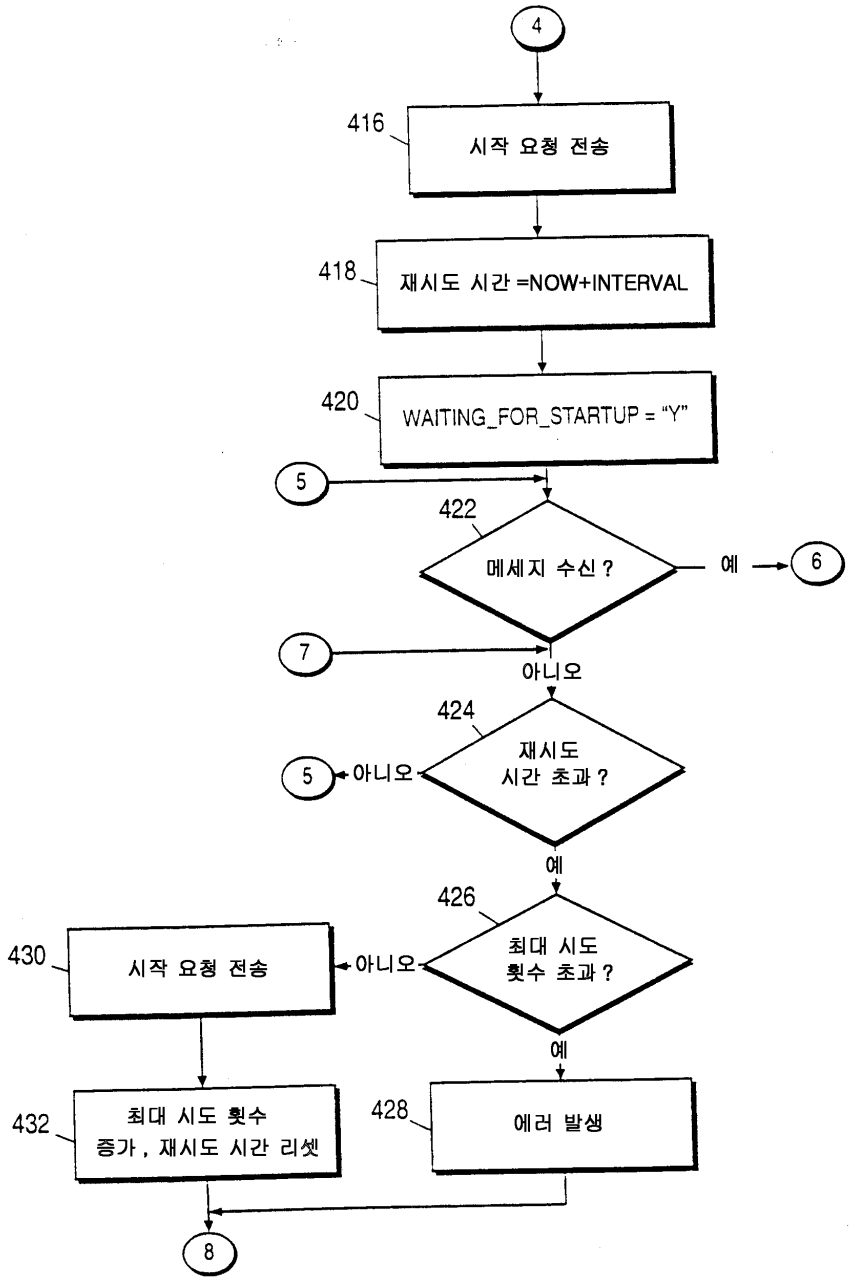
도면3


시작
302
전원연결 ?
NO
YES
304
부트 요청을 보내
접속 검사
310
306
에러 생성
아니오
응답 ?
예
ACK 전송
308
FIG. 4A

