



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0027874
(43) 공개일자 2022년03월08일

(51) 국제특허분류(Int. Cl.)
G06F 21/53 (2013.01) G06F 12/14 (2006.01)
G06F 21/55 (2013.01) G06F 21/57 (2013.01)
G06F 21/79 (2013.01)
(52) CPC특허분류
G06F 21/53 (2013.01)
G06F 12/1441 (2013.01)
(21) 출원번호 10-2021-7042911
(22) 출원일자(국제) 2020년06월08일
심사청구일자 없음
(85) 번역문제출일자 2021년12월28일
(86) 국제출원번호 PCT/US2020/036575
(87) 국제공개번호 WO 2021/006973
국제공개일자 2021년01월14일
(30) 우선권주장
62/872,233 2019년07월09일 미국(US)
16/565,271 2019년09월09일 미국(US)

(71) 출원인
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
(72) 발명자
렌케 맥스웰 크리스토퍼
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 테크놀로지
라이선싱, 엘엘씨
스타크 테일러 제임스
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 테크놀로지
라이선싱, 엘엘씨
(뒷면에 계속)
(74) 대리인
김태홍, 김진희

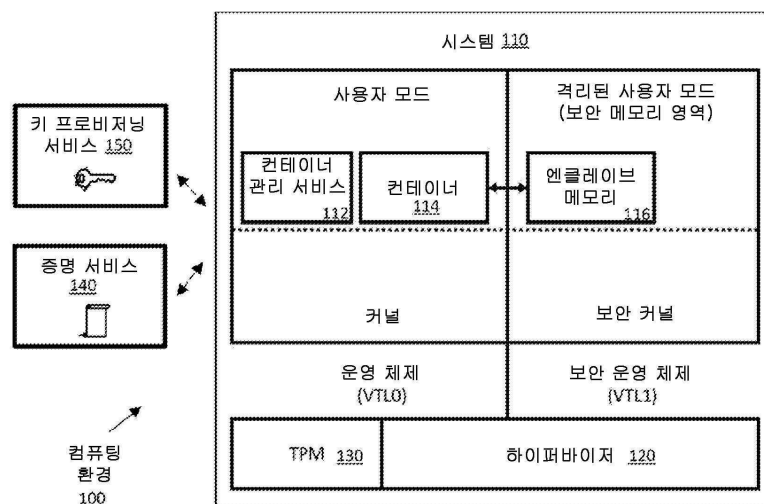
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 프로세스 컨테이너의 컨텍스트에서 보안 메모리 엔클레이브 사용

(57) 요약

메모리는 컨테이너 기반 메모리 엔클레이브로 분할 및 격리된다. 컨테이너 기반 메모리 엔클레이브는 증명 가능한 보안 보장을 갖는다. 컨테이너 이미지로부터 컨테이너 기반 메모리 엔클레이브를 프로비저닝하는 동안, 컨테이너에서 엔클레이브의 메모리 주소로의 의도된 링크는 분할된 메모리 엔클레이브와 같은 호스트의 실제 메모리 주소로 확인 가능하게 링크하도록 수정된다. 일부 경우에, 엔클레이브 증명 보고서는 각각의 새로운 증명 요청에 대해 컨테이너와 원격 증명 서비스 간의 중단 간 증명을 요구하지 않고 유사한 컨테이너로부터의 하나 이상의 이전 컨테이너 증명 보고서에 대한 이전 증명을 기반으로, 해당 증명 요청을 원격 증명 서비스로 전송하지 않고 검증될 수 있다.

대표도 - 도1



(52) CPC특허분류

G06F 21/552 (2013.01)

G06F 21/554 (2013.01)

G06F 21/57 (2013.01)

G06F 21/79 (2013.01)

(72) 발명자

술츠 벤자민 엠

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

비스와나탄 기리드하

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

스미스 프레데릭 저스투스

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

토마스 디푸 찬디

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

플라파카 하리 알

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

구오 엠버 티안키

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 마이크로소프트 테크놀로지 라이선싱, 엘엘씨

명세서

청구범위

청구항 1

컴퓨팅 시스템에 있어서,

하나 이상의 프로세서; 및

컨테이너 기반 메모리 엔클레이브(encclave)를 생성하기 위한 방법을 구현하기 위해 상기 하나 이상의 프로세서에 의해 실행 가능한 컴퓨터 실행 가능 명령어가 저장된 하나 이상의 컴퓨터 판독 가능 저장 매체

를 포함하고, 상기 방법은:

제 1 보안 도메인에서 실행되는 호스트 시스템에 의하여 사용되는 특정 보안 구성을 증명하는 데 사용되는 보안 구성요소를 갖는 컨테이너 이미지를 식별하는 단계 - 상기 보안 구성요소는 상기 호스트 시스템의 메모리 주소에 저장됨 - ;

새로운 컨테이너를 인스턴스화하기 위해 상기 컨테이너 이미지를 사용하는 단계 - 상기 새로운 컨테이너는 상기 호스트 시스템의 메모리 주소에 저장된 보안 구성요소에 기초하여 호스트 시스템의 제 2 보안 도메인에서 엔클레이브 메모리를 이용하도록 구성되고, 상기 새로운 컨테이너는 상기 컨테이너로부터 상기 호스트 시스템에 저장된 상기 보안 구성요소의 메모리 주소로의 의도된 링크인 특정 메모리 주소를 갖는 상기 보안 구성요소의 사본을 포함함 - ; 및

상기 새로운 컨테이너를 인스턴스화하는 동안 또는 이후에, 상기 컨테이너로부터 상기 호스트 시스템에 저장된 상기 보안 구성요소의 상기 특정 메모리 주소로의 의도된 링크를 수정하여, 상기 컨테이너로부터 상기 메모리 주소로의 상기 의도된 링크가 상기 컨테이너를 상기 호스트 시스템의 메모리 주소에 확인 가능하게 링크하도록 수정되는 단계

를 포함하는 것인, 컴퓨팅 시스템.

청구항 2

제 1 항에 있어서,

상기 의도된 링크를 수정하는 단계는 상기 호스트 시스템에 의해 사용되는 상기 보안 구성요소가 로드되는 상기 호스트 시스템의 주소로 상기 새로운 컨테이너의 상기 보안 구성요소의 상기 특정 주소를 덮어쓰는 단계를 포함하는 것인, 컴퓨팅 시스템.

청구항 3

제 1 항에 있어서,

상기 의도된 링크를 수정하는 단계는 상기 새로운 컨테이너의 상기 보안 구성요소의 상기 특정 주소로부터 상기 호스트 시스템에 의해 사용되는 상기 보안 구성요소가 로드되는 상기 호스트 시스템의 주소로의 링크를 생성하는 단계를 포함하는 것인, 컴퓨팅 시스템.

청구항 4

제 1 항에 있어서,

상기 의도된 링크를 수정하는 단계는 상기 호스트 시스템에 의해 사용되는 상기 보안 구성요소가 로드되는 상기 호스트 시스템의 주소를 포함하거나 참조하도록 상기 특정 주소와 연관된 매핑 테이블의 참조 주소를 업데이트하는 단계를 포함하는 것인, 컴퓨팅 시스템.

청구항 5

제 4 항에 있어서,

상기 방법은,

상기 매핑 테이블을 생성하는 단계; 및

상기 새로운 컨테이너를 인스턴스화하는 동안 또는 이후에, 상기 참조 주소를 상기 매핑 테이블에 추가하기 위해 상기 새로운 컨테이너와 연관된 메타데이터를 사용하는 단계

를 더 포함하는 것인, 컴퓨팅 시스템.

청구항 6

제 1 항에 있어서,

상기 보안 구성요소는 라이브러리 파일을 포함하는 것인, 컴퓨팅 시스템.

청구항 7

제 1 항에 있어서,

상기 보안 구성요소는 상기 새로운 컨테이너에 대한 상기 보안 구성에 관한 증명 보고서를 생성하는 데 사용되는 데이터를 포함하는 것인, 컴퓨팅 시스템.

청구항 8

제 7 항에 있어서,

상기 방법은,

상기 새로운 컨테이너에 대한 상기 보안 구성과 연관된 보안 보장의 증명을 수행하는 단계

를 더 포함하는 것인, 컴퓨팅 시스템.

청구항 9

컴퓨팅 시스템에 있어서,

하나 이상의 프로세서; 및

컨테이너 기반 메모리 엔클레이브에 대한 증명을 수행하기 위한 방법을 구현하기 위해 상기 하나 이상의 프로세서에 의해 실행 가능한 컴퓨터 실행 가능 명령어가 저장된 하나 이상의 컴퓨터 판독 가능 저장 매체

를 포함하고, 상기 방법은:

하나 이상의 컨테이너를 식별하는 단계 — 상기 하나 이상의 컨테이너 각각은 호스트 시스템의 제 1 보안 도메인과 상이한 보안 도메인에 상주하는 엔클레이브 메모리를 참조하도록 구성되고 상기 엔클레이브 메모리에 대한 특정 보안 구성을 증명하기 위해 보안 구성요소를 사용하며, 상기 보안 구성요소는 상기 호스트 시스템의 메모리 주소에 로드되고 상기 하나 이상의 컨테이너에서 상기 보안 구성요소에 대한 하나 이상의 참조에 의해 링크됨 — ;

상기 보안 구성요소에 기초하여 상기 하나 이상의 컨테이너의 보안 구성의 보안 보장에 관한 단일 증명 보고서를 원격 증명 서비스로부터 획득하는 단계;

상기 원격 증명 서비스로부터 상기 단일 증명 보고서를 수신한 후 상기 하나 이상의 컨테이너 중 특정 컨테이너에 대한 보안 보장을 증명하기 위한 요청을 수신하는 단계; 및

상기 증명 요청을 수신하기 전에 수신된 상기 단일 증명 보고서에 기초하여 상기 특정 컨테이너에 대한 보안 보장을 증명하는 단계

를 포함하는 것인, 컴퓨팅 시스템.

청구항 10

제 9 항에 있어서,

상기 방법은,

상기 단일 증명 보고서에 기초하여 상기 호스트 시스템의 메모리 주소에 로드된 상기 보안 구성요소에 대한 참조를 각각 포함하는 복수의 컨테이너에 대한 보안 보장을 증명하는 단계를 더 포함하는 것인, 컴퓨팅 시스템.

청구항 11

제 10 항에 있어서,

상기 단일 증명 보고서는 상기 호스트 시스템의 주소에 로드된 상기 보안 구성요소에 대한 참조를 갖는 상이한 컨테이너의 보안 보장에 대한 증명 요청에 응답하여 획득되는 것인, 컴퓨팅 시스템.

청구항 12

제 9 항에 있어서,

상기 방법은,

상기 컨테이너에 대한 할당된 엔클레이브 메모리의 재할당에 후속하여 생성되는, 엔클레이브 메모리가 할당된 특정 컨테이너에 대한 새로운 증명 요청을 수신하는 단계; 및

상기 컨테이너의 무결성을 확인하고, 상기 원격 증명 서비스에 상기 새로운 증명 요청이 제출되도록 요구하지 않고 새로운 증명 보고서를 발행하는 단계

를 더 포함하는 것인, 컴퓨팅 시스템.

청구항 13

제 9 항에 있어서,

상기 방법은,

컨테이너 상태를 모니터링하는 단계; 및

손상된 것으로 결정된 컨테이너를 디스에이블하는 단계

를 더 포함하는 것인, 컴퓨팅 시스템.

청구항 14

제 9 항에 있어서,

상기 방법은,

후속 컨테이너 증명 요청을 수신할 때, 상기 후속 컨테이너 증명 요청과 연관된 특정 컨테이너의 속성을 결정하는 단계; 및

상기 특정 컨테이너의 속성에 기초하여 상기 후속 컨테이너 증명 요청을 원격 서비스로 결정적으로 (deterministically) 라우팅하거나, 대안적으로 상기 특정 컨테이너의 속성에 적어도 부분적으로 기초하여 상기 후속 컨테이너 증명 요청을 상기 원격 서비스로 라우팅하지 않고 증명 보고서를 생성하는 단계

를 더 포함하는 것인, 컴퓨팅 시스템.

청구항 15

제 14 항에 있어서,

상기 후속 컨테이너의 속성은 상기 컨테이너의 수명을 포함하고, 제 1 컨테이너 세트는 제 2 컨테이너 세트보다 상대적으로 더 긴 수명을 가지며, 상기 제 1 컨테이너 세트와 연관된 하나 이상의 컨테이너 증명 요청은 상기 원격 서비스로 라우팅되는 반면, 상기 제 2 컨테이너 세트를 상기 원격 서비스로 라우팅하지 않고 상기 제 1 컨테이너 세트보다 상대적으로 더 짧은 수명을 갖는 후속 컨테이너를 포함하는 상기 제 2 컨테이너 세트 각각에 대해 하나 이상의 증명 보고서가 생성되는 것인, 컴퓨팅 시스템.

