(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년07월13일
(11) 등록번호 10-2133606
(24) 등록일자 2020년07월07일

(51) 국제특허분류(Int. Cl.)
G06F 21/60 (2013.01) *G06F 21/30* (2013.01)
(21) 출원번호 10-2014-7019250
(22) 출원일자(국제) 2013년01월10일
심사청구일자 2017년12월06일
(85) 번역문제출일자 2014년07월11일
(65) 공개번호 10-2014-0109952
(43) 공개일자 2014년09월16일
(86) 국제출원번호 PCT/US2013/020913
(87) 국제공개번호 WO 2013/106492
국제공개일자 2013년07월18일
(30) 우선권주장
13/350,360 2012년01월13일 미국(US)
(56) 선행기술조사문헌
JP2000267565 A*
JP2010524410 A*
JP4489712 B2*
KR100979576 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 테크놀로지 라이센싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
렝가나탄 벤카타라만
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
카버 브라이언 토마스
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
(뒷면에 계속)
(74) 대리인
제일특허법인(유)

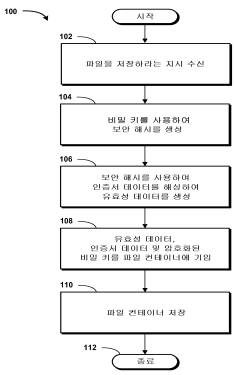
전체 청구항 수 : 총 20 항 심사관 : 구대성

(54) 발명의 명칭 **유효하지 않은 에스크로 키의 검출 기법**

(57) 요 약

보안 해시(예, HMAC(Hash-based Message Authentication Code))가 비밀 정보 조각(예, 비밀 키) 및 각각의 에스크로 키에 특정된 공개 정보 조각(예, 인증서 해시 또는 공개 키)을 사용하여 생성된다. 비밀 키를 사용하는 것은 에스크로 키 유효성 데이터가 비밀 키를 알고 있는 것에 의해서 생성될 수 있다는 것을 보증하고, 이는 공격자가 적합한 에스크로 키 유효성 데이터를 생성하지 못하게 막는다. 인증서 해시 또는 공개 키를 공개 데이터로 사용하는 것은 각각의 에스크로 키 유효성 데이터를 특정한 인증서로 연결하고, 이로써 공격자가 다른 에스크로 키로부터의 유효성 데이터를 간단히 복사하는 것을 방지한다. 유효하지 않은 것으로 발견된 임의의 에스크로 키는 파일 컨테이너로부터 제거될 수 있고, 시스템 감사 로그(system audit log)가 생성될 수 있어 회사, 개인 또는 다른 엔티티가 가능한 보안 침해 시도를 인지할 수 있게 된다.

대 표 도



(72) 발명자

점프 다니엘 브라운

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

레블랭크 데이비드 찰스

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

웨이스 사무엘 아이라

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

명 세 서

청구범위

청구항 1

에스크로 키 유효성 데이터(escrow key validation data)를 생성하는 컴퓨터 구현 방법으로서,

컴퓨터를 사용하여, 파일을 저장하라는 지시(instruction)를 수신하는 단계와,

상기 컴퓨터를 사용하여, 상기 파일을 암호화하는 데 사용되는 비밀 키를 사용하여 보안 해시를 생성하는 단계와,

상기 컴퓨터를 사용하여, 상기 보안 해시를 사용하여 공개 키를 해싱하여 에스크로 키(escrow key)에 대한 유효성 데이터를 생성하는 단계 - 상기 유효성 데이터는 상기 파일에 대해 수행되는 후속 동작 동안 상기 에스크로 키를 검증하고 상기 에스크로 키가 유효하다고 판정되면 상기 파일 동작을 수행하기 위한 유효성 동작의 일부로서 사용하기 위해 생성됨 - 와,

상기 컴퓨터를 사용하여, 상기 유효성 데이터, 상기 공개 키 및 상기 비밀 키를 저장하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 2

제1항에 있어서,

상기 보안 해시는 HMAC(Hash-based Message Authentication Code)인

컴퓨터 구현 방법.

청구항 3

제1항에 있어서,

상기 파일에 대해 수행되는 후속 동작은 열기 동작, 재저장 동작 또는 백그라운드 작업(background task)인

컴퓨터 구현 방법.

청구항 4

제3항에 있어서,

상기 후속 동작은 상기 컴퓨터 상에서 수행되는

컴퓨터 구현 방법.

청구항 5

제3항에 있어서,

상기 후속 동작은 또 다른 컴퓨터 상에서 수행되는

컴퓨터 구현 방법.

청구항 6

제1항에 있어서,

상기 파일을 열라는 지시를 수신하는 단계와,

상기 파일을 열라는 지시의 수신에 응답하여, 비밀 키를 해독하기 위한 사용자 입력을 요청하는 단계와,

사용자 입력을 수신하는 단계와,

상기 사용자 입력이 유효한지 판정하는 단계와,

상기 사용자 입력이 유효하다는 판정에 응답하여,

상기 비밀 키를 해독하는 단계와,

상기 비밀 키를 사용하여 추가 보안 해시를 생성하는 단계와,

상기 추가 보안 해시를 사용하여 상기 공개 키를 해싱하여 상기 에스크로 키에 대한 추가 유효성 데이터를 생성하는 단계와,

상기 유효성 데이터를 상기 추가 유효성 데이터에 비교하는 단계와,

상기 유효성 데이터가 상기 추가 유효성 데이터와 일치하면, 상기 에스크로 키가 유효하다고 판정하고 상기 파일을 여는 단계와,

상기 유효성 데이터가 상기 추가 유효성 데이터와 일치하지 않으면, 상기 에스크로 키가 유효하지 않다고 판정하고 유효하지 않은 상기 에스크로 키를 상기 파일에서 삭제하는 단계를 더 포함하는

컴퓨터 구현 방법.

청구항 7

제6항에 있어서,

상기 유효하지 않은 에스크로 키를 기록(log)하는 단계를 더 포함하는

컴퓨터 구현 방법.

청구항 8

제7항에 있어서,

상기 유효하지 않은 에스크로 키를 기록하는 단계는 시스템 감사정보(system audit)를 생성하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 9

제1항에 있어서,

상기 에스크로 키에 대한 솔트 값(salt value)을 생성하는 단계와,

상기 솔트 값을 상기 보안 해시에 추가하는 단계를 더 포함하는

컴퓨터 구현 방법.

청구항 10

제1항에 있어서,

상기 유효성 데이터, 상기 공개 키 및 상기 비밀 키를 저장하는 단계는 상기 유효성 데이터, 상기 공개 키 및 상기 비밀 키를 상기 파일에 저장하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 11

제1항에 있어서,

상기 유효성 데이터, 상기 공개 키 및 상기 비밀 키를 저장하는 단계는 상기 유효성 데이터, 상기 공개 키 및 상기 비밀 키를 상기 파일을 포함하는 파일 컨테이너에 저장하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 12

에스크로 키를 검증하는 컴퓨터 구현 방법으로서,

컴퓨터를 사용하여, 파일에 대한 동작을 수행하라는 지시를 수신하는 단계 - 상기 파일은 상기 파일과 연관된 에스크로 키에 대한 유효성 데이터를 가지고 있고, 상기 유효성 데이터는 보안 해시를 사용하여 인증서 데이터를 해싱함으로써 생성되며, 상기 보안 해시는 상기 파일에 대해 수행되는 저장 동작 동안 상기 파일을 암호화하는 데 사용되는 비밀 키로부터 생성되었음 - 와,

상기 파일에 대한 동작을 수행하라는 지시의 수신에 응답하여, 상기 컴퓨터를 사용하여, 상기 비밀 키의 해독을 위한 사용자 입력을 요청하는 단계와,

상기 컴퓨터를 사용하여, 사용자 입력을 수신하는 단계와,

상기 컴퓨터를 사용하여, 상기 사용자 입력이 유효한지 판정하는 단계와,

상기 사용자 입력이 유효하다는 판정에 응답하여,

상기 컴퓨터를 사용하여, 상기 비밀 키를 해독하는 단계와,

상기 컴퓨터를 사용하여, 상기 비밀 키를 사용하여 추가 보안 해시를 생성하는 단계와,

상기 컴퓨터를 사용하여, 상기 추가 보안 해시를 사용하여 상기 인증서 데이터를 해싱하여 상기 에스크로 키에 대한 추가 유효성 데이터를 생성하는 단계와,

상기 컴퓨터를 사용하여, 상기 유효성 데이터를 상기 추가 유효성 데이터에 비교하는 단계와,

상기 유효성 데이터가 상기 추가 유효성 데이터와 일치하면, 상기 컴퓨터를 사용하여, 상기 에스크로 키가 유효하다고 판정하고 상기 파일에 대한 동작을 수행하는 단계와,

상기 유효성 데이터가 상기 추가 유효성 데이터와 일치하지 않으면, 상기 컴퓨터를 사용하여, 상기 에스크로 키가 유효하지 않다고 판정하고 유효하지 않은 상기 에스크로 키를 상기 파일과의 연관(association)으로부터 제거하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 13

제12항에 있어서,

상기 보안 해시 및 상기 추가 보안 해시는 HMAC(Hash-based Message Authentication Code)인

컴퓨터 구현 방법.

