



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2017년12월06일

(11) 등록번호 10-1804996

(24) 등록일자 2017년11월29일

- (51) 국제특허분류(Int. Cl.)  
G06F 21/51 (2013.01) G06F 17/30 (2006.01)  
G06F 21/52 (2013.01) H04L 29/06 (2006.01)
- (52) CPC특허분류  
G06F 21/51 (2013.01)  
G06F 17/30943 (2013.01)
- (21) 출원번호 10-2016-7031725(분할)
- (22) 출원일자(국제) 2012년12월28일  
심사청구일자 2016년11월14일
- (85) 번역문제출일자 2016년11월14일
- (65) 공개번호 10-2016-0134867
- (43) 공개일자 2016년11월23일
- (62) 원출원 특허 10-2014-7023375  
원출원일자(국제) 2012년12월28일  
심사청구일자 2014년08월21일
- (86) 국제출원번호 PCT/US2012/072191
- (87) 국제공개번호 WO 2013/115927  
국제공개일자 2013년08월08일
- (30) 우선권주장  
61/595,021 2012년02월03일 미국(US)  
(뒷면에 계속)
- (56) 선행기술조사문헌  
KR1020060079854 A\*  
KR1020090001398 A\*  
KR1020010109271 A  
JP2002544637 A
- \*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
**애플 인크.**  
미합중국 95014 캘리포니아 쿠파티노 인피니트 루프 1
- (72) 발명자  
**키에트라이버, 피터**  
미국 95014 캘리포니아주 쿠파티노 엠/에스 3-씨  
오에스 인피니트 루프 1  
**비드린, 자케스, 에이.**  
미국 95014 캘리포니아주 쿠파티노 엠/에스 3-씨  
오에스 인피니트 루프 1  
(뒷면에 계속)
- (74) 대리인  
**양영준, 백만기**

전체 청구항 수 : 총 25 항

심사관 : 구대성

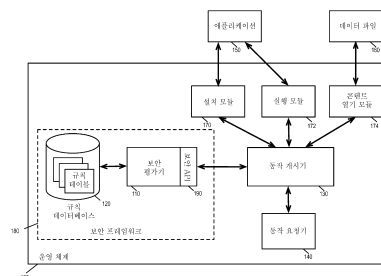
(54) 발명의 명칭 **집중형 동작 관리**

(57) 요약

디바이스의 운영 체제의 일부인 신규한 보안 프레임워크가 제공된다. 프레임워크는 디바이스 상에서 실행되는 애플리케이션에 대하여 수행되도록 요구하는 상이한 동작들에 대한 보안 정책 평가들을 수행하는 보안 평가를 포함한다. 그러한 동작들의 예들은 애플리케이션의 설치, 애플리케이션의 실행, 및 애플리케이션에 의한 콘텐츠

(뒷면에 계속)

## 대표도 - 도1



파일들의 열기(예컨대, 문서들의 열기)를 포함한다.

(52) CPC특허분류

**G06F 21/52** (2013.01)

**H04L 63/20** (2013.01)

(72) 발명자

**린, 크리스토퍼, 에스.**

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 41-1엔

에스 인피니트 루프 1

**살던저, 랜디, 디.**

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 303-3

케이티 인피니트 루프 1

**토마스, 브레이든, 제이.**

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 302-3

오에스3 인피니트 루프 1

(30) 우선권주장

13/624,836 2012년09월21일 미국(US)

13/624,832 2012년09월21일 미국(US)

13/624,828 2012년09월21일 미국(US)

## 명세서

### 청구범위

#### 청구항 1

디바이스 상에서 실행되는 운영 체제(operating system)에 의해 수행되는 동작들의 보안을 평가하는 방법으로서,

동작을 수행하라는 요청을 수신하는 단계;

상기 동작과 연관되는 데이터 오브젝트를 검색하는 단계;

상기 데이터 오브젝트가 상기 디바이스 외부의 소스로부터의 것임을 나타내기 위한 태그 식별자와 상기 데이터 오브젝트가 연관되는 경우, (1) 상기 운영 체제의 보안 정책들에 기초하여 상기 데이터 오브젝트에 대한 보안 평가를 수행하고, (2) 상기 데이터 오브젝트가 상기 보안 평가를 통과하는 경우 요청된 상기 동작을 수행하는 단계; 및

상기 데이터 오브젝트가 상기 태그 식별자와 연관되지 않은 경우, 상기 데이터 오브젝트에 대한 상기 보안 평가를 수행하지 않고 상기 동작을 수행하는 단계

를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

요청되는 상기 동작은, 상기 운영 체제에 의해 제공되는 집중형 보안 평가기(centralized security assessor)에 의해 평가되는 복수의 동작들 중 하나인, 방법.

#### 청구항 3

제1항에 있어서,

상기 보안 평가를 수행하는 것은, 수신된 상기 요청 이전에 상기 데이터 오브젝트에 대해 보안 평가가 수행되지 않은 경우, 상기 데이터 오브젝트에 대한 상기 보안 평가를 수행하는 것을 더 포함하는, 방법.

#### 청구항 4

제1항에 있어서,

상기 보안 평가의 결과에 따라 상기 태그 식별자를 업데이트하는 단계를 더 포함하는 방법.

#### 청구항 5

제1항에 있어서,

상기 데이터 오브젝트는 콘텐츠 파일, 애플리케이션 프로그램, 및 설치 프로그램 중 하나인, 방법.

#### 청구항 6

제1항에 있어서,

요청된 상기 동작을 수행하는 것은, (1) 상기 데이터 오브젝트가 보안 정책을 준수함을 상기 보안 평가가 나타내는 경우, 요청된 상기 동작을 허용하는 것, 및 (2) 상기 데이터 오브젝트가 보안 정책을 준수하지 못함을 상기 보안 평가가 나타내는 경우, 요청된 상기 동작을 불허하는 것을 포함하는, 방법.

#### 청구항 7

제1항에 있어서,

상기 데이터 오브젝트에 대해 사전 보안 평가(previous security assessment)가 이미 수행되었음을 상기 태그

식별자가 나타내는 경우, 상기 보안 평가를 건너뛰는(bypassing) 단계를 더 포함하는 방법.

#### 청구항 8

제7항에 있어서,

(1) 상기 데이터 오브젝트가 보안 정책을 준수함을 상기 태그 식별자가 나타내는 경우, 요청된 상기 동작을 허용하는 단계 및 (2) 상기 데이터 오브젝트가 상기 보안 정책을 준수하지 못함을 상기 태그 식별자가 나타내는 경우, 요청된 상기 동작을 불허하는 단계를 더 포함하는 방법.

#### 청구항 9

프로그램을 저장하는 비밀시적 기계 판독가능 매체로서, 상기 프로그램은 디바이스의 적어도 하나의 프로세싱 유닛에 의해 실행되는 경우, 상기 디바이스 상에서 실행되는 운영 체제에 의해 수행되는 동작들의 보안을 평가하며, 상기 프로그램은,

동작을 수행하라는 요청을 수신하고,

상기 동작과 연관되는 데이터 오브젝트가 검역 비트(quarantine bit)를 포함하는지를 결정하고,

상기 데이터 오브젝트가 상기 디바이스 외부의 소스로부터의 것임을 나타내기 위해 상기 검역 비트와 상기 데이터 오브젝트가 연관되는 경우, (1) 상기 운영 체제의 보안 정책들에 기초하여 상기 데이터 오브젝트에 대한 보안 평가를 수행하고, (2) 상기 데이터 오브젝트를 상기 검역 비트와 연관 해제시키고, (3) 상기 데이터 오브젝트가 상기 보안 평가를 통과하는 경우 요청된 상기 동작을 수행하고,

상기 데이터 오브젝트가 상기 검역 비트와 연관되지 않은 경우, 상기 데이터 오브젝트에 대한 상기 보안 평가를 수행하지 않고 상기 동작을 수행하기 위한

명령어들의 세트를 포함하는, 기계 판독가능 매체.

#### 청구항 10

제9항에 있어서,

상기 보안 평가는, 상기 데이터 오브젝트에 관련된 사용자 입력에 기초하는, 기계 판독가능 매체.

#### 청구항 11

제9항에 있어서,

상기 프로그램은,

상기 외부의 소스로부터 상기 데이터 오브젝트를 수신하고,

상기 검역 비트와 상기 데이터 오브젝트를 연관시키고,

상기 데이터 오브젝트를 저장하기 위한

명령어들의 세트를 더 포함하는, 기계 판독가능 매체.

#### 청구항 12

제11항에 있어서,

상기 검역 비트와 수신된 상기 데이터 오브젝트를 연관시키기 위한 명령어들의 세트는, 상기 검역 비트를 상기 데이터 오브젝트에 첨부하기 위한 명령어들의 세트를 포함하는, 기계 판독가능 매체.

#### 청구항 13

제11항에 있어서,

상기 데이터 오브젝트를 저장하기 위한 명령어들의 세트는, 태그된 데이터 오브젝트들의 리스트를 유지하는 태그 테이블에 상기 데이터 오브젝트를 기록하기 위한 명령어들의 세트를 포함하는, 기계 판독가능 매체.

#### 청구항 14

제9항에 있어서,

상기 보안 평가는 외부의 데이터 오브젝트들을 처리하기 위한 규칙들을 구체화하는 보안 정책들에 기초하는, 기계 판독가능 매체.

#### 청구항 15

제14항에 있어서,

상기 보안 평가는 상기 데이터 오브젝트와 연관되는 상기 검역 비트의 사용자 오버라이드(override)를 수신하는 것을 포함하는, 기계 판독가능 매체.

#### 청구항 16

보안 평가 시스템으로서,

디지털 오브젝트를, 상기 시스템의 외부의 소스로부터 상기 디지털 오브젝트가 다운로드 되는지를 나타내는 태그와 연관시키기 위한 태그 모듈;

(1) 상기 디지털 오브젝트와 연관되는 특정 동작을 개시하기 위한 요청을 수신하고 (2) 연관된 상기 태그에 기초하여 상기 디지털 오브젝트의 보안 평가를 요청하는지를 판정하기 위한 동작 개시기; 및

상기 디지털 오브젝트가 상기 시스템의 외부의 소스로부터의 것임을 나타내기 위해 상기 태그와 연관되는 경우 상기 시스템의 보안 정책들에 기초하여 상기 디지털 오브젝트의 보안 평가를 실시하기 위한 보안 평가기

를 포함하고,

상기 디지털 오브젝트가 상기 태그와 연관되지 않은 경우, 상기 동작 개시기는 상기 보안 평가기에게 상기 디지털 오브젝트의 보안 평가를 실시할 것을 요청하지 않는, 시스템.

#### 청구항 17

제16항에 있어서,

디지털 오브젝트들과 연관되는 태그들을 조사하기 위해 상기 보안 평가기에 규칙들을 제공하는 규칙 데이터베이스를 더 포함하는 시스템.

#### 청구항 18

제16항에 있어서,

상기 보안 평가는 외부의 디지털 오브젝트들을 처리하기 위한 규칙들을 구체화하는 보안 정책들에 기초하는, 시스템.

#### 청구항 19

제16항에 있어서,

상기 디지털 오브젝트가 상기 시스템의 외부의 소스로부터의 것이 아님을 상기 디지털 오브젝트의 상기 태그가 나타내는 경우, 상기 보안 평가기가 상기 특정 동작을 허용하는, 시스템.

#### 청구항 20

제16항에 있어서,

상기 보안 평가기는 상기 시스템에서 일어나는 모든 동작들에 대해 집중화된 보안 평가를 수행하는, 시스템.

#### 청구항 21

제16항에 있어서,

상기 디지털 오브젝트를 연관된 태그와 함께 저장하기 위한 저장소를 더 포함하는 시스템.

## 청구항 22

보안 평가 방법으로서,

보안 동작들을 평가하는 운영 체제를 제공하는 단계;

상기 운영 체제 내에, 상기 운영 체제에서 수행될 동작들의 보안 평가들을 실시하기 위한 보안 평가기를 제공하는 단계;

상기 보안 평가들의 결과들을 결정하기 위한 규칙 데이터베이스를 제공하는 단계

를 포함하고, 상기 규칙 데이터베이스는,

제1 복수의 엔트리를 저장하기 위한 권한 테이블 - 상기 권한 테이블 내의 각 엔트리는 우선순위 값과, 상기 운영 체제의 보안 정책들을 실행하는 규칙들과 연관됨 -; 및

제2 복수의 엔트리를 저장하기 위한 캐시 테이블 - 상기 캐시 테이블 내의 각 엔트리는 상기 보안 평가들의 결과들을 포함함 -

을 포함하고,

상기 보안 평가기는, 상기 권한 테이블의 엔트리들과 연관되는 우선순위 값들에 기초하는 순서에 따라 상기 권한 테이블의 엔트리들을 조사하는 것에 의해 상기 보안 평가들을 실시하는, 방법.

## 청구항 23

제22항에 있어서,

상기 캐시 테이블의 각 엔트리는, 고유 데이터 오브젝트에 대응하고 상기 데이터 오브젝트의 사전 보안 평가의 결과를 저장하며,

상기 동작에 대응하는 상기 캐시 테이블의 엔트리는, 상기 동작을 수행하는데 필요한 데이터 오브젝트에 기초하여 생성되는 해시 값에 의해 인덱싱되는, 방법.

## 청구항 24

보안 평가 방법으로서,

운영 체제에서 특정 동작을 수행하기 위한 명령을 수신하는 단계; 및

권한 테이블 및 캐시 테이블을 포함하는 규칙 데이터베이스에 질의하는 것에 의해 상기 특정 동작을 위한 보안 평가를 수행하는 단계

를 포함하고, 상기 규칙 데이터베이스에 질의하는 것은,

상기 캐시 테이블이 상기 특정 동작이 필요로 하는 데이터 오브젝트에 대응하는 유효 엔트리를 가지는 경우, 상기 캐시 테이블로부터 상기 유효 엔트리를 검색하는 것; 및

상기 캐시 테이블이 상기 데이터 오브젝트에 대응하는 유효 엔트리를 가지지 않는 경우, 상기 데이터 오브젝트에 매칭되는 규칙을 가지는 엔트리를 찾기 위해 상기 권한 테이블을 조사하는 것 - 상기 권한 테이블은 상기 권한 테이블에서의 각 엔트리에 대한 우선순위 값에 기초하여 조사됨 -

을 포함하는, 방법.

## 청구항 25

제24항에 있어서,

상기 캐시 테이블의 상기 유효 엔트리는, 상기 특정 동작이 필요로 하는 데이터 오브젝트에 기초하는 사전 보안 평가의 결과를 저장하며, 상기 특정 동작은 데이터 오브젝트와 연관되며, 상기 캐시 테이블 내의 상기 엔트리는 상기 데이터 오브젝트를 기초로 생성되는 해시 값에 의해 인덱싱되는, 방법.

## 발명의 설명

### 기술 분야

[0001] 본원은 집중형 동작 관리에 관한 것이다.

### 배경 기술

[0002] 컴퓨터 시스템 상에 저장되어 실행되는 프로그램들의 아이덴티티(identity) 및 진위(authenticity)는 사용자들을 위한 기본적인 보안 문제이다. 본질적으로, 사용자들은 프로그램들이 광고될 때 수행하기 위해 사용자들이 함께 상호 작용하는 프로그램들을 원한다. 사용자들은 특정 프로그램을 신뢰할 때 문제점들에 직면할 수 있는 반면, 프로그램은 예상치 않은 방식으로 거동한다. 몇몇 상황들에서, 프로그램은 의도적으로 변경되거나, 바이러스 또는 다른 보안 문제가 존재할 수 있다. 종종, 단순하게는 프로그램은 사용자가 초기에 예상했던 것과는 상이하게 거동하는데, 그 이유는 프로그램이 신뢰되는 소스로부터 오지 않기 때문에, 프로그램이 다른 컴포넌트들과 호환 가능하지 않게 하는 몇몇 요소에서 변경되었기 때문에, 또는 프로그램이 몇몇 다른 이유들로 인해 비효과적이기 때문이다.

[0003] 따라서 이러한 문제를 다루려고 시도하는 많은 보안 프로그램들이 현대의 컴퓨터들 상에서 실행되고 있다. 이들 애플리케이션들 각각은 컴퓨터의 한 세트의 보안 요구들을 다룬다. 이들은 바이러스 방지 애플리케이션, 방화벽, 멀웨어(malware) 검출 프로그램, 서명 검사 메커니즘 등을 포함한다. 이들 프로그램들 중 몇몇은 운영 체제의 일부로서 제공되며, 몇몇은 인터넷 브라우저 내로 통합되는 반면, 다른 것들은 제3자인 판매사로부터 구매되거나 또는 심지어 다운로드된다.

[0004] 그러나, 이들 프로그램들은 아이덴티티들의 인증을 확립하기 위한 일관된 메커니즘들이 아니다. 이들은, 시스템에서 실행되는 임의의 동작이 신뢰할 수 없는 소스들로부터 와서 컴퓨터에 손상을 입히는 데이터 오브젝트들을 도입할 수 있을지라도, 운영 체제에서 실행되는 모든 동작들을 철저히 검사하도록 신뢰되지 않을 수 있다. 이들 프로세스들은 종종 운영 체제에서 최적으로 수행하지 못하며 전체 시스템 속도를 늦춘다. 이들 보안 프로그램들의 뒤죽박죽 특성은 소프트웨어 개발자들을 위한 한 세트의 확실적인 보안 인터페이스 프로그래밍을 도입하는 것을 어렵게 하거나 불가능하게 한다. 더 나쁜 건, 보안 평가 프로그램들의 이들 뒤죽박죽 모음들 중 몇몇은 아직 신뢰할 수 있는 것으로서 적절히 증명되지 않은 소스들로부터 왔을 수 있다.

[0005] 요구되는 것은 아이덴티티들의 인증을 확립하기 위한 일관된 메커니즘이다. 구체적으로, 요구되는 것은 일관된 성능, 운영 체제에서 발생하는 동작들의 포괄적인 검사, 및 확실적인 소프트웨어 개발 지원을 제공하기 위해 운영 체제와 완전히 통합되는 보안 평가 메커니즘이다.

### 발명의 내용

[0006] 본 발명의 몇몇 실시예들은 디바이스를 위한 신규한 보안 프레임워크를 제공한다. 몇몇 실시예들에서, 이러한 프레임워크는 디바이스의 운영 체제의 일부이다. 몇몇 실시예들의 프레임워크는 디바이스 상에서 실행되는 애플리케이션에 대하여 수행되도록 요구하는 상이한 동작들에 대한 보안 정책 평가들을 수행하는 보안 평가기(security assessor)를 포함한다. 그러한 동작들의 예들은 애플리케이션의 설치와 실행, 및 상기 애플리케이션에 의한 콘텐츠 파일들의 열기(예컨대, 문서들의 열기)를 포함한다.

[0007] 몇몇 실시예들에서, 디바이스는 애플리케이션에 대하여 상이한 동작들을 위한 요청들을 수신하는 동작 개시기(operation initiator)를 갖는다. 그러한 요청을 수신할 때, 동작 개시기는 보안 정책 데이터 저장장치에 저장되는 보안 정책 규칙들에 기초하여, 요청된 동작의 실행 가능성을 검사하도록 보안 평가기에 지시한다. 몇몇 실시예들에서의 보안 정책 규칙들은, 프로그램의 소스 또는 발행자를 검증하는 등등을 위해, 다운로드된 프로그램들의 유효성을 검사하기 위해 사용된다. 보안 정책 규칙이 요청된 동작을 허용한다면, 보안 평가기는 동작이 승인되었음을 동작 개시기에 알리며, 동작 개시기는 요청된 동작을 수행하도록 적절한 동작 핸들러(handler)(예컨대, 설치기, 실행기, 또는 파일 오픈너(file opener))에 지시한다.

[0008] 몇몇 실시예들에서, 보안 정책들은 규칙 데이터베이스에 저장되는 다양한 규칙들 또는 명령어들로 구체화된다. 규칙 데이터베이스에서의 규칙들은 이들이 컴퓨터의 보안 정책을 준수하기 위해 데이터 오브젝트들(예컨대, 프로그램 코드 또는 열게 될 문서들)에 요구되는 것을 규정한다. 몇몇 실시예들에서, 규칙 데이터베이스는 규칙 데이터베이스에 대한 질의들을 보다 효율적으로 만들기 위해 다양한 테이블들을 포함한다. 권한 테이블

(authority table) 및 캐시 테이블(cache table)은 그러한 테이블들의 2개의 예들이다.

[0009] 몇몇 실시예들에서, 보안 정책 하에서 데이터 오브젝트의 보안을 평가하는 것은 그들의 우선순위들에 의해 제시된 순서로 규칙 데이터베이스에서의 권한 테이블의 엔트리들을 조사하는 것을 요구한다. 이들 실시예들에서 데이터 오브젝트에 대한 임의의 주어진 보안 평가에 대해, 보다 낮은 우선순위 규칙들은 보다 높은 우선순위 규칙들이 적용 가능하지 않을 때만 적용 가능해진다. 보다 높은 우선순위 규칙은 이들이 중첩하는 보다 낮은 우선순위 규칙들을 대체한다. 그러므로, 보다 높은 우선순위 규칙에 대한 임의의 변경은 잠재적으로 보다 낮은 우선순위 규칙들의 적용 가능성을 변경한다.

[0010] 몇몇 실시예들에서의 동작 개시기는 적용을 위해 요청되는 각각의 동작에 대한 보안 평가를 수행한다. 다른 실시예들에서, 동작 개시기는 애플리케이션을 위해 요청되는 동작들 중 몇몇에 대해서만 보안 평가를 수행한다. 예를 들어, 몇몇 실시예들에서, 평가는 애플리케이션에 의해 새롭게 수신된 파일의 초기 열기에 대해서만 또는 새롭게 수신된 애플리케이션의 설치에 대해서만 수행된다. 이러한 접근법을 용이하게 하기 위해, 몇몇 실시예들의 디바이스는 그러한 파일들이 먼저 디바이스로 로딩되거나 처음으로 열릴 때(예컨대, 먼저 네트워크를 통해 또는 디바이스의 주변 인터페이스를 통해 다운로드된) 새롭게 수신된 애플리케이션들 및 새롭게 수신된 파일들과 태그를 연관시킨다.

[0011] 몇몇 실시예들의 보안 평가기는 데이터 파일들을 포함한 ("아카이브(archive)"와 같은) 데이터 구조의 유효성을 검증함으로써 하나 이상의 데이터 파일들의 유효성을 승인하거나 또는 승인하지 않는다. 그러한 몇몇 실시예들에서, 아카이브들은 데이터 파일들을 안전하게 인증하며 이들 아카이브들에 포함된 데이터 파일들의 소스를 식별하기 위해 사용될 수 있는 아이덴티티의 서명을 지닌다. 보다 구체적으로, 이들 실시예들의 운영 체제는 이들 문서들 또는 데이터 파일들을 포함한 아카이브 구조들에 내장된 서명에 기초하여 문서들 또는 데이터 파일들을 승인하거나 승인하지 않는 규칙들을 규칙 데이터베이스에서 포함한다. 승인된 아카이브 내의 문서는 자동적으로 승인된다.

[0012] 전술된 요약은 본 발명의 몇몇 실시예들에 대한 간단한 소개로서 역할하도록 의도된다. 이는 본 문서에 개시된 모든 발명의 요지의 소개 또는 개괄이도록 의도되지 않는다. 이어지는 상세한 설명 및 상세한 설명에서 참조되는 도면들은 요약에서 설명된 실시예들뿐만 아니라 다른 실시예들을 추가로 설명할 것이다. 따라서, 본 문서에 의해 설명된 실시예들 모두를 이해하기 위해, 요약, 상세한 설명, 및 도면들의 전체적인 검토가 요구된다. 더구나, 청구된 요지들은 요지들의 사상으로부터 벗어남이 없이 다른 특정 형태들로 구체화될 수 있기 때문에, 청구된 요지들은 요약, 상세한 설명, 및 도면들에서의 예시적인 상세 사항들에 의해 제한되는 것이 아니라, 오히려 첨부된 특허청구범위에 의해 한정된다.

### 도면의 간단한 설명

[0013] 본 발명의 신규한 특징들이 첨부된 특허청구범위에 제시되어 있다. 그러나, 설명을 위해, 본 발명의 수 개의 실시예들이 다음의 도면들에 제시되어 있다.

<도 1>

도 1은 수행하도록 요청되는 상이한 동작들의 보안을 평가하는 컴퓨팅 디바이스의 운영 체제를 도시하는 도면.

<도 2>

도 2는 요청된 동작들을 취급하고 승인하기 위해 운영 체제가 사용하는 프로세스를 개념적으로 도시하는 도면.

<도 3>

도 3은 운영 체제의 보안 정책들에 기초하여 요청된 동작의 보안 평가를 수행하기 위한 예시적인 프로세스를 개념적으로 도시하는 도면.

<도 4 내지 도 6>

도 4 내지 도 6은 보안 평가기 및 규칙 데이터베이스를 사용하여 보안 평가를 수행할 때 운영 체제에서의 데이터의 흐름을 도시하는 도면.

<도 7>

도 7은 운영 체제의 보안 정책들을 결정하기 위한 규칙들 또는 명령어들을 저장하는 권한 테이블을 개념적으로

도시하는 도면.

<도 8>

도 8은 부가적인 필드(field)를 포함하는 권한 테이블을 개념적으로 도시하는 도면.

<도 9>

도 9는 각각의 레코드(record) 또는 엔트리에서 단 하나의 규칙 필드를 포함하는 권한 테이블을 개념적으로 도시하는 도면.

<도 10>

도 10은 다수의 엔트리들이 단일 보안 정책을 표현하기 위해 사용된 권한 테이블을 도시하는 도면.

<도 11>

도 11은 예시적인 보안 정책을 위한 한 세트의 벤 다이어그램(Venn diagram)들을 도시하는 도면.

<도 12>

도 12는 다른 예시적인 보안 정책을 위한 한 세트의 벤 다이어그램들을 도시하는 도면.

<도 13>

도 13은 권한 테이블을 사용함으로써 보안 평가를 수행하기 위한 프로세스를 개념적으로 도시하는 도면.

<도 14>

도 14는 권한 테이블 및 캐시 테이블을 포함하는 규칙 데이터베이스를 도시하는 도면.

<도 15>

도 15는 동작 개시기로부터의 요청에 기초하여 보안 평가를 하기 위해 권한 테이블 및 캐시 테이블 둘 모두를 사용하는 보안 평가기를 도시하는 도면.

<도 16>

도 16은 보안 평가기에 의해 데이터 오브젝트를 위한 캐시 테이블 엔트리를 생성하고 저장하는 것을 도시하는 도면.

<도 17>

도 17은 보안 평가를 수행하기 위한 권한 테이블 및 캐시 테이블 둘 모두를 사용하는 프로세스를 개념적으로 도시하는 도면.

<도 18>

도 18은 사용자에게 의한 규칙 데이터베이스에서의 변화를 따라 규칙 테이블을 유지하는 프로세스를 개념적으로 도시하는 도면.

<도 19>

도 19는 보안 평가를 수행하기 위해 그 자신의 식별자를 삽입하는 운영 체제를 갖는 컴퓨터를 도시하는 도면.

<도 20>

도 20은 보안 평가 동작의 일부로서 태그(tag)를 검사하는 운영 체제를 도시하는 도면.

<도 21 및 도 22>

도 21 및 도 22는 데이터 오브젝트들에서 태그들을 식별하는 동작 개시기를 사용하여 보안 평가를 수행할 때 운영 체제에서의 데이터의 흐름을 도시하는 도면.

<도 23>

도 23은 태그 비트들을 프로세싱하기 위한 규칙들을 갖는 규칙 데이터베이스를 포함하는 운영 체제를 도시하는 도면.

<도 24>

도 24는 서명들을 포함하는 데이터 아카이브들을 수신 및 저장하는 컴퓨터를 도시하는 도면.

<도 25>

도 25는 데이터 아카이브들에서의 서명들에 기초하여 데이터 파일들 또는 문서들의 보안 평가들을 하기 위해 규칙 데이터베이스에서 규칙들을 포함하는 운영 체제를 도시하는 도면.

<도 26>

도 26은 문서 열기 동작들의 보안 평가들을 수행하는 프로세스를 개념적으로 도시하는 도면.

<도 27>

도 27은 본 발명의 몇몇 실시예들을 구현시키는 전자 시스템을 개념적으로 도시하는 도면.