발명의 설명

기술분야

배경기술

- [0001] 컴퓨터 기술은 이제 컴퓨터가 거의 모든 산업에 영향을 미치고 서로 다른 유형의 데이터를 저장하고 처리할 수 있게 하는 유비쿼터스 수준으로 발전했다.
- [0002] 데이터를 안전하게 저장하고 처리하기 위해, 일부 운영 체제와 컴퓨터 하드웨어는 메모리 분할을 제공하도록 구성되었다. 메모리 분할은 단일 컴퓨팅 시스템 내에서 및/또는 복수 컴퓨팅 시스템의 분산 네트워크 내에서 데이터의 저장 및 처리를 분리하는 한 가지 방법이다.
- [0003] 예를 들어, 시스템은 민감하지 않은 다른 컴퓨팅 프로세스 및 데이터로부터, 암호화 키, 사용자 ID, 사용자 자격 증명서 및 기타 민감한 데이터와 같은 "보안 중요" 데이터를, 본 명세서에서 엔클레이브(enclave)로 지칭되는 격리된 메모리 영역 내에 격리할 수 있다. 컴퓨팅 업계는 때때로 이러한 유형의 격리된 컴퓨팅 환경을, 예를 들어, "신뢰할 수 있는 실행" 환경 및 "기밀 컴퓨팅" 환경으로 지칭한다.
- [0004] 메모리를 엔클레이브로 분할하는 기술은 일반적으로 여러 형태로 존재한다. 한 가지 예는 Intel의 SGX, ARM Holdings의 TrustZone 등으로 구현된 하드웨어 기반 분할이다. 또 다른 예는 가상화 기반 보안(Virtualization Based Security; VBS)과 같은 운영 체제 또는 하이퍼바이저를 기반으로 한다.
- [0005] VBS는 하이퍼바이저 또는 다른 자원 관리 계층을 사용하여 컴퓨팅 시스템 내에 추가 메모리 파티션을 생성하여 그 안에 배치된 보안 중요 데이터를 격리하고 보호한다. VBS를 사용하면, 사용자 모드와 커널 모드 모두에서 격리를 제공하는 데 사용될 수 있는 여러 가상 신뢰 레벨(Virtual Trust Level; VTL)을 정의할 수 있다. 예를 들어, 일반적인 운영 체제는 기존 사용자 모드 및 커널 모드 드라이버, 애플리케이션, 서비스 등을 지원하기 위해 제 1 VTL(예를 들어, VTL0)에서 실행될 수 있는 반면, 추가 메모리 파티션은 격리된 사용자 모드(Isolated User Mode; IUM) 및 보안 커널(Secure Kernel; SK)의 구현을 지원하기 위해 동일한 시스템의 제 2 VTL(예를 들어, VTL1)에서 실행될 수 있다.
- [0006] VTL 사이의 보안 경계는 하이퍼바이저 또는 다른 관리 계층에서 시행할 수 있어, 예를 들어, VTL1 내의 메모리에 매핑되고 VTL0이 액세스할 수 없는 격리된 메모리 영역을 프로세스 내에 생성하여 VTL0이 VTL1에 액세스하는 것을 방지할 수 있다. 이 격리된 메모리 영역이 엔클레이브이다. 이러한 엔클레이브는 신뢰할 수 없는 환경에서 안전하고 증명 가능한 계산 수단을 제공한다. 따라서, VTL1에서 실행되는 프로세스가 사용자 모드나 커널 모드에 의해 공격을 받더라도, VTL1은 엔클레이브의 내용을 보호할 수 있다.
- [0007] 하드웨어 기반 분할은 메모리에 격리된 엔클레이브를 생성하는 데 사용할 수 있는 다른 일반적인 유형의 메모리 분할이다. 하드웨어 분할의 한 예는 Intel의 SGX(Secure Guard Extensions)를 포함하며, 이는 VBS와 유사한 방식으로 사용자 모드 코드가 엔클레이브를 포함하는 전용 메모리 영역을 할당할 수 있도록 하는 프로세서 명령어 세트를 포함한다.
- [0008] 엔클레이브는 VBS에 의해 보호되든 또는 기타 하드웨어 기반 엔클레이브 기술에 의해 보호되든, 일반적으로 호스트 프로세스의 컨텍스트 내에 존재하여 엔클레이브와의 임의의 상호 작용이 호스트 프로세스의 기존 보호에 종속된다는 것을 이해할 것이다.
- [0009] 특히 가상 머신 기반 가상화를 사용하는 분할의 한 가지 이점은 서버 자원의 고밀도 배치를 가능하게 하고 클라우드 컴퓨팅을 가능하게 하는 방식으로 데이터 및 애플리케이션을 분할할 수 있다는 것이다.
- [0010] 컨테이너 기반 가상화(때때로 네임스페이스 기반 가상화라고도 함)는 확장된 서비스를 위해 훨씬 더 향상된 호환성 및 증가된 밀도를 제공할 수 있는 분할 유형 중 하나이다. 이것은 호환성이 높을수록 소프트웨어 개발 비용이 낮아지고 밀도가 높을수록 시설, 작업 및 하드웨어의 동일한 상대 비용으로 더 많은 수익을 올릴 수 있기 때문에 좋다.
- [0011] 컨테이너 및 컨테이너 기반 가상화를 사용하여, 격리되고 자원 제어되며 이식 가능 운영 환경을 제공하는 것이 또한 가능하다. 예를 들어, 소프트웨어 애플리케이션이 컨테이너 내에서 실행될 때, 시스템의 나머지 부분에 영향을 미치지 않고 실행될 수 있고 시스템은 해당 애플리케이션에 영향을 미치지 않는다. 컨테이너는 하나의 애플리케이션에 특유하거나, 여러 애플리케이션에 런타임을 제공하여 추가 운영 체제 기능 및 서비스(예를 들어,

입출력 기능, 보안 등)를 제공할 수 있다.

[0012] 그러나, 애플리케이션 및 서비스를 종래의 컨테이너로 분할하는 것은 애플리케이션 또는 서비스의 유연성 및 이식성을 증가시키는 것을 포함하여 많은 이점을 갖지만, 전통적인 컨테이너 보안 경계는 다소 구멍이 있다. 그 이유 중 하나는 기존 컨테이너 기술이 특정 하드웨어 보안 보장보다 기본 운영 체제의 보안에 의존하고, 기존 컨테이너의 구조가 애플리케이션 호환성을 용이하게 하기 위해 의도적으로 유연하여 컨테이너 내의 데이터를 보호하는 방법에 격차가 있기 때문이다. 이와 관련하여, 컨테이너는 앞서 언급한 VBS 및 하드웨어 분할 엔클레이브 기술과 동일한 수준의 증명 가능한 보안 보장을 제공할 수 없다.

[0013] 그러나, IoT 및 데이터베이스 워크로드에서 컨테이너형 애플리케이션을 위한 보안 메모리 솔루션이 필요한 많은 새로운 컨테이너 시나리오가 개발되고 있다. 그러나 불행하게도, 특히 효율적이고 확장 가능한 방식으로 이러한 보안 메모리를 제공하는 데 존재하는 여러 문제로 인해 이러한 유틸리티를 제공하기 위한 시스템이 아직 만들어지지 않았다. 이러한 단점은 분산 시스템에서, 특히 강화된 보안이 필요한 경우, 컨테이너 사용의 광범위한 채택을 효과적으로 제한했다.

[0014] 예를 들어, 특히 컨테이너 보안 시나리오에 대해 강력한 모니터링 및 증명과 함께 보안 메모리 할당을 제공/시행할 수 있는 운영 체제는 현재 없다. 컨테이너의 라이프 사이클이 또한 짧으므로, VBS 및 하드웨어 분할에 사용되는 것과 같은 다른 유형의 분할 기술에 사용되는 프로비저닝, 모니터링, 중단 간 보안 키 관리 및 증명 기술을 사용하는 것이 다소 금지된다.

[0015] 따라서, 안전하고 증명 가능한 방식으로, 특히 컨테이너의 기능적 이점을 사용할 수 있는 방식으로, 메모리의 분할 및 격리를 가능하게 하는 시스템 및 방법이 계속해서 필요하다. 일부 실시예에서, 보안 및 격리는 가상 머신과 같은 더 큰 형태의 가상화를 사용하여 달성될 수 있다. 일부 실시예에서, 일부 하이퍼바이저 기반 컨테이너 구현이 엔클레이브를 필요로 할 수 있다. 일부 실시예에서, 작은 컨테이너 이미지를 검증하도록 증명이 사용될 수 있고, 메모리 엔클레이브에서만 보안 집약적 워크로드를 호스팅하도록 컨테이너 기반 가상화 기술(격리가 덜함)이 사용될 수 있기 때문에, 더 큰 형태의 가상화 기술이 필요하지 않다.

[0016] 그러나, 본 명세서에 청구된 주제는 기존 시스템과 관련된 임의의 단점을 반드시 해결하거나 위에서 설명된 것과 같은 환경에서만 작동하는 실시예로 제한되지 않는다는 점을 유념해야 한다. 오히려, 이 배경기술은 단지 본 명세서에서 설명된 일부 실시예가 실시될 수 있는 하나의 예시적인 기술 분야를 설명하기 위해 제공된다.

발명의 내용

[0017] 개시된 실시예는 컨테이너 기반 메모리 엔클레이브를 포함하고 증명 가능한 보안 보장을 갖는 컨테이너를 프로비저닝하기 위한 실시예에 관한 것이다. 개시된 실시예는 또한 컨테이너 기반 메모리 엔클레이브를 포함하는 프로비저닝된 컨테이너에 대한 보안 보장을 증명하기 위한 실시예에 관한 것이다.

[0018] 개시된 실시예는 컨테이너 기반 메모리 엔클레이브의 형태로 보안 메모리 격리를 갖는 컨테이너를 프로비저닝하기 위한 새로운 시스템 및 방법을 제공하고, 효율적이고 확장 가능한 방식으로 컨테이너의 보안 보장에 대한 증명을 효율적으로 관리하기 위한 증명 인프라를 제공함으로써 기존 컨테이너 프로비저닝 기술에 비해 기술적 이점을 제공한다.

[0019] 일부 실시예는 컨테이너 기반 메모리 엔클레이브를 생성하기 위한 방법 및 시스템을 포함한다. 이러한 실시예에서, 컴퓨팅 시스템은 제 1 보안 도메인에서 실행되는 호스트 시스템에 의해 사용되는 특정 보안 구성을 증명하는 데 사용되는 보안 구성요소를 갖는 컨테이너 이미지를 식별하고, 보안 구성요소는 호스트 시스템의 메모리 주소에 저장된다. 일부 실시예에서, 가상 머신 또는 컨테이너 호스트는 VTL1 및 VTL0 메모리를 포함하는 호스트 시스템을 구현한다는 점을 유념한다.

[0020] 그런 다음, 컴퓨팅 시스템은 컨테이너 이미지를 사용하여 제 1 보안 도메인과 상이한 제 2 보안 도메인에서 새로운 컨테이너를 인스턴스화하고, 새로운 컨테이너는 호스트 시스템의 메모리 주소에 저장된 보안 구성요소를 기반으로 호스트 시스템의 엔클레이브 메모리를 사용하도록 구성되고, 새로운 컨테이너는 컨테이너로부터 호스트 시스템에 저장된 보안 구성요소의 메모리 주소로의 의도된 링크인 특정 메모리 주소를 가진 보안 구성요소의 사본을 포함한다.

[0021] 추가로, 새로운 컨테이너의 인스턴스화 동안 또는 이후에, 컴퓨팅 시스템은 컨테이너로부터 호스트 시스템에 저장된 보안 구성요소의 특정 메모리 주소로의 의도된 링크를 수정하여 컨테이너로부터 메모리 주소로의 의도된 링크는 컨테이너를 호스트 시스템의 메모리 주소에 확인 가능하게 링크하도록 수정되어 컨테이너에 대해 증명

가능한 상태를 생성한다.

[0022] 일부 실시예는 또한 컨테이너 기반 메모리 엔클레이브에 대한 증명을 수행하기 위한 방법 및 시스템을 포함한다. 이들 실시예에서, 컴퓨팅 시스템은 호스트 시스템의 제 1 보안 도메인과 상이한 보안 도메인에 상주하는 하나 이상의 컨테이너를 식별하고, 하나 이상의 컨테이너 각각은 호스트 시스템의 엔클레이브 메모리를 참조하도록 구성되고 엔클레이브 메모리에 대한 특정 보안 구성을 증명하기 위해 보안 구성요소를 사용하며, 보안 구성요소는 호스트 시스템의 메모리 주소에 저장되고 하나 이상의 컨테이너에서 보안 구성요소에 대한 하나 이상의 참조에 의해 링크된다.

[0023] 그런 다음, 컴퓨팅 시스템은 보안 구성요소를 기반으로 하나 이상의 컨테이너의 보안 구성의 보안 보장에 관한 증명 보고서를 원격 증명 서비스로부터 획득한다. 일부 경우에, 이것은 여러 증명 보고서가 아니라 단일 증명 보고서이다. 그런 다음, 원격 증명 서비스로부터 증명 보고서를 수신한 후, 컴퓨팅 시스템은 하나 이상의 컨테이너 중 특정 컨테이너에 대한 보안 보장을 증명하기 위한 요청을 수신한다. 그런 다음, 컴퓨팅 시스템은 컨테이너의 새로운 중단 간 증명과 새로운 요청에 대한 원격 증명 서비스를 요구하지 않고, 증명 요청을 수신하기 전에 수신된 증명 보고서를 기반으로 특정 컨테이너에 대한 보안 보장을 증명한다.

