(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0038774
(43) 공개일자 2024년03월25일

(51) 국제특허분류(Int. Cl.)
G06F 21/62 (2013.01) *G06F 21/53* (2013.01)
G06F 9/455 (2018.01)
(52) CPC특허분류
G06F 21/6209 (2013.01)
G06F 21/53 (2013.01)
(21) 출원번호 10-2024-7006343
(22) 출원일자(국제) 2022년09월12일
심사청구일자 2024년02월23일
(85) 번역문제출일자 2024년02월23일
(86) 국제출원번호 PCT/EP2022/075220
(87) 국제공개번호 WO 2023/041462
국제공개일자 2023년03월23일
(30) 우선권주장
17/474,220 2021년09월14일 미국(US)

(71) 출원인
인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드
(72) 발명자
브래드버리, 조나단
미국 뉴욕 12601, 포킵시, 사우스 로드 2455, 씨/오 아이비엠 코포레이션
헨델, 토스텐
독일 보에블링겐 71032, 쇄나이처 스트리트 220, 씨/오 아이비엠 독일 리서치 앤드 디벨롭먼트 지엠비에이치
(뒷면에 계속)
(74) 대리인
허정훈

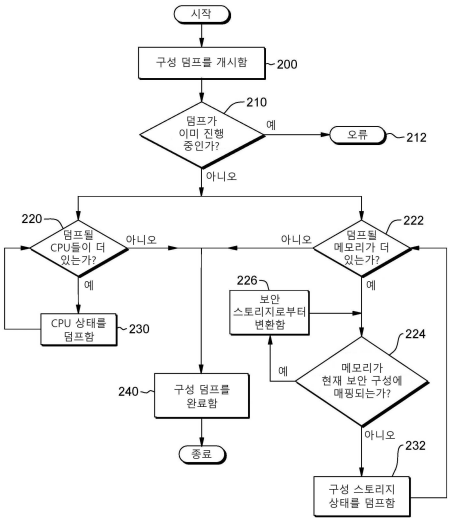
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **보안 가상 머신들의 진단 상태 저장**

(57) 요 약

가상 머신의 진단 상태(diagnostic state)를 저장하기 위한 적어도 하나의 요청이 획득된다. 상기 가상 머신의 저장된 진단 상태를 제공하기 위해 상기 가상 머신의 진단 상태의 저장이, 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 수행된다. 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체(an untrusted entity)가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 상기 저장을 수행하는 단계는 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함한다.

대 표 도 - 도2



(52) CPC특허분류
G06F 9/45558 (2013.01)
G06F 2009/45587 (2019.08)
(72) 발명자
부엔겐, 라인하르트
독일 보에블링겐 71032, 쇄나이처 스트리트 220, 씨/오 아이비엠 독일 리서치 앤드 디벨롭먼트 지엠비에이치
임브렌다, 클라우디오
독일 보에블링겐 71032, 쇄나이처 스트리트 220, 씨/오 아이비엠 독일 리서치 앤드 디벨롭먼트 지엠비에이치
본트래거, 크리스찬
독일 보에블링겐 71032, 쇄나이처 스트리트 220, 씨/오 아이비엠 독일 리서치 앤드 디벨롭먼트 지엠비에이치
프랭크, 야노쉬 안드레아스
독일 보에블링겐 71032, 쇄나이처 스트리트 220, 씨/오 아이비엠 독일 리서치 앤드 디벨롭먼트 지엠비에이치

명 세 서

청구범위

청구항 1

컴퓨팅 환경 내에서 처리를 용이하게 하기 위한 컴퓨터 프로그램 제품에 있어서, 상기 컴퓨터 프로그램 제품은:

하나 또는 그 이상의 컴퓨터 판독 가능 스토리지 매체 및 방법을 수행하기 위해 상기 하나 또는 그 이상의 컴퓨터 판독 가능 스토리지 매체에 집합적으로 저장된 프로그램 명령을 포함하고, 상기 방법은:

가상 머신의 진단 상태(diagnostic state)를 저장하기 위한 적어도 하나의 요청을 획득하는 단계; 및

상기 가상 머신의 저장된 진단 상태를 제공하기 위해 상기 가상 머신의 진단 상태의 저장을, 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 수행하는 단계를 포함하고, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체(an untrusted entity)가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 상기 저장을 수행하는 단계는 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함하는

컴퓨터 프로그램 제품.

청구항 2

제1항에 있어서, 상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 메모리의 내용들(contents of memory)을 포함하는

컴퓨터 프로그램 제품.

청구항 3

제1항 또는 제2항에 있어서, 상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 하나 또는 그 이상의 프로세서 레지스터들의 내용들을 포함하는

컴퓨터 프로그램 제품.

청구항 4

제1항 내지 제3항의 항들 중 어느 한 항에 있어서, 상기 방법은 상기 가상 머신의 진단 상태의 저장을 개시하기 위해 진단 상태 저장 개시 요청(an initiate store diagnostic state request)을 획득하는 단계를 더 포함하며, 상기 진단 상태 저장 개시 요청은 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태의 적어도 일부를 암호화하는 데 사용될 제1 암호화 키를 획득하는 데 사용되는

컴퓨터 프로그램 제품.

청구항 5

제4항에 있어서, 상기 적어도 하나의 요청을 획득하는 단계는 진단 프로세서 상태 저장 요청(a store diagnostic processor state request)을 획득하는 단계를 포함하고, 상기 진단 프로세서 상태 저장 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

암호화된 프로세서 내용들을 제공하기 위해 상기 제1 암호화 키를 사용하여 상기 가상 머신의 적어도 하나의 프로세서의 적어도 선택 내용들(at least select contents)을 암호화하는 단계; 및

상기 암호화된 프로세서 내용들을 저장하는 단계를 포함하는

컴퓨터 프로그램 제품.

청구항 6

제1항 내지 제5항의 항들 중 어느 한 항에 있어서, 상기 적어도 하나의 요청을 획득하는 단계는 보안 메모리로

부터의 변환 요청(a convert from secure memory request)을 획득하는 단계를 포함하고, 상기 보안 메모리로부터 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

메모리의 암호화된 내용들(encrypted contents of memory)을 제공하기 위해 제2 암호화 키를 사용하여 암호화되지 않은 상기 가상 머신의 메모리의 적어도 선택 내용들을 암호화하는 단계; 및

메모리의 상기 암호화된 내용들을 저장하는 단계를 포함하는

컴퓨터 프로그램 제품.

청구항 7

제6항에 있어서, 보안 메모리로부터의 상기 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

내용들이 저장될 메모리의 유닛(a unit of memory)이 암호화되었는지를 결정하는 단계; 및

상기 메모리의 유닛이 암호화되지 않았다고 결정하는 것에 기초하여, 상기 제2 암호화 키를 사용하여 상기 메모리의 유닛을 암호화하는 단계를 포함하고, 암호화된 메모리의 유닛들은 재-암호화되지 않는

컴퓨터 프로그램 제품.

청구항 8

제1항 내지 제7항의 항들 중 어느 한 항에 있어서, 상기 저장된 진단 상태는 메모리의 암호화된 내용들을 포함하고, 상기 방법은:

진단 메모리 상태 저장 요청(a store diagnostic memory state request)을 획득하는 단계; 및

상기 진단 메모리 상태 저장 요청을 획득하는 것에 기초하여, 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 메타데이터를 저장하는 단계를 더 포함하는

컴퓨터 프로그램 제품.

청구항 9

제8항에 있어서, 상기 메타데이터는 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 하나 또는 그 이상의 선택 값들을 생성하기 위해 사용될 하나 또는 그 이상의 선택 컴포넌트 값들을 포함하는

컴퓨터 프로그램 제품.

청구항 10

제1항 내지 제9항의 항들 중 어느 한 항에 있어서, 상기 방법은:

진단 상태의 저장을 완료하기 위해 완료 요청(a complete request)을 획득하는 단계; 및

상기 완료 요청을 획득하는 것에 기초하여, 상기 가상 머신의 암호화되어 저장된 진단 상태를 해독하는 데 사용될 데이터를 제공하는 단계를 더 포함하는

컴퓨터 프로그램 제품.

청구항 11

제10항에 있어서, 상기 데이터는 메모리의 암호화된 내용들의 해독(decryption)에 사용되는 제2 암호화 키와 암호화에 사용될 하나 또는 그 이상의 선택 값들을 생성하기 위해 사용되는 논스 값(a nonce value)을 포함하고, 상기 데이터의 적어도 일부는 암호화되는

컴퓨터 프로그램 제품.

청구항 12

컴퓨팅 환경 내에서 처리를 용이하게 하기 위한 컴퓨터 시스템에 있어서, 상기 컴퓨터 시스템은:

메모리(a memory); 및

상기 메모리와 통신하는 적어도 하나의 프로세서를 포함하고, 상기 컴퓨터 시스템은 방법을 수행하도록 구성되며, 상기 방법은:

가상 머신의 진단 상태(diagnostic state)를 저장하기 위한 적어도 하나의 요청을 획득하는 단계; 및

상기 가상 머신의 저장된 진단 상태를 제공하기 위해 상기 가상 머신의 진단 상태의 저장을, 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 수행하는 단계를 포함하고, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체(an untrusted entity)가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 상기 저장을 수행하는 단계는 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함하는

컴퓨터 시스템.

청구항 13

제12항에 있어서, 상기 방법은 상기 가상 머신의 진단 상태의 저장을 개시하기 위해 진단 상태 저장 개시 요청(an initiate store diagnostic state request)을 획득하는 단계를 더 포함하며, 상기 진단 상태 저장 개시 요청은 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태의 적어도 일부를 암호화하는 데 사용될 제1 암호화 키를 획득하는 데 사용되는

컴퓨터 시스템.

청구항 14

제13항에 있어서, 상기 적어도 하나의 요청을 획득하는 단계는 진단 프로세서 상태 저장 요청(a store diagnostic processor state request)을 획득하는 단계를 포함하고, 상기 진단 프로세서 상태 저장 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

암호화된 프로세서 내용들을 제공하기 위해 상기 제1 암호화 키를 사용하여 상기 가상 머신의 적어도 하나의 프로세서의 적어도 선택 내용들(at least select contents)을 암호화하는 단계; 및

상기 암호화된 프로세서 내용들을 저장하는 단계를 포함하는

컴퓨터 시스템.

청구항 15

제12항 내지 제14항의 항들 중 어느 한 항에 있어서, 상기 적어도 하나의 요청을 획득하는 단계는 보안 메모리로부터 변환 요청(a convert from secure memory request)을 획득하는 단계를 포함하고, 상기 보안 메모리로부터의 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

메모리의 암호화된 내용들(encrypted contents of memory)을 제공하기 위해 제2 암호화 키를 사용하여 암호화되지 않은 상기 가상 머신의 메모리의 적어도 선택 내용들을 암호화하는 단계; 및

메모리의 상기 암호화된 내용들을 저장하는 단계를 포함하는

컴퓨터 시스템.

청구항 16

제12항 내지 제15항의 항들 중 어느 한 항에 있어서, 상기 저장된 진단 상태는 메모리의 암호화된 내용들을 포함하고, 상기 방법은:

진단 메모리 상태 저장 요청(a store diagnostic memory state request)을 획득하는 단계; 및

상기 진단 메모리 상태 저장 요청을 획득하는 것에 기초하여, 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 메타데이터를 저장하는 단계를 더 포함하는

컴퓨터 시스템.

청구항 17

컴퓨팅 환경 내에서 처리를 용이하게 하는 컴퓨터-구현 방법에 있어서, 상기 컴퓨터-구현 방법은:

가상 머신의 진단 상태(diagnostic state)를 저장하기 위한 적어도 하나의 요청을 획득하는 단계; 및

상기 가상 머신의 저장된 진단 상태를 제공하기 위해 상기 가상 머신의 진단 상태의 저장을, 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 수행하는 단계를 포함하고, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체(an untrusted entity)가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 상기 저장을 수행하는 단계는 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함하는

컴퓨터-구현 방법.

청구항 18

제17항에 있어서, 상기 방법은 상기 가상 머신의 진단 상태의 저장을 개시하기 위해 진단 상태 저장 개시 요청(an initiate store diagnostic state request)을 획득하는 단계를 더 포함하며, 상기 진단 상태 저장 개시 요청은 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태의 적어도 일부를 암호화하는 데 사용될 제1 암호화 키를 획득하는 데 사용되는

컴퓨터-구현 방법.

청구항 19

제18항에 있어서, 상기 적어도 하나의 요청을 획득하는 단계는 진단 프로세서 상태 저장 요청(a store diagnostic processor state request)을 획득하는 단계를 포함하고, 상기 진단 프로세서 상태 저장 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

암호화된 프로세서 내용들을 제공하기 위해 상기 제1 암호화 키를 사용하여 상기 가상 머신의 적어도 하나의 프로세서의 적어도 선택 내용들(at least select contents)을 암호화하는 단계; 및

상기 암호화된 프로세서 내용들을 저장하는 단계를 포함하는

컴퓨터-구현 방법.

청구항 20

제17항 내지 제19항의 항들 중 어느 한 항에 있어서, 상기 적어도 하나의 요청을 획득하는 단계는 보안 메모리로부터의 변환 요청(a convert from secure memory request)을 획득하는 단계를 포함하고, 상기 보안 메모리로부터 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:

메모리의 암호화된 내용들(encrypted contents of memory)을 제공하기 위해 제2 암호화 키를 사용하여 암호화되지 않은 상기 가상 머신의 메모리의 적어도 선택 내용들을 암호화하는 단계; 및

메모리의 상기 암호화된 내용들을 저장하는 단계를 포함하는

컴퓨터 구현 방법.

발명의 설명

기 술 분 야

[0001] [0001] 본 발명의 하나 또는 그 이상의 실시예들은, 일반적으로 컴퓨팅 환경내에서 처리(processing)를 용이하게 하는 것에 관한 것이며, 특히 그러한 처리를 개선하는 것에 관한 것이다.

배 경 기 술

[0002] [0002] 가상 머신의 진단 상태 저장(A storing of diagnostic state)(가상 머신 덤프라고도 함)은 추후 분석을 위해 파일에 가상 머신의 상태를 제공한다. 이는 가상 머신 오류의 근본 원인을 찾는 데 자주 사용된다. 가상 머신의 상태는, 예를 들어, 가상 머신의 메모리 사본뿐만 아니라 가상 머신의 하나 또는 그 이상의 중앙 처리

장치들의 선택 레지스터들의 사본(a copy of select registers)도 포함한다.

발명의 내용

해결하려는 과제

[0003] [0003] 일부 가상 머신들은 보안 가상 머신들(secure virtual machines)이고, 이들 보안 가상 머신들에서는 컴퓨터 시스템의 하드웨어 및 펌웨어가 가상 머신이 적극적으로 공유로 지정하는 특수 메모리 영역들을 제외하고 보안 가상 머신의 상태에 대한 액세스를 거부한다. 그러한 가상 머신은, 일반적으로 가능할 수 있는, 가상 머신을 호스팅하는 운영 체제의 관리자나 하이퍼바이저에 의한 액세스가 가능하지 않다. 가상 머신의 상태에 액세스한다는 것은 호스트 시스템 상에서 승인된 주체(an authorized entity)가 가상 머신 메모리로부터 암호화 키들 또는 기밀 문서들과 같은, 민감한 정보를 읽을 수 있다는 것을 의미하며, 이는 보안 위험을 초래할 수 있다.

[0004] [0004] 보안 가상 머신과 같은, 가상 머신의 진단 상태를 저장하는 것을 포함하는, 가상 머신과 관련된 처리가 용이하게 될 수 있다.

과제의 해결 수단

[0005] [0005] 종래 기술의 단점들은 극복되고, 컴퓨팅 환경 내에서 처리를 용이하게 하기 위한 컴퓨터 프로그램 제품의 제공을 통해 추가적인 이점들이 제공된다. 상기 컴퓨터 프로그램 제품은 하나 또는 그 이상의 컴퓨터 판독 가능 스토리지 매체들 및 방법을 수행하기 위해 하나 또는 그 이상의 컴퓨터 판독 가능 스토리지 매체들에 집합적으로 저장된 프로그램 명령들을 포함한다. 상기 방법은 가상 머신의 진단 상태(diagnostic state)를 저장하기 위한 적어도 하나의 요청을 획득하는 단계를 포함한다. 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 상기 가상 머신의 진단 상태의 저장이 상기 가상 머신의 저장된 진단 상태를 제공하기 위해 수행된다. 상기 저장을 수행하는 단계는, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체(an untrusted entity)가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함한다.

[0006] [0006] 가상 머신(예를 들어, 보안 가상 머신)의 진단 상태는 진단 목적들을 위해 저장되는 가상 머신 상태의 보안을 유지하면서 저장될 수 있다.

[0007] [0007] 예들로서, 상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 메모리의 내용들(contents of memory)을 포함하거나 및/또는 상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 하나 또는 그 이상의 프로세서 레지스터들의 내용들을 포함한다.

[0008] [0008] 일 예에서, 상기 가상 머신의 진단 상태의 저장을 개시하기 위해 진단 상태 저장 개시 요청(an initiate store diagnostic state request)이 획득된다. 상기 진단 상태 저장 개시 요청은 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태의 적어도 일부를 암호화하는 데 사용될 제1 암호화 키를 획득하는 데 사용된다.

[0009] [0009] 일 예에서, 상기 적어도 하나의 요청을 획득하는 단계는 진단 프로세서 상태 저장 요청(a store diagnostic processor state request)을 획득하는 단계를 포함한다. 상기 진단 프로세서 상태 저장 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는, 예를 들어, 암호화된 프로세서 내용들을 제공하기 위해 상기 제1 암호화 키를 사용하여 상기 가상 머신의 적어도 하나의 프로세서의 적어도 선택 내용들(at least select contents)을 암호화하는 단계, 및 상기 암호화된 프로세서 내용들을 저장하는 단계를 포함한다.

[0010] [0010] 일 예에서, 상기 적어도 하나의 요청을 획득하는 단계는 보안 메모리로부터 변환 요청(a convert from secure memory request)을 획득하는 단계를 포함한다. 상기 보안 메모리로부터의 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는, 예를 들어, 메모리의 암호화된 내용들(encrypted contents of memory)을 제공하기 위해 제2 암호화 키를 사용하여 암호화되지 않은 상기 가상 머신의 메모리의 적어도 선택 내용들을 암호화하는 단계, 및 메모리의 상기 암호화된 내용들을 저장하는 단계를 포함한다.

[0011] [0011] 일 예로서, 상기 진단 상태의 저장을 수행하는 단계는, 보안 메모리로부터의 상기 변환 요청을 획득하는 것에 기초하여, 예를 들어, 내용들이 저장될 메모리의 유닛(a unit of memory)이 암호화되었는지를 결정하는 단계, 및 상기 메모리의 유닛이 암호화되지 않았다고 결정하는 것에 기초하여, 상기 제2 암호화 키를 사용하여 상기 메모리의 유닛을 암호화하는 단계를 포함하며, 암호화된 메모리의 유닛들은 재-암호화되지 않는다(not re-

encrypted).

[0012] [0012] 암호화되지 않은 내용들을 암호화함으로써(그리고 암호화된 상태를 재-암호화하지 않음으로써) 처리 사이클들과 복잡성이 감소되고, 시스템 성능이 향상된다.

[0013] [0013] 일 예에서, 상기 저장된 진단 상태는 메모리의 암호화된 내용들을 포함한다. 또한, 진단 메모리 상태 저장 요청(a store diagnostic memory state request)이 획득되며, 그리고 상기 진단 메모리 상태 저장 요청을 획득하는 것에 기초하여, 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 메타데이터가 저장된다.

[0014] [0014] 일 예로서, 상기 메타데이터는 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 하나 또는 그 이상의 선택 값들(one or more select values)을 생성하기 위해 사용될 하나 또는 그 이상의 선택 컴포넌트 값들(one or more select component values)을 포함한다.

[0015] [0015] 일 예에서, 진단 상태의 저장을 완료하기 위해 완료 요청(a complete request)이 획득되며, 그리고 상기 완료 요청을 획득하는 것에 기초하여, 상기 가상 머신의 암호화되어 저장된 진단 상태를 해독하는 데 사용될 데이터가 제공된다.

[0016] [0016] 일 예로서, 상기 데이터는 메모리의 암호화된 내용들의 해독(decryption)에 사용되는 제2 암호화 키와 암호화에 사용될 하나 또는 그 이상의 선택 값들을 생성하기 위해 사용되는 논스 값(a nonce value)을 포함한다. 또한, 일 예에서, 상기 데이터의 적어도 일부는 암호화된다.

[0017] [0017] 하나 또는 그 이상의 실시 예들과 관련된 컴퓨터 구현 방법들 및 시스템들도 여기에 설명되고 청구된다. 또한, 하나 또는 그 이상의 실시 예들과 관련된 서비스도 설명되고 여기에 청구될 수 있다.

[0018] [0018] 추가의 특징들 및 이점들이 여기에 기술된 기술들을 통해 실현된다. 다른 실시 예들 및 특징들이 본 명세서에서 상세하게 설명되고 청구된 실시 예들의 일부로 간주된다.

도면의 간단한 설명

[0019] [0019] 하나 혹은 그 이상의 실시 예들이 명세서의 결론에 있는 청구범위들의 예들로서 구체적으로 기재되고 명확하게 청구된다. 본 발명의 하나 또는 그 이상의 실시 예들의 전술한 내용 및 목적들, 특징들, 그리고 이점들은 첨부 도면들과 함께 취해진 다음의 상세한 설명으로부터 명백하다:

도 1은 본 발명의 하나 또는 그 이상의 실시예들을 포함하고 사용하기 위한 컴퓨팅 환경의 한 예를 도시한다;

도 2는, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 구성 덤프를 수행하는 것으로도 지칭되는, 구성(예를 들어, 가상 머신)의 진단 상태를 저장하는 단계의 일례를 도시한다.

도 3은, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 구성 덤프 개시 제어 블록(an initiate configuration dump control block)의 일례를 도시한다.

