(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0070105
(43) 공개일자 2015년06월24일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) *H04L 29/06* (2006.01)
(52) CPC특허분류
G06F 21/577 (2013.01)
H04L 63/0428 (2013.01)
(21) 출원번호 10-2015-7006867
(22) 출원일자(국제) 2013년08월16일
심사청구일자 없음
(85) 번역문제출일자 2015년03월18일
(86) 국제출원번호 PCT/US2013/055449
(87) 국제공개번호 WO 2014/031494
국제공개일자 2014년02월27일
(30) 우선권주장
61/684,743 2012년08월18일 미국(US)
(뒷면에 계속)

(71) 출원인
루미날, 인크.
미국 21701 메릴랜드주 프레데릭 이스트 올 세인츠 스트리트 47
(72) 발명자
스텔라, 조슈아
미국 25443 웨스트 버지니아주 셰퍼즈타운 노스 처치 스트리트 112
집필리, 도미닉
미국 25403 웨스트 버지니아주 마틴스버그 테이블러 스테이션 로드 2183
브링크맨, 메튜
미국 25403 웨스트 버지니아주 마틴스버그 브레번 드라이브 614
(74) 대리인
양영준, 백만기, 정은진

전체 청구항 수 : 총 80 항

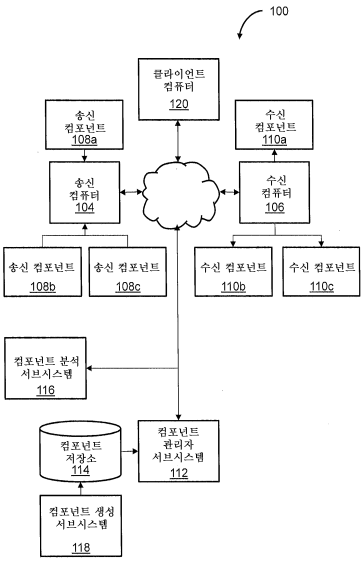
(54) 발명의 명칭 **안전한 컴퓨터 환경을 제공하는 시스템 및 방법**

(57) 요 약

안전한 컴퓨터 환경을 제공하는 방법이 개시된다. 소정의 구현에서, 소프트웨어 컴포넌트 내의 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하고; 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진(known-good) 소프트웨어 컴포넌트로 교체

(뒷면에 계속)

대 표 도 - 도1



하고; 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고/하거나 각기 다른 암호 체계를 사용하여 이들 슬라이스를 암호화하고; 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하고; 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 사용하여 정보를 안전하게 하고; 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 처리함으로써; 또는 다른 방법들을 이용함으로써, 안전한 컴퓨터 환경이 제공된다.

(30) 우선권주장
61/684,744 2012년08월18일 미국(US)
61/684,745 2012년08월18일 미국(US)
61/684,746 2012년08월18일 미국(US)

명 세 서

청구범위

청구항 1

컴포넌트 생성 환경에서 생성되는 소프트웨어 컴포넌트들을 상기 컴포넌트 생성 환경과 분리된 컴포넌트 저장소에 놓기 전에 상기 소프트웨어 컴포넌트들에서의 공격 가능(exploitable)하거나 잠재적으로 공격 가능한 서브 컴포넌트들을 제한하는 컴퓨터 구현 방법으로서, 상기 방법은 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하는 컴퓨터 시스템에 의해 구현되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 컴포넌트 생성 환경에서 적어도 제1 소프트웨어 컴포넌트를 식별하는 단계 - 상기 제1 소프트웨어 컴포넌트는 상기 제1 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 기능을 제공하는 제1 서브 컴포넌트를 포함함 - ;

상기 컴퓨터 시스템에 의해, 상기 제1 서브 컴포넌트를 식별하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능이 상기 제1 소프트웨어 컴포넌트가 실행될 때 상기 제1 소프트웨어 컴포넌트를 통해 사용가능하지 않도록 상기 제1 서브 컴포넌트를 무력화하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 소프트웨어 컴포넌트가 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능이 사용될 가능성 없이 상기 컴포넌트 저장소에 놓이도록 상기 제1 서브 컴포넌트가 무력화된 후에 상기 제1 소프트웨어 컴포넌트를 상기 컴포넌트 저장소에 놓는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 단계는 상기 제1 서브 컴포넌트를 상기 제1 소프트웨어 컴포넌트로부터 제거하는 단계를 포함하는 방법.

청구항 3

제1항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 단계는 상기 제1 서브 컴포넌트를 상기 제1 소프트웨어 컴포넌트로부터 제거하지 않고서 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능을 턴 오프하는 단계를 포함하는 방법.

청구항 4

제1항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 단계는 상기 제1 서브 컴포넌트가 무력화 명령에 기초하여 상기 제1 소프트웨어 컴포넌트로부터 제거되도록 상기 제1 소프트웨어 컴포넌트를 상기 무력화 명령과 관련시키는 단계를 포함하는 방법.

청구항 5

제1항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 단계는 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능이 무력화 명령에 기초하여 턴 오프되도록 상기 제1 소프트웨어 컴포넌트를 상기 무력화 명령과 관련시키는 단계를 포함하는 방법.

청구항 6

제1항에 있어서,

상기 제1 소프트웨어 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 제2 서브 컴포넌트를 더 포함하고, 상기 제1 서브 컴포넌트는 상기 제1 서브 컴포넌트가 상기 제1 소프트웨어 컴포넌트의 기능을 위해 필요하지 않다는 결정에 기초하여 무력화되고, 상기 제2 서브 컴포넌트는 상기 제2 서브 컴포넌트가 상기 제1 소프트웨어 컴포넌트의 기능을 위해 필요하다는 결정에 기초하여 무력화되지 않는 방법.

청구항 7

제1항에 있어서,

상기 제1 소프트웨어 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 제2 서브 컴포넌트를 더 포함하고, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 제2 서브 컴포넌트가 공격 가능하거나 잠재적으로 공격 가능하게 하는 상기 제2 서브 컴포넌트와 관련된 코드 세트를 식별하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제2 서브 컴포넌트가 상기 코드 세트를 더 이상 포함하지 않도록 상기 제2 서브 컴포넌트로부터 상기 코드 세트를 제거하는 단계를 더 포함하고, 상기 제1 소프트웨어 컴포넌트는 상기 제1 소프트웨어 컴포넌트가 상기 제2 환경에 놓일 때 상기 코드 세트 없이 상기 제2 서브 컴포넌트를 포함하는 방법.

청구항 8

제1항에 있어서,

상기 제1 서브 컴포넌트에 의해 제공된 상기 기능은 상기 제1 소프트웨어 컴포넌트를 유지하는데 사용되는 유지 기능 또는 관리 기능을 포함하고, 상기 제1 서브 컴포넌트는 상기 제1 서브 컴포넌트에 의해 제공된 상기 유지 기능 또는 상기 관리 기능에 기초하여 식별되는 방법.

청구항 9

제1항에 있어서,

컴포넌트 작성 서브시스템에 의해, 상기 컴포넌트 생성 환경에서 상기 제1 소프트웨어 컴포넌트를 발생하는 단계; 및

상기 컴포넌트 작성 서브시스템에 의해, 제1 프로필 정보를 상기 제1 소프트웨어 컴포넌트와 관련시키는 단계를 더 포함하고, 상기 제1 프로필 정보는 특정한 소프트웨어 컴포넌트의 특정한 서브 컴포넌트가 무력화되어야 하는지를 결정하는데 사용되고,

상기 제1 서브 컴포넌트를 식별하는 단계는, 컴포넌트 실링 서브시스템에 의해, 상기 제1 프로필 정보에 기초하여 상기 제1 서브 컴포넌트를 식별하는 단계를 포함하는 방법.

청구항 10

제1항에 있어서,

컴포넌트 작성 서브시스템에 의해, 상기 제1 소프트웨어 컴포넌트가 상기 제1 서브 컴포넌트 및 제2 서브 컴포넌트를 포함하도록 상기 컴포넌트 생성 환경에서 상기 제1 소프트웨어 컴포넌트를 발생하는 단계; 및

상기 컴포넌트 작성 서브시스템에 의해, 제1 프로필 정보를 상기 제1 소프트웨어 컴포넌트와 관련시키는 단계를 더 포함하고, 상기 제1 프로필 정보는 특정한 소프트웨어 컴포넌트의 특정한 서브 컴포넌트가 무력화되어야 하는지를 결정하는데 사용되고, 상기 제2 서브 컴포넌트는 상기 제1 프로필 정보에 기초하여 무력화되지 않는 방법.

청구항 11

제1항에 있어서,

컴포넌트 엑스포팅 서브시스템에 의해, 상기 제1 서브 컴포넌트가 무력화된 후에 그리고 상기 제1 소프트웨어

컴포넌트가 상기 컴포넌트 저장소에 놓이기 전에 상기 제1 소프트웨어 컴포넌트에 대한 무결성 정보를 발생하는 단계를 더 포함하고, 상기 무결성 정보는 상기 무결성 정보가 상기 제1 소프트웨어 컴포넌트에 대해 발생된 후에 상기 제1 소프트웨어 컴포넌트가 수정되었는지를 결정하는데 사용되는 방법.

청구항 12

컴포넌트 생성 환경에서 생성되는 소프트웨어 컴포넌트들을 상기 컴포넌트 생성 환경과 분리된 컴포넌트 저장소에 놓기 전에 상기 소프트웨어 컴포넌트들에서의 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트들을 제한하는 시스템으로서, 상기 시스템은

하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하고,

상기 하나 이상의 프로세서는

상기 컴포넌트 생성 환경에서 적어도 제1 소프트웨어 컴포넌트를 식별하고 - 상기 제1 소프트웨어 컴포넌트는 상기 제1 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 기능을 제공하는 제1 서브 컴포넌트를 포함함 -;

상기 제1 서브 컴포넌트를 식별하고;

상기 제1 서브 컴포넌트에 의해 제공된 상기 기능이 상기 제1 소프트웨어 컴포넌트가 실행될 때 상기 제1 소프트웨어 컴포넌트를 통해 사용가능하지 않도록 상기 제1 서브 컴포넌트를 무력화하고;

상기 제1 소프트웨어 컴포넌트가 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능이 사용될 가능성 없이 상기 컴포넌트 저장소에 놓이도록 상기 제1 서브 컴포넌트가 무력화된 후에 상기 제1 소프트웨어 컴포넌트를 상기 컴포넌트 저장소에 놓도록 프로그램된 시스템.

청구항 13

제12항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 것은 상기 제1 서브 컴포넌트를 상기 제1 소프트웨어 컴포넌트로부터 제거하는 것을 포함하는 시스템.

청구항 14

제12항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 것은 상기 제1 서브 컴포넌트를 상기 제1 소프트웨어 컴포넌트로부터 제거하지 않고서 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능을 턴 오프하는 것을 포함하는 시스템.

청구항 15

제12항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 것은 상기 제1 서브 컴포넌트가 무력화 명령에 기초하여 상기 제1 소프트웨어 컴포넌트로부터 제거되도록 상기 제1 소프트웨어 컴포넌트를 상기 무력화 명령과 관련시키는 것을 포함하는 시스템.

청구항 16

제12항에 있어서,

상기 제1 서브 컴포넌트를 무력화하는 것은 상기 제1 서브 컴포넌트에 의해 제공된 상기 기능이 무력화 명령에 기초하여 턴 오프되도록 상기 제1 소프트웨어 컴포넌트를 상기 무력화 명령과 관련시키는 것을 포함하는 시스템.

청구항 17

제12항에 있어서,

상기 제1 소프트웨어 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 제2 서브 컴포넌트를 더 포함하고,

상기 제1 서브 컴포넌트는 상기 제1 서브 컴포넌트가 상기 제1 소프트웨어 컴포넌트의 기능을 위해 필요하지 않다는 결정에 기초하여 무력화되고, 상기 제2 서브 컴포넌트는 상기 제2 서브 컴포넌트가 상기 제1 소프트웨어 컴포넌트의 기능을 위해 필요하다는 결정에 기초하여 무력화되지 않는 시스템.