청구항 14

제12항에 있어서,

상기 동작은 열기 동작, 재저장 동작 또는 백그라운드 작업인

컴퓨터 구현 방법.

청구항 15

제12항에 있어서,

상기 유효하지 않은 에스크로 키를 기록(log)하는 단계를 더 포함하는

컴퓨터 구현 방법.

청구항 16

제15항에 있어서,

상기 유효하지 않은 에스크로 키를 기록하는 단계는 시스템 감사정보(system audit)를 생성하는 단계를 포함하는

컴퓨터 구현 방법.

청구항 17

제12항에 있어서,

상기 파일에 대한 동작을 수행하라는 지시를 수신하기 전에, 상기 파일이 저장되었고, 상기 파일은 상기 유효성 데이터, 상기 에스크로 키, 상기 인증서 데이터 및 상기 비밀 키를 포함하는

컴퓨터 구현 방법.

청구항 18

제12항에 있어서,

상기 파일에 대한 동작을 수행하라는 지시를 수신하기 전에, 상기 파일이 파일 컨테이너에 저장되었고, 상기 파일 컨테이너는 상기 유효성 데이터, 상기 에스크로 키, 상기 인증서 데이터 및 상기 비밀 키를 포함하는

컴퓨터 구현 방법.

청구항 19

컴퓨터 판독가능 명령어를 저장한 컴퓨터 저장 장치로서,

상기 컴퓨터 판독가능 명령어는 컴퓨터에 의해 실행되는 경우에, 상기 컴퓨터로 하여금,

파일을 열라는 지시를 수신하고 - 상기 파일은 상기 파일과 연관된 에스크로 키에 대한 유효성 데이터를 가지고 있고, 상기 유효성 데이터는 HMAC(Hash-based Message Authentication Code)를 사용하여 인증서 데이터를 해싱함으로써 생성되며, 상기 HMAC는 상기 파일에 대해 수행되는 저장 동작 동안 상기 파일을 암호화하는 데 사용되는 비밀 키로부터 생성되었음 -,

상기 파일을 열라는 지시의 수신에 응답하여, 상기 비밀 키의 해독을 위한 인증 자격증명(authentication credential)을 요청하고,

인증 자격증명을 수신하고,

상기 인증 자격증명이 유효한지 판정하고,

상기 인증 자격증명이 유효하다는 판정에 응답하여,

상기 비밀 키를 해독하고,

상기 비밀 키를 사용하여 추가 HMAC를 생성하고,

상기 추가 HMAC를 사용하여 상기 인증서 데이터를 해싱하여 상기 에스크로 키에 대한 추가 유효성 데이터를 생성하고,

상기 유효성 데이터를 상기 추가 유효성 데이터에 비교하고,

상기 유효성 데이터가 상기 추가 유효성 데이터와 일치하면, 상기 에스크로 키가 유효하다고 판정하고 상기 파일을 열라는 지시에 응답하여 상기 파일을 열며,

상기 유효성 데이터가 상기 추가 유효성 데이터와 일치하지 않으면, 상기 에스크로 키가 유효하지 않다고 판정하고 유효하지 않은 상기 에스크로 키를 상기 파일과의 연관(association)으로부터 제거하며, 상기 유효하지 않은 에스크로 키를 기록(log)하게 하는

컴퓨터 저장 장치.

청구항 20

제19항에 있어서,

상기 컴퓨터에 의해 실행되는 경우에 상기 컴퓨터로 하여금

상기 파일을 열라는 지시를 수신하기 전에, 상기 파일을 저장하라는 지시를 수신하고,

상기 파일을 암호화하는 데 사용되는 상기 비밀 키로부터 상기 HMAC를 생성하고,

상기 HMAC를 사용하여 상기 인증서 데이터를 해싱하여 상기 에스크로 키에 대한 유효성 데이터를 생성하며,

상기 유효성 데이터, 상기 인증서 데이터 및 상기 비밀 키를 상기 파일 또는 상기 파일을 포함하는 파일 컨테이너에 기입하고 저장하게 하는

컴퓨터 판독가능 명령어를 더 저장한

컴퓨터 저장 장치.

발명의 설명

기 술 분 야

배 경 기 술

[0001] 워드 프로세싱 애플리케이션, 스프레드시트 애플리케이션 및 프레젠테이션 애플리케이션과 같은 문서 생성 애플리케이션은 때때로 문서 암호화 메커니즘(이를 통해 사용자는 문서에 포함된 정보에 대한 승인되지 않은 액세스를 막도록 문서를 암호화하는 데 사용되는 패스워드를 제공할 수 있음)을 제공한다. 암호화된 문서에 포함되는 정보가 소정의 시점 또는 다른 시점에 패스워드에 대한 권한이 없는(not privy to) 다른 사람에 의해 요구될 수 있다. 예를 들어, 문서가 회사의 고용인에 의해 생성된 패스워드에 의해 암호화되고 고용인이 그 회사를 떠나거나 단순히 패스워드를 잃어버린 경우에, 인적 자원 또는 금융 정보와 같은 중요한 기밀 회사 정보를 포함할 수 있는 문서에 아무도 액세스할 수 없다.

[0002] 패스워드를 알지 못해도 암호화된 문서의 해독을 가능하게 하기 위한 노력으로서, 에스크로 키 매커니즘(escrow key mechanism)이 사용될 수 있다. 에스크로 키 메커니즘은 인증서 기반 키(에스크로 키라 함)를 패스워드 보호 문서에 자동으로 추가하도록 구성가능한 메커니즘이다. 에스크로 키 메커니즘은 패스워드를 알 필요없이, 패스워드 보호 문서가 인증서를 사용하여 암호화되는 것을 가능하게 하여 전술한 바와 같은 경우에 문서 복구 시나리오를 가능하게 한다.

[0003] 암호화된 문서는 흔히 2 단계 시스템을 사용한다. 예를 들어, 문서가 저장될 때마다, 랜덤하게 생성된 비밀 키(secret key)가 사용되어 전체 문서를 암호화한다. 사용자에 의해 제공된 패스워드는 새로운 키를 유도하는 데 사용되고, 새로운 키는 비밀 키를 암호화하는 데 사용된다. 암호화된 비밀 키는 문서에 평문(plain text)으로서 저장될 수 있다. 문서를 해독하기 위해, 키가 사용자 입력 패스워드로부터 유도되고 비밀 키를 해독하는 데 사용되며, 이는 이후에 문서를 해독하는 데 사용된다.

[0004] 에스크로 키 메커니즘은 문서에 대해 구성된 공개 키(인증서에 포함되거나 포함되지 않을 수 있음)를 사용하여 비밀 키를 암호화하는 것에 의해 작동한다. 암호화된 비밀 키(즉, 에스크로 키)는 또한 패스워드 암호화 비밀 키와 함께 문서에 추가된다. 개인 키(private key)를 가진 사람은 누구나 이후에 비밀 키를 해독할 수 있고, 이에 따라 문서를 해독할 수 있다. 비밀 키는 문서가 저장될 때마다 변하기 때문에, 문서 내의 현존하는 에스크로 키는 어느 것이나 새로운 비밀 키로 업데이트될 필요가 있다. 에스크로 키 메커니즘은 보호 문서에 대해 액세스를 획득하려고 하는 공격자에게 매우 취약하다.

[0005] 본 명세서에 기술된 개시내용이 제시하는 것은 이러한 그리고 다른 고려사항에 관한 것이다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0006] 본 명세서에서 유효하지 않은 에스크로 키(예를 들면, 공격자에 의해 파일에 삽입된 것)를 검출하는 개념 및 기법이 설명된다. 예시로서 그리고 전술한 에스크로 키 메커니즘의 사용에 있어서의 잠재적 취약점(potential vulnerability)을 설명하기 위해, 공격자가 패스워드 보호 파일에 대한 액세스를 획득하나 패스워드를 갖고 있지 않고 이에 따라 그 파일을 열 수 없는 시나리오를 고려한다. 공격자는 자신의 에스크로 키를 삽입하기 위해 파일 컨테이너(file container)를 변경할 수 있다. 공격자는 비밀 키를 알지 못하기 때문에, 새롭게 추가된 악성 에스크로 키는 유효하지 않다. 다르게 설명하면, 악성 에스크로 키는 실제 중간 암호 키와 다른 키를 포함한다. 합법적 사용자가 이후에 그 파일을 열고 이후에 재 저장하는 경우에, 파일 컨테이너 내의 에스크로 키는 새로운 에스크로 키로 업데이트될 수 있다. 이러한 업데이트의 결과로, 이제 공격자의 악성 에스크로 키가 원래 포함되었던 맞지 않는 키 대신에 유효한 암호화된 비밀 키(즉, 새로운 비밀 키)를 가진다. 따라서 공격자는 파일을 해독 및 액세스할 수 있고, 이에 따라 에스크로 키 메커니즘을 우회한다.