### 발명을 실시하기 위한 구체적인 내용

- [0014] 다음의 설명에서, 다수의 상세 사항들이 설명의 목적을 위해 제시된다. 그러나, 당업자는 본 발명이 이들 특정 상세 사항들의 사용 없이 실시될 수 있음을 인식할 것이다. 다른 경우들에서, 불필요한 상세 사항으로 본 발명의 설명을 모호하게 하지 않기 위하여 잘 알려진 구조물들 및 디바이스들이 블록도 형태로 도시된다.
- [0015] I. 보안 평가를 갖는 운영 체제
- [0016] 도 1은 수행하도록 요청되는 상이한 동작들의 보안을 평가하는 컴퓨팅 디바이스의 운영 체제(100)를 도시한다. 요청된 동작들은 애플리케이션들의 실행, 애플리케이션들의 설치, 문서들의 열기 등일 수 있다. 요청된 동작의 보안은 운영 체제에 의해 제공된 보안 프레임워크의 일부인 보안 평가기에 의해 평가된다. 이러한 보안 평가에 기초하여, 운영 체제는 요청된 동작을 종료하거나 또는 그 동작이 진행되게 한다.
- [0017] 도 1에 도시된 바와 같이, 운영 체제(100)는 보안 평가기(110), 규칙 데이터베이스(120), 동작 개시기(130), 동작 요청기(140), 설치 모듈(170), 실행 모듈(172), 및 콘텐츠 열기 모듈(174)을 포함한다. 도 1은 또한 운영 체제(100)가 보안 정책을 평가하게 되는 애플리케이션(150) 및 데이터 파일(160)을 도시한다. 보안 평가기(110) 및 규칙 데이터베이스(120)는 애플리케이션 프로그래밍 인터페이스(API)(190)를 제공하는 보안 프레임워크(180)의 일부이다.
- [0018] 운영 체제(100)는 컴퓨터 하드웨어 자원들을 관리하며 상이한 소프트웨어 애플리케이션 프로그램들을 위해 공통 서비스들을 제공하는 한 세트의 프로그램들이다. 운영 체제(100)는 입력, 출력, 및 메모리 할당과 같은 기능들을 위한 컴퓨터 하드웨어와 애플리케이션 프로그램들 사이에서 중재자로서 작용한다. 몇몇 실시예들에서, 프로그램의 애플리케이션 코드는 하드웨어에 의해 실행되며, 운영 체제로부터 수신된 인터럽트들 또는 운영 체제에 대해 이루어진 호출들을 통해 운영 체제와 상호 작용한다.
- [0019] 애플리케이션(150)은 운영 체제 상에서 기능들을 동작시키며 수행하기 위한 실행 가능한 데이터 오브젝트이다. 애플리케이션(150)의 예들은 워드 프로세서, 웹 브라우저, 게임, 미디어 편집 애플리케이션 등을 포함한다. 몇몇 실시예들에서, 애플리케이션(150)은 운영 체제(100)가 실행되고 있는 컴퓨터 상에 하나 이상의 실행 가능한 프로그램들을 설치하는 설치 프로그램이다.
- [0020] 몇몇 실시예들에서, 애플리케이션(150)은 애플리케이션의 사용자가 애플리케이션의 진위를 판단하게 하기 위해 아이덴티티의 하나 이상의 서명들을 내장한다. 애플리케이션(150)에 내장되는 서명은 애플리케이션의 소스 또는 발행자의 보안된 식별자이다. 데이터 오브젝트(예컨대, 애플리케이션)는, 데이터 오브젝트의 서명자에게만 알려져 있는 비밀 사설 키(secret private key)를 사용함으로써 데이터 오브젝트의 콘텐츠에 기초하여 서명이 생성될 때, "서명"된다. 따라서, 그러한 서명은 데이터 오브젝트의 서명자 또는 소스를 식별하며 데이터 오브젝트의 완전성을 보호할 수 있다. 데이터 오브젝트의 서명을 인증하기 위해, 즉 서명이 실제로 그의 알려진 소스/서명자에 의해 데이터 오브젝트로부터 생성됨을 검증하기 위해, 데이터 오브젝트의 수신인(즉, 운영 체제)은 데이터 오브젝트의 콘텐츠와 함께 서명을 인증하기 위해 공개 키를 사용한다. 공개 키는 알려진 소스/서명자로부터 온 것으로 알려져 있으며, 수신인은 (예컨대, 신뢰된 인증 기관에 질의함으로써 또는 수신인 자신의 데이터베이스를 검사함으로써) 공개 키가 실제로 알려진 소스/서명자에 의해 발행됨을 독립적으로 검증할 수 있다.
- [0021] 몇몇 실시예들에서, 데이터 오브젝트의 서명은 데이터 오브젝트와 연관된 공개 키 증명서(public key certificate)로부터 추출된다. 공개 키 증명서(또한, 디지털 증명서 또는 아이덴티티 증명서로서 알려짐)는 아

이덴티티 - 개인 또는 조직의 명칭, 그들의 주소 등과 같은 정보 - 와 공개 키를 결합하기 위해 디지털 서명을 사용하는 전자 문서이다. 증명서는 공개 키가 개인에 속함을 검증하기 위해 사용될 수 있다. 증명서 상의 서명들은 아이덴티티 정보와 공개 키가 함께 속하는 증명서 서명자에 의한 증명들이다. 증명서에서의 정보 및 애플리케이션 데이터 오브젝트의 콘텐츠 둘 모두에 기초하여 생성되거나 서명되는 서명은, 인증될 때, 애플리케이션 데이터 오브젝트의 콘텐츠가 변조(tempered)되지 않았으며, 증명서에서의 정보가 진짜임을 증명한다. 서명 및 소스 식별자들(예컨대, 벤더 ID 및 공개 키)에 더하여, 증명서는 또한 서명이 어떻게 인증될 수 있는지에 대한 정보(예컨대, 서명을 생성하기 위해 사용되는 알고리즘의 ID)를 포함한다.

[0022] 도 1의 예에서, 운영 체제(100)는 서명 인증 동작을 수행한다. 몇몇 실시예들에서, 운영 체제(100)는 운영 체제가 애플리케이션의 서명을 인증하는 데 실패한다면 애플리케이션(150)이 실행되거나 설치되는 것을 방지한다. 몇몇 실시예들에서, 운영 체제(100)는, 애플리케이션이 한 세트의 보안 정책들에 따라 신뢰할 수 있는 것으로 결정된 근원으로부터 오지 않았다면, 심지어 인증된 애플리케이션의 동작이 진행되는 것을 방지한다.

[0023] 데이터 파일(160)은 운영 체제에서 실행될 수 있는 하나 이상의 애플리케이션들과 연관된 데이터 오브젝트이다. 데이터 파일(160)의 예들은 텍스트 파일, 워드 프로세서 문서, 스프레드시트, 미디어 파일 등을 포함한다. 데이터 파일(160)은 데이터 파일을 관독하거나 열 수 있는 애플리케이션에 의해 관독되거나 열린다. 몇몇 실시예들에서, 데이터 파일(160)은 서명들 또는 증명서들을 포함하지 않는 문서들을 포함한다. 데이터 파일은 또한 데이터 파일이 서명들 또는 증명서들과 연관될 수 있도록 아카이브 구조에 위치될 수 있다. 이하의 도 25 및 도 26은 몇몇 실시예들의 보안 평가가 어떻게 데이터 아카이브들에서의 데이터 파일들을 취급하는지를 설명한다.

[0024] 설치 모듈(170)은 애플리케이션(150)과 같은 애플리케이션들의 설치를 감독하는 운영 체제(100) 내에서의 모듈이다. 몇몇 실시예들에서, 설치 모듈(170)은 운영 체제를 통한 설치를 위해 애플리케이션(150)의 프로세싱을 시작하기 전에 동작 개시기(130)로부터의 "진행" 명령을 대기한다. 이들 실시예들 중 몇몇에서, 설치 모듈은 한 세트의 보안 정책들에 따른 한 세트의 요건들의 준수뿐만 아니라 애플리케이션의 서명을 검증하기 위해 애플리케이션 데이터 오브젝트(150)를 보안 평가기(110)에 전달한다. 설치 모듈(170)은 애플리케이션(150)이 인증되고 그의 소스가 검증됨을 표시하는 통신을 동작 개시기(130)로부터 수신할 때만 애플리케이션(150)을 설치하도록 진행한다.

[0025] 실행 모듈(172)은 운영 체제(100)에서의 실행을 위해, 애플리케이션(150)과 같은 애플리케이션들을 시작하는 운영 체제(100) 내에서의 모듈이다. 설치 모듈(170)과 같이, 몇몇 실시예들에서의 실행 모듈(172)은 운영 체제(100)를 통한 실행을 위해 애플리케이션(150)을 시작하기 전에 동작 개시기(130)로부터의 "진행" 명령을 대기한다. 이들 실시예들 중 몇몇에서, 실행 모듈은 애플리케이션의 서명을 검증하기 위해 그리고 한 세트의 보안 정책들에 따른 한 세트의 요건들을 만족시키기 위해 보안 평가기(110)에 애플리케이션 데이터 오브젝트(150)를 전달한다. 실행 모듈(172)은 애플리케이션(150)이 인증되고 그의 소스가 검증된 후 실행 모듈이 동작 개시기(130)로부터 시작 명령을 수신할 때만 애플리케이션(150)을 실행하도록 진행한다.

[0026] 콘텐츠 열기 모듈(174)은 데이터 파일(160)의 열기를 감독하는 운영 체제(100) 내에서의 모듈이다. 몇몇 실시예들에서, 콘텐츠 열기는 데이터 파일(160) 내의 콘텐츠에 대해 적합한 애플리케이션을 식별하고 이어서 애플리케이션에 콘텐츠를 전달하는 동작을 수반한다. 전송된 바와 같이, 데이터 파일의 콘텐츠는 서명이 동반될 수 있거나 동반되지 않을 수 있으며, 인증될 필요가 있거나 인증될 필요가 없을 수 있다. 인증될 필요가 있는 문서들에 대해, 문서는 문서의 서명을 검증하기 위해 그리고 한 세트의 보안 정책들에 따라 한 세트의 요건들을 만족시키기 위해 보안 평가기(110)에 전달된다. 콘텐츠 열기 모듈(174)은 데이터 파일(160)이 인증되고 그의 소스가 검증된 후에 콘텐츠 열기 모듈이 동작 개시기(130)로부터 시작 명령을 수신할 때만 문서 데이터 파일(160)을 열도록 진행한다.

[0027] 몇몇 실시예들에서, 데이터 파일(160)의 열기는 애플리케이션의 실행과 연관된다. 이들 실시예들 중 몇몇에서 콘텐츠 열기 모듈(174)은 데이터 파일(160)을 열기 위한 애플리케이션을 먼저 식별한다. 콘텐츠 열기 모듈(174)에 의해 식별된 애플리케이션이 운영 체제에서 이미 실행 중이라면, 콘텐츠 열기 모듈은 실행될 실행 모듈(172)에 데이터 파일을 전달한다. 식별된 애플리케이션이 현재 실행 중이지 않다면, 콘텐츠 열기 모듈(174)은 실행 모듈(172)이 애플리케이션을 실행하기 시작하게 한다. 식별된 애플리케이션의 실행은 운영 체제(100)가 애플리케이션이 인증되고 그의 소스가 검증됨을 결정한 후에만 발생한다.

[0028] 동작 요청기(140)는 운영 체제(100)에 의해 특정 동작의 성능을 요청하는 모듈이다. 그러한 동작들은 애플리케이션의 설치, 애플리케이션의 실행, 데이터 파일들의 열기 등을 포함할 수 있다. 몇몇 실시예들에서, 동작 요

청기(140)는 (예컨대, 문서 또는 애플리케이션에 대한 그래픽 아이콘을 선택함으로써) 문서를 열거나 특정 애플리케이션을 실행하기 위한 사용자 명령을 수신하는 (GUI와 같은) 사용자 인터페이스이다. 동작 요청기(140)는 또한, 운영 체제 내에서 실행되고 수행될 동작을 위해 동작 개시기(130)에게 요청을 하는 상이한 프로그램일 수 있다. 몇몇 실시예들에서, 동작 요청기(140)는 또한, 운영 체제(100)의 보안 정책이 동작의 발생을 허용하지 않는다면(예컨대, 애플리케이션(150)의 서명이 인증되는 데 실패하거나 애플리케이션(150)의 소스가 신뢰되지 않는다면), 동작 개시기(130)로부터 통지를 수신한다.

[0029] 동작 개시기(130)는 특정 동작이 발생하게 하거나 또는 이를 방지함으로써 운영 체제의 보안 정책을 시행하는 모듈이다. 동작 개시기(130)는 동작 요청기(140)로부터 특정 동작을 위한 요청을 수신한다. 요청에 기초하여, 동작 개시기(130)는 필요한 데이터 오브젝트들(예컨대, 애플리케이션(150) 또는 데이터 파일(160))을 획득한다. 동작 개시기(130)는 이어서 운영 체제(100)의 보안 정책들 하에서 요청된 동작을 평가하도록 보안 평가기(110)에 요청한다. 동작 개시기는 또한 요청된 보안 평가 동작들의 일부로서 보안 평가기에 수신된 데이터 오브젝트를 전달한다.

[0030] 보안 평가기(110)는 동작이 운영 체제의 보안 정책들을 준수하는지 여부를 결정하는 모듈이다. 보안 평가기(110)는 특정 동작에 관한 보안 정책의 평가를 요청하는 질의를 동작 개시기(130)로부터 수신한다. 보안 평가기(110)는 동작과 연관된 데이터 오브젝트들을 조사함으로써 동작이 보안 정책들을 준수하는지 여부를 결정한다. 예를 들어, 애플리케이션(150)을 설치하는 동작에 기초한 요청은 보안 평가기가 조사할 수 개의 데이터 오브젝트들(예컨대, 설치 프로그램 자체뿐만 아니라 애플리케이션(150)을 위한 설치 패키지의 설치 파일들)을 포함할 수 있다. 다른 예에서, 애플리케이션(150)에 의한 데이터 파일(160) 열기 요청은 보안 평가기(110)에 의해 조사될 애플리케이션(150)의 실행 가능한 프로그램 및 데이터 파일(160) 둘 모두를 위한 데이터 오브젝트들을 포함할 수 있다. 몇몇 실시예들에서, 애플리케이션(150) 및 데이터 파일(160)은 보안 평가의 대상으로 지칭된다. 보안 평가기(110)는 이어서 동작 개시기(130)로부터 수신된 질의에 응답한다. 예를 들어, 특정 동작이 보안 정책의 조건들을 만족한다면, 보안 평가기는 보안 정책이 만족됨을 동작 개시기(130)에 알린다. 다른 한편, 특정 동작이 보안 정책의 조건들을 만족하는 데 실패한다면, 보안 평가기는 특정 동작이 보안 정책에 실패했다고 동작 개시기에 응답한다.

[0031] 몇몇 실시예들에서, 보안 평가기(110) 및 규칙 데이터베이스(120)는 운영 체제(100)를 위한 보안 평가를 수행하는 보안 프레임워크(180)의 일부로서 구현된다. 운영 체제의 다른 컴포넌트들은 동작들의 보안 평가를 수행하기 위해 보안 프레임워크(180)에 의해 제공된 보안 API(190)를 사용할 수 있다. 도 1의 예에서, 동작 개시기(130)는 API(190)를 통해 보안 평가기(110)와 통신하는 운영 체제(100)의 컴포넌트이다.

[0032] 일단 동작 개시기(130)가 질의에 대해 보안 평가기(110)로부터 응답을 수신한다면, 동작 개시기(130)는 특정 동작이 진행되게 하거나 진행되지 않게 함(예컨대, 방지함)으로써 보안 정책을 시행한다. 동작이 진행되게 된다면, 동작 개시기(130)는 설치 모듈(170), 실행 모듈(172), 또는 콘텐츠 열기 모듈(174)이 동작 요청기(140)에 의해 요청된 동작을 수행하기 시작하게 한다. 동작이 허용되지 않는다면, 동작 개시기(130)는 동작 요청기(140)에 통지하며, 동작 요청기는 이어서 사용자에게 통지하고 요청된 동작을 종료한다. 이하의 도 2는 동작 개시기(130)에 의해 수행되는 예시적인 프로세스를 설명한다.

[0033] 데이터 오브젝트에 내장된 서명은, 그의 수신인에 의해 인증될 때, 데이터 오브젝트가 특정 소스로부터 온 것임을 검증한다. 몇몇 실시예들에서, 보안 평가기(110)는 데이터 오브젝트에서의 서명을 인증하며 데이터 오브젝트의 소스를 알아낸다. 보안 평가기에 의해 인증될 수 있는 서명을 생성하는 데 실패한(예컨대, 결함이 있거나 불완전한) 데이터 오브젝트는 신뢰할 수 없는 소스로부터 온 가능성이 있다. 그러한 상황에서, 보안 평가기(110)는 운영 체제의 보안 정책들 하에서, 데이터 오브젝트가 실행되거나 열리는 것을 방지하도록 동작 개시기(130)에 알린다 (몇몇 실시예들에서, 인증되는 데 실패한 서명을 갖는 문서 또는 프로그램은 결코 진행되지 않게 할 수 있다). 반대로, 보안 평가기(110)가 데이터 오브젝트의 서명을 인증할 수 있다면, 보안 평가기는, 보안 정책 하에서 다른 조건들이 또한 충족된다고 가정하면(예컨대, 데이터 오브젝트의 소스가 신뢰될 수 있다면), 요청된 동작을 계속하는 것이 안전함을 동작 개시기에 알릴 수 있다.

[0034] 몇몇 실시예들에 의해 구현된 보안 정책은 상이한 데이터 오브젝트들을 상이하게 조사한다. 몇몇 유형들의 데이터 오브젝트들에 대해, 운영 체제의 보안 정책들은 서명의 인증에 더하여 만족될 필요가 있는 추가 조건들을 가질 수 있다. 예를 들어, 보안 정책들은 일련의 증명서들에 기초하여 서명의 인증을 요구할 수 있거나, (코드 서명이 성공적으로 인증될지라도) 데이터 오브젝트의 소스가 특정 벤더가 될 것을 요구할 수 있다. 몇몇 실시예들에서, 보안 정책들은 사용자에 의해 변경될 수 있다. 예를 들어, 컴퓨터의 신뢰된 관리자는 보안 평가기가

임의의 서명들을 검증하거나 인증하지 않고 특정 데이터 오브젝트를 수용하도록 운영 체제의 보안 정책들을 변경할 수 있다. 몇몇 실시예들에서, 보안 평가기(110)는 동작 개시기(130)에 의해 제출되는 임의의 데이터 오브젝트에 관한 임의의 보안 정책을 구현하도록 임의로 프로그래밍될 수 있다.

[0035] 몇몇 실시예들에서, 데이터 오브젝트를 평가하기 위해 사용된 보안 정책들은 데이터 오브젝트가 보안 정책들을 준수하는지 여부를 결정하기 위해 보안 평가기(110)가 실행하는 명령어들의 세트로서 구현된다. 도 1의 예시적인 운영 체제에서, 보안 정책들을 구현하는 명령어들의 세트는 규칙 데이터베이스(120)에 저장된다. 보안 평가기(110)는 규칙 데이터베이스(120)로부터 명령어들의 세트를 검색하며, (i) 데이터 오브젝트를 분석하기 위해 (예컨대, 데이터 오브젝트의 소스가 허용 가능한지 여부를 결정하기 위해) 그리고 (ii) 분석에 기초하여 동작 개시기(130)에 응답을 발행하기 위해 검색된 명령어들을 사용한다. 규칙 데이터베이스에 저장된 하나 이상의 규칙들의 각각의 세트는, 운영 체제의 보안 정책을 준수하기 위해 데이터 오브젝트(예컨대, 애플리케이션(150) 또는 데이터 파일(160)과 같은 실행 가능한 프로그램 코드)에 요구되는 것을 규정하기 때문에, 때때로 "코드 요건"으로 지칭된다. 이하의 도 3은 보안 평가기(110)에 의해 수행되는 예시적인 프로세스를 설명한다. 코드 요건들은 도 9 및 도 13을 참조하여 추가로 상세히 이하에 설명될 것이다.

[0036] 규칙 데이터베이스(120)는 운영 체제(100)의 보안 정책들을 구현하기 위한 명령어들의 세트들을 저장하는 데이터베이스이다. 보안 평가기(110)는 동작 개시기(130)에 의해 이루어진 보안 질의에 응답하기 위해 규칙 데이터베이스(120)로부터 명령어들의 세트들을 검색한다. 규칙 데이터베이스는 도 7 내지 도 17을 참조하여, 이하에 추가로 설명된다.

[0037] 도 2는 몇몇 실시예들의 운영 체제가 요청된 동작들을 취급 및 승인하기 위해 사용하는 프로세스(200)를 개념적으로 도시한다. 구체적으로, 프로세스는 요청된 동작들을 위한 보안 평가들을 요청하며 승인된 동작들을 시작한다. 몇몇 실시예들에서, 프로세스(200)는 도 1의 동작 개시기(130)와 같은, 운영 체제(100) 내에서의 모듈에 의해 수행된다.

[0038] 프로세스(200)는 (예컨대, 동작 요청기(140)로부터) 동작 요청을 수신할 때(210에서) 시작한다. 몇몇 실시예들에서, 요청은 문서를 열거나 애플리케이션을 실행하기 위한 사용자 명령이다. 요청은 또한 운영 체제에서 특정 동작을 수행하기 위해 현재 실행 중인 애플리케이션 또는 시스템 프로세스에 의해 프롬프팅될 수 있다.

[0039] 다음으로, 프로세스는 요청을 보안 평가기에 전달한다(220에서). 도 1의 예에서, 요청은 또한 보안 정책 하에서 요청의 보안을 평가하기 위해 보안 평가기에 대해 요구되는 데이터 오브젝트들(예컨대, 애플리케이션(150) 또는 데이터 파일(160))을 포함한다. 예를 들어, 문서를 열기 위해 사용자 명령에 의해 개시된 동작 요청은 몇몇 경우들에서 문서를 여는 것이 애플리케이션의 호출을 요구하기 때문에 보안 평가기에 전달될 다수의 데이터 오브젝트들을 요구할 수 있으며, 따라서 보안 평가기에 전달된 요청은 문서 및 애플리케이션 둘 모두의 파일 취급들을 포함할 것이다.

[0040] 프로세스는 다음으로 보안 평가기(예컨대, 110)로부터 응답을 수신하며(230에서) 요청된 동작을 승인할지 여부를 결정한다(240에서). 몇몇 실시예들에서, 보안 평가기로부터의 응답은 요청이 운영 체제의 보안 정책을 통과하였는지 여부를 (예컨대, 동작 개시기(130)에 알림으로써) 표시한다. 프로세스(200)는 몇몇 실시예들에서, 응답이 요청이 보안 평가를 통과하였음(예컨대, 인증된 서명을 갖고 허용 가능한 소스로부터 옴)을 표시한다면, 요청된 동작이 진행되게 한다. 몇몇 실시예들에서, 프로세스(200)는, 응답이 달리 표시한다면(예컨대, 보안 평가기가 서명을 인증할 수 없거나 데이터 오브젝트가 허용 가능한 소스로부터 오지 않음), 요청된 동작을 종료하거나 중단한다.

[0041] 요청이 승인된다면, 프로세스(200)는 요청을 수신하고 승인된 동작을 수행할 수 있는 동작 핸들러에 요청을 전달한다(250에서). 도 1에 예시된 바와 같이, 설치 모듈(170), 실행 모듈(172), 및 콘텐츠 열기 모듈(174)을 포함한, 수 개의 동작 핸들러들이 요청들을 수신하고 동작들을 수행할 수 있다. 동작 핸들러에 요청을 전달한 후, 프로세스(200)는 종료한다.

[0042] 다른 한편, 프로세스(200)가 요청을 승인하지 않는다면, 프로세스는 디폴트 메시지를 동작 요청기(예컨대, 도 1의 140)로 반환하고(260에서), 동작 요청기는 이어서 사용자에게 통지한다(도시되지 않음). 더욱이, 프로세스(200)는 요청이 승인되지 않을 때 동작 핸들러에 요청을 전달하지 않는다. 따라서, 170, 172, 및 174와 같은 동작 핸들러들 중 어떤 것도 요청이 승인되지 않을 때 요청된 동작을 수행하지 않을 것이다. 동작 요청기에 디폴트 메시지를 반환한 후, 프로세스(200)는 종료한다.

[0043] 몇몇 실시예들은 프로세스(200)의 변형들을 수행한다. 예를 들어, 프로세스(200)의 특정 동작들은 도시되고 설

명된 정확한 순서로 수행되지 않을 수 있다. 특정 동작들은 하나의 연속적인 시리즈의 동작들로 수행되지 않을 수 있으며, 상이한 특정 동작들이 상이한 실시예들에서 수행될 수 있다.

- [0044] 도 3은 운영 체제의 보안 정책들에 기초하여 요청된 동작의 보안 평가를 수행하기 위한 예시적인 프로세스(300)를 개념적으로 도시한다. 구체적으로, 프로세스(300)는 요청된 동작에 관한 보안 평가를 프로세스가 할 수 있게 하는 규칙에 요청을 매칭시키기 위한 규칙 데이터베이스를 사용한다. 몇몇 실시예들에서, 프로세스(300)는 도 1의 보안 평가기(110)와 같은, 운영 체제에서의 모듈에 의해 수행된다.
- [0045] 프로세스(300)는 동작 개시기로부터 요청을 수신할 때(310에서) 시작한다. 요청은 특정한 요청된 동작을 위해 보안 평가가 요구된다는 것을 표시한다.
- [0046] 다음으로, 프로세스는 매칭되는 것을 발견하도록 규칙 데이터베이스에 질의한다(320에서). 프로세스(300)는 매칭되거나 요청에 적용 가능한 규칙들에 대한 규칙 데이터베이스에서의 엔트리들을 조사한다. 요청에 매칭되지 않는 규칙은 요청을 위한 보안 평가를 하는 데는 사용될 수 없다. 예를 들어, 특정 명칭을 갖는 애플리케이션을 검사하기 위한 규칙은 상이한 명칭을 갖는 애플리케이션을 검사하는 데는 사용될 수 없다. 이하에 추가로 논의되는 바와 같이, 규칙 데이터베이스에서의 다수의 엔트리들은 요청을 동시에 매칭시킬 수 있으며, 단지 최고 우선순위를 갖는 규칙만이 적용될 것이다.
- [0047] 프로세스는 그 후 매칭되거나 적용 가능한 규칙이 규칙 데이터베이스에서 발견되었는지 여부를 결정한다(330에서). 프로세스(300)가 적어도 하나의 매칭되는 규칙을 찾는다면, 프로세스는 340으로 진행한다. 그렇지 않다면 프로세스는 360으로 진행한다.
- [0048] 360에서, 프로세스는 요청에 관한 어떤 적용 가능한 규칙들 또는 명령어들이 없음을 표시하기 위해 디폴트 응답을 동작 개시기에 반환한다. 몇몇 실시예들에서, 디폴트 보안 정책은 규칙 데이터베이스에서 매칭되는 규칙들이 없는 동작에 대한 요청을 허용하지 않는 것이다. 몇몇 실시예들에서, 디폴트 보안 정책은 규칙 데이터베이스에 의해 구체적으로 금지되지 않은 요청들을 허용한다. 몇몇 실시예들에서, 디폴트 정책은 운영 체제가 사용자에게 통지하도록 요구한다. 디폴트 응답을 반환한 후, 프로세스(300)는 종료한다.
- [0049] 340에서, 프로세스는 요청에 관한 보안 평가를 한다. 몇몇 실시예들에서, 프로세스는 요청과 연관된 데이터 오브젝트들(예컨대, 애플리케이션(150) 및/또는 데이터 파일(160))에 매칭되는 규칙들 또는 적용 가능한 명령어들의 세트를 적용함으로써 보안 평가를 수행한다. 몇몇 실시예들에서, 매칭되는 규칙들의 세트는 실행 가능한 프로그램으로서 보안 평가기에서 실행한다. 몇몇 실시예들에서, 매칭되는 규칙들의 세트는 보안 평가기가 데이터 오브젝트의 소스, 데이터 오브젝트의 아이덴티티, 및 데이터 오브젝트의 유형과 같은 변수들에 기초하여 보안 평가를 하게 한다.
- [0050] 일단 보안 평가가 이루어졌다면, 프로세스는 요청된 동작이 진행되게 할지 진행되지 않게 할지 여부를 결정하기 위해 동작 개시기에 대한 보안 평가를 반환한다(350에서). 프로세스(300)는 동작 개시기에 보안 평가를 반환한 후 종료한다.
- [0051] 몇몇 실시예들은 프로세스(300)의 변형들을 수행한다. 예를 들어, 프로세스(300)의 특정 동작들은 도시되고 설명된 정확한 순서로 수행되지 않을 수 있다. 특정 동작들은 하나의 연속적인 시리즈의 동작들로 수행되지 않을 수 있으며, 상이한 특정 동작들이 상이한 실시예들에서 수행될 수 있다.
- [0052] 도 4 내지 도 6은 보안 평가기 및 규칙 데이터베이스를 사용하여 보안 평가를 수행할 때 운영 체제(100)에서의 데이터의 흐름을 도시한다. 도 1에서처럼, 도 4 내지 도 6에서의 운영 체제(100)는 보안 평가기(110), 규칙 데이터베이스(120), 동작 개시기(130), 동작 요청기(140), 설치 모듈(170), 실행 모듈(172), 및 콘텐츠 열기 모듈(174)을 포함한다.
- [0053] 도 4는 요청된 동작이 애플리케이션을 설치하기 위한 것일 때 운영 체제(100) 내에서의 데이터 흐름을 도시한다. 동작 데이터 흐름은 애플리케이션 X를 설치하기 위해 동작 개시기(130)에 요청할 때 동작 요청기(140)에서 시작한다(동작 '1'). 이러한 요청은, 몇몇 실시예들에서, 애플리케이션 X를 설치하기 위한 사용자 명령에 기초한다. 사용자 명령은 사용자 인터페이스에 의해 수신될 수 있다.
- [0054] 동작 개시기(130)는 이어서 운영 체제의 보안 정책들 하에서 애플리케이션 X 설치의 보안을 평가하도록 보안 평가기(110)에 요청한다(동작 '2'). 몇몇 실시예들에서, 보안 평가기(110)에 대한 이러한 요청은 설치 파일들 및 패키지들과 같은, 애플리케이션 X를 설치하기 위해 필요한 데이터 오브젝트들에 대한 취급들을 포함한다.
- [0055] 보안 평가기(110)는, 요청을 수신할 때, 보안 정책 하에서 설치 요청에 매칭되는 규칙들 또는 명령어들에 대해

규칙 데이터베이스(120)에 질의한다(동작 '3'). 규칙 데이터베이스는 데이터베이스에서의 질의에 매칭되는 규칙들 또는 명령어들을 보안 평가기(110)에 제공한다(동작 '4'). 보안 평가기(110) 및 규칙 데이터베이스(120) 사이에서의 이러한 질의 프로세스는, 매칭 규칙이 규칙 데이터베이스에서 발견되거나 규칙 데이터베이스에서 적용 가능한 규칙들이 없다는 결정이 이루어질 때까지 계속된다.

