

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁷
G06F 9/46

(11) 공개번호 특허2001-0041365
(43) 공개일자 2001년05월15일

(21) 출원번호	10-2000-7009482	(87) 국제공개번호	W0 1999/44115
(22) 출원일자	2000년08월25일	(87) 국제공개일자	1999년09월02일
번역문제출일자	2000년08월25일		
(86) 국제출원번호	PCT/US1999/03522		
(86) 국제출원출원일자	1999년02월18일		
(81) 지정국	AP ARIP0특허 : 케냐 레소토 말라위 수단 스와질랜드 우간다 가나 감비아 짐바브웨 EA 유라시아특허 : 아르메니아 아제르바이잔 벨라루스 키르기즈 카자흐 스탄 몰도바 러시아 타지키스탄 투르크메니스탄 EP 유럽특허 : 오스트리아 벨기에 스위스 독일 덴마크 스페인 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투 칼 스웨덴 핀란드 사이프러스 OA OAPI특허 : 부르키나파소 베냉 중앙아프리카 콩고 코트디부아르 카 메룬 가봉 기네 말리 모리타니 니제르 세네갈 차드 토고 기네비소 국내특허 : 인도네시아 유고슬라비아 알바니아 아르메니아 오스트리아 오스트레일리아 아제르바이잔 보스니아-헤르체고비나 바베이도스 불가 리아 브라질 벨라루스 캐나다 스위스 중국 쿠바 체코 독일 덴마크 에스토니아 스페인 핀란드 영국 그루지야 헝가리 이스라엘 아이슬란 드 일본 케냐 키르기즈 북한 대한민국 카자흐스탄 세인트루시아 스 리랑카 라이베리아 레소토 리투아니아 룩셈부르크 라트비아 몰도바 마다가스카르 마케도니아 몽고 말라위 멕시코 노르웨이 뉴질랜드 슬 로베니아 슬로바키아 타지키스탄 투르크메니스탄 터키 트리니다드토 바고 우크라이나 우간다 우즈베키스탄 베트남 폴란드 포르투갈 루마 니아 러시아 수단 스웨덴 싱가포르 그레나다 가나 크로아티아 짐바 브웨 인도 시에라리온 감비아		
(30) 우선권주장	60/076,048 1998년02월26일 미국(US)		
	09/044,837 1998년03월20일 미국(US)		
(71) 출원인	선 마이크로시스템즈, 인코포레이티드 케네쓰 올센		
	미합중국 캘리포니아 94303 팔로 알토 샌 안토니오 로드 901		
(72) 발명자	웨이플러로버트		
	미국, 메사추세츠02144, 소머빌, 노쓰스트리트96		
(74) 대리인	허상훈		

심사청구 : 없음

(54) 보안 조건의 방법당 지정

요약

하나의 시스템은 하나의 서버상에 호출된 다수의 방법에 대한 각각의 보안 조건을 개별적으로 지정한다. 상기 시스템은 하나의 보안 클래스 라이브러리 및 하나의 실행 장치를 포함한다. 상기 보안 클래스 라이브러리는 다수의 방법에 대응하는 다수의 보안 객체를 저장한다. 상기 보안 객체 각각은 상기 대응되는 방법을 위한 보안 조건을 저장한다. 상기 실행 장치는 상기 클라이언트의 보안 조건을 포함하는 방법 호출 요구를 클라이언트로부터 수신한다. 상기 요구로부터, 상기 실행 장치는 상기 요구된 방법에 대응하는 상기 보안 클래스 라이브러리에 있는 상기 보안 객체의 하나를 확인하며, 상기 클라이언트 보안 조건과 상기 보안 객체에 저장된 보안 조건의 비교에 바탕을 둔 상기 요구된 방법을 호출할지를 결정한다.

대표도

도4b

색인어

분산 처리 시스템, 컴퓨터 시스템 보안 측정, 독립적인 보안 조건

명세서

기술분야

본 발명은 컴퓨터 시스템에서 보안 측정에 관한 것이며, 특히 독립적인 보안 조건에 개별적인 원격 방법을 결합시키는 서버를 가능케 하는 시스템 및 방법에 관한 것이다.

배경기술

관련 출원

1998. 2. 26일자 가특허출원 제 60/076,048호 (명칭: 분산 컴퓨팅 시스템),

본 출원과 동일자 특허출원 제 09/044,923호 (명칭: 저장장치 임대 시스템 및 방법),

본 출원과 동일자 특허출원 제 09/044,838호 (명칭: 분산 시스템에서 위촉 증서를 임대하기 위한 방법과 장치 및 산물),

본 출원과 동일자 특허출원 제 09/044,834호 (명칭: 분산 시스템에서 그룹 회원증을 임대하기 위한 방법과 장치 및 물건),

본 출원과 동일자 특허출원 제 09/044,916호 (명칭: 고장 검출 임대 방법),

본 출원과 동일자 특허출원 제 09/044,933호 (명칭: 이벤트 기본 시스템에서 행동을 이송하기 위한 방법),

본 출원과 동일자 특허출원 제 09/044,919호 (명칭: 분산 시스템에서 이벤트 통지를 위한 원격 로딩 및 객체의 지연 복구),

본 출원과 동일자 특허출원 제 09/044,938호 (명칭: 원격 방법 호출을 위한 장치 및 방법),

본 출원과 동일자 특허출원 제 09/045,652호 (명칭: 원격 방법 식별을 위한 결정 해쉬용 시스템 및 방법),

본 출원과 동일자 특허출원 제 09/044,790호 (명칭: 분산 시스템에서 원격 객체의 상태를 결정하기 위한 방법 및 장치),

본 출원과 동일자 특허출원 제 09/044,930호 (명칭: 분산 시스템에서 원격 프로시저 호출과 연관된 처리를 수행하기 위한 다운로드 가능한 스마트 프록시),

본 출원과 동일자 특허출원 제 09/044,917호 (명칭: 원격 방법의 중단 및 계속),

본 출원과 동일자 특허출원 제 09/044,835호 (명칭: 데이터 베이스에서 다중 엔트리 및 다중 템플릿 정합을 위한 방법 및 시스템),

본 출원과 동일자 특허출원 제 09/044,839호 (명칭: 데이터 베이스에서 인-플레이스 변경을 위한 방법 및 시스템),

본 출원과 동일자 특허출원 제 09/044,945호 (명칭: 데이터 베이스에서 타입세이프 속성 매칭 방법 및 시스템),

본 출원과 동일자 특허출원 제 09/044,931호 (명칭: 분산 시스템에서 동적 탐색 서비스),

본 출원과 동일자 특허출원 제 09/044,939호 (명칭: 분산 시스템의 디바이스와 통신하는데 사용하기 위한 다운로드 가능한 코드를 제공하는 방법 및 장치),

본 출원과 동일자 특허출원 제 09/044,826호 (명칭: 탐색 서비스로 접근을 용이하게 하는 방법 및 시스템),

본 출원과 동일자 특허출원 제 09/044,932호 (명칭: 분산 시스템에서 동적으로 정보를 확인하기 위한 방법 및 장치),

1998. 2. 26일자 특허출원 제 09/030,840호 (명칭: 네트워크 상에서 동적 분산 컴퓨팅을 위한 방법 및 장치),

본 출원과 동일자 특허출원 제 09/044,936호 (명칭: 연속성 공유 메모리 공간 형성을 위한 대화형 설계틀),

본 출원과 동일자 특허출원 제 09/044,934호 (명칭: 다형성 토큰 기초 제어),

본 출원과 동일자 특허출원 제 09/044,915호 (명칭: 스택 기초 액세스 제어),

본 출원과 동일자 특허출원 제 09/044,944호 (명칭: 스택 기초 보안 조건).