[0007] 본 명세서에 개시된 개념 및 기법은, 전술한 공격 시나리오를 가능하게 하지 않으면서 파일에 현존하는 에스크로 키가 파일이 저장된 후에 리프레시 및 유지될 수 있게 하는 메커니즘을 제공한다. 인증서 소유자만이 에스크로 키를 해독할 수 있기 때문에 에스크로 키는 유효성을 확인하기 위해 해독될 수 없다. 본 명세서에 기술된 개념 및 기법은 또한, 인증서의 개인 키에 액세스할 필요 없이, 파일 컨테이너의 각각의 에스크로 키를 검증하는 메커니즘을 제공한다.

[0008] 일 측면에서, 보안 해시(예, HMAC(Hash-based Message Authentication Code))가 비밀 정보 조각(예, 비밀 키) 및 각각의 에스크로 키에 특정된 공개 정보 조각(예, 인증서 해시 또는 공개 키)을 사용하여 생성된다. 비밀 키를 사용하는 것은 에스크로 키 유효성 데이터가 비밀 키를 아는 것에 의해서만 생성될 수 있다는 것을 보증하고, 이는 공격자가 적합한 에스크로 키 유효성 데이터를 생성하지 못하게 막는다. 인증서 해시 또는 공개 키를 공개 데이터로 사용하는 것은 각각의 에스크로 키 유효성 데이터를 특정한 인증서로 연결하고, 이로써 공격자가 다른 에스크로 키로부터의 유효성 데이터를 간단히 복사하는 것을 방지한다. 유효하지 않은 것으로 발견된 임의의 에스크로 키는 파일 컨테이너로부터 제거될 수 있고, 시스템 감사 로그(system audit log)가 생성될 수 있어 회사, 개인 또는 다른 엔티티가 가능한 보안 침해 시도를 인지할 수 있게 된다. 문서가 이전의 합법적인 에스크로 키를 업데이트할 필요성을 인지하지 못하는 소프트웨어 버전에 의해 편집되면 유사한 (그러나 악성은 아님) 상태가 발생할 수 있다. 어느 경우에든, 제공된 에스크로 키가 업데이트되어야 하고 그렇지 않으면 검출될 수 있다.

[0009] 다른 측면에서, 파일을 저장하기 위해 수행되는 저장 동작 중에, 유효성 데이터가 파일 컨테이너 내의 각각의 에스크로 키에 추가된다. 이후에, 파일을 열기 위해 수행되는 열기 동작 중에, 유효성 데이터가 생성되고 저장 동작 중에 각각의 에스크로 키에 추가된 유효성 데이터와 비교된다. 일치성(match)이 있는 경우에, 에스크로 키는 유효하다고 판정된다. 그렇지 않으면, 애스크로 키는 유효하지 않은 것으로 판정되고, 에스크로 키가 파일 컨테이너로부터 제거될 수 있고, 기록(log)될 수 있다.

[0010] 전술한 발명의 대상은 컴퓨터로 제어되는 장치, 컴퓨터 프로세스, 컴퓨팅 시스템 또는 컴퓨터 판독가능 저장 매체와 같은 제조 물품으로서 구현될 수 있다는 것을 이해해야 한다. 이러한 그리고 다양한 다른 특징은 다음의 상세한 설명의 판독 및 연관된 도면의 검토에 의해 분명해질 것이다.

[0011] 본 요약은 상세한 설명에서 이하에 추가로 설명되는 개념의 선택사항을 간략한 형태로 소개하기 위해 제공된다. 본 요약은 청구된 발명의 대상의 주요 특징이나 핵심 특징을 식별시키기 위한 것이 아니며, 본 요약은 청구된 발명의 대상의 범주를 제한하는 데 사용하려는 것이 아니다. 또한, 청구된 발명의 대상은 본 명세서의 임의의 부분에 언급된 임의의 또는 모든 문제점을 해결하는 구현예에 한정되지 않는다.

도면의 간단한 설명

[0012] 도 1은 도시된 실시예에 따라 하나 이상의 에스크로 키에 대한 유효성 데이터를 문서에 추가하는 방법의 특징을 나타내는 흐름도이다.

도 2a 및 2b는 도시된 실시예에 따라 하나 이상의 에스크로 키를 검증하는 방법의 특징을 나타내는 흐름도이다.

도 3은 본 명세서에 제시되는 실시예의 특징을 구현할 수 있는 컴퓨터 시스템에 대한 예시적인 컴퓨터 하드웨어 및 소프트웨어 아키텍처를 나타내는 컴퓨터 아이키텍처 도면이다.

발명을 실시하기 위한 구체적인 내용

[0013] 다음의 상세한 설명은 유효하지 않은 에스크로 키를 검출하기 위한 개념 및 기법에 관한 것이다. 본 명세서에 설명된 개념 및 기법의 일 측면에 따르면, 보안 해시(예, HMAC)가 비밀 정보의 조각(예, 비밀 키) 및 각각의 에스크로 키에 특정된 공개 정보 조각(예, 인증서 해시)을 사용하여 생성된다. 비밀 키는 중간 키(intermediate key)에 의해 보호되는 암호화된 정보에 저장될 수 있고, 이는 문서를 해독하도록 승인된 사람에게만 알려진다. 비밀 키를 사용하는 것은 에스크로 키 유효성 데이터가 비밀 키를 알고 있는 것에 의해서만 생성될 수 있다는 것을 보증하고, 이는 공격자가 적합한 에스크로 키 유효성 데이터를 생성하지 못하게 막는다. 인증서 해시 또는 공개 키를 공개 데이터로 사용하는 것은 각각의 에스크로 키 유효성 데이터를 특정한 인증서로 연결하고, 이로써 공격자가 다른 에스크로 키로부터의 유효성 데이터를 간단히 복사하는 것을 방지한다. 유효하지 않은 것으로 발견된 에스크로 키는 어느 것이나 파일 컨테이너로부터 제거될 수 있고, 시스템 감사 로그(system audit log)가 생성될 수 있어 회사, 개인 또는 다른 엔티티가 가능한 보안 침해(security breach) 시도를 인지할 수 있다.

[0014] 본 명세서에 설명된 발명의 대상은 컴퓨터 시스템 상의 운영체제 및 애플리케이션 프로그램의 실행과 함께 실행되는 프로그램 모듈에 대한 일반적인 맥락으로 제시되나, 해당 분야의 기술자는 다른 구현예가 다른 유형의 프로그램 모듈과 함께 수행될 수 있다는 것을 이해할 것이다. 일반적으로, 프로그램 모듈은 루틴, 프로그램, 컴포넌트, 데이터 구조 및 특정한 태스크를 수행하거나 특정한 추상 데이터 유형을 구현하는 다른 유형의 구조를 포함한다. 또한, 본 발명이 속하는 분야의 기술자는 본 명세서에 기술된 발명의 대상이 다른 컴퓨터 시스템 구성(핸드 헬드 장치, 멀티프로세서 시스템, 마이크로프로세서 기반의 또는 프로그램가능한 가전기기, 미니컴퓨터, 메인프레임 컴퓨터 등을 포함함)으로 실현 실현될 수 있다는 것을 이해할 것이다.

[0015] 다음의 상세한 설명에서, 본 명세서의 일부를 구성하며 특정한 실시예 또는 예시를 설명하기 위해 도시된 첨부된 도면에 대한 참조가 이루어진다. 지금부터 도면(도면에서 동일한 번호는 수 개의 도면 전체에서 동일한 구성요소를 나타냄)을 참조하여, 컴퓨팅 시스템, 컴퓨터 판독가능 저장 매체 및 유효하지 않은 에스크로 키를 검출하기 위해 컴퓨터로 구현되는 방법론의 여러 측면이 제시될 것이다.

[0016] 이제, 도 1를 참조하여, 하나 이상의 에스크로 키에 유효성 데이터를 추가하는 방법(100)의 측면들이 상세히 설명될 것이다. 본 명세서에 기술된 방법의 동작은 반드시 임의의 특정한 순서로 제시될 필요는 없으며 대안적인 순서로 동작의 일부 또는 전부를 수행하는 것이 가능하며 고려될 수 있다. 동작은 설명 및 예시를 쉽게 하기 위해 기술된 순서로 제시되었다. 첨부된 청구범위의 범주를 벗어나지 않는 범위에서, 동작이 추가, 삭제 및/또는 동시에 수행될 수 있다.