청구항 18

제12항에 있어서,

상기 제1 소프트웨어 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 제2 서브 컴포넌트를 더 포함하고, 상기 하나 이상의 프로세서는

상기 제2 서브 컴포넌트가 공격 가능하거나 잠재적으로 공격 가능하게 하는 상기 제2 서브 컴포넌트와 관련된 코드 세트를 식별하고;

상기 제2 서브 컴포넌트가 상기 코드 세트를 더 이상 포함하지 않도록 상기 제2 서브 컴포넌트로부터 상기 코드 세트를 제거하도록 프로그램되고, 상기 제1 소프트웨어 컴포넌트는 상기 제1 소프트웨어 컴포넌트가 상기 제2 환경에 놓일 때 상기 코드 세트 없이 상기 제2 서브 컴포넌트를 포함하는 시스템.

청구항 19

제12항에 있어서,

상기 제1 서브 컴포넌트에 의해 제공된 상기 기능은 상기 제1 소프트웨어 컴포넌트를 유지하는데 사용되는 유지 기능 또는 관리 기능을 포함하고, 상기 제1 서브 컴포넌트는 상기 제1 서브 컴포넌트에 의해 제공된 상기 유지 기능 또는 상기 관리 기능에 기초하여 식별되는 시스템.

청구항 20

제12항에 있어서,

상기 하나 이상의 프로세서는

상기 컴포넌트 생성 환경에서 상기 제1 소프트웨어 컴포넌트를 발생하고;

제1 프로필 정보를 상기 제1 소프트웨어 컴포넌트와 관련시키도록 프로그램되고, 상기 제1 프로필 정보는 특정한 소프트웨어 컴포넌트의 특정한 서브 컴포넌트가 무력화되어야 하는지를 결정하는데 사용되고,

상기 제1 서브 컴포넌트를 식별하는 것은 상기 제1 프로필 정보에 기초하여 상기 제1 서브 컴포넌트를 식별하는 것을 포함하는 시스템.

청구항 21

제12항에 있어서,

상기 하나 이상의 프로세서는

상기 제1 소프트웨어 컴포넌트가 상기 제1 서브 컴포넌트 및 제2 서브 컴포넌트를 포함하도록 상기 컴포넌트 생성 환경에서 상기 제1 소프트웨어 컴포넌트를 발생하고;

제1 프로필 정보를 상기 제1 소프트웨어 컴포넌트와 관련시키도록 프로그램되고, 상기 제1 프로필 정보는 특정한 소프트웨어 컴포넌트의 특정한 서브 컴포넌트가 무력화되어야 하는지를 결정하는데 사용되고, 상기 제2 서브 컴포넌트는 상기 제1 프로필 정보에 기초하여 무력화되지 않는 시스템.

청구항 22

제12항에 있어서,

상기 하나 이상의 프로세서는

상기 제1 서브 컴포넌트가 무력화된 후에 그리고 상기 제1 소프트웨어 컴포넌트가 상기 컴포넌트 저장소에 놓이기 전에 상기 제1 소프트웨어 컴포넌트에 대한 무결성 정보를 발생하도록 프로그램되고, 상기 무결성 정보는 상기 무결성 정보가 상기 제1 소프트웨어 컴포넌트에 대해 발생된 후에 상기 제1 소프트웨어 컴포넌트가 수정되었

는지를 결정하는데 사용되는 시스템.

청구항 23

런타임 환경에서 실행하고 있는 소프트웨어 컴포넌트들을, 상기 런타임 환경에서 실행하고 있는 상기 소프트웨어 컴포넌트들이 손상되었거나 잠재적으로 손상되었는지에 관계없이, 대응하는 양호한 것으로 알려진(known-good) 소프트웨어 컴포넌트들로 교체하는 컴퓨터 구현 방법으로서, 상기 방법은 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하는 컴퓨터 시스템에 의해 구현되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 런타임 환경에서 실행하고 있는 적어도 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 나타내는 적어도 제1 이벤트를 결정하는 단계 - 상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 결정됨 -;

상기 컴퓨터 시스템에 의해, 상기 런타임 환경과 분리된 컴포넌트 저장소로부터 적어도 제2 소프트웨어 컴포넌트를 얻는 단계 - 상기 제2 소프트웨어 컴포넌트는 상기 제1 소프트웨어 컴포넌트에 대응함 -; 및

상기 컴퓨터 시스템에 의해, 상기 제2 소프트웨어 컴포넌트가 상기 제1 이벤트 후에 상기 런타임 환경에서 사용가능하고 상기 제1 소프트웨어 컴포넌트가 상기 제1 이벤트 후에 상기 런타임 환경에서 더 이상 사용가능하지 않도록 상기 제1 이벤트에 기초하여 상기 제1 소프트웨어 컴포넌트를 상기 제2 소프트웨어 컴포넌트로 교체하는 단계

를 포함하는 방법.

청구항 24

제23항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 런타임 환경에서 상기 제2 소프트웨어 컴포넌트를 인스턴스화하는 단계를 더 포함하고, 상기 제2 소프트웨어 컴포넌트는 상기 제2 소프트웨어 컴포넌트가 상기 런타임 환경에서 사용가능하지 않도록 비활성 경로에서 인스턴스화되고,

상기 제1 소프트웨어 컴포넌트를 상기 제2 소프트웨어 컴포넌트로 교체하는 단계는

상기 컴퓨터 시스템에 의해, 상기 제2 소프트웨어 컴포넌트가 상기 런타임 환경에서 사용가능하도록 상기 제2 소프트웨어 컴포넌트를 활성 경로에 놓는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 소프트웨어 컴포넌트를 상기 활성 경로 밖으로 이동시키는 단계를 포함하는 방법.

청구항 25

제24항에 있어서,

상기 제1 소프트웨어 컴포넌트는 상기 제2 소프트웨어 컴포넌트가 상기 활성 경로에 놓일 때 상기 활성 경로 밖으로 이동되는 방법.

청구항 26

제24항에 있어서,

상기 제1 소프트웨어 컴포넌트 및 상기 제2 소프트웨어 컴포넌트는 각각 가상 머신을 포함하고, 상기 제2 소프트웨어 컴포넌트를 상기 활성 경로에 놓는 단계는 상기 제2 소프트웨어 컴포넌트가 상기 런타임 환경에서 사용가능하도록 상기 제2 소프트웨어 컴포넌트와 관련된 네트워크 설정을 갱신하는 단계를 포함하는 방법.

청구항 27

제24항에 있어서,

상기 제1 소프트웨어 컴포넌트 및 상기 제2 소프트웨어 컴포넌트는 각각 운영 체제 처리를 포함하고, 상기 제2 소프트웨어 컴포넌트를 상기 활성 경로에 놓는 단계는 상기 제2 소프트웨어 컴포넌트를 처리 간 통신 설정을 통해 상기 활성 경로로 이동시키는 단계를 포함하는 방법.

청구항 28

제24항에 있어서,

상기 제1 소프트웨어 컴포넌트 및 상기 제2 소프트웨어 컴포넌트는 각각 실행가능한 코드를 포함하고, 상기 제2 소프트웨어 컴포넌트를 상기 활성 경로에 놓는 단계는 상기 제2 소프트웨어 컴포넌트와 관련된 메모리 어드레스 포인터 또는 룩업 테이블을 갱신하는 단계를 포함하는 방법.

청구항 29

제23항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제2 소프트웨어 컴포넌트와 관련된 무결성 정보를 얻는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 무결성 정보에 기초하여 상기 제2 소프트웨어 컴포넌트를 유효화하는 단계를 더 포함하고, 상기 제1 소프트웨어 컴포넌트의 교체는 상기 제2 소프트웨어 컴포넌트의 유효화에 기초하는 방법.

청구항 30

제23항에 있어서,

컴포넌트 분석 서브시스템에 의해, 상기 제1 소프트웨어 컴포넌트가 교체된 후에 상기 제1 소프트웨어 컴포넌트를 분석하는 단계; 및

컴포넌트 분석 서브시스템에 의해, 상기 분석에 기초하여 상기 제1 소프트웨어 컴포넌트가 손상되었는지 그리고/또는 얼마나 손상되었는지를 결정하는 단계를 더 포함하는 방법.

청구항 31

제23항에 있어서,

상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트가 하나 이상의 교체 시간 간격의 경과 후에 교체되도록 상기 하나 이상의 교체 시간 간격의 경과를 포함할 수 있고, 상기 제2 소프트웨어 컴포넌트가 교체되어야 하는 것을 나타내는 제2 이벤트는 상기 제2 소프트웨어 컴포넌트가 상기 하나 이상의 교체 시간 간격의 제2 경과 후에 교체되도록 상기 하나 이상의 교체 시간 간격의 제2 경과를 포함할 수 있는 방법.

청구항 32

제23항에 있어서,

상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과했다는 결정에 기초하는 방법.

청구항 33

제23항에 있어서,

상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트의 갱신된 버전이 사용가능하다는 결정에 기초하고, 상기 제2 소프트웨어 컴포넌트는 상기 갱신된 버전을 포함하는 방법.

청구항 34

제23항에 있어서,

컴포넌트 생성 서브시스템에 의해, 상기 제2 소프트웨어 컴포넌트를 상기 컴포넌트 저장소에 부가하는 단계를 더 포함하고, 상기 컴포넌트 생성 서브시스템은 상기 컴포넌트 생성 서브시스템이 상기 런타임 환경으로부터 손상될 수 없도록 상기 런타임 환경으로부터 접근 불가능한 방법.

청구항 35

제34항에 있어서,

상기 컴포넌트 생성 서브시스템은 상기 컴포넌트 저장소에 판독 접근하지 못하는 방법.

청구항 36

제23항에 있어서,

상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 단계는 컴포넌트 관리자 서브시스템에 의해, 상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 단계를 포함하고, 상기 컴포넌트 관리자 서브시스템은 상기 컴포넌트 저장소에 기입 접근하지 못하는 방법.

청구항 37

제23항에 있어서,

컴포넌트 생성 서브시스템에 의해, 상기 제2 소프트웨어 컴포넌트를 상기 컴포넌트 저장소에 부가하는 단계를 더 포함하고, 상기 컴포넌트 생성 서브시스템은 상기 컴포넌트 생성 서브시스템이 상기 런타임 환경으로부터 손상될 수 없도록 상기 런타임 환경으로부터 접근 불가능하고,

상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 단계는 상기 컴포넌트 관리자 서브시스템에 의해, 상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 단계를 포함하고, 상기 컴포넌트 관리자 서브시스템은 상기 컴포넌트 저장소에 기입 접근하지 못하는 방법.

청구항 38

런타임 환경에서 실행하고 있는 소프트웨어 컴포넌트들을, 상기 런타임 환경에서 실행하고 있는 상기 소프트웨어 컴포넌트들이 손상되었거나 잠재적으로 손상되었는지에 관계없이, 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트들로 교체하는 시스템으로서, 상기 시스템은

하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하고,

상기 하나 이상의 프로세서는

상기 런타임 환경에서 실행하고 있는 적어도 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 나타내는 적어도 제1 이벤트를 결정하고 - 상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 결정됨 -;

상기 런타임 환경과 분리된 컴포넌트 저장소로부터 적어도 제2 소프트웨어 컴포넌트를 얻고 - 상기 제2 소프트웨어 컴포넌트는 상기 제1 소프트웨어 컴포넌트에 대응함 -;

상기 제2 소프트웨어 컴포넌트가 상기 제1 이벤트 후에 상기 런타임 환경에서 사용가능하고 상기 제1 소프트웨어 컴포넌트가 상기 제1 이벤트 후에 상기 런타임 환경에서 더 이상 사용가능하지 않도록 상기 제1 이벤트에 기초하여 상기 제1 소프트웨어 컴포넌트를 상기 제2 소프트웨어 컴포넌트로 교체하도록

프로그램된 시스템.

청구항 39

제38항에 있어서,

상기 하나 이상의 프로세서는

상기 런타임 환경에서 상기 제2 소프트웨어 컴포넌트를 인스턴스화하도록 프로그램되고, 상기 제2 소프트웨어 컴포넌트는 상기 제2 소프트웨어 컴포넌트가 상기 런타임 환경에서 사용가능하지 않도록 비활성 경로에서 인스턴스화되고,

상기 제1 소프트웨어 컴포넌트를 상기 제2 소프트웨어 컴포넌트로 교체하는 것은

상기 제2 소프트웨어 컴포넌트가 상기 런타임 환경에서 사용가능하도록 상기 제2 소프트웨어 컴포넌트를 활성 경로에 놓고;

상기 제1 소프트웨어 컴포넌트를 상기 활성 경로 밖으로 이동시키는 것을 포함하는 시스템.