발명의 배경

분산 시스템은 통신 네트워크에 의하여 상호 연결된 다수의 상이한 컴퓨터를 일반적으로 포함한다. 종종, 클라이언트-서버 관계는 상호 통신하는 컴퓨터 사이에서 성립된다. 통상, "클라이언트(client)"는 "서버(server)" 프로세스에 의해 위치되는 또는 제어되는 자원을 요구하기 위해 호출하는 프로세스로서 한정된다. 상술에서, 상기 요구하는 프로세스 및 서버 프로세스를 수행하는 컴퓨터는 각각 하나의 클라

이언트 및 서버로서 또한 언급된다. 그러나, 이러한 역할은 정보의 환경 및 특정한 프로세싱 해당 장소에 의존하는 것을 바꾼다.

클라이언트-서버 관계를 조성하는 하나의 메카니즘은 원격 절차 호출(RPC)인데 이곳에서 클라이언트가 서버의 기능을 호출한다. 상기 RPC는 동일한 또는 상이한 컴퓨터를 운용하는 두 개의 프로세스 사이의 통신을 동조시키는 메카니즘이다. RPC 메카니즘은 두 개의 부분으로 구현되는데 한 부분은 클라이언트 쪽 및 다른 한 부분은 서버 쪽이다.

보안은 클라이언트 컴퓨터와 서버 컴퓨터가 통신할 때 언제나 발생하는 이슈이다. 보안에 빈틈이 생기면 클라이언트와 서버 컴퓨터 모두의 운용에 심각한 장애가 일어날 수 있다. 따라서 컴퓨터 시스템을 사용하는 조직은 고의적으로 혹은 무심결에 컴퓨터 시스템에 기능장애를 일으키거나 조직의 기밀을 훔치는 사람들에게 공격을 받기 쉽다.

시스템 연산자는 통상 세 가지 보안 이슈로 어드레스한다: (1) 메시지를 가로채는 것 및 변경하는 것을 방지하는 것; (2) 서버에 접근하는 것을 통제하는 것; (3) 클라이언트에 의해 서버를 인증하는 것. 시스템 연산자는 통신 조건을 설정하는 방법을 제공하는 보안 클래스를 한정함으로서 객체지향의 프로그램 환경으로 이러한 이슈를 인습적으로 어드레스하고 있다. 하나의 객체지향 프로그램 환경은 1997년도, 삼즈.넷(Sams.net) 출판의 자바1.1 개발자 가이드 제 2 편집에서 설명되는 선 마이크로시스템즈사의 자바 상표(Sun MicrosystemsTM JavaTM) 객체지향 프로그램 환경이며, 이 결과로 레퍼런스에 의해 통합된다.

상기 보안 클래스는 다섯 가지 통신 조건을 포함한다: CONFIDENTIALITY, INTEGRITY, ANONYMITY, AUTHENTICATE_SERVER, 및 NO_DELEGATION. CONFIDENTIALITY는 메시지 내용이 비밀로 되는 것을 보증한다. 시스템 연산자는 적절한 해독 키를 가지는 부분만이 상기 메시지를 해독할 수 있는 것을 보증하기 위해 암호 기술을 사용한다. INTEGRITY는 메시지 내용(요구 및 응답 둘 모두)이 변경된 메시지를 처리하기 위해 변경 및 거절되어 졌을 때 검출한다. 시스템 연산자는 상기 클라이언트 및 서버 위치 둘 모두에서 체크섬(checksums) 또는 그와 같은 것의 사용을 통해 이러한 것을 달성한다.

ANONYMITY는 익명으로 남기를 바라는 클라이언트를 나타낸다. 다시 말하면, 상기 클라이언트는 상기 서버에 의해 인증되는 것을 원하지 않는다. AUTHENTICATE_SERVER는 원격 방법을 호출하기 전에 인증 서버가 필요한 클라이언트를 나타낸다. 이러한 통신 조건을 통해, 상기 클라이언트는 상기 해당 서버와 통신하는 것을 보증한다. NO_DELEGATION은 그것을 만드는 호출로서 상기 클라이언트의 신분을 대리하는 것이 허용되지 않는 서버를 언급한다. 다시 말하면, 상기 서버는 상기 클라이언트를 가장하는 다른 컴퓨터 시스템에 호출되는 것을 허가하지 않는다.

상기 클라이언트의 위치에서, 상기 보안 클래스는 각 통신 조건을 위해 단일 비트에 의해 표현된다. CONFIDENTIALITY, INTEGRITY, ANONYMITY, AUTHENTICATE_SERVER, 및 NO_DELEGATION에 대응하는 비트를 설정함으로서, 상기 클라이언트는 비밀성이 보증되는 것, 보존성이 보증되는 것, 상기 클라이언트가 익명으로 남는 것, 상기 서버가 인증되는 것, 그리고 위임성이 제공되지 않는 것을 각각 지정한다. 인습적으로, 하나의 클라이언트는 하나의 전체 정황 또는 하나의 호출마다에 기초를 둔 상기 보안 클래스 선택을 나타낸다. 다시 말하면, 상기 클라이언트는 상기 클라이언트에 의해 만들어진 모든 호출을 위해 사용되는 한 세트의 통신 조건을 명시하거나, 서버의 개별적인 호출을 위해 분리된 통신 조건을 명시한다.

상기 서버의 위치에서, 상기 보안 클래스는 각 통신 조건을 위해 이중 비트에 의해 표현된다. 상기 비트 중 하나(상기 "지원된 비트")는 상기 대응하는 통신 조건이 상기 서버에 의해 지원되는지를 나타낸다. 상기 다른 비트(상기 "요구된 비트")는 상기 대응하는 통신 조건이 상기 서버에 의해 요구되는지를 나타낸다. 다시 말하면, 상기 지원된 비트는 상기 서버가 할 수 있는 것을 지정하며, 상기 요구된 비트는 상기 서버가 해야할 것을 지정한다.

CONFIDENTIALITY에 대하여, 상기 지원된 비트가 설정되면, 상기 서버는 상기 클라이언트가 그것을 요구할 때 비밀성을 제공한다. 상기 비트가 지원되지 않으면 상기 서버는 비밀성을 지원할 수 없다. 상기 요구된 비트가 설정될 때, 상기 서버는 항상 비밀성을 보증한다.

INTEGRITY에 대하여, 상기 지원된 비트가 설정되면, 상기 서버는 상기 클라이언트가 그것을 요구할 때 보존성을 제공한다. 만일 상기 비트가 설정되지 않으면, 상기 서버는 보존성을 제공할 수 없다. 상기 요구된 비트가 설정될 때, 상기 서버는 항상 보존성을 보증한다.

ANONYMITY에 대하여, 상기 지원된 비트가 설정되면, 상기 서버는 상기 클라이언트가 익명성을 요구할 때 익명의 클라이언트(즉, 상기 서버는 클라이언트를 인증하지 않으려고 한다)를 수용하려고 한다. 반대로, 상기 지원된 비트가 설정되지 않으면, 상기 서버는 상기 클라이언트를 항상 인증한다. 상기 요구된 비트가 설정될 때, 상기 서버는 익명의 클라이언트를 다만 수용한다. 다시 말하면, 상기 요구된 비트를 설정될 때, 상기 서버는 상기 클라이언트를 결코 인증하지 않는다.

AUTHENTICATE_SERVER에 대하여, 상기 지원된 비트가 설정되면, 상기 서버는 상기 클라이언트가 그것을 요구할 때 그 자체를 인증하려고 한다. 반대로, 상기 지원된 비트가 설정되지 않으면, 상기 서버는 그 자체를 인증할 수 없다. 서버는 인증하기 위해 그 자체를 결코 요구하지 않는다. 그래서 상기 요구된 비트는 결코 설정되지 않는다.

NO_DELEGATION에 대하여, 상기 지원된 비트가 설정되면, 상기 서버는 위임없이 동작하려고 한다. 반대로, 상기 지원된 비트가 설정되지 않으면, 상기 서버는 위임을 항상 요구한다. 상기 요구된 비트가 설정될 때, 상기 서버는 위임할 필요가 결코 없다.