[0056] 일단 매칭 규칙이 발견된다면, 보안 평가기(110)는 애플리케이션 X의 설치와 관련된 데이터 오브젝트들에 매칭 규칙들을 적용함으로써 보안 평가를 수행한다. 몇몇 실시예들에서, 규칙은 데이터 오브젝트들을 프로세싱하고 데이터 오브젝트들이 운영 체제의 보안 정책들을 준수하는지 여부를 결정하기 위한 프로그램 명령어들의 형태를 취한다. 데이터 오브젝트들의 프로세싱은, 몇몇 실시예들에서, 데이터 오브젝트들 내에서, 애플리케이션 X의 설치와 관련된 서명들을 검증하는 것을 포함한다. 몇몇 실시예들에서, 보안 정책들의 준수는 서명이 성공적으로 인증되었는지 여부 및 서명이 신뢰되는 소스로부터 온 것인지 여부에 기초하여 결정된다. 보안 평가는 이어서 동작 개시기(130)로 다시 전달된다(동작 '5'). 도 4에 도시된 예에서, 보안 평가기는 애플리케이션 X를 설치하는 것이 OK임을 표시한다. 이는 애플리케이션 X의 설치와 관련된 데이터 오브젝트들이 규칙 데이터베이스에 저장된 보안 정책들을 준수하는 것으로 발견되었기 때문에 보안 평가이다.

[0057] 애플리케이션 X를 설치하는 것이 OK인 보안 평가를 수신한 후, 동작 개시기(130)는 설치 모듈(170)에게 설치를 시작하고 애플리케이션 X를 컴퓨터에 설치하라고 명령한다. 실행 모듈(172) 및 콘텐츠 열기 모듈(174) 주위에서의 파선들은 이러한 동작이 이들 2개의 모듈들을 수반하지 않음을 표시한다.

[0058] 도 5는, 설치 동작을 위한 데이터 흐름을 도시하는 대신에, 도 5가 요청된 동작이 애플리케이션을 실행(예컨대, 시작 또는 호출)하기 위한 것일 때 운영 체제(100) 내에서의 데이터 흐름을 도시한다는 것을 제외하고는, 도 4와 유사하다. 동작 데이터 흐름은, 동작 요청기가 애플리케이션 X 실행 요청을 동작 개시기(130)에 할 때 동작 요청기(140)에서 시작한다(동작 '1'). 이러한 요청은, 몇몇 실시예들에서, 애플리케이션 X를 실행하기 위한 사용자 명령이다. 사용자 명령은 사용자 인터페이스에 의해 수신될 수 있다.

[0059] 동작 개시기(130)는 이어서 운영 체제의 보안 정책들 하에서 애플리케이션 X 실행의 보안을 평가하도록 보안 평가기에 요청한다(동작 '2'). 몇몇 실시예들에서, 이러한 요청은 애플리케이션 X의 실행 가능한 기계 코드와 같은, 애플리케이션 X를 실행하기 위해 필요한 데이터 오브젝트들에 대한 취급들을 포함한다.

[0060] 보안 평가기(110)는, 요청을 수신할 때, 보안 정책 하에서 애플리케이션을 실행하라는 요청에 매칭되는 규칙들 또는 명령어들에 대해 규칙 데이터베이스(120)에 질의한다(동작 '3'). 규칙 데이터베이스는 데이터베이스에서의 질의를 충족시키는 규칙들 또는 명령어들을 보안 평가기(110)에 제공한다(동작 '4'). 보안 평가기(110) 및 규칙 데이터베이스(120) 사이에서의 이러한 질의 프로세스는, 매칭 규칙이 규칙 데이터베이스에서 발견되거나 규칙 데이터베이스에서 적용 가능한 규칙들이 없다는 결정이 이루어질 때까지 계속된다.

[0061] 일단 매칭 규칙이 발견된다면, 보안 평가기(110)는 애플리케이션 X의 실행과 관련된 데이터 오브젝트들에 매칭 규칙들을 적용함으로써 보안 평가를 수행한다. 몇몇 실시예들에서, 규칙은 데이터 오브젝트들을 프로세싱하고 데이터 오브젝트들이 운영 체제의 보안 정책들을 준수하는지 여부를 결정하기 위한 프로그램 명령어들의 형태를 취한다. 데이터 오브젝트들의 프로세싱은, 몇몇 실시예들에서, 데이터 오브젝트들 내에서, 애플리케이션 X의 실행과 관련된 서명들을 검증하는 것을 포함한다. 몇몇 실시예들에서, 보안 정책들의 준수는 서명이 성공적으로 인증되었는지 여부 및 서명이 신뢰되는 소스로부터 온 것인지 여부에 기초하여 결정된다. 보안 평가는 이어서 동작 개시기(130)로 다시 전달된다(동작 '5'). 도 5의 예에서, 보안 평가기는 애플리케이션 X를 실행하는 것이 OK임을 표시한다. 이는 애플리케이션 X의 실행과 관련된 데이터 오브젝트들이 규칙 데이터베이스에 저장된 보안 정책들을 준수하는 것으로 발견되었기 때문에 보안 평가이다.

[0062] 애플리케이션 X를 실행하는 것이 OK인 보안 평가를 수신한 후, 동작 개시기(130)는 실행 모듈(172)에게 애플리케이션 X를 시작하라고 명령한다. 설치 모듈(170) 및 콘텐츠 열기 모듈(174) 주위에서의 파선들은 이러한 동작이 이들 2개의 모듈들을 수반하지 않음을 표시한다.

[0063] 도 6은 도 6의 데이터 흐름이 문서를 열라는 요청에 기초한다는 것을 제외하고, 이전의 2개의 도면들과 유사하다. 데이터 흐름은 동작 요청기가 동작 개시기(130)에 문서 데이터 파일 X를 열기 위한 요청을 할 때 동작 요청기(140)에서 시작한다(동작 '1'). 이러한 요청은, 몇몇 실시예들에서, 데이터 파일 X를 열기 위한 사용자 명령이다. 사용자 명령은 사용자 인터페이스에 의해 수신될 수 있다.

[0064] 동작 개시기(130)는 이어서 운영 체제의 보안 정책들 하에서 데이터 파일 X 열기의 보안을 평가하도록 보안 평가기에 요청한다(동작 '2'). 몇몇 실시예들에서, 이러한 요청은 데이터 파일 X에 대한 취급들을 포함한다. 테

이터 파일 X가 애플리케이션 Y에 의해 열리는 문서이면, 이러한 요청은 몇몇 실시예들에서 또한 애플리케이션 Y의 실행 가능한 이진 파일들에 대한 취급들을 포함한다. 이러한 경우에, 문서 X 및 애플리케이션 Y 둘 모두는 문서 X를 열기 위해 필요한 데이터 오브젝트들로 간주된다. 그러므로, 보안 평가는 이들 데이터 오브젝트들 둘 모두에 대해 수행된다. 그러나, 몇몇 실시예들에서, 데이터 파일 X를 열기 위해 필요한 애플리케이션은 이미 실행 중이다(그리고 따라서 도 5의 예에서와 같이, 보안 평가를 이미 통과하였다). 따라서, 데이터 파일 X를 위한 취급만이 요청의 일부로서 보안 평가기에 전달된다.

[0065] 보안 평가기(110)는, 요청을 수신할 때, 보안 정책 하에서 문서 열기 요청에 매칭되는 규칙들 또는 명령어들에 대해 규칙 데이터베이스(120)에 질의한다(동작 '3'). 규칙 데이터베이스는 규칙 데이터베이스에서의 규칙들 또는 명령어들을 보안 평가기(110)에 제공한다(동작 '4'). 보안 평가기(110) 및 규칙 데이터베이스(120) 사이에서의 이러한 질의 프로세스는, 매칭 규칙이 규칙 데이터베이스에서 발견될 때까지 또는 규칙 데이터베이스에서 적용 가능한 규칙들이 없다는 결정이 이루어질 때까지 계속된다.

[0066] 일단 매칭 규칙이 발견되었다면, 보안 평가기(110)는 데이터 파일 X의 열기와 관련있는 데이터 오브젝트들(예컨대, 애플리케이션 Y)에 매칭 규칙들을 적용한다. 몇몇 실시예들에서, 규칙들은 데이터 오브젝트들을 프로세싱하고 이들 데이터 오브젝트들이 운영 체제의 보안 정책들을 준수하는지 여부에 대해 결정하는 프로그램 명령어들의 형태를 취한다. 데이터 오브젝트들의 프로세싱은, 몇몇 실시예들에서, 데이터 파일 X의 열기와 관련있는 데이터 오브젝트들 내에서의 서명들을 검증하는 것을 포함한다. 몇몇 실시예들에서, 서명이 성공적으로 인증되었는지 여부 및 서명이 신뢰되는 소스로부터 온 것인지 여부에 기초하여 결정이 이루어진다. 보안 평가는 이어서 동작 개시기(130)로 다시 전달된다(동작 '5'). 도 6의 예에서, 보안 평가기는 데이터 파일 X를 여는 것이 OK임을 표시한다. 이것은 데이터 파일 X의 열기와 관련있는 데이터 오브젝트들(예컨대, 데이터 파일 X 자체 및 애플리케이션 Y)이 규칙 데이터베이스에 저장된 보안 정책들을 준수하는 것으로 발견되었기 때문에 보안 평가이다.

[0067] 데이터 파일 X를 여는 것이 OK인 보안 평가를 수신한 후, 동작 개시기(130)는 콘텐츠 열기 모듈(174)에게 애플리케이션 X를 시작하라고 명령한다. 실행 모듈(172) 및 설치 모듈(170) 주위에서의 파선들은 이러한 동작이 이들 2개의 모듈들을 수반하지 않음을 표시한다. 몇몇 실시예들에서, 문서의 열기가 컴퓨터에서 현재 실행 중이지 않은 애플리케이션의 실행을 요구할 때, 동작 개시기는 또한 실행 모듈(172)을 시작할 것이다.

## [0068] II. 규칙 데이터베이스

[0069] 상기 설명된 예들에서, 요청된 동작들과 연관된 데이터 오브젝트들은 운영 체제의 보안 정책들에 따라 평가된다. 이들 보안 정책들은 규칙 데이터베이스에 저장된 상이한 규칙들 또는 명령어들에서 구체화된다. "운영 체제의 보안 정책들"은 따라서 운영 체제에서 발생하는 동작들을 제어하는 프로그래밍 가능한 규칙들의 세트이지만, 운영 체제 자체의 프로그램들의 일부는 아니다. 섹션 II-C에서 논의되는 바와 같이, 이들 "운영 체제의 보안 정책들"은 컴퓨터 시스템의 벤더 또는 사용자들에 의해 변경될 수 있다.

[0070] 몇몇 실시예들에서, 규칙 데이터베이스는 규칙 데이터베이스에 대한 질의들을 보다 효율적으로 만들기 위해 다양한 테이블들을 포함한다. 2개의 그러한 테이블들 - 권한 테이블 및 캐시 테이블- 이 이하에서 설명된다.

### [0071] A. 권한 테이블

[0072] 몇몇 실시예들에서 규칙 데이터베이스는 권한 테이블을 포함한다. 그러한 몇몇 실시예들에서, 권한 테이블은 특정 동작이 운영 체제에서 발생하도록 허용할지 여부를 결정하기 위해 보안 평가기가 검색하는 규칙들을 포함한다. 권한 테이블에서의 규칙들은 이들이 컴퓨터의 보안 정책을 준수하기 위해 데이터 오브젝트들(예컨대, 프로그램 코드 또는 열게 될 문서들)에 요구되는 것을 규정한다. 따라서, 이들 규칙들 중 하나 이상의 규칙들의 각각의 세트는 몇몇 실시예들에서 "코드 요건"으로 지칭된다.

[0073] 코드 요건들은 특정 신뢰되는 소스에 속하도록 증명서에 벤더 ID 및/또는 공개 키를 요구하는 것과 같은, 대상에 서명하기 위해 사용된 증명서들 상의 임의의 상태들을 표현할 수 있다. 코드 요건은 또한, 이들 프로그램들의 일체형 부분들인 환경설정 사전(configuration dictionary)들에서의 엔트리들과 같은 다른 것들을 요구할 수 있다. 예를 들어, 코드 요건은 그 서명과 관계가 없는 "주소록 사용을 허용하는 자격을 가져야 함"(자격들은 프로그램들 내에서의 포함을 위해 일부에 의해 규정된 환경설정 사전들 중 하나임)을 서술할 수 있다.

[0074] 도 7은 운영 체제의 보안 정책들을 결정하기 위한 규칙들 또는 명령어들을 저장하는 권한 테이블(700)을 개념적으로 도시한다. 권한 테이블(700)은 엔트리들 또는 레코드들의 세트를 포함하며, 각각의 엔트리 또는 레코드는 코드 요건을 규정하는 하나 이상의 규칙들 또는 명령어들의 세트를 저장한다. 특정 보안 정책은 권한 테이블

내의 하나 이상의 엔트리들에 의해 표현될 수 있다.

- [0075] 도시된 바와 같이, 저장 디바이스(710)는 권한 테이블(700)을 저장하며, 권한 테이블은 레코드(721, 722, 723)들을 포함한 수 개의 엔트리들 또는 레코드들을 포함한다. 레코드들은 정책(731, 732, 733)들을 포함한 수 개의 보안 정책들을 표현한다. 각각의 레코드 또는 엔트리는 또한 필드(741 내지 744)들의 세트를 포함한다. 필드(741 내지 743)들은 '규칙 1' 내지 '규칙 n'으로 라벨링되어 있다. 필드(744)는 "동작"으로 라벨링되어 있다.
- [0076] 레코드의 각각의 필드는 운영 체제의 보안 평가자에게 논리 또는 산술 연산들의 세트를 수행하고 논리 또는 산술 연산들의 세트에 따라 결정하라고 지시하는 규칙들 또는 명령어들의 세트이다. 예를 들어, 레코드(721)의 필드(741)에서의 'X'는 보안 평가가 애플리케이션을 위한 것인지 여부를 알기 위해 검사하는 연산들의 세트일 수 있으며, 레코드(721)의 필드(742)에서의 'Y'는 증명서의 만료일을 검사하는 연산들의 세트일 수 있으며, 레코드(722)의 필드(742)에서의 'Z'는 애플리케이션이 신뢰되는 벤더 또는 소스로부터 온 것인지 여부를 검증하는 규칙들일 수 있는 반면, 필드(741)에서의 'W'는 열리는 파일이 인터넷으로부터 다운로드된 파일인지 여부를 검사하는 규칙일 수 있다. 엔트리는 모든 필드들을 채울 필요는 없다. 예를 들어, 엔트리(723)는 단지 하나의 채워진 규칙 필드(741)만을 갖는다. 레코드(723)를 위한 상기 필드에서의 명령어는 열기가 요청되는 문서가 인터넷으로부터 다운로드되었는지 여부를 검사할 수 있다.
- [0077] 레코드의 상이한 필드들에서의 명령어들의 상이한 세트들이 공동으로 단일 결정을 한다. 이러한 단일 결정은 '동작' 필드(744)에서 서술된 동작이 착수되어야 하는지 여부를 결정하기 위해 사용된다. 예를 들어, 레코드(722)의 동작 필드(744)에서의 '승인'은, 평가되는 데이터 오브젝트가 애플리케이션이고, X.509 표준 하에서 검증된 증명서를 가지며, 신뢰되는 벤더로부터 온다면, 특정 애플리케이션을 시작하라는 요청을 보안 평가기가 승인하게 할 것이다. 다른 한편, 엔트리(723)의 '미승인'은, 특정 콘텐츠 아이템이 인터넷으로부터 다운로드된다면, 특정 콘텐츠 아이템을 개방하라는 요청을 보안 평가기가 승인하지 않게 한다.
- [0078] 명령어들 또는 조건들의 그러한 집합은 보안 정책을 구성한다. 도 7의 예에서, 레코드(721)에서의 규칙들 또는 명령어들은 보안 정책 A(731)를 구성하고, 레코드(722)에서의 규칙들 또는 명령어들은 보안 정책 B(732)를 구성하며, 레코드(723)에서의 규칙들 또는 명령어들은 보안 정책 C(733)를 구성한다. 몇몇 실시예들에서, 보안 정책은 권한 테이블 내의 둘 이상의 레코드들에 걸치는 규칙들의 집합이다. 그러한 권한 테이블은 도 10에 대한 참조에 의해 이하에서 추가로 논의된다.
- [0079] 몇몇 실시예들에서, 규칙들의 몇몇은 매칭을 위한 것이며 규칙들의 몇몇은 검증을 위한 것이다. 매칭을 위한 규칙들은 특정 엔트리가 보안 평가를 위한 요청에 적용 가능한지 여부를 결정한다. 검증을 위한 규칙들은 요청이 규칙 테이블의 특정 엔트리 또는 레코드에 의해 구체화된 보안 정책을 준수하는지 여부를 결정한다. 예를 들어, 특정 벤더로부터 데이터 오브젝트의 유형을 거절하는 규칙은 검증을 위한 규칙인 반면, 게임 애플리케이션들에 대한 규칙의 적용 가능성을 제한하는 규칙은 매칭을 위한 규칙이다.
- [0080] 규칙 필드들 및 동작을 위한 필드에 더하여, 몇몇 실시예들은 다른 필드들을 포함한다. 도 8은 도 7에 도시된 규칙 필드들 및 동작 필드에 더하여, 권한 테이블(800)이 부가적인 필드를 포함한다는 점을 제외하고는, 권한 테이블(700)과 유사한 권한 테이블(800)을 개념적으로 도시한다. 권한 테이블(800)은 또한 다수의 엔트리(821 내지 823)들을 가지며, 각각의 엔트리는 다수의 필드(840 내지 844)들을 갖는다. 이들 필드들은 규칙 필드(841 내지 843)들 및 동작 필드(844)를 포함한다. 그러나, 권한 테이블(700)과 달리, 권한 테이블(800)에서의 각각의 엔트리는 또한 우선순위 필드(840)를 포함한다.
- [0081] 몇몇 실시예들의 우선순위 필드(840)는 요청에 매칭되는 규칙들을 찾기 위해 권한 테이블에서의 어떤 엔트리가 먼저 조사될지를 결정한다. 우선순위 필드에서 보다 높은 우선순위 값을 갖는 엔트리들은 보다 낮은 우선순위 값들을 갖는 엔트리들 전에 조사될 것이다. 도 8에 도시된 바와 같이, 레코드(822)는 우선순위 필드(840)에서 1.75의 우선순위 값을 갖는 반면, 레코드(823)는 1.25의 우선순위 값을 갖는다. 따라서, 레코드(822)는 레코드(823) 전에 탐색될 것이지만, 레코드(821)(2.70의 우선순위 값을 가짐)은 레코드(822, 823)들 둘 모두 전에 탐색될 것이다. 엔트리들의 우선순위는 다른 방식들로(예컨대, 정수를 사용함으로써, 최고 우선순위로부터 최저 우선순위까지의 연결 리스트(linked list)에 엔트리들을 배열함으로써, 또는 우선순위에 따라 엔트리들을 분류함으로써) 표현될 수 있다.
- [0082] 임의의 주어진 보안 평가 요청에 대해, 권한 테이블의 다수의 엔트리들이 요청에 매칭될 수 있다. 그러나, 몇몇 실시예들에서, 보안 평가기는 매칭 엔트리가 발견될 때 권한 테이블을 탐색하는 것을 정지한다. 예를 들어,

다수의 매칭들을 가질 수 있는 탐색에 대해, 보안 평가기는 간단히 최고 우선순위 매칭 엔트리인 제1 매칭 엔트리를 사용할 것이다. 따라서, 우선순위 필드의 사용은 다수의 매칭들의 가능성을 완화시켜, 단일 매칭 엔트리가 보안 평가를 위해 사용됨을 보장한다.

- [0083] 도 7 및 도 8의 예들에서, 권한 테이블에서의 엔트리는 논리 또는 산술 연산들을 수행하기 위한 명령어들의 세트를 각각 포함하는 수 개의 규칙 필드들을 포함한, 수 개의 필드들을 포함한다. 그러나, 몇몇 실시예들은 권한 테이블에서 단일 규칙 필드만을 갖고, 모든 규칙 필드들의 명령어들의 세트들을 단일 규칙으로서 표현한다. 권한 테이블의 레코드에서의 단일 규칙 필드는 특정 보안 정책 하에서 보안 평가를 하기 위해 필요한 명령어들 모두를 포함한다.
- [0084] 도 9는 각각의 레코드 또는 엔트리에 하나의 규칙 필드만을 포함하는 몇몇 다른 실시예들의 권한 테이블(900)을 개념적으로 도시한다. 권한 테이블(900)은 권한 테이블(800)과 유사하여, 우선순위 필드(941), 동작 필드(943), 및 다수의 엔트리(921 내지 923)들을 포함한다. 그러나, 수 개의 규칙 필드들을 포함하는 권한 테이블(800)과는 달리, 권한 테이블(900)은 하나의 규칙 필드(942)만을 포함한다.
- [0085] 각각의 레코드 또는 엔트리를 위한 규칙 필드(942)를 포함하는 코드 요건은 매칭 및 검증 명령어들 둘 모두를 포함한다, 요청에 대한 보안 평가를 수행하기 위해 필요한 명령어들 모두를 포함한다. 몇몇 실시예들에서, 이전에 언급된 바와 같이, 코드 요건은 하나 이상의 연결된 규칙들의 세트일 수 있다. 그러한 몇몇 실시예들에서, 코드 요건은 데이터 오브젝트의 소스를 검증하기 위해 요청된 데이터 오브젝트에 적용된다. 예를 들어, 엔트리(921)를 위한 규칙 필드(942)를 포함하는 코드 요건에서의 명령어들의 세트는 다음의 결정들 각각을 할 수 있다: (1) 보안 평가를 위한 요청이 애플리케이션을 위한 것인지 여부; (2) 애플리케이션이 신뢰되는 벤더 또는 소스로부터 온 것인지 여부; 및 (3) 증명서에서의 소스 식별 정보(예컨대, 벤더 ID, 공개 키 등)가 실제로 신뢰된 벤더 또는 소스에 속하는지 여부. 3개의 결정들 각각이 그의 대응하는 규칙을 만족할 때만, 동작 필드(943)에서의 동작이 적용될 것이다. 다른 한편, 3개의 결정들 중 하나 이상이 그의 대응하는 규칙을 만족시키는데 실패할 때, 레코드의 동작 필드는 적용되지 않을 것이다. 따라서, 동작 필드에 의해 규정된 동작은 레코드를 위한 단일 규칙에 기초하며, 보안 평가기에 전달될 것이다(예컨대, 요청된 동작의 승인 또는 미승인).
- [0086] 권한 테이블은 도 7 내지 도 10에 도시되는 것들과는 상이한 포맷들을 가질 수 있다. 예를 들어, 몇몇 실시예에서, 권한 테이블 엔트리는 "디스에이블"이라 불리는 필드를 포함할 수 있어, 디스에이블 필드가 소정 값을 포함할 때, 엔트리가 건너뛰어질 수 있게 하고, 보안 평가기가 탐색에서 엔트리를 고려하지 않게 할 것이다.
- [0087] 도 7을 참조하여 이전에 언급된 바와 같이, 운영 체제의 보안 정책들은 규칙 데이터베이스에서의 규칙들 및 명령어들에 의해 표현된다. 몇몇 실시예들에서, 권한 테이블에서의 각각의 엔트리는 하나의 특정 보안 정책(예컨대, 특정 벤더로부터 유효한 서명을 갖는 애플리케이션들만을 허용함)을 표현하기에 충분하다. 몇몇 실시예들에서, 보안 정책은 권한 테이블에서 둘 이상의 엔트리들에 의해 표현될 수 있다.
- [0088] 도 10은 다수의 엔트리들이 단일 보안 정책을 표현하기 위해 사용된 권한 테이블(1000)을 도시한다. 도 10은 도 11 및 도 12를 참조하여 설명될 것이다. 권한 테이블(1000)은 권한 테이블(900)과 유사하다. 권한 테이블은 다수의 엔트리(1021 내지 1027)들을 가지며, 엔트리들 각각은 3개의 필드(우선순위 필드(1041), 규칙 필드(1042), 및 동작 필드(1043))들을 갖는다. 그러나, 도 10은 또한 권한 테이블의 3개의 엔트리들(엔트리(1021, 1024, 1026)들)에 의해 구현되는 예시적인 보안 정책(1031)('보안 정책 1')을 도시한다. 보안 정책의 3개의 엔트리들은 상이한 우선순위(P1, P5, P8)들을 가져, 최고 매칭 우선순위를 갖는 엔트리만이 보안 평가를 하기 위해 사용되게 한다. 이 예에서, P1은 최고 우선순위를 갖고, P5의 우선순위는 P1의 것보다 낮으며, P8은 최저 우선순위를 갖는다.
- [0089] 도 11은 도 10의 예시적인 보안 정책(1031)에 대한 벤 다이어그램들의 세트를 도시한다. 다이어그램들은 3개의 권한 테이블 엔트리(P1, P5, P8)들 중에서 논리적 관계들을 도시한다. 다이어그램들은 또한 우선순위들의 사용에 의해 예시적인 보안 정책(1031)의 구성을 도시한다.
- [0090] 도 11은 보안 정책(1031), 2차원 벤 다이어그램(1120), 및 3차원 다이어그램(1130)을 포함한다. 보안 정책(1031)은 우선순위(P1)에 대한 레코드(또는 명령어들의 세트), 우선순위(P5)에 대한 레코드, 및 우선순위(P8)에 대한 레코드를 포함한다. 우선순위(P1)를 갖는 레코드에서의 규칙들(즉, 1021)은 "회사(X)로부터 유효한 증명서를 갖는 모든 워드 프로세싱 앱들"을 허용할 것이다. 우선순위(P5)를 갖는 레코드에서의 규칙들(즉, 1024)은 "회사(X)로부터의 어떠한 앱들도 없음"을 허용할 것이다. 우선순위(P8)를 갖는 레코드에서의 규칙들(즉, 1026)은 모든 애플리케이션들의 열기를 허용할 것이다. 몇몇 실시예들에서, 증명서는 서명이 인증되었다면 "유효

한" 것으로 간주된다.