청구항 40

제39항에 있어서,

상기 제1 소프트웨어 컴포넌트는 상기 제2 소프트웨어 컴포넌트가 상기 활성 경로에 놓일 때 상기 활성 경로 밖으로 이동되는 시스템.

청구항 41

제39항에 있어서,

상기 제1 소프트웨어 컴포넌트 및 상기 제2 소프트웨어 컴포넌트는 각각 가상 머신을 포함하고, 상기 제2 소프트웨어 컴포넌트를 상기 활성 경로에 놓는 것은 상기 제2 소프트웨어 컴포넌트가 상기 런타임 환경에서 사용가능하도록 상기 제2 소프트웨어 컴포넌트와 관련된 네트워크 설정을 갱신하는 것을 포함하는 시스템.

청구항 42

제39항에 있어서,

상기 제1 소프트웨어 컴포넌트 및 상기 제2 소프트웨어 컴포넌트는 각각 운영 체제 처리를 포함하고, 상기 제2 소프트웨어 컴포넌트를 상기 활성 경로에 놓는 것은 상기 제2 소프트웨어 컴포넌트를 처리 간 통신 설정을 통해 상기 활성 경로로 이동시키는 것을 포함하는 시스템.

청구항 43

제39항에 있어서,

상기 제1 소프트웨어 컴포넌트 및 상기 제2 소프트웨어 컴포넌트는 각각 실행가능한 코드를 포함하고, 상기 제2 소프트웨어 컴포넌트를 상기 활성 경로에 놓는 것은 상기 제2 소프트웨어 컴포넌트와 관련된 메모리 어드레스 포인터 또는 룩업 테이블을 갱신하는 것을 포함하는 시스템.

청구항 44

제38항에 있어서,

상기 하나 이상의 프로세서는

상기 제2 소프트웨어 컴포넌트와 관련된 무결성 정보를 얻고;

상기 무결성 정보에 기초하여 상기 제2 소프트웨어 컴포넌트를 유효화하도록 프로그램되고, 상기 제1 소프트웨어 컴포넌트의 교체는 상기 제2 소프트웨어 컴포넌트의 유효화에 기초하는 시스템.

청구항 45

제38항에 있어서,

상기 하나 이상의 프로세서는

상기 제1 소프트웨어 컴포넌트가 교체된 후에 상기 제1 소프트웨어 컴포넌트를 분석하고;

상기 분석에 기초하여 상기 제1 소프트웨어 컴포넌트가 손상되었는지 그리고/또는 얼마나 손상되었는지를 결정하도록 프로그램된 시스템.

청구항 46

제38항에 있어서,

상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트가 하나 이상의 교체 시간 간격의 경과 후에 교체되도록 상기 하나 이상의 교체 시간 간격의 경과를 포함할 수 있고, 상기 제2 소프트웨어 컴포넌트가 교체되어야 하는 것을 나타내는 제2 이벤트는 상기 제2 소프트웨어 컴포넌트가 상기 하나 이상의 교체 시간 간격의 제2 경과 후에 교체되도록 상기 하나 이상의 교체 시간 간격의 제2 경과를 포함할 수 있는 시스템.

청구항 47

제38항에 있어서,

상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과했다는 결정에 기초하는 시스템.

청구항 48

제38항에 있어서,

상기 제1 이벤트는 상기 제1 소프트웨어 컴포넌트의 갱신된 버전이 사용가능하다는 결정에 기초하고, 상기 제2 소프트웨어 컴포넌트는 상기 갱신된 버전을 포함하는 시스템.

청구항 49

제38항에 있어서,

상기 하나 이상의 프로세서는

컴포넌트 생성 서브시스템을 통해 상기 소프트웨어 컴포넌트를 상기 컴포넌트 저장소에 부가하도록 프로그램되고, 상기 컴포넌트 생성 서브시스템은 상기 컴포넌트 생성 서브시스템이 상기 런타임 환경으로부터 손상될 수 없도록 상기 런타임 환경으로부터 접근 불가능한 시스템.

청구항 50

제49항에 있어서,

상기 컴포넌트 생성 서브시스템은 상기 컴포넌트 저장소에 판독 접근하지 못하는 시스템.

청구항 51

제38항에 있어서,

상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 것은 컴퓨터 관리자 서브시스템을 통해 상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 것을 포함하고, 상기 컴포넌트 관리자 서브시스템은 상기 컴포넌트 저장소에 기입 접근하지 못하는 시스템.

청구항 52

제38항에 있어서,

상기 하나 이상의 프로세서는

컴포넌트 생성 서브시스템을 통해 상기 제2 소프트웨어 컴포넌트를 상기 컴포넌트 저장소에 부가하도록 프로그램되고, 상기 컴포넌트 생성 서브시스템은 상기 컴포넌트 생성 서브시스템이 상기 런타임 환경으로부터 손상될 수 없도록 상기 런타임 환경으로부터 접근 불가능하고,

상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 것은 컴포넌트 관리자 서브시스템을 통해 상기 컴포넌트 저장소로부터 상기 제2 소프트웨어 컴포넌트를 얻는 것을 포함하고, 상기 컴포넌트 관리자 서브시스템은 상기 컴포넌트 저장소에 기입 접근하지 못하는 시스템.

청구항 53

송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스들로 조합하고 상기 정보의 안전한 송신을 위해 각기 다른 암호 체계들을 사용하여 상기 슬라이스들을 암호화하는 컴퓨터 구현 방법으로서, 상기 방법은 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하는 컴퓨터 시스템에 의해 구현되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 제1 송신 컴포넌트와 관련되어 송신될 제1 정보를 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 송신 컴포넌트와 다른 제2 송신 컴포넌트와 관련되어 송신될 제2 정보를 수신하는 단계;

상기 컴퓨터 시스템에 의해, 제1 암호 체계에 기초하여 상기 제1 정보의 적어도 일부를 나타내는 제1 데이터 슬

라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 암호 체계와 다른 제2 암호 체계에 기초하여 상기 제2 정보의 적어도 일부를 나타내는 제2 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 슬라이스용 상기 제1 암호 체계 및 상기 제2 데이터 슬라이스용 상기 제2 암호 체계를 지정하는 제1 헤더를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 헤더, 상기 제1 데이터 슬라이스, 및 상기 제2 데이터 슬라이스를 포함하는 제1 데이터 패킷을 발생하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷을 송신하는 단계

를 포함하는 방법.

청구항 54

제53항에 있어서,

상기 제1 헤더를 발생하는 단계는 제3 암호 체계에 기초하여 상기 제1 헤더를 발생하는 단계를 포함하는 방법.

청구항 55

제54항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제3 암호 체계에 기초하여 제1 핸드셰이크 요구를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 핸드셰이크 요구를 제2 컴퓨터 시스템에 송신하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제2 컴퓨터 시스템으로부터 제1 응답을 수신하는 단계를 더 포함하고,

상기 제1 헤더는 헤더들을 암호화하기 위한 상기 제3 암호 체계의 사용을 확인하는 상기 제1 응답에 기초하여 상기 제3 암호 체계를 사용하여 암호화되는 방법.

청구항 56

제55항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷이 송신되기 전에 상기 제2 컴퓨터 시스템에 송신된 데이터 패킷을 상기 제2 컴퓨터 시스템이 처리하지 못한 것에 기초하여 리셋 요구를 수신하는 단계를 더 포함하고,

상기 제1 핸드셰이크 요구는 상기 리셋 요구에 기초하여 발생되고, 상기 제1 응답은 상기 제1 헤더용 상기 제1 데이터 패킷 내의 제1 위치를 지정하는 방법.

청구항 57

제54항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제3 암호 체계와 다른 제4 암호 체계에 기초하여 제1 핸드셰이크 요구를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 핸드셰이크 요구를 제2 컴퓨터 시스템에 송신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 컴퓨터 시스템으로부터 제1 응답을 수신하는 단계 - 상기 제1 응답은 헤더 인덱스를 지정함 -;

상기 컴퓨터 시스템에 의해, 상기 제3 암호 체계에 대응하는 상기 헤더 인덱스에 기초하여 상기 제3 암호 체계를 사용하여 제2 핸드셰이크 요구를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 컴퓨터 시스템에 상기 제2 핸드셰이크 요구를 송신하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제2 컴퓨터 시스템으로부터 제2 응답을 수신하는 단계를 더 포함하고, 상기 제1 헤더는 헤더들을 암호화하기 위한 상기 제3 암호 체계의 사용을 확인하는 상기 제2 응답에 기초하여 상기 제3 암호 체계를 사용하여 암호화되는 방법.

청구항 58

제53항에 있어서,

상기 제1 헤더는 상기 제1 데이터 패킷 내의 상기 제1 데이터 슬라이스의 위치 및 상기 제1 데이터 패킷 내의 상기 제2 데이터 슬라이스의 위치를 지정하는 방법.

청구항 59

제53항에 있어서, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 제1 암호 체계에 기초하여 상기 제1 정보의 적어도 제2 부분을 나타내는 제3 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 암호 체계에 기초하여 상기 제2 정보의 적어도 제2 부분을 나타내는 제4 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제3 데이터 슬라이스용 상기 제1 암호 체계 및 상기 제4 데이터 슬라이스용 상기 제2 암호 체계를 지정하는 제2 헤더를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 헤더, 상기 제3 데이터 슬라이스, 및 상기 제4 데이터 슬라이스를 포함하는 제2 데이터 패킷을 발생하는 단계 - 상기 제1 헤더는 상기 제1 데이터 패킷 내의 제1 위치에 위치하고, 상기 제2 헤더는 상기 제1 헤더에 의해 지정된 상기 제2 데이터 패킷 내의 제2 위치에 위치하고, 상기 제2 헤더는 상기 제2 데이터 패킷 내의 상기 제3 데이터 슬라이스의 위치 및 상기 제2 데이터 패킷 내의 상기 제4 데이터 슬라이스의 위치를 지정함 -; 및

상기 컴퓨터 시스템에 의해, 상기 제2 데이터 패킷을 송신하는 단계를 더 포함하는 방법.

청구항 60

제59항에 있어서,

상기 제1 정보 및 상기 제2 정보는 제1 세션 동안에 수신되고, 상기 제2 헤더는 제3 헤더를 위한 제3 위치를 지정하고, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 제1 세션과 다른 제2 세션 동안에 상기 제1 송신 컴포넌트와 관련되어 송신될 제3 정보를 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 암호 체계에 기초하여 상기 제3 정보의 적어도 일부를 나타내는 제5 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제5 데이터 슬라이스용 상기 제1 암호 체계를 지정하는 상기 제3 헤더를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제3 위치의 상기 제1 헤더, 및 상기 제5 데이터 슬라이스를 포함하는 제3 데이터 패킷을 발생하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제3 데이터 패킷을 송신하는 단계를 더 포함하는 방법.

청구항 61

제53항에 있어서,

상기 제1 데이터 슬라이스 또는 상기 제2 데이터 슬라이스의 크기와 다른 크기를 갖는 랜덤하게 발생된 필러 데이터를 포함하는 제3 데이터 슬라이스를 발생하는 단계를 더 포함하고, 상기 제1 데이터 패킷은 상기 제3 데이터 슬라이스를 포함하는 방법.