종래에, 서버는 상기 서버가 허용하는 최소한의 조건을 요구하는 단일 세트의 통신 조건을 포함한다. 상기 서버는 상기 서버에 호출된 모든 방법 및 기능을 처리하는 한 세트의 통신 조건을 지정한다. 통상, 그러나, 상기 방법 및 기능은 그들의 호출을 위해 허용할 수 있는 연속적으로 변하는 조건 세트를 포함한다. 예를 들면, 하나의 서버가 두 방법을 포함한다고 가정한다: 클라이언트 인증을 요구하는 삭제 방

법 및 클라이언트 인증을 요구하지 않는 검색 방법. 상기 서버 통신 조건은 상기 방법에 의해 요구되는 가장 엄격한 통신 조건을 포함하며, 이러한 경우에 클라이언트 인증을 항상 필요로 한다.

그러한 통신 조건의 일괄 지정은 방법을 호출하는데 필요한 처리 시간의 총합을 증가시키며, 모든 방법이 동일한 통신 조건에, 그러한 통신 조건을 요구하지 않는 것들조차도 종속되기 때문에 시스템 자원을 낭비한다. 따라서, 통신 시스템에서 보안 조건 지정을 개선하는 것이 바람직하다.

발명의 상세한 설명

발명의 요약

본 발명과 일치된 시스템 및 방법은 하나의 방법마다에 바탕을 둔 통신 조건을 지정하는 서버를 허용함으로써 종래 시스템의 개선을 나타낸다. 즉, 상기 서버는 클라이언트 또는 또 다른 서버에 대신하여 호출된 각 방법을 위해 하나의 분리된 세트의 통신 조건을 지정한다.

본 발명의 원리에 일치하는 시스템은 하나의 컴퓨터에 호출된 다수의 방법 각각을 위해 보안 조건을 개별적으로 지정한다. 상기 시스템은 하나의 보안 클래스 라이브러리 및 하나의 실행 장치를 포함한다. 상기 보안 클래스 라이브러리는 상기 다수의 방법에 대응하는 다수의 보안 객체를 저장한다. 상기 보안 객체 각각은 상기 대응하는 방법을 위한 상기 보안 조건을 저장한다. 상기 실행 장치는 하나의 클라이언트로부터 상기 클라이언트의 보안 조건을 포함하는 하나의 방법 호출 요구를 수신한다. 상기 요구로부터, 상기 실행 장치는 상기 보안 클래스 라이브러리에서 상기 요구된 방법에 대응하는 하나의 상기 보안 객체를 확인한다. 상기 실행 장치는 상기 클라이언트 보안 조건과 상기 보안 객체에 저장된 보안 조건의 비교에 바탕을 둔 상기 요구된 방법을 호출할 것인지를 결정한다.

도면의 간단한 설명

첨부 도면은 본 명세서의 일부로 결합되었으며, 본 발명의 목적, 장점 및 원리들에 대해 설명하기 위한 것이다.

도 1은 본 발명에 따른 시스템을 사용하기에 적절한 컴퓨터 네트워크를 나타낸 다이어그램;

도 2는 본 발명의 원리에 따른 구현으로서 도 1에 보인 서버를 나타낸 다이어그램;

도 3은 도 2에 보인 메모리를 나타낸 상세한 다이어그램;

도 4a는 도 3에 보인 보안 클래스 라이브러리를 나타낸 상세한 다이어그램;

도 4b는 도 4에 보인 상기 보안 클래스 라이브러리로부터 방법 보안 객체를 나타낸 더 상세한 다이어그램;

도 5a 및 도 5b는 본 발명의 원리에 따른 클라이언트 프로세싱 활동을 나타낸 플로우차트; 그리고

도 6은 본 발명의 원리에 따른 서버 프로세싱 활동을 나타낸 플로우차트.

실시예

개요

본 발명의 하기에 설명된 명세서는 첨부된 도면에 대해 언급한다. 상이한 도면에서 상기 동일한 참조 번호는 동일 또는 유사한 요소로 간주한다. 또한, 하기에 설명된 명세서는 본 발명을 제한하지 않는다. 그 대신에, 본 발명의 범위는 첨부된 청구항에 의해 한정된다.

본 발명의 원리에 따른 시스템 및 방법은 한 클라이언트 또는 다른 서버에 대신하여 호출되는 각각의 방법을 위해 상이한 한 세트의 통신 조건을 지정하는 서버를 허용한다. 이러한 방법당 지정은 요구된 방법을 호출하기 위해 요구된 처리 시간 및 상기 호출에 필요한 시스템 자원의 총합 둘 모두를 절약한다.

분산 시스템

도 1은 본 발명에 일치하는 시스템과 함께 사용되는 전형적인 컴퓨터 네트워크(1000)를 나타낸 다이어그램이다. 컴퓨터 네트워크(1000)는 네트워크를 경유하는 서버 컴퓨터(1300)에 연결된 클라이언트 컴퓨터(1100 및 1200)를 포함한다. 클라이언트 컴퓨터(1100 및 1200)는 예를 들어 네트워크 컴퓨터, 워크스테이션, 퍼스널, 컴퓨터, 메인프레임 컴퓨터 심지어는 덤 터미널("dumb" terminals)까지도 포함하는 종래의 클라이언트 컴퓨터이다.

클라이언트 컴퓨터(1100)는 LAN과 같은 로컬 네트워크(1120)에 쌍방향 데이터 통신 커플링을 제공해주는 통신 인터페이스를 포함한다. 숙련된 기술자는 무선 연결이 커플링을 위해서도 사용될 수 있다는 것을 인정할 것이다. 어떠한 장비에 있어서도 클라이언트 컴퓨터(1100)는 다양한 형태의 정보들을 나타내는 전기적 전자기적, 혹은 광학적 신호들을 송수신한다.

로컬 네트워크(1120)는 통상 다른 데이터 네트워크 또는 장비에 데이터 통신을 제공한다. 예를 들어, 로컬 네트워크(1120)는 호스트 컴퓨터(1140) 또는, 인터넷 서비스 제공자(ISP)(1420)에 의해 운용되는 데이터 장비를 접속시킬 수 있다. ISP(1420)은 차례로 인터넷(1140)을 통해 데이터 통신 서비스를 제공할 수 있다. 로컬 네트워크(1120)와 인터넷(1440)은 모두 데이터송신을 위해서 전기적 전자기적 또는, 광학적 신호를 사용한다. 다양한 네트워크를 통과한 신호와 클라이언트 컴퓨터(1100)에 데이터를 송수신하는 신호는 정보를 운송하는 전형적인 형태의 캐리어 신호 이다.

클라이언트 컴퓨터(1100)는 로컬 네트워크를 경유하는 인터넷(1440)에 연결할 필요는 없지만 ISP(1420)

에 직접 연결될 필요가 있을지도 모른다. 클라이언트 컴퓨터(1200)는 그러한 접속만을 필요로 한다. 이러한 접속은 모뎀이나 이더넷(Ethernet) 접속 같은 데이터 접속을 사용함으로써 수행될 수 있다.

클라이언트 컴퓨터(1100 및 1200)는 네트워크를 통해서 프로그램 코드를 포함하는 메시지를 보내고 데이터를 받을 수 있다. 인터넷(1440) 또는 로컬 네트워크(1120)상에서 통신할 때 클라이언트 컴퓨터(1100 및 1200)는 서버 컴퓨터(1300)와 같은 서버 컴퓨터로부터 그러한 코드를 요구하거나 서버 컴퓨터의 방법과 기능을 호출할 수도 있다.

도 2는 본 발명의 원리에 일치하는 구현에 사용되는 서버 컴퓨터(1300)를 나타낸 다이어그램이다. 클라이언트 컴퓨터(1100 및 1200)는 유사하게 구성될 수 있다.