[0017] 또한, 설명된 방법이 임의의 시점에 종료될 수 있고 이들이 각각 완전하게 수행될 필요는 없다는 것을 이해해야 한다. 방법의 일부 또는 모든 동작 및/또는 실질적으로 등가인 동작이 이하에 정의된 컴퓨터 저장 매체 상에 포함되는 컴퓨터 판독가능 명령어를 실행함으로써 수행될 수 있다. 상세한 설명 및 청구범위에서 사용된 "컴퓨터 판독가능 명령어"라는 용어 및 이들의 변형은 루틴, 애플리케이션, 애플리케이션 모듈, 프로그램 모듈, 프로그램, 컴포넌트, 데이터 구조, 알고리즘 등을 포함하도록 본 명세서에서 폭넓게 사용된다. 컴퓨터 판독가능 명령어는 다양한 시스템 구성(단일 프로세서 또는 멀티프로세서 시스템, 미니컴퓨터, 메인프레임 컴퓨터, 개인 컴퓨터, 핸드헬드 컴퓨팅 장치, 프로세서 기반의 프로그램 가능한 가전 기기, 이들 조합 등을 포함함)으로 구현될 수 있다.

[0018] 따라서, 본 명세서에 기술된 로직 동작(logical operations)은 (1) 컴퓨팅 시스템상에서 실행되는 일련의 컴퓨터 구현 동작 또는 프로그램 모듈로서 및/또는 (2) 컴퓨팅 시스템 내에서 상호연결된 머신 로직 회로 또는 회로 모듈로서 구현되는 것으로 이해되어야 한다. 구현은 컴퓨팅 시스템의 성능 및 다른 필요조건에 의존하는 선택의 문제이다. 따라서, 본 명세서에서 기재된 로직 동작은 동작, 구조 장치, 액트, 또는 모듈로써 다양하게 지칭된다. 이들 동작, 구조 장치, 액트 및 모듈은 소프트웨어, 펌웨어, 전용 디지털 로직, 및 그들의 임의의 조합으로 구현될 수 있다.

[0019] 방법(100)은 파일을 저장하기 위해 실행되는 저장 동작 중에 수행되는 것으로 설명된다. 본 명세서에서 이하에 기술된 특징은 애플리케이션, 애플리케이션 유형, 파일 또는 파일 유형으로 반드시 특정될 필요는 없다. 일부

실시예에서, 애플리케이션은 제1 상태에서 파일을 열고, 사용자 입력(예, 편집이나 다른 상호작용)을 수신하며, 제1 상태와 다른 제2 상태에서 파일을 저장하도록 구성된다. 다른 실시예에서는, 파일이 저장되나 파일 데이터에 아무런 변경이 이루어지지 않는다. 일부 실시예에서, 파일은 메타데이터, 포매팅 파라미터(formatting parameter) 또는 파일 데이터가 아닌 파일 내에 포함된 다른 데이터에 대한 변경과 함께 저장된다. 일부 실시예에서, 애플리케이션이 파일을 저장하도록 구성되나, 열기 및 저장을 넘어선 동작을 수행할 능력(capability)을 갖거나 갖지 않을 수 있다. 예를 들어, 일부 실시예에서 애플리케이션은 임의의 방식으로 파일을 편집하도록 구성될 수 없다. 일부 실시예에서, 저장 동작은 복사 및 붙이기 동작을 포함하고, 제1 파일이 복사되고 붙이기 되며 이로써 새로운 파일이 생성되고 저장한다.

[0020] 일부 실시예에서, 파일은 문서(예, 워드 프로세서 문서, 스프레드시트 문서, 프레젠테이션 문서, 드로잉 문서 또는 협업 문서)이다. 일부 실시예에서, 그러한 문서를 생성 및/또는 보기(view)하는 데 사용되는 애플리케이션은 각각, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT WORD와 같은 워드 프로세싱 애플리케이션, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT EXCEL과 같은 스프레드시트 애플리케이션, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT POWERPOINT와 같은 프레젠테이션 애플리케이션, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT VISIO와 같은 드로잉 애플리케이션, 또는 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT SHAREPOINT와 같은 협업 애플리케이션이다. 일부 실시예에서, 애플리케이션은 사용자로 하여금 보기를 허용하나 문서를 편집하는 것은 허용하지 하도록 구성되는 리드 온리 애플리케이션이다. 다른 실시예에서, 애플리케이션은 사용자로 하여금 문서를 보기 및 편집하는 것을 허용하도록 구성되는 판독/기입 애플리케이션이다. 애플리케이션은 컴퓨터 시스템 상에 로컬로 설치된 단독 애플리케이션(stand-alone application), 컴퓨터 시스템에 의해 원격으로 액세스되는 원격 시스템에 설치된 원격 애플리케이션 또는 웹 애플리케이션일 수 있다. 다른 문서 유형 및 관련 애플리케이션이 고려된다.

[0021] 또한, 방법(100)은 하나 이상의 에스크로 키를 파일에 관해 기술된다. 선택적으로 또는 추가로, 파일을 포함하는 파일 컨테이너가 하나 이상의 에스크로 키를 포함할 수 있다. 일부 에스크로 키는 파일 컨테이너에 저장될 수 있는 반면에 다른 에스크로 키는 파일 자체에 저장될 수 있다. 파일 컨테이너는 컨테이너 또는 래퍼 메타 파일 포맷(container or wrapper meta-file format)이고, 이의 상세내역(specification)은 서로 다른 데이터 요소 및 메타데이터가 파일에 어떻게 공존하는지를 기술한다.

[0022] 방법(100)은 단계(102)에서 시작하고, 여기서 파일을 저장하라는 지시(instruction)가 수신된다. 파일을 저장하라는 지시는 그래픽 유저 인터페이스(GUI) 또는 파일을 저장하도록 구성되는 애플리케이션에 의해 또는 파일 저장을 위해 제시되는 다른 사용자 인터페이스를 통해 수신될 수 있다. GUI는 메뉴나 메뉴의 일부, 아이콘, 리본 인터페이스(ribbion interface)나 리본 인터페이스의 일부, 팝업 GUI, 이들의 일부 조합 등으로서 표현될 수 있다. 파일을 저장하라는 지시는 인터페이스 장치(예, 키보드, 키패드, 마우스, 게임패드, 원격 제어 장치)를 통해 또는 임의의 다른 인터페이스 장치(이는 하나 이상의 버튼, 터치스크린, 터치패드, 마이크로폰, 또는 이를 통해 사용자가 파일을 저장하라고 애플리케이션에 지시하는 메커니즘을 제공하는 다른 맨-머신 인터페이스)를 통해 수신될 수 있다. 이러한 인터페이스 장치 중 하나 이상 장치상의 전용의 또는 프로그램된 물리적 저장 버튼이 고려된다.

[0023] 방법(100)의 나머지 동작이 동작(102)에서 수신된 파일을 저장하라는 지시에 의해 시작되는 저장 동작 중에 수행된다. 방법(100)은 동작(102)으로부터 동작(104)으로 진행하고, 여기서 보안 해시가 비밀 키를 사용하여 생성된다. 일부 실시예에서, 비밀 키는 랜덤하게 또는 유사 랜덤하게 생성되고, 파일을 암호화하는 데 사용된다. 일부 실시예에서, 암호화된 비밀 키는 평문으로 파일에 저장된다. 일부 실시예에서, 암호화된 비밀 키가 평문(plain text)으로 텍스트에 저장된다. 대안으로, 암호화된 비밀 키가 일부 다른 형식으로 파일에 저장될 수 있다.

[0024] 일부 실시예에서, 동작(104)에서 생성된 보안 해시는 해시화된 메시지 인증 코드(Hashed Message Authentication Code)(다르게는, 해시화된 메시지 인증 체크섬(Hashed Message Authentication Checksum), 즉, "HMAC"이라 함)이다. 일부 실시예에서, HMAC는 비밀 키를 비밀 정보 및 일부 공개 정보(일부 공개 정보는 HMAC가 생성되고 있는 중인 특정한 에스크로 키에 특정되며, 예를 들면 인증서 데이터 또는 공개 키임)로서 사용하여 생성된다. 대안적인 실시예에서, 비밀 키 및 인증서 데이터(또는 공개 키)의 정규 해시(regular hash)가 HMAC 대신에 사용된다. 정규 해시를 사용하는 실시예의 결과 해시가 암호법적으로 더 약할 수 있으나, 그럼에도 소정의 시나리오에서 정규 해시의 사용으로 인해 애플리케이션을 찾을 수 있다.

[0025] 비밀 키가 각각의 저장 동작 중에 변경되지 않는 경우에 또는 다른 예측불허의 목적으로, 일부 실시예에서 각각의 에스크로 키에 대한 랜덤 솔트(salt) 값이 생성되고, 경우에 따라 해시 또는 HMAC에 포함된다. 이러한 솔트 값은 이후에 에스크로 키 데이터의 나머지 부분과 함께 파일 컨테이너에 저장된다.