도 4a는, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 중앙 처리 장치(CPU) 상태 덤프 제어 블록(a dump central processing unit (CPU) state control block)의 일례를 도시한다.

도 4b는, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 보안 중앙 처리 장치 덤프 영역의 한 예를 도시한다.

도 4c는 본 발명의 하나 또는 그 이상의 실시예들에 따라 사용되는 보안 스토리지 제어 블록으로부터의 변환(a convert from secure storage control block)의 일례를 도시한다.

도 5는, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 덤프 구성 스토리지 상태 제어 블록(a dump configuration storage state control block)의 일례를 도시한다.

도 6은, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 구성 덤프 완료 제어 블록(a complete configuration dump control block)의 일례를 도시한다.

도 7a-7c는, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 컴퓨팅 환경 내에서 처리를 용이하게 하는 일 예를 도시한다;

도 8a는 본 발명의 하나 또는 그 이상의 실시예들을 통합하고 사용하기 위한 컴퓨팅 환경의 또 다른 예를 도시한다;

도 8b는,본 발명의 하나 또는 그 이상의 실시예들에 따라, 도 7a의 메모리의 추가 세부사항을 도시한다;

도 9는, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 클라우드 컴퓨팅 환경의 일 실시예를 도시한다; 그리고

도 10은, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 추상화 모델 계층들의 일 예를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0020] [0020] 본 발명의 하나 혹은 그 이상의 실시예들에 따라, 컴퓨팅 환경 내에서 처리를 용이하게 하는 능력이 제공된다. 일 예로서, 이 능력은 저장되어 있는 가상 머신 상태의 보안을 유지하면서 가상 머신(예: 보안 가상 머신)의 진단 상태가 저장될 수 있는(또한 덤프 될 수 있다고 함) 기술을 포함한다. 일 예로서, 상기 진단 상태는 신뢰할 수 없는 주체(예: 하이퍼바이저)가 읽을 수 있기 전에 암호화된다.

[0021] [0021] 일례에서, 진단 분석을 위해 저장될(즉, 덤프될) 가상 머신의 메모리 내용들은, 예를 들어, 덤프 키(또는 제1 덤프 키라고도 함)라고 하는, 선택된 키를 사용하여서 암호화될 수 있다. 예를 들어, 페이지 아웃되어(paged-out) 페이징 키(또한 제2 암호화 키라고도 함)로 현재 암호화된 메모리는 상기 선택된 키로 해독되고 재-암호화된다(decrypted and re-encrypted). 또 다른 예로서, 본 발명의 일 실시예에 따라, 사용되는 중앙 처리 장치 사이클들의 수와 복잡성을 감소시키기 위해, 암호화된 메모리(페이징 키를 사용하여 암호화된)를 재-암호화하는 대신, 메모리의 유닛들(예: 메모리의 페이지들)에 대한 선택 고유 값들(select unique values)(예: 튀크(tweak) 값들)이 제공되고 페이징 키는 선택된 키에 의해 암호화된다. 따라서, 페이지 아웃된 페이지들은 덤프를 위해 재-암호화될 필요가 없으며 아직 암호화되지 않은 페이지들만 덤프를 위해 암호화된다. 더 자세히 설명하면, 메모리의 각 유닛에는, 예를 들어, 부호 없는 정수인 하나의 값이 제공되는데, 이는 튀크 값 또는 선택 고유 값(the tweak value or select unique value)이라 한다. 이들 값들은 임의의 정수(an arbitrary integer)에서 시작하여 연속적으로 할당된다(assigned consecutively). 이 값은, 예를 들어, 리틀 엔디안 바이트 어레이(a little-endian byte array)로 변환될 수 있으며, 이 값의 암호화는, 예를 들어, AES(Advanced Encryption Standard) 알고리즘을 사용하여 수행될 수 있다.

[0022] [0022] 하나 또는 그 이상의 실시예들에서, 가상 머신(예를 들어, 보안 가상 머신)의 덤프 또는 진단 저장을 보안으로(securely) 용이하게 하기 위해, 울트라바이저 호출(Ultravisor Call)이라고 불리는 명령을 채용하는, 애플리케이션 프로그래밍 인터페이스(an application programming interface)가 제공되고 사용된다. 울트라바이저 호출 명령은 보안 가상 머신의 진단 상태를 저장하는 데 사용되는 하나 또는 그 이상의 커맨드들을 개시하는 데(initiate) 사용된다. 이들 커맨드들은, 예를 들어, 구성 덤프 개시 커맨드(an initiate configuration dump command), 중앙 처리 장치 상태 덤프 커맨드(a dump central processing unit state command), 보안 스토리지로부터 변환 커맨드(a convert from secure storage command), 구성 스토리지 상태 덤프 커맨드(a dump configuration storage state command) 및 구성 덤프 완료 커맨드(a complete configuration dump command)를 포함하며, 이들 각각에 대해서는 아래에서 자세히 설명한다.

[0023] [0023] 본 발명의 하나 또는 그 이상의 실시 예들을 통합하고 사용하기 위한 컴퓨팅 환경의 일 실시예가 도 1을 참조하여 기술된다. 일 예로, 상기 컴퓨팅 환경은 뉴욕주, 아몽크 소재의, 인터내셔널 비즈니스 머신즈 코포레이션에서 제공하는, z/Architecture® 명령 세트 아키텍처를 기반으로 한다. z/Architecture 명령 세트 아키텍처의 일 실시예는 "z/Architecture Principles of Operation", IBM 간행물 번호 SA22-7832-12, 13판, 2019년 9월에 설명되어 있으며, 전체 내용이 여기에 참조로 통합된다. 그러나, z/Architecture 명령 세트 아키텍처는 한 가지 예시적인 아키텍처일뿐이며 및/또는 인터내셔널 비즈니스 머신즈 코포레이션 및/또는 다른 주체들의 다른 유형들의 컴퓨팅 환경들도 본 발명의 하나 또는 그 이상의 실시예들을 포함할 수 있고 및/또는 사용할 수 있다. z/Architecture 및 IBM은 적어도 하나의 관할 지역에서 인터내셔널 비즈니스 머신즈 코포레이션의 상표들 또는 등록 상표들이다.

[0024] [0024] 도 1을 참조하면, 일 예에서, 컴퓨팅 환경(100)은 중앙 프로세서 콤플렉스(CEC)(102)를 포함한다. 중앙 프로세서 콤플렉스(102)는, 예를 들어, IBM Z® 서버(또는 다른 서버 또는 인터내셔널 비즈니스 머신즈 코포레이션 또는 다른 업체들에 의해서 제공되는 머신)이고, 하나 또는 그 이상의 프로세서 유닛들(또한 프로세서들이라 함)(110) 및 입력/출력(I/O) 서브시스템(111)에 결합된, 예를 들어, 메모리(104)(일명, 시스템 메모리, 메인 메모리, 메인 스토리지, 중앙 스토리지, 스토리지)와 같은, 복수의 컴포넌트들을 포함한다. 예시적 프로세서 유닛들(110)은 하나 또는 그 이상의 범용 프로세서들(일명 중앙 처리 유닛들(CPU들)) 및/또는 하나 또는 그 이상의 다른 프로세서들을 포함한다. IBM Z는 적어도 하나의 관할 지역에서 인터내셔널 비즈니스 머신즈 코포레이션

의 상표 또는 등록 상표이다.

[0025] [0025] I/O 서브시스템(111)은 중앙 프롯서 콤플렉스 복합체의 일부이거나 그로부터 분리될 수 있다. 이것은 메인 스토리지(104)와 입/출력 컨트롤 유닛들(108) 및 중앙 프로세서 콤플렉스에 결합된 입/출력(I/O) 디바이스들(106) 사이의 정보 흐름을 지시한다.

[0026] [0026] 많은 유형의 I/O 디바이스들이 사용될 수 있다. 하나의 특정 유형은 데이터 스토리지 디바이스(140)이다. 데이터 스토리지 디바이스(140)는 하나 또는 그 이상의 프로그램들(142), 하나 또는 그 이상의 컴퓨터 판독 가능 프로그램 명령들(144) 및/또는 데이터 등을 저장할 수 있다. 상기 컴퓨터 판독 가능 프로그램 명령은 본 발명의 실시예들의 기능들(functions)을 수행하도록 구성될 수 있다.

[0027] [0027] 중앙 프로세서 콤플렉스(102)는 착탈식/비-착탈식, 휘발성/비-휘발성 컴퓨터 시스템 스토리지 매체를 포함할 수 있고 및/또는 이 들에 결합될 수 있다. 중앙 전자 콤플렉스(101)는, 예를 들어, 비-착탈식, 비-휘발성 자기 매체(일반적으로 "하드 드라이브"라고 함), 착탈식, 비-휘발성 자기 디스크(예를 들어, "플로피 디스크")로부터 읽고 이에 쓰기 위한 자기 디스크 드라이브, 및/또는, CD-ROM, DVD-ROM 또는 기타 광 매체와 같은, 착탈식, 비-휘발성 광 디스크로부터 읽거나 또는 이에 쓰기 위한 광 디스크 드라이브를 포함할 수 있고 및/또는 이들에 결합될 수 있다. 다른 하드웨어 및/또는 소프트웨어 컴포넌트들이 중앙 프로세서 콤플렉스(102)와 함께 사용될 수 있음을 이해해야 한다. 예들에는, 다음이 포함되지만, 이에 제한되지는 않는다: 마이크로코드, 장치 드라이버들, 중복 처리 유닛들, 외부 디스크 드라이브 어레이들, RAID 시스템들, 테이프 드라이브들 및 데이터 보관 스토리지 시스템 등.

[0028] [0028] 또한, 중앙 프로세서 콤플렉스(102)는 다수의 다른 범용 또는 특수-목적 컴퓨팅 시스템 환경들 또는 구성들과 함께 동작할 수 있다. 중앙 프로세서 콤플렉스(102)와 함께 사용하기에 적합할 수 있는 잘 알려진 컴퓨팅 시스템들, 환경들, 및/또는 구성들의 예들은, 개인용 컴퓨터(PC) 시스템들, 서버 컴퓨터 시스템들, 씬 클라이언트들, 씩 클라이언트들, 핸드헬드 또는 랩톱 디바이스들, 멀티프로세서 시스템들, 마이크로프로세서-기반 시스템들, 셋톱 박스들, 프로그래밍 가능한 소비자 전자 제품들, 네트워크 PC들, 미니 컴퓨터 시스템들, 메인프레임 컴퓨터 시스템들, 및, 위의 시스템들 또는 디바이스들 중 어느 하나, 등을 포함하는 분산 클라우드 컴퓨팅 환경들을 포함하지만 이들로 제한되지는 않는다.

[0029] [0029] 중앙 프로세서 콤플렉스(102)는 하나 또는 그 이상의 실시예들에서 가상화 지원을 제공하고, 여기서 메모리(104)는, 예를 들어, 하나 또는 그 이상의 가상 머신들(112)(또한 게스트들이라고도 함), 가상 머신들을 관리하는, 하이퍼바이저(114)와 같은, 가상 머신 관리자, 신뢰된 실행 환경(trusted execution environment)(115) 및 프로세서 펌웨어(116)를 포함한다. 하이퍼바이저(114)의 한 예는 미국 뉴욕 아몽크 소재의 인터내셔널 비즈니스 머신즈 코포레이션에 의해서 제공되는 z/VM® 이다. 상기 하이퍼바이저는 때때로 호스트라 한다. z/VM은 적어도 하나의 관할권에서 인터내셔널 비즈니스 머신즈 코포레이션의 상표 또는 등록 상표이다.

[0030] [0030] 하나 또는 그 이상의 실시예들에서, 신뢰된 실행 환경(trusted execution environment)(115)은, 예를 들어, 본 명세서에 설명된 것과 같은 프로세스들을 수행하도록 구성된 하드웨어 및/또는 펌웨어에서, 적어도 부분적으로, 구현될 수 있다. 상기 신뢰된 실행 환경은 메모리 보호를 강화하기 위해 메모리 보호 하드웨어를 사용하는 신뢰된 펌웨어 및/또는 하드웨어이다. 게스트 소유자는, 호스트 키 문서에 임베드된, 공개 호스트 키를 사용하여 상기 신뢰된 실행 환경에 정보를 안전하게 전달할 수 있다(예를 들어, IBM Secure Execution을 사용하여). 기밀 정보를 처리하기 위해, 상기 신뢰된 실행 환경은 일치하는 개인 호스트 키(a matching private host key)를 사용한다. 상기 개인 호스트 키는 서버(예를 들어, IBM Z® 서버)에 특정하고(specific) 하드웨어 보호된다.

[0031] [0031] 프로세서 펌웨어(116)는, 예를 들어, 프로세서의 마이크로코드 또는 밀리코드를 포함한다. 예를 들어, 상기 프로세서 펌웨어는 더 높은 수준의 머신 코드 구현에 사용되는 하드웨어 수준 및/또는 데이터 구조를 포함한다. 일 실시예에서, 상기 프로세서 펌웨어는, 예를 들어, 기본 하드웨어에 특정한 신뢰된 소프트웨어, 마이크로코드 또는 밀리코드를 포함하고 시스템 하드웨어에 대한 운영 체제 액세스를 제어하는 마이크로코드 또는 밀리코드로 일반적으로 전달되는 독점 코드를 포함한다.

[0032] [0032] 중앙 프로세서 컴플렉스의 가상 머신 지원은, 각각 다른 프로그램들(120)과 작동할 수 있고, Linux® 운영 체제와 같은, 게스트 운영 체제(122)를 실행할 수 있는 많은 수의 가상 머신들(112)을 작동하는 능력을 제공한다. 각각의 가상 머신(112)은 별도의 시스템으로서 기능할 수 있다. 즉, 각 가상 머신은 독립적으로 재설정되

고 게스트 운영 체제를 실행하며 다른 프로그램과 함께 작동할 수 있다. 가상 머신에서 실행되는 운영 체제 또는 애플리케이션은 완전하고 완전한 시스템(a full and complete system)에 액세스할 수 있는 것처럼 보이지만 실제로는 일부만 사용될 수 있다. z/VM 및 Linux가 예들로서 제공되지만, 다른 가상 머신 관리자들 및/또는 운영 체제들이 본 발명의 하나 또는 그 이상의 실시예들에 따라 사용될 수 있다. 등록 상표 Linux®는 전 세계적으로 해당 상표를 소유하고 있는 Linus Torvalds의 독점 사용권자인 Linux Foundation의 2차 라이선스에 따라 사용된다.

[0033] [0033] 일 실시예에서, 하나 또는 그 이상의 가상 머신들(112)은 보안 가상 머신들(secure virtual machines)이다. 보안 가상 머신은 하이퍼바이저(예: 하이퍼바이저(114))가 보안 가상 머신의 상태(예: 메모리, 레지스터들, 등)를 관찰할 수 없는 방식으로 시작된다. 예를 들어, 기밀 컴퓨팅(confidential computing)의 일 실시예에서, 하이퍼바이저는 보안 가상 머신을 시작/중지할 수 있고, 하이퍼바이저는 보안 가상 머신을 시작하는 데 사용되는 데이터가 어디에 있는지는 알 수 있지만 실행 중인 보안 가상 머신을 들여다볼 수는 없다. 보안 가상 머신을 로드/시작하는 데 사용되는 데이터는 하이퍼바이저가 보안 머신을 볼 수 없는 방식으로 암호화될 수 있다. 보안 가상 머신 이미지의 소유자는 보안 게스트 메타데이터에 기밀 데이터를 배치한 다음 보안 게스트 메타데이터와 함께 보안 가상 머신 이미지를 생성한다. 보안 가상 머신이 로드된 후, 보안 가상 머신의 상태와의 모든 상호 작용은 신뢰된 실행 환경(trusted execution environment)(115)과 같은 신뢰된 실행 환경에 의해 처리된다.

[0034] [0034] 본 발명의 일 실시예에 따라, 가상 머신 상태의 보안을 유지하면서 보안 가상 머신의 진단 상태(diagnostic state of a secure virtual machine)가 저장될 수 있다(즉, 보안 가상 머신이 덤프될 수 있다). 일 예에서, 진단 상태는 신뢰할 수 없는 주체(예: 하이퍼바이저)가 읽을 수 있기 전에 암호화된다. 하나 또는 그 이상의 실시예들에 따라, 보안 가상 머신의 덤프를 보안으로 용이하게 하기 위해, 하드웨어 및 펌웨어에 대해 애플리케이션 프로그래밍 인터페이스(an application programming interface)가 사용된다. 애플리케이션 프로그래밍 인터페이스를 통해 하이퍼바이저는 가상 머신 소유자에게 전달될 수 있는 가상 머신 상태의 암호화된 버전에 액세스할 수 있다. 그러면, 상기 소유자(예: 독점적)는 상태를 해독하고 분석을 위해 암호화되지 않은 가상 머신 덤프를 가져올 수 있는데, 이는 상기 소유자가 암호화된 덤프를 위한 암호 해독 키가 있는 인스턴스(예: 유일한 인스턴스)이기 때문이다. 예를 들어, 가상 머신을 시작하는 데 사용되는 메타데이터는 소유자 키 또는 통신 키라고 하는 기밀 키를 포함하며, 이는, 예를 들어, 신뢰된 펌웨어만이 기밀 소유자 키를 획득할 수 있도록 보호된다. 이 키는, 예를 들어, 암호화된 진단 데이터를 해독하기 위해 배타적으로 사용된다.

[0035] [0035] 일례로서, 애플리케이션 프로그래밍 인터페이스는, 예를 들어, 구성 덤프 개시 커맨드(an initiate configuration dump command), 중앙 처리 장치 상태 덤프 커맨드(a dump central processing unit state command), 보안 스토리지로부터의 변환 커맨드(a convert from secure storage command), 구성 스토리지 상태 덤프 커맨드(a dump configuration storage state command) 및 구성 덤프 완료 커맨드(a complete configuration dump command)을 포함하는, 복수의 커맨드들을 위한 복수의 울트라바이저 호출들(Ultravisor Calls)로 나누어진다. 예를 들어, 각 커맨드는, 일 실시예에서, 울트라바이저 호출 연산을 명시하는 오피 코드(an opcode)와 복수의 오퍼랜드들을 포함하는 울트라바이저 호출 명령의 실행에 명시된다. 일 예로서, 상기 명령의 제3 오퍼랜드(예를 들어, 직접 필드(an immediate field))가 0을 명시할 때, 정상 커맨드(a normal command)가 명시되어 상기 명령의 제2 오퍼랜드(예를 들어, 레지스터)는 울트라바이저 제어 블록을 가리키는데, 상기 울트라바이저 제어 블록은 울트라바이저(또는 다른 신뢰된 주체)에 의해 실행될 커맨드를 명시하는 스토리지에 위치한다.

[0036] [0036] 울트라바이저 호출 명령에 의해 명시될 수 있는 진단 상태를 저장하기 위해 사용되는 예시적인 커맨드들 및 제어 블록들이 도 2-6을 참조하여 설명된다. 예를 들어, 도 2는 상기 커맨드들을 채용하는 처리 흐름을 도시하고, 도 3-6은 본 발명의 하나 또는 그 이상의 실시예들에 따라 상기 커맨드들에 의해 사용되는 다양한 제어 블록들/정보를 도시한다.

[0037] [0037] 먼저 도 2를 참조하면, 구성 덤프 개시 커맨드(an initiate configuration dump command)(또한 진단 저장 개시 요청(an initiate store diagnostic request)이라고도 함)가 단계(200)에서 호출된다. 예를 들어, 상기 울트라바이저 호출 명령은, 예를 들어 하이퍼바이저(예: 하이퍼바이저 114)에 의해 발행되며, 이는, 예를 들어, 울트라바이저와 같은, 신뢰된 주체에 의해 실행될 구성 덤프 개시 커맨드를 포함하는 제어 블록에 포인터(a pointer)를 명시한다. 그러한 제어 블록의 일 예가 도 3을 참조하여 설명된다.

[0038] [0038] 일례에서, 구성 덤프 개시 울트라바이저 제어 블록(an initiate configuration dump ultravisor

control block)(300)은, 예를 들어, 다음 필드들을 포함한다:

[0039] [0039] 길이(302): 이 필드(예를 들어, 바이트 0-1)는, 예를 들어, 부호 없는 이진 정수(예를 들어, 16비트)를 포함하며, 이 값은 제어 블록(파라미터 블록이라고도 함)의 길이를 바이트로 명시한다. 상기 길이는 선택 값(예: 40 hex)을 명시하고; 그렇지 않으면 선택 응답 코드(예: 0005 hex)가 적용된다.

[0040] [0040] 커맨드 코드(304): 이 필드(예를 들어, 바이트 2-3)는 부호 없는 이진 정수(예를 들어, 16비트)를 포함하며, 이 값은 구성 덤프 개시 울트라바이저 호출 커맨드(the initiate configuration dump Ultravisor Call command)를 위한 커맨드 코드를 명시한다.

[0041] [0041] 응답 코드(306): 이 필드(예: 바이트 4-5)는 부호 없는 이진 정수(예: 16비트)를 포함하며, 이는 응답 코드를 명시한다. 이 값은 연산이 완료되면 저장된다.

[0042] [0042] 반환 이유 코드(Return Reason Code)(308): 이 필드(예를 들어, 바이트 6-7)는 반환 이유 코드를 포함한다.

[0043] [0043] 보안 구성 컨텍스트 핸들(310): 이 필드(예를 들어, 바이트 24-31)는 덤프 프로세스를 초기화할 보안 구성(예를 들어, 가상 머신)을 식별하는 값(예를 들어, 64비트 값)을 포함한다.

[0044] [0044] 만일 특별 조건(a special condition)이 존재한다면, 울트라바이저 호출 제어 블록에는 선택 코드(예: 0001 hex) 이외의 응답 코드가 저장된다. 구성 덤프 개시 울트라바이저 호출(the initiate configuration dump Ultravisor Call)에 대한 특별 조건들은, 예를 들어, 다음을 포함한다:

[0045] [0045] 0003 hex: 구성 덤프 개시 울트라바이저 호출 커맨드가 성공적으로 실행된 울트라바이저 호출 초기화 커맨드(a successfully executed initialize Ultravisor Call command)를 발행하지 않은 구성에서 발행되었다.