서버 컴퓨터(1300)는 정보교환을 위해서 버스(2100) 또는 다른 통신 메카니즘을 포함한다. 그리고 프로세서(2200)는 정보처리를 위해서 버스(2100)에 연결된다. 프로세서(2200)는 운영자가 정보를 입력할 수 있도록 하기 위하여 버스(2100)를 경유하여 키보드, 마우스, 음성인식 메카니즘, 또는 트랙볼 같은 입력 디바이스(2300)에 연결될 수도 있다. 그리고, 운영자에게 정보를 보여주기 위하여 CRT같은 출력 디바이스(2400)에도 연결될 수 있다.

서버 컴퓨터(1300)는 또한 RAM이나 다른 형태의 재기록 가능한 저장 디바이스와 같은 메인 메모리(2500)를 포함하는데 메인 메모리(2500)는 정보 및 프로세서(2200)에 의해 실행된 명령을 저장하기 위하여 버스(2100)에 연결된다. 또한, 메인 메모리(2500)는 프로세서(2200)가 명령을 실행할 동안 임시 변수나 다른 중간 정보를 저장하기 위하여 사용될 수도 있다. 더 나아가 서버 컴퓨터(1300)는 ROM(2600)이나 다른 형태의 정적 저장 디바이스를 포함하는데, 이들은 정적 정보 및 프로세서(2200)를 위한 명령을 저장하기 위해 버스(2100)에 연결된다. 저장 디바이스(2700)는 정보 또는 명령을 저장하기 위해 제공되는데 버스(2100)에 연결된다.

여기에 사용된 용어 "컴퓨터-읽기 가능한 매체"는 실행을 위한 프로세서(2200)에 명령을 제공하는데 관여하는 어떠한 매체를 언급한다. 그러한 매체는 제한되지는 않지만 비휘발성 매체, 휘발성 매체 및 전송 매체를 포함하는 많은 형태를 가진다. 비휘발성 매체는 저장 디바이스(2700)와 같은 예를 들어 광학 또는 자기 디스크를 포함한다. 휘발성 매체는 메인 메모리(2500)와 같은 반복 쓰기가 가능한 메모리를 포함한다. 전송 매체는 동축 케이블, 동선 및 버스(2100)를 포함하는 선을 포함하는 광섬유를 포함한다. 전송 매체는 전파 및 적외선 데이터 통신 동안 생성되는 것과 같은 음향 또는 광 파동의 형태를 가진다.

컴퓨터-읽기 가능한 매체의 공통 형태는 예를 들어 플로피 디스크, 유동성 디스크, 하드 디스크, 자기 테이프 또는 다른 자기 매체, CD ROM, 다른 광학 매체, 천공카드, 종이테이프, 홀의 형태를 가지는 다른 물리적 매체, RAM, PROM, EPROM, FLASH EPROM, 다른 메모리 칩 또는 카트리지, 캐리어 웨이브 또는 컴퓨터가 읽을 수 있는 다른 매체를 포함한다.

다양한 형태의 컴퓨터 읽기 가능한 매체는 실행을 위한 프로세서(2200)에 하나 또는 그 이상의 명령의 하나 또는 그 이상의 순서를 실행하는데 관련된다. 예를 들면, 상기 명령은 원격 컴퓨터의 자기 디스크에 최초로 저장된다. 상기 원격 컴퓨터는 상기 명령을 반복 쓰기 가능한 메모리로 로드할 수 있으며 상기 명령을 모뎀을 사용하여 전화선을 통해 보낼 수 있다.

서버 컴퓨터(1300)에 위치한 모뎀은 전화선상의 데이터를 수신할 수 있으며 상기 데이터를 적외선 신호로 변환하는 적외선 송신기를 사용할 수 있다. 버스(2100)에 연결된 적외선 검출기는 상기 적외선 신호로 전송된 데이터를 수신할 수 있으며 상기 데이터를 버스(2100)상에 놓을 수 있다. 프로세서(2200)가 상기 명령을 회복하여 실행하는 것으로부터, 버스(2100)는 상기 데이터를 메인 메모리(2500)로 전송한다. 메인 메모리(2500)에 의해 수신된 명령은 프로세서(2200)에 의해 수행 전 또는 후 중 하나에서 저장 디바이스(2700)에 임의로 저장된다.

서버 컴퓨터(1300)는 버스(2100)에 연결된 통신 인터페이스(2800)를 포함한다. 통신 인터페이스(2800)는 인터넷 링크(2820)를 통해 인터넷(1440)에 연결하는 양방향 데이터 통신을 제공한다. 통신 인터페이스(2800)는 로컬 네트워크(1120)에 연결하는 양방향 데이터 통신을 선택적으로 제공한다. 어떤 구현에서는, 통신 인터페이스(2800)는 다양한 종류의 정보를 나타내는 데이터를 운반하는 전기적, 전자기적 또는, 광학적 신호를 전송 및 수신한다.

도 3은 메인 메모리(2500)를 포함하는 메모리(3000)를 상세하게 나타낸 다이어그램이다. 메모리(3000)는 자바 상표의 런타임 시스템(3100), 애플리케이션 영역(3200) 및 데이터 메모리(3300)를 포함한다. JVM(3120)은 동작 시스템 및 컴퓨터 하드웨어에 의해 제공된 성능을 사용하여 동작하는 소프트웨어로 구현된 마이크로프로세서이다. 상기 JVM은, 예를 들어 팀 린드홀름(Tim Lindholm) 및 프랭크 옐린(Frank Yellin)이 저술한 에디슨 웨슬리(Addison Wesley)사 1996년 출판, 자바 가상 머신 스펙에서 설명되어 있으며, 본 명세서에 참조로써 통합된다.

JVM(3120)은 자바 상표의 원격 방법 요청(RMI)(3140)을 포함한다. RMI(3140)는 애플리케이션 영역(3200)에 저장된 방법의 요청에 대한 호출에 응하는 RPC 메카니즘이다. RMI(3140) 및 JVM(3120) 둘 모두는 미국 캘리포니아 마운틴 뷰에 소재한 선 마이크로시스템스사로부터 구입할 수 있는 자바 상표의 소프트웨어 개발 키트의 일부분으로써 제공된다.

애플리케이션 영역(3200)은 프로세서(220)에 의해 실행 및 JVM(3120)에 의해 해석되는 프로그램을 포함한다. 상기 프로그램은 자바 프로그래밍 언어의 애플릿과 같은 객체지향 애플릿을 포함하며 또한 다른 객체지향 프로그램, 멀티플랫폼 프로그램, 보안 프로그램, 분산 네트워킹 프로그램, 멀티스레드된 프로그램, 웹 프로그램 등등을 포함한다. 상기 프로그램 중 일부는 클라이언트의 요구로써 RMI(3140)에 의해 요청된 방법을 포함한다. 그리고, 상기 방법의 일부는 데이터 메모리(3300)에 저장된 객체 및 클래스상에서 동작한다. 상기 방법은 예를 들어 저장, 삭제, 업데이트, 검색 등등을 포함한다.

데이터 메모리(3300)는 프로세서(2200) 및 JVM(3120)에 의해 사용된 데이터를 저장하며, 보안 클래스 라

이브러리(3320)를 포함한다. 도 4a는 보안 클래스 라이브러리(3320)를 더 상세하게 나타낸 다이어그램이다. 보안 클래스 라이브러리(3320)는 확인된 방법에 의해 요구된 통신 조건을 포함하는 다중 방법 보안 객체(4100-4400)를 저장한다.

도 4b는 본 발명에 일치하는 구현으로써의 방법 보안 객체(4100)의 내용을 나타낸 다이어그램이다. 방법 보안 객체(4100)는 방법 식별자(4110) 및 통신 조건 플래그를 포함하며, 비밀성 비트(4120), 보존성 비트(4130), 익명성 비트(4140), 서버 인증 비트(4150) 및 위임성 비트(4160)를 포함한다.

방법 식별자(4110)는 애플리케이션 영역(3200)에서 상기 방법의 포인터 또는 참조로 인용된 미국 특허출원 제 09/045,652호(명칭: 원격 확인 방법의 결론적인 해시를 위한 방법 및 시스템)에 설명된 결론적인 해싱 메커니즘과 같은 대응하는 방법에 대한 증명을 포함한다.