- [0091] 2D 벤 다이어그램(1120)은 3개의 타원형(1121 내지 1123)들을 포함한다. 3D 다이어그램(1130)은 동일한 3개의 타원들을 3차원 방식으로 도시한다. 타원(1121)은 우선순위(P1)를 갖는 규칙들을 나타내고, 타원(1122)은 우선순위(P5)를 갖는 규칙들을 나타내며, 타원(1123)은 우선순위(P8)를 갖는 규칙들을 나타낸다.
- [0092] 2D 벤 다이어그램(1120)은 따라서 3개의 권한 테이블 엔트리들 중에서 논리적 관계들을 도시한다. 벤 다이어그램(1120) 내에서, 타원(1123)은 타원(1122, 1121)들을 포함하며, 타원(1122)은 타원(1121)을 포함한다. 이는 3개의 레코드들에서의 규칙들 사이에서의 논리적 관계들에 대응하는데, 여기서 P1 규칙인 "회사 X로부터의 유효한 증명서를 갖는 모든 워드 프로세싱 앱들"의 적용 가능성은 P5 규칙인 "회사 X로부터의 앱들"의 적용 가능성의 서브세트이며, P5 규칙의 적용 가능성은 P8 규칙인 "모든 앱들"의 적용 가능성의 서브세트이다.
- [0093] 3D 다이어그램은, 권한 테이블에서의 우선순위 필드의 사용이 보안 정책(1031) 내의 규칙들 중 어떤 것이 보안 평가를 하기 위해 적용되어야 하는지를 결정하는 방법을 도시한다. 도시된 바와 같이, 타원(1121)은 가장 높으며 보안 정책(1031)에서 최고 우선순위 규칙(P1)에 대응한다. 타원(1122)은 두 번째로 가장 높으며, 보안 정책(1031)에서 다음으로 가장 높은 우선순위 규칙(P5)에 대응한다. 타원(1123)은 가장 낮으며 보안 정책(1031)에서 가장 낮은 우선순위 규칙(P8)에 대응한다. 보다 높은 우선순위 규칙은, 교차할 때마다 보다 높은 우선순위 타원들이 보다 낮은 우선순위 타원들을 가리는 것처럼, 이들이 교차하는 경우에 보다 낮은 우선순위 규칙들을 대체한다. 보다 높은 우선순위 규칙들의 적용 가능성은 보다 낮은 우선순위 규칙들의 적용 가능성에서의 결점을 효과적으로 지적한다.
- [0094] 보안 정책(1031)은 따라서 보안 평가들을 수행할 때 논리적 불일치 없이 권한 테이블(1000)에서 다수의 엔트리들을 사용하여 구성될 수 있다. 우선순위 필드의 사용은 권한 테이블에서의 상이한 엔트리들이 동일한 요청에 적용될 때 어느 규칙들이 사용될지의 문제를 해결한다.
- [0095] 도 12는 다른 예시적인 보안 정책(1231)에 대한 벤 다이어그램들의 세트를 도시한다. 다이어그램들은 3개의 권한 테이블 엔트리(P1, P5, P8)들 중에서 논리적 관계들을 도시한다. 다이어그램들은 또한 우선순위들의 사용에 의해 예시적인 보안 정책(1231)의 구성을 도시한다. 보안 정책(1031)에서의 권한 테이블 엔트리들과 달리, 보안 정책(1231)에서의 권한 테이블 엔트리들은 반드시 서로의 서브세트들인 것은 아니다.
- [0096] 도 12는 보안 정책(1231), 2차원 벤 다이어그램(1220), 및 3차원 다이어그램(1230)을 포함한다. 보안 정책(1231)은 우선순위(P1)에 대한 레코드(또는 명령어들의 세트), 우선순위(P5)에 대한 레코드, 및 우선순위(P8)에 대한 레코드를 포함한다. 우선순위(P1)를 갖는 레코드에서의 규칙들은 "유효한 증명서를 갖는 모든 워드 프로세싱 앱들"을 허용할 것이다. 우선순위(P5)를 갖는 레코드에서의 규칙들은 "회사 X로부터 아무것도 없음"을 허용할 것이다. 우선순위(P8)를 갖는 레코드에서의 규칙들은 유효한 증명서를 갖는 모든 애플리케이션들의 열기를 허용할 것이다.
- [0097] 벤 다이어그램(1220)은 3개의 타원(1221 내지 1223)들을 포함한다. 3D 다이어그램(1230)은 동일한 3개의 타원들을 3차원 방식으로 도시한다. 타원(1221)은 우선순위(P1)를 갖는 규칙들을 나타내고, 타원(1222)은 우선순위(P5)를 갖는 규칙들을 나타내며, 타원(1223)은 우선순위(P8)를 갖는 규칙들을 나타낸다.
- [0098] 2D 벤 다이어그램은 따라서 3개의 권한 테이블 엔트리들 중에서 논리적 관계들을 도시한다. 벤 다이어그램(1220) 내에서, 타원(1223)은 타원(1221)을 포함하지만, 타원(1222)은 타원(1221, 1223)들 둘 모두와 중첩한다. 이는 3개의 레코드들 내의 규칙들 사이에서의 논리적 관계들에 대응한다. 구체적으로, P1 규칙인 "유효한 증명서를 갖는 모든 워드 프로세싱 앱들"의 적용 가능성은 P8 규칙인 "유효한 증명서를 갖는 모든 앱들"의 적용 가능성의 서브세트이다. P5 규칙인 "회사 X로부터 아무것도 없음"의 적용 가능성은 서브세트 또는 수퍼세트 없이 P1 및 P8 레코드들 둘 모두와 교차한다.
- [0099] 3D 다이어그램(1230)은, 권한 테이블에서의 우선순위 필드의 사용이 보안 정책(1231) 내의 규칙들 중 어떤 것이 보안 평가를 하기 위해 적용되어야 하는지를 결정하는 방법을 도시한다. 도시된 바와 같이, 타원(1221)은 가장 높으며 보안 정책(1231)에서 최고 우선순위 규칙(P1)에 대응한다. 타원(1222)은 두 번째로 가장 높으며, 보안 정책(1231)에서 다음으로 가장 높은 우선순위 규칙(P5)에 대응한다. 타원(1223)은 가장 낮으며 보안 정책(1231)에서 가장 낮은 우선순위 규칙(P8)에 대응한다. 보다 높은 우선순위 규칙은, 교차할 때마다 보다 높은 우선순위 타원이 보다 낮은 우선순위 타원들을 가리는 것처럼, 이들이 교차할 때 보다 낮은 우선순위 규칙들을 대체한다.
- [0100] 보안 정책(1231)은 따라서 보안 평가를 수행할 때 논리적 불일치 없이 권한 테이블에서 다수의 엔트리들을 사용

하여 구성될 수 있다. 우선순위 필드의 사용은 권한 테이블에서의 상이한 엔트리들이 동일한 요청에 적용될 때 어느 규칙들이 사용될지의 문제를 해결한다. 예를 들어, 회사 X로부터 유효한 증명서를 갖는 워드 프로세서에 대한 요청은, "유효한 증명서를 갖는 모든 워드 프로세싱 앱들"에 대한 규칙이 "회사 X로부터 아무것도 없음"보다 더 높은 우선순위를 갖기 때문에, 워드 프로세서가 승인되게 할 것이다. 다른 예로서, 회사 X로부터 유효한 증명서를 갖는 웹 브라우저 애플리케이션에 대한 요청은, "회사 X로부터 아무것도 없음"에 대한 규칙이 "유효한 증명서를 갖는 모든 앱들"보다 높은 우선순위를 갖기 때문에, 승인되지 않을 것이다.

- [0101] 몇몇 실시예들에 대해, 도 13은 권한 테이블을 사용함으로써 보안 평가를 수행하기 위한 프로세스(1300)를 개념적으로 도시한다. 프로세스는 보안 평가 요청을 수신하며 권한 테이블의 엔트리들(또는 레코드들)을 상기 요청과 매칭시킨다. 프로세스는 이어서 요청의 데이터 오브젝트가 권한 테이블에 저장된 코드 요건을 준수하는지 여부를 검증함으로써 보안 평가를 수행한다.
- [0102] 프로세스(1300)는 보안 평가를 하라는 요청 또는 질의를 수신한 때 시작한다(1310에서). 요청은 권한 테이블에 저장된 코드 요건이 적용될 데이터 오브젝트(예컨대, 애플리케이션 프로그램 코드, 설치 패키지, 또는 열릴 문서)가 동반된다.
- [0103] 프로세스는 다음으로 데이터 오브젝트로부터 코드 서명을 추출한다(1320에서). 몇몇 실시예들에서, 서명은 데이터 오브젝트의 소스에 의해 발행된 공개 키 증명서로부터 추출된다. 프로세스는 이어서 추출된 서명이 인증되는지 여부를 결정한다(1325에서). 몇몇 실시예들에서, 보안 평가기 내의 별개의 모듈이 코드 서명 인증(예컨대, 코드 서명 검증기 모듈)을 수행한다. 일단 코드 서명이 추출되면, 프로세스는 코드 서명이 성공적으로 인증되었는지 여부를 검사하기 위해 코드 서명 검증기 모듈을 사용한다. 서명이 성공적으로 인증된다면, 프로세스는 1330으로 진행한다. 서명이 성공적으로 인증되지 않는다면, 프로세스(1300)는 종료한다.
- [0104] 프로세스는 다음으로 권한 테이블에서 최고 우선순위 엔트리를 찾는다(1330에서). 몇몇 실시예들에서, 프로세스(1300)는 어떤 엔트리가 최고 우선순위를 갖는지를 결정하기 위해 권한 테이블에서 각각의 엔트리의 우선순위 필드들을 비교한다. 프로세스는 이어서 엔트리가 발견되었는지 여부를 결정한다(1340에서). 몇몇 실시예들에서, 이는 어떤 엔트리들이 이미 조사되었는지를 기록함으로써 달성된다. 권한 테이블에서의 엔트리들 모두가 이미 조사되었다면, 더 이상 발견될 엔트리는 없다. 최고 우선순위 엔트리가 발견되었다면, 프로세스는 1350으로 진행한다. 권한 테이블에서 더 이상 조사될 엔트리가 없다면, 프로세스(1300)는 종료한다.
- [0105] 1350에서, 프로세스는 발견된 권한 테이블 엔트리로부터 코드 요건을 검색한다. 프로세스는 이어서 데이터 오브젝트의 소스를 검증하기 위해 검색된 코드 요건을 사용한다(1360에서). 그러한 검증은 서명을 생성하기 위해 사용되는 증명서의 만료일 및 벤더 ID의 검증을 포함할 수 있다. 검색된 코드 요건은 또한 데이터 오브젝트의 다른 요구된 속성들, 데이터 오브젝트의 유형, 데이터 오브젝트의 명칭과 같은 속성들, 또는 데이터 오브젝트와 연관될 수 있는 임의의 다른 속성들을 검증하기 위해 사용된다. 따라서, 예를 들어, 프로세스가 데이터 오브젝트로부터의 서명을 인증할 수 있을지라도, 프로세스는 데이터 오브젝트가 코드 요건을 만족하지 않는다면(예컨대, 데이터 오브젝트가 정확한 유형을 소유하지 않거나 신뢰할 수 있는 것으로 간주되는 소스로부터 오지 않았다면) 데이터 오브젝트를 여전히 거절할 수 있다.
- [0106] 프로세스는 이어서 코드 요건이 질의와 매칭되는지 또는 질의에 적용 가능한지 여부를 결정한다(1370에서). 보안 평가가 요청에 적용 가능하지 않은 코드 요건에 기초하여 이루어질 수는 없다. 몇몇 실시예들은 이러한 프로세스의 이 시점까지 코드 요건의 적용 가능성을 검사하지 않는데, 그 이유는 프로세스가 요청의 데이터 오브젝트를 프로세싱하기 위해 코드 요건을 사용할 때까지 코드 요건의 적용 가능성이 때때로 결정될 수 없기 때문이다. 몇몇 실시예들에서, 프로세스는 충분한 정보가 이용 가능하자마자 코드 요건의 적용 가능성을 검사한다. 코드 요건이 질의와 매칭된다면(즉, 적용 가능하다면), 프로세스는 1390으로 진행한다. 그렇지 않다면, 프로세스는 1380으로 진행한다.
- [0107] 1380에서, 프로세스는 다음으로 가장 높은 우선순위 엔트리에 대한 권한 테이블을 탐색한다. 몇몇 실시예들은 이전에 조사된 보다 높은 우선순위 엔트리들을 마킹하며, 다음의 가장 높은 우선순위 엔트리는 간단하게는 마킹되지 않은 최고 우선순위 엔트리이다. 프로세스는 다음으로 엔트리가 발견되었는지 여부를 결정한다(1325에서). 그렇다면, 프로세스는 1330으로 귀환한다. 그렇지 않다면, 프로세스(1300)는 종료한다.
- [0108] 1390에서, 프로세스는 동작 개시기에 의해 취해져야 하는 동작을 결정하기 위해 코드 요건을 사용한다. 몇몇 실시예들에서, 이러한 동작은, 데이터 오브젝트가 코드 요건을 만족하는지 여부에 의존하여, 매칭된 엔트리의 동작 필드(즉, 승인함 또는 승인하지 않음)에 의해 규정된다. 요청에 응답한 후, 프로세스(1300)는 종료한다.

[0109] B. 캐시 테이블

[0110] 몇몇 실시예들에서 권한 테이블 내의 엔트리들을 사용하여 보안 평가를 수행하는 것은 많은 계산들을 포함할 수 있다. 증명서에 내장된 정보를 검증하는 규칙은 몇몇 실시예들에서 계산 집중적 동작들을 포함할 수 있다. 더욱이, 권한 테이블 내의 많은 엔트리들에 걸친 탐색은 요청과 매칭되는 엔트리에 대한 탐색을 완료하는 데 필요할 수 있다. 예를 들어, 요청이 매칭 엔트리들을 갖지 않는다는 것을 보장하기 위해, 권한 테이블의 완전한 탐색이 필요할 수 있다. 몇몇 실시예들은 아직 평가되지 않은 데이터 오브젝트들만에 대한 보안 평가 동작들을 수행함으로써 보안 평가 시간을 절약한다. 다시 말하면, 질의는 애플리케이션 프로그램이 처음으로 시작될 때만 또는 데이터 파일이 처음으로 열릴 때만 권한 테이블에 대해 이루어진다.

[0111] 보안 평가 동작들을 추가로 가속화하기 위해, 몇몇 실시예들의 규칙 데이터베이스는 권한 테이블에 더하여 캐시 테이블을 통합한다. 몇몇 실시예들에서, 캐시 테이블은 그 데이터 오브젝트에 고유한 해시 값(hash value)으로 인덱싱된 어드레스 위치에 데이터 오브젝트에 대한 요구된 동작을 저장한다. 예를 들어, 고유 코드 디렉토리 해시 값이 애플리케이션 프로그램 또는 데이터 파일에 대해 계산될 수 있다. 따라서, 애플리케이션의 실행(또는 설치) 또는 아카이브(도 24 및 도 25를 참조하여 섹션 IV에서 이하에 보다 상세히 설명됨)의 열기를 위한 보안 평가에 대한 요청은 실행 가능한 애플리케이션, 설치 파일 또는 아카이브의 해시 값의 계산만을 요구한다. 동작(예컨대, 승인 또는 미승인)은 이어서 해시 인덱스 값에 의해 지시된 캐시 테이블의 어드레스 위치로부터 검색된다.

[0112] 권한 테이블의 질의와는 달리, 캐시 테이블의 질의는 규칙들에 따라 다수의 엔트리들의 탐색도 너무 긴 계산도 요구하지 않는다. 캐시 테이블에서의 대응하는 엔트리를 갖는 데이터 오브젝트를 수반한 동작의 평가에 대한 요청은 따라서 크게 가속화될 수 있다. 몇몇 실시예들에서, 보안 평가에 대한 요청은 먼저 권한 테이블을 탐색하기 전에 캐시 테이블에서의 매칭 엔트리를 찾으려고 시도한다. 캐시 테이블은 또한, 캐시 테이블에서의 각각의 엔트리가 하나의 데이터 오브젝트(예컨대, 실행 가능한 애플리케이션 또는 문서)에 대응하기 때문에, 오브젝트 테이블로서 불린다. 사용자들은 그들이 최근에 상호 작용한 데이터 오브젝트(및 그에 따라 캐시 테이블에서의 엔트리들)와 상호작용하려는 경향이 있기 때문에, 캐시 테이블의 사용은 보안 평가 시간을 상당히 감소시킨다.

[0113] 몇몇 실시예들에 대해, 도 14는 권한 테이블(1410) 및 캐시 테이블(1460)을 포함하는 규칙 데이터베이스(1400)를 도시한다. 도면은 또한 캐시 테이블의 예시적인 엔트리들, 및 캐시 테이블 엔트리들과 권한 테이블 엔트리들 사이에서의 관계들을 도시한다. 몇몇 실시예들에서, 권한 테이블 및 캐시 테이블은 별개의 물리적 저장 장치들에 저장되는 반면, 다른 실시예들에서, 권한 테이블 및 캐시 테이블은 단일 물리적 저장 장치에 저장된다.

[0114] 권한 테이블(1410)은 도 9의 권한 테이블(900)과 동일한 포맷 및 콘텐츠를 가져, 레코드 엔트리(1421 내지 1423)들, 우선순위 필드(1440), 규칙 필드(1441), 및 동작 필드(1442)를 포함한다.

[0115] 캐시 테이블(1460)은 3개의 엔트리(1471 내지 1473)들을 포함한다. 권한 테이블의 엔트리들과는 달리, 캐시 테이블의 엔트리들은 규칙 필드를 포함하지 않는다. 그러나, 캐시 테이블은 해시 ID 필드(1491), 만료일 필드(1492), 동작 필드(1493), 및 기준 필드(1494)를 포함한 다른 필드들을 포함한다.

[0116] 캐시 테이블(1460)의 엔트리(1471)는 권한 테이블(1410)의 엔트리(1421)에 매칭되는 데이터 오브젝트(데이터 오브젝트 A)에 기초한다. 캐시 테이블(1460)의 엔트리(1472)는 권한 테이블(1410)의 엔트리(1422)에 매칭되는 다른 데이터 오브젝트(데이터 오브젝트 B)에 기초한다. 몇몇 실시예들에서, 이들 캐시 테이블 엔트리들은 보안 평가 요청들이 데이터 오브젝트(A, B)들에 기초하여 이루어질 때 만들어진다. 엔트리(1473)는 권한 테이블(1410)에서의 임의의 엔트리와 매칭되지 않는 데이터 오브젝트(데이터 오브젝트 C)에 기초한다. 부정 엔트리(1473)는 권한 테이블(1410)이 데이터 오브젝트 C를 수반하는 보안 평가 요청에 대한 매칭 규칙을 산출하는 데 실패할 때 캐시 테이블(1460)에서 만들어진다. 따라서, 캐시 테이블에서의 이러한 부정 엔트리의 존재는 권한 테이블(1410)의 길고 철저한 탐색을 방지한다.

[0117] 캐시 테이블의 부정 엔트리(1473)는 권한 테이블(1410)에서의 매칭 엔트리 없이 오브젝트로부터 생성되기 때문에 권한 테이블로부터 임의의 필드들을 물려받지 않는다. 몇몇 실시예들에서, 부정 엔트리는 부정 엔트리이고 데이터 오브젝트에 대한 권한 테이블(1410)에서의 매칭 엔트리가 없다는 표시를 포함한다. 몇몇 그러한 실시예들에서, 권한 테이블은 정상적으로 프로세싱되지 않지만 다시 참조될 캐시 테이블에서의 부정 엔트리들에 대한 앵커들로서 작용하는 "가상" 엔트리들을 포함한다. 다시 말하면, 어떠한 규칙들도 적용하지 않음을 결정하는 결과로서 생성되는 캐시 레코드들(즉, 부정 엔트리들)은, 권한 테이블에 저장된 가상의 "권한 없음" 규칙(즉,

가상 엔트리들)을 참조할 것이다. 이는 데이터 구조들에서의 일관성(consistency)을 보장하기 위해 주로 행해지며 이들 실시예들의 가시적인 거동에 영향을 미치지 않는다.

[0118] 캐시 테이블 엔트리에서의 동작 필드(1493)는 데이터 오브젝트에 대해 실제로 취해졌던 동작을 기록한다. 이러한 동작은 데이터 오브젝트를 승인하거나 또는 승인하지 않는, 권한 테이블(1410)에 저장된 매칭 규칙들의 적용에 기초한다. 예를 들어, 캐시 테이블 엔트리(1472)의 동작 필드는 오브젝트 B에 대해 "승인하지 않음"을 표시한다. 이는 (권한 테이블 엔트리(1422)에 저장된) 오브젝트 B에 대한 매칭 규칙이 그의 코드 요건을 충족시키는 데 실패한 것으로서 오브젝트 B를 평가하며 오브젝트 B가 승인되지 않음을 표시한다. 오브젝트 B에 기초한 다음 요청에 의한 캐시 테이블에서의 이러한 엔트리의 검색은 이어서 또한 승인되지 않을 것이다. 다시 말하면, 각각의 캐시 테이블 엔트리의 동작 필드(1493)는 상기 엔트리의 데이터 오브젝트의 보안 평가를 저장한다.

[0119] 만료일 필드(1492)는 캐시 엔트리가 만료한 때를 표시한다. 몇몇 실시예들에서, 데이터 오브젝트에 대한 캐시 테이블 엔트리의 만료일은 데이터 오브젝트의 콘텐츠에 의해 결정된다. 몇몇 실시예들은 캐시 엔트리에서의 만료일로서 데이터 오브젝트의 증명서의 만료일을 사용한다. 그 서명이 일련의 증명서들로부터 도출되는 데이터 오브젝트들에 대해, 가장 빨리 만료하는 증명서의 만료일이 이들 실시예들의 몇몇에서 사용될 것이다. 몇몇 실시예들에서, 만료일은 권한 테이블에서의 규칙들에 의해 결정되며 캐시 테이블 엔트리가 생성될 때 캐시 테이블로 삽입된다. 몇몇 실시예들에서, 운영 체제는 데이터 오브젝트들의 증명서들이 아닌 정보에 기초하여 스스로 만료일을 규정한다. 몇몇 실시예들에서의 그러한 정보는 벤더들에 의해 공급된 디폴트 값, 요청된 동작의 유형, 또는 운영 체제에 이용 가능한 임의의 다른 정보일 수 있다.

[0120] 몇몇 실시예들에서, 보안 평가는 엔트리를 적용하기 전에 캐시 테이블 엔트리의 만료일 필드를 검사한다. 만료일에 따라 만료된 캐시 테이블 엔트리는 보안 평가에 의해 무시될 것이다(그리고 캐시-미스(cache-miss)로서 처리될 것이다). 이는 보안 평가가 캐시 테이블(1460)에서의 더 이상 쓸모가 없는 엔트리들에 기초하여 부정확한 보안 평가를 하는 것을 방지한다. 몇몇 실시예들에서, 만료된 캐시 테이블 엔트리는 액세스될 때 폐기된다. 몇몇 실시예들은 주기적으로 만료된 엔트리들에 대한 캐시 테이블을 검사하며 테이블로부터 이들을 제거하는 폐기 메커니즘을 포함한다.

[0121] 해시 ID 필드(1491)는 해시 값을 저장한다. 몇몇 실시예들에서, 캐시 테이블 엔트리의 해시 ID 필드는 캐시 테이블 엔트리의 데이터 오브젝트의 해시와 연관되는 값을 저장한다. 몇몇 다른 실시예들은 캐시 테이블에 이러한 필드를 포함하지 않는다.

[0122] 몇몇 실시예들에서, 기준 필드(1494)는 권한 테이블에서의 엔트리로의 링크 백(link back)을 저장한다. 구체적으로, 각각의 캐시 테이블 엔트리의 기준 필드(1494)는 캐시 테이블 엔트리를 생성하기 위해 사용되는 권한 테이블 엔트리로의 링크 백을 저장한다(즉, 권한 테이블 엔트리의 규칙들은 캐시 테이블 엔트리와 연관된 데이터 오브젝트의 보안 평가를 수행하기 위해 사용된다). 예를 들어, 캐시 테이블 엔트리(1471)의 기준 필드는 캐시 테이블 엔트리(1471)가 권한 테이블 엔트리(1421)로부터 생성된다는 것을 표시할 것이며, 캐시 테이블 엔트리(1472)의 상기 기준 필드는 캐시 테이블 엔트리(1472)가 권한 테이블 엔트리(1422)로부터 생성된다는 것을 표시할 것이다.

[0123] 몇몇 실시예들은 권한 테이블에서의 변화들로 인해 더 이상 쓸모가 없는 엔트리들의 캐시 테이블을 폐기하기 위해 캐시 테이블에서의 기준 필드를 사용한다. 몇몇 실시예들에서, 캐시 테이블은 또한 권한 테이블(도시되지 않음)로부터 물려받는 우선순위 필드를 포함한다. 몇몇 실시예들은 권한 테이블에서의 변화들로 인해 더 이상 쓸모가 없는 엔트리들의 캐시 테이블을 폐기하기 위해 캐시 테이블에서의 우선순위 필드를 사용한다. 캐시 테이블 엔트리들의 폐기는 섹션 II-C에서 그리고 도 18을 참조하여 이하에 추가로 설명될 것이다.

[0124] 몇몇 실시예들에서, 캐시 테이블(1460)은 다른 필드들을 포함한다. 이들 캐시 테이블 필드들의 몇몇의 콘텐츠는 캐시 테이블 엔트리를 생성하기 위해 사용되는 권한 테이블 엔트리로부터 물려받는다. 몇몇 실시예들에서, 캐시 테이블에서의 엔트리들은, 그들의 어드레스들이 이들 엔트리들을 생성하기 위해 사용되는 데이터 오브젝트들의 해시 값들에 의해 결정되기 때문에, 반드시 인접하는 것은 아니다. 몇몇 실시예들에서, 데이터 오브젝트에 대한 해시 값은 데이터 오브젝트에 대한 해시 동작을 수행함으로써 생성된다. 몇몇 실시예들에서, 해싱 동작은 임의의 다른 데이터 오브젝트로부터 데이터 오브젝트를 고유하게 식별하는 해시 값을 생성한다. 몇몇 실시예들에서, 해싱 동작은 증분적 해싱 동작을 포함한다. 증분적 해싱 동작의 설명들을 본 명세서에 참고로 포함된, 예컨대 미국 특허 제7,103,779호에서 볼 수 있다.

- [0125] 몇몇 실시예들에서, 해싱 동작은 코드 디렉토리 해싱 동작을 포함한다. 코드 디렉토리 해시는 몇몇 실시예들에서 캐시 테이블에서의 매 고유 엔트리를 참조하는 고유 값이다. 따라서, 대응하는 코드 요건을 갖는 각각의 데이터 오브젝트는, 고유 코드 디렉토리 해시에 의해, 캐시 테이블에서 고유하게 식별된다. 코드 디렉토리 및 그의 함수들의 해시의 설명들을 본 명세서에 참고로 포함된 미국 특허 출원 공개 제2008/0168553호에서 볼 수 있다.
- [0126] 도 14의 예에서, 오브젝트 A(1471)에 대한 엔트리는 오브젝트 A의 해시 값에 의해 인덱싱되는 위치에 저장되고, 오브젝트 B(1472)에 대한 엔트리는 오브젝트 B의 해시 값에 의해 인덱싱되는 위치에 저장되며, 오브젝트 C(1473)에 대한 엔트리는 오브젝트 C의 해시 값에 의해 인덱싱되는 위치에 저장된다.
- [0127] 몇몇 실시예들에 대해, 도 15는 동작 개시기로부터의 요청에 기초하여 보안 평가를 하기 위해 권한 테이블 및 캐시 테이블 둘 모두를 사용하는 보안 평가기(1500)를 도시한다. 도시된 바와 같이, 보안 평가기(1500)는 동작 개시기(1505)로부터 요청을 수신하며 권한 테이블(1510) 및 캐시 테이블(1515) 둘 모두에 질의한다. 보안 평가기(1500)는 권한 테이블에 질의하기 위한 질의 관리기(1520) 및 캐시 테이블에 질의하기 위한 질의 관리기(1525)를 포함한다. 보안 평가기(1500)는 또한 프로세서(1530) 및 캐시 테이블 레코드 생성기(1540)를 포함한다. 권한 테이블(1510)에 질의하기 위한 질의 관리기(1520)는 또한 규칙 엔진(1550), 레코드 선택기(1560), 및 서명 검증기(1570)를 포함한다.
- [0128] 동작 개시기(1505)는 보안 평가 요청들을 하며 보안 평가기의 프로세서(1530)로부터 오는 보안 평가들의 결과들을 기다린다. 프로세서(1530)는 동작 개시기(1505)로부터 보안 평가 요청들을 수신한다. 이러한 수신된 보안 평가 요청에 기초하여, 프로세서는 보안 평가 요청을 승인할지 여부를 결정하기 위해 질의 관리기(1520)(권한 테이블용) 및 질의 관리기(1525)(캐시 테이블용) 둘 모두와 통신한다. 프로세서는 또한 캐시 테이블(1515)에 캐시-미스가 있을 때 새로운 캐시 엔트리들을 생성하기 위해 캐시 테이블 레코드 생성기(1540)를 사용한다. 캐시 테이블 레코드 생성기는 또한 요청의 대상인 데이터 오브젝트들(즉, 프로그램 코드들 또는 문서들)을 해싱함으로써 캐시 테이블(1515)의 물리적 메모리에 대한 어드레스를 생성하기 위해 사용된다.
- [0129] 캐시 테이블에 질의하기 위한 질의 관리기(1525)는 캐시 테이블(1515)을 포함하는 물리적 메모리와 프로세서(1530) 사이에 인터페이스를 제공한다. 권한 테이블에 질의하기 위한 질의 관리기(1520)는 권한 테이블(1510)을 포함하는 물리적 메모리와 프로세서(1530) 사이에 인터페이스를 제공한다. 질의 관리기(1520)는 또한 권한 테이블(1510)로부터 엔트리를 선택 및 검색하기 위해 레코드 선택기(1560)를 사용한다. 검색된 엔트리는 이어서 파싱되고(parsed) 실행되도록 규칙 엔진(1550)에 중계된다. 몇몇 실시예들에서, 질의 관리기(1520)는 데이터 오브젝트의 코드 서명을 검증하기 위해 서명 검증기(1570)를 사용한다. 그러한 몇몇 실시예들에서, 코드 서명이 임의의 이유로 적절히 인증하는 데 실패한다면, 보안 평가기(1500)는 동작 개시기(1505)에 미승인 메시지를 반환한다. 몇몇 실시예들에서, 규칙 엔진(1550), 레코드 선택기(1560), 및 서명 검증기(1570)의 기능들은 질의 관리기(1520) 대신에 프로세서(1530)에 의해 수행된다.
- [0130] 서명 검증기 모듈(1570)은 데이터 오브젝트의 코드 서명을 인증한다. 몇몇 실시예들에서, 서명 검증기 모듈은 데이터 오브젝트 자체의 콘텐츠뿐만 아니라 공개 키 증명서에서의 정보를 사용함으로써 서명을 인증한다. 서명 검증기 모듈은 증명서에 의해 식별되는 서명 인증 알고리즘에 따라 서명 인증 동작을 수행한다. 서명 인증 프로세스의 완료 시, 서명 검증기 모듈은 데이터 오브젝트의 서명이 성공적으로 인증되었는지 여부를 보안 평가기(1500)에 알린다.
- [0131] 도 16은 보안 평가기(1600)에 의해 데이터 오브젝트를 위한 캐시 테이블 엔트리를 생성하고 저장하는 것을 도시한다. 구체적으로, 도면은 요청이 데이터 오브젝트에 기초하여 보안 평가기에 대해 이루어질 때 캐시 테이블 엔트리에서의 다양한 필드들의 생성을 도시한다. 도 16은 권한 테이블(1605), 캐시 테이블(1610), 및 데이터 오브젝트(1625)를 위해 보안 평가기(1600)에 요청하는 동작 개시기(1620)를 도시한다. 캐시 테이블은 엔트리(1611 내지 1613)들을 포함한다.
- [0132] 보안 평가기(1600)는 권한 테이블을 위한 질의 관리기(1630), 해시 모듈(1650), 메모리 인터페이스(1660), 및 만료일 추출 모듈(1670)을 포함한다. 몇몇 실시예들에 대해, 질의 관리기(1630)는 도 15에서의 질의 관리기(1520)와 유사한 반면, 해시 모듈(1650), 및 만료일 추출 모듈(1670)의 기능들은 도 15의 프로세서(1530)와 유사한 프로세서에 의해 수행된다. 보안 평가기(1600)는 또한 도면의 간소화를 위해 도시되지 않는 서명 검증기 모듈과 같은 다른 모듈들을 포함한다.
- [0133] 동작 개시기(1620)는 데이터 오브젝트(1625)('데이터 오브젝트 X')의 사용을 요구하는 동작에 관한 보안 평가를

위해 보안 평가기(1600)에 요청을 한다. 데이터 오브젝트(1625)는 질의 관리기(1630), 해시 모듈(1650), 및 만료일 추출 모듈(1670)을 포함한, 보안 평가기(1600) 내의 수 개의 모듈들로 중계된다. 질의 관리기(1630)는 데이터 오브젝트(1625)에 매칭되는 엔트리를 권한 테이블(1605)로부터 찾는다. 질의 관리기는 이어서 데이터 오브젝트(1625)에 대한 적절한 동작을 결정하기 위해 매칭 엔트리의 규칙들을 적용한다. 질의 관리기에 의해 취해진 이러한 동작은 새로운 캐시 테이블 엔트리(1612)에 대한 동작 필드(1642)를 형성한다.