청구항 62

제53항에 있어서,

제1 정책 및 제2 정책이 상기 컴퓨터 시스템에 또는 상기 컴퓨터 시스템과 관련하여 저장되고, 상기 제1 헤더는 상기 제1 정책을 지정함으로써 상기 제1 데이터 슬라이스용 상기 제1 암호 체계 및 상기 제2 데이터 슬라이스용

상기 제2 암호 체계를 지정하고, 상기 방법은

상기 컴퓨터 시스템에 의해, 제2 컴퓨터 시스템으로부터 제2 데이터 패킷을 수신하는 단계 - 상기 제2 데이터 패킷은 제2 헤더 및 제3 데이터 슬라이스를 포함하고, 상기 제3 데이터 슬라이스는 제3 정보의 일부를 나타냄 -;

상기 컴퓨터 시스템에 의해, 상기 제2 정책을 지정하는 상기 제2 헤더에 기초하여, 상기 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 상기 제1 암호 체계를 식별하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제3 정보의 상기 일부를 얻기 위해 상기 제1 암호 체계에 기초하여 상기 제3 데이터 슬라이스를 처리하는 단계를 더 포함하는 방법.

청구항 63

제62항에 있어서,

상기 제1 정책은 상기 제1 암호 체계와의 제1 암호 체계 인덱스의 관련 및 상기 제2 암호 체계와의 제2 암호 체계 인덱스의 관련을 지정하고, 상기 제2 정책은 상기 제1 암호 체계와의 제3 암호 체계 인덱스의 관련을 지정하고, 상기 제1 헤더는 상기 제1 정책, 상기 제1 데이터 슬라이스용 상기 제1 암호 체계 인덱스, 및 상기 제2 데이터 슬라이스용 상기 제2 암호 체계 인덱스를 지정함으로써 상기 제1 데이터 슬라이스용 상기 제1 암호 체계 및 상기 제2 데이터 슬라이스용 상기 제2 암호 체계를 지정하고, 상기 제1 암호 체계는 상기 제2 정책 및 상기 제3 데이터 슬라이스용 상기 제3 암호 체계 인덱스를 지정하는 상기 제2 헤더에 기초하여 상기 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 식별되는 방법.

청구항 64

각기 다른 암호 체계들을 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스들을 갖는 데이터 패킷을 처리하는 컴퓨터 구현 방법으로서, 상기 방법은 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하는 컴퓨터 시스템에 의해 구현되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 제1 헤더, 제1 송신 컴포넌트와 관련된 제1 정보의 적어도 일부를 나타내는 제1 데이터 슬라이스, 및 상기 제1 송신 컴포넌트와 다른 제2 송신 컴포넌트와 관련된 제2 정보의 적어도 일부를 나타내는 제2 데이터 슬라이스를 포함하는 제1 데이터 패킷을 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 헤더에 기초하여, 상기 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계 및 상기 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계를 식별하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 정보의 상기 일부를 얻기 위해 상기 제1 암호 체계에 기초하여 상기 제1 데이터 슬라이스를 처리하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 정보의 상기 일부를 얻기 위해 상기 제2 암호 체계에 기초하여 상기 제2 데이터 슬라이스를 처리하는 단계

를 포함하고, 상기 제2 암호 체계는 상기 제1 암호 체계와 다른 방법.

청구항 65

제64항에 있어서,

상기 컴퓨터 시스템에 의해, 제2 헤더, 상기 제1 정보의 적어도 제2 부분을 나타내는 제3 데이터 슬라이스, 및 상기 제2 정보의 적어도 제2 부분을 나타내는 제4 데이터 슬라이스를 포함하는 제2 데이터 패킷을 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 헤더에 기초하여, 상기 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 상기 제1 암호 체계 및 상기 제4 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 상기 제2 암호 체계를 식별하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 정보의 상기 제2 부분을 얻기 위해 상기 제1 암호 체계에 기초하여 상기 제3 데이터 슬라이스를 처리하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 정보의 상기 제2 부분을 얻기 위해 상기 제2 암호 체계에 기초하여 상기 제4 데이터 슬라이스를 처리하는 단계;

상기 컴퓨터 시스템에 의해, 제1 수신 컴포넌트를 위한 상기 제1 정보를 형성하기 위해 상기 제1 정보의 상기 일부와 상기 제1 정보의 상기 제2 부분을 조합하는 단계; 및

상기 컴퓨터 시스템에 의해, 제2 수신 컴포넌트를 위한 상기 제2 정보를 형성하기 위해 상기 제2 정보의 상기 일부와 상기 제2 정보의 상기 제2 부분을 조합하는 단계를 더 포함하는 방법.

청구항 66

제65항에 있어서,

상기 제1 헤더는 상기 제1 데이터 패킷 내의 제1 위치에 위치하고, 상기 제1 헤더는 상기 제2 헤더용 상기 제2 데이터 패킷 내의 제2 위치를 지정하는 방법.

청구항 67

제64항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 상기 제1 암호 체계를 식별하고 상기 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 상기 제2 암호 체계를 식별하기 위해 제3 암호 체계를 사용하여 상기 제1 헤더를 처리하는 단계를 더 포함하는 방법.

청구항 68

제67항에 있어서,

상기 제1 데이터 패킷은 제2 컴퓨터 시스템으로부터 수신되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 제2 컴퓨터 시스템으로부터 제1 핸드셰이크 요구를 수신하는 단계 - 상기 제1 핸드셰이크 요구는 상기 제3 암호 체계에 기초하여 발생됨 -;

상기 컴퓨터 시스템에 의해, 헤더들을 발생하기 위한 상기 제3 암호 체계의 사용을 확인하는 제1 응답을 발생하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 응답을 상기 제2 컴퓨터 시스템에 송신하는 단계를 더 포함하고, 상기 제1 헤더는 상기 제1 응답에 기초하여 상기 제3 암호 체계를 사용하여 발생되는 방법.

청구항 69

제68항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷이 수신되기 전에 수신된 데이터를 처리하지 못한 것에 기초하여 리셋 요구를 발생하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 리셋 요구를 상기 제2 컴퓨터 시스템에 송신하는 단계를 더 포함하고,

상기 제1 핸드셰이크 요구는 상기 리셋 요구의 송신에 기초하여 상기 제2 컴퓨터 시스템으로부터 수신되고, 상기 제1 응답은 상기 제1 헤더용 상기 제1 데이터 패킷 내의 제1 위치를 지정하는 방법.

청구항 70

제64항에 있어서,

상기 제1 데이터 패킷은 랜덤하게 발생된 필러 데이터를 포함하는 방법.

청구항 71

다른 데이터 패킷들에 대해 데이터 패킷 내의 각기 다른 위치들에서 시작하는 헤더들을 갖는 데이터 패킷들을 사용하여 정보를 안전하게 하는 컴퓨터 구현 방법으로서, 상기 방법은 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하는 컴퓨터 시스템에 의해 구현되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 송신될 제1 정보를 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 정보의 적어도 제1 부분을 나타내는 제1 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 정보의 적어도 제2 부분을 나타내는 제2 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 제1 헤더 및 상기 제1 데이터 슬라이스를 포함하는 제1 데이터 패킷을 발생하는 단계 - 상기 제1 헤더는 상기 제1 데이터 패킷 내의 제1 위치에 위치함 -;

상기 컴퓨터 시스템에 의해, 제2 헤더 및 상기 제2 데이터 슬라이스를 포함하는 제2 데이터 패킷을 발생하는 단계 - 상기 제2 헤더는 상기 제2 데이터 패킷 내의 제2 위치에 위치하고, 상기 제1 헤더는 상기 제2 헤더를 위한 상기 제2 위치를 지정함 -; 및

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷 및 제2 데이터 패킷을 송신하는 단계

를 포함하는 방법.

청구항 72

제71항에 있어서,

상기 컴퓨터 시스템에 의해, 제1 암호 체계에 기초하여 상기 제1 헤더를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 암호 체계에 기초하여 상기 제2 헤더를 발생하는 단계를 더 포함하고,

상기 제1 데이터 슬라이스를 발생하는 단계는 상기 제1 암호 체계와 다른 제2 암호 체계에 기초하여 상기 제1 데이터 슬라이스를 발생하는 단계를 포함하고, 상기 제2 데이터 슬라이스를 발생하는 단계는 상기 제2 암호 체계에 기초하여 상기 제2 데이터 슬라이스를 발생하는 단계를 포함하는 방법.

청구항 73

제72항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷이 송신되기 전에 제2 컴퓨터 시스템에 송신된 데이터 패킷을 상기 제2 컴퓨터 시스템이 처리하지 못한 것에 기초하여 리셋 요구를 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 리셋 요구에 기초하여 상기 제1 암호 체계를 사용하여 제1 핸드셰이크 요구를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 핸드셰이크 요구를 상기 제2 컴퓨터 시스템에 송신하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제2 컴퓨터 시스템으로부터 제1 응답을 수신하는 단계를 더 포함하고, 상기 제1 헤더는 헤더들을 암호화하기 위한 상기 제1 암호 체계의 사용을 확인하는 상기 제1 응답에 기초하여 상기 제1 암호 체계를 사용하여 암호화되는 방법.

청구항 74

제71항에 있어서,

상기 제1 정보는 제1 세션 동안에 수신되고, 상기 제2 헤더는 제3 헤더용 제3 데이터 패킷 내의 제3 위치를 지정하고, 상기 방법은

상기 컴퓨터 시스템에 의해, 상기 제1 세션과 다른 제2 세션 동안에 제2 정보를 수신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 정보의 적어도 일부를 나타내는 제3 데이터 슬라이스를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제3 헤더 및 상기 제3 데이터 슬라이스를 포함하는 상기 제3 데이터 패킷을 발생하는 단계 - 상기 제3 헤더는 상기 제3 위치에 위치함 -; 및

상기 컴퓨터 시스템에 의해, 상기 제3 데이터 패킷을 송신하는 단계를 더 포함하는 방법.

청구항 75

제71항에 있어서,

상기 제1 데이터 패킷은 랜덤하게 발생된 필러 데이터를 포함하는 방법.

청구항 76

다른 데이터 패킷들에 대해 데이터 패킷 내의 각기 다른 위치들에서 시작하는 헤더들을 갖는 데이터 패킷들을 처리하는 컴퓨터 구현 방법으로서, 상기 방법은 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 프로세서를 포함하는 컴퓨터 시스템에 의해 구현되고, 상기 방법은

상기 컴퓨터 시스템에 의해, 제1 데이터 패킷 및 제2 데이터 패킷을 수신하는 단계 - 상기 제1 데이터 패킷은 제1 헤더 및 제1 정보의 제1 부분을 나타내는 제1 데이터 슬라이스를 포함하고, 상기 제2 데이터 패킷은 제2 헤더 및 상기 제1 정보의 제2 부분을 나타내는 제2 데이터 슬라이스를 포함함 -;

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷 내의 제1 위치에서 상기 제1 헤더를 식별하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제2 헤더를 위한 제2 위치를 지정하는 상기 제1 헤더에 기초하여 상기 제2 데이터 패킷 내의 제2 위치에서 상기 제2 헤더를 식별하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 헤더에 기초하여 상기 제1 데이터 패킷 내의 상기 제1 데이터 슬라이스를 식별하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제2 헤더에 기초하여 상기 제2 데이터 패킷 내의 상기 제2 데이터 슬라이스를 식별하는 단계

를 포함하는 방법.

청구항 77

제76항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 정보의 상기 제1 부분을 얻기 위해 상기 제1 데이터 슬라이스를 처리하는 단계;

상기 컴퓨터 시스템에 의해, 상기 제1 정보의 상기 제2 부분을 얻기 위해 상기 제2 데이터 슬라이스를 처리하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 정보를 형성하기 위해 상기 제1 정보의 상기 제1 부분과 상기 제1 정보의 상기 제2 부분을 조합하는 단계를 더 포함하는 방법.

청구항 78

제77항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 헤더를 암호화하는데 사용되는 제1 암호 체계를 식별하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 슬라이스를 암호화하는데 사용되는 제2 암호 체계를 식별하기 위해 상기 제1 암호 체계를 사용하여 상기 제1 헤더를 식별하는 단계를 더 포함하고, 상기 제2 암호 체계는 상기 제1 암호 체계와 다르고,

상기 제1 데이터 슬라이스를 처리하는 단계는 상기 제1 정보의 상기 제1 부분을 얻기 위해 상기 제2 암호 체계를 사용하여 상기 제1 데이터 슬라이스를 처리하는 단계를 포함하고,

상기 제2 데이터 슬라이스를 처리하는 단계는 상기 제1 정보의 상기 제2 부분을 얻기 위해 상기 제2 암호 체계를 사용하여 상기 제2 데이터 슬라이스를 처리하는 단계를 포함하는 방법.