비밀성 비트(4120)는 상기 대응하는 방법이 기밀인 관련 통신을 허용 또는 요구하는지를 규정하는 지원된 비트(4122) 및 요구된 비트(4124)를 포함한다. 클라이언트가 상기 방법의 요청을 요구하며 지원된 비트(4122)가 설정되면, 서버 컴퓨터(1300)는 상기 클라이언트가 그것을 요구할 때 비밀성을 제공한다. 만일, 지원된 비트(4122)가 설정되지 않으면, 서버 컴퓨터(1300)는 비밀성을 제공할 수 없다. 이에 반해서, 만일 요구된 비트(4124)가 설정되면, 서버 컴퓨터(1300)는 항상 비밀성을 보증한다.

보존성 비트(4130)는 상기 방법이 관련 통신에서 보존성 확인을 허용 또는 요구하는지를 규정하는 지원된 비트(4132) 및 요구된 비트(4134)를 포함한다. 클라이언트가 상기 방법의 요청을 요구하며 지원된 비트(4132)가 설정되면, 서버 컴퓨터(1300)는 상기 클라이언트가 그것을 요구할 때 보존성을 제공한다. 만일, 지원된 비트(4132)가 설정되지 않으면, 서버 컴퓨터(1300)는 보존성을 제공할 수 없다. 이에 반해서, 만일 요구된 비트(4134)가 설정되면, 서버 컴퓨터(1300)는 항상 보존성을 보증한다.

익명성 비트(4140)는 상기 방법이 클라이언트의 익명을 요구 또는 지원하는지를 규정하는 지원된 비트(4142) 및 요구된 비트(4144)를 포함한다. 클라이언트가 상기 방법의 요청을 요구하며 지원된 비트(4142)가 설정되면, 서버 컴퓨터(1300)는 상기 클라이언트가 그것을 요구할 때 익명성을 허용한다. 만일, 지원된 비트(4142)가 설정되지 않으면, 서버 컴퓨터(1300)는 클라이언트의 인증을 요구한다. 요구된 비트(4144)가 설정되면, 서버 컴퓨터(1300)는 익명의 클라이언트를 다만 수락한다. 다시 말하면, 요구된 비트(4144)가 설정되면, 서버 컴퓨터(1300)는 상기 클라이언트를 결코 인증하지 않는다.

서버 인증 비트(4150)는 상기 방법이 클라이언트에 의해 요구될 때 서버 컴퓨터(1300)의 인증을 지원하는지를 규정하는 지원된 비트(4152) 및 요구된 비트(4154)를 포함한다. 상기 클라이언트가 상기 방법의 요청을 요구하며 지원된 비트(4152)가 설정되면, 서버 컴퓨터(1300)는 상기 클라이언트가 그것을 요구할 때 자신을 인증한다. 만일, 지원된 비트(4152)가 설정되지 않으면, 서버 컴퓨터(1300)는 자신을 인증할 수 없다. 방법은 자신을 인증하는 서버 컴퓨터(1300)를 결코 요구하지 않는다. 그러므로, 요구된 비트(4154)는 결코 설정되지 않는다.

위임성 비트(4160)는 서버 컴퓨터(1300)가 다른 시스템을 호출할 때 서버 컴퓨터(1300)가 상기 클라이언트인 체할 필요가 있는지를 규정하는 지원된 비트(4162) 및 요구된 비트(4164)를 포함한다. 상기 클라이언트가 상기 방법의 요청을 요구하며 지원된 비트(4162)가 설정되면, 만일 상기 클라이언트가 그것을 허용하지 않는다면, 서버 컴퓨터(1300)는 위임성 없이 동작한다. 만일, 지원된 비트(4162)가 설정되지 않으면, 서버 컴퓨터(1300)는 위임성을 요구한다. 이에 반해서, 만일 요구된 비트(4164)가 설정되면, 서버 컴퓨터(1300)는 위임성을 결코 필요로 하지 않는다.

시스템 프로세싱

도 5a, 도 5b 및 도 6d는 클라이언트가 본 발명의 원리에 일치하는 구현으로서 서버의 기능에 액세스를 요구할 때 일어나는 프로세싱을 나타낸 플로우 차트이다. 도 5a 및 도 5b는 상기 클라이언트에 의해 수행되는 프로세싱을 나타낸다. 도 6은 상기 서버에 의해 수행되는 프로세싱을 나타낸다.

클라이언트 컴퓨터(1100)(도 1)상에서 동작하는 프로세스와 같은 클라이언트 프로세스는 서버 컴퓨터(1300)와 같은 서버상의 기능에 액세스를 요구한다. 상기 클라이언트 프로세스는 클라이언트 컴퓨터(1100)상에 RMI를 요청한다. 기능에 요구된 액세스는 서버(1300)상의 방법 및 몇 가지 다른 유사 동작을 위한 요구를 요청하는 호출을 포함한다.

상기 클라이언트 프로세스로부터, 상기 클라이언트 RMI는 요청되어진 서버(1300)상의 방법을 확인한다[단계(5110)]. 상기 클라이언트 RMI는 클라이언트 컴퓨터(1100)가 보존하는 서버(1300)의 보안 클래스 라이브러리(3320)의 사본으로부터 통신 조건을 수신함으로써 요구된 방법을 위해 서버(1300)의 통신 조건을 결정한다[단계(5120)].

상기 클라이언트 RMI는 서버(1300)의 보안 클래스 라이브러리의 사본으로부터 요구된 방법의 통신 조건에 상기 클라이언트 프로세스의 통신 조건을 결합시킨다[단계(5130)]. 상기 결합은 상기 클라이언트 프로세스의 통신 조건 및 상기 요구된 방법의 통신 조건 둘 모두를 만족하는 한 세트의 통신 조건으로 끝난다.

상기 클라이언트 RMI는 상기 요구된 방법이 상기 클라이언트 프로세스의 통신 조건을 지원하는지를 결정한다[단계(5140)]. 만일, 상기 요구된 방법이 상기 클라이언트 프로세스의 통신 조건을 지원하지 않는다면, 상기 클라이언트 RMI는 이러한 사실을 상기 클라이언트 프로세스에 통보한다[단계(5150)]. 상기 클라이언트 RMI는 예외를 배제함으로써 이를 행한다.

만일, 상기 요구된 방법이 상기 클라이언트 프로세스의 통신 조건을 지원한다면, 상기 클라이언트 RMI는 상기 결합된 통신 조건을 만족하는 통신 프로토콜을 선택한다[단계(5210)](도 5b). 하나의 서버의 클라이언트에 의해 만들어진 모든 원격 방법 요청은 근원적인 통신 프로토콜의 일부를 사용함으로써 수행된다. 상기 프로토콜의 상세 설명은 네트워크의 종류 및 바람직한 보안 종류에 따라서 매우 폭 넓다. 일반적으로, 상기 프로토콜은 클라이언트와 서버 사이의 실제 연결을 제공한다. 상기 연결은 클라이언트와

서버 사이의 전2중식 데이터 통신을 제공한다.

통상, 상기 클라이언트 및 상기 서버에 유용한 몇 가지 통신 프로토콜이 있다. 상기 클라이언트 RMI는 결합된 통신 조건을 만족시키는 유용한 통신 프로토콜 중에서 하나의 통신 프로토콜을 선택한다. 다시 말하면, 하나의 통신 프로토콜은 상기 결합된 통신 조건을 만족하는 또는 초과하는 것으로 선택된다. 만일, 상기 결합된 통신 조건을 만족하는 프로토콜이 없다면, 상기 클라이언트 RMI는 이러한 사실을 상기 클라이언트 프로세스에 알린다. 상기 클라이언트 RMI는 예외를 배제함으로써 이것을 행한다.