[0134] 해시 모듈(1650)은 데이터 오브젝트(1625)를 수신하며 수신된 데이터 오브젝트에 대한 해시 동작을 수행한다. 몇몇 실시예들에서, 해시 동작은 증분적 해싱 동작이다. 해시 모듈(1650)은 데이터 오브젝트 X에 대한 새롭게 생성된 캐시 테이블 엔트리를 위한 캐시 테이블 내의 위치를 규정하기 위한 인덱스('X를 위한 인덱스')로서 작용하는 고유 코드 디렉토리 해시 값(1655)을 생성한다. 메모리 인터페이스(1660)는 이어서 캐시 테이블에 대한 물리적 어드레스로서 이 값을 사용한다. 몇몇 실시예들에서, 이러한 고유하게 생성된 코드 디렉토리 해시 값은 데이터 오브젝트 X에 대한 엔트리(1612)의 해시 ID 필드에 저장될 것이다.

[0135] 또한 데이터 오브젝트(1625)를 수신하는 만료일 추출 모듈(1670)은 데이터 오브젝트(1625)로부터 만료일을 계산 또는 추출한다. 몇몇 실시예들에서, 데이터 오브젝트의 만료일은 데이터 오브젝트의 서명을 생성하기 위해 사용되는 증명서에 의해 규정된다. 데이터 오브젝트는 하나 이상의 증명서 또는 심지어 일련의 증명서들로부터 도출된 서명을 가질 수 있다. 그러한 데이터 오브젝트에 대해, 몇몇 실시예들은 캐시 테이블 엔트리에 대한 만료로서 이들 증명서들 중에서 가장 빠른 만료일을 사용한다. 추출된 만료일은 만료일 필드(1675)가 된다. 몇몇 실시예들에서, 만료일 추출 모듈(1670)은 만료일들을 결정하기 위한 기초로서 다른 정보를 사용한다. 몇몇 다른 실시예들에서, 만료일 필드(1675)는 권한 테이블로부터 물려받으며 만료일 추출 모듈(1670)에 의해 계산되지 않는다.

[0136] 동작 필드(1642) 및 만료일 필드(1675)는, 데이터 오브젝트 X에 대한 해시 ID 필드 및 기준 필드와 함께, 데이터 오브젝트 X에 대한 새로운 캐시 테이블 엔트리(1612)를 형성하기 위해 함께 연결된다. 캐시 테이블(1610)은 2개의 다른 오브젝트들에 대한 2개의 다른 엔트리들(데이터 오브젝트 Y에 대한 엔트리(1611) 및 오브젝트 Z에 대한 엔트리(1613))을 갖는다.

[0137] 몇몇 실시예들에서, 캐시 테이블의 콘텐츠들은 유사한 운영 체제들을 실행하는 다양한 컴퓨터들에 걸쳐 공유될 수 있다. 컴퓨터는 다른 컴퓨터에 의해 생성되는 캐시 테이블, 또는 운영 체제의 벤더에 의해 공급되는 사전-패키징된 캐시 테이블을 들여올 수 있다. 그러한 사전에 채워진 캐시 테이블을 사용하는 컴퓨터는 그 자신의 캐시 엔트리들을 생성할 필요 없이 그의 보안 평가 동작들을 즉시 가속화할 수 있다.

[0138] 몇몇 실시예들에 대해, 도 17은 보안 평가를 수행하기 위해 권한 테이블 및 캐시 테이블 둘 모두를 사용하는 프로세스(1700)를 개념적으로 도시한다. 프로세스는 먼저 캐시 테이블에서 매칭 엔트리를 찾으려고 시도한다. 프로세스가 캐시 테이블에서 그러한 엔트리를 찾을 수 없다면(즉, 캐시-미스), 프로세스는 대신에 매칭 엔트리를 찾기 위해 권한 테이블에 질의할 것이다. 권한 테이블의 질의는 캐시 테이블에서의 새로운 엔트리를 초래한다. 이러한 프로세스는 몇몇 실시예들에서 운영 체제의 보안 평가기에 의해 수행된다.

[0139] 프로세스(1700)는 동작(예컨대, 애플리케이션의 실행, 애플리케이션의 설치, 또는 문서의 열기)에 대한 요청이 동작 개시기에 의해 이루어질 때 시작한다. 프로세스는 요청된 동작에 의해 요구되는 데이터 오브젝트와 함께 이러한 요청을 수신한다(1710에서).

[0140] 프로세스는 다음으로 데이터 오브젝트에 대한 해시 값(즉, 고유 코드 디렉토리 해시 값)을 결정한다(1720에서). 해시 값은 몇몇 실시예들에서 증분적 해싱 동작에 의해 계산된다. 계산된 해시 값들은 데이터 오브젝트에 대한 캐시 테이블에 엔트리를 위치시키기 위한 인덱스로서 동작한다. 다음으로, 프로세스는 해시 값을 사용함으로써 캐시 테이블 엔트리를 찾을 수 있는지 여부를 결정한다(1730에서). 이러한 결정은, 몇몇 실시예들에서, 해시 값에 의해 지시된 위치에 저장된 유효 엔트리가 있는지 여부에 기초한다. 이들 실시예들 중 몇몇에서, 각각의 캐시 테이블 엔트리는 캐시 테이블 엔트리가 유효한지 여부를 표시하는 비트와 연관된다. 캐시 테이블 엔트리는, 데이터 오브젝트에 대한 캐시 엔트리에 의해 채워지지 않았다면 또는 캐시 테이블 엔트리가 폐기되어 아직 새로운 캐시 엔트리로 채워지지 않았다면, 유효하지 않다. 프로세스가 데이터 오브젝트의 해시 값을 사용하여 유효 엔트리를 찾을 수 있다면(캐시 히트(cache hit)), 프로세스는 1740으로 진행한다. 그렇지 않다면(캐시 미스), 프로세스는 1750으로 진행한다.

[0141] 1740에서, 프로세스는 매칭 캐시 테이블 엔트리의 콘텐츠에 기초하여 보안 평가 요청에 대한 응답을 결정한다. 프로세스는 이어서 응답이 요청된 동작의 승인인지 여부를 결정한다(1780에서). 응답이 동작을 승인한다면, 프

로세스는 1790으로 진행하며, 여기서 동작 개시기는 도 1의 실행 모듈(172), 설치 모듈(170), 또는 콘텐츠 열기 모듈(174)과 같은 동작 핸들러에 요청을 전달한다. 응답이 동작을 승인하지 않는다면, 동작 개시기는 동작 핸들러에 요청을 전달하지 않음으로써 동작이 진행되는 것을 정지한다. 캐시 테이블 엔트리의 콘텐츠들에 기초하여 동작 개시기에 응답한 후, 프로세스(1700)는 종료한다.

[0142] 1750에서, 프로세스는 캐시 테이블에서의 캐시-미스 때문에 권한 테이블에 질의한다. 몇몇 실시예들에서, 이러한 동작은 요청에 매칭되는 엔트리에 대해 권한 테이블을 탐색하는 도 13의 프로세스(1300)에 의해 수행된다.

[0143] 다음으로, 프로세스는 권한 테이블에서 매칭 엔트리를 발견할 수 있는지 여부를 결정한다(1755에서). 그렇다면, 프로세스는 1760으로 진행한다. 그렇지 않다면, 프로세스는 데이터 오브젝트에 대한 부정 캐시 엔트리를 생성하도록 1770으로 진행하고 이어서 종료한다. 위에서 논의된 바와 같이, 데이터 오브젝트에 대한 부정 캐시 엔트리는 보안 평가기가 운영 체제의 보안 정책들이 특정 요청에 적용 가능한 규칙들을 갖지 않는다는 것을 빨리 확인할 수 있게 한다.

[0144] 1760에서, 프로세스는 권한 테이블의 매칭 엔트리로부터의 응답을 결정한다. 프로세스는 다음으로 권한 테이블로부터의 이러한 매칭 엔트리에 기초하여 캐시 테이블에 새로운 엔트리를 생성한다(1765에서). 프로세스는 이어서 요청된 동작을 승인할지 또는 승인하지 않을지 여부를 결정하기 위해 1780으로 진행한다. 권한 테이블 엔트리의 콘텐츠들에 기초하여 동작 개시기에 응답한 후, 프로세스(1700)는 종료한다.

[0145] C. 사용자 오버라이드(user override)

[0146] 몇몇 실시예들에서, 충분한 특권들을 갖는 사용자들이 권한 테이블에서 엔트리들을 추가, 삭제, 또는 수정함으로써 규칙 데이터베이스에 대한 변경들을 만들 수 있게 된다. 예를 들어, 관리자 특권을 갖는 사용자는 특정 애플리케이션을 실행하기 위한 서명 인증을 바이패스할 수 있다. 사용자는 보다 낮은 우선순위 규칙들을 대체하는 권한 테이블 내로 보다 높은 우선순위 엔트리를 추가하고, 이에 의해 특정 애플리케이션이 실행되게 함으로써 그렇게 할 수 있다.

[0147] 그러나, 상기 도 10에 도시된 바와 같이, 보안 정책은 종종 우선순위들에 의해 함께 연결되는 다수의 테이블 엔트리들을 포함한다. 따라서, 보안 정책 하에서 데이터 오브젝트를 평가하는 것은 이들 테이블 엔트리들을 이들의 우선순위들에 의해 제시된 순서로 조사하는 것을 요구한다. 데이터 오브젝트에 대한 임의의 주어진 보안 평가에 대해, 보다 낮은 우선순위 규칙들은 보다 높은 우선순위 규칙들이 적용 가능하지 않을 때만 적용 가능해진다. 그러므로, 보다 높은 우선순위 규칙에 대한 임의의 변경은 잠재적으로 보다 낮은 우선순위 규칙들의 적용 가능성을 변경한다.

[0148] 캐시 테이블은 권한 테이블에서의 변경 이전에 규칙들의 적용 가능성을 반영하는 캐시 엔트리들을 저장하기 때문에, 몇몇 실시예들에서, 특정 우선순위의 권한 테이블에서의 변경은 캐시 테이블에서 동일하거나 보다 낮은 우선순위 엔트리들 모두의 폐지를 트리거한다. 예를 들어, 우선순위 값(2.3)을 갖는 새로운 권한 테이블 엔트리는 2.3 이하의 우선순위들 값들을 갖는 모든 캐시 테이블 엔트리들의 폐지를 트리거할 것이다. 몇몇 실시예들에서, 이러한 폐지 프로세스는 캐시 테이블 엔트리를 생성하기 위해 사용되는 권한 테이블 엔트리의 우선순위를 학습하기 위해 캐시 테이블 엔트리들 각각에서의 기준 필드(예컨대, 도 14의 1494)를 사용한다. 몇몇 다른 실시예들에서, 각각의 캐시 테이블 엔트리는 이러한 목적을 위해 권한 테이블로부터 물려받는 우선순위 필드를 지닌다.

[0149] 몇몇 실시예들에서, 부정 캐시 엔트리는 최저 우선순위를 갖는 것으로 가정된다. 부정 엔트리가 권한 테이블에서 매칭 엔트리를 갖지 않는 데이터 오브젝트의 캐시 테이블 엔트리이기 때문에, 권한 테이블에서의 임의의 변경은 부정 엔트리의 데이터 오브젝트에 영향을 미칠 수 있다(즉, 권한에서의 변경은 잠재적으로 데이터 오브젝트가 권한 테이블에서 매칭 엔트리를 갖게 하며, 따라서 부정 엔트리를 더 이상 쓸모가 없게 만든다). 이들 실시예들 중 몇몇에서, 권한 테이블에 대한 임의의 변경은 항상 부정 캐시 엔트리들이 폐지되게 한다.

[0150] 몇몇 실시예들에 대해, 도 18은 관리 사용자에 의해 규칙 데이터베이스에서의 변경을 따르는 규칙 테이블을 유지하는 프로세스를 개념적으로 도시한다. 프로세스는 캐시 테이블로부터 보다 낮은 우선순위 엔트리들을 폐지한다. 프로세스(1800)는 특권을 가진 사용자(예컨대, 관리자)가 규칙 데이터베이스를 변경시킨 때 시작한다. 프로세스는 특권을 가진 사용자로부터 규칙 변경을 수신하며(1810에서) 규칙 변경이 권한 테이블에 이미 있는 규칙에 대한 것인지 여부를 결정한다(1820에서). 규칙 변경이 권한 테이블 엔트리에 이미 있는 규칙을 위한 것이라면, 프로세스는 권한 테이블 엔트리를 업데이트하기 위해 1840으로 진행한다. 그렇지 않다면, 프로세스는 권한 테이블에 새로운 엔트리를 만들기 위해 1830으로 진행한다.

- [0151] 프로세스는 다음으로 새롭게 부가되거나 업데이트된 규칙보다 낮은 우선순위를 갖는 규칙 데이터베이스 내의 엔트리들이 있는지 여부를 결정한다(1850에서). 그렇다면, 프로세스는 1860으로 진행한다. 그렇지 않다면, 프로세스(1800)는 종료한다.
- [0152] 1860에서, 프로세스는 캐시 테이블에서 보다 낮은 우선순위 엔트리들 모두를 삭제한다. 몇몇 실시예들에서, 프로세스는 캐시 테이블에서의 모든 점유된 엔트리들을 조사하며 새롭게 부가되거나 업데이트된 권한 테이블 엔트리보다 낮거나 같은 우선순위들을 갖는 엔트리들을 퍼징한다. 보다 낮은 우선순위 캐시 테이블 엔트리들(임의의 부정 엔트리들을 포함함) 모두를 삭제한 후, 프로세스(1800)는 종료한다.
- [0153] III. 다운로드된 콘텐츠의 보안 평가
- [0154] 상기 설명된 실시예들 중 몇몇은 이들 데이터 오브젝트들에 대한 보안 평가를 하기 위해 데이터 오브젝트들에 내장되는 증명서들을 사용한다. 대부분의 경우에, 이들 방법들은 이들 데이터 오브젝트들의 소스에 의해 제공되는 식별자들(예컨대, 증명서들에서의 서명들 및 벤더 ID들)에 의존한다. 그러나, 이들 소스-제공된 식별자들이 항상 이용 가능하거나 신뢰 가능한 것은 아니다. 따라서, 보안 평가를 수행하기 위해 이들 소스-제공된 식별자들에 의존하는 것에 더하여 또는 그 대신에, 몇몇 실시예들의 운영 체제는 데이터 오브젝트에 대한 운영 체제 자신의 식별자를 제공한다. 이들 실시예들 중 몇몇에서, 운영 체제는 인터넷, USB 플래시 드라이브와 같은 외부 소스들로부터, 또는 디바이스의 임의의 다른 주변 인터페이스를 통해 들어오는 데이터 오브젝트와 태그를 연관시킨다. 태그는 데이터 오브젝트가 외부 소스로부터 온다는 것을 표시하기 위해 검역 비트(quarantine bit)를 포함한다. 검역 비트의 존재는 시스템이 데이터 오브젝트에 대한 보안 평가를 수행하지 않았음을 추가로 표시한다. 일단 운영 체제가 데이터 오브젝트에 대한 보안 평가를 수행하였다면, 태그는 운영 체제의 보안 정책 하에서 데이터 오브젝트가 이미 평가되었으며, 데이터 오브젝트가 운영 체제의 보안 정책을 통과하였음(또는 실패하였음)을 표시하기 위해 (예컨대, 검역 비트를 제거함으로써) 업데이트된다.
- [0155] 도 19는 보안 평가를 수행하기 위해 그 자신의 식별자를 사용하는 운영 체제를 갖는 컴퓨터(1900)를 도시한다. 구체적으로, 컴퓨터(1900)의 운영 체제는 예를 들어 인터넷으로부터 다운로드되는 콘텐츠 및/또는 애플리케이션들에 태그 또는 검역 비트를 부가한다. 컴퓨터(1900)는 저장 디바이스(1910), 태그 모듈(1920), 컴퓨터(1900) 상에서 현재 실행되는 애플리케이션(1930), 네트워크 인터페이스(1940), 및 사용자 인터페이스(UI) 모듈(1950)을 포함한다. UI 모듈(1950)은 입력 디바이스 드라이버들(1960) 및 디스플레이 모듈(1970)을 통해 사용자와의 통신을 제어한다.
- [0156] 네트워크 인터페이스(1940)는 네트워크를 통해 컴퓨터(1900)와 외부 세계 사이에서의 통신 인터페이스를 제공한다. 네트워크는 컴퓨터(1900)가 인터넷(1980)을 통해 통신하게 하고 다른 컴퓨터들로부터 애플리케이션 파일(1981, 1982)들 및 콘텐츠 파일(1983, 1984)들과 같은 데이터 오브젝트들에 액세스하게 한다. 이들 콘텐츠 파일들은 텍스트 파일들, 워드 프로세서 문서들, 스프레드시트들, 미디어 파일들 등일 수 있다. 애플리케이션 파일들은 설치 및 프로그램 파일들을 포함할 수 있다. 이들 애플리케이션 및 콘텐츠 파일들은 이어서 컴퓨터(1900)로 다운로드될 수 있다. 도면에 도시되지 않았지만, 컴퓨터(1900)는 또한 인터넷이 아닌 (외부 플래시 드라이브와 같은) 외부 소스들로부터 데이터 오브젝트들을 다운로드할 수 있다.
- [0157] 애플리케이션(1930)은 운영 체제를 통해 컴퓨터(1900) 상에서 현재 실행되고 있는 프로그램 또는 프로세스들의 세트이다. 애플리케이션(1930)은 인터넷 브라우저, 워드 프로세서, 비디오 게임, 미디어 편집 애플리케이션, 또는 컴퓨터(1900) 상에서 동작할 수 있는 임의의 프로그램일 수 있다. 애플리케이션(1930)은 네트워크 인터페이스를 경유하여 인터넷을 통해 소스들로부터 데이터 오브젝트(1981 내지 1984)들을 다운로드하는 것을 포함한, 인터넷을 통해 통신을 요구하는 동작들을 수행할 수 있다. 일단 파일들이 다운로드되었다면, 애플리케이션(1930)은 저장 디바이스(1910)에 다운로드된 파일들을 저장한다. 몇몇 실시예들에서, 다운로드된 파일들은 저장되기 전에 태그 모듈(1920)을 거쳐야 한다.
- [0158] 태그 모듈(1920)은 태그와 각각의 다운로드된 파일을 연관시킨다. 몇몇 실시예들에서, 그러한 태그는 다운로드된 파일이 외부 소스로부터 온 것이고 운영 체제의 보안 정책 하에서 조사되지 않았음을 표시하기 위해 검역 비트를 포함한다. 몇몇 실시예들에서, 태그 모듈은 태그를 다운로드된 파일에 삽입하거나 첨부한다. 몇몇 실시예들에서, 태그 모듈은 어떤 파일들이 인터넷 또는 다른 외부 소스들로부터 온 것인지를 기록하는 태그 테이블을 유지한다. 컴퓨터 상에서 실행되는 다른 프로세스들 또는 모듈들은 어떤 파일들 또는 데이터 오브젝트들이 인터넷과 같은 외부 소스들로부터 온 것인지를 찾아내기 위해 태그 테이블에 액세스할 수 있다. 몇몇 실시예들은 파일이 애플리케이션(1930)에 의해 다운로드되자마자 그리고 파일이 저장 디바이스(1910)에

저장되기 전에 외부 소스로부터의 파일과 태그를 연관시킨다.

- [0159] 저장 디바이스(1910)는 컴퓨터(1900)를 위한 다양한 데이터 오브젝트들을 저장한다. 이들 데이터 오브젝트들은 인터넷 또는 다른 외부 소스들로부터 다운로드되는 콘텐츠 파일들 및 애플리케이션 파일들을 포함할 수 있다. 몇몇 실시예들에서, 이들 다운로드된 애플리케이션들 및 콘텐츠 파일들은 태그 모듈(1920)에 의해 공급되는 태그들과 연관된다.
- [0160] 일단 다운로드된 파일들이 저장되고 태깅되면, 운영 체제는 데이터 오브젝트가 다운로드되었는지 여부를 고려하는 보안 정책들에 기초하여 보안 평가를 수행할 수 있다. 도 20은 보안 정책이 태그들 또는 검역 비트들의 존재를 검사하는 것을 요구하는 운영 체제(2000)를 도시한다. 구체적으로, 운영 체제(2000)는 보안 평가 동작들을 수행하기 전에 데이터 오브젝트가 태깅되었는지 여부를 식별하기 위한 메커니즘을 포함한다. 보안 평가 동작들은 상기 섹션(I, II)들에 설명된 것들과 유사하다.
- [0161] 도시된 바와 같이, 운영 체제(2000)는 동작 요청기(2005), 보안 평가기(2020), 규칙 데이터베이스(2030), 및 동작 개시기(2010)를 포함한다. 동작 개시기는 프로세서(2040) 및 태그 식별자 모듈(2050)을 포함한다. 운영 체제는 또한 설치 모듈(2060) 및 콘텐츠 열기 모듈(2070)을 포함한다.
- [0162] 몇몇 실시예들에서, 운영 체제(2000)는 도 1의 운영 체제(100)와 유사하다. 구체적으로, 동작 요청기(140)와 같이, 동작 요청기(2005)는 동작 개시기(130)에 대해 동작 수행 요청들을 한다. 설치 모듈(170) 및 콘텐츠 열기 모듈(174)과 같이, 설치 모듈(2060) 및 콘텐츠 열기 모듈(2070)은 승인된 요청들을 수신하고 승인된 동작들을 수행하는 동작 핸들러들이다.
- [0163] 보안 평가기(2020) 및 규칙 데이터베이스(2030)는 도 1에 도시된 보안 평가기(110) 및 규칙 데이터베이스(120)와 유사한 동작들을 수행한다. 보안 평가기 및 규칙 데이터베이스의 기능들은 또한 도 7 내지 도 17을 참조하여 섹션 II에서 전술되어 있다. 예를 들어, 규칙 데이터베이스(2030)는 몇몇 실시예들에서 권한 테이블 및 캐시 테이블 둘 모두를 포함한다. 몇몇 실시예들에서, 보안 평가기(2020)는 사용자 입력에 기초하여 태깅된 오브젝트의 보안 평가를 수행한다. 예를 들어, 태깅된 오브젝트에 대한 보안 평가 요청을 수신할 때, 보안 평가기(2020)는 요청된 동작이 다운로드된 오브젝트를 수반한다고 사용자에게 알리며 위험들을 알고서 진행할지 여부를 사용자에게 묻는다.
- [0164] 동작 개시기(130)처럼, 동작 개시기(2010)는 특정 동작이 발생하게 하거나 발생하지 않게 함으로써 운영 체제의 보안 정책을 시행한다. 동작 개시기는 동작 요청기(2005)로부터 특정 동작을 위한 요청을 수신하며, 보안 평가기(2020)에게 운영 체제(2000)의 보안 정책들 하에서 요청된 동작을 평가하라고 요청한다. 그러나, 도 1의 동작 개시기(130)와 달리, 동작 개시기(2010)는 요청된 동작에 대한 데이터 오브젝트가 인터넷 또는 임의의 다른 외부 소스들로부터 온 것으로서 태깅되었는지 여부를 검사한다. 외부 소스들로부터 온 것으로 태깅된(예컨대, 검역 비트가 태깅된) 데이터 오브젝트들은 보안 평가의 대상이 된다. 이미 평가된 데이터 오브젝트들에 대한 반복된 보안 평가를 피하기 위해, 몇몇 실시예들은 데이터 오브젝트가 이미 이전에 평가되었음을 표시하기 위해 보안 평가 후 태그를 업데이트한다. 몇몇 실시예들은 검역 비트를 제거함으로써 이를 달성한다. 몇몇 실시예들에서, 업데이트된 태그는 또한 데이터 오브젝트가 이전 보안 평가를 통과하였거나 실패하였는지 여부를 표시한다.
- [0165] 도 20의 예에서, 동작 개시기(2010) 내부에서의 태그 식별자 모듈(2050)은 태그를 검사하며 프로세서 모듈(2040)에 보고한다. 운영 체제(2000)에서, 보안 평가기(2020)는 태그에 기초하여 보안 평가를 하지 않는다. 이는 요청된 동작을 계속할지 여부를 결정하기 위해 태그를 사용하는 동작 개시기(2010) 내부에서의 프로세서(2040)이다. 이들 실시예들 중 몇몇에서, 동작 개시기(2010)는 요청된 동작과 연관된 데이터 오브젝트가 외부 소스들로부터 온 것이고 그 데이터 오브젝트가 결코 평가되지 않았다고 태깅된 경우에만 보안 평가로 진행한다. 데이터 오브젝트가 이와 같이 태깅되지 않았다면(예컨대, 검역 비트가 제거되었다면, 또는 데이터 오브젝트가 이전에 평가된 것으로 이미 태깅되었다면), 동작 개시기(2010)는 보안 평가기에게 보안 평가를 수행하라고 요청하지 않을 것이다.
- [0166] 도 21 및 도 22는 데이터 오브젝트들에서 태그들을 식별하는 동작 개시기를 사용하여 보안 평가를 수행할 때 운영 체제(2000)에서의 데이터의 흐름을 도시한다. 도 20에서처럼, 도 21 및 도 22에서의 운영 체제(2000)는 보안 평가기(2020), 규칙 데이터베이스(2030), 동작 개시기(2010), 동작 요청기(2005), 설치 모듈(2060), 및 콘텐츠 열기 모듈(2070)을 포함한다.
- [0167] 도 21은 운영 체제(2000)에 애플리케이션을 설치하는 동작의 데이터 흐름을 도시한다. 동작은 동작 요청기가

동작 개시기(2010)에 애플리케이션 X 설치 요청을 할 때 동작 요청기(2005)에서 시작한다(동작 '1'). 프로세서(2040)는 이어서 애플리케이션 X가 태깅되었는지 여부를 묻기 위해 태그 식별자 모듈(2050)에 명령을 전송한다(동작 '2'). 애플리케이션 X가 외부 소스로부터 온 것(예컨대, 검역 비트를 갖는 것)으로서 태깅되어 있다면, 태그 식별자(2050)는 그 확인을 프로세서(2040)에 전송하며(동작 '3'), 프로세서는 이어서 보안 평가기(2020)에게 보안 평가를 수행하라고 요청한다(동작 '4'). 다른 한편, 애플리케이션 X가 태깅되지 않았다면 또는 애플리케이션 X가 이전에 평가된 것(또는 그의 검역 비트가 제거된 것)으로서 태깅되어 있다면, 몇몇 실시예들에서 프로세서(2040)는 보안 평가기(2020)에게 보안 평가를 수행하라고 요청하지 않을 것이다. 몇몇 실시예들에서, 애플리케이션 X가 이전 보안 평가를 실패한 것으로 태깅되어 있다면, 프로세서(2040)는 보안 평가기(2020)를 수반함이 없이 설치 프로세스를 즉시 종료할 수 있다. 반대로, 애플리케이션 X가 이전 보안 평가를 통과한 것으로 태깅되어 있다면, 몇몇 실시예들에서 프로세서(2040)는 애플리케이션 X가 보안 평가기(2020)를 수반함이 없이 즉시 진행되게 할 수 있다.