청구항 79

제78항에 있어서,

상기 컴퓨터 시스템에 의해, 상기 제1 데이터 패킷이 수신되기 전에 수신된 데이터 패킷을 처리하지 못한 것에 기초하여 리셋 요구를 발생하는 단계;

상기 컴퓨터 시스템에 의해, 상기 리셋 요구를 제2 컴퓨터 시스템에 송신하는 단계;

상기 컴퓨터 시스템에 의해, 상기 리셋 요구의 송신에 기초하여 상기 제2 컴퓨터 시스템으로부터 제1 핸드셰이크 요구를 수신하는 단계 - 상기 제1 핸드셰이크 요구는 상기 제1 암호 체계를 사용하여 암호화됨 -;

상기 컴퓨터 시스템에 의해, 헤더들을 암호화하기 위한 상기 제1 암호 체계의 사용을 확인하는 제1 응답을 발생하는 단계; 및

상기 컴퓨터 시스템에 의해, 상기 제1 응답을 송신하는 단계를 더 포함하고, 상기 제1 헤더는 상기 제1 응답에 기초하여 상기 제1 암호 체계를 사용하여 암호화되는 방법.

청구항 80

제76항에 있어서,

상기 제1 데이터 패킷 또는 상기 제2 데이터 패킷은 랜덤하게 발생된 필러 데이터를 포함하는 방법.

발명의 설명

기 술 분 야

[0001] **관련 출원에 대한 상호 참조**

[0002] 본 출원은 (1) 2012년 8월 18일자 출원된 미국 임시 특허 출원 번호 61/684,743호; (2) 2012년 8월 18일자 출원된 미국 임시 특허 출원 번호 61/684,744호; (3) 2012년 8월 18일자 출원된 미국 임시 특허 출원 번호 61/684,745호; 및 (4) 2012년 8월 18일자 출원된 미국 임시 특허 출원 번호 61/684,746호의 우선권을 주장하며, 이들 각각은 전체적으로 본 명세서에 참고로 도입된다.

[0003] 본 출원은 동일자로 제출된 다음의 계류중인 미국 유틸리티 특허 출원에 부가적으로 관련된다: (1) "SYSTEM AND METHOD FOR LIMITING EXPLOITABLE OR POTENTIALLY EXPLOITABLE SUB-COMPONENTS IN SOFTWARE COMPONENTS"라고 하는 미국 특허 출원 번호 13/969,158호 [정리 번호 026409-0424092]; (2) "SYSTEM AND METHOD FOR REPLACING SOFTWARE COMPONENTS WITH CORRESPONDING KNOWN-GOOD SOFTWARE COMPONENTS WITHOUT REGARD TO WHETHER THE SOFTWARE COMPONENTS HAVE BEEN COMPROMISED OR POTENTIALLY COMPROMISED"라고 하는 미국 특허 출원 번호 13/969,181호 [정리 번호 026409-0424094]; 및 (3) "SYSTEM AND METHOD FOR INTERLEAVING INFORMATION INTO SLICES OF A DATA PACKET, DIFFERENTIALLY ENCRYPTING THE SLICES, AND OBFUSCATING INFORMATION IN THE DATA PACKET"이라고 하는 미국 특허 출원 번호 13/969,216호 [정리 번호 026409-0424095]이고, 부가적으로 이들 각각은 전체적으로 본 명세서에 참고로 도입된다.

[0004] 본 발명은 소프트웨어 컴포넌트 내의 공격 가능(exploitable)하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하고, 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진(known-good) 소프트웨어 컴포넌트로 교체하고, 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고 각기 다른 암호 체계를 사용하여 이들 슬라이스를 암호화하고, 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하고, 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 사용하여 정보를 안전하게 하고, 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 처리함으로써, 또는 다른 방법들을 이용함으로써, 컴포넌트 및/또는 정보의 무결성를 보호하고 유지하는 것에 관한 것이다.

배 경 기 술

[0005] 세계가 점차 과학 기술적으로 진보되고 컴퓨터 시스템에 의존함에 따라, 사이버 공격이 정교함 및 강도에 있어서 증가하고 있다. 이들 공격은 독점 정보를 도용하고, 멀웨어를 확산시키고, 서비스를 붕괴하고, 또는 다른 문제를 야기하는 익스플로잇(exploit)의 사용을 포함한다. 방화벽, 멀웨어 방지 검출 시스템 등과 같은 전통적인 방어 해결책은 이들 위협을 식별하고 완화시키는데 중점을 두고 있다. 그러나, 전통적인 방어 해결책은 익스플로잇 또는 다른 위협을 존재하게 하는 중요한 이슈의 많은 것을 무시한다. 예를 들어, 컴포넌트는 일반적으로 오랜 기간 동안 런타임 환경에서 노출되고, 동작에 필요하지 않을 수 있는 관리적 접근 및 특권을 제공한다. 또한, 네트워크는 런타임 환경에서 컴포넌트를 유지하는데 요구되는 접속을 통해 전형적으로 노출된다.

또한, 많은 예에서, 손상된 컴포넌트를 발견하는 것은 어려울 수 있고 최종적으로 발견된 경우에, 막대한 손상 또는 부정적 효과를 이미 받은 상태일 수 있다.

발명의 내용

해결하려는 과제

[0006] 이들 및 다른 단점을 다룬 본 개시 내용은 방법, 장치, 및/또는 시스템에 관한 것이다. 예를 들어, 시스템은 소프트웨어 컴포넌트 내의 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하고, 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하고, 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고, 각기 다른 암호 체계를 사용하여 이들 슬라이스를 암호화하고, 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하고, 다른 데이터 패킷에 대해 데이터 패킷 내의 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 사용하여 정보를 안전하게 하고/하거나, 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 처리함으로써 컴포넌트 및/또는 정보의 보안 및 무결성을 가능하게 할 수 있다.

[0007] 논의된 바와 같이, 소프트웨어 컴포넌트는 전형적으로 동작에 필요하지 않을 수 있는 관리적 접근 또는 특권을 제공한다. 이러한 접근 또는 특권은, 예를 들어, 관리적 접근/특권을 제공하는 소프트웨어 컴포넌트 또는 다른 소프트웨어 컴포넌트를 손상시키도록 멀웨어 또는 다른 위협에 의해 공격당할 수 있다. 전통적인 방어 해결책이 일반적으로 손상된 소프트웨어 컴포넌트를 발견하고 손상된 소프트웨어 컴포넌트에 관련한 부정적 효과를 완화시키는데 중점을 두지만, 이 시스템은 네트워크화된 혹은 접근 가능한 환경에 노출될 수 있는 컴포넌트의 보안 및 무결성을 가능하게 하고/하거나 네트워크를 통해 시스템 컴포넌트로부터 또는 시스템 컴포넌트에 송신되는 정보의 보안을 가능하게 하기 위해 부가적인 또는 대안적인 보호 대책을 제공할 수 있다. 부가적인 또는 대안적인 보호 대책은 개별적으로 또는 다른 보호 대책과 조합하여 사용될 수 있다.

[0008] 한 예로서, 시스템은 소프트웨어 컴포넌트가 생성된 후에 그리고 소프트웨어 컴포넌트가 그것이 손상될 수 있는 환경에 놓이기 전에 소프트웨어 컴포넌트 내의 공격 가능하거나 또는 잠재적으로 공격 가능한 서브 컴포넌트를 제거하거나 혹은 무력화시킴으로써 손상된 소프트웨어 컴포넌트 및 관련된 부정적 효과를 없애거나 혹은 감소시킬 수 있다.

[0009] 다른 예로서, 이러한 발견을 교체를 위한 전제 조건으로 하지 않고 시스템은 런타임 환경에서 사용가능한 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체할 수 있다. 이 교체는, 예를 들어, 런타임 환경에서 사용가능한 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 주기적으로 및/또는 온 디맨드로 수행될 수 있다. 바꾸어 말하면, 시스템은 특정한 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었다는 것을 반드시 결정하지 않고서 특정한 소프트웨어 컴포넌트를 교체할 수 있다. 결과적으로, 어떤 예에서 시스템은 손상된 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체할 수 있다.

[0010] 소프트웨어 컴포넌트가 런타임 환경에서 실행하고 있을 때, 소프트웨어 컴포넌트가 런타임 환경에서 실행하고 있는 동안 손상되었거나 잠재적으로 손상된 소프트웨어 컴포넌트의 모두를 식별한다는 것은 (불가능하지는 않더라도) 부담이 되는 작업일 수 있다. 이와 같이, 다른 이점들 중에서, 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 런타임 환경에서 실행하고 있거나 혹은 사용가능한 소프트웨어 컴포넌트의 빈번한 및/또는 정기적인 교체는 소프트웨어 컴포넌트가 런타임 환경에서 실행하고 있는 동안에 손상되었거나 잠재적으로 손상되었는지 소프트웨어 컴포넌트를 식별할 필요성을 없애거나 혹은 감소시킬 수 있다.

[0011] 다른 예로서, 시스템은 네트워크를 통해 송신되는 정보를 인터리빙하는 것과 난독화하는 것을 제공할 수 있다. 예를 들면, 각 데이터 패킷이 각기 다른 컴포넌트와 관련된 정보의 부분들을 포함할 수 있도록 데이터 패킷이 발생될 수 있다. 데이터 패킷 내의 각 데이터 슬라이스가 정보의 또 하나의 데이터-슬라이스-표현 부분에 관련된 또 하나의 컴포넌트와 다른 컴포넌트와 관련된 정보의 적어도 일부를 나타낼 수 있도록 정보의 각 부분이 암호화된 데이터 슬라이스로서 데이터 패킷에서 표현될 수 있다. 부가적으로, 또는 대안적으로, 데이터 패킷 내의 각 데이터 슬라이스는 데이터 패킷 내의 또 하나의 데이터 슬라이스를 암호화하는데 사용되는 또 하나의 암호 체계와 다른 암호 체계로 암호화될 수 있다. 각기 다른 암호 체계는, 예를 들어, 각기 다른 암호화 타입 및/또는 각기 다른 암호화 키의 사용을 포함할 수 있다.

[0012] 다른 예로서, 데이터 패킷 내의 헤더의 위치가 하나의 데이터 패킷으로부터 다른 데이터 패킷으로 계속 변화할 수 있도록 데이터 패킷이 발생될 수 있다. 한 가지 사용의 경우에, 후속하는 데이터 패킷은 이전의 데이터 패킷 내의 위치와 다른 데이터 패킷 내의 위치에 그 헤더를 가질 수 있다. 후속하는 데이터 패킷의 헤더의 위치는, 예를 들어, 이전의 데이터 패킷의 헤더에 의해 지정될 수 있다. 이 방식으로, 각기 다른 정보의 부분들을 조합하고, 각기 다른 암호 체계를 사용하여 데이터 슬라이스를 암호화하고/하거나, 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 사용함으로써 정보가 안전하게 될 수 있다.

[0013] 본 발명의 다양한 다른 양태, 특징, 및 장점이 본 발명의 상세한 설명 및 여기에 첨부된 도면을 통해 분명해질 것이다. 상기 일반적인 설명과 다음의 상세한 설명은 모두 예시적인 것이고 본 발명의 범위를 제한하는 것은 아니라는 것을 또한 이해하여야 한다. 본 명세서 및 청구 범위에서 사용된 바와 같이, 단수 표현은 문맥이 명확히 달리 표현하지 않는 한 복수의 대상을 포함한다. 또한, 본 명세서 및 청구 범위에서 사용된 바와 같이, "또는"은 문맥이 명확히 달리 표현하지 않는 한 "및/또는"을 의미한다.

도면의 간단한 설명

[0014] 본 발명은 유사한 참조 번호가 유사한 요소를 참조하는 첨부 도면에서 예로서 제한적이 아니게 예시된다.

도 1은 본 발명의 양태에 따른, 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 시스템의 예시도.

도 2는 본 발명의 양태에 따른, 서브시스템, 저장소, 또는 다른 컴포넌트가 존재할 수 있는 환경의 예시도 및 환경에 관련한 동작의 플로우차트.

도 3은 본 발명의 양태에 따른, 컴포넌트 생성 서브시스템의 예시도.