[0046] [0046] 0005 hex: 구성 덤프 개시 울트라바이저 호출 커맨드가 선택 값(예: 40 hex)과 같지 않은 길이로 발행되었다.

[0047] [0047] 0020 hex: 보안 구성 컨텍스트 핸들이 정의된 보안 구성에 대한 유효한 핸들이 아니다.

[0048] [0048] 0101 hex: 구성이 덤프하도록 승인되지 않았다.

[0049] [0049] 0102 hex: 언팩된 이미지 확인 울트라바이저 호출(A verify unpacked image Ultravisor Call)이 명시된 구성에 대해 성공적으로 완료되지 않았다.

[0050] [0050] 0103 hex: 구성을 위해 이미 덤프가 진행 중이다.

[0051] [0051] 구성 덤프 개시 울트라바이저 호출(The initiate configuration dump Ultravisor Call)은 명시된 보안 구성(예: 보안 가상 머신)에 대한 덤프 프로세스를 초기화하는 데 사용된다. 초기화의 일부로서, 선택 보안 구성 데이터(예를 들어, 암호화된 프로세서 상태)를 덤프하기 위해 고유한 덤프 키(a unique dump key)(예를 들어, 암호화 키, 명확성을 위해 여기서는 제1 암호화 키라고 함)가 유도된다(derived). 일 예로서, 상기 커맨드는 저장된 고객 통신 키(예: 소유자 키) 및 키 유도 시드(a key derivation seed)로부터 덤프 키를 유도한다. 이 키는 저장되어 나중에 다른 커맨드들을 통해 제공되는 데이터를 암호화하는 데 사용될 수 있다.

[0052] [0052] 일례에서, 명시된 구성을 목표로 하는 언팩된 이미지 확인 울트라바이저 호출(언팩된 이미지 - 안전하고 해독된 이미지의 무결성을 확인하는 데 사용됨)은 성공적으로 완료되어야 하고; 그렇지 않으면, 연산은 수행되지 않으며 선택된 응답 코드(예: 0101 hex)가 저장된다. 만일 성공적으로 완료된 구성 덤프 완료 울트라바이저 호출 없이(without a successfully completed complete configuration dump Ultravisor Call) 명시된 구성에 대해 다른 구성 덤프 개시 울트라바이저 호출이 성공적으로 완료된다면, 연산은 수행되지 않으며 선택된 응답 코드(예: 0102 hex)가 저장된다.

[0053] [0053] 일례에서, 덤프 프로세스는 보안 가상 머신이 여전히 설정 프로세스에 있다면 개시되지 않는데, 이는 가상 머신의 상태가 제1 명령이 실행되는 지점까지는 소유자에게 알려지기 때문이다. 더 나아가, 표시된 바와 같이, 이 커맨드가 덤프 프로세스를 개시하면, 이는 다른 애플리케이션 프로그래밍 인터페이스 커맨드들이 하이퍼바이저에 의해 호출될 수 있게 한다. 덤프 프로세스가 개시된 후에는 현재 프로세스가 완료될 때까지 이 보안 가상 머신에 대해 제2 덤프 프로세스는 개시되지 않는다. 따라서, 일례에서, 구성 덤프 개시 커맨드를 실행하는 것에 기초하여, 덤프(또한 진단 상태 저장(a store diagnostic state)이라고도 함)가 선택 구성(예를 들어, 가상 머신)에 대해 이미 진행 중인지에 관한 결정이 내려진다(210). 만일 덤프가 구성에 대해 이미 진행 중이면,

일례에서, 이 처리에 대해 오류가 표시된다(212). 그러나, 만일 덤프가 구성에 대해 아직 진행 중이 아니면, 처리가, 여기에 설명된 바와 같이, 계속된다.

[0054] [0054] 예를 들어, 구성 덤프를 개시하는 것에 기초하여, 가상 머신의 상태가 덤프될 것인지에 관한 결정이 내려진다. 도 2로 돌아가면, 하나 또는 그 이상의 중앙 처리 장치들의 상태가 덤프되어야 하는지에 관한 질의가 이루어진다(220). 만일 하나 또는 그 이상의 중앙 처리 장치들의 상태가 덤프되어야 한다면, 중앙 처리 장치 상태 덤프 커맨드(a dump central processing unit state command)(또한 진단 프로세서 상태 저장 요청(a store diagnostic processor state request)이라고도 함)이 개시된다(230). 예를 들어, 울트라바이저 호출 명령(the Ultravisor Call instruction)이, 예를 들어, 하이퍼바이저에 의해 발행되며, 이는, 예를 들어, 신뢰된 주체에 의해 실행될 중앙 처리 장치 상태 덤프 커맨드(the dump central processing unit state command)를 포함하는 제어 블록에 포인터를 명시한다. 그러한 호출 제어 블록의 예가 도 4a를 참조하여 설명된다.

[0055] [0055] 일례에서, 중앙 처리 장치 상태 덤프 울트라바이저 호출 제어 블록(a dump central processing unit state Ultravisor Call control block)(400)은, 예를 들어, 다음 필드들을 포함한다:

[0056] [0056] 길이(402): 이 필드(예를 들어, 바이트 0-1)는, 예를 들어, 부호 없는 이진 정수(예를 들어 16비트)를 포함하며, 이 값은 제어 블록의 길이를 바이트로 명시한다. 상기 길이는 선택 값(예: 50 hex)을 명시하고; 그렇지 않으면, 선택 응답 코드(예: 0005 hex)가 적용된다.

[0057] [0057] 커맨드 코드(404): 이 필드(예를 들어, 바이트 2-3)는 부호 없는 이진 정수(예를 들어, 16비트)를 포함하고, 이 값은 중앙 처리 장치 상태 덤프 울트라바이저 호출 커맨드(the dump central processing unit state Ultravisor Call command)를 위한 커맨드 코드를 명시한다.

[0058] [0058] 응답 코드(406): 이 필드(예: 바이트 4-5)는 부호 없는 이진 정수(예: 16비트)를 포함하고, 이는 응답 코드를 명시한다. 이 값은 연산이 완료되면 저장된다.

[0059] [0059] 반환 이유 코드(408): 이 필드(예: 바이트 6-7)는 반환 이유 코드를 포함한다. 중앙 처리 장치 상태 덤프 울트라바이저 호출 커맨드의 경우, 이 필드는, 예를 들어, 0이다.

[0060] [0060] 보안 중앙 처리 장치 컨텍스트 핸들(Secure Central Processing Unit Context Handle)(410): 이 필드(예를 들어, 바이트 24-31)는 상태가 덤프될 보안 중앙 처리 장치를 식별하는 값(예를 들어, 64비트 값)을 포함한다.

[0061] [0061] 보안 중앙 처리 장치 덤프 영역 원점(Secure Central Processing Unit Dump Area Origin)(412): 선택 수(예를 들어, 12)의 0들이 첨부된 이 필드(예를 들어, 바이트 32-39의 비트 0-51)는 보안 중앙 처리 장치 덤프 영역의 논리 주소(예를 들어, 64비트 논리 주소)를 포함한다.

[0062] [0062] 만일 특별 조건이 존재한다면, 울트라바이저 호출 제어 블록에 선택 코드(예: 0001 hex) 이외의 응답 코드가 저장된다. 중앙 처리 장치 상태 덤프 울트라바이저 호출에 대한 특별 조건들은, 예를 들어, 다음과 같다:

[0063] [0063] 0003 hex: 중앙 처리 장치 상태 덤프 울트라바이저 호출 커맨드(The dump central processing unit state Ultravisor Call command)는 성공적으로 실행된 울트라바이저 호출 초기화 커맨드(a successfully executed initialize Ultravisor Call command)를 발행하지 않은 구성에서 발행되었다.

[0064] [0064] 0005 hex: 중앙 처리 장치 상태 덤프 울트라바이저 호출 커맨드가 선택 값(예: 50 hex)과 같지 않은 길이로 발행되었다.

[0065] [0065] 0021 hex: 보안 중앙 처리 장치 컨텍스트 핸들은 정의된 보안 중앙 처리 장치에 대한 유효한 핸들이 아니다.

[0066] [0066] 0101 hex: 구성 덤프 개시 울트라바이저 호출이 명시된 보안 중앙 처리 장치를 포함하는 보안 구성에서 완료되지 않았다.

[0067] [0067] 0102 hex: 명시된 중앙 처리 장치에 대한 중앙 처리 장치 상태가 이미 덤프되었다.

[0068] [0068] 0103 hex: 명시된 보안 중앙 처리 장치가 현재 실행 중이다.

[0069] [0069] 0104 hex: 보안 중앙 처리 장치 덤프 영역에 액세스하려고 할 때 액세스 예외가 인식되었다.

[0070] [0070] 보안 중앙처리장치 덤프 영역의 포맷의 일례가 도 4b를 참조하여 설명된다. 일 예로서, 보안 중앙 처리

장치 덤프 영역(420)은 다음을 포함한다:

[0071] [0071] 보안 중앙 처리 장치 덤프 영역 버전(422): 이 필드(예: 바이트 0-3)는 부호 없는 이진 정수(예: 32비트)를 포함하며, 이 값은 울트라바이저에 의해 기록된 보안 중앙 처리 장치 덤프 영역의 버전을 명시한다. 버전 번호는 보안 중앙 처리 장치 덤프 영역에 있는 다른 필드들의 레이아웃을 정의한다.

[0072] [0072] 보안 중앙 처리 장치 덤프 영역 길이(424): 이 필드(예: 바이트 4-7)는 부호 없는 이진 정수(예: 32비트)를 포함하며, 이 값은 울트라바이저에 의해 보안 중앙 처리 장치 덤프 영역에 기록된 데이터의 바이트 수를 명시한다.

[0073] [0073] 초기화 벡터(426): 이 필드(예를 들어, 바이트 8-19)는, 예를 들어, 보안 중앙 처리 장치 덤프 영역의 AES 256-GCM(Galois/Counter 모드) 암호화를 수행할 때 울트라바이저에 의해 사용되는 고유한 초기화 벡터를 포함한다.

[0074] [0074] 일반 레지스터들(428): 이 필드(예를 들어, 바이트 32-159)는 명시된 보안 중앙 처리 장치에 대한 선택 일반 레지스터들(예를 들어, 일반 레지스터들 0 내지 15)의 내용들을 포함한다.

[0075] [0075] 현재 프로그램 상태 워드(430): 이 필드(예를 들어, 바이트 160-175)는 명시된 보안 중앙 처리 장치에 대한 현재 프로그램 상태 워드의 내용들을 포함한다.

[0076] [0076] 프리픽스(432): 이 필드(예를 들어, 바이트 184-187)는 명시된 보안 중앙 처리 장치에 대한 프리픽스 레지스터의 선택 비트들(예를 들어, 비트 32-63)을 포함한다.

[0077] [0077] 부동 소수점 제어 레지스터(434): 이 필드(예를 들어, 바이트 188-191)는 명시된 보안 중앙 처리 장치에 대한 부동 소수점 제어 레지스터를 포함한다.

[0078] [0078] TOD(Time-of-Day) 프로그래밍 가능 레지스터(436): 이 필드(예를 들어, 바이트 196-199)는 명시된 보안 중앙 처리 장치에 대한 타임-오브-데이 프로그래 가능 레지스터를 포함한다.

[0079] [0079] 중앙 처리 장치 타이머(438): 이 필드(예를 들어, 바이트 200-207)는 명시된 보안 중앙 처리 장치에 대한 중앙 처리 장치 타이머를 포함한다.

[0080] [0080] 클록 비교기(440): 이 필드(예를 들어, 바이트 209-215)는 명시된 보안 중앙 처리 장치에 대한 클록 비교기의 선택 비트들(예를 들어, 비트 0-55)을 포함한다.

[0081] [0081] 액세스 레지스터들(442): 이 필드(예를 들어, 바이트 224-287)는 명시된 보안 중앙 처리 장치에 대한 선택 액세스 레지스터들(예를 들어, 액세스 레지스터들 0 내지 15)의 내용들을 포함한다.

[0082] [0082] 제어 레지스터들(444): 이 필드(예를 들어, 바이트 288-415)는 명시된 보안 중앙 처리 장치에 대한 선택 제어 레지스터들(예를 들어, 제어 레지스터들 0 내지 15)의 내용들을 포함한다.

[0083] [0083] 벡터 레지스터들(446): 이 필드(예를 들어, 바이트 416-927)는 명시된 보안 중앙 처리 장치에 대한 선택 벡터 레지스터들(예를 들어, 벡터 레지스터들 0 내지 31)의 내용들을 포함한다.

[0084] [0084] 보호된 스토리지 지정 레지스터(Guarded Storage Designation Register)(448): 이 필드(예를 들어, 바이트 1448-1455)는 명시된 보안 중앙 처리 장치에 대한 보호된 스토리지 지정 레지스터의 내용들을 포함한다.

[0085] [0085] 보호된 스토리지 섹션 마스크 레지스터(450): 이 필드(예를 들어, 바이트 1456-1463)는 명시된 보안 중앙 처리 장치에 대한 보호된 스토리지 섹션 마스크 레지스터의 내용들을 포함한다.

[0086] [0086] 보호된 스토리지 이벤트 파라미터 목록 주소 레지스터(452): 이 필드(예를 들어, 바이트 1464-1471)는 명시된 보안 중앙 처리 장치에 대한 보호된 스토리지 이벤트 파라미터 목록 주소 레지스터의 내용들을 포함한다.

[0087] [0087] 덤프 플래그들(454): 이 필드(예를 들어, 바이트 1536-1537)는 덤프된 중앙 처리 장치 내용들에 관한 정보를 전달하는 플래그들을 포함한다. 예시적 플래그(예: 비트 0)는 미처리 보안 명령 가로채기 처리 표시기(an outstanding secure instruction interception processing indicator)를 포함한다. 이 플래그가 설정될 때, 부분 명령 결과들(partial instruction results)이 덤프된 중앙 처리 장치 상태에 포함될 수 있다.

[0088] [0088] 보안 중앙 처리 장치 덤프 영역 태그(456): 이 필드(예를 들어, 보안 중앙 처리 장치 덤프 영역의 마지막 16바이트)는 상기 영역의 내용들의 진위(the authenticity)를 확인하는 데 사용되는 인증 태그(the

authentication tag)를 포함한다.

[0089] [0089] 일례에서, 필드들(422-426)는 인증되고 필드들(428-454)는 암호화된다.

[0090] [0090] 중앙 처리 장치 덤프 울트라바이저 커맨드(The dump central processing unit ultravisor command)는 가상 머신의 모든 중앙 처리 장치(또는 그 서브세트)에 대해 실행될 수 있으며 이는 중앙 처리 장치의 암호화된 상태를 생성한다. 가상 머신 상태의 암호화는, 예를 들어, 덤프 키를 사용하여 수행된다. 암호화된 중앙 처리 상태는 일반 레지스터들, 제어 레지스터들, 부동 소수점 레지스터들, 벡터 레지스터들뿐만아니라, 타이머들의 내용들, 부동 소수점 제어 레지스터 및/또는 현재 명령 주소를 포함할 수 있지만 이에 국한되지는 않는다. 또한, 일례에서, 그 것은 암호화되지 않은 영역을 포함한다. 이 영역에는, 데이터 암호화에 사용되는 초기화 벡터뿐만 아니라, 나중에 덤프 데이터의 암호 해독 및 해석을 용이하게 하기 위한 길이 및 버전 표시도, 예들로서, 저장된다.

[0091] [0091] 이 커맨드는 명시된 보안 중앙 처리 장치(또는 다른 프로세서 장치)의 암호화된 내용들을 덤프하는 데 사용된다. 보안 중앙처리장치 덤프 영역은, 예를 들어, 평문 헤더(a plain text header)를 포함하며, 이는 버전 정보, 암호화된 데이터의 크기, 중앙처리장치 상태의 암호화에 사용되는 초기화 벡터 등을 포함한다. 중앙 처리 장치 상태는, 예를 들어, 256비트 덤프 키를 갖는 AES-GCM을 사용하여 암호화된다. 암호화된 중앙처리장치 덤프 영역에 대한 인증 태그는 암호화된 영역 바로 뒤에 저장된다. 보안 중앙 처리 장치 덤프 영역의 길이는, 예를 들어, 질의 울트라바이저 호출 커맨드(a query Ultravisor Call command)에 의해 반환된 보안 중앙 처리 장치 스토리지 길이보다 작거나 같다.

[0092] [0092] 구성 덤프 개시 울트라바이저 호출 커맨드(An initiate configuration dump Ultravisor Call command)는 중앙 처리 장치 상태 덤프 울트라바이저 호출 커맨드(a dump central processing unit state Ultravisor Call command)를 발행하기 전에 성공적으로 완료되어야 한다; 그렇지 않으면 선택 응답 코드(예: 0101 hex)가 저장된다. 만일 중앙 처리 장치 상태 덤프 울트라바이저 호출 커맨드가 구성 덤프 개시 울트라바이저 호출 커맨드와 구성 덤프 완료 울트라바이저 호출 커맨드 사이의 명시된 보안 중앙 처리 장치 컨텍스트 핸들에 대해 두 번 이상 발행된다면, 선택 응답 코드(예: 0102 hex)가 저장된다. 만일 명시된 보안 중앙 처리 장치가 현재 실행 중이면, 선택 응답 코드(예: 0103 hex)가 저장된다.

[0093] [0093] 도 2로 돌아가서, 일 예에서, 구성 덤프를 개시하는 것에 기초하여(200), 가상 머신의 메모리가 덤프될 것인지에 관한 결정도 내려진다(222). 만일 메모리가 덤프되어야 한다면, 일 예에서, 덤프하는 단계는 페이지 아웃 프로세스(a page-out process) 동안 발생한다. 예를 들어, 덤프되는 메모리가 현재 보안 구성(예를 들어, 보안 가상 머신)에 매핑되는지에 관한 결정이 내려진다(224). 만일 덤프되는 메모리가 보안 구성에 매핑된다면, 한 예에서, 보안 스토리지로부터의 변환 커맨드(a convert from secure storage command)가 개시된다(226). 예를 들어, 상기 울트라바이저 호출 명령은, 예를 들어, 보안 스토리지로부터의 변환 제어 블록(a convert from secure storage control block)을 명시하는 하이퍼바이저에 의해 발행되고, 이는, 예를 들어, 울트라바이저에 의해 실행될 보안 스토리지로부터의 변환 커맨드를 명시한다. 그러한 제어 블록의 일례가 도 4c를 참조하여 설명된다.

[0094] [0094] 일례에서, 보안 스토리지로부터의 변환 울트라바이저 호출 제어 블록(a convert from secure storage Ultravisor Call control block)(470)은, 예를 들어, 다음 필드들을 포함한다:

[0095] [0095] 길이(472): 이 필드(예를 들어, 바이트 0-1)는, 예를 들어, 부호 없는 이진 정수(예를 들어, 16비트)를 포함하며, 이 값은 제어 블록의 길이를 바이트로 명시한다. 길이는 선택 값(예: 20hex)을 명시하여야 하고; 그렇지 않으면 선택 응답 코드(예: 0005 16진수)가 적용된다.

[0096] [0096] 커맨드 코드(474): 이 필드(예: 바이트 2-3)는 부호 없는 이진 정수(예: 16비트)를 포함하며, 이 값은 보안 스토리지로부터의 변환 울트라바이저 호출 커맨드를 위한 커맨드 코드를 명시한다.

[0097] [0097] 응답 코드(476): 이 필드(예: 바이트 4-5)는 부호 없는 이진 정수(예: 16비트)를 포함하며, 이는 응답 코드를 명시한다. 이 값은 연산이 완료될 때 저장된다.

[0098] [0098] 반환 이유 코드(478): 이 필드(예: 바이트 6-7)는 반환 이유 코드를 포함한다. 예를 들어, '0001'hex의 응답 코드가 저장될 때, 예를 들어,'0000'hex의 반환 이유 코드가 저장되며, 이 때 메모리의 명시된 유닛(예를 들어, 스토리지의 블록)의 보안 속성이 수정된다. 만일 메모리의 명시된 유닛의 보안속성이 이미 원하는 상태로 설정되어 있다면, 반환 이유 코드, 예를들어, '0001'이 저장된다.

[0099] [0099] 호스트 절대 주소(480): 오른쪽에 추가된 선택 수, 예를 들어 12개의 0 비트들을 갖는 이 필드(예를 들어, 바이트 32-39의 비트 0-51)는 비보안 스토리지로 변환할 메모리 유닛의 절대 주소(예를 들어, 4KB 스토리지 블록)을 형성한다.

[0100] [00100] 만일 특별 조건이 존재한다면, 선택 코드(예를 들어, 0001 hex) 이외의 응답 코드가 울트라바이저 호출 제어 블록에 저장된다. 보안 스토리지로부터의 변환 울트라바이저 호출을 위한 특별 조건들은, 예를 들어, 다음과 같다:

[0101] [00101] 0003 hex: 보안 스토리지로부터의 변환 울트라바이저 호출 커맨드(The convert from secure storage Ultravisor Call command)가 성공적으로 실행된 초기화 울트라바이저 호출 커맨드를 발행하지 않은 구성에서 발행되었다.

[0102] [00102] 0005 hex: 보안 스토리지로부터의 변환 울트라바이저 호출 커맨드가 선택 값(예: 20 hex)과 같지 않은 길이로 실행되었다.

[0103] [00103] 0030 hex: 홈 주소 공간 제어 엘리멘트(Home address space control element)는, 예를 들어, 1로 설정된 R 비트(예를 들어, 비트 58)를 갖는다.

[0104] [00104] 0031 hex: 변환 예외(Translation exception)는 구성 변수 스토리지 영역(configuration variable storage area)에 액세스하려 할 때 발생했다.

[0105] [00105] 0032 hex: 보안 게스트 변수 스토리지 영역은 구역-프레임 절대 주소 또는 세그먼트-프레임 절대 주소(a region-frame absolute address or a segment-frame absolute address)로 변환되는 가상 주소를 포함한다.