[0026] 방법(100)이 동작(104)에서 동작(106)으로 진행하고, 여기서 인증서 데이터는 동작(104)에서 생성된 보안 해시를 사용하여 해시화되어 에스크로 키에 대한 유효성 데이터를 생성한다. 대안적으로, 공개 키는 인증서 데이터 대신에 사용될 수 있다. 이후에, 방법(100)은 동작(108)으로 진행되고, 유효성 데이터, 인증서 데이터 및 암호화된 비밀 키가 파일에 대한 파일 컨테이너 내에 기입된다. 대안적으로, 이러한 데이터의 전부 또는 일부가 직접 파일에 기입된다. 방법(100)은 이어서 동작(110)으로 진행되고, 파일 컨테이너가 저장된다. 방법(100)은 동작(110)에서 동작(112)으로 진행되고, 여기서 방법(100)이 종료된다.

[0027] 이제, 도 2a 및 도 2b를 참조하여, 하나 이상의 에스크로 키를 검증하는 방법(200)의 특징이 상세히 설명될 것이다. 방법(200)은 도 1에 도시된 방법(100)을 참조하여 전술한 저장 동작의 실행 중에 저장된 파일을 열기 위한 열기 동작 중에 수행되는 것으로 설명된다. 본 명세서에서 이하에 설명된 특징들이 반드시 소정의 애플리케이션, 애플리케이션 유형, 파일 또는 파일 유형으로 특정될 필요는 없다. 일부 실시예에서, 애플리케이션은 파일을 열고 파일을 편집하는 것을 허용하도록 구성된다. 일부 실시예에서, 애플리케이션은 파일을 열도록 구성되나 열기를 넘어서는 동작을 수행하는 기능을 갖거나 갖지 않을 수 있다. 예를 들어, 애플리케이션은 일부 실시예에서 임의의 방식으로 파일을 편집하도록 구성될 수 있다. 일부 실시예에서, 방법(200)에 따라 파일을 여는 데 사용되는 애플리케이션은 전술한 방법(100)에 따라 파일을 저장하는 데 사용하는 애플리케이션과 동일하다. 다른 실시예에서는, 이러한 애플리케이션이 서로 다르다. 파일을 저장하는 데 사용되는 컴퓨터 시스템 또는 장치는 파일을 여는 데 사용되는 컴퓨터 시스템 또는 장치와 동일하거나 다를 수 있다.

[0028] 먼저 도 2a를 참조하면, 방법(200)이 시작되고 동작(202)으로 진행되며, 여기서 파일을 열라는 지시가 수신된다. 파일을 열라는 지시에 응답하여, 비밀 키를 해독하기 위한 사용자 입력이 요청된다. 일부 실시예에서, 사용자 입력에 대한 요청은 사용자에게 사용자 입력을 제공하도록 프롬프팅하는 애플리케이션 내에 제시되는 통지(notification)이나, 요청은 대안적 형태를 취할 수 있다. 일부 실시예에서, 요청된 사용자 입력은 인증 자격증명(authentication credential)(예를 들면, 패스워드, 패스코드, 개인 식별 번호(personal identification number), 보안 질문/대답, 패스프레이즈(passphrase), 음성 패스프레이즈, 다른 보안 인증서, 이들의 조합 등이 있으나 이에 한정되는 것은 아님)이다. 일부 실시예에서, 요청되는 사용자 입력은, 다음의 카테고리, 즉 가진 것(something one has), 중요한 것(something one is), 알고 있는 것(something one knows), 행한 것(something one has done) 및 위치한 곳(somewhere one is located) 중 둘 이상으로부터 다중-인자 인증 자격증명에 대한 요청을 포함한다.

[0029] 파일을 열라는 지시가 GUI 또는 파일을 열도록 구성되는 애플리케이션에 의해 또는 파일을 열기 위해 제시되는 다른 사용자 인터페이스를 통해 수신될 수 있다. GUI는 메뉴나 메뉴의 일부, 아이콘, 리본 인터페이스나 리본 인터페이스의 일부, 팝업 GUI나 팝업 GUI의 일부, 이들의 일부 조합 등으로서 표현될 수 있다. 대안적으로, 파일을 열라는 지시는 인터페이스 장치(예, 키보드, 키패드, 마우스, 게임패드, 원격 제어 장치)를 통해 또는 임의의 다른 인터페이스 장치(이는 하나 이상의 버튼, 터치스크린, 터치패드, 마이크로폰, 또는 이를 통해 사용자가 파일을 열라고 애플리케이션에 지시하는 메커니즘을 제공하는 다른 맨-머신 인터페이스)를 통해 수신될 수 있다. 이러한 인터페이스 장치 중 하나 이상의 장치 상의 전용의 또는 프로그램된 물리적 열기 버튼이 고려된다.

[0030] 방법(200)이 동작(202)에서 동작(204)으로 진행되고, 여기서 동작(202)의 사용자 입력에 대한 요청에 응답하는 사용자 입력이 수신된다. 이어서, 방법(200)은 동작(206)으로 진행되고, 여기서 동작(204)에서 수신된 사용자 입력이 유효한지 아닌지 여부에 관한 판정이 이루어진다. 이러한 판정은 예상되는 사용자 입력(예, 예상되는 패스워드 또는 다른 예상되는 인증 자격증명)과 동작(204)에서 수신된 사용자 입력의 비교에 기초하여 이루질 수 있다. 예상되는 사용자 입력은 애플리케이션이 실행 중인 동일한 컴퓨터 또는 장치에 저장될 수 있거나, 예를 들면 동작(204)에서 수신된 사용자 입력을 검증하도록 구성되는 인증 서버 상에 원격으로 저장될 수 있다.

[0031] 동작(206)에서, 사용자 입력이 유효하지 않은 것으로 판정되는 경우에, 방법(200)은 동작(208)으로 진행하고, 여기서 사용자 입력이 유효하지 않다고 나타내는 메시지가 제시된다. 다르게는, 사용자 입력이 유효하지 않다는 것을 나타내는 메시지가 제시되지 않는다. 일부 실시예에서, 애플리케이션은 사용자 입력이 유효하지 않다는 판정이 이루어질 시에 또는 그 후에 애플리케이션이 종료된다. 어느 경우에든, 방법(200)은 이어서 동작

(210)으로 진행되고, 여기서 방법(200)이 종료된다.

[0032] 동작(206)에서, 사용자 입력이 유효하다는 판정이 이루어지면, 방법(200)은 동작(212)으로 진행되고, 여기서 비밀 키가 암호화된다. 방법(200)은 동작(212)에서 동작(214)으로 진행되며, 보안 해시가 비밀 키를 사용하여 생성된다. 일부 실시예에서, 동작(214)에서 생성된 보안 해시는 HMAC이다. 일부 실시예에서, HMAC는 비밀 키를 비밀 정보 조각 및 일부 공개 정보(이는 HMAC가 생성되고 있는 중인 특정한 에스크로 키에 특정되며, 예를 들면 인증서 데이터 또는 공개 키임)로서 사용하여 생성된다. 대안적인 실시예에서, 비밀 키 및 인증서 데이터(또는 공개 키)의 정규 해시(regular hash)가 HMAC 대신에 사용된다. 정규 해시를 사용하는 실시예의 결과 해시가 암호법적으로 더 약할 수 있으나, 그럼에도 소정의 시나리오에서 정규 해시의 사용으로 인해 애플리케이션을 찾을 수 있다.

[0033] 동작(214)으로부터, 방법(200)이 동작(216)으로 진행되고, 여기서 인증서 데이터는 동작(214)에서 생성된 보안 해시를 사용하여 해시화되어 에스크로 키에 대한 유효성 데이터를 생성한다. 대안적으로, 공개 키가 인증서 데이터 대신에 사용될 수 있다. 이후에 방법(200)은 도 2b(구체적으로 동작(218))로 진행된다. 동작(218)에서는, 동작(216)에서 생성된 유효성 데이터가 도 1의 동작(110)에서 파일 컨테이너에 저장된 유효성 데이터와 비교된다. 방법(200)은 동작(218)에서 동작(220)으로 진행되고, 여기서, 두 개의 유효성 데이터 세트 사이에 일치성이 존재하는지 아닌지 여부에 관한 판정이 이루어진다.

[0034] 동작(220)에서, 유효성 데이터 세트 간에 일치성이 존재하지 않으면, 방법(200)은 동작(222)으로 진행되고, 여기서 에스크로 키가 유효하지 않다고 판정되며, 유효하지 않은 에스크로 키는 파일로부터 제거된다. 이어서, 방법(200)은 동작(224)로 진행되고, 여기서 유효하지 않은 에스크로 키가 기록(log)된다. 일부 실시예에서, 시스템 감사정보(system audit)가 생성되어 회사 또는 파일의 보안에 관심이 있는 다른 엔티티가 가능한 보안 침해 시도를 인지할 수 있게 된다. 대안적으로, 유효하지 않은 에스크로 키는 기록되지 않는다. 어느 경우에든, 방법(200)은 다시 도 2a(구체적으로 동작(210))로 돌아가고, 여기서 방법(200)이 종료된다.