[0168] 보안 평가 요청이 이루어졌다면, 보안 평가기(2020)는 규칙 데이터베이스(2030)에 질의한다(동작 '5'). 규칙 데이터베이스(2030)는 데이터베이스에서의 규칙들 또는 명령어들을 보안 평가기(2020)에 제공한다(동작 '6'). 보안 평가기(2020) 및 규칙 데이터베이스(2030) 사이에서의 이러한 질의 프로세스는, 매칭 규칙이 규칙 데이터베이스에서 발견될 때까지 또는 규칙 데이터베이스에서 적용 가능한 규칙들이 없다는 결정이 이루어질 때까지 계속된다.

[0169] 일단 매칭 규칙이 발견되었다면, 보안 평가기(2020)는 애플리케이션 X의 설치와 관련있는 데이터 오브젝트들에 매칭 규칙을 적용한다. 보안 평가는 이어서 동작 개시기(2010)로 다시 전달된다(동작 '7'). 도 21의 예에서, 보안 평가기로부터 다시 동작 개시기로의 전달은, 애플리케이션 X의 설치와 관련있는 데이터 오브젝트들이 규칙 데이터베이스에 저장된 보안 정책들을 준수하는 것으로 발견되었기 때문에, 애플리케이션 X를 설치하는 것이 OK임을 표시한다.

[0170] 애플리케이션 X가 보안 평가기(2020)에 의해 수행된 보안 평가를 통과하였음을 알 때, 프로세서(2040)는 애플리케이션 X의 설치를 계속할지 여부를 결정한다. 보안 평가기에 의해 이루어진 보안 평가에 더하여, 몇몇 실시예들에서 프로세서(2040)는 애플리케이션 X가 태깅되었는지 여부를 고려한다. 프로세서가 애플리케이션 X의 설치를 계속하는 것이 OK임을 결정한다면, 프로세서는 애플리케이션 X의 설치를 시작하기 위해 설치 모듈(2060)에 명령을 전송한다(동작 '8'). 콘텐츠 열기 모듈(2070) 주위에서의 파선들은 콘텐츠 열기 모듈이 이러한 동작에 수반되지 않는다는 것을 표시한다.

[0171] 도 22는 운영 체제(2000)에서 문서를 여는 동작에 대한 데이터 흐름도를 도시한다. 동작은 동작 요청기가 동작 개시기(2010)에 데이터 파일 X를 열기 위한 요청을 할 때 동작 요청기(2005)에서 시작한다(동작 '1'). 프로세서(2040)는 이어서 데이터 파일 X가 태깅되었는지 여부를 묻기 위해 태그 식별자 모듈(2050)에 명령을 전송한다(동작 '2'). 데이터 파일 X가 외부 소스로부터 온 것(예컨대, 검역 비트를 갖는 것)으로 태깅되어 있다면, 태그 식별자(2050)는 그 확인을 프로세서(2040)에 전송하며(동작 '3'), 프로세서는 이어서 보안 평가기(2020)에게 보안 평가를 수행하라고 요청한다(동작 '4'). 다른 한편, 데이터 파일 X가 태깅되지 않았다면 또는 데이터 파일 X가 이전에 평가된 것으로 태깅되어 있다면, 몇몇 실시예들에서 프로세서(2040)는 보안 평가기(2020)에게 보안 평가를 수행하라고 요청하지 않을 것이다. 몇몇 실시예들에서, 데이터 파일 X가 이전 보안 평가를 실패한 것으로 태깅되어 있다면, 프로세서(2040)는 보안 평가기(2020)를 수반함이 없이 파일 열기 프로세스를 즉시 종료할 수 있다. 반대로, 데이터 파일 X가 이전 보안 평가를 통과한 것으로 태깅되어 있다면, 몇몇 실시예들에서 프로세서(2040)는 데이터 파일 X의 열기가 보안 평가기(2040)를 수반함이 없이 즉시 진행되게 할 수 있다.

[0172] 보안 평가 요청이 이루어졌다면, 보안 평가기(2020)는 이어서 규칙 데이터베이스(2030)에 질의한다(동작 '5'). 규칙 데이터베이스(2030)는 데이터베이스에서의 규칙들 또는 명령어들을 보안 평가기(2020)에 제공한다(동작 '6'). 보안 평가기(2020) 및 규칙 데이터베이스(2030) 사이에서의 이러한 질의 프로세스는, 매칭 규칙이 규칙 데이터베이스에서 발견될 때까지 또는 규칙 데이터베이스에서 적용 가능한 규칙들이 없다는 결정이 이루어질 때까지 계속된다.

[0173] 일단 매칭 규칙이 발견되었다면, 보안 평가기(2020)는 데이터 파일 X의 열기와 관련있는 데이터 오브젝트들에 매칭 규칙들을 적용한다. 보안 평가는 이어서 동작 개시기(2010)로 다시 전달된다(동작 '7'). 도 22의 예에서, 보안 평가기로부터 다시 동작 개시기로의 전달은, 데이터 파일 X가 규칙 데이터베이스에 저장된 보안 정책들을 준수하는 것으로 발견되었기 때문에, 데이터 파일 X를 여는 것이 OK임을 표시한다.

[0174] 데이터 파일 X가 보안 평가기(2020)에 의해 수행된 보안 평가를 통과하였음을 알 때, 프로세서(2040)는 데이터

파일 X의 열기를 계속할지 여부를 결정한다. 보안 평가기에 의해 이루어진 보안 평가에 더하여, 몇몇 실시예들에서 프로세서(2040)는 또한 데이터 파일 X가 태깅되었는지 여부를 고려한다. 프로세서가 데이터 파일 X의 열기를 계속하는 것이 OK임을 결정한다면, 프로세서는 데이터 파일 X의 열기를 시작하기 위해 콘텐츠 열기 모듈(2070)에 명령을 전송한다(동작 '8'). 설치 모듈(2060) 주위에서의 파선들은 설치 모듈이 이러한 동작에 수반되지 않음을 표시한다.

[0175] 도 20 내지 도 22는 데이터 오브젝트가 인터넷과 같은 외부 소스로부터 온 것으로서 태깅되어 있는지 여부를 결정하는 동작 개시기(2010)를 갖는 운영 체제(2000)를 도시한다. 운영 체제(2000)의 동작 개시기(2010)는 또한 태그에 기초하여 동작을 계속할지 여부에 대한 결정을 한다. 규칙 데이터베이스에 저장된 규칙들은 태그를 고려하거나 프로세싱하지 않는다. 그러나, 몇몇 다른 실시예들에서, 운영 체제의 규칙 데이터베이스는 태그를 식별하고 식별된 태그에 기초하여 보안 평가들을 하는 규칙들을 포함한다.

[0176] 몇몇 실시예들에 대해, 도 23은 다운로드된 데이터 오브젝트들과 연관된 태그들을 프로세싱하기 위한 규칙들을 갖는 규칙 데이터베이스를 포함하는 운영 체제(2300)를 도시한다. 몇몇 그러한 실시예들에서의 동작 개시기는 요청된 동작과 연관되는 데이터 오브젝트의 태그를 검사하지 않는다. 운영 체제(2000)와 같이, 운영 체제(2300)는 동작 요청기(2305), 보안 평가기(2320), 규칙 데이터베이스(2330), 및 동작 개시기(2310)를 포함한다. 운영 체제는 또한 설치 모듈(2360) 및 콘텐츠 열기 모듈(2370)을 포함한다.

[0177] 동작 개시기(2310)는 동작 요청기(2305)로부터 동작 요청들을 수신하고 보안 평가기(2320)로 보안 평가 요청들을 한다. 그러나, 동작 개시기(2010)와는 달리, 동작 개시기(2310)는 다운로드 태그들의 검사를 수행하지 않는다.

[0178] 보안 평가기(2320)는 동작 개시기(2310)로부터 보안 평가 요청들을 수신하고 각각의 보안 평가 요청과 연관된 데이터 오브젝트를 매칭시키는 규칙들에 대해 규칙 데이터베이스에 질의한다.

[0179] 규칙 데이터베이스(2330)는 보안 평가를 수행하기 위한 규칙들을 저장한다. 규칙 데이터베이스(2030)와는 달리, 규칙 데이터베이스(2330)는 또한 데이터 오브젝트들이 태깅되어 있는지를 알기 위해 데이터 오브젝트들을 분석하기 위한 규칙들을 포함한다. 데이터 오브젝트가 태깅되어 있다면, 몇몇 실시예들은 이어서 데이터 오브젝트에 대한 보안 평가를 수행하기 위해 규칙 데이터베이스에서의 규칙들을 적용하도록 진행하며 요청된 동작을 승인/미승인한다. 몇몇 실시예들에서, 규칙 데이터베이스(2330)는 섹션 II에서 이전에 설명된 바와 같이, 권한 테이블 및 캐시 테이블 둘 모두를 포함한다. 예를 들어, 데이터 오브젝트의 검역 비트 때문에 이전 보안 평가에 의해 승인되지 않았던 데이터 오브젝트는 데이터 오브젝트를 승인하지 않은 동작 필드를 갖는 캐시 엔트리를 가질 수 있다.

#### [0180] IV. 문서들의 보안 평가

[0181] 데이터 오브젝트의 소스를 식별하기 위한 서명이 데이터 오브젝트의 콘텐츠에 부분적으로 기초하여 도출되므로, 서명들은 데이터 오브젝트가 변경되었는지 여부를 시험하기 위해 사용될 수 있다. 이는 또한 콘텐츠가 사용자들에 의해 빈번하게 변경되는 데이터 오브젝트(예컨대, 문서)가 그 자신의 서명을 가질 수 없음을 의미한다. 그러나, 데이터 파일은 서명을 포함하는 증명서를 지닐 수 있는(아카이브와 같은) 구조로 배치될 수 있다. 그러한 서명은 구조에서 데이터 파일들의 소스를 안전하게 식별하기 위해 사용될 수 있다. 몇몇 실시예들에서, 운영 체제는 아카이브 구조들에 내장된 증명서들에서의 식별자들에 기초하여 문서들 또는 데이터 파일들을 승인하거나 승인하지 않는 규칙들을 규칙 데이터베이스에 포함한다. 승인된 아카이브 내의 소정 문서는 또한, 이들 실시예들 중 몇몇에 의해, 승인된 것으로 간주된다.

[0182] 도 24는 증명서들을 갖는 데이터 아카이브들을 수신 및 저장하는 컴퓨터(2400)를 도시한다. 증명서들(서명들을 가짐)은 이어서 아카이브 내의 콘텐츠의 소스를 안전하게 식별하기 위해 컴퓨터(2400)의 운영 체제에 의해 사용될 수 있다. 예시된 바와 같이, 컴퓨터(2400)는 저장 디바이스(2410), 컴퓨터(2400) 상에서 현재 실행되고 있는 애플리케이션(2430), 네트워크 인터페이스(2440), 및 사용자 인터페이스(UI) 모듈(2450)을 포함한다. UI 모듈(2450)은 입력 디바이스 드라이버(2460)들 및 디스플레이 모듈(2470)을 통해 사용자와의 통신을 제어한다.

[0183] 네트워크 인터페이스(2440)는 네트워크를 통해 컴퓨터(2400)와 외부 세계 사이에 통신 인터페이스를 제공한다. 네트워크는 컴퓨터(2400)가 인터넷을 통해 통신하게 하고 다른 컴퓨터들로부터 데이터 아카이브(2481 내지 2484)들과 같은 데이터 오브젝트들에 액세스하게 한다. 이들 아카이브들의 콘텐츠는 텍스트 파일들, 워드 프로세서 문서들, 스프레드시트들, 미디어 파일들 등일 수 있다. 이들 아카이브들의 콘텐츠는 또한 실행 가능한 애플리케이션들 또는 애플리케이션 설치 파일들일 수 있다. 이들 데이터 아카이브들 중 몇몇 또는 모두는 그들의

소스를 식별하는 서명들을 갖는 증명서들을 갖는다. 데이터 아카이브는 (데이터 아카이브(2481)와 같은) 하나의 콘텐츠 또는 (데이터 아카이브(2484)와 같은) 다수의 콘텐츠들 또는 파일들을 포함할 수 있다. 도면에 도시되지 않지만, 컴퓨터(2400)는 또한 외부 플래시 드라이브와 같은 데이터 통신의 다른 수단들을 통해 다른 컴퓨터들로부터 데이터 아카이브들에 액세스할 수 있다.

[0184] 애플리케이션(2430)은 운영 체제를 통해 컴퓨터(2400) 상에서 현재 실행되고 있는 프로그램 또는 프로세스들의 세트이다. 애플리케이션(2430)은 인터넷 브라우저, 워드 프로세서, 비디오 게임, 미디어 편집 애플리케이션, 또는 컴퓨터(2400)에서 동작할 수 있는 임의의 프로그램일 수 있다. 애플리케이션(2430)은 네트워크 인터페이스(2440)를 통해 인터넷으로부터 데이터 아카이브(2481 내지 2484)들을 다운로드하는 것을 포함한, 인터넷에 의한 통신을 요구하는 동작들을 수행할 수 있다.

[0185] 데이터 아카이브들이 다운로드되었다면, 애플리케이션(2430)은 저장 디바이스(2410)에 다운로드된 파일들을 저장한다. 저장 디바이스(2410) 내로의 데이터 아카이브들의 저장 동작은 도 19를 참조하여 이전에 설명된 바와 같은 태깅되거나 검역된 데이터의 저장 동작과 유사하다. 저장 디바이스(2410)는 데이터 아카이브들을 저장한다. 일단 데이터 아카이브들이 저장되면, 운영 체제는 서명들을 인증하고 데이터 아카이브들에 내장된 증명서들을 조사함으로써 보안 평가들을 수행할 수 있다.

[0186] 도 25는 데이터 아카이브들에서의 증명서들에 기초하여 데이터 파일들 또는 문서들의 보안 평가들을 하기 위해 규칙 데이터베이스에서 규칙들을 포함하는 운영 체제를 도시한다. 구체적으로, 도 25는 운영 체제(2500)에서의 예시적인 데이터 파일 열기 요청을 도시한다. 데이터 파일이 데이터 아카이브에 내장된다.

[0187] 운영 체제(2500)는 동작 요청기(2505), 보안 평가기(2520), 규칙 데이터베이스(2530), 동작 개시기(2510) 및 콘텐츠 열기 모듈(2570)을 포함한다. 운영 체제(2500)는 또한 데이터 아카이브(2580)를 포함한 수 개의 데이터 아카이브들을 저장하는 저장 디바이스(2560)에 액세스한다. 몇몇 실시예들에서, 저장 디바이스(2560)는 도 24의 저장 장치(2410)를 포함하며, 저장 장치는 인터넷 또는 다른 외부 소스들로부터 다운로드된 데이터 아카이브들을 저장한다.

[0188] 몇몇 실시예들에서, 운영 체제(2500)는 도 1의 운영 체제(100)와 유사하다. 구체적으로, 동작 요청기(140)와 같이, 동작 요청기(2505)는 동작 개시기(2510)에 대해 동작 수행 요청들을 한다. 동작 개시기(2510)는 동작 요청기(2505)로부터 요청을 수신하고 보안 평가기(2520)로 보안 평가 요청을 한다. 도 25에 의해 도시된 예에서, 요청은 데이터 아카이브(2580)에 내장된 문서 또는 데이터 파일을 열기 위한 것이다. 동작 개시기(2510)는 데이터 아카이브(2580)를 보안 평가기에 전달함으로써 보안 평가기(2520)로 요청을 한다. 일단 동작 개시기(2510)가 보안 평가기(2520)로부터 응답을 수신하면, 동작 개시기는 보안 평가에 기초하여 콘텐츠 열기 모듈(2570)을 시작하거나 요청된 동작을 종료한다.

[0189] 보안 평가기(2520) 및 규칙 데이터베이스(2530)는 보안 평가기(110) 및 규칙 데이터베이스(120)와 유사한 동작들을 수행한다. 몇몇 실시예들에서, 규칙 데이터베이스는 도 7 내지 도 17을 참조하여 섹션 II에 기술된 바와 같이 권한 테이블 및 캐시 테이블 둘 모두를 포함한다. 데이터 파일들 또는 문서들의 소스를 안전하게 식별하기 위해, 보안 평가기(2520)는 규칙 데이터베이스(2530)에 저장된 규칙들을 적용하기 전에 서명을 인증한다.

[0190] 콘텐츠 열기 모듈(2570)은 도 1의 콘텐츠 열기 모듈(174)과 유사하다. 일단 콘텐츠 열기 모듈이 동작 개시기(2510)로부터 시작 명령을 수신한다면, 콘텐츠 열기 모듈(2570)은 요청된 데이터 파일을 열도록 진행한다. 도 25의 예에서, 열리도록 요청된 데이터 파일은 저장 디바이스(2560)에 저장된 데이터 아카이브(2580)에 있다. 데이터 파일을 열기 위해, 몇몇 실시예들에서 콘텐츠 열기 모듈(2570)은 아카이브 구조로부터 데이터 파일을 추출하고 이어서 (예컨대, 데이터 파일을 개방하기 위해 필요한 애플리케이션에 데이터 파일을 전송함으로써) 데이터 파일을 연다.

[0191] 몇몇 실시예들에 대해, 도 26은 문서 열기 동작들의 보안 평가들을 수행하는 프로세스(2600)를 개념적으로 도시한다. 구체적으로, 프로세스(2600)는 데이터 아카이브들에 내장된 서명들을 갖는 증명서들에 기초하여 보안 평가 결정을 하도록 운영 체제에 의해 수행된다.

[0192] 프로세스(2600)는 데이터 파일(또는 문서) 열기 요청을 수신할 때 시작한다. 프로세스는 개방될 것이 요청되는 문서를 수신한다(2610에서). 프로세스는 이어서 문서가 안전한 유형의 것인지 여부를 결정한다(2620에서). 몇몇 실시예들에서, 컴퓨터를 손상시킬 것 같지 않은 것으로 간주되는 유형들의 문서들 또는 파일들이 그의 소스에 관계없이 열리게 된다. 예를 들어, 텍스트 파일들은 안전한 것으로 고려되며 소스의 어떠한 서명 또는 다른 식별들도 요구하지 않는다. 문서가 안전한 유형의 것으로 간주되면, 프로세스는 문서를 열기 위해 (도 25의 콘

텐트 열기 모듈(2570)과 같은) 동작 핸들러에 문서를 전달하기 위해 2670으로 진행한다. 문서가 안전한 유형의 것으로 간주되지 않는다면, 프로세스는 2630으로 진행한다. 실행 가능한 문서들이 몇몇 실시예들에서는 안전한 것으로 간주되지 않는다. 소정 유형들의 비-실행 가능한 문서들은 또한, 그러한 비-실행 가능한 문서들이 이들 문서들을 열어 사용하는 애플리케이션들을 통해 컴퓨터를 손상시킬 수 있는 악성 코드를 지닐 수 있으므로, 몇몇 실시예들에서 소스 검증을 요구한다.

[0193] 다음으로, 프로세스는 문서가 아카이브에 있는지 여부를 결정한다(2630에서). 전술된 바와 같이, 몇몇 데이터 파일들 또는 문서들은 소스 식별 서명을 갖지 않지만, 데이터 파일들은 서명을 지닌 아카이브 구조 내에 배치될 수 있다. 그러나, 몇몇 실시예들에서, 소정 유형들의 문서들은 아카이브 내에 있지 않고 그들 자신의 서명들을 포함할 수 있다(예컨대, 거의 변경되지 않는 문서들은 파일이 변경되었는지 여부를 테스트하는 서명을 지닐 수 있다). 문서가 아카이브 내에 있다면, 프로세스(2600)는 2650으로 진행한다. 그렇지 않다면, 프로세스는 문서의 보안 평가를 그것만으로 수행하기 위해 2640으로 진행한다.

[0194] 2640에서, 프로세스는 문서 자신의 증명서에 기초하여 보안 평가를 수행한다. 동작은 문서의 증명서에 내장되는 문서의 서명의 인증뿐만 아니라 증명서에서의 다른 정보에 기초한 부가적인 보안 평가들을 포함한다. 몇몇 실시예들에서, 프로세스는 문서에 대한 보안 평가를 하는 데 사용될 수 있는 매칭 규칙들을 위해 규칙 데이터베이스에 질의한다. 몇몇 실시예들에서 이러한 동작은 문서에 코드 요건들을 적용하는 것을 포함한다. 코드 요건들은 문서에 내장된 증명서들을 조사하라는 그리고 소스가 신뢰되는지를 검사하라는 명령어들을 포함할 수 있다. 보안 평가를 수행한 후, 프로세스는 매칭 규칙들에 기초하여 열기에 대해 문서를 승인할지 여부를 결정한다(2660에서). 그렇다면, 프로세스는 열기 위해 문서를 (도 25의 콘텐츠 열기 모듈(2570)과 같은) 동작 핸들러에 전달한다(2670에서). 그렇지 않다면, 프로세스는 종료한다.

[0195] 2650에서, 프로세스는 아카이브의 증명서에 기초하여 보안 평가를 수행한다. 동작은 아카이브의 증명서에 내장되는 서명의 인증뿐만 아니라 증명서에서의 다른 정보에 기초한 부가적인 보안 평가들을 포함한다. 몇몇 실시예들에서, 프로세스는 아카이브에 대한 보안 평가를 하는 데 사용될 수 있는 매칭 규칙들을 위해 규칙 데이터베이스에 질의한다. 몇몇 실시예들에서 이러한 동작은 아카이브에 코드 요건들을 적용하는 것을 포함한다. 코드 요건들은 아카이브에서 증명서들을 조사하라는 그리고 소스가 신뢰되는지를 검사하라는 명령어들을 포함할 수 있다. 프로세스는 이어서 매칭 규칙들에 기초하여 아카이브 및 따라서 열기 위한 아카이브 내의 문서들 모두를 승인할지 여부를 결정한다(2660에서). 그렇다면, 프로세스는 열기 위해 문서를 (도 25의 콘텐츠 열기 모듈(2570)과 같은) 동작 핸들러에 전달한다(2670에서). 그렇지 않다면, 프로세스는 종료한다.

## [0196] V. 전자 시스템

[0197] 전술된 특징들 및 애플리케이션들 중 많은 것은 컴퓨터 판독가능 저장 매체(또한 컴퓨터 판독가능 매체로 지칭됨) 상에 기록된 명령어들의 세트로서 규정되는 소프트웨어 프로세스들로서 구현된다. 이들 명령어들이 하나 이상의 계산 또는 프로세싱 유닛(들)(예컨대, 하나 이상의 프로세서들, 프로세서들의 코어들, 또는 다른 프로세싱 유닛들)에 의해 실행될 때, 명령어들은 프로세싱 유닛(들)이 명령어들에서 지시된 동작들을 수행하게 한다. 컴퓨터 판독가능 미디어의 예들은, CD-ROM, 플래시 드라이브, RAM(random access memory) 칩, 하드 드라이브, EPROM(erasable programmable read only memory), EEPROM(electrically erasable programmable read-only memory) 등을 포함하지만 이로 한정되지 않는다. 컴퓨터 판독가능 미디어는 무선으로 또는 유선 연결들을 통해 전달하는 반송파(carrier wave) 및 전자 신호를 포함하지 않는다.

[0198] 본 명세서에서, 용어 "소프트웨어"는 ROM(read-only memory)에 상주하는 펌웨어 또는 프로세서에 의한 프로세싱을 위해 메모리 내로 판독될 수 있는 자기 저장 장치에 저장된 애플리케이션들을 포함하도록 의도된다. 또한, 몇몇 실시예들에서, 다수의 소프트웨어 발명들은 별개의 소프트웨어 발명들에 남아 있으면서 보다 큰 프로그램의 하위-부분들로서 구현될 수 있다. 몇몇 실시예들에서, 다수의 소프트웨어 발명들은 또한 별개의 프로그램들로서 구현될 수 있다. 마지막으로, 본 명세서에 설명된 소프트웨어 발명을 함께 구현하는 별개의 프로그램들의 임의의 조합은 본 발명의 범주 내에 있다. 몇몇 실시예들에서, 소프트웨어 프로그램들은, 하나 이상의 전자 시스템들 상에서 동작하도록 설치된 때, 소프트웨어 프로그램들의 동작들을 실행 및 수행하는 하나 이상의 특정 기계 구현예들을 한정한다.

[0199] 도 27은 본 발명의 몇몇 실시예들을 구현시키는 전자 시스템(2700)을 개념적으로 도시한다. 전자 시스템(2700)은 컴퓨터(예컨대, 데스크탑 컴퓨터, 개인용 컴퓨터, 태블릿 컴퓨터 등), 전화기, PDA, 또는 임의의 다른 종류의 전자 디바이스일 수 있다. 그러한 전자 시스템은 다양한 유형들의 컴퓨터 판독가능 미디어 및 다양한 다른 유형들의 컴퓨터 판독가능 미디어를 위한 인터페이스들을 포함한다. 전자 시스템(2700)은 버스(2705), 프로

세싱 유닛(들)(2710), 그래픽 프로세싱 유닛(GPU)(2715), 시스템 메모리(2720), 네트워크(2725), ROM(2730), 영구 저장 디바이스(2735), 입력 디바이스(2740), 및 출력 디바이스(2745)를 포함한다.