[0106] [00106] 0103 hex: 호스트 절대 주소는 호스트 구성에서 이용할 수 없다.

[0107] [00107] 0104 hex: 호스트 절대 주소는 울트라바이저 스토리지이다.

[0108] [00108] 0105 hex: 대응 호스트 가상 주소가 보안 스토리지로부터 너무 많이 변환되었다.

[0109] [00109] 일례로서, 보안 스토리지로부터의 변환 울트라바이저 커맨드는 보안 메모리를 덤프하는 데 사용되며, 여기서 덤프되는 메모리 유닛들(예를 들어, 메모리 페이지들)은, 예를 들어, 페이징 키를 사용하여 암호화되어 저장된다. 보안 스토리지로부터의 변환 울트라바이저 커맨드는 보안 구성과 연관된 메모리 유닛(예: 4KB의 저장소 블록)의 보안 속성들을 비-보안 스토리지로 수정한다. 만일 메모리의 명시된 유닛이 이미 비-보안 스토리지로 정의되었다면, 연산은 추가 작업 없이 완료되고 선택 응답 코드(예: '0001' hex) 및 선택 반환 이유 코드(예: '0001' hex)가 저장된다. 만일 메모리 유닛이 정의된-공유 스토리지가 아니면, 메모리 유닛의 내용들로부터 다이제스트(a digest)가 생성된다. 일 예에서, 이 다이제스트는 신뢰된 주체(예: 울트라바이저)에 의해 보존된다. 메모리 유닛의 내용들을 암호화(예: 페이징 키를 사용하여)한 다음, 메모리 유닛은 비-보안으로 만든다. 만일 메모리 유닛이 공유 스토리지로 정의되었다면, 메모리 유닛은 메모리 유닛의 내용들에 대한 수정없이 비-보안 상태가 된다.

[0110] [00110] 만일 명시된 호스트 절대 주소가 신뢰된 개체에 의해 소유되었다면, 메모리의 명시된 유닛은 변경되지 않고 선택된 응답 코드(예: '0104' hex)가 저장된다.

[0111] [00111] 보안 스토리지로부터의 변환한 후, 도 2를 참조하면, 처리는 계속되어 질의(224)로 나아간다. 만일 덤프될 메모리가 현재 보안 구성에 매핑되지 않는다면, 구성 스토리지 상태 덤프 커맨드(a dump configuration storage state command)(또한 진단 메모리 상태 저장 요청이라고도 함)이 개시된다(232). 예를 들어, 울트라바이저 호출 명령은, 예를 들어, 하이퍼바이저에 의해 발행되며, 상기 하이퍼바이저는, 예를 들어, 울트라바이저에 의해 실행될 구성 스토리지 상태 덤프 커맨드(the dump configuration storage state command)를 명시하는 구성 스토리지 상태 덤프 제어 블록(a dump configuration storage state control block)에 포인터를 명시한다. 그러한 제어 블록의 일례가 도 5를 참조하여 설명된다.

[0112] [00112] 일례에서, 구성 스토리지 상태 덤프 울트라바이저 호출 제어 블록(a dump configuration storage state Ultravisor Call control block)(500)은, 예를 들어, 다음 필드들을 포함한다:

[0113] [00113] 길이(502): 이 필드(예를 들어, 바이트 0-1)는, 예를 들어, 부호 없는 이진 정수(예를 들어 16비트)를 포함하며, 이 값은 제어 블록의 길이를 바이트 로 명시한다. 상기 길이는 선택 값(예: 58 hex)을 명시하여야 하고; 그렇지 않으면, 선택 응답 코드(예: 0005 hex)가 적용된다.

[0114] [00114] 커맨드 코드(504): 이 필드(예를 들어, 바이트 2-3)는 부호 없는 이진 정수(예를 들어, 16비트)를 포함하면, 이 값은 구성 스토리지 상태 덤프 울트라버이저 호출 커맨드(the dump configuration storage state Ultravisor Call command)를 위한 커맨드 코드를 명시한다.

[0115] [00115] 응답 코드(506): 이 필드(예: 바이트 4-5)는 부호 없는 이진 정수(예: 16비트)를 포함하며, 이는 응답 코드를 명시한다. 이 값은 연산이 완료될 때 저장된다.

[0116] [00116] 반환 이유 코드(508): 이 필드(예: 바이트 6-7)는 반환 이유 코드를 포함한다. 구성 스토리지 상태 덤프 울트라바이저 호출 커맨드의 경우, 이 필드는, 예를 들어, 0이다.

[0117] [00117] 보안 구성 컨텍스트 핸들(510): 이 필드(예를 들어, 바이트 24-31)는 상태가 덤프되어야 하는 보안 구성을 식별하는 값(예를 들어, 64비트 값)을 포함한다.

[0118] [00118] 보안 구성 스토리지 덤프 영역 원점(512): 선택 수(예를 들어, 12)의 0들이 추가된 이 필드(예를 들어, 바이트 32-39의 비트 0-51)는 보안 구성 스토리지 덤프 영역의 논리 주소(예를 들어, 64비트 논리 주소)를 포함한다.

[0119] [00119] 보안 구성 절대 주소(514): 선택 수(예를 들어, 20)의 0들이 추가된 이 필드(예를 들어, 바이트 40-47의 비트 0-43)는, 예를 들어, 선택 고유 컴포넌트들(예: 튀크 컴포넌트들(tweak components))를 저장하기 위한 스토리지의 256 프레임들 중 제1 프레임의 보안 구성 절대 주소(예를 들어, 64비트 주소)를 포함한다.

[0120] [00120] 만일 특별 조건이 존재한다면, 선택 코드(예를 들어, 0001 hex) 이외의 응답 코드가 울트라바이저 호출 제어 블록에 저장된다. 구성 스토리지 상태 덤프 울트라바이저 호출(the dump configuration storage state Ultravisor Call)을 위한 특별 조건들은 다음과 같다:

[0121] [00121] 0003 hex: 구성 스토리지 상태 덤프 울트라바이저 호출 커맨드는 성공적으로 실행된 초기화 울트라바이저 호출 커맨드를 발행하지 않은 구성에서 발행되었다.

[0122] [00122] 0005 hex: 구성 스토리지 상태 덤프 울트라바이저 호출 커맨드가 선택 값(예: 50 hex)과 같지 않은 길이로 발행되었다.

[0123] [00123] 0020 hex: 보안 구성 컨텍스트 핸들이 정의된 보안 구성에 대해 유효한 핸들이 아니다.

[0124] [00124] 0031 hex: 변환 예외가 구성 가상 스토리지 영역들에 액세스하려고 할 때 인식되었다.

[0125] [00125] 0101 hex: 구성 덤프 개시 울트라바이저 호출이 명시된 보안 구성에서 완료되지 않았다.

[0126] [00126] 0102 hex: 액세스 예외가 보안 구성 스토리지 덤프 영역에 액세스할 때 인식되었다.

[0127] [00127] 0103 hex: 보안 구성 절대 주소는, 예를 들어, 보안 구성 스토리지 길이보다 크다.

[0128] [00128] 만일 특별 선택 고유 컴포넌트(예: 특별 튀크 컴포넌트)가 저장될 때, 선택 바이트들(예: 바이트 14-15)는 메모리의 대응 유닛(예: 4K 바이트 스토리지 블록)을 해석하는 방법에 관한 정보를 제공하는 플래그들을 포함한다. 예시적 플래그들은 다음을 포함한다:

[0129] [00129] 비트설명

[0130] [00130] 15 제로 페이지

[0131] [00131] 메모리의 대응 유닛(4K 바이트 스토리지 블록)은 보안 구성에 의해 저장되지 않았으며 그 내용들은 0들(또는 다른 선택 값)으로 가정될 수 있다.

[0132] [00132] 14 공유 페이지

[0133] [00133] 메모리의 대응 유닛(4K 바이트 스토리지 블록)은 공유 스토리지로 정의되며 암호화되지 않는다.

[0134] [00134] 13 페이지 매핑됨

[0135] [00135] 메모리의 대응 유닛(4K 바이트 스토리지 블록)은 보안 페이지에 대한 유효한 매핑을 갖는다.

[0136] [00136] 구성 스토리지 상태 덤프 커맨드(The dump configuration storage state command)는 주어진 양의 게스트 메모리에 대해 페이지당 선택 고유 컴포넌트 값들(예: 튀크 컴포넌트 값들) 및 상태 플래그들을 덤프한다.

[0137] [00137] 일례에서, 구성 스토리지 상태 덤프 울트라바이저 호출 커맨드(the dump configuration storage state

Ultravisor Call command)는, 예를 들어 256 선택 고유 컴포넌트 값들(예: 튀크 컴포넌트 값들)을 보안 구성 절대 주소에 의해 명시된 메모리 유닛(예: 4K바이트 스토리지 블록)에 대한 선택 고유 값(예: 튀크)으로 시작하는 보안 구성 덤프 영역에 저장한다. 명시된 구성 덤프 영역의 크기는, 예를 들어, 울트라바이저 정보 질의 울트라바이저 호출 커맨드(a query ultravisor information Ultravisor Call command)로부터 획득될 수 있다.

[0138] [00138] 구성 스토리지 상태 덤프 울트라바이저 호출(The dump configuration storage state Ultravisor Call)은 보안 구성 스토리지의 메모리 유닛(예: 4K바이트 블록들)의 내용들을 암호화하거나 해독할 때 사용되는 선택 고유 값들(예: 튀크 값들)을 유도하는 데 사용되는, 예를 들어, 16바이트 선택 고유 컴포넌트 값들(예: 튀크 컴포넌트 값들)을 덤프하는 데 사용된다. 선택 고유 컴포넌트 값들은 보안 구성 내에서, 예를 들어, 절대 주소 0에서 시작하는 메모리 유닛에 대응하는 값으로 시작하는 순서대로 출력된다. 만일, 예를 들어, 선택 고유 컴포넌트 값의 바이트 0-3이, 예를 들어, 'FFFFFFFF' hex이면, 상기 메모리 유닛은 명시된 선택 고유 값을 갖지 않고, 예를 들어, 상기 선택 고유 컴포넌트 값의 바이트 14-15는 메모리의 대응 유닛이 해석되는 방법에 관한 많은 정보를 제공하는 플래그들을 포함한다. 만일 선택 고유 컴포넌트 값이 다른 값이면, 선택 고유 컴포넌트 값과, 예를 들어, 구성 덤프 완료 울트라바이저 호출(a complete configuration dump Ultravisor Call)로부터 획득된 해독된 선택 고유 값 논스(nonce)(예: Tweak nonce)의 OR이 AES-XTS 연산을 위한 선택 고유 값으로 사용되어야 한다.

[0139] [00139] 도 2로, 그리고 특히, 질의 단계들(220, 222)로 돌아가면, 만일 덤프할 중앙 처리 장치가 더 이상 없다면 또는 만일 덤프할 메모리가 더 이상 없다면, 구성 덤프 완료 호출(a complete configuration dump call)(또한 완료 요청이라고도 함)이 이루어진다(240). 예를 들어, 상기 울트라바이저 호출 명령은, 예를 들어, 하이퍼바이저에 의해 발행되며, 상기 하이퍼바이저는 울트라바이저에 의해 실행될 구성 덤프 완료 커맨드를 명시하는 구성 덤프 완료 제어 블록을 명시한다. 그러한 제어 블록의 일례가 도 6을 참조하여 설명된다.

[0140] [00140] 일례에서, 구성 덤프 완료 울트라바이저 호출 제어 블록(a complete configuration dump Ultravisor Call control block)(600)은, 예를 들어, 다음 필드들을 포함한다:

[0141] [00141] 길이(602): 이 필드(예를 들어, 바이트 0-1)는, 예를 들어, 부호 없는 이진 정수(예를 들어 16비트)를 포함하며, 이 값은 제어 블록의 길이를 바이트로 명시한다. 상기 길이는 선택 값(예: 64 + N 바이트 값)을 명시해야 하고; 그렇지 않으면 선택 응답 코드(예: 0005 hex)가 적용된다. 값 N은, 예를 들어, 울트라바이저 정보 질의 울트라바이저 호출 커맨드(a query ultravisor information Ultravisor Call command)에 의해 반환된 구성 마무리 덤프 길이(the configuration finalize dump length)로부터 획득될 수 있다.

[0142] [00142] 커맨드 코드(604): 이 필드(예를 들어, 바이트 2-3)는 부호 없는 이진 정수(예를 들어, 16비트)를 포함하며, 이 값은 구성 상태 덤프 완료 울트라바이저 호출 커맨드(the complete configuration state dump Ultravisor Call command)를 위한 커맨드 코드를 명시한다.

[0143] [00143] 응답 코드(606): 이 필드(예: 바이트 4-5)는 부호 없는 이진 정수(예: 16비트)를 포함하며, 이는 응답 코드를 명시한다. 이 값은 연산이 완료될 때 저장된다.

[0144] [00144] 반환 이유 코드(608): 이 필드(예: 바이트 6-7)는 반환 이유 코드를 포함한다. 구성 덤프 완료 울트라바이저 호출 커맨드의 경우, 이 필드는, 예를 들어, 0이다.

[0145] [00145] 보안 구성 컨텍스트 핸들(609): 이 필드(예를 들어, 바이트 24-31)는 덤프가 완료되어야 할 보안 구성을 식별하는 값(예를 들어, 64비트 값)을 포함한다.

[0146] [00146] 키 유도 시드(Key Derivation Seed)(610): 이 필드(예를 들어, 바이트 64-127)는 중앙 처리 장치 상태뿐만 아니라 이 제어 블록의 데이터를 암호화하는 데 사용되는 덤프 키를 유도하는 데 사용되는 시드를 포함한다.

[0147] [00147] 초기화 벡터(612): 이 필드(예를 들어, 바이트 128-143)는 이 울트라바이저 호출에 의해 저장된 데이터를 암호화할 때 사용되는 초기화 벡터를 포함한다.

[0148] [00148] 튀크 논스(Tweak Nonce)(일명, 선택 고유 논스)(614): 이 필드(예를 들어, 바이트 144-159)는 선택 고유 값들(예를 들어, 튀크 값들)에 사용되는 논스 값을 포함한다. 이 값은 메모리 유닛(예: 4K바이트 스토리지 블록)을 암호화하는 데 사용되는 선택 고유 값들을 생성하기 위해 각각의 비-특별 선택 고유 컴포넌트(예: 조정 컴포넌트)와 OR 되어야(Ored) 한다.

[0149] [00149] 스토리지 암호화 키 1(616): 이 필드(예를 들어, 바이트 160-191)는 선택고유 값들(예를 들어, 튀크 값

들)을 암호화하는 데 사용되는 제1 AES-256 XTS 키를 포함하며, 이 선택 고유 값들은 이후에 덤프된 메모리 유닛(예: 4K바이트 스토리지 블록)을 해독하는 데 사용된다.

[0150] [00150] 스토리지 암호화 키 2(618): 이 필드(예: 바이트 192-223)는 메모리 유닛(예를 들어, 4K바이트 스토리지 블록들)을 해독하기 위해 암호화된 선택 고유 값(예: 튀크 값)과 함께 사용되는 제2 AES-256 XTS 키를 포함한다. 한 예에서, 스토리지 암호화 키 2와 결합된 스토리지 암호화 키 1은 페이징 키이다.

[0151] [00151] 인증 태그(620): 이 필드(예를 들어, 바이트 256-271)는 인증된 값들(예를 들어, 바이트 64-143)을 추가로 인증된 데이터로 사용하고 태그의 나머지 부분을 암호화된 필드들(예: 바이트 144-255)로부터 계산함으로써 유도된 AES-GCM 인증 태그를 포함한다.

[0152] [00152] 만일 특별 조건이 존재한다면, 선택 코드 이외의 응답 코드(예: 0001 hex)가 울트라바이저 호출 제어 블록에 저장된다. 구성 덤프 완료 울트라바이저 호출을 위한 특별 조건들은, 예를 들어, 다음을 포함한다:

[0153] [00153] 0003 hex: 구성 덤프 완료 울트라바이저 호출 커맨드(The complete configuration dump Ultravisor Call command)가 성공적으로 실행된 초기화 울트라바이저 호출 커맨드를 발행하지 않은 구성에서 발행되었다.

[0154] [00154] 0005 hex: 구성 덤프 완료 울트라바이저 호출 커맨드가 선택 값(예: 130 hex)과 같지 않은 길이로 발행되었다.

[0155] [00155] 0020 hex: 보안 구성 컨텍스트 핸들이 정의된 보안 구성에 대한 유효한 핸들이 아니다.

[0156] [00156] 0101 hex: 구성 덤프 개시 울트라비아이저 호출이 명시된 보안 구성에서 완료되지 않았다.

[0157] [00157] 구성 덤프 완료 울트라바이저 호출(The complete configuration dump Ultravisor Call)은 명시된 보안 구성에 대한 덤프 처리를 완료하는 데 사용된다. 이 호출은 고객에 의해서 덤프를 해독하고 무결성을 체크하는 데 사용되는 데이터를 반환한다. 또한 상기 호출은 덤프 프로세스를 완료하므로 구성 덤프 개시 호출을 통해 새로운 덤프 프로세스의 개시를 할 수 있게 한다. 예를 들어, 상기 호출은 가상 머신 소유자가 덤프 키 자체를 계산할 수 있도록 덤프 키를 생성하는 데 사용된 키 유도 시드(the key derivation seed)를 포함한다. 또한, 상기 호출은 반환된 데이터의 암호화된 부분을 암호화하는 데 사용되는 초기화 벡터와 인증되고 암호화된 데이터의 내용들을 확인하는 데 사용되어야 하는 인증 태그를 포함한다. 상기 암호화된 부분들은, 예를 들어, 상기 메모리 상태 덤프 호출로부터 선택 고유 갑들의 각각으로 OR되어야 하는 선택된 고유 값 논스(the selected unique value nonce)뿐만아니라, 이들 선택 고유 값들을 암호화하는 데 사용되는 AES XTS(Advanced Encryption Standard XEX Tweakable Block Cipher with Ciphertext Stealing) 키를 포함한다.

[0158] [00158] 본 명세서에 설명된 바와 같이, 보안 가상 머신은 신뢰할 수 없는 주체가 덤프된 가상머신의 상태를 읽을 수 있기 전에 덤프를 암호화함으로써 덤프된 가상 머신 상태의 보안을 유지하면서 덤프될 수 있다. 보안 가상 머신의 덤프를 보안으로 용이하게 수행하기 위해, 덤프 애플리케이션 프로그래밍 인터페이스가 채용된다. 상기 인터페이스는 복수의 호출들을 사용하고 각 호출은 제어 블록을 가리키며, 제어블록의 예들이 여기에서 설명된다. 본 명세서에 설명된 각각의 제어 블록/영역은 하나 또는 그 이상의 실시예들에서 추가의, 더 적은 수의 및/또는 다른 필드들을 포함할 수 있다. 또한, 각 필드는 제어 블록/영역 내에서 서로 다른 위치에 있을 수 있고 및/또는 크기나 데이터 유형이 다를 수 있다. 또한, 제어 블록/영역은 하나 또는 그 이상의 유보된 필드들을 포함할 수 있다. 다양한 변형들이 가능하다.

[0159] [00159] 하나 또는 그 이상의 실시예들에서, 가상 머신의 메모리는 다수의 기술들을 사용하여 덤프될 수 있다. 예를 들어, 한 가지 기술은 덤프 키를 사용하여 메모리를 직접 암호화하는 것을 포함한다. 중앙 처리 장치 메모리 암호화라고 하는, 이 기술을 사용하여, 보안 가상 머신의 페이지가 암호화된다. 보안된 가상 머신의 키와 하이퍼바이저의 키가 다르기 때문에, 만일 하이퍼바이저가 결과를 읽거나 쓰게 되면, 그 것은 랜덤 데이터가 된다. 최적의 보안을 위해, 암호화 키는 중앙 처리 장치의 메모리 컨트롤러를 떠나지 않으므로 그 것은 보안 가상 머신의 메모리를 덤프하는 데 사용될 수 없는데, 그 이유는 가상 머신이 현재 실행하고 있는 중앙 처리 장치만이 그 페이지들을 해독할 수 있기 때문이다. 또한, 상기 키는 보안 가상 머신이 중지될 때 클리어된다(cleared).

[0160] [00160] 이러한 보호 메커니즘은 보안 가상 머신 메모리를 덤프하는 하나의 기술을 필요로 하는데, 그 이유는 보안 가상 머신 메모리에 액세스하는 데 사용되는 키는 일반적으로 내보낼 수 없기 때문이며: 따라서, 내보낼 수 없는 액세스 키(the non-exportable access key)를 대체하기 위해 내보낼 수 있는 특별 덤프 키로 이를 암호화하는 것이다. 만일 페이지가 디스크로 스왑되었다면(swapped), 그 것을 해독하고 재-암호화하기 위해 게스트

의 메모리로 그 페이지를 다시 가져와야 한다. 이로 인해 추가적인 메모리 압력이 발생하고 암호화/암호 해독에 프로세서 사이클들이 소요된다.

[0161] [00161] 하나 또는 그 이상의 실시예들에 따라, 보안 가상 머신의 메모리를 덤프하는 데 사용되는 또 다른 기술은 메모리 보호라고 하며, 이는 하이퍼바이저가 펌웨어에 요청한 후 그 것이 암호화될 때까지 페이지가 속한 보안 가상 머신과 신뢰된 펌웨어(예: 울트라바이저)를 제외하고 보안 가상 머신의 페이지에 대한 액세스를 거부한다. 암호화 후, 상기 페이지는 더 이상 보안 가상 머신에 이용 가능하지 않게 되지만 하이퍼바이저에 의해 디스크에 기록될 수는 있다. 유사하게, 펌웨어에 대한 요청이 암호화된 페이지를 해독하고, 무결성을 체크하며, 보안 가상 머신에 다시 매핑하여 다시 그 것을 액세스할 수 있도록 하는 데 사용된다.