[0024] 이 요약은 아래의 상세한 설명에서 추가 설명되는 개념들의 선택을 간략화된 형태로 소개하기 위해 제공된다. 이 요약은 청구된 주제의 중요 특징 또는 필수 특징을 식별하기 위한 것도 아니고, 청구된 주제의 범위를 결정하기 위해 이용되는 것도 아니다.

[0025] 추가 특징 및 장점은 다음의 설명에서 기재될 것이며, 부분적으로는 설명으로부터 명백해지거나, 또는 본 명세서의 교시의 실시예에 의해 학습될 수 있다. 본 발명의 특징 및 장점은 특히 첨부된 청구범위에서 지적된 기기 및 조합에 의해 실현되고 획득될 수 있다. 본 발명의 특징은 다음의 설명 및 첨부된 청구범위로부터 보다 완전하게 명백해질 것이며, 또는 이하에 기재된 바와 같이 본 발명의 실시예에 의해 학습될 수 있다.

도면의 간단한 설명

[0026] 앞서 언급한 장점과 특징 및 다른 장점과 특징이 획득될 수 있는 방식을 설명하기 위해서, 위에서 간략하게 설명된 주제의 더욱 구체적인 설명이 첨부된 도면에 도시된 특정 실시예를 참조하여 제공될 것이다. 이러한 도면은 단지 전형적인 실시예를 도시하고, 따라서 그 범위를 제한하는 것으로 간주되어서는 안 된다는 것을 이해하고, 실시예는 첨부된 도면의 사용을 통해 추가적인 특이성 및 세부 사항으로 설명되고 해석될 것이다.

도 1은 키 프로비저닝 서비스 및 증명 서비스가 컴퓨팅 시스템과 통신하는 컴퓨팅 환경을 도시하며, 컴퓨팅 시스템은 제 1 도메인(예를 들어, VTL0)에서 작동하는 사용자 모드 및 보안 커널을 사용하는 격리된 사용자 모드를 제공하고, 격리된 메모리 엔클레이브는 제 2 도메인(예를 들어, VTL1)에서 작동한다.

도 2는 증명 가능한 상태를 갖는 컨테이너 및 엔클레이브를 사용하고 컨테이너 관리 서비스, 증명 서비스 및 키 프로비저닝 서비스를 포함하는 시스템의 설계 개요의 일 실시예를 도시한다.

도 3은 개시된 실시예의 양태를 포함 및/또는 구현하도록 구성된 컴퓨팅 시스템을 도시한다.

도 4는 키 프로비저닝 서비스 및 증명 서비스 인프라와 청구된 발명의 양태를 포함 및/또는 구현하도록 구성된 다른 컴퓨팅 시스템을 도시한다.

도 5는 개시된 실시예에 따라 컨테이너를 프로비저닝하기 위한 방법과 연관된 복수의 동작을 갖는 흐름도의 일 실시예를 도시한다.

도 6은 개시된 실시예에 따라 컨테이너의 보안 보장을 증명하기 위한 방법과 연관된 복수의 동작을 갖는 흐름도의 일 실시예를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0027] 개시된 실시예는 컨테이너 기반 메모리 엔클레이브를 포함하고 증명 가능한 보안 보장을 갖는 컨테이너를 프로비저닝하기 위한 실시예에 관한 것이다. 개시된 실시예는 또한 컨테이너 기반 메모리 엔클레이브를 포함하는 프로비저닝된 컨테이너에 대한 보안 보장을 증명하기 위한 실시예에 관한 것이다.

[0028] 일부 경우에, 개시된 실시예는 컨테이너 기반 메모리 엔클레이브로서 컨테이너를 프로비저닝하기 위한 새로운 시스템 및 방법을 제공하고, 효율적이고 확장 가능한 방식으로 컨테이너의 보안 보장에 대한 증명을 효율적으로 관리하기 위한 증명 인프라를 제공함으로써 종래의 컨테이너 프로비저닝 기술 및 기존 엔클레이브 기술에 비해

기술적 이점을 제공한다.

- [0029] 컨테이너 기반 엔클레이브의 생성은 기존 VBS 및 기존 하드웨어 기반 엔클레이브에서 가능한 것보다 훨씬 더 많은 메모리 파티션 격리를 생성한다. 예를 들어, 기존 엔클레이브는 VBS에 의해 보호되든 또는 기타 하드웨어 기반 엔클레이브 기술에 의해 보호되든, 일반적으로 호스트 프로세스의 컨텍스트 내에 존재하여 엔클레이브와의 임의의 상호 작용이 호스트 프로세스의 기존 보호에 종속된다는 것을 유념한다. 그러나, 일반적인 운영 체제 프로세스는 전통적으로 강력한 격리 메커니즘을 갖고 있지 않지만, 본 명세서에 설명된 방식으로 프로세스 컨테이너의 컨텍스트 내에 엔클레이브 호스트 프로세스를 캡슐화함으로써, 시스템에 추가적인 격리 레벨을 추가하는 것이 가능하다. 컨테이너 기반 엔클레이브의 사용은 또한 효율적이고 확장 가능한 방식으로 보안 보장 증명을 제공함으로써 모니터링 및 검증될 수 있다.
- [0030] 컨테이너 기반 엔클레이브의 프로비저닝 및 모니터링뿐만 아니라 효율적이고 확장 가능한 방식으로 컨테이너 기반 엔클레이브의 보안 보장을 증명하는 것을 포함하는 개시된 실시예의 추가적인 이점 및 기능이 아래에서 설명될 것이다.
- [0031] 먼저, 개시된 실시예의 양태를 구현하도록 구성된 컴퓨팅 시스템(110)을 포함하는 컴퓨팅 환경(100)을 도시하는 도 1에 주의를 기울일 것이다. 이 컴퓨팅 시스템(110)은, 예를 들어, 컨테이너(114) 및 컨테이너 기반 메모리 엔클레이브(116)를 제공하도록 구성된 컨테이너 관리 서비스(112)를 포함한다. 컴퓨팅 시스템(110)은 또한 제 1 도메인(예를 들어, 운영 체제 VTL0)에서 동작하는 사용자 모드에서 컨테이너 관리 서비스(112) 및 컨테이너(114)를 구현하고, 제 2 도메인(예를 들어, 운영 체제 VTL1)에서 격리된 컨테이너 기반 메모리 엔클레이브(116) 및 대응하는 보안 커널을 구현하도록 구성된다.
- [0032] 하이퍼바이저(120) 또는 다른 관리 계층이 제 1 도메인(VTL0)과 제 2 도메인(VTL1) 사이의 보안 경계를 구현하도록 구성된다. TPM(Trusted Platform Module; 신뢰 플랫폼 모듈)(130)이 증명 서비스(140)에 상태를 증명하도록 구성된다. 일부 경우에, 이는 키 프로비저닝 서비스(150) 또는 다른 키 프로비저닝 서비스와 같이 TPM 제조업체에 의해 프로비저닝된 인증된 보증 키(endorsement key; EK)를 제공함으로써 달성된다.
- [0033] 사용 중에 TPM을 실행하는 호스트는 증명 ID 키(attestation identity key; AIK)를 생성하며, 이는 나중에 호스트의 부팅 상태를 무결성 보호하고 증명하는 데 사용할 수 있는 AIK 인증서 또는 기타 보고서에서 TPM 클레임(예를 들어, TPM의 프리미티브를 증명하는 보안 보장)에 서명하는 데 사용할 수 있다. 증명 보고서 요청에 응답하여, AIK 인증서는 처음에 TPM(130) 및/또는 컴퓨팅 시스템(110)에 제공된다. TPM에 의해 프로비저닝된 인증된 보증 키(EK) 및 AIK의 공존을 검증 및 증명하는 것에 응답하여, AIK 인증서는 증명 서비스 또는 기타 인증서 서비스에 의해 제공된다.
- [0034] 증명 서비스는 또한 특정 컨테이너 기반 엔클레이브에 대한 보안 보장을 검증하거나 증명할 수 있다(예를 들어, 격리 보안 보장 및 프리미티브에 대한 증명). 예를 들어, 호스트는 TPM 플랫폼 구성 레지스터(Platform Configuration Register; PCR)에서 운영 체제 바이너리 해시와 시스템 부팅 체인을 측정하고, 이는 AIK로 디지털 서명되고 증명 서비스에 의해 검증된 증명 보고서에서 엔클레이브 바이너리를 입증하고 증명하는 데 사용될 수 있다. 보안 커널 부팅 중에 TPM에서 시스템 ID 키(system identity key; IDK)가 생성되고 측정된다. 일부 실시예에서, 부팅 순서는 또한 시스템 부팅 체인의 일부로서 측정된다.
- [0035] 부팅 후, 엔클레이브는 증명 서비스에 컨테이너에 대한 증명 보고서를 요청한다. 선택적으로, 엔클레이브는 엔클레이브가 클라이언트와 공유할 수 있는 키를 컨테이너 증명 보고서에 추가할 수 있다. 이 키를 구현하기 위해, 엔클레이브는 먼저 키 쌍을 생성하고 엔클레이브 증명 요청에서 시스템 ID 키(IDK)로 엔클레이브 키 쌍의 공개 키에 디지털 서명하며, 엔클레이브 증명 요청은 호스트 TPM 부팅 로그 및 PCR과 함께 증명 서비스(예를 들어, 증명 서비스(140) 또는 다른 증명 서비스)로 전송되며, 모두 AIK 및 AIK 인증서로 서명된다. 엔클레이브 공개 키가 컨테이너 증명 보고서에 추가되면, 이 증명 보고서는 엔클레이브 증명 보고서로 처리될 수 있고 키는 이 엔클레이브와 안전하게 통신하는 데 사용될 수 있다.
- [0036] 증명 서비스(140)는 TPM 부팅 로그가 AIK/인증서로 올바르게 서명되었고 PCR에서 측정된 부팅 체인이 TPM 부팅 로그와 일치하는지 검증한다. 또한, 부팅 로그의 IDK가 증명 요청에서 엔클레이브 공개 키에 서명하는 데 사용된 IDK와 일치하는지 검증한다. 증명 서비스는 또한 부팅 로그의 IDK가 증명 요청에서 엔클레이브 공개 키에 서명하는 데 사용된 IDK와 일치하는지 확인한다. 그렇다면, 증명 서비스는 시스템 상태와 함께 서명된 증명 보고서를 생성한다. 이 서명된 보고서는 엔클레이브 증명 보고서를 검증하고 현재 타임 스탬프와 함께 엔클레이브의 공개 키를 포함하므로, 컨테이너 기반 엔클레이브의 보안 보장을 증명한다.