- [0200] 버스(2705)는 전자 시스템(2700)의 다수의 내부 디바이스들을 통신적으로 연결하는 모든 시스템, 주변 장치, 및 칩셋 버스들을 총괄하여 표현한다. 예를 들어, 버스(2705)는 ROM(2730), GPU(2715), 시스템 메모리(2720), 및 영구 저장 디바이스(2735)와 프로세싱 유닛(들)(2710)을 통신적으로 연결한다.
- [0201] 이들 다양한 메모리 유닛들로부터, 프로세싱 유닛(들)(2710)은 본 발명의 프로세스들을 실행하기 위해 실행할 명령어들과 프로세싱할 데이터를 검색한다. 프로세싱 유닛(들)은 상이한 실시예들에서 단일 프로세서 또는 멀티-코어 프로세서일 수 있다. 몇몇 명령어들은 GPU(2715)에 전달되어 이에 의해 실행된다. GPU(2715)는 다양한 계산들을 없애거나 프로세싱 유닛(들)(2710)에 의해 제공된 이미지 프로세싱을 보완한다. 몇몇 실시예들에서, 그러한 기능은 코어이미지(CoreImage)의 커널 셰이딩 언어(kernel shading language)를 사용하여 제공될 수 있다.
- [0202] ROM(2730)은 프로세싱 유닛(들)(2710) 및 전자 시스템의 다른 모듈들에 의해 요구되는 정적 데이터 및 명령어들을 저장한다. 다른 한편, 영구 저장 디바이스(2735)는 판독-기록 메모리 디바이스이다. 이러한 디바이스는 전자 시스템(2700)이 오프일 때도 명령어들과 데이터를 저장하는 비-휘발성 메모리 유닛이다. 본 발명의 몇몇 실시예들은 영구 저장 디바이스(2735)로서 (자기 또는 광학 디스크 및 그의 대응하는 디스크 드라이브와 같은) 대용량 저장 디바이스를 사용한다.
- [0203] 다른 실시예들은 영구 저장 디바이스로서 (플로피 디스크, 플래시 메모리 디바이스 등 및 그의 대응하는 디스크 드라이브와 같은) 이동식 저장 디바이스를 사용한다. 영구 저장 디바이스(2735)와 같이, 시스템 메모리(2720)는 판독-기록 메모리 디바이스이다. 그러나, 저장 디바이스(2735)와 달리, 시스템 메모리(2720)는 RAM과 같은 휘발성 판독-기록 메모리이다. 시스템 메모리(2720)는 프로세서가 런타임 시 요구하는 명령어들과 데이터의 일부를 저장한다. 몇몇 실시예들에서, 본 발명의 프로세스들은 시스템 메모리(2720), 영구 저장 디바이스(2735), 및/또는 ROM(2730)에 저장된다. 예를 들어, 다양한 메모리 유닛들이 몇몇 실시예들에 따라 멀티미디어 클립들을 프로세싱하기 위한 명령어들을 포함한다. 이들 다양한 메모리 유닛들로부터, 프로세싱 유닛(들)(2710)은 몇몇 실시예들의 프로세스들을 실행하기 위해 실행할 명령어들과 프로세싱할 데이터를 검색한다.
- [0204] 버스(2705)는 또한 입력 및 출력 디바이스(2740, 2745)들에 연결한다. 입력 디바이스(2740)들은 사용자가 전자 시스템으로 정보를 전달하게 하고 전자 시스템에 대한 명령어들을 선택하게 할 수 있다. 입력 디바이스(2740)들은 영숫자 키보드 및 포인팅 디바이스(또한 "커서 제어 디바이스"로 불림), 카메라(예컨대, 웹캠), 마이크로폰 또는 음성 명령어들을 수신하기 위한 유사한 디바이스 등을 포함한다. 출력 디바이스(2745)들은 전자 시스템에 의해 생성된 이미지들을 디스플레이하거나 그렇지 않으면 데이터를 출력한다. 출력 디바이스(2745)들은 프린터 및 음극선관(CRT) 또는 액정 디스플레이(LCD)와 같은 디스플레이 디바이스뿐만 아니라 스피커 또는 유사한 오디오 출력 디바이스를 포함한다. 몇몇 실시예들은 입력 및 출력 디바이스들 둘 모두로서 기능하는 터치스크린과 같은 디바이스들을 포함한다.
- [0205] 마지막으로, 도 27에 도시된 바와 같이, 버스(2705)는 또한 네트워크 어댑터(도시되지 않음)를 통해 전자 시스템(2700)을 네트워크(2725)에 결합시킨다. 이러한 방식으로, 컴퓨터는 컴퓨터들의 네트워크(예를 들어, 근거리 네트워크("LAN"), 광역 네트워크("WAN"), 인트라넷, 또는 인터넷과 같은 네트워크들의 네트워크)의 일부일 수 있다. 전자 시스템(2700)의 임의의 또는 모든 컴포넌트들은 본 발명과 함께 사용될 수 있다.
- [0206] 몇몇 실시예들은 전자 컴포넌트들, 예를 들어 마이크로프로세서, 기계 판독가능 또는 컴퓨터-판독가능 매체(대안적으로, 컴퓨터-판독가능 저장 미디어, 기계-판독가능 미디어, 또는 기계-판독가능 저장 미디어로 불림)에 컴퓨터 프로그램 명령어들을 저장하는 메모리 및 저장 장치를 포함한다. 그러한 컴퓨터-판독가능 미디어의 몇몇 예들은 RAM, ROM, CD-ROM(read-only compact disc), CD-R(recordable compact disc), CD-RW(rewritable compact disc), 판독-전용 DVD(digital versatile disc)(예컨대, DVD-ROM, 듀얼-레이어 DVD-ROM), 다양한 기록가능/재기록가능 DVD(예컨대, DVD-RAM, DVD-RW, DVD+RW 등), 플래시 메모리(예컨대, SD 카드, 미니-SD 카드, 마이크로-SD 카드 등), 자기 및/또는 솔리드 스테이트 하드 드라이브, 판독-전용 및 기록가능 블루-레이(Blu-Ray)(등록상표) 디스크, 초고밀도 광 디스크, 임의의 다른 광 또는 자기 미디어, 및 플로피 디스크를 포함한다. 컴퓨터-판독가능 미디어는 적어도 하나의 프로세싱 유닛에 의해 실행 가능하며 다양한 동작들을 수행하기 위한 명령어들의 세트들을 포함하는 컴퓨터 프로그램을 저장할 수 있다. 컴퓨터 프로그램들 또는 컴퓨터 코드의 예들은, 컴파일러에 의해 생성되는 바와 같은 기계 코드; 및 인터프리터를 사용하여 컴퓨터, 전자 컴포넌트 또는 마이크로프로세서에 의해 실행되는 상위-레벨 코드를 포함한 파일들을 포함한다.

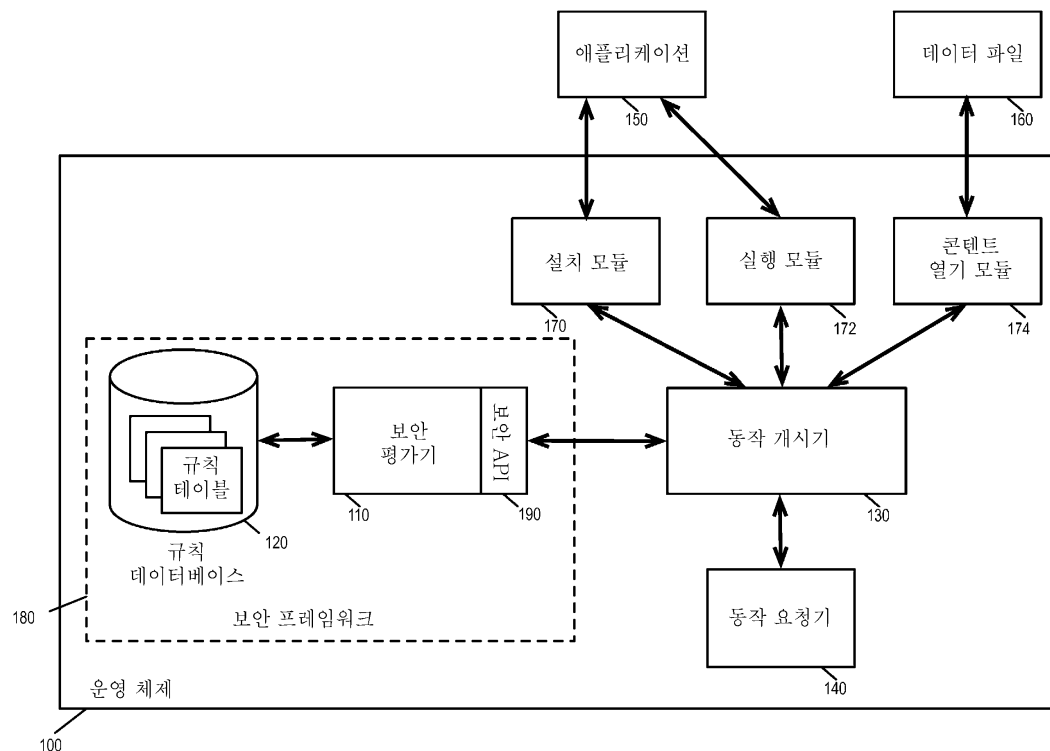
[0207] 상기 논의는 소프트웨어를 실행시키는 마이크로프로세서 또는 멀티-코어 프로세서들을 주로 언급하지만, 몇몇 실시예들은 ASIC(application specific integrated circuit) 또는 FPGA(field programmable gate array)와 같은 하나 이상의 집적 회로들에 의해 수행된다. 몇몇 실시예들에서, 그러한 집적 회로들은 회로 자체 상에 저장된 명령어들을 실행시킨다. 게다가, 몇몇 실시예들은 프로그래밍 가능한 논리 디바이스(programmable logic device, PLD), ROM, 또는 RAM 디바이스들에 저장된 소프트웨어를 실행시킨다.

[0208] 본 출원의 이러한 명세서 및 임의의 청구항들에 사용되는 바와 같이, 용어 "컴퓨터", "서버", "프로세서", 및 "메모리" 모두는 전자 디바이스 또는 다른 기술적 디바이스를 언급한다. 이들 용어들은 사람 또는 사람들의 그룹들을 배제한다. 명세서의 목적들을 위해, 용어 "디스플레이" 또는 "디스플레이하는"은 전자 디바이스 상에서 디스플레이하는 것을 의미한다. 본 출원의 이러한 명세서 및 임의의 청구항들에 사용되는 바와 같이, 용어들 "컴퓨터 판독가능 매체", "컴퓨터 판독가능 미디어", 및 "기계 판독가능 미디어"는 전체적으로 컴퓨터에 의해 판독가능한 형태로 정보를 저장하는 유형적인 물리적 오브젝트들로 제한된다. 이들 용어들은 임의의 무선 신호들, 유선 다운로드 신호들, 및 임의의 다른 임시 신호들을 제외한다.

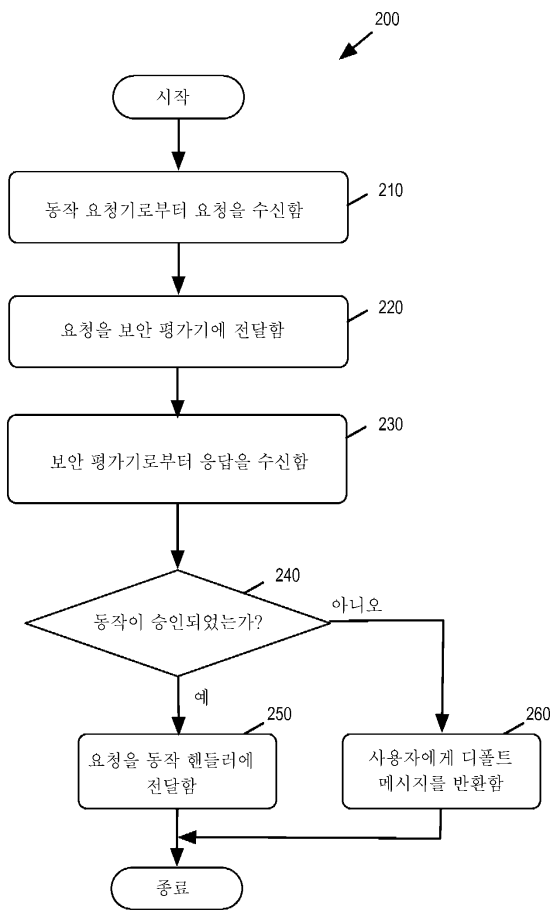
[0209] 본 발명은 다수의 특정 상세 사항들을 참조하여 설명되었지만, 당업자는 본 발명이 본 발명의 사상으로부터 벗어남이 없이 다른 특정 형태들로 구체화될 수 있다는 것을 인식할 것이다. 게다가, 도면들 중 적어도 몇몇(도 2, 도 3, 도 13, 도 17, 도 18, 및 도 26을 포함함)은 프로세스들을 개념적으로 도시한다. 이들 프로세스들의 특정 동작들은 도시되고 설명된 정확한 순서로 수행되지 않을 수 있다. 특정 동작들은 하나의 연속적인 시리즈의 동작들로 수행되지 않을 수 있으며, 상이한 특정 동작들이 상이한 실시예들에서 수행될 수 있다. 더욱이, 프로세스는 수 개의 서브-프로세스들을 사용하여 또는 보다 큰 매크로 프로세스의 일부로서 구현될 수 있다. 따라서, 당업자는 본 발명이 상기 예시적인 상세 사항들에 의해 제한되는 것이 아니라 오히려 첨부된 특허청구 범위에 의해 한정된다는 것을 이해할 것이다.