[0162] [00162] 이 보호 메커니즘은 덤프를 위한 스와핑을 위해 페이지를 암호화할 때 사용되는 동일한 키를 사용할 수 있다. 즉, 일단 페이지들이 암호화되면 게스트가 그 페이지들에 다시 액세스하려는 경우에만 그 페이지들을 게스트로 다시 가져오면 된다. 아직 암호화되지 않은 페이지들은 암호화되어야 하지만, 이미 암호화된 페이지들은 그들이 아직 메모리에 있거나 디스크에 있는지 여부에 관계없이 덤프에 직접 기록될 수 있다. 이로 인해 만일 페이지들이 이미 디스크에 페이징되어 있다면, 덤프에 필요한 CPU 사이클들에서 상당한 감소가 초래된다. 나중에 덤프를 검사할 때 그 페이지들을 해독하기 위해 암호화 특정 데이터(Encryption specific data)를 내보내야 한다. 예를 들어, 그러한 데이터에는 메모리의 각 페이지에 대해 선택 고유 값들(예: 튀크 값들)을 포함할 수 있다.

[0163] [00163] 가상 머신 내부의 실패한 커널에 의해 기록된 진단 상태의 저장(a store of diagnostic state)과 대조적으로, 하나 또는 그 이상의 실시예들에서, 본 명세서에 설명된 진단 상태 저장은 하이퍼바이저에 의해 가상 머신 외부에서 생성된다. 그 것은 가상 머신의 소유자에 의해 트리거되거나 하이퍼바이저가 가상 머신에 오류가 발생하여 코드가 정상적으로 실행되고 있지 않는다고 결정할 때 자동으로 트리거된다.

[0164] [00164] 본 발명의 하나 또는 그 이상의 실시예들은 컴퓨터 기술과 불가분하게 연결되어 있으며 컴퓨터 내 처리를 용이하게 하여 성능을 향상시킨다. 보안을 유지하면서 덜 복잡하고 더 적은 처리 사이클들을 사용하며 성능을 향상시키는 방식으로 가상 머신(예: 보안 가상 머신)의 진단 상태를 보안으로 저장함으로써 처리가 용이해진다.

[0165] [00165] 본 발명의 하나 또는 그 이상의 실시예들과 관련된 컴퓨팅 환경 내에서 처리를 용이하게 하는 일 실시예의 추가 세부사항들은 도 7a-7c를 참조하여 설명된다.

[0166] [00166] 도 7a를 참조하면, 일 실시예에서, 가상 머신의 진단 상태(diagnostic state)를 저장하기 위한 적어도 하나의 요청이 획득된다(700). 상기 가상 머신의 저장된 진단 상태를 제공하기 위해 상기 가상 머신의 진단 상태의 저장이, 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 수행된다(702). 상기 저장을 수행하는 단계는, 예를 들어, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체(an untrusted entity)가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함한다(704).

[0167] [00167] 가상 머신(예를 들어, 보안 가상 머신)의 진단 상태는 진단 목적들을 위해 저장되는 가상 머신 상태의 보안을 유지하면서 저장될 수 있다.

[0168] [00168] 예들로서, 상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 메모리의 내용들(contents of memory)을 포함하거나(706) 및/또는 상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 하나 또는 그 이상의 프로세서 레지스터들의 내용들을 포함한다(708).

[0169] [00169] 일 예에서, 상기 가상 머신의 진단 상태의 저장을 개시하기 위해 진단 상태 저장 개시 요청(an initiate store diagnostic state request)이 획득된다(710). 상기 진단 상태 저장 개시 요청은 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태의 적어도 일부를 암호화하는 데 사용될 제1 암호화 키를 획득하는 데 사용된다(712).

[0170] [00170] 일 예에서, 도 7b를 참조하면, 상기 적어도 하나의 요청을 획득하는 단계는 진단 프로세서 상태 저장 요청(a store diagnostic processor state request)을 획득하는 단계를 포함한다(720). 상기 진단 프로세서 상태 저장 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는, 예를 들어, 암호화된 프로세서 내용들을 제공하기 위해 상기 제1 암호화 키를 사용하여 상기 가상 머신의 적어도 하나의 프로세서의 적어도 선택 내용들(at least select contents)을 암호화하는 단계를 포함하고(722), 그리고 상기 암호화된 프로세

서 내용들을 저장하는 단계를 포함한다(723).

[0171] [00171] 일 예에서, 상기 적어도 하나의 요청을 획득하는 단계는 보안 메모리로부터 변환 요청(a convert from secure memory request)을 획득하는 단계를 포함한다(726). 상기 보안 메모리로부터의 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는, 예를 들어, 메모리의 암호화된 내용들(encrypted contents of memory)을 제공하기 위해 제2 암호화 키를 사용하여 암호화되지 않은 상기 가상 머신의 메모리의 적어도 선택 내용들을 암호화하는 단계를 포함하고(730), 그리고 메모리의 상기 암호화된 내용들을 저장하는 단계를 포함한다(732).

[0172] [00172] 일 예로서, 상기 진단 상태의 저장을 수행하는 단계는, 보안 메모리로부터의 상기 변환 요청을 획득하는 것에 기초하여, 예를 들어, 내용들이 저장될 메모리의 유닛(a unit of memory)이 암호화되었는지를 결정하는 단계를 포함하고(736), 그리고 상기 메모리의 유닛이 암호화되지 않았다고 결정하는 것에 기초하여, 상기 제2 암호화 키를 사용하여 상기 메모리의 유닛을 암호화하는 단계를 포함하며, 암호화된 메모리의 유닛들은 재-암호화되지 않는다(738).

[0173] [00173] 암호화되지 않은 내용들을 암호화함으로써(그리고 암호화된 상태를 재-암호화하지 않음으로써) 처리 사이클들과 복잡성이 감소되고, 시스템 성능이 향상된다.

[0174] [00174] 일 예에서, 도 7c를 참조하면, 상기 저장된 진단 상태는 메모리의 암호화된 내용들을 포함한다(750). 또한, 진단 메모리 상태 저장 요청(a store diagnostic memory state request)이 획득되며(752), 그리고 상기 진단 메모리 상태 저장 요청을 획득하는 것에 기초하여, 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 메타데이터가 저장된다(754).

[0175] [00175] 일 예로서, 상기 메타데이터는 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 하나 또는 그 이상의 선택 값들(one or more select values)을 생성하기 위해 사용될 하나 또는 그 이상의 선택 컴포넌트 값들(one or more select component values)을 포함한다(756).

[0176] [00176] 일 예에서, 진단 상태의 저장을 완료하기 위해 완료 요청(a complete request)이 획득되며(760), 그리고 상기 완료 요청을 획득하는 것에 기초하여, 상기 가상 머신의 암호화되어 저장된 진단 상태를 해독하는 데 사용될 데이터가 제공된다(762).

[0177] [00177] 일 예로서, 상기 데이터는 메모리의 암호화된 내용들의 해독(decryption)에 사용되는 제2 암호화 키와 암호화에 사용될 하나 또는 그 이상의 선택 값들을 생성하기 위해 사용되는 논스 값(a nonce value)을 포함한다(770). 또한, 일 예에서, 상기 데이터의 적어도 일부는 암호화된다(772).

[0178] [00178] 다른 변형들 및 실시예들이 가능하다.

[0179] [00179] 본 발명의 실시예들은 많은 유형의 컴퓨팅 환경들에 의해 사용될 수 있다. 본 발명의 하나 또는 그 이상의 실시예들을 포함하고 사용하기 위한 컴퓨팅 환경의 다른 실시예가 도 8a를 참조하여 기술된다. 이 예에서, 컴퓨팅 환경(36)은, 예컨대, 네이티브 중앙 처리 유닛(CPU)(37), 메모리(38), 및, 예를 들어, 하나 또는 그 이상의 버스들(40) 및/또는 다른 접속들을 통해, 서로 연결된 하나 또는 그 이상의 입출력 디바이스들 및/또는 인터페이스들(39)을 포함한다. 예들로서, 컴퓨팅 환경(36)은 미국 뉴욕 주 아몽크에 있는 인터내셔널 비즈니스 머신즈 코포레이션에 의해 공급되는 PowerPC® 프로세서; 미국 캘리포니아 주 팔로 알토에 있는 휼렛 패커드 사에 의해 공급되는 Intel® Itanium® II 프로세서들을 갖는 HP 수퍼돔 및/또는 인터내셔널 비즈니스 머신즈 코포레이션, 휼렛 패커드, Intel Corporation, 오라클, 및/또는 기타 회사들에 의해 공급되는 아키텍처들에 기반하는 기타 머신들을 포함할 수 있다. PowerPC는 적어도 한 국가(관할 구역)에서 인터내셔널 비즈니스 머신즈 코포레이션의 상표 또는 등록 상표이다. Intel 및Itanium은 미국 및 기타 국가들에서 인텔 코포레이션 또는 그 자회사들의 상표 또는 등록 상표이다.

[0180] [00180] 네이티브 중앙 처리 유닛 (37)는 환경 내에서 처리 중에 사용되는 하나 또는 그 이상의 범용 레지스터들 및/또는 하나 또는 그 이상의 특수-목적 레지스터들과 같은 하나 또는 그 이상의 네이티브 레지스터들(41)을 포함한다. 이들 레지스터들은 특정 시점에서 환경 상태를 표시하는 정보를 포함한다.

[0181] [00181] 또한, 네이티브 중앙 처리 유닛(the native central processing unit) (37)는 메모리(38)에 저장된 명령들과 코드를 실행한다. 한 특정 예에서, 중앙 처리 유닛(the central processing unit)는 메모리(38)에 저장된 에뮬레이터 코드(42)를 실행한다. 이 코드는 한 아키텍처에서 구성된 컴퓨팅 환경이 다른 아키텍처를 에뮬레이트를 할 수 있게 한다. 예를 들어, 에뮬레이터 코드(42)를 사용하면 PowerPC 프로세서들, HP 수퍼돔 서버들

또는 다른 서버들과 같은 z/Architecture 명령 세트 아키텍처 이외의 아키텍처들을 기반으로 하는 머신들에서 z/Architecture 명령 세트 아키텍처를 에뮬레이트하고 z/Architecture 명령 세트 아키텍처를 기반으로 개발된 소프트웨어 및 명령들을 실행할 수 있다.

[0182] [00182] 에뮬레이터 코드(42)에 관한 더 상세한 설명들은 도 8b를 참조하여 기술된다. 메모리(38)에 저장된 게스트 명령들(43)은 네이티브 CPU(37)의 아키텍처가 아닌 아키텍처에서 실행되도록 개발된 소프트웨어 명령들(예를 들어, 머신 명령들에 관련되는)을 포함한다. 예를 들어, 게스트 명령들(43)은 z/Architecture 명령 세트 아키텍처에 기반한 프로세서 상에서 실행하도록 설계되었지만, 대신에, 예를 들어, Intel Itanium II 프로세서일 수 있는, 네이티브 CPU (37)상에서 에뮬레이트 될 수 있다. 한 예에서, 에뮬레이터 코드(42)는 메모리(38)로부터 하나 또는 그 이상의 게스트 명령들(43)을 획득하고 획득된 명령들에 대한 로컬 버퍼링을 선택적으로 제공하기 위한 명령 페치 루틴(44)을 포함한다. 또한, 획득된 게스트 명령의 유형을 결정하고 상기 게스트 명령을 하나 또는 그 이상의 대응하는 네이티브 명령들(46)로 변환하기 위한 명령 변환 루틴(45)을 포함한다. 이 변환은, 예를 들어, 상기 게스트 명령에 의해 수행될 함수를 식별하는 것과 그 함수를 수행하기 위한 네이티브 명령(들)을 선택하는 것을 포함한다.

[0183] [00183] 또한, 에뮬레이터 코드(42)는 네이티브 명령들이 실행되도록 하는 에뮬레이션 컨트롤 루틴(47)을 포함한다. 에뮬레이션 컨트롤 루틴(47)은 네이티브 CPU (37)로 하여금 하나 또는 그 이상의 이전에 획득 된 게스트 명령들을 에뮬레이트하는 네이티브 명령들의 루틴을 실행하게 하고, 그러한 실행의 종료 시에, 다음 게스트 명령 또는 일 군의 게스트 명령들을 획득하는 것을 에뮬레이트 하기 위해 상기 명령 페치 루틴에 컨트롤을 반환(return)하게 할 수 있다. 네이티브 명령들(46)의 실행은 메모리(38)로부터 레지스터로 데이터를 로드하는 단계; 레지스터로부터 데이터를 메모리에 다시 저장하는 단계; 또는 변환 루틴에 의해 결정되는, 어떤 유형의 산술 또는 논리 연산을 수행하는 단계를 포함할 수 있다.

[0184] [00184] 각 루틴은, 예를 들어, 소프트웨어로 구현되고, 상기 소프트웨어는 메모리에 저장되며, 네이티브 중앙처리 유닛(37)에 의해 실행된다. 다른 예들에서, 하나 또는 그 이상의 루틴들 또는 연산들은 펌웨어, 하드웨어, 소프트웨어 또는 이들의 조합으로 구현된다. 에뮬레이트된 프로세서의 레지스터들은 상기 네이티브 CPU의 레지스터들(41)을 사용하여 또는 메모리(38) 내의 위치들을 사용하여 에뮬레이트 될 수 있다. 실시 예들에서, 게스트 명령들(43), 네이티브 명령들 (46) 및 에뮬레이터 코드(42)는 동일한 메모리 내에 상주하거나 또는 다른 메모리 디바이스들 사이에서 분산될 수 있다.

[0185] [00185] 에뮬레이트될 수 있는 예시적 명령은, 본 발명의 일 실시예에 따라, 본 명세서에 기술된 울트라바이저 호출들(the Ultravisor Calls)을 포함한다. 또한, 본 발명의 하나 또는 그 이상의 실시예들에 따라, 본 발명의 다른 명령들, 커맨드들, 함수들(functions), 연산들(operations) 및/또는 하나 또는 그 이상의 실시예들이 에뮬레이트될 수 있다.

[0186] [00186] 전술한 컴퓨팅 환경들은 사용될 수 있는 컴퓨팅 환경들의 예들일 뿐이다. 파티션되지 않은 환경들, 파티션된 환경들, 클라우드 환경들 및/또는 에뮬레이트된 환경들을 포함하되 이에 국한되지 않는 다른 환경들이 사용될 수 있다; 실시예들은 어느 하나의 환경으로 제한되지 않는다. 여기에서 컴퓨팅 환경들의 다양한 예들이 설명되지만, 본 발명의 하나 또는 그 이상의 실시예들은 많은 유형의 환경들과 함께 사용될 수 있다. 여기에 제공된 컴퓨팅 환경들은 단지 예들일 뿐이다.

[0187] [00187] 각각의 컴퓨팅 환경은 본 발명의 하나 또는 그 이상의 실시예들을 포함하도록 구성될 수 있다.

[0188] [00188] 본 발명의 하나 또는 그 이상의 실시 예들은 클라우드 컴퓨팅과 관련될 수 있다.

[0189] [00189] 본 명세서는 클라우드 컴퓨팅에 관해서 상세한 설명들을 포함하지만, 여기서 기술된 그러한 가르침들의 구현은 클라우드 컴퓨팅 환경에만 한정되는 것은 아님을 이해하여야 한다. 오히려, 본 발명의 실시 예들은 지금 알려져 있거나 또는 나중에 개발될 모든 다른 유형의 컴퓨팅 환경과 함께 구현될 수 있다.

[0190] [00190] 클라우드 컴퓨팅은, 최소한의 관리 노력 또는 서비스 제공자와의 상호작용으로 빠르게 제공되고 해제될 수 있는, 구성 가능한(configurable) 컴퓨팅 자원들(예를 들어, 네트워크, 네트워크 대역폭, 서버, 처리, 메모리, 스토리지, 애플리케이션, 가상 머신, 및 서비스)의 공유 풀에 대한 편리한 주문형(on-demand) 네트워크 액세스를 가능하게 하는 서비스 전달 모델이다. 이 클라우드 모델은 적어도 5가지의 특성(characteristics), 적어도 3가지 서비스 모델(service models), 및 적어도 4가지 배치 모델(deployment models)을 포함할 수 있다.

[0191] [00191] 특성들은 다음과 같다:

[0192] [00192] 주문형 셀프-서비스(On-demand self-service): 클라우드 소비자는, 서비스 제공자와의 인적 상호작용을 필요로 하지 않고 필요한 만큼 자동적으로, 서버 시간(server time) 및 네트워크 스토리지 같은 컴퓨팅 기능들을 일방적으로 제공(provision)할 수 있다.

[0193] [00193] 광역 네트워크 액세스(Broad network access): 혼성의 씬 또는 씩 클라이언트 플랫폼들(heterogeneous thin or thick client platforms)(예를 들어, 모바일폰들, 랩탑들, 및 PDA들)에 의한 사용을 장려하는 표준 메커니즘들을 통해 액세스되는 기능들을 네트워크를 통해서 이용할 수 있다.

[0194] [00194] 자원 풀링(Resource pooling): 제공자의 컴퓨팅 자원들은 멀티-테넌트 모델(a multi-tenant model)을 이용하여, 각기 다른 물리적 및 가상 자원들을 요구(demand)에 따라 동적으로 할당 및 재할당하면서, 다수의 소비자들에게 서비스할 수 있도록 풀에 넣어둔다(pooled). 소비자는 일반적으로 제공된 자원들의 정확한 위치를 컨트롤할 수 없거나 그에 대한 지식이 없지만 더 높은 추상 레벨에서(예를 들어, 국가, 주, 또는 데이터센터) 위치를 명시할 수 있다는 점에서 위치 독립성이 있다.

[0195] [00195] 기민한 탄력성(Rapid elasticity): 용량들(capabilities)이 기민하게 탄력적으로 제공되어 (어떤 경우엔 자동으로) 신속히 규모를 확장할 수도 있고(scale out) 그리고 탄력적으로 해제되어 신속히 규모를 축소할 수도 있다(scale in). 소비자에게 제공할 수 있는 가능성이 종종 무제한이고 언제든지 원하는 수량으로 구매할 수 있는 것처럼 보인다.

[0196] [00196] 측정 가능한 서비스(Measured service): 클라우드 시스템들은 자원 사용을 자동으로 컨트롤하고 최적화하는데, 서비스의 유형(예를 들어, 스토리지, 처리, 대역폭, 및 활성 사용자 계정)에 적절한 추상화 레벨에서(at some level of abstraction) 계측 기능을 이용하여서 그렇게 한다. 자원 사용량은 모니터되고, 컨트롤되고, 그리고 보고될 수 있으며 이로써 이용하는 서비스의 제공자와 사용자 모두에게 투명성을 제공한다.

[0197] [00197] 서비스 모델들(Service Models)은 다음과 같다:

[0198] [00198] 소프트웨어 서비스(Software as a Service)(SaaS): 소비자에게 제공되는 서비스는 클라우드 인프라스트럭처 상에서 실행되는 제공자의 애플리케이션들을 사용하게 해주는 것이다. 애플리케이션들은 웹 브라우저(예를 들어, 웹기반 이메일) 같은 씬(thin) 클라이언트 인터페이스를 통해 여러 클라이언트 장치들에서 액세스 가능하다. 소비자는 네트워크, 서버들, 운영 체제들, 스토리지, 또는 개별 애플리케이션 능력들을 포함하는 하부 클라우드 인프라스트럭처를 관리하거나 컨트롤하지 않는다. 단, 제한된 사용자-특화 애플리케이션 구성 세팅들은 예외로서 가능하다.

[0199] [00199] 플랫폼 서비스(Platform as a Service)(PaaS): 소비자에게 제공되는 서비스는 제공자에 의해 지원되는 프로그래밍 언어들 및 도구들을 이용하여 생성된 소비자-생성 또는 획득 애플리케이션들을 클라우드 인프라스트럭처에 배치하게 해주는 것이다. 소비자는 네트워크, 서버들, 운영 체제들, 또는 스토리지를 포함하는 하부 클라우드 인프라스트럭처를 관리하거나 컨트롤하지 않는다. 그러나 배치된 애플리케이션들에 대해서 그리고 가능한 경우 애플리케이션 호스팅 환경 구성들에 대해서 컨트롤할 수 있다.

[0200] [00200] 인프라스트럭처 서비스(Infrastructure as a Service)(IaaS): 소비자에게 제공되는 서비스는 처리, 스토리지, 네트워크, 및 기타 기본 컴퓨팅 자원들을 제공하여 주는 것이며, 여기서 소비자는 임의의 소프트웨어를 배치 및 실행할 수 있고, 이 소프트웨어에는 운영 체제들과 애플리케이션들이 포함될 수 있다. 소비자는 하부 클라우드 인프라스트럭처를 관리하거나 컨트롤하지 않지만, 운영 체제들, 스토리지, 배치된 애플리케이션들, 및 가능한 경우 선택된 네트워킹 컴포넌트들의 제한적인 컨트롤(예를 들어, 호스트 방화벽들)에 대하여 컨트롤할 수 있다.

[0201] [00201] 배치 모델들(Deployment Models)은 다음과 같다:

[0202] [00202] 사설 클라우드(Private cloud): 클라우드 인프라스트럭처는 오직 한 조직(an organization)을 위해서 운영되고, 그 조직 또는 제3자에 의해 관리될 수 있으며 옥내(on-premises) 또는 옥외(off-premises)에 위치할 수 있다.

[0203] [00203] 커뮤니티 클라우드(Community cloud): 클라우드 인프라스트럭처는 여러 조직들에 의해 공유되고 관심사(예를 들어, 선교, 보안 요건, 정책, 및 규정 준수 심사)를 공유하는 특정 커뮤니티를 지원하며, 여러 조직들 또는 제3자에 의해 관리될 수 있으며 옥내(on-premises) 또는 옥외(off-premises)에 위치할 수 있다.

[0204] [00204] 공공 클라우드(Public cloud): 클라우드 인프라스트럭처는 일반 대중 또는 대규모 산업 집단에서 이용

할 수 있으며 클라우드 서비스를 판매하는 조직이 소유한다.