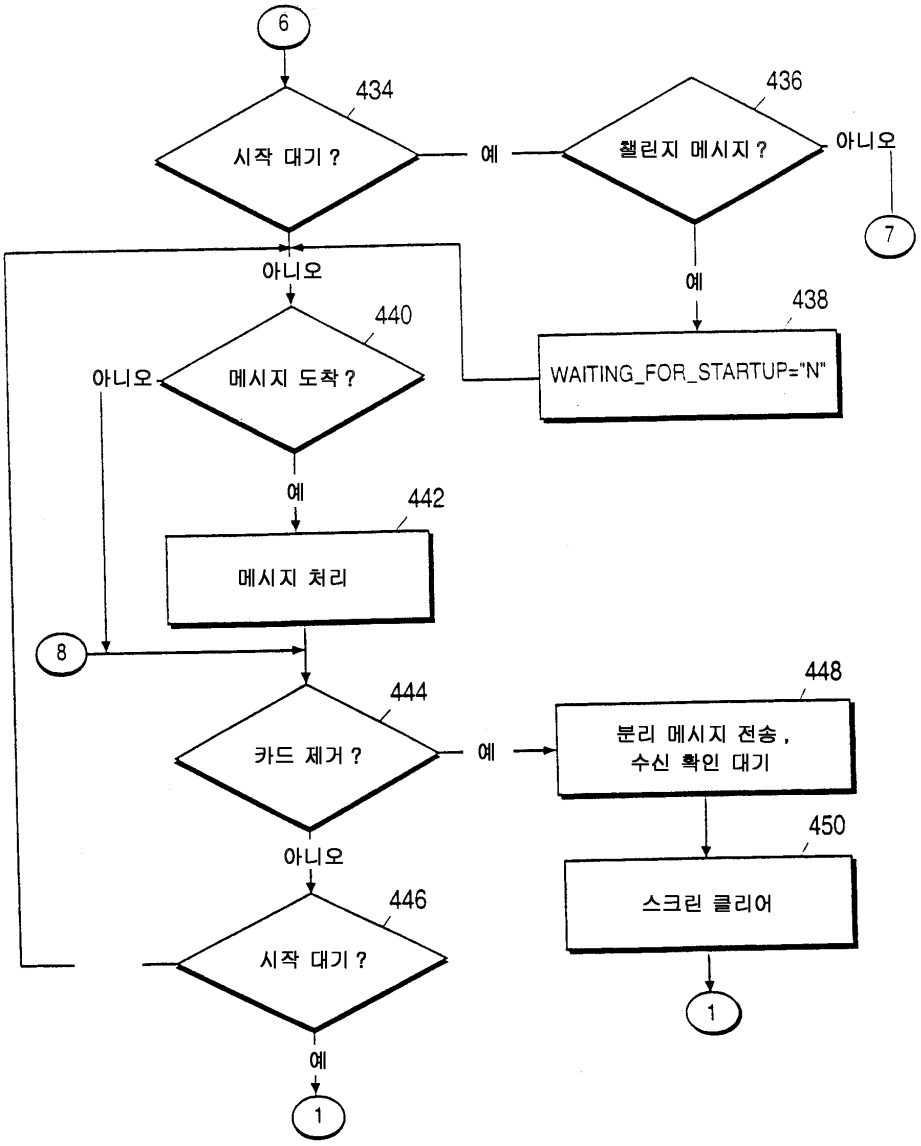
도면4A


시작
1
402
아니오
카드 입력 ?
예
2
부트 요청 전송
404
406
응답 또는
타임 아웃
타임 아웃
응답
ACK 전송
408
4
에러 발생
410
예
412
최대 시도
횟수 초과 ?
아니오
최대 시도 횟수 증가
414
2