[0035] 다른 실시예(도시되지 않음)에서, 방법(200)은 동작(224)에서 동작(228)으로 진행되고, 여기서 파일이 열린다. 이러한 실시예에서, 파일이 조작되었을 수 있다는 것을 나타내는 경고가 사용자에게 제시될 수 있다. 이어서 방법(200)은 도 2a(구체적으로 동작(210))로 돌아가고, 여기서 방법(200)이 종료된다.

[0036] 동작(220)에서, 유효성 데이터 간에 일치성이 존재하는 것으로 판정되면, 방법(200)이 동작(226)으로 진행되고, 여기서 에스크로 키가 유효하지 않다고 판정된다. 이후에 방법(200)은 동작(228)으로 진행되고, 여기서 파일이 열린다. 방법(200)은 동작(228)에서 다시 도 2a(구체적으로 동작(210))로 돌아가고, 여기서 방법(200)이 종료된다.

[0037] 에스크로 키가 도 2에서 열기 동작 중에 검증되는 것으로 설명되었으나, 선택적으로 에스크로 키는 저장 동작 중에 백그라운드 작업으로서 또는 특정한 사전 지정된 입력에 응답하여 검증될 수 있다는 점을 이해해야 한다.

[0038] 도 3은 유효하지 않은 에스크로 키의 검출에 관해 본 명세서에 설명된 소프트웨어 컴포넌트를 실행할 수 있는 장치에 대한 예시적인 컴퓨터 아키텍처(300)를 나타낸다. 따라서, 도 3에 도시된 컴퓨터 아키텍처(300)는 서버 컴퓨터, 모바일 전화, PDA, 스마트 폰, 데스크톱 컴퓨터, 넷북, 테이블 컴퓨터 및/또는 랩톱 컴퓨터에 대한 아키텍처를 나타낸다. 컴퓨터 아키텍처(300)는 본 명세서에 제시된 소프트웨어 컴포넌트의 임의의 특징을 실행하는 데 사용될 수 있다.

[0039] 도 3에 도시된 컴퓨터 아키텍처(300)는 중앙 처리 장치(302)(CPU), 시스템 메모리(304)(랜덤 액세스 메모리(RAM)(306) 및 리드 온리 메모리(ROM)(308)를 포함함) 및 CPU(302)에 메모리를 연결하는 시스템 버스(310)를 포함한다. 예를 들면, 시동(startup) 동안 컴퓨터 아키텍처(300) 내 요소들 간에 정보를 전달하는 것을 돕는 기본 루틴을 포함하는 기본 입/출력 시스템이 ROM(308)에 저장된다. 컴퓨터 아키텍처(300)는 운영체제(314), 애플리케이션 프로그램(316) 및 데이터(318)를 더 포함한다. 데이터(318)는 하나 이상의 에스크로 키(310), 하나 이상의 비밀 키(322), 하나 이상의 인증서 또는 공개 키(324), 하나 이상의 보안 해시(326), 유효성 데이터(328) 및 파일 데이터(330)(예를 들면, 본 명세서에 전술한 에스크로 키, 비밀 키, 인증서, 보안 해시, 유효성 데이터 및 파일 데이터)를 포함한다.

[0040] CPU(302)는 운영체제(314)를 실행하도록 구성된다. 운영체제(314)는 컴퓨터 아키텍처(300)의 동작을 제어하는 애플리케이션 프로그램이다. 애플리케이션(316)은 본 명세서에 기술된 다양한 기능성을 제공하도록 운영체제(314)의 최상위에서 실행되도록 구성되는 실행가능 프로그램이다. 예를 들어, 애플리케이션(316)은 에스크로 키를 검증하는 것에 관하여 각각 도 1 및 도 2a/2b를 참조하여 전술한 저장 동작 및 열기 동작을 제공할 수 있

다. 일부 실시예에서, 애플리케이션(316)은 제1 상태에서 파일을 열고, 편집 또는 다른 상호작용과 같은 사용자 입력을 수신하며, 제1 상태와 다른 제2 상태에 파일을 저장하도록 구성된다. 다른 실시예에서, 애플리케이션(316)은 파일 데이터(330)에 아무런 변경이 이루어지지 않는 경우에도 파일을 저장하도록 구성된다. 일부 실시예에서, 파일은 메타데이터, 포매팅 파라미터 또는 파일 데이터가 아닌 파일 내에 포함되는 다른 데이터에 대한 변경과 함께 저장된다. 일부 실시예에서, 애플리케이션(316)은 파일을 저장하도록 구성되나, 열기 및 저장을 넘어선 동작을 수행할 능력(capability)을 갖거나 갖지 않을 수 있다. 예를 들어, 일부 실시예에서 애플리케이션(316)은 임의의 방식으로 파일을 편집하도록 구성될 수 없다.

[0041] 일부 실시예에서, 애플리케이션(316)은 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT WORD와 같은 워드 프로세싱 애플리케이션, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT EXCEL과 같은 스프레드시트 애플리케이션, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT POWERPOINT와 같은 프레젠테이션 애플리케이션, 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT VISIO와 같은 드로잉 애플리케이션, 또는 워싱턴 주 레드몬드에 소재한 마이크로소프트 사로부터 입수할 수 있는 MICROSOFT SHAREPOINT와 같은 협업 애플리케이션이다. 선택적으로 또는 추가로, 애플리케이션(316)은 전술한 문서 생성 애플리케이션 중 하나 이상의 웹 기반 버전이고, 이러한 실시예에서, 컴퓨터 아키텍처(300)는 애플리케이션(316)을 웹 기반 애플리케이션으로서 제공하도록 구성되는 서버 컴퓨터의 아키텍처의 기능을 한다.

[0042] 대용량 저장 장치(312)는 버스(310)에 연결된 대용량 저장 컨트롤러(도시되지 않음)를 통해 CPU(302)에 연결된다. 대용량 저장 장치(312) 및 이의 연관된 컴퓨터 판독가능 매체는 컴퓨터 아키텍처(300)에 대한 비휘발성 저장소를 제공한다. 본 명세서에 포함된 컴퓨터 판독가능 매체에 대한 기술내용이 대용량 저장 장치(예, 하드 디스크 또는 CD-ROM 드라이브)를 지칭하지만, 해당 분야의 통상의 기술자라면 컴퓨터 판독가능 매체는 컴퓨터 아키텍처(300)에 의해 액세스될 수 있는 임의의 이용 가능한 컴퓨터 저장 매체 또는 통신 매체일 수 있음을 이해할 것이다.

[0043] 통신 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는, 반송파나 다른 전송 메커니즘과 같은 변조된 데이터 신호의 다른 데이터를 포함한다. "변조된 데이터 신호"라는 용어는 정보를 신호로 인코딩하는 것과 같은 방식으로 자신의 특성이 변경되거나 설정된 신호를 의미한다. 제한이 아닌 예시로서, 통신 매체는 유선 네트워크나 직접 유선 연결과 같은 유선 매체 및 음향, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 포함한다. 전술한 것의 임의의 조합도 또한 컴퓨터 판독가능 매체의 범주에 포함되어야 한다.

[0044] 제한이 아닌 예시로서, 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 다른 데이터를 저장하기 위한 임의의 방법 또는 기법으로 구현된 휘발성 및 비휘발성, 이동식 및 비 이동식 매체를 포함할 수 있다. 예를 들어, 컴퓨터 매체는 RAM, ROM, EPROM, EEPROM, 플래시 메모리 또는 그 밖의 다른 솔리드 스테이트 메모리 기법, CD-ROM, 디지털 다목적 디스크("DVD"), HD-DVD, BLU-RAY, 또는 그 밖의 다른 광학 저장소, 자기 카세트, 자기 테이프, 자기 디스크 저장소 또는 그 밖의 다른 자기 저장 장치, 또는 원하는 정보를 저장하기 위해 사용될 수 있고 컴퓨터 아키텍처(300)에 의해 액세스될 수 있는 임의의 다른 매체를 포함하나, 이에 한정되는 것은 아니다. 청구범위의 목적을 위해, "컴퓨터 저장 매체"라는 문구 및 이들의 변형은 파장, 신호 및/또는 다른 일시적/무형의 통신 매체 등을 포함하지 않는다.