만일 통신 프로토콜이 발견되면, 상기 클라이언트 RMI는 상기 요구를 위한 선택된 프로토콜을 사용하기 위해 종래의 메커니즘을 사용하는 상기 서버와 교섭한다. 그리고, 상기 클라이언트 RMI는 상기 프로토콜의 사용을 시작한다[단계(5220)]. 상기 통신 프로토콜이 지원하는 통신 조건에 의존함으로써, 프로토콜의 시작은 암호화를 위한(비밀성을 위한) 준비로서 암호 키를 계산하는 정보 또는 체크섬(보존성을 위한)을 생성하는 정보와 같은 정보를 교환하는 것과 관련된다; 그리고 프로토콜의 시작은 서버에서 클라이언트 및 또는 클라이언트에서 서버를 인증하는 다중 도전-응답 통신에 관련된다. 상기 클라이언트 및 서버는 일정 시간 주기동안 실제 연결을 열린 상태를 유지하는데 또한 동의한다. 그래서 실제 연결은 클라이언트로부터 서버로 순차적인 통신을 위해 재 사용되며, 서버상에 동일 또는 상이한 방법을 위한 요구를 포함한다.

일단 상기 통신 프로토콜은 초기화되어 있으면, 상기 클라이언트 RMI는 서버에 상기 요구를 전송한다[단계(5230)]. 서버(1300)가 상기 클라이언트 RMI로부터 상기 요구를 수신할 때, 서버(1300)의 RMI(즉, RMI(3140)(도 3))는 상기 요구로부터 상기 요구된 방법 및 상기 클라이언트의 통신 조건을 확인한다[단계(6110)](도 6). RMI(3140)는 상기 요구된 방법을 보안 클래스 라이브러리(3320)에 위치시킨다. 그리고, RMI(3140)는 상기 대응하는 방법 보안 객체로부터 상기 방법에 관련된 통신 조건을 확인한다. RMI(3140)는 상기 요구된 방법이 상기 클라이언트의 통신 조건을 지원하는지를 결정하는 상기 방법 보안 객체에 저장된 통신 조건과 상기 클라이언트의 통신 조건을 비교한다[단계(6130)].

상기 요구된 방법이 상기 클라이언트의 통신 조건을 지원하면, RMI(3140)는 상기 요구를 위한 사용으로써 상기 통신 프로토콜이 상기 클라이언트의 및 상기 요구된 방법의 통신 조건 둘 모두를 지원하는지를 결정한다[단계(6140)]. 만일 상기 프로토콜이 통신 조건의 세트 둘 모두를 지원한다면, JVM(3120)는 상기 클라이언트 프로세스 상기 요구된 방법을 호출한다(즉, 상기 기능에 액세스를 승인한다)[단계(6150)].

만일 상기 요구된 방법이 상기 클라이언트의 통신 조건을 지원하지 않거나 상기 통신 프로토콜이 통신 조건의 세트 둘 모두를 지원하지 않는다면, RMI(3140)는 상기 클라이언트에 상기 요구가 거절된다는 것을 통보한다[단계(6160)]. 이러한 경우에, RMI(3140)는 비인가된 액세스 시도의 발생을 기록한다.

예를 들어, 클라이언트 컴퓨터(1100)상에 실행하는 클라이언트 프로세스가 서버 컴퓨터(1300)내의 데이터 메모리(3300)에 저장된 객체의 업데이트를 위한 익명성을 요구한다고 가정한다. 만일 익명성 비트(4140)의 지원된 비트(4142)가 상기 업데이트 방법을 위해 설정되지 않으면, 상기 방법은 데이터 메모리(3300)에서 상기 객체의 업데이트를 허용하기 전에 상기 클라이언트 프로세스의 인증을 요구한다. 그러므로, 비록 상기 통신 프로토콜이 익명성을 지원할지라도, 상기 클라이언트의 통신 조건이 상기 클라이언트 인증 조건을 만족하지 않기 때문에 RMI(3140)는 상기 클라이언트의 요구를 거절한다.

결론

본 발명의 원리에 일치하는 시스템 및 방법은 클라이언트로부터 요구의 제공에서 보안을 강화하는 방법 당으로 보안 조건을 지정하는 서버를 허용한다.

본 발명의 바람직한 실시예의 상기 설명은 도시 및 설명을 제공하지만, 드러낸 정확한 형태로 본 발명을 보존히 규명 또는 제한하는 것은 아니다. 변조 및 다양성은 상기의 설명에 따라 가능하거나 본 발명의 연습으로부터 습득된다. 본 발명의 범위는 청구항 및 그것의 동치에 의해 한정된다.

예를 들면, 도 1의 전형적인 컴퓨터 네트워크는 설명적인 목적으로만 존재한다. 본 명세서에 포함된 설명 및 개념은 참고로 본 출원에 포함되어진 서류중인 미국 특허출원 제 09/044,944호 (명칭: 스택 기초 보안 조건)에서 설명되는 전형적인 분산 시스템과 같은, 도 1과 비슷하지 않은 구조를 가지는 다른 컴퓨터 네트워크에 똑같이 적용된다.

산업상이용가능성

비록, 본 발명에 일치하는 시스템 및 방법은 자바 프로그래밍 환경에서 동작하는 것으로 설명되지만, 당업자는 본 발명이 다른 시스템 및 다른 프로그래밍 환경에서 실시될 수 있다는 것을 인정할 것이다. 부가적으로, 비록 본 발명의 양태가 메모리에 저장된 것으로 설명되지만, 당업자는 이러한 양태가 하드 디스크와 같은 제 2 저장 장치, 플로피 디스크 또는 CD-ROM; 인터넷으로부터 캐리어 웨이브; 또는 RAM 또는 ROM의 다른 형태와 같은 컴퓨터-읽기 가능한 매체의 다른 종류에 저장되거나 그것으로부터 읽을 수 있다는 것을 인정할 것이다. 선, 선 마이크로시스템즈, 선 로고, 자바 및 자바 기초의 상표는 미합중국 및 다른 국가에서 선 마이크로시스템즈 사의 상표 또는 등록 상표이다.

(57) 청구의 범위

청구항 1

하나의 컴퓨터상에 호출된 다수의 방법 각각을 위한 보안 조건을 개별적으로 지정하는 시스템에 있어서, 상기 다수의 방법에 대응하는, 각각의 상기 보안 객체가 상기 대응하는 방법을 위한 보안 조건을 저장하는, 다수의 보안 객체를 저장하는 보안 클래스 라이브러리와; 그리고

클라이언트의 보안 조건을 가지는 방법 요청 요구를 상기 클라이언트로부터 수신하는, 상기 요구된 방법에 대응하는 상기 보안 클래스 라이브러리에서 상기 보안 객체 중 하나를 확인하는, 그리고 상기 클라이언트 보안 조건과 상기 하나의 보안 객체에 저장된 보안 조건의 비교에 기초하여 요구된 방법을 호출할 것인지를 결정하는 실행 장치를 포함하는 시스템.

청구항 2

제 1 항에 있어서, 상기 보안 조건은 상기 컴퓨터 및 상기 클라이언트가 통신하는 동안 비밀을 보증함으로써 비밀성 조건을 포함하는 것을 특징으로 하는 시스템.

청구항 3

제 1 항에 있어서, 상기 보안 조건은 상기 컴퓨터 및 상기 클라이언트가 전송하는 동안 통신이 변경되지 않고 보존됨을 보증함으로써 보존성 조건을 포함하는 것을 특징으로 하는 시스템.

청구항 4

제 1 항에 있어서, 상기 보안 조건은 상기 클라이언트가 상기 컴퓨터에 익명으로 남음으로써 익명성 조건을 포함하는 것을 특징으로 하는 시스템.

청구항 5

제 1 항에 있어서, 상기 보안 조건은 상기 클라이언트가 스스로를 인증하는 컴퓨터를 요구함으로써 서버 인증 조건을 포함하는 것을 특징으로 하는 시스템.