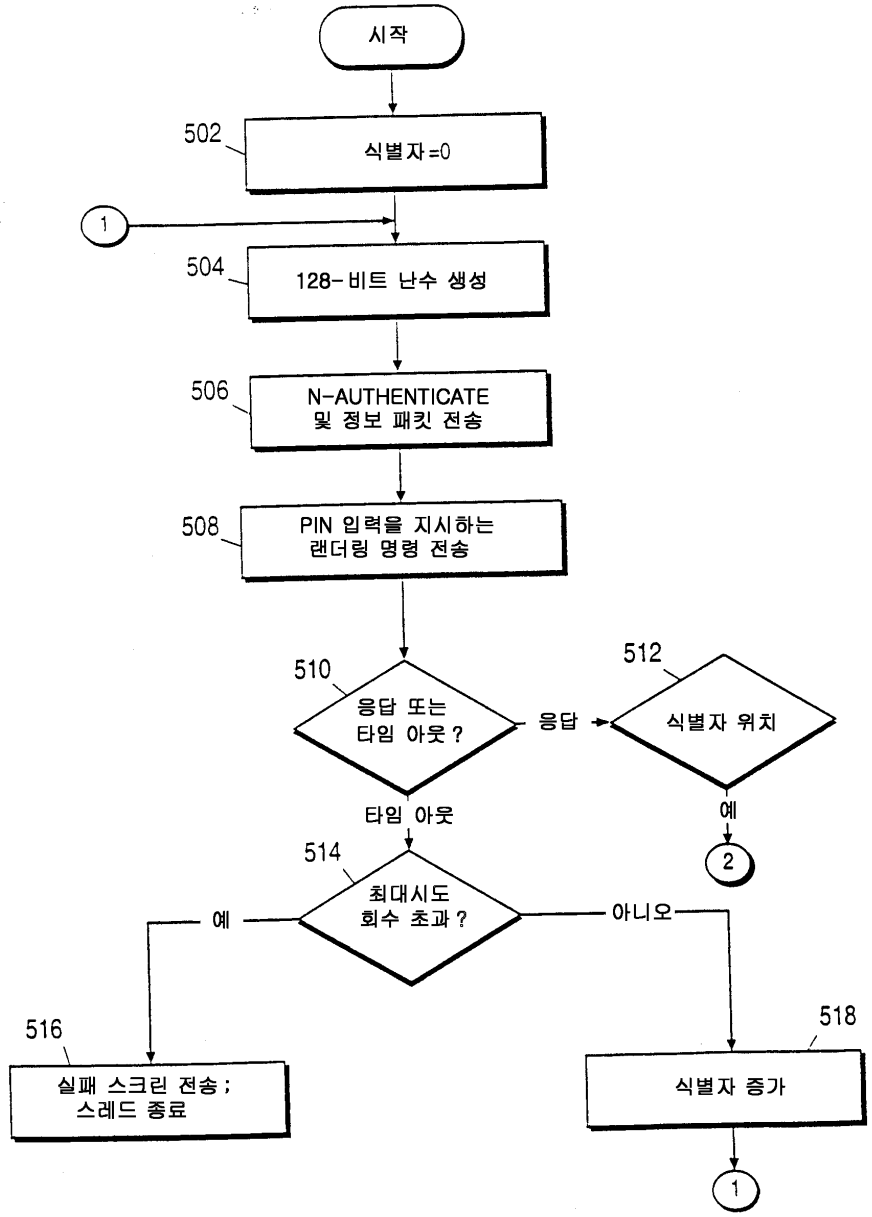
도면4B


4
416
시작 요청 전송
418
재시도 시간 =NOW+INTERVAL
420
WAITING_FOR_STARTUP = "Y"
5
422
메세지 수신 ?
예
6
7
아니오
424
재시도
시간 초과 ?
5
아니오
예
426
최대 시도
횟수 초과 ?
430
시작 요청 전송
아니오
예
432
최대 시도 횟수
증가 , 재시도 시간 리셋
428
에러 발생
8