- [0037] 도 2는 증명 가능한 상태를 갖는 컨테이너 및 엔클레이브를 사용하고 컨테이너 관리 서비스, 증명 서비스 및 키 프로비저닝 서비스를 포함하는 시스템의 설계 개요를 도시한다. 이들 구성요소 사이의 관계는 위에서 설명되었으며 도 3 및 도 4를 참조하여 이어지는 설명에서 훨씬 더 명백해질 것이다.
- [0038] 컨테이너 관리 서비스는 시스템에서 실행 중인 모든 컨테이너의 상태를 프로비저닝, 부팅, 모니터링 및 유지하도록 구성되어 임의의 N개의 컨테이너를 생성하기 위한 확장 가능한 인프라를 제공한다는 점을 유념한다.
- [0039] 각 컨테이너는 또한 자신의 엔클레이브를 생성, 초기화 및 할당하는 역할을 한다.
- [0040] 엔클레이브 및 컨테이너는 컨테이너 관리 서비스에 의해 추적되는 호스트 OS 커널을 통해 바인딩된다. 이 바인딩은 호스트 OS의 컨텍스트에서 신뢰할 수 있고 증명할 수 있지만 엔클레이브의 보안 경계를 유지하지 않는다.
- [0041] 증명 서비스는 적절한 엔클레이브 코드가 임의의 주어진 엔클레이브 내에서 실행되고 있음을 보장하는 데 사용될 수 있다.
- [0042] 일부 실시예에서, 커널은 시스템/플랫폼 증명 보고서를 증명 서비스에 보고할 수 있고, 따라서 엔클레이브(VBS, SGX 등)를 생성하는 데 사용되는 플랫폼의 무결성을 검증할 수 있다. 일부 실시예에서, 컨테이너는 엔클레이브 공개 키를 시스템 증명 보고서에 추가하여 엔클레이브 증명 보고서를 생성한다. 일부 실시예에서, 엔클레이브 내용에 대해 전체 증명이 실행되어 엔클레이브 증명 보고서를 생성한다.
- [0043] 엔클레이브/컨테이너는 자신의 엔클레이브 증명 보고서를 생성하고 그 보고서를 증명 서비스(위에서 설명됨)에 제공하는 역할을 한다. 증명 서비스는 컨테이너 관리 서비스와 협력하여 컨테이너와 해당 컨테이너와 상호 작용하는 프로세스 모두에 증명을 제공할 수 있다.
- [0044] 도 3 및 도 4는 컨테이너 기반 엔클레이브를 프로비저닝하고 증명하기 위해 개시된 실시예의 양태를 포함 및/또는 구현하도록 구성된 컴퓨팅 시스템을 도시한다. 도 3의 시스템(300A)은 도 5를 참조하여 도시되고 설명된 흐름도(500)에 구체적으로 대응하고, 도 4의 시스템(300B)은 도 6을 참조하여 도시되고 설명된 흐름도(600)에 구체적으로 대응한다.
- [0045] 도시된 바와 같이, 도시된 시스템(300A 및 300B)은 하나 이상의 하드웨어 프로세서(들)(310) 및 저장소(320)를 포함한다. 특히, 저장소(320)는 예를 들어 하드 드라이브에 대응할 수 있는 경계 실린더로서 일반적으로 도시된다. 그러나, 저장소(320)는 단순한 하드 드라이브 구성으로 제한되지 않는다는 것을 이해할 것이다. 대신, 저장소(320)는 하나 이상의 컴퓨팅 시스템 및 저장 위치에 대한 상이한 유형의 독립형 및/또는 분산 저장소를 포함할 수 있다.
- [0046] 일부 경우에, 저장소(320)는 비휘발성 저장소를 포함한다. 다른 경우에, 저장소는 비휘발성 및 휘발성 저장소의 임의의 조합을 포함한다. 저장소(320)는 또한, 예를 들어, 휘발성 메모리 및/또는 플래시 또는 다른 비휘발성 메모리를 포함할 수 있는 메모리를 포함할 수 있다.
- [0047] 일부 경우에, 메모리는 분할된 메모리 컨테이너(예를 들어, 새로운 컨테이너(315)) 및 기타 저장 구조(예를 들어, 컨테이너 메타데이터(322), 매핑 테이블(324), 컨테이너 이미지(326), 컨테이너 이미지 파일(예를 들어, .d11(328)) 및 보안 데이터(329))와 같은 분할된 메모리뿐만 아니라 컴퓨팅 시스템(300A 및 300B)을 위한 OS(운영 체제) 및 커널을 포함하는 실행 가능 코드 및 애플리케이션을 저장 및 구현하는 데 사용된다.
- [0048] 저장소는 또한 VTL0 메모리(380) 및 VTL1 메모리(390)를 포함하는 메모리 파티션을 포함할 수 있다. VTL0 메모리(380)는 사용자 모드를 구현하고 컨테이너의 프로비저닝을 개시하기 위해 시스템 OS에 의해 사용되는 제 1 기본 저장 도메인을 포함한다. VTL0 메모리는 또한 프로비저닝된 컨테이너(들)의 주소(들)를, 격리된 보안 메모리 엔클레이브를 포함하는 강화된/격리된 보안 저장 도메인 VTL1 메모리(390)의 대응하는 컨테이너 엔클레이브(들)에 매핑하기 위한 매핑 테이블(아래에서 더 자세히 설명됨)을 저장할 수 있다.
- [0049] 저장소(320)는 또한 하나 이상의 네트워크 연결(399)을 통해 컨테이너 및 심지어 원격 시스템(예컨대, 원격 키 프로비저닝 서비스(395) 및 원격 증명 서비스(396))을 프로비저닝, 모니터링, 로딩 또는 인터페이스하는 데 사용되는 다른 도시된 시스템 구성요소(예를 들어, 매핑(330), 컨테이너 증명 서비스(394), 컨테이너 관리 서비스(340), 메모리 관리자(360), 로더(370), 키 프로비저닝 서비스(397))를 인스턴스화하거나 구현하기 위한 실행 가능 코드 및 애플리케이션을 저장한다.
- [0050] 다양한 시스템 구성요소의 도시된 분리는 단지 개시된 실시예의 설명을 용이하게 하기 위해 제공되므로 달리 명시되지 않는 한 저장소 및 서로에 대한 시스템 구성요소의 임의의 물리적 경계 또는 관계를 부과하는 것으로 해

석되어서는 안 된다는 것이 이해될 것이다.

- [0051] 전술한 내용과 관련하여, 시스템의 사용자 모드 및 격리된 사용자 모드를 각각 구현하는 데 사용되는 상이한 도메인(VTL0 메모리(380)와 VTL1 메모리(390)) 사이에 부과되고 시행되는 물리적 및/또는 가상 경계가 있음을 유념한다.
- [0052] 분할된 도메인 사이에 설정된 하나 이상의 경계는 예를 들어 본 명세서에 설명된 바와 같이 컨테이너를 프로비저닝하는 동안 생성되고 시행될 수 있다. 일부 실시예에서, 이 시행은 하이퍼바이저에 의해 유지된다. 다른 실시예에서, 다른 관리 계층이 상이한 도메인에 대해 생성되는 파티션 및 메모리 격리 파티션을 시행하고 유지하는 데 사용된다. 일부 경우에, 컨테이너 관리 서비스(340)는 상이한 컨테이너 사이의 보안/격리 경계를 식별한다. 보안/격리 경계는 하드웨어 또는 하이퍼바이저(도시되지 않음)를 통해 시행된다.
- [0053] 제 1 보안 도메인(예를 들어, VTL0 메모리(380))의 단일 컨테이너와 제 2 보안 도메인(예를 들어, VTL1 메모리(390))의 대응하는 컨테이너 기반 엔클레이브 파티션만을 포함하는 2개의 상이한 메모리 파티션만이 도 3 및 도 4에 도시되어 있지만, 임의의 수의 상이한 파티션이 있을 수 있다는 것이 이해될 것이다. 이러한 파티션은 또한 동일한 및/또는 상이한 보안 도메인(예컨대, 비제한적으로, VBS의 가상 신뢰 레벨의 상이한 신뢰 레벨) 내에 설정된 컨테이너의 수량을 지원하기 위해 상이한 보안 도메인의 수량을 사용/구성할 수 있다.
- [0054] 추가로, 도시되지는 않았지만, 상이한 파티션은 또한 Intel의 SGX 엔클레이브 기술과 같은 하드웨어 분할뿐만 아니라 그 안에 포함된 컨테이너에 대한 하드웨어 및 가상 메모리 분할의 임의의 조합에 포괄적으로 또는 배타적으로 기반할 수 있다.
- [0055] 상이한 보안 도메인을 참조하면, 각각의 보안 도메인은 상이한 요구 및 선호도를 수용하고 컨테이너 기반 메모리 엔클레이브가 원하는 상이한 보안 메모리 솔루션(기존 및 아직 개발되지 않은 IoT, 데이터베이스, 엔터프라이즈 워크로드 및 저장 시나리오를 포함하지만 이에 제한되지 않음)에 대한 상이한 레벨의 보안 보장을 갖는 것을 용이하게 하도록 구성될 수 있다는 것이 이해될 것이다.
- [0056] 도시되지 않은 일부 예에서, 개시된 시스템은 1개, 2개 또는 2개 이상의 상이한 보안 도메인 내에서 컨테이너 기반 메모리 엔클레이브를 격리하며, 각각의 상이한 보안 도메인은 상이한 가상 보안 도메인(예를 들어, VTL1, VTL2, 등)을 포함하고, 기본 호스트 사용자 모드 보안 도메인(예를 들어, VTL0)과 상이한 보안 보장을 가지며, 하나 이상의 하드웨어 저장 장치(들)의 동일/공유 세트(예를 들어, 하나 이상의 드라이브(들), 디스크(들), 랙(들), 데이터 센터(들) 등의 동일/공유 세트) 내에서 구현 및 분배된다. 이러한 시나리오는, 예를 들어, 상이한 테넌트가 동일한 백엔드 컨테이너 저장 시스템 및 저장 장치를 사용하지만 각 테넌트가 상이한 보안 계약을 갖고 컨테이너 기반 메모리 엔클레이브를 프로비저닝하기 위해 상이한 컨테이너 이미지를 활용하는 다중 테넌트 실시예를 지원할 수 있다.
- [0057] 도시되지 않은 일부 예에서, 개시된 시스템은 1개, 2개 또는 2개 이상의 상이한 보안 도메인 내에서 컨테이너 기반 메모리 엔클레이브를 격리하며, 각각의 상이한 보안 도메인은 기본 호스트 사용자 모드 보안 도메인과 상이한 보안 보장을 갖고, 각각의 상이한 보안 도메인은 상이한/물리적으로 분리되고 공유되지 않은 하드웨어 저장 장치(예를 들어, 하나 이상의 드라이브(들), 디스크(들), 랙(들), 데이터 센터(들) 등의 상이한 세트) 상에서 물리적으로 격리된다. 이러한 실시예는 서로 다른 위치의 물리적 저장소를 사용하는 분산 네트워크에 특히 유용하다.
- [0058] 일부 경우에, 개시된 시스템은 기본 호스트 사용자 모드 보안 도메인과 상이한 동일한 보안 도메인 내에서 복수의 컨테이너 기반 메모리 엔클레이브를 격리하며, 여기서 복수의 컨테이너 기반 메모리 엔클레이브 각각은 공통 컨테이너 이미지로부터 모두 프로비저닝된다.
- [0059] 일부 대안 및/또는 추가 실시예에서, 적어도 2개 이상의 상이한 컨테이너 기반 메모리 엔클레이브는 적어도 2개의 상이한 대응하는 컨테이너 이미지로부터 프로비저닝된다.
- [0060] 도 3 내지 도 4에 반영된 바와 같이, 그리고 이전에 언급된 바와 같이, 개시된 시스템은 컨테이너 관리 서비스(340)를 포함한다. 이 컨테이너 관리 서비스(340)는 새로운 컨테이너(예를 들어, 새로운 컨테이너(315))를 프로비저닝하고 프로비저닝된 컨테이너와 그 보안 구성 및 메모리 영역을 모니터링하도록 구성된다.
- [0061] 컨테이너 증명 서비스(394)가 또한 시스템(300A/300B)에 의해 제공된다. 이 컨테이너 증명 서비스(394)는 일부 실시예에서 하나 이상의 원격 증명 서비스(들)(예를 들어, 원격 증명 서비스(396))와의 통신을 용이하게 하기 위한 인터페이스이다.