청구항 6

제 1 항에 있어서, 상기 보안 조건은 상기 클라이언트가 상기 클라이언트의 보증하에 위임하는 상기 컴퓨터를 인가함으로써 위임성 조건을 포함하는 것을 특징으로 하는 시스템.

청구항 7

제 1 항에 있어서, 상기 보안 조건 각각은 상기 보안 조건이 상기 컴퓨터상의 대응하는 방법으로 지원되는지 또는 요구되는지를 규정하는 비트를 포함하는 것을 특징으로 하는 시스템.

청구항 8

제 1 항에 있어서, 상기 실행 장치는;

상기 클라이언트 보안 조건과 상기 하나의 보안 객체에 저장된 상기 보안 조건을 비교하는 비교 요소를 포함하는 것을 특징으로 하는 시스템.

청구항 9

제 8 항에 있어서, 상기 실행 장치는;

상기 클라이언트 보안 조건이 하나의 보안 객체에 저장된 상기 보안 조건을 만족할 때 상기 요구된 방법을 호출하는 호출 요소를 더 포함하는 것을 특징으로 하는 시스템.

청구항 10

제 8 항에 있어서, 상기 실행 장치는;

상기 클라이언트 보안 조건이 하나의 보안 객체에 저장된 상기 보안 조건을 만족하는데 실패할 때 인가된 요구를 보고하는 보고 요소를 더 포함하는 것을 특징으로 하는 시스템.

청구항 11

하나의 컴퓨터상에 호출된 다수의 방법 각각을 위한 보안 조건을 개별적으로 지정하는 시스템에 있어서, 상기 다수의 방법에 대응하는, 각각의 상기 보안 객체가 상기 대응하는 방법을 위한 보안 조건을 저장하는, 다수의 보안 객체를 저장하는 수단과;

클라이언트의 보안 조건을 가지는 방법 요청 요구를 상기 클라이언트로부터 수신하는 수단과;

상기 요구된 방법에 대응하는 수단을 저장하고 있는 상기 보안 객체 중 하나를 확인하는 수단과; 그리고

상기 클라이언트 보안 조건과 상기 하나의 보안 객체에 저장된 보안 조건의 비교에 기초하여 요구된 방법을 호출할 것인지를 결정하는 수단을 포함하는 시스템.

청구항 12

하나의 기능과 관련된 보안 조건에 바탕을 둔 컴퓨터상의 상기 기능에 액세스를 승인하는 컴퓨터-구현된 방법에 있어서,

상기 컴퓨터상의 메모리에 다수의 기능을 저장하는 단계와;

상기 기능 각각에 대한 보안 조건을 개별적으로 지정하는 단계와;

하나의 클라이언트의 보안 조건을 가지는 기능 액세스 요구를 상기 클라이언트로부터 수신하는 단계와;

상기 요구된 기능에 대한 지정된 보안 조건을 결정하는 단계와;

상기 지정된 보안 조건에 상기 클라이언트 보안 조건을 비교하는 단계와; 그리고

상기 클라이언트 보안 조건이 상기 지정된 보안 조건을 만족하면 상기 요구된 기능에 액세스를 승인하는 단계를 포함하는 컴퓨터-구현된 방법.

청구항 13

각각의 기능에 대해 개별적으로 지정되는 상기 기능과 관련된 보안 조건에 바탕을 둔 컴퓨터상의 상기 기능에 액세스를 승인하는 방법을 수행하는 컴퓨터를 제어하는 명령을 포함하는 컴퓨터-읽기 가능한 매체에 있어서,

하나의 클라이언트의 보안 조건을 가지는 기능 액세스 요구를 상기 클라이언트로부터 수신하는 단계와;

상기 요구된 기능에 대한 지정된 보안 조건을 결정하는 단계와;

상기 결정된 보안 조건에 상기 클라이언트 보안 조건을 비교하는 단계와; 그리고

상기 클라이언트 보안 조건이 상기 지정된 보안 조건을 만족하면 상기 요구된 기능에 액세스를 승인하는 단계를 포함하는 방법.

청구항 14

여러 방법을 가지는 하나의 프로그램,

상기 프로그램의 방법 각각에 대한 보안 조건을 저장하는 보안 클래스 영역, 그리고

하나의 원격 컴퓨터의 보안 조건을 가지는 방법 요청 요구를 상기 원격 컴퓨터로부터 수신하며, 상기 원격 컴퓨터가 상기 보안 클래스 영역에서 요구된 방법에 대한 보안 조건을 만족하면 상기 요구된 방법을 호출하는 가상 머신을 포함하는 하나의 메모리와; 그리고

상기 가상 머신 및 상기 프로그램을 동작시키는 하나의 프로세서를 포함하는 데이터 처리 시스템.

청구항 15

하나의 보안 클래스 라이브러리 데이터 구조를 포함하는 컴퓨터-읽기 가능한 메모리 디바이스에 있어서, 요구자에 의해 요구된 호출을 하는 디바이스에 저장된 다수의 방법 중 하나에 대응하는 각각의 보안 객체로서, 상기 각각의 보안 객체가:

상기 대응하는 방법을 확인하는 하나의 방법 식별자와, 그리고

상기 대응하는 방법에 대한 보안 조건을 규정하는 하나의 플래그를 포함하는 것을 특징으로 하는 다수의 보안 객체를 포함하는 보안 클래스 라이브러리 데이터 구조.

청구항 16

제 15 항에 있어서, 상기 플래그는 상기 대응하는 방법이 요구자의 익명을 지원하는지를 지정하는 하나의 익명성 플래그를 포함하는 것을 특징으로 하는 보안 클래스 라이브러리 데이터 구조.

청구항 17

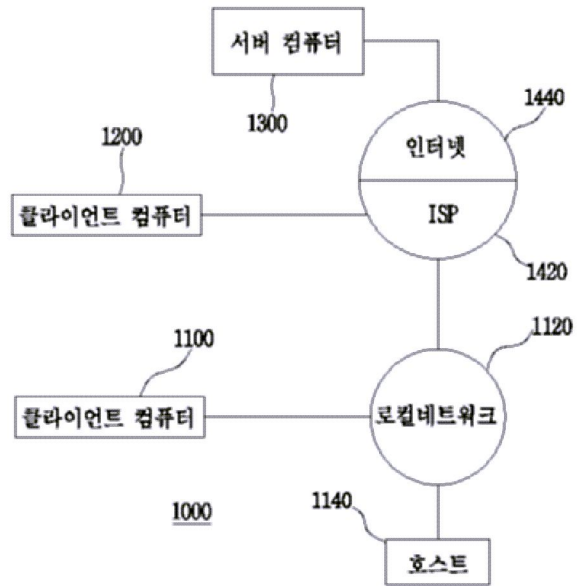
제 15 항에 있어서, 상기 플래그는 상기 대응하는 방법이 디바이스 인증을 지원하는지를 지정하는 하나의 수신자 인증 플래그를 포함하는 것을 특징으로 하는 보안 클래스 라이브러리 데이터 구조.