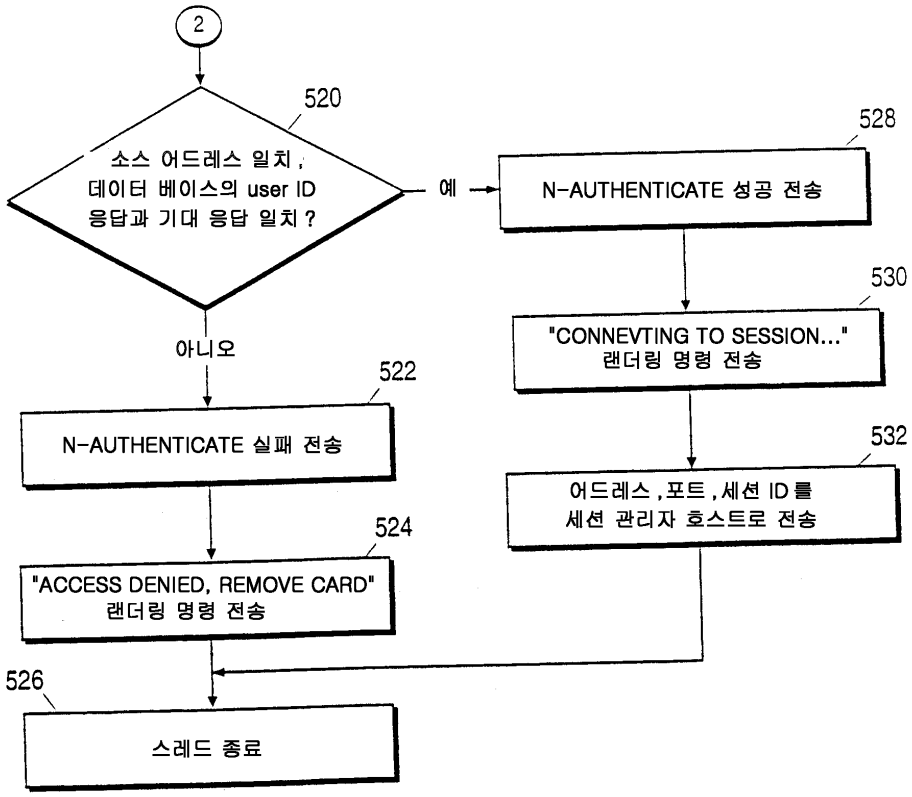
도면4C


6
434
시작 대기 ?
예
436
챌린지 메시지 ?
아니오
7
아니오
예
438
WAITING_FOR_STARTUP="N"
440
아니오
메시지 도착 ?
예
442
메시지 처리
8
444
카드 제거 ?
예
448
분리 메시지 전송 ,
수신 확인 대기
450
스크린 클리어
1
아니오
446
시작 대기 ?
예
1

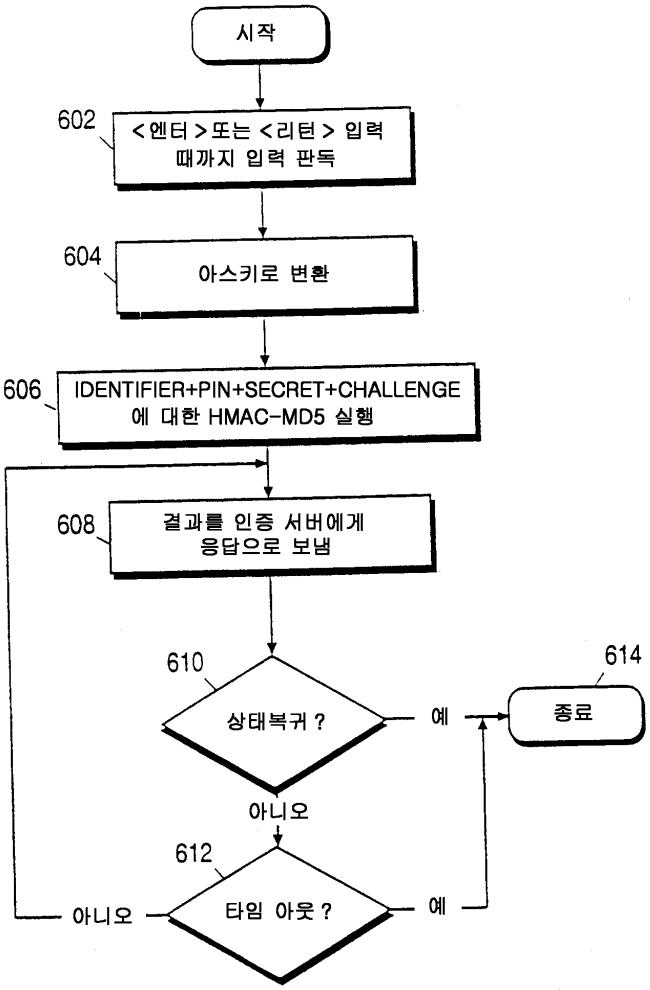
도면5A


시작
502
식별자=0
1
504
128-비트 난수 생성
506
N-AUTHENTICATE
및 정보 패킷 전송
508
PIN 입력을 지시하는
랜더링 명령 전송
510
응답 또는
타임 아웃?
응답
512
식별자 위치
예
2
타임 아웃
514
최대시도
회수 초과?
예
아니오
516
실패 스크린 전송;
스레드 종료
518
식별자 증가
1