[0045] 다양한 실시예에 따르면, 컴퓨터 아키텍처(300)는 네트워크(332)를 통한 원격 컴퓨터로의 로직 연결을 이용해 네트워크 환경에서 동작할 수 있다. 네트워크(322)는 인터넷(Internet), 인트라넷 또는 엑스트라넷일 수 있다. 해당분야의 기술자라면 네트워크(332)에 대한 액세스가 하나 이상의 유선 또는 무선 액세스 네트워크(도시되지 않음)를 통해 제공될 수 있다는 것을 이해할 것이다.

[0046] 컴퓨터 아키텍처(300)는 버스(310)에 연결되는 네트워크 인터페이스 유닛(314)을 통해 네트워크(332)에 연결될 수 있다. 또한 네트워크 인터페이스 유닛(314)이 사용되어 다른 유형의 네트워크 및 원격 컴퓨터 시스템으로 연결될 수 있다는 것을 이해해야 한다. 또한 컴퓨터 아키텍처(300)는 키보드, 마우스, 전자 스타일러스(도 3에 도시되지 않음) 또는 다른 입력 장치(도 3에 도시되지 않음)를 포함하는 복수의 다른 장치로부터의 입력을 수신하고 처리하기 위한 입/출력 컨트롤러(316)를 더 포함할 수 있다. 마찬가지로, 입/출력 컨트롤러(316)가 디스플레이 스크린, 프린터, 또는 다른 유형의 출력 장치(역시 도 3에 도시되지 않음)로 출력을 제공할 수 있다.

[0047] 본 명세서에 기재된 소프트웨어 구성요소는, CPU(302)로 로딩되고 실행되는 경우에 CPU(302) 및 전체 컴퓨터 아키텍처(300)를 범용 컴퓨팅 시스템에서 본 명세서에 제시된 기능을 촉진시키도록 맞춤화된 전용 컴퓨팅 시스템으로 변환한다는 것을 이해해야 한다. CPU(302)는 개별적으로 또는 집합적으로 임의의 개수의 상태를 가정할

수 있는 임의의 개수의 트랜지스터 또는 다른 개별 회로 요소로부터 구성될 수 있다. 더 구체적으로, CPU(302)는 본 명세서에 개시된 소프트웨어 모듈 내에 포함된 실행 가능한 명령어에 응답하여 유한 상태 머신으로서 동작할 수 있다. 이들 컴퓨터 실행형 명령어는 CPU(302)가 상태들 간 전이하는 방식을 특정화하고, 이에 따라 트랜지스터 또는 CPU(302)를 구성하는 다른 개별 하드웨어 요소를 변환시킴으로써 CPU(302)를 변환시킬 수 있다.

[0048] 또한, 본 명세서에 제공되는 소프트웨어 모듈을 인코딩하는 것을 본 명세서에 제공되는 컴퓨터 판독가능 매체의 물리적 구조를 변환시킬 수 있다. 물리적 구조의 특정 변환은 본 상세한 설명의 서로 다른 구현예에서 다양한 인자에 따라 달라질 수 있다. 이러한 인자들의 비-제한적 예시로는, 컴퓨터 판독가능 매체가 주 저장장치로서 특성화되었는지 보조 저장장치로서 특성화되었는지 등에 무관하게, 컴퓨터 판독가능 매체를 구현하기 위해 사용되는 기술이 포함된다. 예를 들어, 컴퓨터 판독가능 매체가 반도체 기반 메모리로서 구현되는 경우, 반도체 메모리의 물리적 상태를 변환함으로써 본 명세서에 개시되는 소프트웨어는 컴퓨터 판독가능 매체 상에 인코딩될 수 있다. 예를 들어, 소프트웨어는 반도체 메모리를 구성하는 트랜지스터, 커패시터, 또는 그 밖의 다른 이산 회로 요소의 상태를 변환할 수 있다. 또한 소프트웨어는 데이터를 저장하기 위해 이러한 구성요소의 물리적 상태를 변환시킬 수 있다.

[0049] 또 다른 예를 들면, 본 명세서에 개시된 컴퓨터 판독가능 매체판독가능 매체 광학 기법을 이용해 구현될 수 있다. 이러한 구현예에서, 본 명세서에 제공되는 소프트웨어는, 소프트웨어가 자기 또는 광학 매체에 인코딩될 때 상기 자기 또는 광학 매체의 물리적 상태를 변환할 수 있다. 이들 변환은 특정 자기 매체 내 특정 위치의 자기 특성을 변경하는 것을 포함할 수 있다. 또한 이들 변환은 특정 광학 매체 내 특정 위치의 물리적 특징 또는 특성을 변경하여, 이들 위치의 광학 특성을 변화시키는 것을 포함할 수 있다. 물리적 매체의 그 밖의 다른 변환이 본 명세서의 사상과 범위 내에서 가능하며, 전술한 예시는 설명을 용이하게 하기 위한 것에 불과하다.

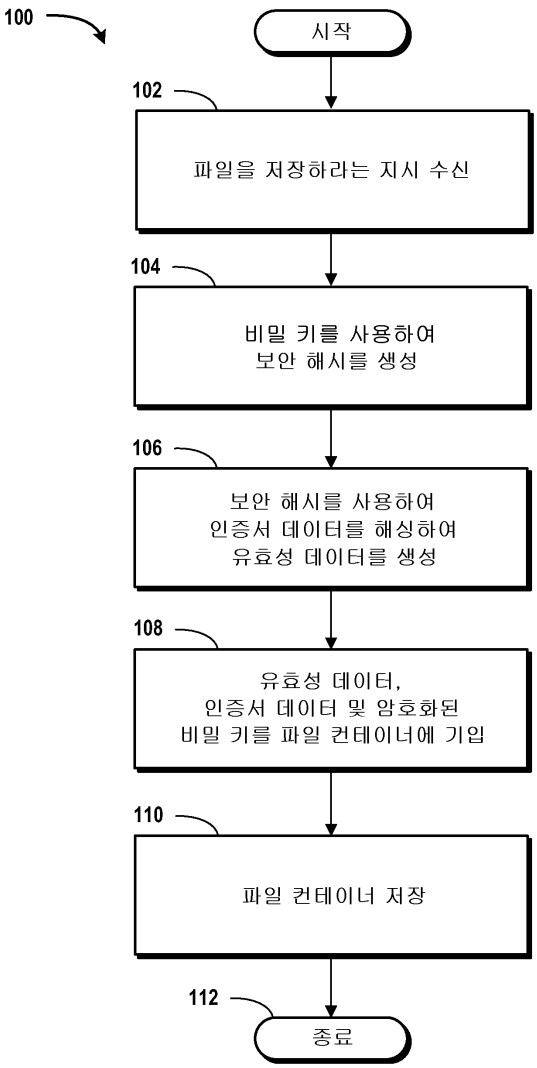
[0050] 이와 관련하여, 본 명세서에 제공되는 소프트웨어 구성요소를 저장 및 실행하기 위해 많은 유형의 물리적 변환이 컴퓨터 아키텍처(300)에서 발생함을 이해해야 한다. 컴퓨터 아키텍처(300)는 그 밖의 다른 유형의 컴퓨팅 장치, 가령, 핸드-헬드 컴퓨터, 임베디드 컴퓨터 시스템, 개인 디지털 보조기(PDA), 및 해당 분야에 알려진 른 유형의 컴퓨팅 장치를 포함할 수 있다. 컴퓨터 아키텍처(300)는 도 3에 도시된 구성요소를 모두 포함하지 않거나, 도 3에 명시적으로 나타나지 않는 그 밖의 다른 구성요소를 포함하거나, 도 3에 도시된 것과 완전히 상이한 아키텍처를 이용할 수 있다.

[0051] 전술한 내용에 기초하여, 유효하지 않은 에스크로 키를 검출하는 기법이 본 명세서에 설명되었다는 것을 이해할 것이다. 본 명세서에 제시된 발명의 대상은 컴퓨터 구조적 특징, 방법론적인 변환가능 액트, 특정한 컴퓨팅 기계 및 컴퓨터 판독가능 매체에 특정된 언어로 설명되었으나, 첨부된 청구범위에 정의된 발명은 본 명세서에 기술된 구체적인 특징, 액트 또는 매체에 한정될 필요가 없다는 것을 이해할 수 있을 것이다. 오히려, 구체적인 특징, 액트 또는 매체는 청구범위를 구현하는 예시적인 형식으로서 기술된 것이다.