도 4a는 본 발명의 양태에 따른, 송신 컴퓨터의 예시도.

도 4b는 본 발명의 양태에 따른, 수신 컴퓨터의 예시도.

도 5는 본 발명의 양태에 따른, 2개의 컴퓨터 시스템 간의 정보 송신의 예시도.

도 6은 본 발명의 양태에 따른, 데이터 패킷 헤더의 예시도.

도 7은 본 발명의 양태에 따른, 런타임 환경에서 실행하고 있는 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하는 방법의 플로우차트의 예시도.

도 8은 본 발명의 양태에 따른, 소프트웨어 컴포넌트에서 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하는 방법의 플로우차트의 예시도.

도 9는 본 발명의 양태에 따른, 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고 정보의 안전한 송신을 위해 각기 다른 암호 체계를 사용하여 슬라이스를 암호화하는 방법의 플로우차트의 예시도.

도 10은 본 발명의 양태에 따른, 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하는 방법의 플로우차트의 예시도.

도 11은 본 발명의 양태에 따른, 컴포넌트의 서브 컴포넌트를 제한하고, 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하고, 네트워크를 통해 컴포넌트에 또는 컴포넌트로부터 송신되는 정보를 안전하게 하고 난독화함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도.

도 12는 본 발명의 양태에 따른, 컴포넌트의 서브 컴포넌트를 제한하고 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도.

도 13은 본 발명의 양태에 따른, 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하고 네트워크를 통해 컴포넌트에 또는 컴포넌트로부터 송신되는 정보를 안전하게 하고 난독화함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도.

도 14는 본 발명의 양태에 따른, 컴포넌트의 서브 컴포넌트를 제한하고 네트워크를 통해 컴포넌트에 또는 컴퓨터로부터 송신되는 정보를 안전하게 하고 난독화함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도.

발명을 실시하기 위한 구체적인 내용

[0015] 다음의 설명에서, 설명의 목적을 위해, 많은 특정한 상세가 본 발명의 구현의 철저한 이해를 제공하기 위해 기술된다. 그러나, 본 기술 분야의 통상의 기술자는 이들 특정한 상세 없이 또는 균등한 구성으로 실시될 수 있다는 것을 이해할 것이다. 다른 예에서, 공지된 구조 및 장치는 본 발명의 구현을 불필요하게 애매하게 하는 것을 피하기 위해서 블럭도 형태로 도시되어 있다.

[0016] 도 1은 본 발명의 한 양태에 따른, 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 시스템(100)의 예시도이다. 시스템(100)은 네트워크화된 혹은 접근 가능한 환경에 노출될 수 있는 컴포넌트의 보안 및 무결성을 가능하게 하고/하거나 시스템 컴포넌트로부터 또는 시스템 컴포넌트에 네트워크를 통해 송신되는 정보의 보안을 가능하게 하기 위해 각기 다른 보호 대책을 제공하는 하나 이상의 컴퓨터 및 서브시스템을 포함할 수 있다. 각기 다른 보호 대책은 개별적으로 또는 다른 보호 대책과 조합하여 사용될 수 있다. 보호 대책은, 예를 들어, 소프트웨어 컴포넌트의 소정의 인터페이스 또는 다른 서브 컴포넌트를 무력화하고, 소프트웨어 컴포넌트가 손상되었는지에 관계없이 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하고, 네트워크를 통해 송신되는 정보를 인터리빙하고 난독화하는 것, 및/또는 다른 보호 대책을 포함할 수 있다.

[0017] 어떤 구현에서, 시스템(100)은 손상되기 쉬운 특정한 컴포넌트의 인터페이스 또는 다른 서브 컴포넌트를 무력화할 수 있다. 예를 들어, 시스템(100)은 인스턴스화될 때 런타임 환경에서 동작하고 네트워크를 통해 접근 가능한 컴포넌트를 생성하는 컴포넌트 생성 서브시스템(118)을 포함할 수 있다. 특정한 컴포넌트는, 예를 들어, 특정한 컴포넌트에 대한 유지 또는 관리 동작을 수행하기 위해 시스템 관리자에 의해 통상적으로 사용되지만 특정한 컴포넌트를 손상시키도록 공격자 등에 의해 자주 공격당하는 컴포넌트 인터페이스와 같은, 서브 컴포넌트를 포함할 수 있다. 어떤 구현에서, 컴포넌트 생성 서브시스템(118)은 서브 컴포넌트에 더 이상 접근할 수 없고 따라서 서브 컴포넌트가 공격자 등에 의해 의해 더 이상 공격당하지 않도록 서브 컴포넌트를 무력화하도록 프로그램될 수 있다. 이와 같이, 인스턴스화될 때 컴포넌트 생성 서브시스템(118)에 의해 생성된 컴포넌트는 무력화되어서 더 이상 공격하기 쉽지 않은 소정의 서브 컴포넌트를 가질 수 있다. 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트의 무력화는, 예를 들어, 설정(configurations)(예를 들어, 다른 컴포넌트와의 유일한 관계를 설정하는 사전 설정 또는 다른 설정)이 소프트웨어 컴포넌트 및 소프트웨어 컴포넌트가 송신, 수신, 또는 저장하는 정보의 보안 및 무결성에 부정적으로 영향을 줄 수 있는 방식으로 수정되지 못하게 할 수 있다.

[0018] 컴포넌트 생성 서브시스템(118)은 컴포넌트로부터, 유지 또는 관리 기능을 제공하는 프로그램 코드 또는 명령을 제거하고, 서브 컴포넌트에 더 이상 접근할 수 없도록 컴포넌트를 설정하는 것, 및/또는, 다른 방법으로 서브 컴포넌트에 접근 불가능하게 함으로써(시스템 관리자 등에 의해 접근 불가능하게 하는 것을 포함) 서브 컴포넌트를 무력화하도록 프로그램될 수 있다. 이와 같이, 시스템은 공격당할 수 있는 소정의 서브 컴포넌트를 제거함으로써 시스템의 보안성을 향상시키기 위해서 컴포넌트 레벨에서 컴포넌트를 보호할 수 있다.

[0019] 어떤 구현에서, 컴포넌트 생성 서브시스템(118)은 그것의 서브 컴포넌트 중 하나 이상이 무력화되었는지 여부에 따라, 생성된 컴포넌트를 컴포넌트 저장소(114)에 저장할 수 있다. 이들 구현에서, 시스템(100)은 컴포넌트 저장소(114)로부터 특정한 컴포넌트를 얻고 런타임 환경에서 특정한 컴포넌트에 접근 가능하게 하도록 프로그램된 컴포넌트 관리자 서브시스템(112)을 포함할 수 있다.

[0020] 어떤 구현에서, 시스템(100)은 런타임 환경에서 동작하는 컴포넌트를 대응하는 양호한 것으로 알려진 컴포넌트로 교체할 수 있다. 예를 들어, 컴포넌트 관리자 서브시스템(112)은 런타임 환경에서 동작하는 컴포넌트를 컴포넌트가 손상되었는지에 관계없이 대응하는 양호한 것으로 알려진 컴포넌트로 교체하도록 프로그램될 수 있다. 대응하는 양호한 것으로 알려진 컴포넌트는 컴포넌트 생성 서브시스템(118)에 의해 생성될 수 있다. 어떤 구현에서, 대응하는 양호한 것으로 알려진 컴포넌트의 하나 이상의 서브 컴포넌트는 여기에 설명된 바와 같이 무력화되었다. 이들 구현에서, 시스템은 대응하는 양호한 것으로 알려진 컴포넌트로 교체되는 컴포넌트와 컴포넌트의 제한된 인터페이스를 조합하는 보호 대책을 제공할 수 있다. 다른 구현에서, 대응하는 양호한 것으로 알려진 컴포넌트의 하나 이상의 서브 컴포넌트 중 어느 것도 무력화되지 않았다.

[0021] 어떤 구현에서, 시스템(100)은 교체된 컴포넌트에 대한 분석을 수행할 수 있다. 이들 구현에서, 시스템(100)은 교체 후 분석을 수행하도록 프로그램된 컴포넌트 분석 서브시스템(116)을 포함할 수 있다. 시스템은 이러한 교체 후 분석을 사용하여 교체된 컴포넌트가 손상되었는지, 어느 정도 및/또는 어떻게 손상되었는지를 결정할 수 있다. 분석이 교체 후에 이루어지기 때문에, 분석의 결과는 분석되고 있는 컴포넌트가 교체되어야 하는지를 결정하는데 사용되지 않는다.

[0022] 어떤 구현에서, 시스템(100)은 시스템 컴포넌트에 및/또는 시스템 컴포넌트로부터 네트워크를 통해 정보를 안전하게 송신할 수 있다. 이러한 정보는, 예를 들어, 네트워크를 통해 송신될 수 있는 문서, 메시지, 텍스트, 이미지, 데이터 및/또는 다른 유형의 정보를 포함할 수 있다. 예를 들어, 하나 이상의 송신 컴포넌트(108)(도 1에 송신 컴포넌트(108a, 108b, 108c)로 도시됨)는 대응하는 수신 컴포넌트(110)(도 1에 수신 컴포넌트(110a, 110b, 110c)로 도시됨)에 관련된 정보를 개별적으로 송신하기를 원할 수 있다. 특정한 송신 컴포넌트(108) 또는 특정한 수신 컴포넌트(110)는 각각 어플리케이션, 컴퓨팅 장치, 및/또는 네트워크를 통해 제공된 정보를 발생 또는 수신할 수 있는 다른 컴포넌트를 포함할 수 있다.

[0023] 어떤 구현에서, 시스템(100)은 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고 정보의 안전한 송신을 위해 각기 다른 암호 체계를 사용하여 슬라이스를 암호화하도록 프로그램되는 송신 컴퓨터(104)를 포함할 수 있다. 예를 들어, 송신 컴퓨터(104)는 각각의 데이터 패킷이 송신 컴포넌트(108a)와 관련된 정보의 부분, 송신 컴포넌트(108b)와 관련된 정보의 부분, 및 송신 컴포넌트(108c)와 관련된 정보의 부분을 포함하는 데이터 패킷을 발생할 수 있다. 다른 수의 송신 컴포넌트(108) 및 관련된 정보가 역시 사용될 수 있다. 정보의 각 부분은 데이터 패킷 내의 각 데이터 슬라이스가 정보의 또 하나의 데이터-슬라이스-표현 부분에 관련된 또 하나의 송신 컴포넌트(108)와 다른 송신 컴포넌트(108)와 관련된 정보의 부분을 나타내고/나타내거나, 데이터 패킷 내의 각 데이더 슬라이스가 데이터 패킷 내의 또 하나의 데이터 슬라이스를 암호화하는데 사용되는 또 하나의 암호 체계와 다른 암호 체계로 암호화될 수 있는 암호화된 데이터 슬라이스로서 데이터 패킷 내에 표현될 수 있다. 각기 다른 암호 체계는, 예를 들어, 각기 다른 암호화 타입 및/또는 각기 다른 암호화 키의 사용을 포함할 수 있다. 암호화 타입의 예는 원 타임 패드(OTP), 데이터 암호화 표준(DES), 진보된 암호화 표준(AES), 및/또는 기타 암호화 타입을 포함할 수 있다. 여기에 사용된 바와 같이, 데이터 패킷은 통신의 단위를 지칭할 수 있다. 데이터 패킷은, 예를 들어, 데이터그램, 세그먼트, 블럭, 셀, 프레임, 또는 네트워크를 통해 송신되는 통신의 다른 단위를 지칭할 수 있다.