- [0062] 또한, 엔클레이브는 실행 중인 프로세스의 유형 자체를 쉽게 식별할 수 없고 엔클레이브를 호출하는 프로세스는 신뢰할 수 없는 별도의 도메인(예를 들어, VTL0)에 의해 제어되기 때문에, 원격 증명 서비스(396) 그리고 때때로 로컬 컨테이너 증명 서비스(394)는 엔클레이브를 조정하는 컨테이너 관리 서비스(340)에 대한 인터페이스를 제공하여 주어진 엔클레이브와 함께 사용되는 호스트 프로세스의 유형을 증명한다. 예를 들어, 엔클레이브가 프로세스 격리 컨테이너의 컨텍스트 내에서 실행 중인 경우, 증명 서비스(위에 설명된 플랫폼 증명 보고서 및 엔클레이브 증명 보고서와 함께)는 주어진 엔클레이브의 호스트 프로세스가 시스템의 나머지 부분과 격리되어 있다는 증명 가능한 진술을 할 수 있다.
- [0063] 이러한 방식으로, 원격 증명 서비스(396) 그리고 때때로 로컬 컨테이너 증명 서비스(394)는 위에서 설명된 바와 같이, 특히 도 1 및 도 2를 참조하여 설명된 바와 같이, 시스템 및/또는 컨테이너 기반 메모리 엔클레이브의 보안 보장을 증명하기 위해 시스템(300A/300B) 및/또는 엔클레이브 컨테이너에 의해 제공되는 증명 보고서를 검증/증명하도록 구성된다.
- [0064] 일부 경우에, 컨테이너 증명 서비스(394)는 시스템(300B)과 원격 증명 서비스(들)(396) 사이에서 정보(예를 들어, 증명 보고서, 증명 요청, 검증된 증명 보고서 등)를 단순히 전달한다.
- [0065] 다른 경우에, 컨테이너 증명 서비스(394)는 엔클레이브에 대해 수신된 각각의 증명 요청에 대해 원격 증명 서비스(들)와 인터페이스하지 않고, 증명 자격 증명서를 저장하여 컨테이너 기반 메모리 엔클레이브 보안 보장의 증명을 수행하도록 구성된다. 이러한 방식으로, 각 증명 요청에 대해 엔클레이브와 원격 증명 서비스 간의 종단 간 증명을 요구하지 않고, 컨테이너의 임시 특성을 수용하여 훨씬 더 효율적으로 증명을 수행할 수 있다. 이는 동일한 컨테이너 이미지로부터 모두 프로비저닝되는 복수의 컨테이너/엔클레이브에 대한 증명 보고서를 로컬로 검증하는 데 특히 효율적일 수 있다.
- [0066] 예를 들어, 일부 경우에, 컨테이너 증명 서비스(394)는 검증을 위해 전송되는 각각의 새로운 증명 보고서에 대해 엔클레이브 증명 보고서의 종단 간 검증을 요구하지 않고, 특정 컨테이너 기반 엔클레이브에 대한 보안 보장을 검증하거나 증명(예를 들어, 격리 보안 보장 및 프리미티브를 증명)하도록 구성된다.
- [0067] 이것은, 예를 들어, 컨테이너 증명 서비스(394)가 엔클레이브 증명 보고서를 검증하기 위해 원격 키 프로비저닝 서비스(395) 및/또는 원격 증명 서비스(396)로부터 수신된 AIK 또는 다른 자격 증명서를 포함하는 자격 증명서를 저장하고, 새로운 컨테이너 기반 엔클레이브 보고서가 원격 키 프로비저닝 서비스에 의해 이전에 검증된 컨테이너 기반 엔클레이브 보고서에 대응하거나 유사한 프로파일을 갖는 것으로 결정될 때 (이전 컨테이너 기반 엔클레이브 또는 새로운 컨테이너 기반 엔클레이브로부터) 새로운 엔클레이브 증명 보고서를 검증하는 데 사용할 수 있다.
- [0068] 새로운 컨테이너 기반 엔클레이브 보고서가 이전에 검증된 컨테이너 기반 엔클레이브 보고서에 대응하거나 유사한 프로파일을 갖는다는 결정은, 예를 들어, 새로운 컨테이너 기반 엔클레이브 보고서가 동일한 보안 도메인 및 프로비저닝 프로파일을 갖는(예를 들어, TPM 플랫폼 구성 레지스터(PCR)에서 유사한 시스템 부팅 체인을 가짐, 동일한 IDK를 가짐 및/또는 동일한 기본 인증 보증 키(EK)에 연결됨) 동일하거나 상이한 컨테이너 기반 엔클레이브의 이전 원격 증명으로부터 수신된 동일한 AIK 또는 다른 자격 증명서/보안 정보로 서명되었는지 확인함으로써 이루어질 수 있다.
- [0069] 다른 경우에, 이러한 결정은, 새로운 컨테이너 기반 엔클레이브 보고서가 이전에 프로비저닝된 컨테이너를 프로비저닝하고 원격 증명 서비스(396)에 의해 검증된 대응하는 엔클레이브 증명 보고서를 생성하는 데 사용된 동일한 보안 데이터(329)를 갖는 동일한 컨테이너 이미지로부터 프로비저닝된 새로운 컨테이너로부터 수신되었음을 결정함으로써 이루어질 수 있다.
- [0070] 일부 경우에, 이전/호스트 컨테이너 엔클레이브와 동일한 이미지에 의해 프로비저닝된 새로운/게스트 컨테이너 엔클레이브에 대한 증명 프로세스는 로컬 및/또는 원격 컨테이너 증명 서비스(394)뿐만 아니라 로컬 및/또는 원격 키 프로비저닝 서비스를 사용하여 시스템(300A/B)에 의해 수행되고, 다음과 같은 사항을 포함한다. (1) 호스트는 새로운/게스트 엔클레이브에 전송되는 키를 생성하거나(예를 들어, 키 프로비저닝 서비스(397)를 사용) 키를 획득하고(예를 들어, 원격 키 프로비저닝 서비스를 사용), (2) 새로운/게스트 엔클레이브는 증명 보고서를 생성하고, (3) 새로운/게스트 엔클레이브는 호스트와 Diffie-Hellman 키 교환 프로세스를 시작하고, (4) 새로운/게스트 엔클레이브는 (키 교환 데이터와 함께) 증명 보고서를 호스트 엔클레이브에 제시하고(예컨대, 컨테이너 증명 서비스(394)에 증명 보고서를 제시하며, 증명 보고서는 호스트 엔클레이브와 같은 이전 컨테이너 엔클레이브에 대해 이전에 검증된 증명 보고서에 대응하는 정보를 포함함), (5) 호스트는 Diffie-Hellman 키 교환 프로

세스를 시작하고, (6) 호스트는 동일한 컨테이너 이미지로부터 프로비저닝된 이전에 프로비저닝된 컨테이너의 이전 증명을 기반으로 새로운/게스트 증명 보고서를 컨테이너 증명 서비스(394)로 검증하고, (7) 호스트는 키 프로비저닝 서비스(397)에 의해 제공된 키 정보를 사용하여 새로운/게스트 엔클레이브에 키 교환 정보를 전송하고, (8) 호스트 및 게스트 엔클레이브는 Diffie-Hellman 키 교환을 완료하며, (10) 호스트는 키 프로비저닝 서비스(397)를 사용하여 공유 키로 게스트 엔클레이브에 전송될 키를 암호화한다.

[0071] 이러한 실시예에서, 이전에 프로비저닝되고 원격 증명 서비스에 의해 증명된 동일한 컨테이너 이미지로부터 프로비저닝된 새로운/게스트 엔클레이브의 엔클레이브 증명 보고서를 검증하기 위해 로컬 키 프로비저닝 서비스(397) 및 컨테이너 증명 서비스(394)를 이용하는 것이 가능하다.

[0072] 일부 실시예에서, 새로운 컨테이너(315)는 일단 증명되면 다른 컨테이너에 대한 보고서를 로컬로 생성하거나 다른 컨테이너에 증명을 위임할 수 있다. 일부 경우에, 예를 들어, 이러한 다른 위임 컨테이너는 수명이 더 길 수 있으므로 프로시처럼 작동한다. 이것은 예를 들어 주어진 호스트 시스템(300B)에 많은 임시 컨테이너의 존재를 가능하게 할 뿐만 아니라 컨테이너 및 위임 컨테이너의 상이한 계층 구조 내에서 증명의 내포화를 가능하게 한다.

[0073] 일부 경우에, 증명은 컨테이너의 속성(예를 들어, 컨테이너의 상이한 수명 기간, 컨테이너의 상이한 유형, 상이한 컨테이너 위치, 상이한 컨테이너 테넌트, 상이한 포함 콘텐츠, 상이한 컨테이너 위험 프로필, 상이한 컨테이너에 대한 상이한 서비스 품질 보증, 컨테이너 책임(예를 들어, 위임 또는 기본 컨테이너) 및/또는 임의의 다른 컨테이너 속성)에 기초하여 시스템 및/또는 원격 서비스의 상이한 계층/구성요소를 사용하여 상이하게 수행된다.

[0074] 예를 들어, 더 긴 수명 및/또는 더 높은 위험 레벨을 갖는 컨테이너가 하나 이상의 특정 원격 증명 서비스에서 증명되도록 요구될 수 있는 반면, 더 짧은 수명 및/또는 더 낮은 위험 레벨을 갖는 상이한 컨테이너가 시스템/컨테이너에 더 가까운 상이한 원격 증명 서비스에서 증명되거나 심지어 로컬 컨테이너 증명 서비스(즉, 394)에서 증명되도록 요구될 수 있다.

[0075] 따라서, 일부 경우에, 시스템은 컨테이너 증명에 액세스하고 해당 증명을 생성한 대응하는 컨테이너의 결정된 수명 기간(또는 컨테이너의 임의의 다른 속성(들))에 기초하여 증명을 적절한 증명 서비스로 결정론적으로 라우팅할 수 있다. 예를 들어, 후속 컨테이너 증명을 수신하면, 시스템은 후속 컨테이너 증명과 관련된 특정 컨테이너의 속성을 결정할 수 있으며, 그런 다음 특정 컨테이너의 속성에 기초하여 후속 컨테이너 증명을 원격 서비스로 결정론적으로 라우팅하거나, 대안적으로 특정 컨테이너의 속성에 적어도 부분적으로 기초하여 후속 컨테이너 증명을 원격 서비스로 라우팅하지 않고 증명 보고서를 생성한다. 속성은 컨테이너의 수명을 포함할 수 있기 때문에, 이는 또한 시스템이 제 1 컨테이너 세트가 제 2 컨테이너 세트보다 상대적으로 더 긴 수명을 가지고 있다고 분석하는 것을 포함할 수 있으며, 여기서 제 1 컨테이너 세트와 연관된 하나 이상의 컨테이너 증명이 원격 서비스로 라우팅되는 반면, 예를 들어, 제 1 컨테이너 세트보다 상대적으로 더 짧은 수명을 갖는 후속 컨테이너를 포함하는 제 2 컨테이너 세트 각각에 대해 로컬 증명으로 하나 이상의 증명 보고서를 생성하며 제 2 컨테이너 세트를 원격 서비스로 라우팅하지 않는다.

[0076] 전술한 내용과 관련하여, 컨테이너 증명은 증명 요청(예를 들어, 엔클레이브 증명 요청, 컨테이너 증명 요청, 증명 보고서, 및/또는 다른 증명)일 수 있다는 것이 이해될 것이다. 추가로, 증명을 라우팅하는 데 사용되는 속성은 위에서 설명된 바와 같이 컨테이너 수명 또는 임의의 다른 컨테이너/엔클레이브 속성일 수 있다는 것이 또한 이해될 것이다.

[0077] 컨테이너 기반 메모리 엔클레이브를 갖는 컨테이너를 프로비저닝하기 위한 흐름이 이제 도 5의 흐름도를 참조하여 설명될 것이다. 도시된 바와 같이, 도 5의 흐름도(500)는 개시된 실시예에 따라 컨테이너를 프로비저닝하기 위한 다양한 방법과 연관된 복수의 동작(동작 510, 동작 520 및 동작 530)을 포함한다.

[0078] 도시된 제 1 동작(동작 510)은 시스템(예컨대, 시스템(300A))이 제 1 보안 도메인(예를 들어, VTLO)에서 실행되는 호스트 시스템에서 사용하는 특정 보안 구성을 증명하는 데 사용되는 보안 구성요소(예컨대, .d11(328)와 같은 라이브러리 파일)를 갖는 컨테이너 이미지(예컨대, 컨테이너 이미지(326))를 식별하는 것을 포함한다. 일부 경우에, 컨테이너 이미지는 로더(370)에 의해 식별되고 메모리(예를 들어, VTLO 메모리(380))에 로드된다. 메모리 관리자(360)는 컨테이너를 로드할 위치를 식별하고 프로비저닝 및 해제된 컨테이너를 추적하여 메모리를 효과적으로 관리한다.

[0079] 컨테이너 이미지의 보안 구성요소는 다양한 형태를 취할 수 있으며 라이브러리 파일일 필요는 없음을 유념한다.

대신, 보안 구성요소는 증명 가능하고 엔클레이브 증명 요청을 생성하는 데 사용되거나 엔클레이브 증명 요청에 포함된 정보를 갖는 구성 정보를 가진 임의의 데이터(예를 들어, 데이터 블록, 데이터 구조, 비정형 데이터 등)를 포함 및 구성될 수 있다. 일부 경우에, 보안 구성요소는 컨테이너 이미지(326)의 보안 구성 프로파일 또는 그것이 로드되는 엔클레이브 메모리의 격리 속성을 식별하는 정보를 포함한다.

[0080] 보안 구성요소는 예를 들어 호스트 시스템의 엔클레이브 메모리 파티션 내의 (예를 들어, VTL1 메모리(390) 내의) 위치를 포함할 수 있는 호스트 시스템의 메모리 위치에 로드될 것이다. 필수는 아니지만, 컨테이너 이미지(326)는 키 프로비저닝 서비스(397)에 의해 제공되는 하나 이상의 키로 해독되는 실행 가능 코드 또는 다른 보안 데이터(329)를 포함할 수도 있어, 프로비저닝된 후 실행 가능 코드는 호스트 시스템의 격리된 엔클레이브 메모리 파티션(예를 들어, VTL1) 내에서 자동으로 인스턴스화된다.

[0081] 특히, 보안 구성요소로부터의 정보는 컨테이너가 완전히 프로비저닝된 후에 대응하는 컨테이너 기반 메모리 엔클레이브의 보안 보장을 검증하는 데 사용될 수 있다. 또한, 일부 경우에, 보안 구성요소는 엔클레이브 메모리 파티션(예를 들어, VTL1 메모리(390))에 로드되어 기본 저장소/보안 도메인(예를 들어, VTL0 메모리(380))에 구현된 구성요소/프로세스에 의한 원치 않는 액세스로부터 보안 구성요소를 더욱 격리 및 보호한다는 점을 유념한다.

[0082] 새로운 컨테이너를 프로비저닝하는 초기 단계 동안, 보안 구성요소의 실제/확인 가능한 메모리 주소는 컨테이너 이미지에 의해 지정되지 않았기 때문에 아직 알려지지 않았다. 따라서, 매핑 테이블 및 컨테이너 메타데이터를 업데이트하는 데 사용되는 메모리 주소는, 컨테이너 이미지(326)로부터 새로운 컨테이너(315)를 프로비저닝할 때, 호스트 시스템의 실제/확인 가능한 주소가 아니라 보안 구성요소에 대한 의도된 메모리 주소(잘못된 플레이스 홀더 주소임)일 뿐이다. 이것이 이 시점에서 의도된 메모리 주소로만 언급되는 이유이다.

[0083] 보안 구성요소에 대한 이 의도된 메모리 주소는 매핑 테이블(324) 및/또는 컨테이너 메타데이터와 같은 데이터 구조 내에 컨테이너의 초기 프로비저닝 단계 동안 기록된다.