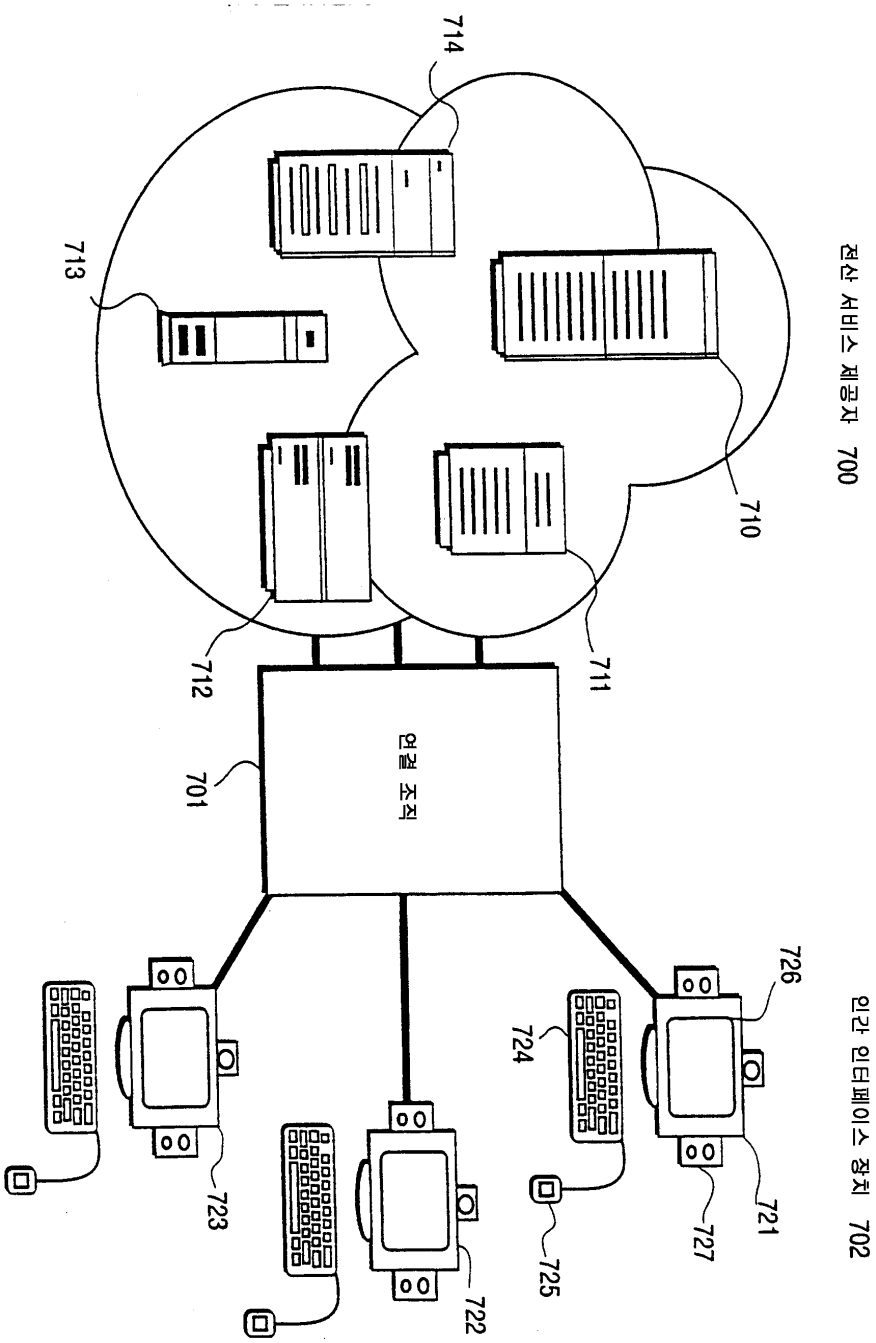
도면5B


2
520
소스 어드레스 일치 ,
데이터 베이스의 user ID
응답과 기대 응답 일치 ?
예
528
N-AUTHENTICATE 성공 전송
530
"CONNEVTING TO SESSION..."
랜더링 명령 전송
532
어드레스 , 포트 , 세션 ID 를
세션 관리자 호스트로 전송
아니오
522
N-AUTHENTICATE 실패 전송
524
"ACCESS DENIED, REMOVE CARD"
랜더링 명령 전송
526
스레드 종료