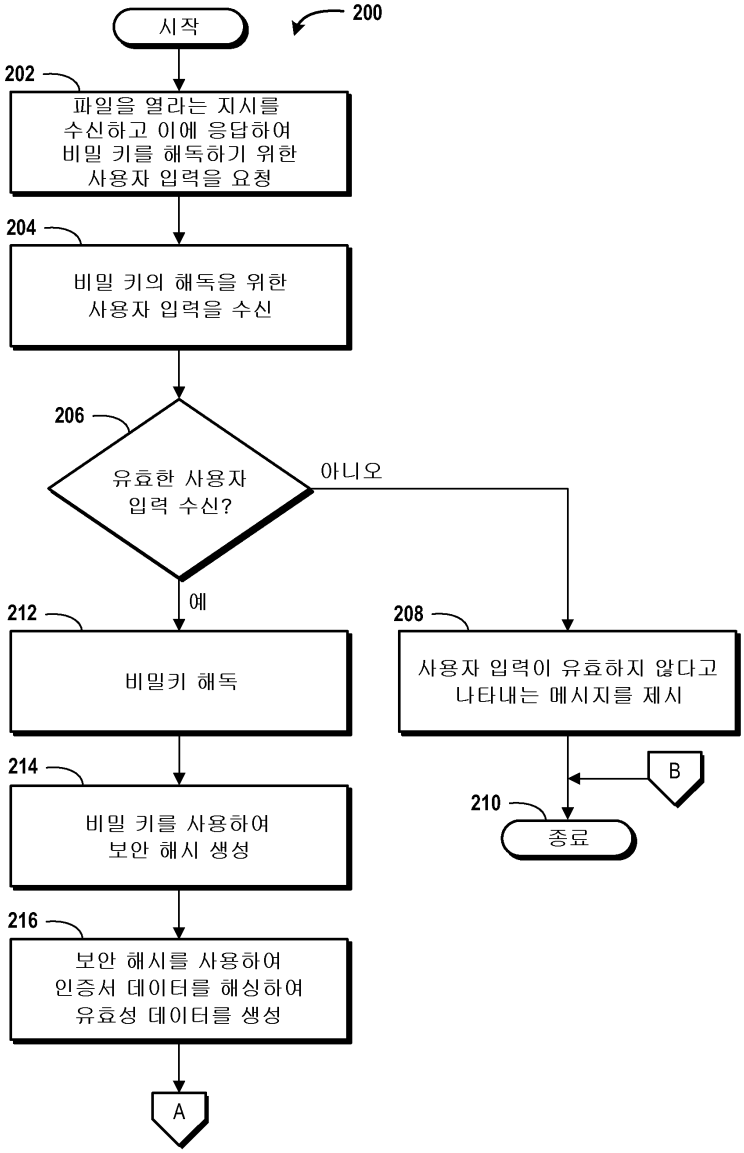
[0052] 전술한 발명의 대상은 단지 예시를 위해 제공된 것이며, 한정하는 것으로 해석해서는 안 된다. 도시되고 기술된 예시적 실시예 및 응용예를 따르지 않고도 이하의 청구범위에 기술된 본 발명의 진정한 사상 및 범주를 벗어나지 않는 범위에서, 본 명세서에 기재된 발명의 대상에 대한 다양한 수정 및 변경이 이루어질 수 있다.

도면

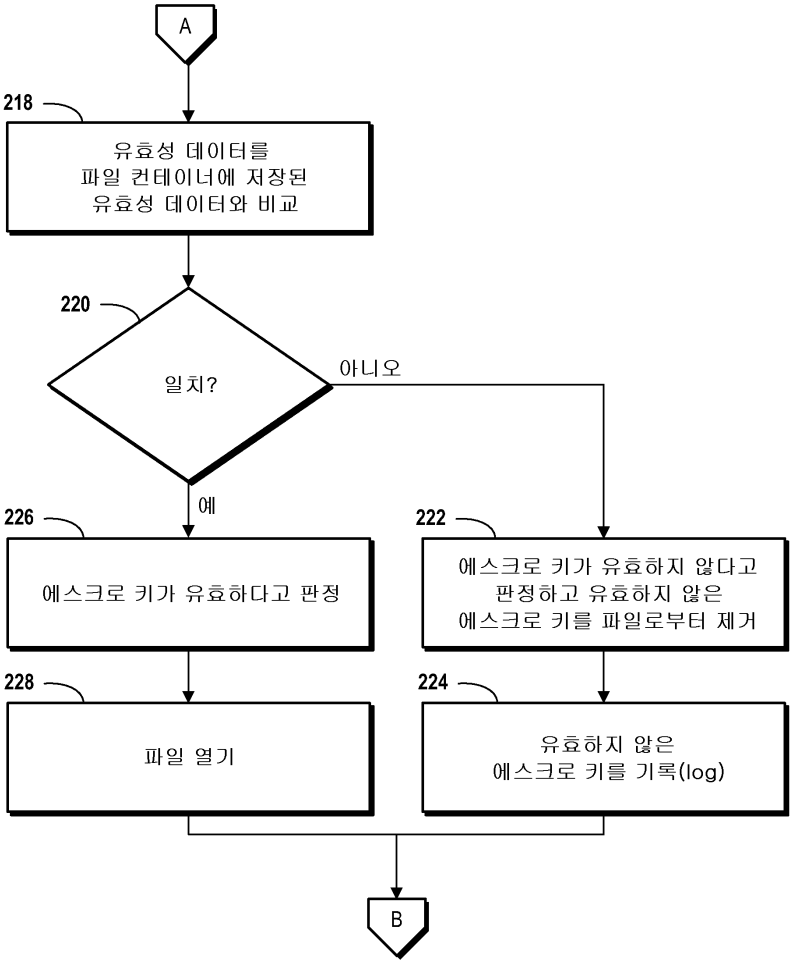
도면1


100
시작
102
파일을 저장하라는 지시 수신
104
비밀 키를 사용하여
보안 해시를 생성
106
보안 해시를 사용하여
인증서 데이터를 해싱하여
유효성 데이터를 생성
108
유효성 데이터,
인증서 데이터 및 암호화된
비밀 키를 파일 컨테이너에 기입
110
파일 컨테이너 저장
112
종료

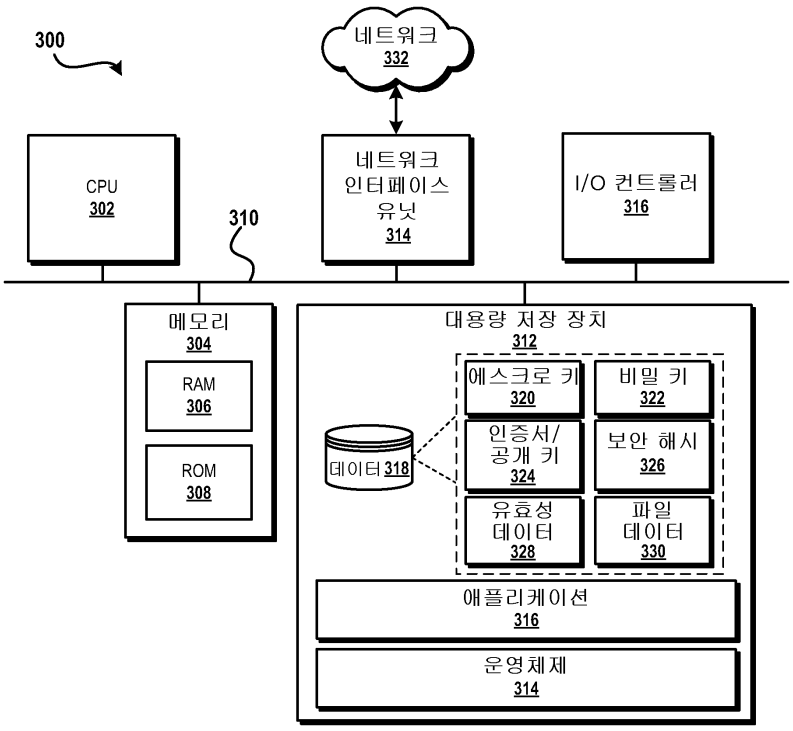
도면2a


시작
200
202
파일을 열라는 지시를 수신하고 이에 응답하여 비밀 키를 해독하기 위한 사용자 입력을 요청
204
비밀 키의 해독을 위한 사용자 입력을 수신
206
유효한 사용자 입력 수신?
아니오
예
212
비밀키 해독
208
사용자 입력이 유효하지 않다고 나타내는 메시지를 제시
B
210
종료
214
비밀 키를 사용하여 보안 해시 생성
216
보안 해시를 사용하여 인증서 데이터를 해싱하여 유효성 데이터를 생성
A

도면2b


A
218
유효성 데이터를
파일 컨테이너에 저장된
유효성 데이터와 비교
220
일치?
아니오
예
226
에스크로 키가 유효하다고 판정
222
에스크로 키가 유효하지 않다고
판정하고 유효하지 않은
에스크로 키를 파일로부터 제거
228
파일 열기
224
유효하지 않은
에스크로 키를 기록(log)
B

도면3


300
네트워크
332
CPU
302
310
네트워크
인터페이스
유닛
314
I/O 컨트롤러
316
메모리
304
RAM
306
ROM
308
대용량 저장 장치
312
데이터 318
에스크로 키
320
비밀 키
322
인증서/
공개 키
324
보안 해시
326
유효성
데이터
328
파일
데이터
330
애플리케이션
316
운영체제
314