[0084] 매핑 테이블(324)은 다양한 형태를 취할 수 있고 단일 독립형 데이터 구조 또는 복수의 분산 테이블/구조일 수 있음을 유념한다. 일부 경우에, 매핑 테이블(324)은 시스템 내의 상이한 컨테이너 및 이들의 할당된 메모리 위치를 식별하는 매핑(330)에 의해 생성 및 유지된다. 일단 생성되면, 매핑 테이블(324)은 컨테이너 관리 서비스(340)에 의해 참조되고 메모리 관리자(360)는 새로운 컨테이너(315) 및 다른 프로비저닝된 컨테이너와의 인터페이스 및 모니터링을 형성한다.

[0085] 컨테이너를 프로비저닝한 후, 컨테이너는 컨테이너 메타데이터 및/또는 매핑 테이블의 보안 구성요소에 대한 주소/포인터를 통해 보안 구성요소에 액세스하여 컨테이너가 엔클레이브 증명 보고서를 생성하는 데 사용할 보안 구성요소 정보를 식별한다. 따라서, 보안 구성요소에 대한 주소가 올바른 주소로 업데이트되지 않으면, 컨테이너는 보안 구성요소의 정보에 액세스할 수 없어 컨테이너/엔클레이브에 대한 적절한 보안 보장을 갖는 엔클레이브 증명 보고서를 생성할 수 없으며 이러한 보안 보장을 증명하려는 임의의/모든 시도는 실패한다.

[0086] 또한, 일부 경우에, 페이징 작업과 같은 컴퓨팅 프로세스가 컨테이너에 대해 할당된 메모리 위치를 변경하는 결과를 초래할 수 있다. 이러한 인스턴스를 처리하기 위해, 매핑 테이블(324)은 할당된 메모리의 변경 사항을 반영하도록 추가 업데이트될 수 있다. 추가적으로, 컨테이너는 증명 서비스(394)에서 증명될 수 있거나, 컨테이너는 컨테이너의 무결성이 컨테이너의 코드 무결성(CI) 정책과 일치하는지 확인하는 호스트 메모리 엔클레이브(예를 들어, VTL1)에 의해 검증될 수 있어 메모리 엔클레이브는 예를 들어 원격 증명 서비스(396)로부터 원격 증명을 얻을 필요 없이 새로운 증명 보고서를 발행할 수 있다. 또한, 이러한 경우에, 호스트 메모리 엔클레이브는 원격 증명 서비스(396)에 대한 위임자 역할을 한다. 일부 실시예에서, 이 증명은 매핑 테이블에서 메타데이터로 표시될 수 있으며, 이것은 또한 메모리 관리자에 의해 추적되어 이 메타데이터는 페이징 작업에 의해 "더티(dirty)"로 표시될 수 있으므로 증명은 이후에 매번 "새로 고침"되어야 한다. 따라서, 이 실시예에서, 메모리 범위는 증명 보고서에 포함될 것이다.

[0087] 따라서, 일부 실시예에서, 컨테이너가 완전히 프로비저닝되고 보안 구성요소가 최종적으로 호스트 시스템의 메모리에, 예컨대, 호스트 시스템의 엔클레이브 메모리 파티션 내에 로드되면 참조된 컨테이너의 보안 구성요소에 대한 메모리 주소를 업데이트하는 것이 중요하다.

[0088] 동작(520)에 의해 반영된 바와 같이, 시스템(300A)은 호스트 시스템의 사용자 모드를 구현하는 데 사용되는 제 1 보안 도메인(예를 들어, VTL0 메모리(380))과 상이한 제 2 보안 도메인(예를 들어, VTL1 메모리(390))에서 새로운 컨테이너(315)(새로운 컨테이너 엔클레이브를 포함함)를 인스턴스화하기 위해 컨테이너 이미지(326)를 사

용한다. 특히, 새로운 컨테이너는 증명 가능한 보안 구성요소(예를 들어, .d11(328))에 지정된 구성을 기반으로 호스트 시스템의 격리된 엔클레이브 메모리 파티션을 사용하도록 구성되어 컨테이너 기반 엔클레이브로서 인스턴스화되고, 결과적으로 컨테이너는 격리된 엔클레이브 메모리를 참조하고 격리된 엔클레이브 메모리에 저장된 데이터를 사용하여 호스트 시스템의 기본 사용자 모드 메모리로부터 격리된 프로세스를 수행하도록 구성된다.

[0089] 일단 프로비저닝되면, 새로운 컨테이너(315)는 호스트 시스템의 특정 메모리 주소를 갖는 보안 구성요소의 사본을 포함하며, 보안 구성요소는 새로운 컨테이너가 참조하는 호스트 시스템의 격리된 엔클레이브 메모리 파티션에 저장된다. 그러나, 위에서 언급한 바와 같이, 새로운 컨테이너(315)에 의해 제공되거나 매핑되는 현재 주소는 매핑 테이블(324) 및/또는 컨테이너 메타데이터에 반영된 바와 같이 초기 컨테이너 이미지를 기반으로 하므로 컨테이너로부터 호스트 시스템에 있는 보안 구성요소의 메모리 주소로의 의도된 링크일 뿐이다(컨테이너 이미지가 새로운 컨테이너를 프로비저닝하는 데 처음 사용될 때 메모리 주소를 알 수 없었기 때문에).

[0090] 이 문제는 다수의 상이한 컨테이너를 프로비저닝하기 위한 단일 컨테이너 이미지의 사용 및 새로운 컨테이너가 처음 프로비저닝되는 VTL0 메모리가 아닌 호스트의 엔클레이브 메모리 파티션에 있는 컨테이너에 대한 보안 구성요소의 저장/격리와 관련하여 특히 두드러진다.

[0091] 이 문제를 해결하기 위해, 시스템은 의도된 링크(예컨대, 보안 구성요소의 기록된/의도된 물리적 메모리 주소)에 액세스하고 수정하여 새로운 컨테이너와 엔클레이브 메모리 파티션의 메모리 주소 간의 매핑된 링크가 컨테이너를 엔클레이브 메모리 파티션의 호스트 시스템의 정확한 메모리 주소에 확인 가능하게 링크할 것이다(동작 530).

[0092] 이 수정은 또한 보안 구성요소가 실제로 저장된 호스트 시스템의 실제 메모리 주소로 새로운 컨테이너의 보안 구성요소의 의도된 메모리 주소를 덮어쓰는 단계, 보안 구성요소가 실제로 저장된 호스트 시스템의 실제 메모리 주소로부터 링크를 생성하는 단계, 매핑 테이블(324) 내의 보안 구성요소와 연관된 의도된 주소를 실제/확인 가능한 주소로 변경하는 단계, 보안 구성요소의 의도된 메모리 주소를 갖는 매핑 테이블(324)을 보안 구성요소의 메모리 주소에 대해 정확한 주소를 갖는 새로운 매핑 테이블(324)로 덮어쓰거나 교체하는 단계, 매핑 테이블(324) 내의 링크 또는 새로운 포인터를 원래의 의도된 링크/주소로부터 엔클레이브 메모리 파티션의 보안 구성요소의 실제 확인 가능한 물리적 주소로 생성하는 단계 중 하나 이상을 포함할 수 있다.

[0093] 수정은 또한 매핑 테이블의 생성 및 엔클레이브 메모리 파티션의 보안 구성요소의 메모리 주소를 매핑 테이블에 추가하기 위해 새로운 컨테이너와 연관된 메타데이터의 사용을 포함하거나 따를 수 있다.

[0094] 이 메모리 주소가 보안 구성요소에 대한 정확하고/확인 가능한 메모리 위치를 가리키도록 매핑 테이블에서 변경, 업데이트 또는 수정되면, 컨테이너는 보안 구성요소에 액세스하고 보안 구성요소에 기초한 정보로 증명 보고서를 생성할 수 있다. 일부 실시예에서, 증명 보고서는 컨테이너 관리 서비스(340) 또는 신뢰 당사자와 공유된다는 점을 유념한다(동작 540). 일부 실시예에서, VTL1은 새로운 컨테이너(315)의 상태 및 시스템(300B)의 상태를 모니터링할 수 있다. 컨테이너 관리 서비스(340)는 또한 일부 경우에 컨테이너 상태 및 시스템 상태를 모니터링할 수 있다. 컨테이너 상태의 모니터링은 증명 요청 액세스 및 평가, 증명 보고서 평가, 및/또는 컨테이너에 대한 업데이트 및 쓰기 요청 모니터링 및/또는 컨테이너의 엔클레이브 메모리의 주소 재지정/재할당을 포함할 수 있다.

[0095] 주어진 새로운 컨테이너(315) 또는 시스템(300B)이 증명에 실패하거나 손상된 것으로 보이는 경우, 컨테이너 관리 서비스(340)는 새로운 컨테이너(315)를 해체하도록 요청할 수 있고/있거나 그러한 손상을 독립적으로 식별하는 경우 이러한 조치를 사전에 취할 수 있다(동작 550).

[0096] 일부 실시예에서, VTL1은 매핑 테이블(324)을 모니터링할 수 있으며, 증명에 실패하거나 손상되는 경우, VTL1 메모리를 새로운 컨테이너(315) 및 유사한 엔티티로 매핑하는 업데이트를 방지할 수도 있다. 일부 실시예에서, 추가 통지가 VTL1 또는 다른 구성요소로부터 원격 증명 서비스(396)와 같은 원격 관리 엔티티로 전송될 수 있다.

[0097] 도 6은 개시된 실시예에 따라 컨테이너의 보안 보장을 증명하기 위한 방법과 연관된 복수의 동작(동작 610, 동작 620, 동작 630 및 동작 640)을 갖는 흐름도(600)의 일 실시예를 도시한다. 이 흐름도(600)는 도 4에 도시된 시스템(300B)에 대응한다.

[0098] 처음에, 시스템(300B) 예시에 반영되고 이전에 설명된 바와 같이, 원격 증명 서비스(396)는 시스템 증명 보고서를 검증하고 개별 엔클레이브/컨테이너 보고서를 검증하는 데 사용될 수 있다. 증명 프로세스에 사용되는 키 및 기타 자격 증명서는 원격 키 프로비저닝 서비스(395)에 의해 제공될 수도 있다. 그러나, 추가로 설명된 바와 같

이, 임시 컨테이너, 특히 동일한 컨테이너 이미지로부터 프로비저닝된 컨테이너에 대한 중단 간 증명 및 키 프로비저닝은 자원 낭비를 나타낸다.

[0099] 현재 시스템은 로컬 키 프로비저닝 서비스(397) 및 로컬 컨테이너 증명 서비스(394)를 제공함으로써 이러한 비효율성의 일부를 해결하며, 이들은 원격 서비스와 관련된 자격 증명서를 포함 및/또는 유지하지만, 새로운 컨테이너가 컨테이너 기반 메모리 엔클레이브를 갖는 이전 및/또는 기존 컨테이너에 대한 엔클레이브 증명 보고서를 프로비저닝 및/또는 증명하는 데 사용되는 이전에 사용된 컨테이너 이미지를 기반으로 하는 경우, 시스템(300B) 외부의 원격 서비스(예컨대, 중단 간 프로비저닝 및/또는 증명에 필요함)에 새로운 컨테이너에 대한 프로비저닝 또는 증명 요청을 전달하지 않고, 프로비저닝 및/또는 증명 동안 컨테이너에 키를 직접 제공하여 프로비저닝된 컨테이너의 보안 보장을 증명하는 데 사용될 수 있다.

[0100] 도 6에 도시된 바와 같이, 시스템에 의해 구현된 제 1 동작은 호스트 시스템 제 1 보안 도메인(예를 들어, VTL0)과 상이한 보안 도메인(예를 들어, VTL1 메모리, 엔클레이브 메모리를 포함함)에 상주하는 하나 이상의 컨테이너를 식별하는 동작이다(동작 610). 특히, 이 동작에서, 각각의 컨테이너는 호스트 시스템의 엔클레이브 메모리를 참조하고 엔클레이브 메모리에 대한 특정 보안 구성을 증명하기 위해 보안 구성요소를 사용하도록 구성된다. 이러한 보안 구성요소는 또한 엔클레이브 메모리와 같은 호스트 시스템의 특정 메모리 주소에 저장되고 하나 이상의 컨테이너에서 보안 구성요소에 대한 하나 이상의 참조에 의해 링크된다.

[0101] 다음 동작은 시스템이 보안 구성요소에 의해 획득되거나 보안 구성요소를 포함하는 정보를 기반으로 하나 이상의 컨테이너의 보안 구성의 보안 보장을 검증하는 단일 증명 보고서를 원격 증명 서비스로부터 획득하는 것을 포함한다(동작 620). 이것은 예를 들어 시스템이 특정 컨테이너 이미지에 의해 프로비저닝되고 호스트 시스템의 엔클레이브 메모리를 사용하며 증명 보고서를 생성하기 위한 보안 구성요소를 사용하는 제 1 컨테이너에서 컨테이너 증명 요청을 원격 증명 서비스에 제출하는 것에 응답하여 발생할 수 있다.