[0205] [00205] 하이브리드 클라우드(Hybrid cloud): 클라우드 인프라스트럭처는 둘 또는 그 이상의 클라우드들(사설, 커뮤니티, 또는 공공)이 혼합된 구성이며, 이들은 고유한 주체들로 있지만 데이터 및 애플리케이션 이식가능성(portability)을 가능하게 해주는 표준화된 또는 소유권 있는 기술(예를 들어, 클라우드들 사이의 부하 균형을 위한 클라우드 버스팅(cloud bursting))에 의해 서로 결합되어 있다.

[0206] [00206] 클라우드 컴퓨팅 환경은 상태 비보존(statelessness), 낮은 결합(low coupling), 모듈 방식(modularity), 및 의미적 상호운용성(semantic interoperability)에 집중하는 서비스를 지향한다. 클라우드 컴퓨팅의 중심에는 상호 연결된 노드들의 네트워크를 포함하는 인프라스트럭처가 있다.

[0207] [00207] 이제 도 9를 참조하면, 예시적인 클라우드 컴퓨팅 환경(50)이 도시된다. 도시된 바와 같이, 클라우드 컴퓨팅 환경(50)은 예를 들어 개인 휴대 정보 단말기(PDA) 또는 휴대폰(54A), 데스크탑 컴퓨터(54B), 랩탑 컴퓨터(54C), 및/또는 자동차용 컴퓨터 시스템(54N)과 통신할 수 있는 것과 같이, 클라우드 소비자가 사용하는 로컬 컴퓨팅 디바이스가 하나 또는 그 이상의 클라우드 컴퓨팅 노드들(52)을 포함한다. 노드들(52)은 서로 통신할 수 있다. 이들은 여기에 기술된 바와 같은 사설, 커뮤니티, 공공, 또는 하이브리드 클라우드들 또는 이들의 조합 등의 하나 또는 그 이상의 네트워크들에서 물리적으로 또는 가상으로 그룹화될 수 있다(도시되지 않음). 이것은 클라우드 소비자가 로컬 컴퓨팅 장치 상에 자원들을 유지할 필요가 없게 클라우드 컴퓨팅 환경(50)이 인프라스트럭처, 플랫폼들 및/또는 소프트웨어를 서비스로서 제공할 수 있게 해준다. 도 9에 도시된 컴퓨팅 장치들(54A-N)의 유형들은 단지 예시의 목적으로 기술한 것이며 컴퓨팅 노드들(52)과 클라우드 컴퓨팅 환경(50)은 모든 유형의 네트워크 및/또는 네트워크 주소지정가능 연결을 통해서 (예를 들어, 웹 브라우저를 사용하여) 모든 유형의 컴퓨터화된 디바이스와 통신할 수 있다는 것을 이해해야 한다.

[0208] [00208] 이제 도 10 참조하면, 클라우드 컴퓨팅 환경(50) (도 9)에 의해 제공되는 일 세트의 기능별 추상화 계층들이 도시된다. 도 10에 도시된 컴포넌트들, 계층들, 및 기능들은 단지 예시의 목적이며 본 발명의 바람직한 실시 예들은 이것에 한정되지 않는다는 것을 미리 이해해야 한다. 도시된 바와 같이, 다음의 계층들과 그에 대응하는 기능들이 제공된다:

[0209] [00209] 하드웨어 및 소프트웨어 계층(60)은 하드웨어 및 소프트웨어 컴포넌트들을 포함한다. 하드웨어 컴포넌트들의 예들에는: 메인프레임들(61); RISC(Reduced Instruction Set Computer) 아키텍처 기반 서버들(62); 서버들(63); 블레이드 서버들(64); 스토리지 디바이스들(65); 그리고 네트워크 및 네트워킹 컴포넌트들(66)이 포함된다. 일부 실시 예들에서, 소프트웨어 컴포넌트들은 네트워크 애플리케이션 서버 소프트웨어(67) 및 데이터베이스 소프트웨어(68)를 포함한다.

[0210] [00210] 가상화 계층(70)은 추상화 계층을 제공하며 이로부터 다음의 가상 주체들의 예들이 제공될 수 있다: 가상 서버들(71); 가상 스토리지(72); 가상 사설 네트워크를 포함하는, 가상 네트워크들(73); 가상 애플리케이션들 및 운영 체제들(74); 및 가상 클라이언트들(75).

[0211] [00211] 한 예에서, 관리 계층(80)은 아래에 기술하는 기능들을 제공한다. 자원 제공(Resource provisioning)(81)은 클라우드 컴퓨팅 환경 내에서 작업들을 수행하는 데 이용되는 컴퓨팅 자원들 및 기타 자원들의 동적 조달을 제공한다. 계측 및 가격 책정(Metering and Pricing)(82)은 자원들이 클라우드 컴퓨팅 환경 내에서 이용될 때 비용 추적, 및 이 자원들의 소비에 대한 요금 청구 또는 송장을 제공한다. 한 예에서, 이 자원들은 애플리케이션 소프트웨어 라이센스를 포함할 수 있다. 보안(Security)은 데이터 및 기타 자원들에 대한 보호뿐 아니라 클라우드 소비자들과 작업들에 대한 신원 확인을 제공한다. 사용자 포털(User portal)(83)은 소비자들 및 시스템 관리자들에게 클라우드 컴퓨팅 환경에 대한 액세스를 제공한다. 서비스 레벨 관리(Service level management)(84)는 요구되는 서비스 레벨이 충족되도록 클라우드 컴퓨팅 자원 할당 및 관리를 제공한다. 서비스 레벨 협약서(SLA) 기획 및 충족(planning and fulfillment)(85)은 SLA에 부합하는 예상되는 미래 요건에 맞는 클라우드 컴퓨팅 자원들의 사전-배치(pre-arrangement) 및 조달(procurement)을 제공한다.

[0212] [00212] 워크로드 계층(90)은 클라우드 컴퓨팅 환경이 이용될 수 있는 기능들의 예들을 제공한다. 이 계층에서 제공될 수 있는 워크로드들과 기능들의 예들은 다음과 같다: 맵핑 및 네비게이션(91); 소프트웨어 개발 및 라이프사이클 관리(92); 가상 교실 교육 전달(93); 데이터 분석 처리(94); 트랜잭션 처리(95); 및 보안 가상 머신 덤프 처리(secure virtual machine dump processing)(96).

[0213] [00213] 본 발명의 실시 예들은 시스템, 방법, 및/또는 통합의 모든 가능한 기술적 세부 레벨에서 컴퓨터 프로그램 제품이 될 수 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 스토리지 매체(또는 미디어)를 포함할 수

있으며, 이 매체 상에 프로세서가 본 발명의 실시 예들을 수행하도록 하는 컴퓨터 판독 가능 프로그램 명령들을 갖는다.

[0214] [00214] 상기 컴퓨터 판독 가능 스토리지 매체는 명령 실행 장치에 의해 사용될 명령들을 유지 및 저장할 수 있는 유형의(tangible) 디바이스일 수 있다. 상기 컴퓨터 판독 가능 스토리지 매체는, 예를 들면, 전자 스토리지 디바이스, 자기 스토리지 디바이스, 광 스토리지 디바이스, 전자기 스토리지 디바이스, 반도체 스토리지 디바이스, 또는 전술한 것들의 모든 적절한 조합일 수 있으며, 그러나 이에 한정되지는 않는다. 컴퓨터 판독 가능 스토리지 매체의 더 구체적인 예들의 비포괄적인 목록에는 다음이 포함될 수 있다: 휴대용 컴퓨터 디스켓, 하드 디스크, 랜덤 액세스 메모리(RAM), 판독-전용 메모리(ROM), 소거 및 프로그램가능 판독-전용 메모리(EPROM 또는 플래시 메모리), 정적 랜덤 액세스 메모리(SRAM), 휴대용 컴팩트 디스크 판독-전용 메모리(CD-ROM), 디지털 다용도 디스크(DVD), 메모리 스틱, 플로피 디스크, 천공-카드들 또는 명령들이 기록된 홈에 있는 융기된 구조들 같이 머신적으로 인코드 된 장치, 및 전술한 것들의 모든 적절한 조합. 본 명세서에서 사용될 때, 컴퓨터 판독 가능 스토리지 매체는 무선 전파들이나 다른 자유롭게 전파되는 전자기파들, 도파관이나 기타 전송 매체(예를 들어, 광섬유 케이블을 통해 전달되는 광 펄스들)를 통해 전파되는 전자기파들, 또는 선(wire)을 통해 전송되는 전기 신호들 같이 그 자체로 일시적인(transitory) 신호들로 해석되지는 않는다.

[0215] [00215] 본 명세서에 기술되는 컴퓨터 판독 가능 명령들은, 예를 들어, 인터넷, 근거리 통신망, 광역 통신망 및/또는 무선 네트워크 등의 통신망(네트워크)을 통해 컴퓨터 판독 가능 스토리지 매체로부터 각각 컴퓨팅/처리 디바이스들로 또는 외부 스토리지 디바이스로부터 외부 컴퓨터로 다운로드 될 수 있다. 상기 통신망은 구리 전송 케이블들, 광 전송 섬유들, 무선 전송, 라우터들, 방화벽들, 스위치들, 게이트웨이 컴퓨터들 및/또는 엣지 서버들을 포함할 수 있다. 각 컴퓨팅/처리 유닛 내 네트워크 어댑터 카드 또는 네트워크 인터페이스는 상기 통신망으로부터 컴퓨터 판독 가능 프로그램 명령들을 수신하고 그 컴퓨터 판독 가능 프로그램 명령들을 각각의 컴퓨팅/처리 디바이스 내의 컴퓨터 판독 가능 스토리지 매체에 저장하기 위해 전송한다.

[0216] [00216] 본 발명의 연산들을 실행하기 위한 컴퓨터 판독 가능 프로그램 명령들은 Smalltalk, C++ 또는 그와 유사 언어 등의 객체 지향 프로그래밍 언어와 "C" 프로그래밍 언어 또는 그와 유사한 프로그래밍 언어 등의 종래의 절차적 프로그래밍 언어들을 포함하여, 하나 또는 그 이상의 프로그래밍 언어들을 조합하여 작성된(written) 어셈블러 명령들, 명령-세트-아키텍처(ISA) 명령들, 머신 명령들, 머신 종속 명령들, 마이크로코드, 펌웨어 명령들, 상태-셋팅 데이터, 집적회로를 위한 구성 데이터, 또는 소스 코드나 목적 코드일 수 있다. 상기 컴퓨터 판독 가능 프로그램 명령들은 전적으로 사용자의 컴퓨터상에서, 부분적으로 사용자의 컴퓨터상에서, 독립형(stand-alone) 소프트웨어 패키지로서, 부분적으로 사용자의 컴퓨터상에서 그리고 부분적으로 원격 컴퓨터상에서 또는 전적으로 원격 컴퓨터나 서버상에서 실행될 수 있다. 위에서 마지막의 경우에, 원격 컴퓨터는 근거리 통신망(LAN) 또는 광역 통신망(WAN)을 포함한 모든 종류의 네트워크를 통해서 사용자의 컴퓨터에 접속될 수 있고, 또는 이 접속은 (예를 들어, 인터넷 서비스 제공자를 이용한 인터넷을 통해서) 외부 컴퓨터에 이루어질 수도 있다. 일부 실시 예들에서, 예를 들어 프로그램 가능 로직 회로, 필드-프로그램 가능 게이트 어레이들(FPGA), 또는 프로그램 가능 로직 어레이들(PLA)을 포함한 전자 회로는 본 발명의 실시 예들을 수행하기 위해 전자 회로를 맞춤화하도록 상기 컴퓨터 판독 가능 프로그램 명령들의 상태 정보를 활용하여 상기 컴퓨터 판독 가능 프로그램 명령들을 실행할 수 있다.

[0217] [00217] 본 발명의 특징들이 본 발명의 실시 예들에 따른 방법들, 장치들(시스템들), 및 컴퓨터 프로그램 제품들의 플로 차트 예시도들 및/또는 블록도들을 참조하여 기술된다. 플로 차트 예시도들 및/또는 블록도들의 각 블록과 플로 차트 예시도들 및/또는 블록도들 내 블록들의 조합들은 컴퓨터 판독 가능 프로그램 명령들에 의해 구현될 수 있다는 것을 이해할 수 있을 것이다.

[0218] [00218] 이들 컴퓨터 판독 가능 프로그램 명령들은 범용 컴퓨터, 특수목적용 컴퓨터, 또는 기타 프로그램가능 데이터 처리 유닛의 프로세서에 제공되어 머신(machine)을 생성하고, 그렇게 하여 그 명령들이 상기 컴퓨터 또는 기타 프로그램가능 데이터 처리 유닛의 프로세서를 통해서 실행되어, 상기 플로 차트 및/또는 블록도의 블록 또는 블록들에 명시된 기능들/연산들을 구현하기 위한 수단을 생성할 수 있다. 이들 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터 판독 가능 스토리지 매체에 저장될 수 있으며, 컴퓨터, 프로그램가능 데이터 처리 유닛 및/또는 기타 디바이스들에 지시하여 명령들이 저장된 상기 컴퓨터 판독 가능 스토리지 매체가 상기 플로 차트 및/또는 블록도의 블록 또는 블록들에 명시된 기능/연산의 특징들을 구현하는 명령들을 포함하는 제조품(an article of manufacture)을 포함하도록 특정한 방식으로 기능하게 할 수 있다.

[0219] [00219] 상기 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터, 기타 프로그램가능 데이터 처리 유닛, 또는 다

른 디바이스에 로드 되어, 상기 컴퓨터, 기타 프로그램가능 장치 또는 다른 디바이스에서 일련의 동작 단계들이 수행되게 하여 컴퓨터 구현 프로세스를 생성하며, 그렇게 하여 상기 컴퓨터, 기타 프로그램가능 장치, 또는 다른 디바이스 상에서 실행되는 명령들이 플로 차트 및/또는 블록도의 블록 또는 블록들에 명시된 기능들/연산들을 구현할 수 있다.

[0220] [00220] 도면들 내 플로 차트 및 블록도들은 본 발명의 여러 실시 예들에 따른 시스템들, 방법들 및 컴퓨터 프로그램 제품들의 가능한 구현들의 아키텍처, 기능(functionality), 및 연산(operation)을 예시한다. 이와 관련하여, 상기 플로 차트 또는 블록도들 내 각 블록은 상기 명시된 논리적 기능(들)을 구현하기 위한 하나 또는 그 이상의 실행 가능한 명령들을 포함한 모듈, 세그먼트 또는 명령들의 일부분을 나타낼 수 있다. 일부 다른 실시 예들에서, 상기 블록에 언급되는 기능들은 도면들에 언급된 순서와 다르게 일어날 수도 있다. 예를 들면, 연속으로 도시된 두 개의 블록들은 실제로는 사실상 동시에 실행될 수도 있고, 또는 이 두 블록들은 때때로 관련된 기능에 따라서는 역순으로 실행될 수도 있다. 블록도들 및/또는 플로 차트 예시도의 각 블록, 및 블록도들 및/또는 플로 차트 예시도 내 블록들의 조합들은 특수목적용 하드웨어 및 컴퓨터 명령들의 명시된 기능들 또는 연산들, 또는 이들의 조합들을 수행하는 특수 목적용 하드웨어-기반 시스템들에 의해 구현될 수 있다는 것에 또한 주목해야 한다.

[0221] [00221] 전술한 것에 추가하여, 본 발명의 하나 또는 그 이상의 실시 예들은 고객 환경들의 관리를 공급하는 서비스 제공자에 의해 제공, 공급, 배치, 관리, 서비스 등이 될 수 있다. 예를 들면, 서비스 제공자는 하나 또는 그 이상의 고객들을 위해 본 발명의 하나 또는 그 이상의 실시 예들을 수행하는 컴퓨터 코드 및/또는 컴퓨터 인프라스트럭처의 제작, 유지, 지원 등을 할 수 있다. 그 대가로, 서비스 제공자는, 예를 들어, 가입제(subscription) 및/또는 수수료 약정에 따라 고객으로부터 대금을 수령할 수 있다. 추가적으로 또는 선택적으로, 서비스 제공자는 하나 또는 그 이상의 제3자들에게 광고 컨텐츠를 판매하고 대금을 수령할 수 있다.

[0222] [00222] 한 예에서, 본 발명의 하나 또는 그 이상의 실시 예들을 수행하기 위해 하나의 애플리케이션이 배치될 수 있다. 한 예로서, 하나의 애플리케이션의 배치는 본 발명의 하나 또는 그 이상의 실시 예들을 수행하기 위해 동작 가능한 컴퓨터 인프라스트럭처를 제공하는 것을 포함할 수 있다.

[0223] [00223] 추가의 실시 예로서, 컴퓨터 판독 가능 코드를 컴퓨팅 시스템으로 통합하는 것을 포함하는 컴퓨팅 인프라스트럭처가 배치될 수 있으며, 상기 컴퓨팅 인프라스트럭처에서 상기 코드는 상기 컴퓨팅 시스템과 결합하여 본 발명의 하나 또는 그 이상의 실시 예들을 수행할 수 있다.

[0224] [00224] 추가의 실시 예로서, 컴퓨터 판독 가능 코드를 컴퓨터 시스템으로 통합시키는 것을 포함하는 컴퓨팅 인프라스트럭처를 통합하기 위한 프로세스가 제공될 수 있다. 상기 컴퓨터 시스템은 컴퓨터 판독 가능 매체를 포함하고, 상기 컴퓨터 시스템에서 컴퓨터 매체는 본 발명의 하나 또는 그 이상의 실시 예들을 포함한다. 상기 코드는 상기 컴퓨터 시스템과 결합하여 본 발명의 하나 또는 그 이상의 실시 예들을 수행할 수 있다.

[0225] [00225] 위에서 다양한 실시 예들이 기술되었지만, 이들은 단지 예시들일 뿐이다. 예를 들면, 다른 아키텍처들의 컴퓨팅 환경들이 본 발명의 하나 또는 그 이상의 실시 예들을 포함하고 사용하는 데 사용될 수 있다. 또한, 다른 명령들, 커맨드들, 함수들(functions), 호출들(calls) 및/또는 연산들이 사용될 수 있다. 많은 변형들이 가능하다.

[0226] [00226] 다양한 실시예들이 본 명세서에 기술되어 있다. 또한, 본 발명의 실시예들의 정신으로부터 벗어나지 않는 범위 내에서 다양한 변형들이 가능하다. 달리 불일치하지 않는 한, 여기에 기술된 각 실시예 또는 특징 및 그 변형들은 모든 다른 실시예 또는 특징과 결합 가능할 수 있음에 유의해야 한다.

[0227] [00227] 또한, 다른 유형의 컴퓨팅 환경들도 유익을 얻을 수 있고 사용될 수 있다. 예로서, 프로그램 코드를 저장 및/또는 실행하기에 적합한 데이터 처리 시스템이 사용될 수 있으며, 이 시스템은 시스템 버스를 통해서 메모리 엘리멘트들에 직접적으로 또는 간접적으로 결합된 적어도 두 개의 프로세서들을 포함한다. 상기 메모리 엘리멘트들은, 예를 들어 프로그램 코드의 실제 실행 동안 사용되는 로컬 메모리, 대용량 스토리지(bulk storage), 및 코드가 실행 동안에 대용량 스토리지로부터 검색되어야 하는 횟수를 감소시키기 위해 적어도 일부 프로그램 코드의 임시 스토리지(temporary storage)를 제공하는 캐시 메모리를 포함한다.

[0228] [00228] 입력/출력 또는 I/O 디바이스들(키보드들, 디스플레이들, 포인팅 디바이스들, DASD, 테이프, CD들, DVD들, 썸 드라이브들 및 기타 메모리 매체 등을 포함하나 이에 한정되지는 않음)은 직접 또는 중개(intervening) I/O 컨트롤기들을 통해서 시스템에 결합될 수 있다. 네트워크 어댑터 또한 상기 시스템에 결합되어 데이터 처리

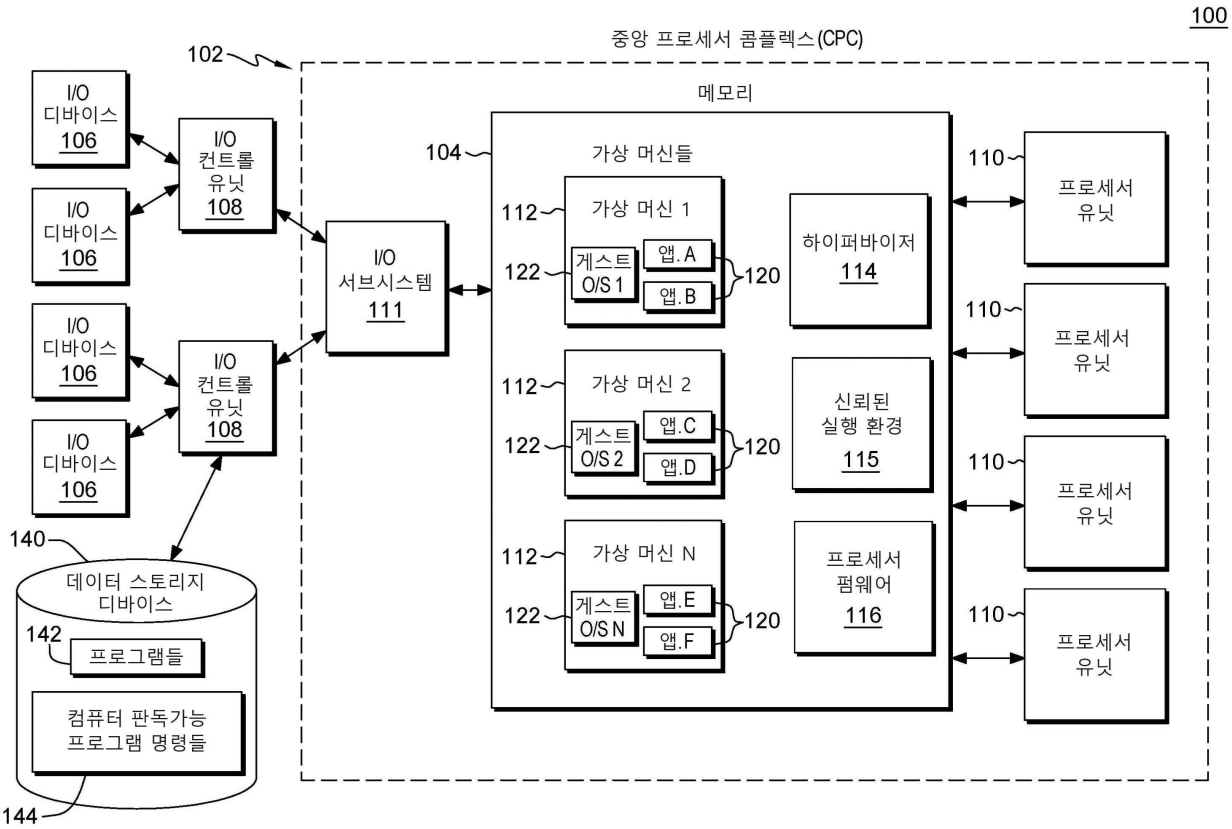
시스템이 중개하는 사설 또는 공공 네트워크를 통해서 기타 데이터 처리 시스템 또는 원격 포인터 또는 스토리지 디바이스에 결합되는 것을 가능하게 한다. 모뎀들, 케이블 모뎀들, 및 이더넷 카드들은 이용 가능한 유형의 네트워크 어댑터들의 단지 일부이다.