## 도면

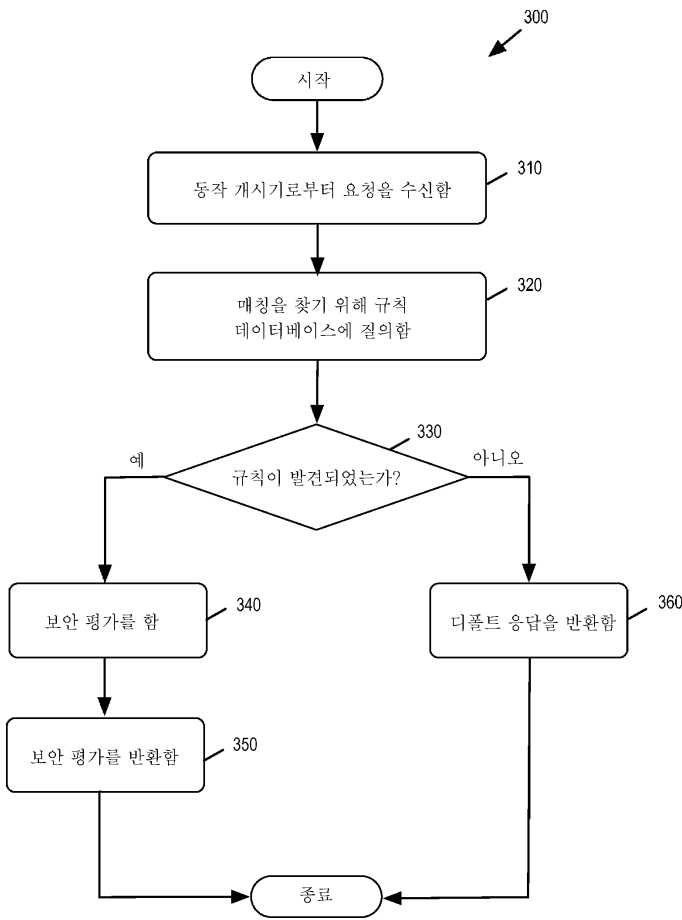
### 도면1



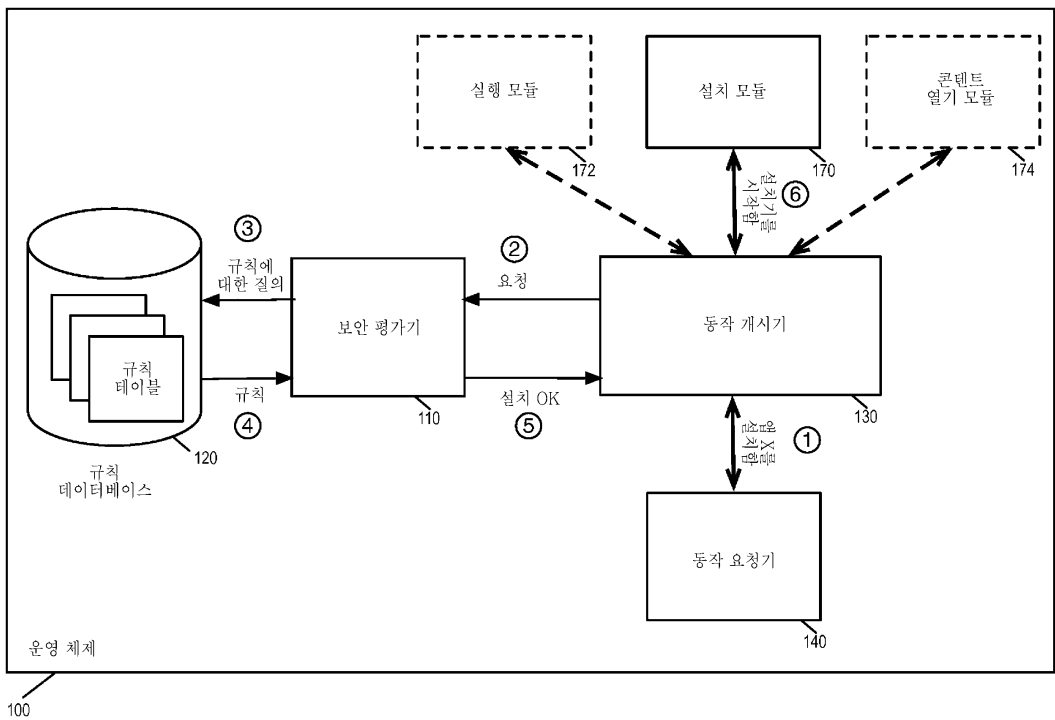
도면2



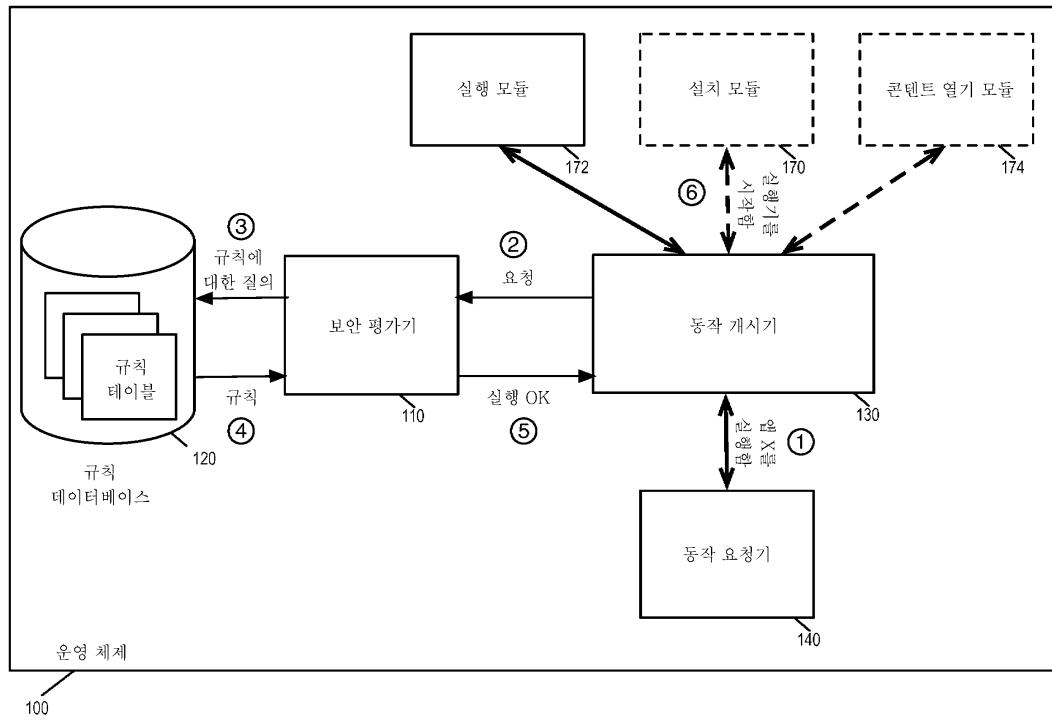
도면3



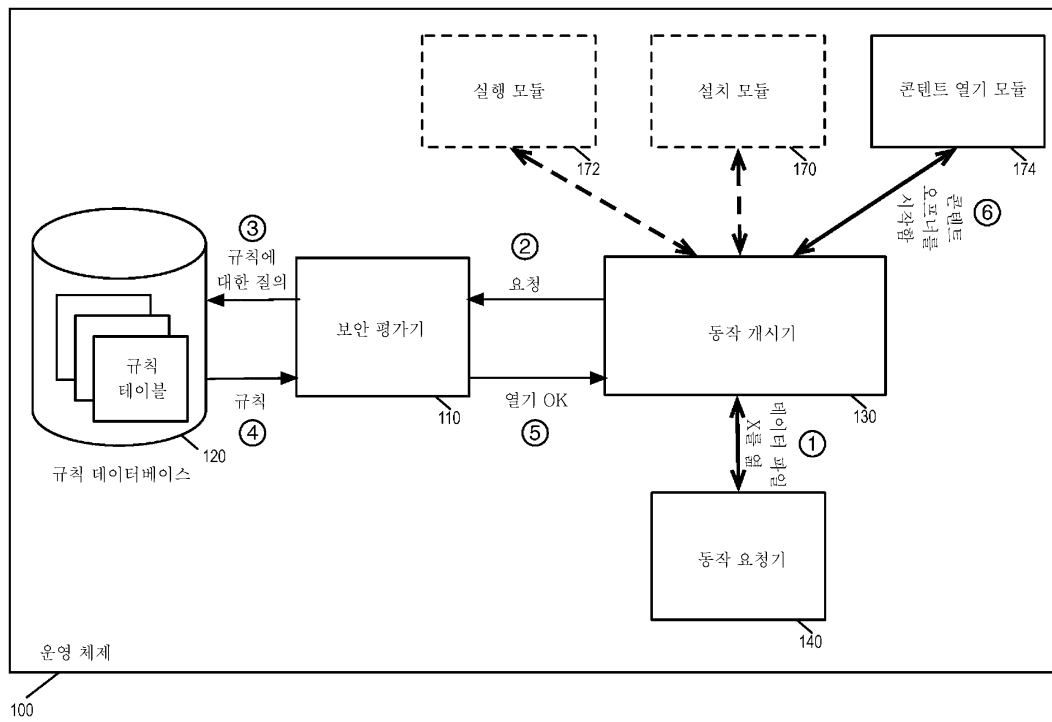
도면4



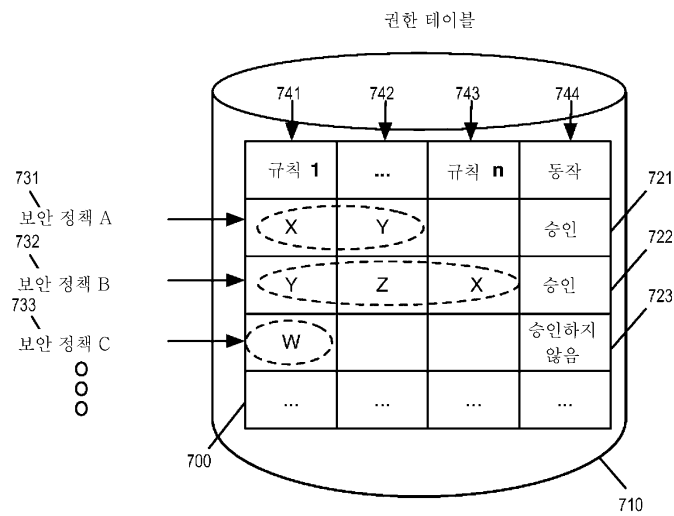
도면5



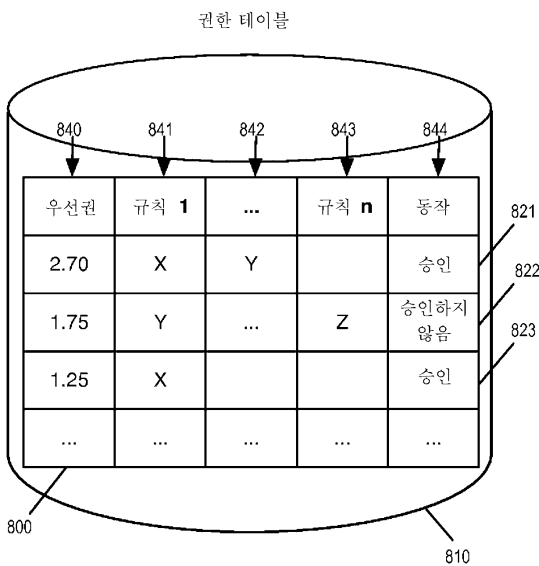
도면6



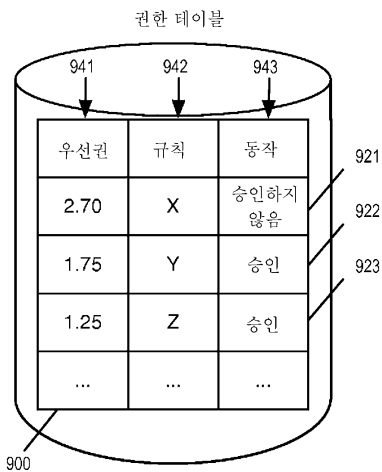
도면7



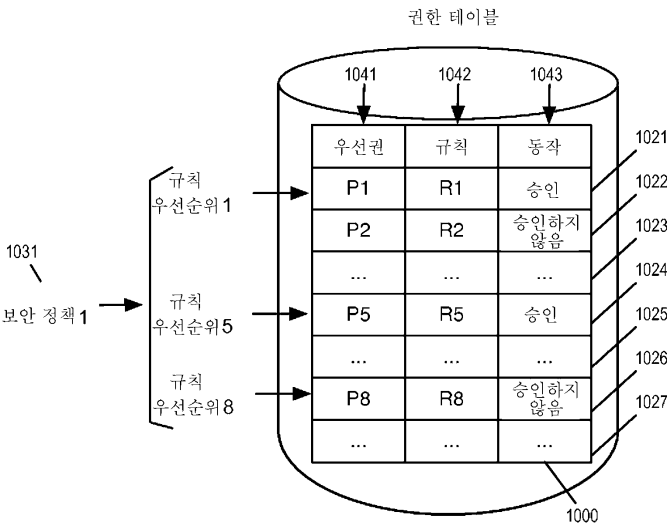
도면8



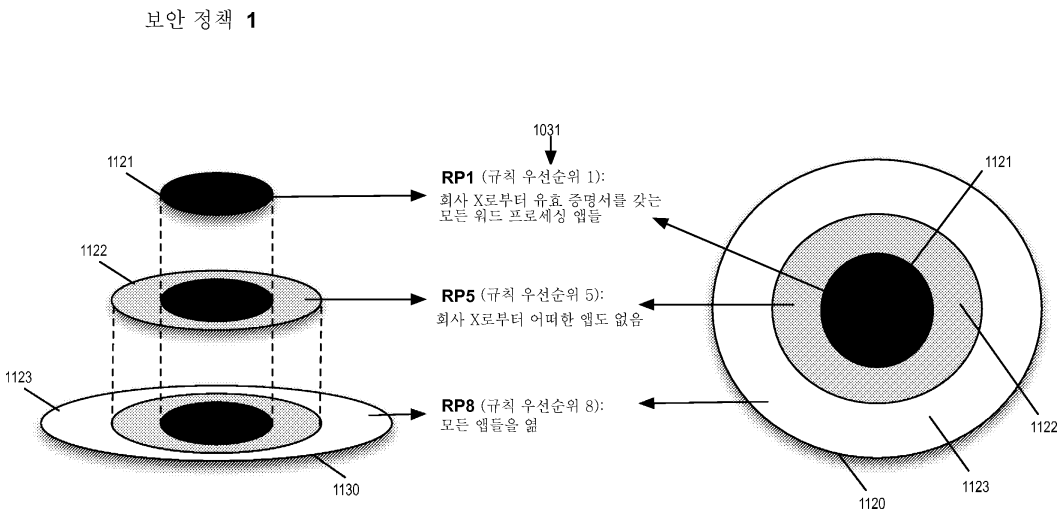
도면9



도면10

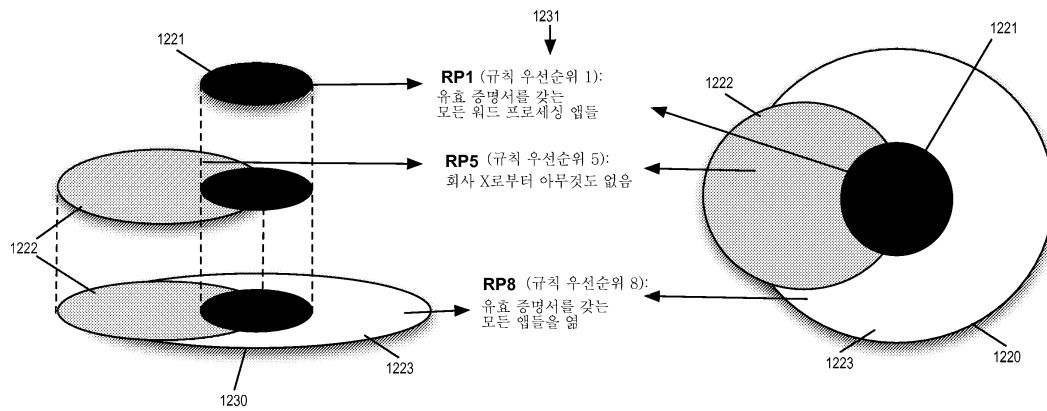


도면11

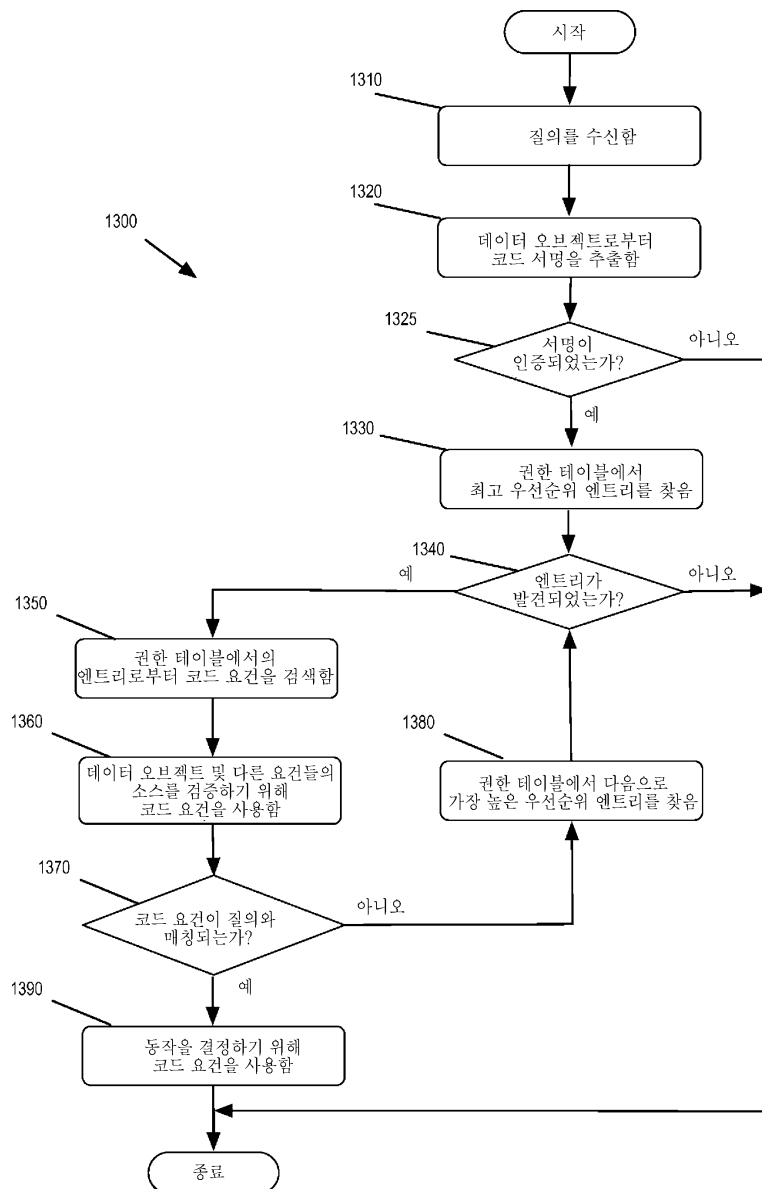


도면12

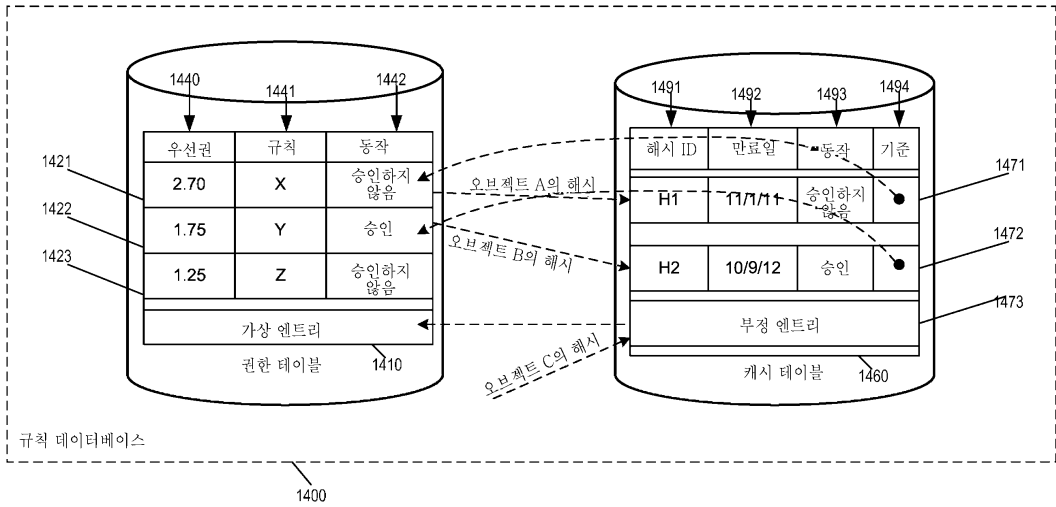
보안 정책 2



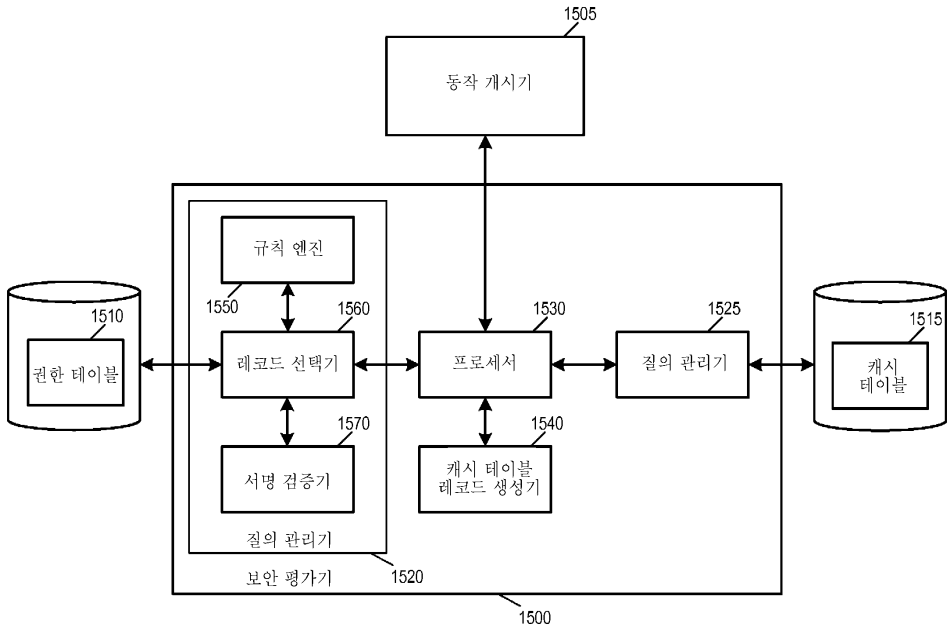
도면13



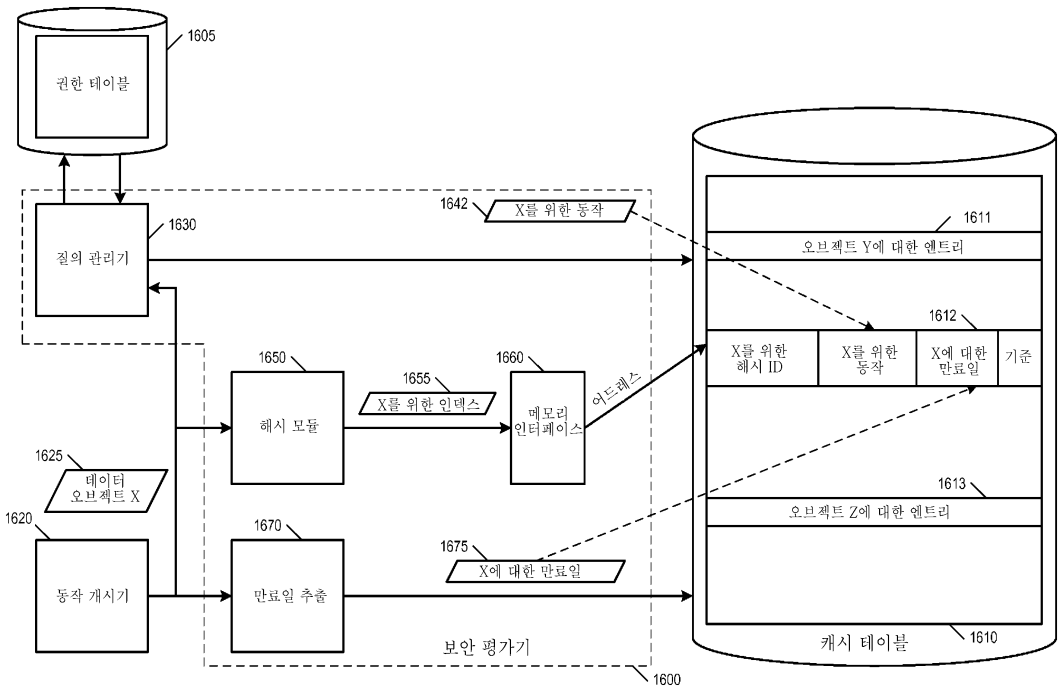
도면14



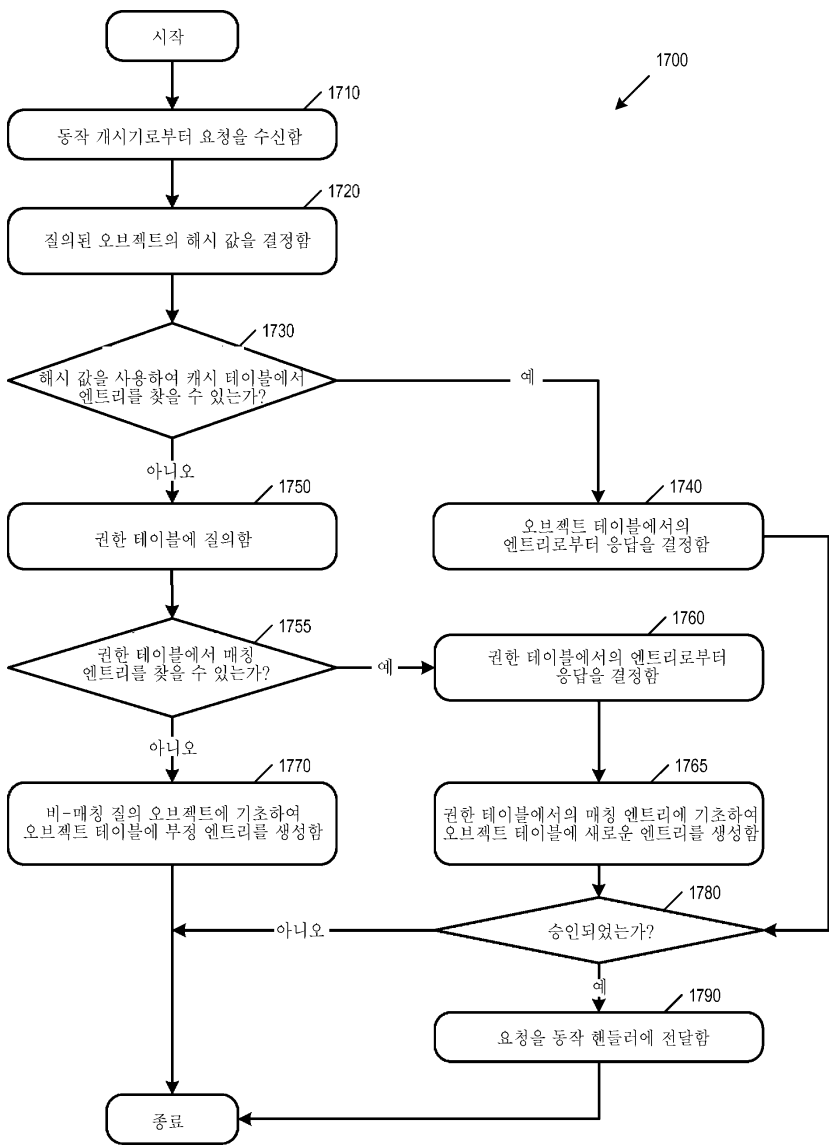
도면15



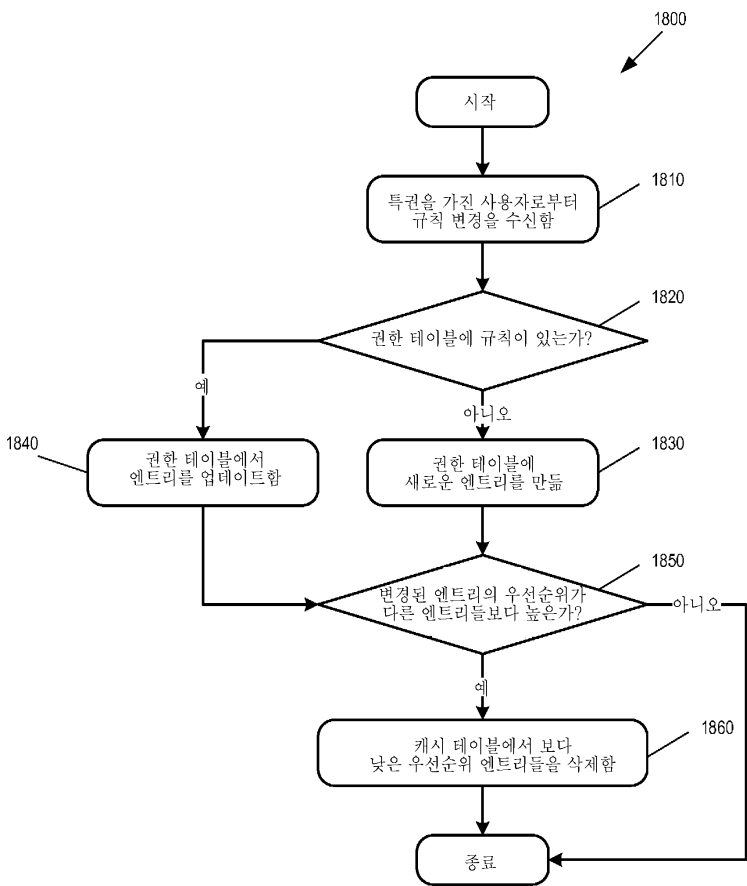
도면16



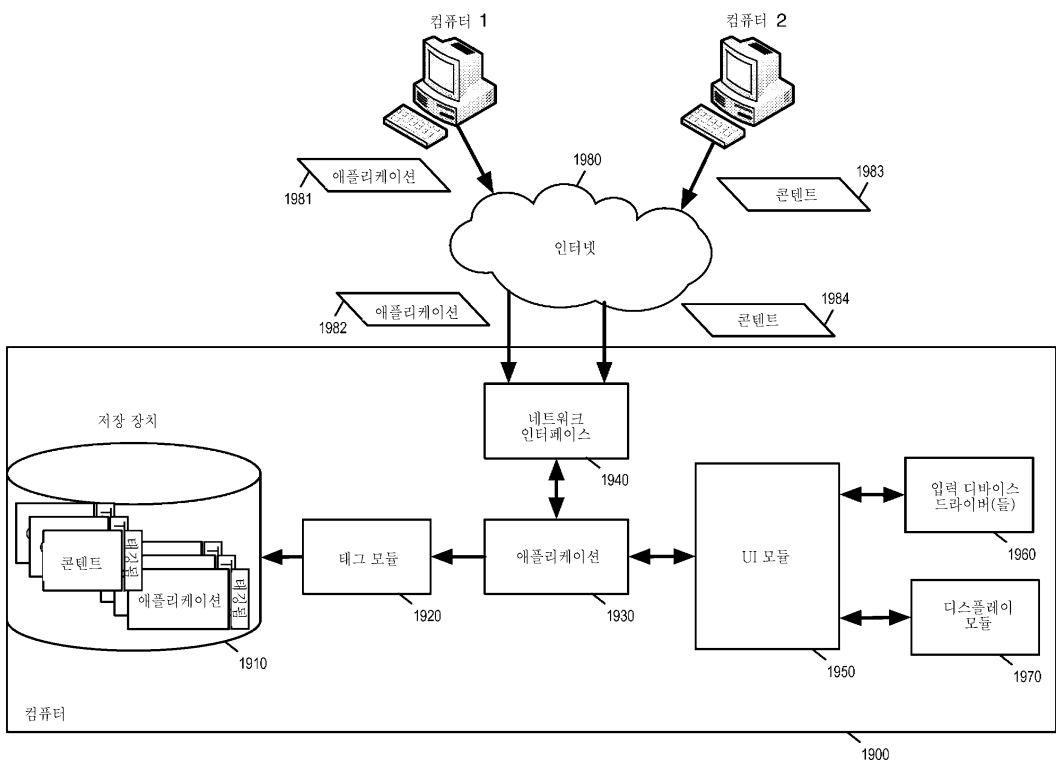
도면17



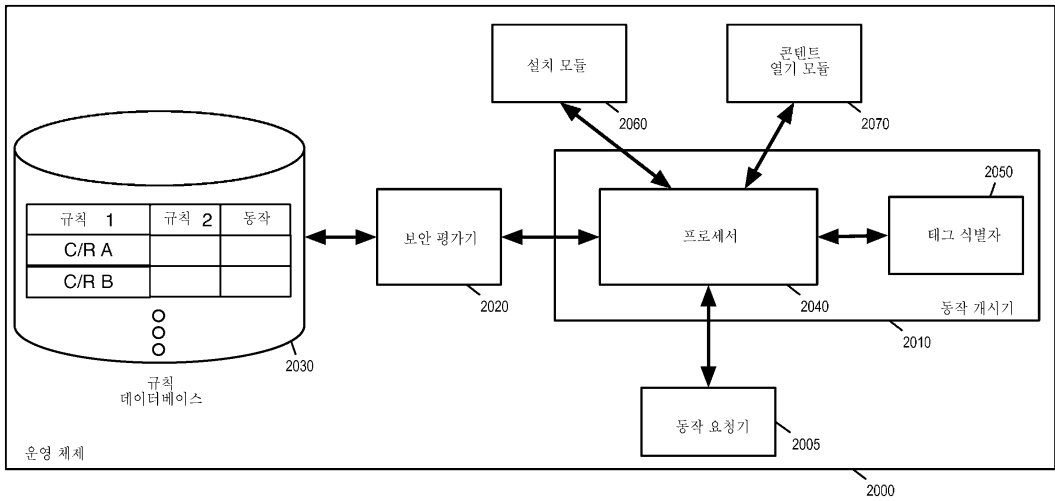
도면18



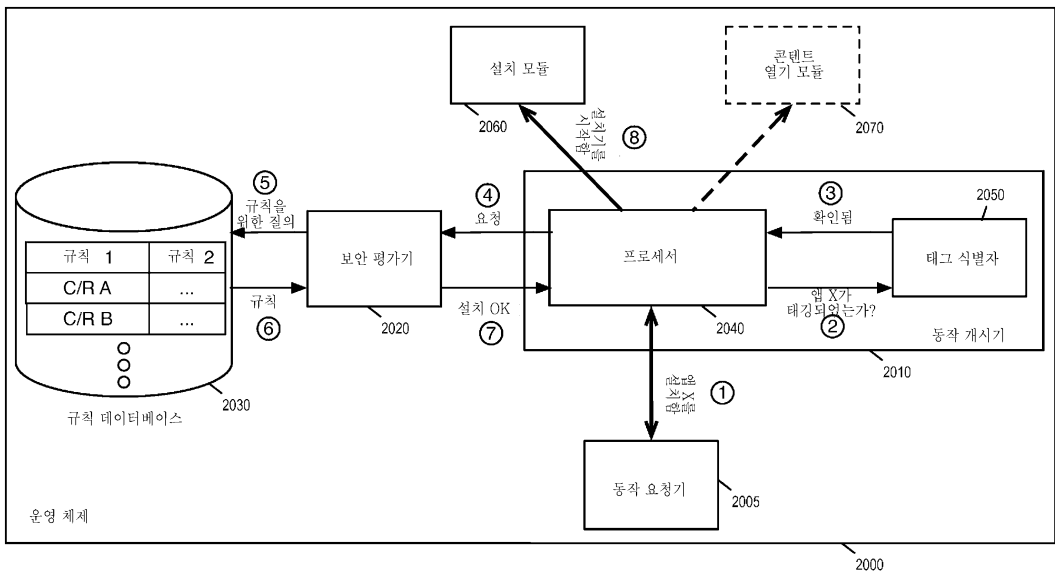
도면19



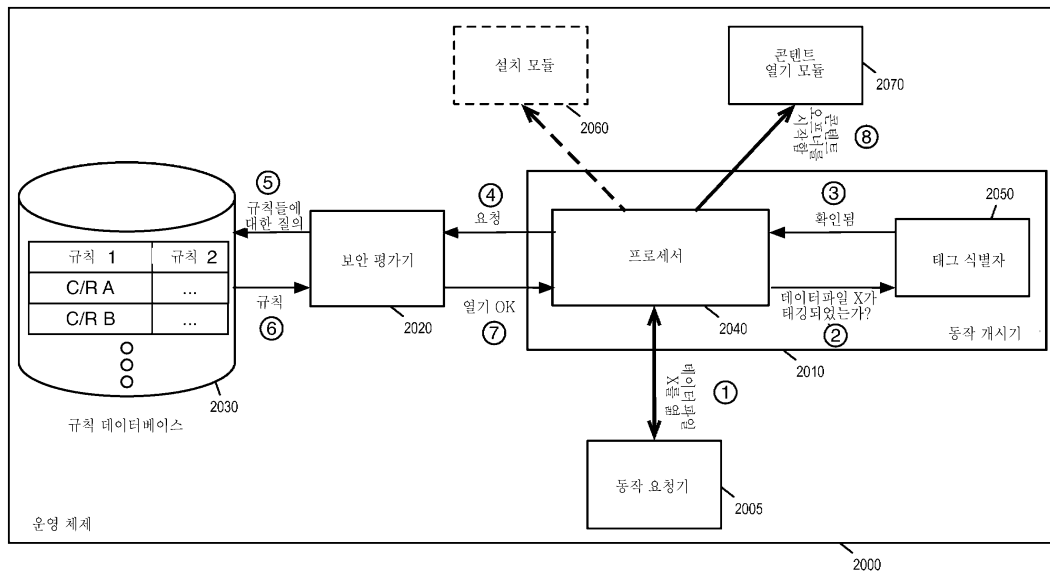
도면20



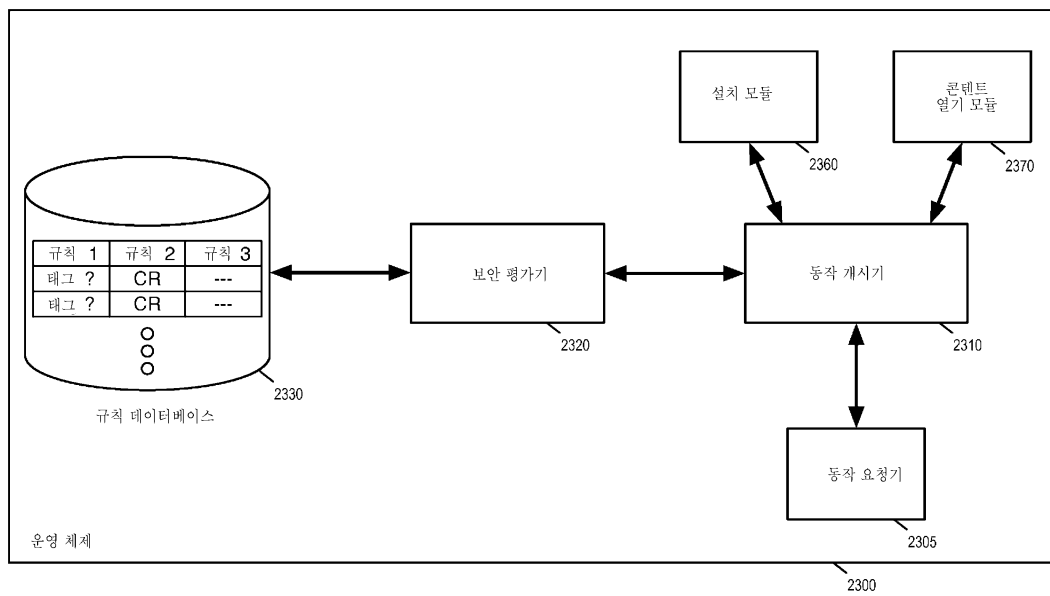
도면21



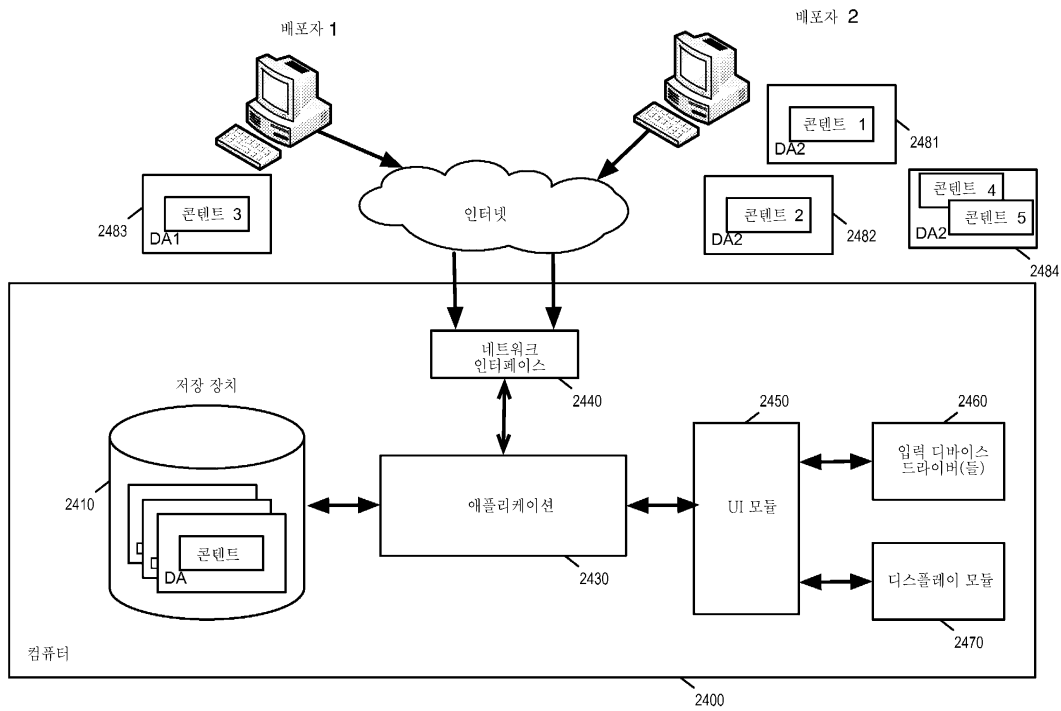
도면22



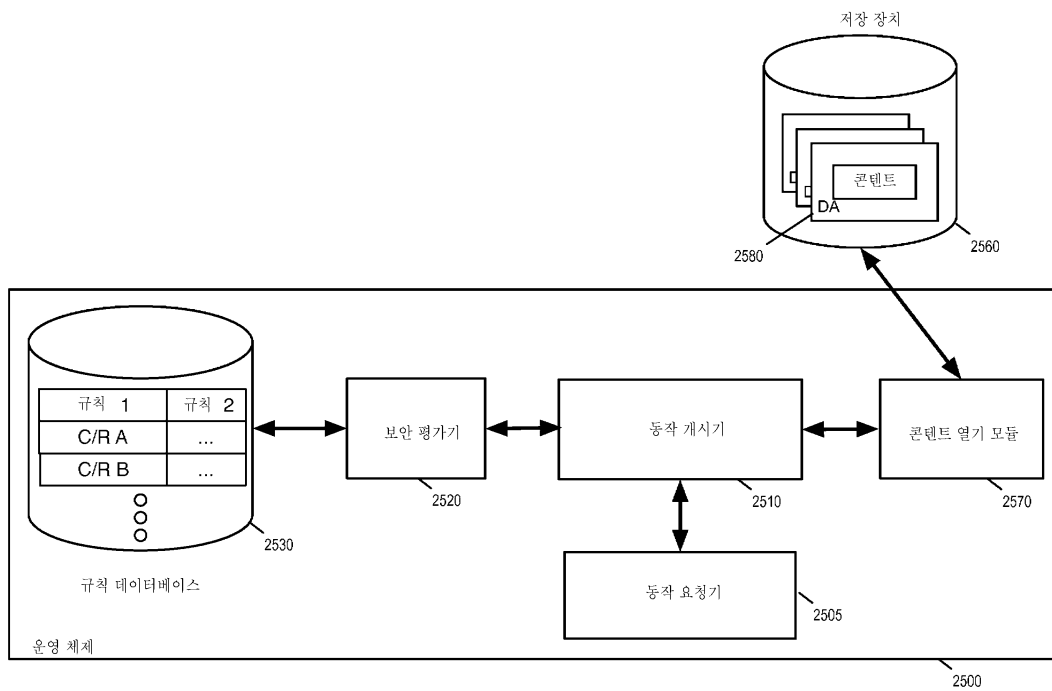
도면23



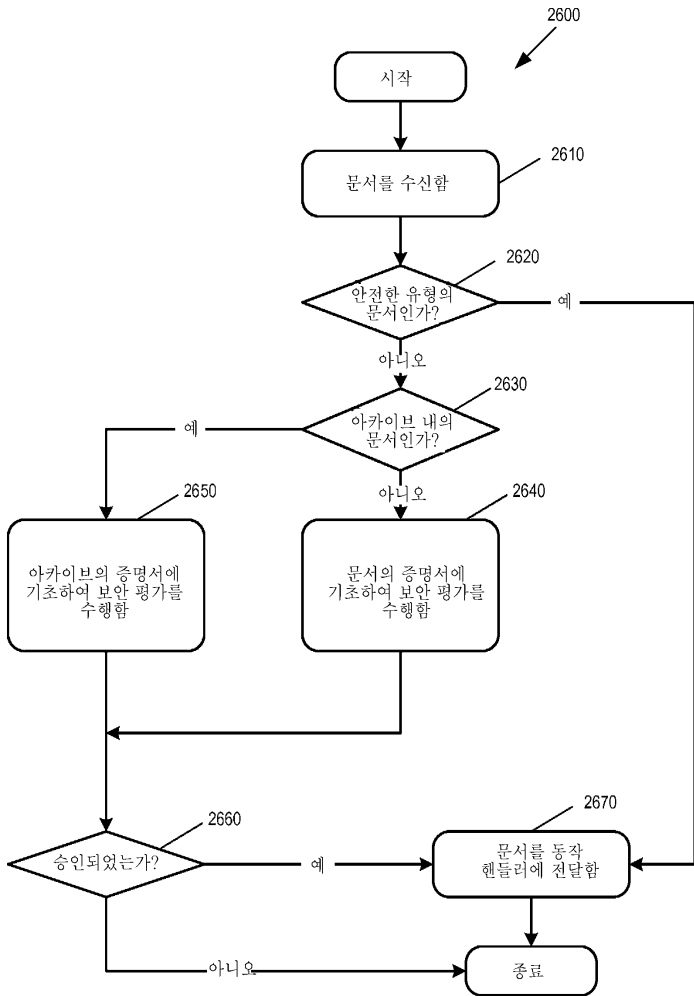
도면24



도면25



도면26



도면27