[0102] 다음으로, 시스템은 원격 증명 서비스로부터 단일 증명 보고서를 수신한 후 하나 이상의 컨테이너 중 특정 컨테이너에 대한 보안 보장을 증명하기 위한 새로운 요청을 수신한다(동작 630). 이것은 예를 들어 새로운/제 2 컨테이너(제 1 컨테이너를 프로비저닝하는 데 사용된 동일한 컨테이너 이미지로부터 프로비저닝됨)가 새로운 증명 요청을 생성하고 제출하는 것에 응답하여 발생할 수 있다. 대안적으로, 특정 컨테이너는 추가 증명 요청을 제출하거나 이전에 제출된 증명 요청을 재전송하는 제 1 컨테이너일 수 있다. 일부 실시예에서, 이 증명 요청은 컨테이너에 대한 것이고, 일부 실시예에서, 이 증명 요청은 컨테이너 및 해당 컨테이너와 연관된 엔클레이브에 대한 것임을 유념한다.

[0103] 일부 실시예에서, 로컬 컨테이너 증명 서비스(394)는 이전 증명 요청에 대한 응답으로 새로운 요청을 가로채고 새로운 증명 요청이 제 1/단일 증명 보고서를 검증하는 데 사용되었던 원격 증명 서비스로 전송되는 것을 방지한다. 일부 실시예에서, 원격 증명 서비스에 대한 네트워크 연결이 실패하거나 시간 초과되거나 신뢰할 수 없으므로, 이전 증명 요청에 대한 응답으로 제 1/단일 증명 보고서를 검증하는 데 사용되었던 원격 증명 서비스로 전송된 새로운 요청은 의도된 목적지에 도달하지 못한다.

[0104] 대신, 시스템(예를 들어, 로컬 컨테이너 증명 서비스(394))은 (원격 증명 서비스로 확인하지 않고) 독립적으로 특정 컨테이너의 보안 구성의 보안 보장을 검증 및/또는 증명한다(동작 640). 이 동작은, 예를 들어, 특정 컨테이너가 단일 증명 보고서로 검증된 증명 보고서를 생성한 이전 컨테이너를 프로비저닝하는 데 사용된 동일한 컨테이너 이미지로부터 프로비저닝되고 새로운 증명 요청을 수신하기 전에 수신된 이전 증명 보고서를 기반으로 한다고 결정하는 것에 응답하여 수행된다.

[0105] 일부 경우에, 동작(630 및 640)은 상이한 컨테이너 증명 요청에 대해 반복되어, 시스템은 호스트 시스템의 메모리 주소에 저장된 동일한 보안 구성요소에 대한 참조를 각각 포함하는 복수의 상이한 컨테이너에 대한 보안 보장을 증명하며, 이들은 모두 동일한 컨테이너 이미지로부터 프로비저닝되고 동일한 컨테이너 이미지로부터 프로비저닝된 다른 컨테이너에 대해 생성된 이전에 수신/서명된 증명 보고서를 통한 보안 보장의 이전 검증을 기반으로 한다.

[0106] 일부 실시예에서, 증명 서비스는 여러 계층의 캐싱으로 계층화될 수 있으며, 예를 들어, 한 계층은 컨테이너 자체에서 실행될 수 있고, 다른 계층은 호스트에서 실행될 수 있으며, 제 3 계층은 원격 증명 서비스(396)와 같은 원격 서비스에서 실행될 수 있는 식이다. 이 경우, 컨테이너 이미지의 검증은 다른 계층으로 보고(또는 전파)되고, 그런 다음 "캐싱"되어 증명 서비스의 고가용성을 보장할 수 있다.

[0107] 전술한 내용과 관련하여, 상이한 계층 내의 상이한 증명의 내포화는 또한 각 계층에서 구조화된 및/또는 결정론

적 증명을 포함할 수 있다는 것이 이해될 것이다. 예를 들어, 각 계층의 증명에 대해, 시스템은 증명할 증명 보고서를 생성하는 컨테이너의 속성을 독립적으로 식별할 수 있다. 이러한 속성을 기반으로 하여, 해당 계층 레벨에서 증명을 담당하는 구성요소/서비스(예를 들어, 컨테이너, 위임 컨테이너, 호스트, 원격 서비스 등)가 동일한 계층 내에서 증명을 수행할 수 있거나 증명 요청을 다른 계층/증명 서비스 또는 구성요소로 범프할 수 있다.

[0108] 전술한 실시예는 종래의 메모리 분할 기술과 관련된 많은 단점을 극복하기 위해 사용될 수 있고, 확장 가능하고 효율적인 방식으로 컨테이너 기반 메모리 엔클레이브를 제공하기 위해 컨테이너의 계층화된 격리를 활용하는 방식으로 사용될 수 있다는 것이 이해될 것이다.

[0109] 본 명세서에 설명된 방법은 위에서 설명된 것과 청구된 것과 같은 컴퓨팅 시스템에 의해 실행될 수 있으며, 컴퓨팅 시스템은 하나 이상의 프로세서(예를 들어, 하드웨어 프로세서) 및 컴퓨터 판독 가능 매체를 포함하며, 컴퓨터 판독 가능 매체는 컴퓨터 실행 가능 명령어를 저장하는 컴퓨터 메모리 및/또는 하나 이상의 다른 하드웨어 저장 장치(들)를 포함하며, 컴퓨터 실행 가능 명령어는 하드웨어 프로세서에 의해 실행 가능하고 하드웨어 프로세서에 의해 실행될 때 대응하는 컴퓨팅 시스템이 본 명세서에 설명된 방법에 언급된 동작의 실행을 포함하는 다양한 기능을 수행하게 하는 것이 이해될 것이다.

[0110] 본 발명의 실시예는 이하에서 더 상세히 논의되는 바와 같이 컴퓨터 하드웨어를 포함하는 특수 목적 또는 범용 컴퓨터를 포함하거나 이용할 수 있다. 본 발명의 범위 내의 실시예는 또한 컴퓨터 실행 가능 명령어 및/또는 데이터 구조를 운반하거나 저장하기 위한 물리적 및 다른 컴퓨터 판독 가능 매체를 포함할 수 있다. 이러한 컴퓨터 판독 가능 매체는 범용 또는 특수 목적 컴퓨터 시스템에 의해 액세스될 수 있는 임의의 사용 가능한 매체일 수 있다. 컴퓨터 실행 가능 명령어를 저장하는 컴퓨터 판독 가능 매체는 물리적 저장 매체이다. 컴퓨터 실행 가능 명령어를 운반하는 컴퓨터 판독 가능 매체는 전송 매체이다. 따라서, 제한이 아닌 예로써, 본 발명의 실시예는 물리적 컴퓨터 판독 가능 저장 매체 및 전송 컴퓨터 판독 가능 매체와 같은, 적어도 2개의 명백하게 상이한 종류의 컴퓨터 판독 가능 매체를 포함할 수 있다.

[0111] 물리적 컴퓨터 판독 가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 저장 장치(예컨대, CD, DVD 등), 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 컴퓨터 실행 가능 명령어 또는 데이터 구조의 형태로 원하는 프로그램 코드 수단을 저장하는 데 사용할 수 있고 범용 또는 특수 목적 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다.

[0112] "네트워크"는 컴퓨터 시스템 및/또는 모듈 및/또는 다른 전자 장치 간의 데이터 전송을 가능하게 하는 하나 이상의 데이터 링크로 정의된다. 정보가 네트워크 또는 다른 통신 연결(유선, 무선, 또는 유선과 무선의 조합)을 통해 컴퓨터에 전송되거나 제공되는 경우, 컴퓨터는 해당 연결을 전송 매체로 적절하게 간주할 수 있다. 전송 매체는 컴퓨터 실행 가능 명령어 또는 데이터 구조의 형태로 원하는 프로그램 코드 수단을 운반 또는 전송하는 데 사용될 수 있고 범용 또는 특수 목적 컴퓨터에 의해 액세스될 수 있는 네트워크 및/또는 데이터 링크를 포함할 수 있다. 위의 조합이 또한 컴퓨터 판독 가능 매체의 범위 내에 포함된다.

[0113] 또한, 다양한 컴퓨터 시스템 구성요소에 도달하면, 컴퓨터 실행 가능 명령어 또는 데이터 구조 형태의 프로그램 코드 수단이 전송 컴퓨터 판독 가능 매체로부터 물리적 컴퓨터 판독 가능 저장 매체로(또는 그 반대로) 자동으로 전송될 수 있다. 예를 들어, 네트워크 또는 데이터 링크를 통해 수신된 컴퓨터 실행 가능 명령어 또는 데이터 구조는 네트워크 인터페이스 모듈(예를 들어, "NIC") 내의 RAM에 버퍼링될 수 있고, 그런 다음 결국 컴퓨터 시스템 RAM 및/또는 컴퓨터 시스템에서의 덜 휘발성인 컴퓨터 판독 가능 물리적 저장 매체로 전송될 수 있다. 따라서, 컴퓨터 판독 가능 물리적 저장 매체는 전송 매체를 또한 (또는 심지어 주로) 사용하는 컴퓨터 시스템 구성요소에 포함될 수 있다.

[0114] 컴퓨터 실행 가능 명령어는, 예를 들어, 범용 컴퓨터, 특수 목적 컴퓨터, 또는 특수 목적 처리 장치 하에 특정 기능 또는 기능 그룹을 수행하게 하는 명령어 및 데이터를 포함한다. 컴퓨터 실행 가능 명령어는, 예를 들어, 바이너리, 어셈블리 언어와 같은 중간 포맷 명령어 또는 소스 코드일 수 있다. 본 주제가 구조적 특징 및/또는 방법론적 동작에 특유한 언어로 설명되었지만, 첨부된 청구범위에 정의된 주제는 반드시 위에서 설명된 특징 또는 동작으로 제한될 필요는 없음을 이해해야 한다. 오히려, 설명된 특징 및 동작은 청구범위를 구현하는 예시적인 형태로 개시된다.

[0115] 당업자라면, 본 발명이 퍼스널 컴퓨터, 데스크톱 컴퓨터, 랩톱 컴퓨터, 메시지 프로세서, 휴대용 장치, 다중 프로세서 시스템, 마이크로프로세서 기반 또는 프로그램 가능 가전 제품, 네트워크 PC, 미니 컴퓨터, 메인프레임 컴퓨터, 이동 전화기, PDA, 호출기, 라우터, 스위치 등을 포함하는 많은 유형의 컴퓨터 시스템 구성을 갖는 네

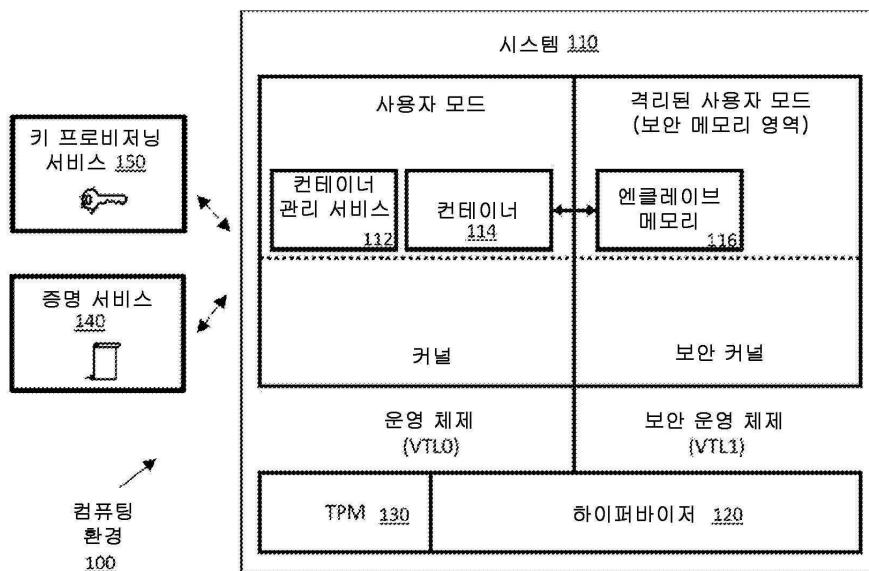
트위크 컴퓨팅 환경에서 실시될 수 있음을 이해할 것이다. 본 발명은 또한 네트워크를 통해 연결된 (유선 데이터 링크, 무선 데이터 링크 또는 유선 데이터 링크와 무선 데이터 링크의 조합에 의해) 로컬 및 원격 컴퓨터 시스템 모두가 작업을 수행하는 분산 시스템 환경에서 실시될 수 있다. 분산 시스템 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 양자 모두에 위치할 수 있다.

[0116] 대안적으로, 또는 추가로, 본 명세서에 설명된 기능은 하나 이상의 하드웨어 로직 구성요소에 의해 적어도 부분적으로 수행될 수 있다. 예를 들어, 비제한적으로, 사용될 수 있는 예시적인 유형의 하드웨어 로직 구성요소는 필드 프로그램 가능 게이트 어레이(Field-Programmable Gate Array; FPGA), 프로그램 지정 집적 회로(Program-Specific Integrated Circuit; ASIC), 프로그램 지정 표준 제품(Program-Specific Standard Product; ASSP), 시스템 온 칩 시스템(System-On-a-chip; SOC), 복합 프로그램 가능 로직 디바이스(Complex Programmable Logic Device; CPLD) 등을 포함한다.