[0229] [00229] 본 명세서에서 사용된 용어들은 단지 본 발명의 특정 실시 예들을 기술할 목적으로 사용된 것이지 한정하려는 의도로 사용된 것은 아니다. 본 명세서에서 사용할 때, 단수 형태 “하나", "한", "그"는 그 컨텍스트에서 그렇지 않은 것으로 명확히 명시되어 있지 않으면, 복수 형태도 또한 포함할 의도로 기술된 것이다. 또한, "포함한다" 및/또는 "포함하는" 이라는 말들은 본 명세서에서 사용될 때, 언급되는 특징들, 정수들, 단계들, 연산들, 엘리멘트들, 및/또는 컴포넌트들의 존재를 명시하지만, 하나 또는 그 이상의 다른 특징들, 정수들, 단계들, 연산들, 엘리멘트들, 컴포넌트들 및/또는 이들의 그룹들의 존재 또는 추가를 배제하는 것은 아니라는 것을 이해할 수 있을 것이다.

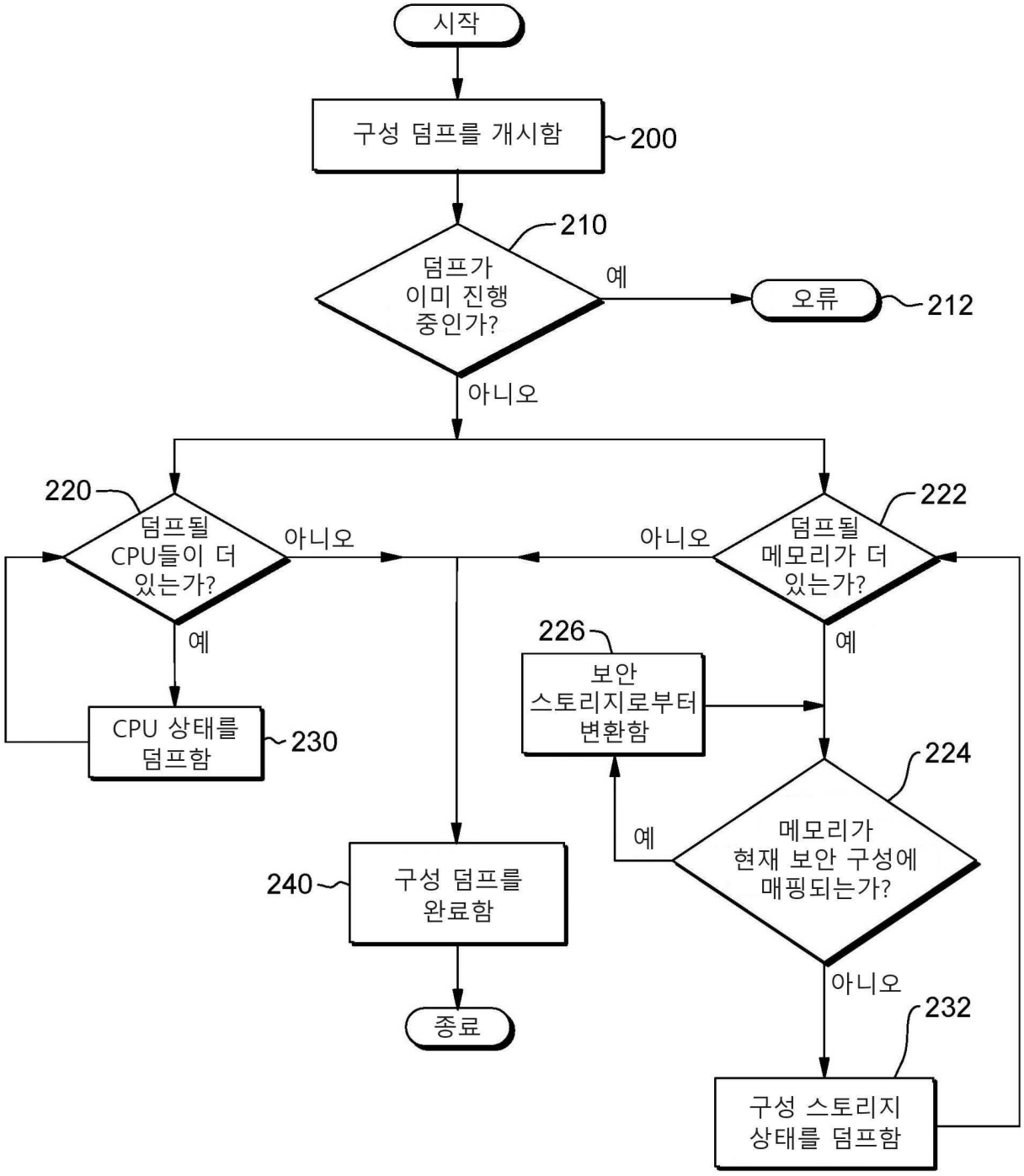
[0230] [00230] 이하의 청구항들에서, 대응하는 구조들(structures), 재료들(materials), 동작들(acts), 및 모든 수단의 등가물들 또는 단계 플러스 기능 엘리멘트들은, 만일 있다면, 구체적으로 청구되는 다른 청구된 엘리멘트들과 조합하여 그 기능을 수행하기 위한 구조, 재료, 또는 동작을 포함할 의도가 있다. 본 발명의 하나 또는 그 이상의 실시 예들에 대한 설명은 예시와 설명의 목적으로 제공되는 것이며, 개시되는 형태로 빠짐없이 만들어지거나 한정하려는 의도가 있는 것은 아니다. 이 기술 필드에서 통상의 지식을 가진 자라면 많은 수정들 및 변형들이 있을 수 있다는 것을 알 수 있다. 본 발명의 실시 예는 여러 특징들 및 실제 응용을 가장 잘 설명하기 위해 그리고 고려되는 특정 용도에 적합하게 여러 수정들을 갖는 다양한 실시 예들을 이 기술 필드에서 통상의 지식을 가진 자들이 이해할 수 있도록 하기 위해, 선택되고 기술되었다.

도면

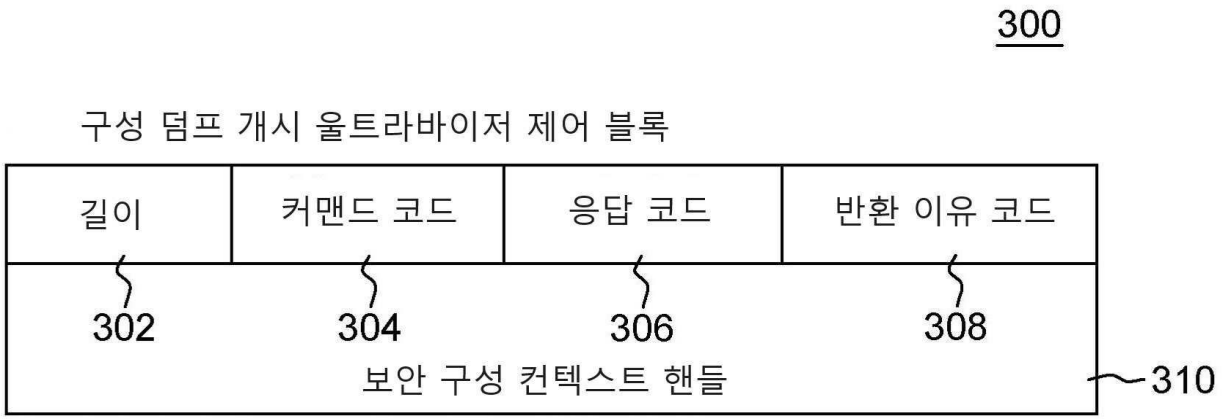
도면1



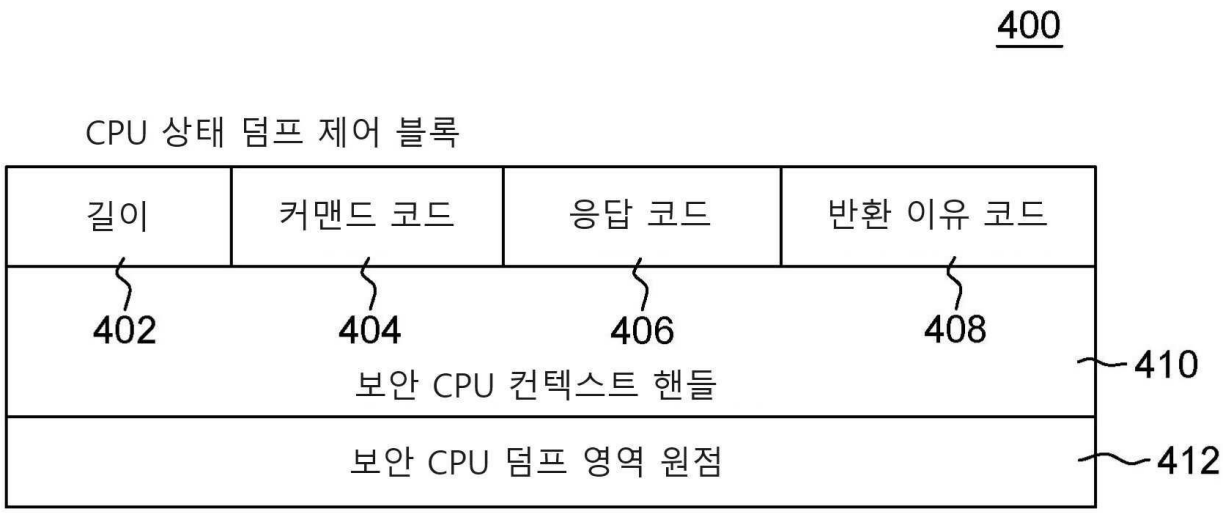
도면2


시작
구성 덤프를 개시함
200
210
덤프가 이미 진행 중인가?
예
오류
212
아니오
220
덤프될 CPU들이 더 있는가?
아니오
222
덤프될 메모리가 더 있는가?
아니오
예
예
226
보안 스토리지로부터 변환함
CPU 상태를 덤프함
230
224
메모리가 현재 보안 구성에 매핑되는가?
예
240
구성 덤프를 완료함
아니오
232
구성 스토리지 상태를 덤프함
종료

도면3


300
구성 덤프 개시 울트라바이저 제어 블록
길이
커맨드 코드
응답 코드
반환 이유 코드
302
304
306
308
보안 구성 컨텍스트 핸들
310

도면4a


400
CPU 상태 덤프 제어 블록
길이
커맨드 코드
응답 코드
반환 이유 코드
402
404
406
408
보안 CPU 컨텍스트 핸들
410
보안 CPU 덤프 영역 원점
412

도면4b

420

보안 CPU 덤프 영역

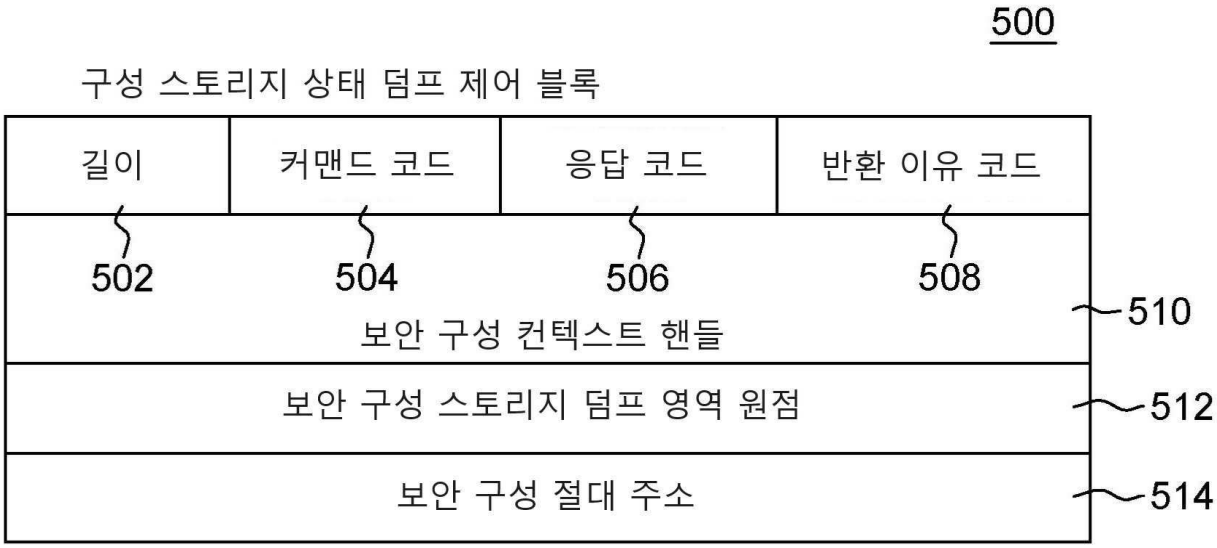
보안 CPU 덤프 영역 버전 ~ 422	보안 CPU 덤프 영역 길이 ~ 424
초기화 벡터 ~ 426	
일반 레지스터들 ~ 428	
현재 프로그램 상태 워드 ~ 430	
프리픽스 ~ 432	부동 소수점 제어 레지스터 ~ 434
TOD 프로그래밍 가능 레지스터 ~ 436	
CPU 타이머 ~ 438	
클록 비교기 ~ 440	
액세스 레지스터들 ~ 442	
제어 레지스터들 ~ 444	
벡터 레지스터들 ~ 446	
보호된 스토리지 지정 레지스터 ~ 448	
보호된 스토리지 섹션 마스크 레지스터 ~ 450	
보호된 스토리지 이벤트 파라미터 목록 주소 레지스터 ~ 452	
. . .	
덤프 플래그들 ~ 454	
보안 CPU 덤프 영역 태그 ~ 456	

도면4c

470

보안 스토리지로부터의 변환 제어 블록

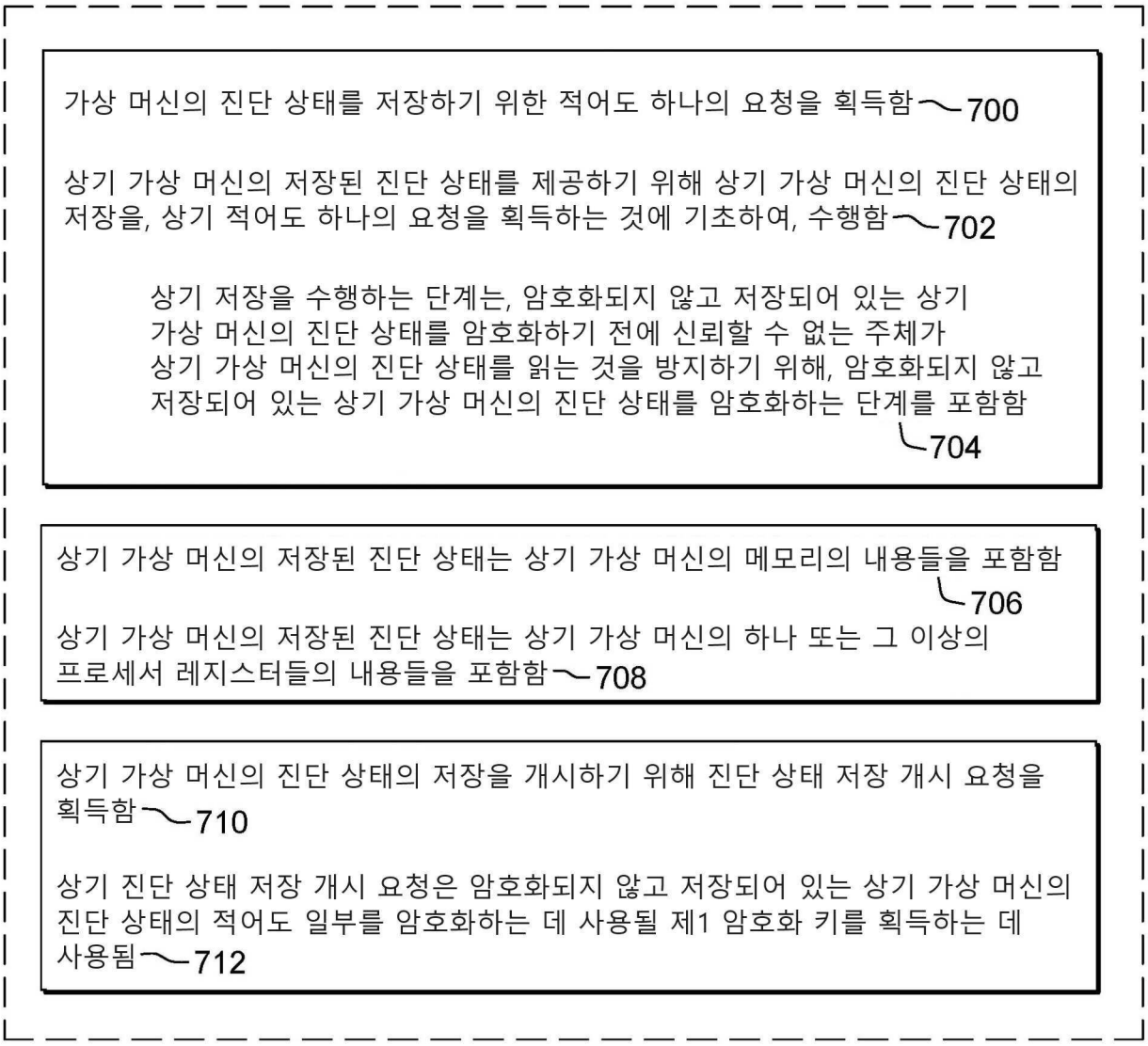
도면5


500
구성 스토리지 상태 덤프 제어 블록
길이
커맨드 코드
응답 코드
반환 이유 코드
502
504
506
508
보안 구성 컨텍스트 핸들
510
보안 구성 스토리지 덤프 영역 원점
512
보안 구성 절대 주소
514

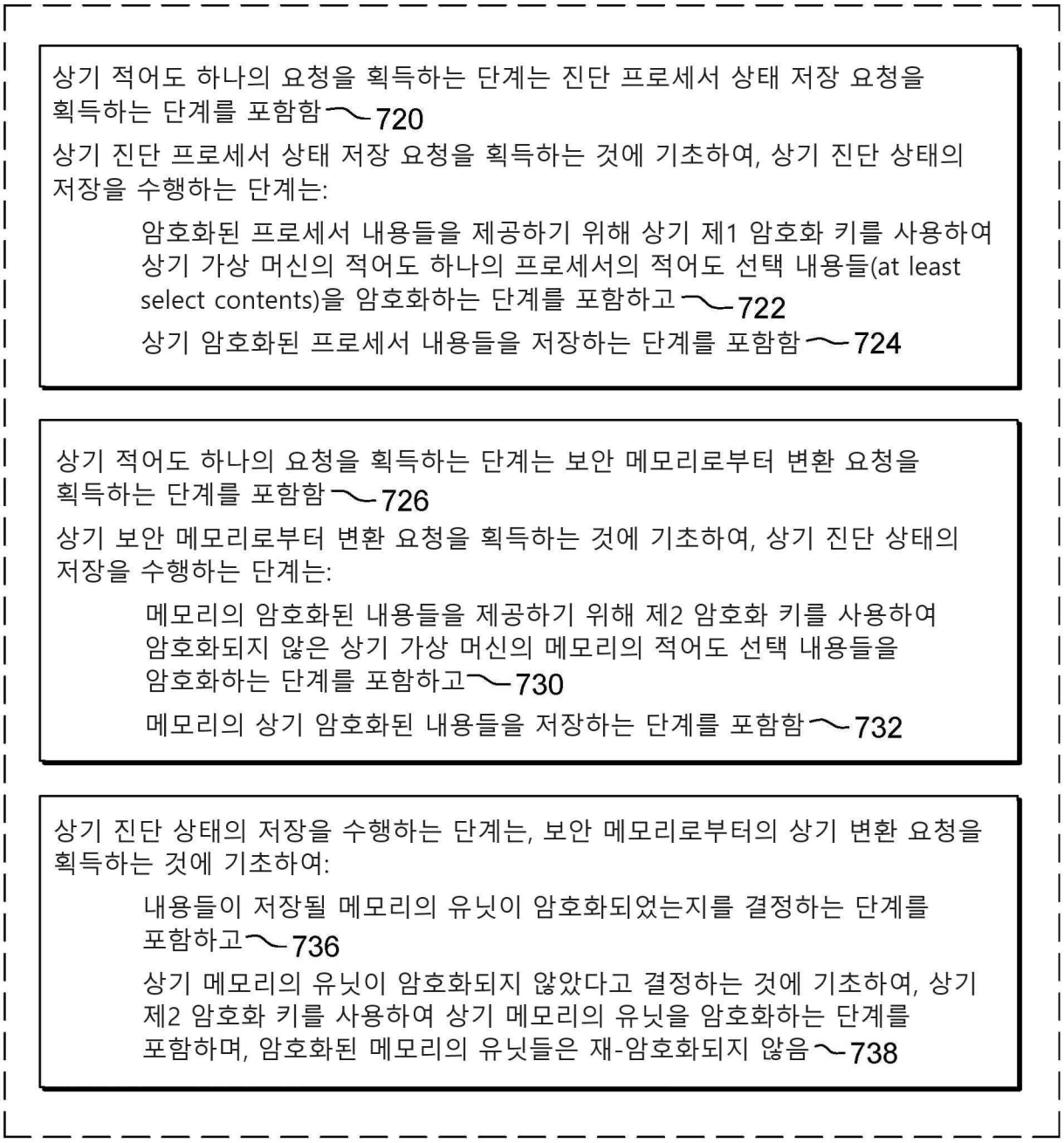
도면6


600
구성 덤프 완료 제어 블록
길이
커맨드 코드
응답 코드
반환 이유 코드
602
604
606
608
보안 구성 컨텍스트 핸들
609
키 유도 시드
610
초기화 벡터
612
튀크 논스
614
스토리지 암호화 키 1
616
스토리지 암호화 키 2
618
인증 태그
620