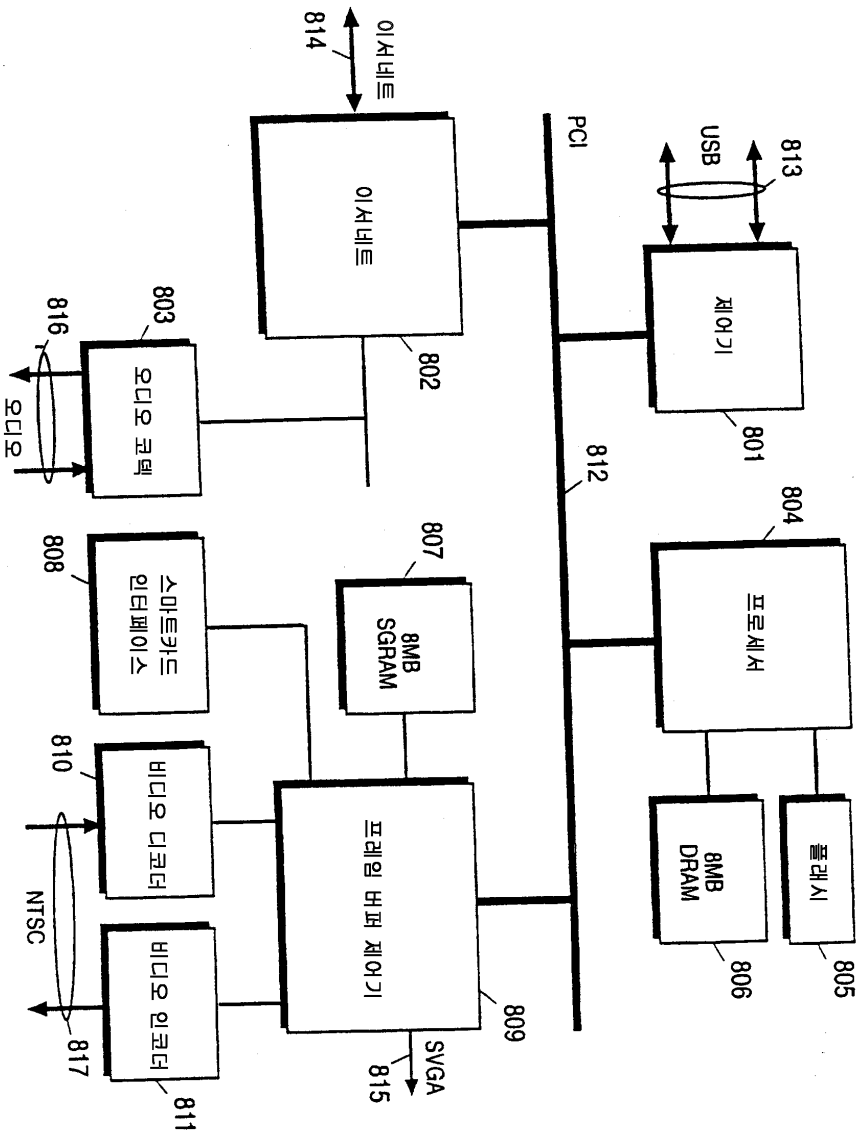
도면6


시작
602
<엔터>또는 <리턴> 입력 때까지 입력 판독
604
아스키로 변환
606
IDENTIFIER+PIN+SECRET+CHALLENGE 에 대한 HMAC-MD5 실행
608
결과를 인증 서버에게 응답으로 보냄
610
상태복귀?
예
614
종료
아니오
612
타임 아웃?
예
아니오

도면7


714
713
710
711
712
701
연결 조직
723
722
724
725
726
727
721
전산 서비스 제공자 700
인간 인터페이스 장치 702

도면8


이서네트
814
이서네트
802
PCI
USB
813
제어기
801
812
804
프로세서
803
816
오디오 코덱
오디오
807
8MB SGRAM
808
스마트카드 인터페이스
810
비디오 디코더
NTSC
817
비디오 인코더
811
프레임 버퍼 제어기
809
SVGA
815
8MB DRAM
806
플래시
805