[0024] 어떤 구현에서, 수신 컴퓨터(106)는 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하도록 프로그램될 수 있다. 예를 들어, 송신 컴퓨터(104) 및 수신 컴퓨터(106)는 각 데이터 패킷이 각기 다른 정보와 관련되고 각기 다른 암호 체계를 사용하여 암호화된 데이터 슬라이스를 포함하는 데이터 패킷을 통해 서로 통신하도록 사전 프로그램될 수 있다. 어떤 시나리오에서, 송신 컴퓨터(104) 및 수신 컴퓨터(106)의 사전 설정은 2개의 컴퓨터(104 및 106) 간의 유일한 관계를 설정할 수 있다. 2개의 컴퓨터 간의 관계는, 예를 들어: (i) 2개의 컴퓨터에 사용가능한 정책; (ii) 패드(예를 들어, OTP)의 집합 또는 헤더, 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 암호화하는데 사용되는 다른 암호 체계; (iii) 관계의 상태(예를 들어, 다음의 데이터 패킷 내의 헤더가 위치하여야 하는 곳, 어느 패드가 현재 사용되어야 하는지, 또는 기타 상태 정보); 또는 (iv) 관계의 다른 특성을 정의할 수 있다. 송신 컴퓨터(104)가 다른 컴퓨터와 다른 관계를 가질 수 있지만, 다른 컴퓨터는 각기 다른 정책, 암호 체계, 관계 상태, 또는 다른 관계 특성을 정의하는 다른 관계들의 결과로서 송신 컴퓨터(104)로부터 수신 컴퓨터(106)로 향하는 데이터 패킷을 처리할 수 없다. 한편, 수신 컴퓨터(106)는 각기 다른 송신 컴포넌트(108)와 관련된 정보의 부분들을 얻고 얻어진 부분들을 조합하여 이후에 적절한 수신 컴포넌트(110)에 제공되는 정보를 형성하기 위해 송신 컴퓨터(104)로부터의 데이터 패킷을 처리할 수 있다.

[0025] 특정한 송신 컴포넌트(108), 특정한 수신 컴포넌트(110), 송신 컴퓨터(104), 및 수신 컴퓨터(106)는 하나 이상의 무력화된 인터페이스를 갖고/갖거나 컴포넌트가 여기에 설명된 바와 같이 손상되었는지에 관계없이 시스템에 의해 교체된 컴포넌트를 각각 포함하거나 포함하지 않을 수 있다는 점에 주목하여야 한다. 이와 같이, 시스템은 단독으로 또는 여기에 또한 설명되는 하나 이상의 다른 보호 대책과 조합하여 여기에 설명된 바와 같이 네트워크를 통해 송신될 정보를 안전하게 하고/하거나 난독화할 수 있다.

[0026] 어떤 구현에서, 컴포넌트 관리자 서브시스템(112)은 송신 컴퓨터(104)와 관련된 소프트웨어 컴포넌트 및/또는 수신 컴퓨터(106)와 관련된 소프트웨어 컴포넌트를 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하도록 프로그램될 수 있다. 예를 들어, 교체 시간 간격의 경과 시에, 컴포넌트 관리자 서브시스템(112)은 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고 각기 다른 암호 체계를 사용하여 슬라이스의 암호화를 수행하는 송신 컴퓨터(104)에서의 컴포넌트(예를 들어, 정보 송신 서브시스템)를 교체할 수 있다. 교체 시간 간격의 경과 시에, 컴포넌트 관리자 서브시스템(112)은 또한 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷의 처리를 수행하는 수신 컴퓨터(106)에서의 컴포넌트(예를 들어, 정보 수신 서브시스템)를 교체할 수 있다.

[0027] 송신 컴퓨터(104)에서의 컴포넌트 및 수신 컴포넌트에서의 컴포넌트의 교체 컴포넌트는 각각 각 데이터 패킷이 각기 다른 정보와 관련되고 각기 다른 암호 체계를 사용하여 암호화된 데이터 슬라이스를 포함하는 데이터 패킷을 통해 서로 통신하도록 사전 프로그램될 수 있다. 교체 컴포넌트의 사전 설정은, 예를 들어, 교체된 컴포넌트들 간의 관계와 동일한 관계, 또는 교체 컴포넌트들 간의 관계와 다른 유일한 관계를 설정할 수 있다. 교체 컴포넌트 및/또는 다른 양호한 것으로 알려진 소프트웨어 컴포넌트가 송신 컴퓨터(104), 수신 컴퓨터(106), 그들 각각의 컴포넌트, 또는 다른 컴포넌트가 실행하고 있는 런타임 환경과 분리된 컴포넌트 저장소(114)로부터 얻어질 수 있다. 어떤 시나리오에서, 사전 설정된 관계를 갖는 양호한 것으로 알려진 소프트웨어 컴포넌트로의 컴포넌트의 정기적인 교체가 수행될 수 있으므로, 정책, 암호 체계, 및 관계 상태가 정기적으로 변화하게 함으로써, 컴포넌트 및 정보의 더 나은 보안 및 무결성을 제공한다.

[0028] 어떤 구현에서, 컴포넌트 관리자 서브시스템(112)은 교체된 컴포넌트를 교체된 컴포넌트가 이전에 실행하였던 런타임 환경과 분리된 분석 환경으로 이동시키도록 프로그램될 수 있다. 교체된 컴포넌트가 분석 환경으로 이동된 후에, 컴포넌트 분석 서브시스템(116)은 교체된 컴포넌트가 손상되었는지 및/또는 어떻게 손상되었는지를 결정하기 위해 교체된 컴포넌트를 분석하도록 프로그램될 수 있다. 이 방식으로, 다른 이점들 중에서, 손상에 보다 강한 소프트웨어 컴포넌트의 새로운 버전이 덜 강한 대응하는 소프트웨어 컴포넌트를 교체하도록 개발될 수 있다.

[0029] 도 1에 도시한 바와 같이, 시스템(100)은 클라이언트 컴퓨터(120)(또는 다수의 클라이언트 컴퓨터(120))를 더 포함할 수 있다. 클라이언트 컴퓨터(120)는 임의 형태의 이동 단말기, 고정 단말기, 또는 다른 장치를 포함할 수 있다. 예를 들어, 클라이언트 컴퓨터(120)는 데스크탑 컴퓨터, 노트북 컴퓨터, 넷북 컴퓨터, 태블릿 컴퓨터, 스마트폰, 내비게이션 장치, 전자 책 장치, 게임기, 또는 다른 클라이언트 컴퓨터를 포함할 수 있다. 사용자는, 예를 들어, 하나 이상의 클라이언트 컴퓨터(120)를 이용하여 송신 컴퓨터(104), 수신 컴퓨터(106), 송신 컴포넌트(108), 수신 컴포넌트(110), 또는 시스템(100)의 다른 컴포넌트와 상호작용할 수 있다. 다른 예로서, 클라이언트 컴퓨터(120)는 송신 컴퓨터(104), 수신 컴퓨터(106), 또는 시스템(100)의 다른 컴포넌트를 포함할 수 있다.

[0030] 다양한 시스템 컴포넌트 및 그들의 전반적인 기능에 대해 설명하였지만, 이제부터는 주의를 각 보호 대책의 보다 상세한 설명으로 돌릴 것이다. 여기에 설명된 바와 같이, 시스템은 이러한 보호 대책을 개별로 단독으로 또는 하나 이상의 다른 보호 대책과 조합하여 사용할 수 있다.

[0031] **컴포넌트를 대응하는 양호한 것으로 알려진 컴포넌트로 교체**

[0032] 도 2는 본 발명의 양태에 따른, 서브시스템, 저장소, 또는 다른 컴포넌트가 존재할 수 있는 환경(102)의 예시도 및 환경(102)에 관련한 동작의 플로우차트이다. 도시한 바와 같이, 환경(102)은 컴포넌트 관리자 서브시스템(112), 컴포넌트 저장소(114), 컴포넌트 분석 서브시스템(116), 컴포넌트 생성 서브시스템(118), 또는 다른 서브시스템, 저장소, 또는 컴포넌트를 포함할 수 있다. 도 2가 분리된 환경(102)을 나타내지만, 컴포넌트 관리자 서브시스템(112), 컴포넌트 저장소(114), 컴포넌트 분석 서브시스템(116), 및 컴포넌트 생성 서브시스템(118) 중 하나 이상이 동일한 환경 또는 분리된 환경에 존재할 수 있다는 점에 주목하여야 한다. 여기에 설명된 바와 같이, 한 환경에서의 서브시스템 또는 컴포넌트는 다른 환경에 접근하는데 제한이 있을 수 있거나 접근하지 못한다(예를 들어, 판독 접근 제한, 기입 접근 제한, 실행 접근 제한, 삭제 접근 제한, 및/또는 다른 접근 제한)는 것은 유리할 수 있다.

[0033] 어떤 구현에서, 보안 및/또는 무결성이 런타임 환경에서 실행하고 있는 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체함으로써 가능해질 수 있다. 소프트웨어 컴포넌트의 교체는 런타임 환경에서 실행하고 있는 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 수행될 수 있다. 예를 들어, 소프트웨어 컴포넌트(예를 들어, 컴포넌트(202a, 202b), 또는 다른 컴포넌트)의 교체와 관련한 동작은 동작(204, 206, 208, 210, 212, 214) 중 하나 이상, 또는 이후 더 상세히 설명되는 다른 동작을 포함할 수 있다.

[0034] 한 구현에서, 컴포넌트 관리자 서브시스템(112)은 소프트웨어 컴포넌트를 양호한 것으로 알려진 소프트웨어 컴포너트로 교체하도록 프로그램될 수 있다. 컴포넌트 관리자 서브시스템(112)은 런타임 환경(102a)에서 실행하고 있는 적어도 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 나타내는 적어도 제1 이벤트를 결정할 수 있다. 컴포넌트 관리자 서브시스템(112)은 제1 이벤트에 기초하여 제1 소프트웨어 컴포넌트를 제1 소프트웨어 컴포넌트에 대응하는 제2 소프트웨어 컴포넌트로 교체할 수 있다. 제1 소프트웨어 컴포넌트는 제2 소프트웨어 컴포넌트가 제1 이벤트 이후에 런타임 환경(102a)에서 사용가능하고 제1 소프트웨어 컴포넌트가 제1 이벤트 이

후에 런타임 환경(102a)에서 더 이상 사용가능하지 않도록 제2 소프트웨어 컴포넌트로 교체될 수 있다. 제2 소프트웨어 컴포넌트는 런타임 환경(102a)과 분리된 소프트웨어 저장소(114)로부터 컴포넌트 관리자 서브시스템(112)에 의해 얻어질 수 있다(예를 들어, 컴포넌트 저장소(114)는 런타임 환경(102a)과 분리된 환경(102c)에 존재할 수 있다).

[0035] 한 예로서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 이 둘 다의 소프트웨어 컴포넌트가 동일한 소프트웨어 컴포넌트의 인스턴스이므로 서로 대응할 수 있고, 제2 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트의 갱신된 버전이고, 또는 제2 소프트웨어 컴포넌트는 그렇지 않은 경우에 제1 소프트웨어 컴포넌트를 위한 교체 컴포넌트이다. 런타임 환경(102a)에 놓이고 실행되기 전에, 제1 소프트웨어 컴포넌트는 또한 컴포넌트 저장소(114)에 양호한 것으로 알려진 소프트웨어 컴포넌트로서 저장되어 있을 수 있다.

[0036] 어떤 구현에서, 도 2와 관련하여, 제1 소프트웨어 컴포넌트는 컴포넌트(202a)일 수 있고 제2 소프트웨어 컴포넌트는 컴포넌트(202b)일 수 있다. 컴포넌트(202a 및 202b)는 컴포넌트 생성 서브시스템(118)에 의해 컴포넌트 생성 환경(102d)에서 생성된 동일한 소프트웨어 컴포넌트의 대응하는 인스턴스일 수 있다. 동작(204 및 206)에서, 대응하는 컴포넌트(202a 및 202b)는 교체 컴포넌트로서 컴포넌트 저장소(114)에 부가될 수 있다. 동작(208)에서, 컴포넌트(202a)는 런타임 환경(102a)에서 실행하고 있는 또 하나의 대응하는 컴포넌트를 교체하기 위해서 컴포넌트 관리자 서브시스템(112)에 의해 얻어질 수 있고, 그 후 컴포넌트(202a)가 다른 대응하는 컴포넌트 대신에 런타임 환경(102)에서 실행한다. 동작(210)에서, 컴포넌트(202b)는 컴포넌트 관리자 서브시스템(112)에 의해 얻어질 수 있고 런타임 환경(102a)에서 실행하고 있는 컴포넌트(202a)를 위한 교체 컴포넌트로서 런타임 환경(102a)에 놓여진다.