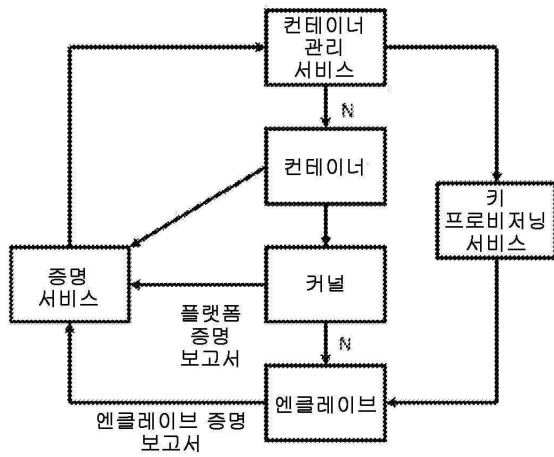
[0117] 본 발명은 본 발명의 사상 또는 특징을 벗어나지 않고 다른 특정한 형태로 구현될 수 있다. 설명된 실시예는 모든 면에서 제한이 아닌 단지 예시적인 것으로서 간주되어야 한다. 따라서, 본 발명의 범위는 전술한 설명보다는 첨부된 청구범위에 의해 나타난다. 청구범위의 등가적인 범위 및 의미에 속하는 모든 변경은 본 발명의 범위 내에 포함되어야 한다.

도면

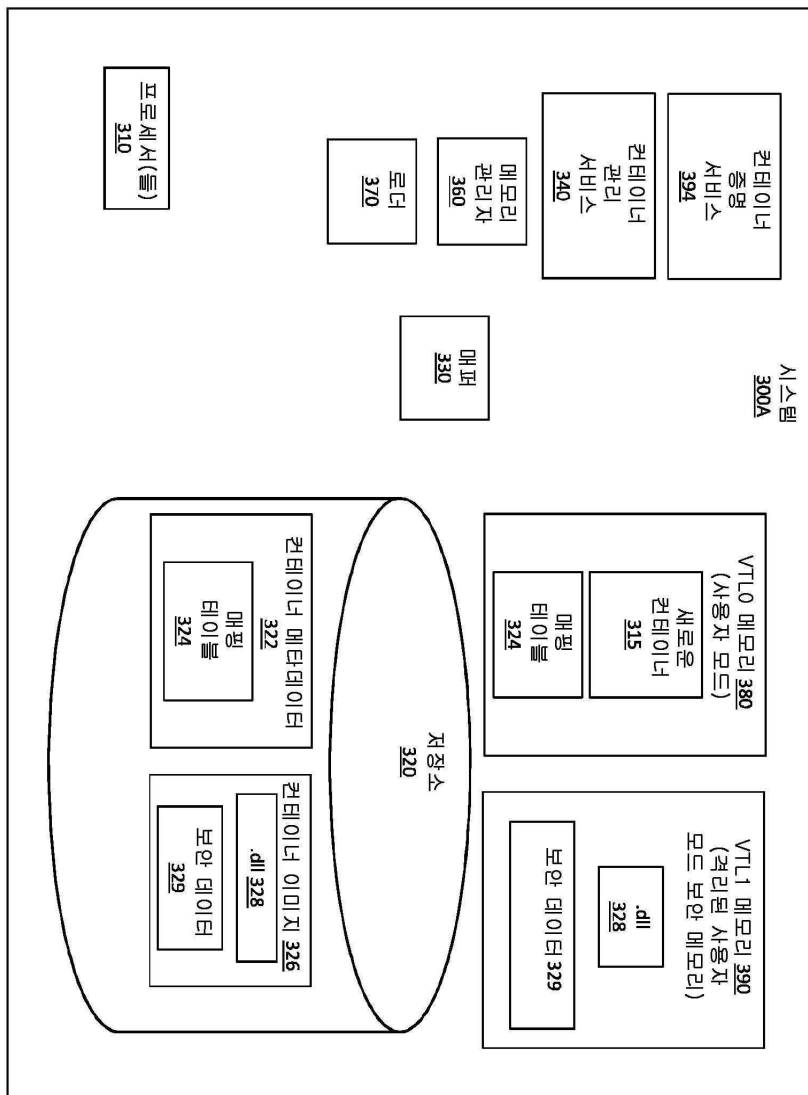
도면1



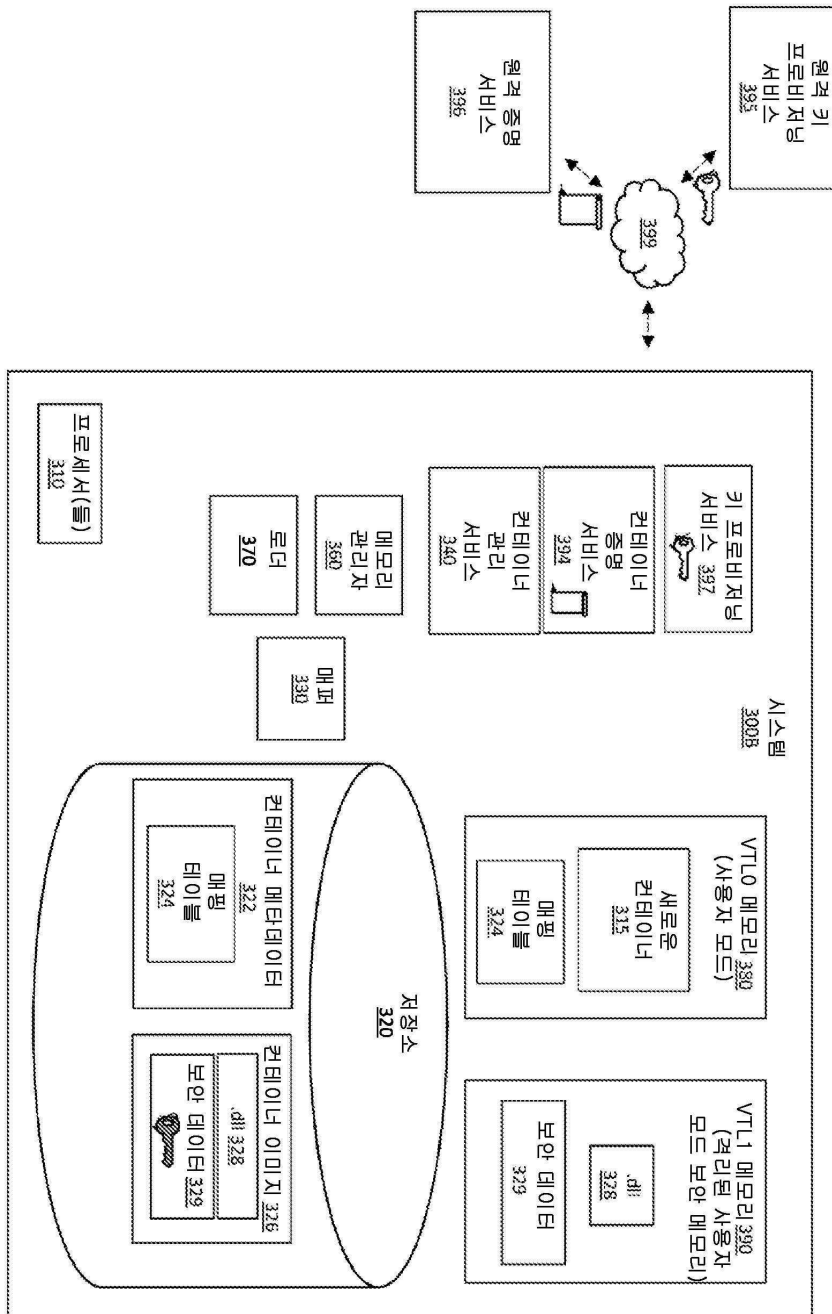
도면2



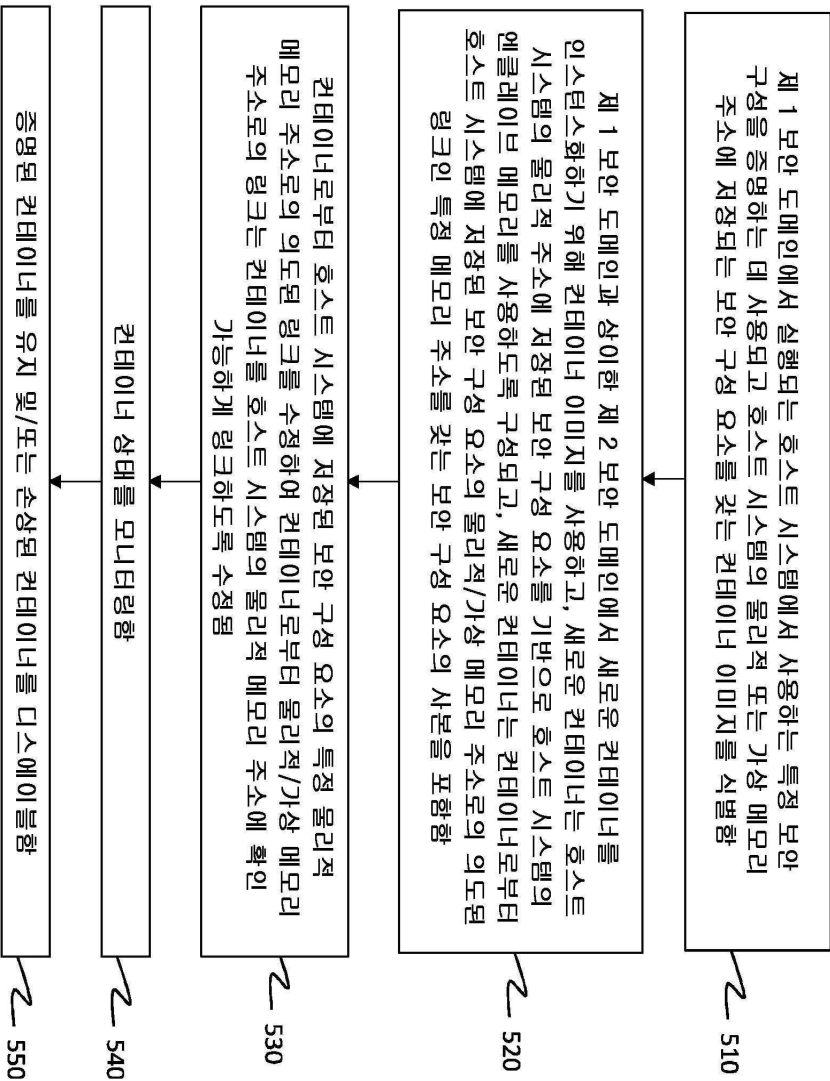
도면3



도면4

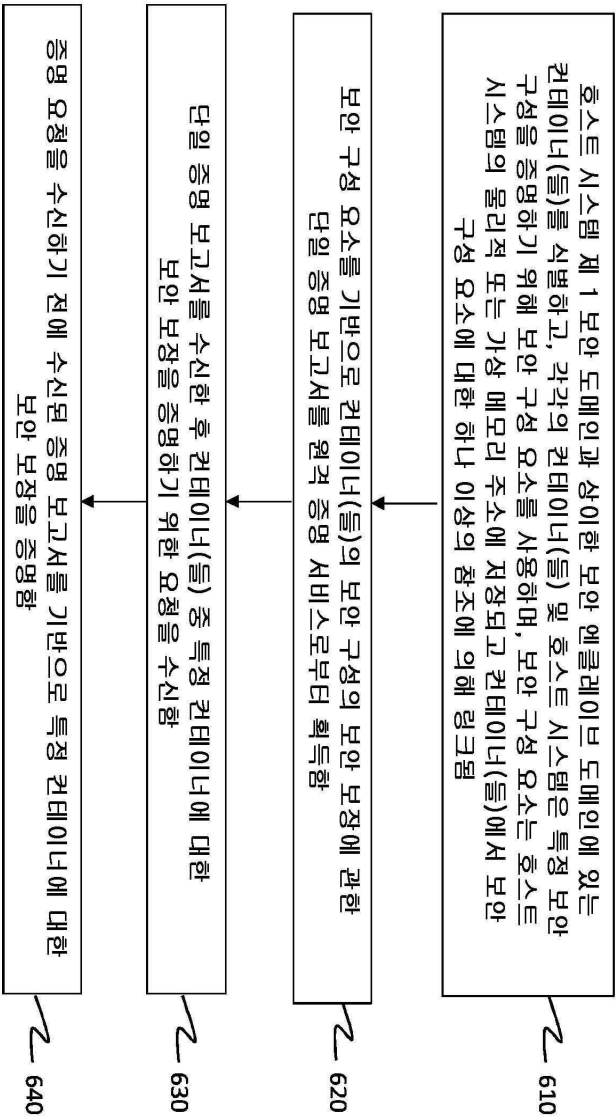


500



도면5

600



도면6