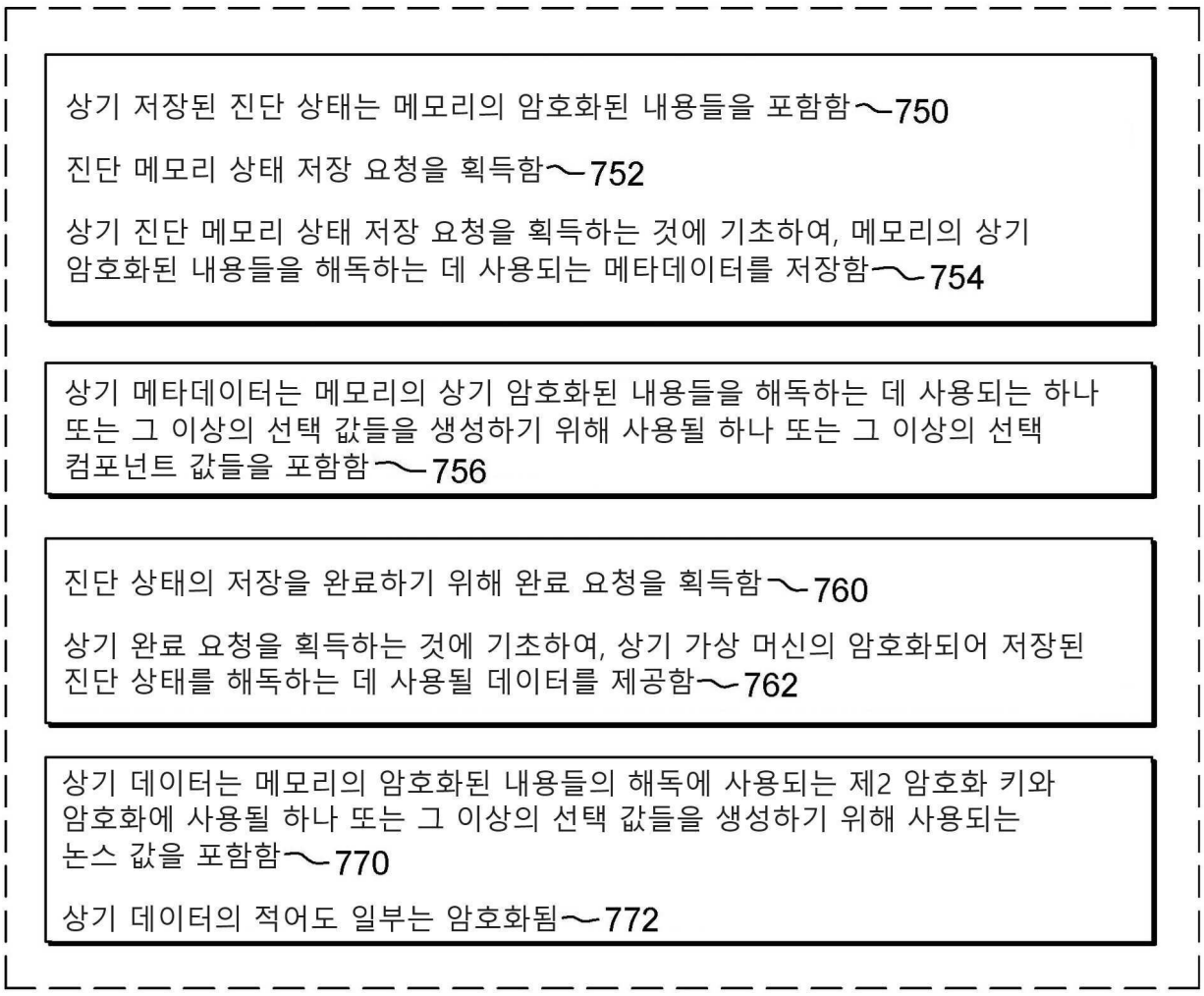
도면7a


가상 머신의 진단 상태를 저장하기 위한 적어도 하나의 요청을 획득함
700
상기 가상 머신의 저장된 진단 상태를 제공하기 위해 상기 가상 머신의 진단 상태의 저장을, 상기 적어도 하나의 요청을 획득하는 것에 기초하여, 수행함
702
상기 저장을 수행하는 단계는, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하기 전에 신뢰할 수 없는 주체가 상기 가상 머신의 진단 상태를 읽는 것을 방지하기 위해, 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태를 암호화하는 단계를 포함함
704
상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 메모리의 내용들을 포함함
706
상기 가상 머신의 저장된 진단 상태는 상기 가상 머신의 하나 또는 그 이상의 프로세서 레지스터들의 내용들을 포함함
708
상기 가상 머신의 진단 상태의 저장을 개시하기 위해 진단 상태 저장 개시 요청을 획득함
710
상기 진단 상태 저장 개시 요청은 암호화되지 않고 저장되어 있는 상기 가상 머신의 진단 상태의 적어도 일부를 암호화하는 데 사용될 제1 암호화 키를 획득하는 데 사용됨
712

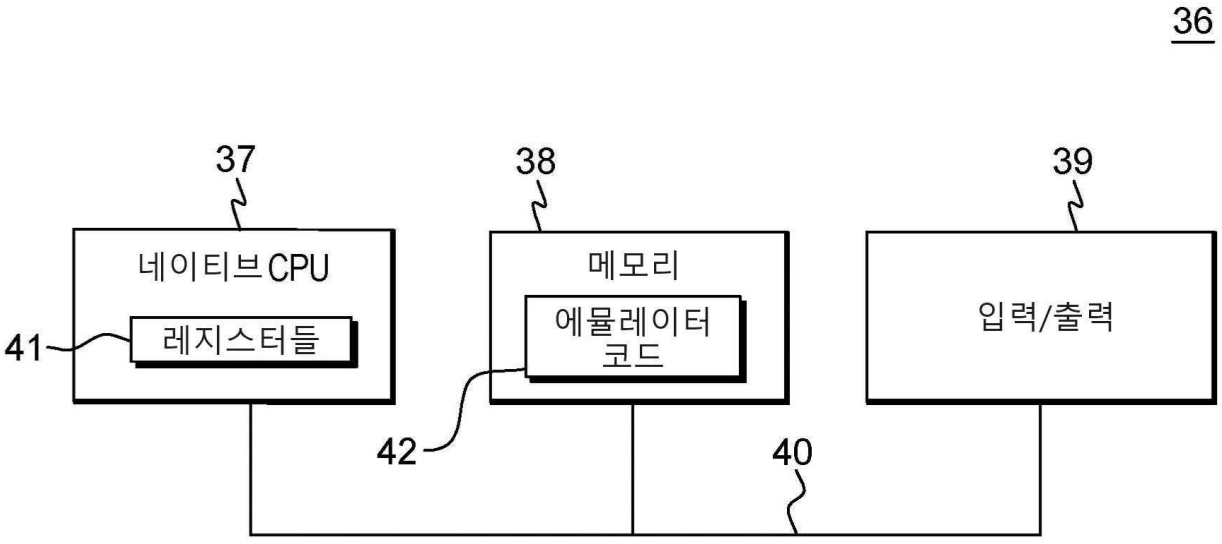
도면7b


상기 적어도 하나의 요청을 획득하는 단계는 진단 프로세서 상태 저장 요청을 획득하는 단계를 포함함 720
상기 진단 프로세서 상태 저장 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:
암호화된 프로세서 내용들을 제공하기 위해 상기 제1 암호화 키를 사용하여 상기 가상 머신의 적어도 하나의 프로세서의 적어도 선택 내용들(at least select contents)을 암호화하는 단계를 포함하고 722
상기 암호화된 프로세서 내용들을 저장하는 단계를 포함함 724
상기 적어도 하나의 요청을 획득하는 단계는 보안 메모리로부터 변환 요청을 획득하는 단계를 포함함 726
상기 보안 메모리로부터 변환 요청을 획득하는 것에 기초하여, 상기 진단 상태의 저장을 수행하는 단계는:
메모리의 암호화된 내용들을 제공하기 위해 제2 암호화 키를 사용하여 암호화되지 않은 상기 가상 머신의 메모리의 적어도 선택 내용들을 암호화하는 단계를 포함하고 730
메모리의 상기 암호화된 내용들을 저장하는 단계를 포함함 732
상기 진단 상태의 저장을 수행하는 단계는, 보안 메모리로부터의 상기 변환 요청을 획득하는 것에 기초하여:
내용들이 저장될 메모리의 유닛이 암호화되었는지를 결정하는 단계를 포함하고 736
상기 메모리의 유닛이 암호화되지 않았다고 결정하는 것에 기초하여, 상기 제2 암호화 키를 사용하여 상기 메모리의 유닛을 암호화하는 단계를 포함하며, 암호화된 메모리의 유닛들은 재-암호화되지 않음 738

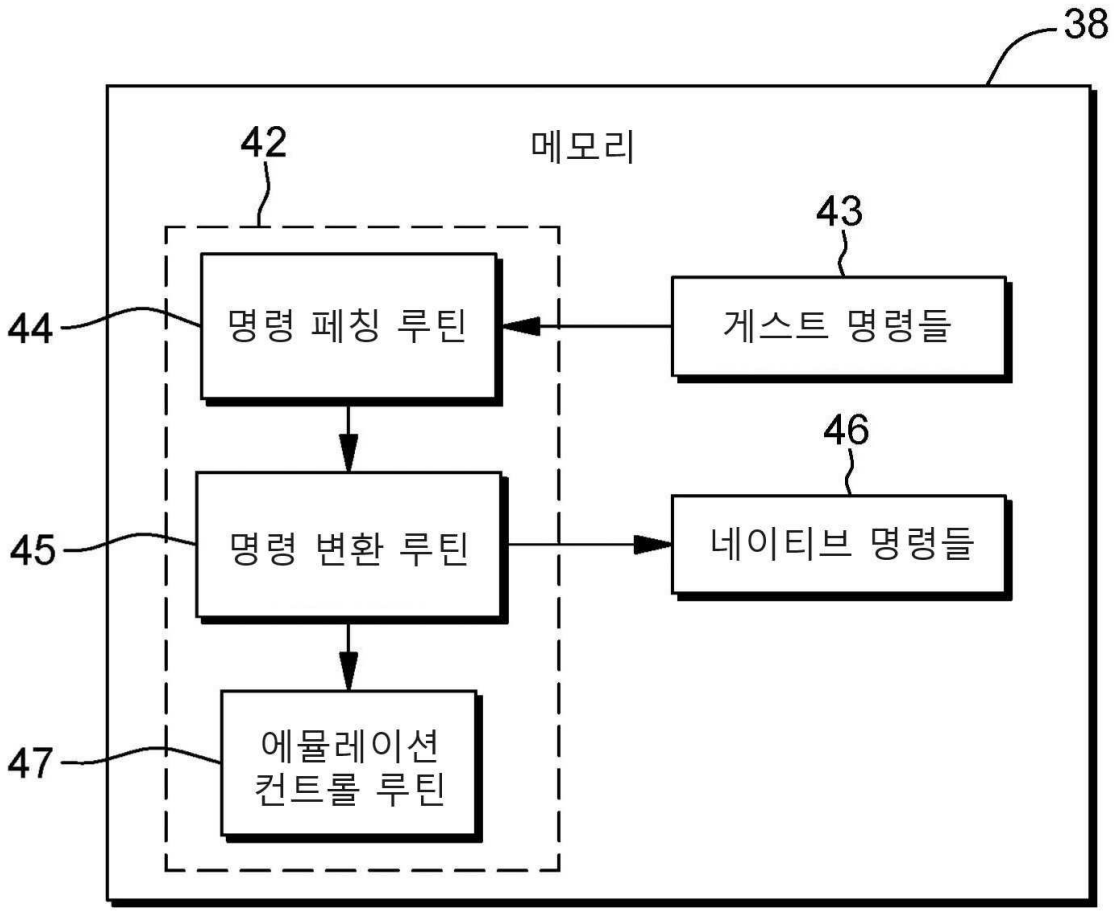
도면7c


상기 저장된 진단 상태는 메모리의 암호화된 내용들을 포함함 750
진단 메모리 상태 저장 요청을 획득함 752
상기 진단 메모리 상태 저장 요청을 획득하는 것에 기초하여, 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 메타데이터를 저장함 754
상기 메타데이터는 메모리의 상기 암호화된 내용들을 해독하는 데 사용되는 하나 또는 그 이상의 선택 값들을 생성하기 위해 사용될 하나 또는 그 이상의 선택 컴포넌트 값들을 포함함 756
진단 상태의 저장을 완료하기 위해 완료 요청을 획득함 760
상기 완료 요청을 획득하는 것에 기초하여, 상기 가상 머신의 암호화되어 저장된 진단 상태를 해독하는 데 사용될 데이터를 제공함 762
상기 데이터는 메모리의 암호화된 내용들의 해독에 사용되는 제2 암호화 키와 암호화에 사용될 하나 또는 그 이상의 선택 값들을 생성하기 위해 사용되는 논스 값을 포함함 770
상기 데이터의 적어도 일부는 암호화됨 772

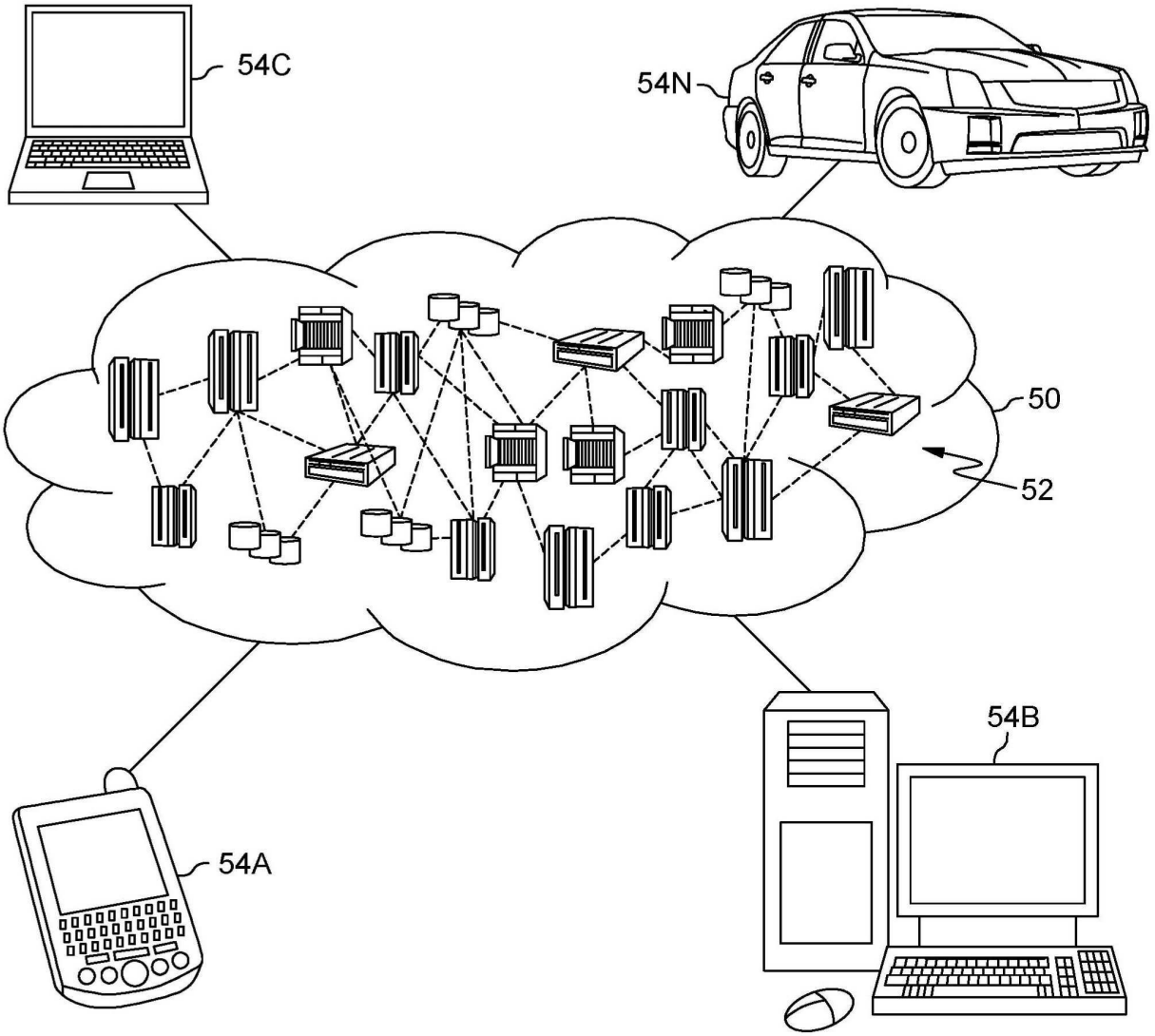
도면8a


36
37
네이티브 CPU
41
레지스터들
38
메모리
에뮬레이터 코드
42
39
입력/출력
40

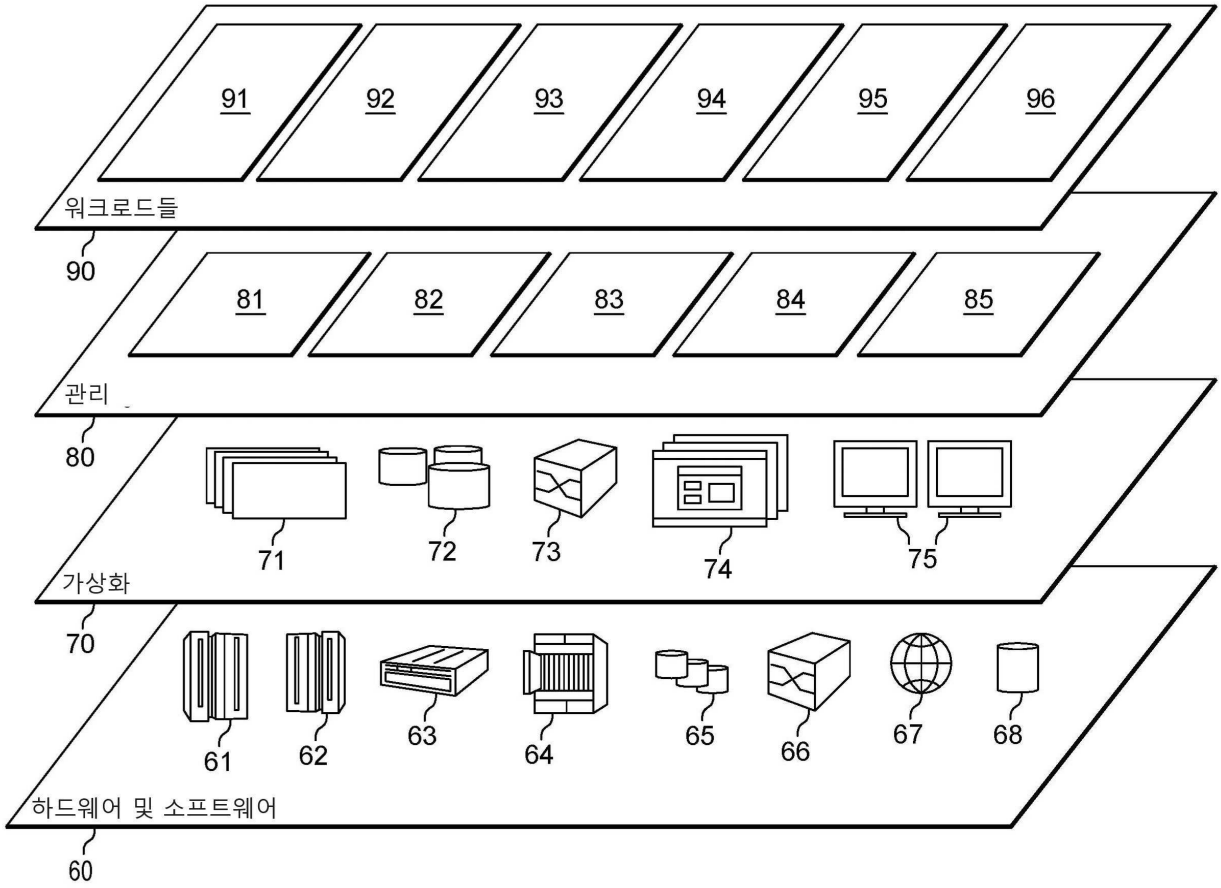
도면8b


38
메모리
42
43
44
명령 페칭 루틴
게스트 명령들
46
45
명령 변환 루틴
네이티브 명령들
47
에뮬레이션
컨트롤 루틴

도면9



도면10


91
92
93
94
95
96
워크로드들
90
81
82
83
84
85
관리
80
71
72
73
74
75
가상화
70
61
62
63
64
65
66
67
68
하드웨어 및 소프트웨어
60