청구항 18

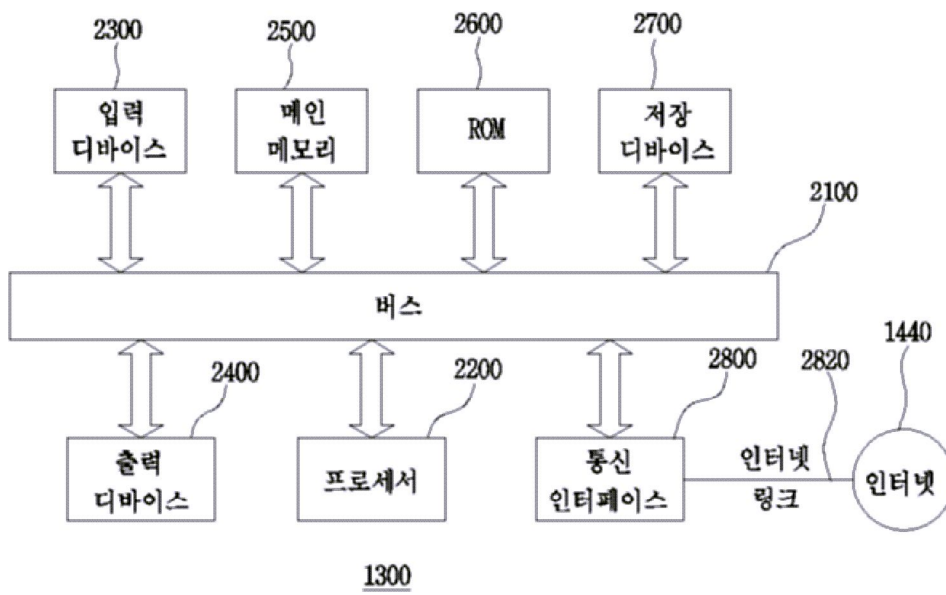
제 15 항에 있어서, 상기 플래그는 상기 대응하는 방법이 상기 요구자의 증명하에 위임성을 지원하는지를 지정하는 하나의 위임성 플래그를 포함하는 것을 특징으로 하는 보안 클래스 라이브러리 데이터 구조.

도면

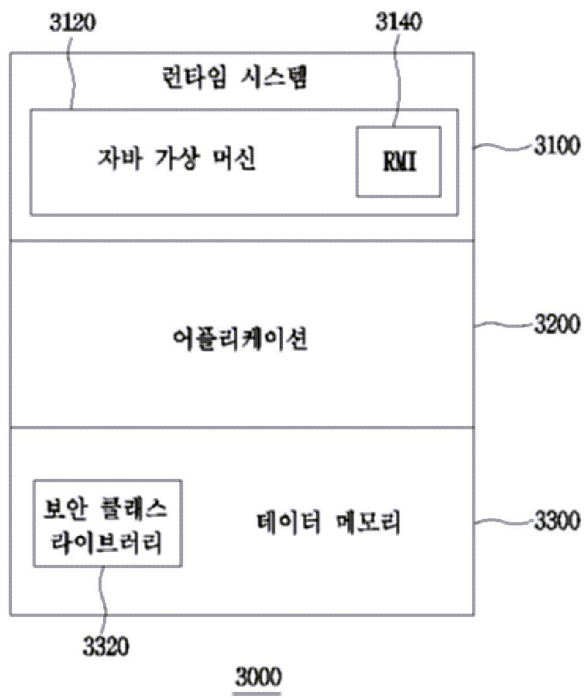
도면1



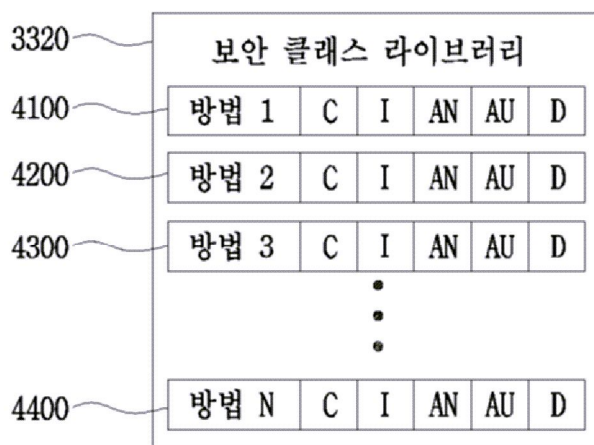
도면2



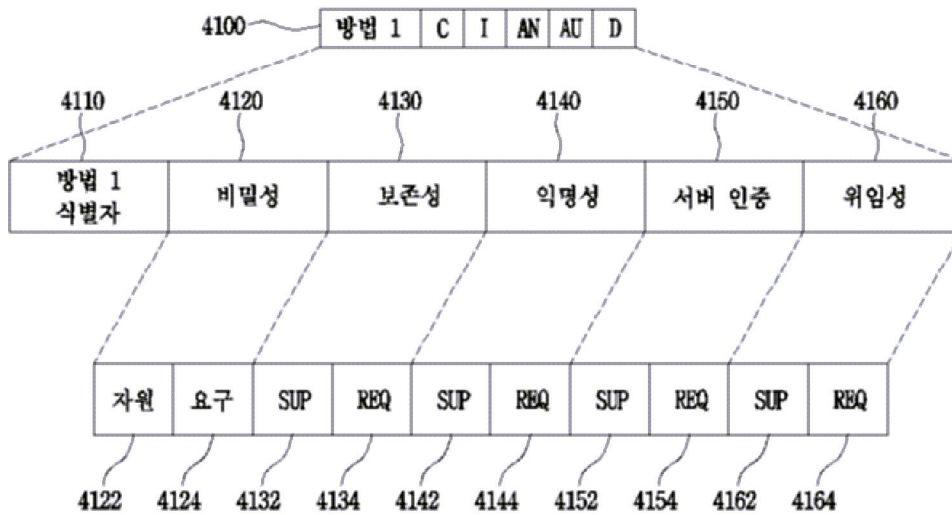
도면3



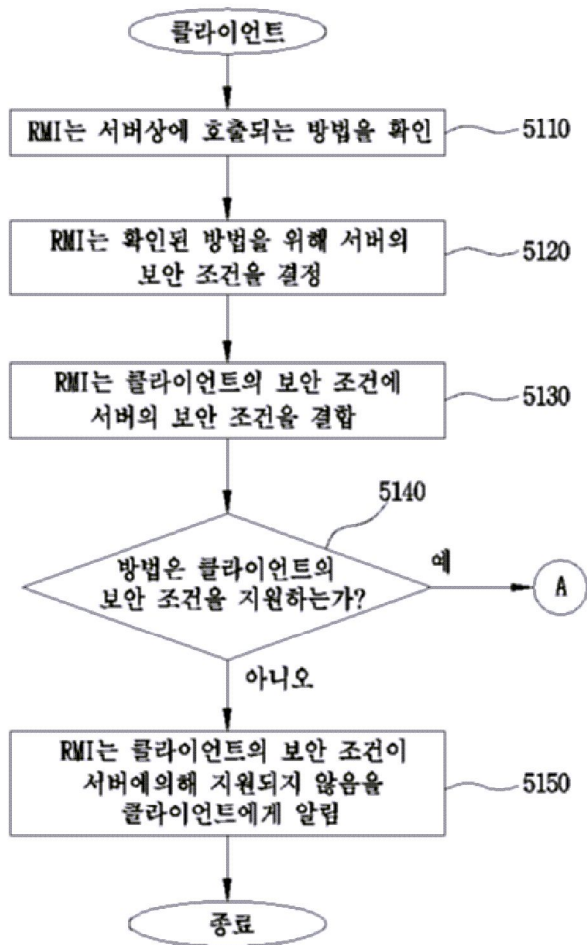
도면4a



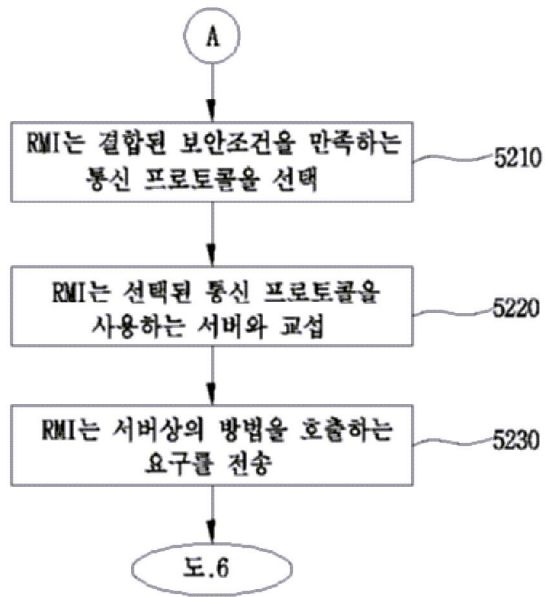
도면4b



도면5a



도면5b



도면6