[0037] 제1 이벤트(제1 소프트웨어 컴포넌트의 교체의 기초가 됨)는 제1 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 표시할 수 있다. 어떤 구현에서, 제1 이벤트는 하나 이상의 교체 시간 간격의 경과, 제1 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과하는 것, 제1 소프트웨어 컴포넌트의 갱신된 버전의 사용가능성, 및/또는 다른 이벤트를 포함할 수 있다.

[0038] 어떤 구현에서, 제1 소프트웨어 컴포넌트는 하나 이상의 교체 시간 간격의 경과 이후에 교체될 수 있고, 제2 소프트웨어 컴포넌트는 하나 이상의 교체 시간 간격의 제2 경과 이후에 교체될 수 있다. 한 예로서, 런타임 환경(102a)에서 실행하고 있는 소프트웨어 컴포넌트는 주기적으로(예를 들어, 매일, 매시, 매분, 매초 등) 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체될 수 있다. 다른 사용의 경우에, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과한 후에 교체될 수 있다. 이 방식으로, 다른 이점들 중에서, 런타임 환경(102a)에서 실행하고 있는 소프트웨어 컴포넌트의 무결성이 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 정기적으로 교체함으로써 유지될 수 있다. 이와 같이, 런타임 환경(102a)에서 실행하고 있는 소프트웨어 컴포넌트가 손상되더라도, 손상된 소프트웨어 컴포넌트와 관련된 어떤 부정적 효과가 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 소프트웨어 컴포넌트의 대응하는 양호한 것으로 알려진 컴포넌트로의 정기적인 교체를 통해 감소될 수 있다.

[0039] 다른 사용의 경우에, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트의 갱신된 버전(예를 들어, 제2 소프트웨어 컴포넌트)이 사용가능하게 된 이후에 교체될 수 있다. 그러므로, 다른 이점들 중에서, 런타임 환경(102a)은 멀웨어 또는 다른 보안 문제에 의한 공격에 보다 더 강한 소프트웨어 컴포넌트의 버전을 포함하는, 사용가능한 소프트웨어 컴포넌트의 가장 최근의 버전을 반영할 수 있다.

[0040] 다양한 구현에서, 컴포넌트 관리자 서브시스템(112)은 런타임 환경(102a)에서 제2 소프트웨어 컴포넌트를 인스턴스화하도록 프로그램될 수 있다. 제2 소프트웨어 컴포넌트는, 예를 들어, 제2 소프트웨어 컴포넌트가 런타임 환경(102a)에서 사용가능하지 않도록 비활성 경로에서 인스턴스화될 수 있다. 제1 소프트웨어 컴포넌트의 제2 소프트웨어 컴포넌트로의 교체는 제2 소프트웨어 컴포넌트가 런타임 환경(102a)에서 사용가능하게 되도록 컴포넌트 관리자 서브시스템(112)이 제2 소프트웨어 컴포넌트를 활성 경로에 놓는 것을 포함할 수 있다. 제1 소프트웨어 컴포넌트의 교체는 컴포넌트 관리자 서브시스템(112)이 제1 소프트웨어 컴포넌트를 활성 경로 밖으로 이동시키는 것을 더 포함할 수 있다.

[0041] 예를 들어, 런타임 환경(102a)은 특정한 소프트웨어 컴포넌트의 하나 이상의 활성 인스턴스 및 소프트웨어 컴포넌트의 하나 이상의 비활성 인스턴스를 포함할 수 있다. 소프트웨어 컴포넌트의 비활성 인스턴스는, 예를 들어, 소프트웨어 컴포넌트의 활성 인스턴스의 교체일 수 있다. 이와 같이, 활성 인스턴스 중 하나의 교체가 일어날 때, 활성 인스턴스는 간단히 비활성 인스턴스 중 하나로 교체될 수 있다. 소프트웨어 컴포넌트의 부가

적인 대응하는 양호한 것으로 알려진 인스턴스가 컴포넌트 저장소(114)로부터 얻어지고 런타임 환경(102a)에 바로 사용가능한 교체 인스턴스를 공급하기 위해 비활성 인스턴스로서 런타임 환경(102a)에서 인스턴스화될 수 있다. 어떤 구현에서, 제1 소프트웨어 컴포넌트는 특정한 소프트웨어 컴포넌트의 활성 인스턴스일 수 있고, 제2 소프트웨어 컴포넌트는 소프트웨어 컴포넌트의 비활성 인스턴스일 수 있다. 제2 소프트웨어 컴포넌트는 활성 경로 내의 제1 소프트웨어 컴포넌트를 교체하기 위해 활성 경로 내로 순환될 수 있고, 제1 소프트웨어 컴포넌트는 활성 경로 밖으로 순환될 수 있다. 이와 같이, 제2 소프트웨어 컴포넌트는 제2 소프트웨어 컴포넌트를 활성 경로 내로 순환시킴으로써 활성화될 수 있고, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트를 활성 경로 밖으로 순환시킴으로써 비활성화될 수 있다.

[0042] 어떤 구현에서, 컴포넌트 관리자 서브시스템(112)은 제2 소프트웨어 컴포넌트를 활성 경로에 놓는 것과 동시에 또는 그 이후에 제1 소프트웨어 컴포넌트를 활성 경로 밖으로 이동시키도록 프로그램될 수 있다. 한 예로서, 제1 소프트웨어 컴포넌트의 제2 소프트웨어 컴포넌트로의 교체는 교체 전에 제1 소프트웨어 컴포넌트에 의해 이전에 제공된 연속적인 원활한 동작에 의존할 수 있는 사용자 또는 다른 컴포넌트에 의해 다운 시간이 경험되지 않도록 끊김없는 방식으로 수행될 수 있다.

[0043] 어떤 구현에서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 나타내는 제1 이벤트 시에 제1 소프트웨어 컴포넌트에 의해 취급되고 있던 작업을 완료하도록 허용될 수 있다. 그러나, 새로운 작업이 취급할 제1 소프트웨어 컴포넌트에 할당되지 않고, 대신에 교체-표시 이벤트가 아직 일어나지 않은 제1 소프트웨어 컴포넌트에 대응하는 다른 활성 소프트웨어 컴포넌트(예를 들어, 제1 소프트웨어 컴포넌트의 다른 활성 인스턴스)에 할당될 수 있다. 제1 이벤트 시에 제1 소프트웨어 컴포넌트에 의해 취급되고 있는 작업의 완료 시에, 컴포넌트 관리자 서브시스템은 제1 소프트웨어 컴포넌트를 활성 경로 밖으로 이동시키는 동안 제2 소프트웨어 컴포넌트를 동시에 활성 경로에 놓을 수 있다. 새로운 작업은 다음에 제2 소프트웨어 컴포넌트에 대응하는 다른 활성 소프트웨어 컴포넌트(제1 소프트웨어 컴포넌트 또는 제2 소프트웨어 컴포넌트의 다른 활성 인스턴스) 뿐만 아니라 제2 소프트웨어 컴포넌트에 할당될 수 있다.

[0044] 소정의 구현에서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 각각 가상 머신을 포함할 수 있다. 소프트웨어 컴포넌트의 정의는 하나 이상의 디스크 영상을 포함할 수 있다. 런타임 환경(102a)은 제1 소프트웨어 컴포넌트가 교체 전에 실행하고 제2 소프트웨어 컴포넌트가 교체 후에 실행하는 하이퍼바이저를 포함할 수 있다. 제2 소프트웨어 컴포넌트를 활성 경로에 놓는 것은 제2 소프트웨어 컴포넌트가 런타임 환경에서 사용가능하도록 컴포넌트 관리자 서브시스템(112)이 제2 소프트웨어 컴포넌트와 관련된 네트워크 설정을 갱신하는 것을 포함할 수 있다.

[0045] 어떤 구현에서, 하나 이상의 외부 IP 어드레스는 이 외부 IP 어드레스로 다이렉트된 요구가 활성 서브넷 어드레스에 할당된 소프트웨어 컴포넌트로 리다이렉트되도록 하나 이상의 활성 서브넷 어드레스와 관련될 수 있다. 활성 서브넷 어드레스에 할당된 소프트웨어 컴포넌트는 런타임 환경(102a)에서 사용가능하게 될 수 있고, 비활성 서브넷 어드레스에 할당된 소프트웨어 컴포넌트는 런타임 환경(102a)에서 사용 가능하게 될 수 없다. 제1 소프트웨어 컴포넌트는 교체 전에 활성 서브넷 어드레스 중 하나에 할당될 수 있고, 제2 소프트웨어 컴포넌트는 교체 전에 비활성 서브넷 어드레스 중 하나에 할당될 수 있다. 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 나타내는 제1 이벤트에 응답하여, 제2 소프트웨어 컴포넌트에 할당된 비활성 서브넷 어드레스는 활성화될 수 있고, 또는 다른 활성 서브넷 어드레스가 제2 소프트웨어 컴포넌트에 할당될 수 있으므로, 외부 IP 어드레스로 다이렉트된 요구는 그것의 할당된 활성 서브넷 어드레스를 통해 제2 소프트웨어 컴포넌트로 리다이렉트될 수 있다.

[0046] 다양한 구현에서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 각각 운영 체제 처리를 포함할 수 있다. 제2 소프트웨어 컴포넌트를 활성 경로에 놓는 것은 컴포넌트 관리자 서브시스템(112)이 제2 소프트웨어 컴포넌트를 처리 간 통신 설정을 통해 활성 경로로 이동시키는 것을 포함할 수 있다.

[0047] 어떤 구현에서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 각각 실행가능한 코드를 포함할 수 있다. 제2 소프트웨어 컴포넌트를 활성 경로에 놓는 것은 컴포넌트 관리자 서브시스템(112)이 제2 소프트웨어 컴포넌트와 관련된 메모리 어드레스 포인터 또는 룩업 테이블을 갱신하는 것을 포함할 수 있다. 예를 들어, 언어 라이브러리는 언어 라이브러리에서 사용가능한 기능(예를 들어, 소프트웨어 컴포넌트)과 관련된 메모리 어드레스 포인터 및 룩업 테이블을 이용하여 기능 호출을 가능하게 할 수 있다. 메모리 어드레스 포인터 또는 룩업 테이블은, 예를 들어, 교체 실행가능한 코드가 존재하는 메모리 내의 다른 위치를 포인트하도록 각각의 기능 호출 이후에 갱신될 수 있다. 한 시나리오에서, 제1 소프트웨어 컴포넌트는 언어 라이브러리 내의 메모리 어드레

스 포인터 또는 룩업 테이블이 포인트하는 제1 메모리 위치에 있는 특정한 기능과 관련된 실행가능한 코드일 수 있다. 언어 라이브러리는 소프트웨어 컴포넌트가 교체 소프트웨어 컴포넌트(예를 들어, 기능의 교체 실행가능한 코드)로 교체되기 전에 소프트웨어 컴포넌트(예를 들어, 기능과 관련된 실행가능한 코드)가 한 번만 호출될 수 있도록 설정될 수 있다. 이와 같이, 제1 소프트웨어 컴포넌트가 한번 호출된 후에, 제1 메모리 위치를 포인트하는 메모리 어드레스 포인터 또는 룩업 테이블은 기능과 관련된 교체 실행가능한 코드가 존재하는 제2 메모리 위치를 포인트하도록 갱신될 수 있다.

[0048] 소정의 구현에서, 컴포넌트 관리자 서브시스템(112)은 제2 소프트웨어 컴포넌트와 관련된 무결성 정보를 얻도록 프로그램될 수 있다. 제2 소프트웨어 컴포넌트는 무결성 정보에 기초하여 컴포넌트 관리자 서브시스템(112)에 의해 유효화될 수 있다. 제2 소프트웨어 컴포넌트의 유효화는, 예를 들어, 컴포넌트 관리자 서브시스템(112)이 제2 소프트웨어 컴포넌트를 이용하여 제1 소프트웨어 컴포넌트를 교체하기 전에 수행될 수 있다. 어떤 구현에서, 무결성 정보는 제2 소프트웨어 컴포넌트가 양호한 것으로 알려진 소프트웨어 컴포넌트인 동안에 제2 소프트웨어 컴포넌트에 대해 발생될 수 있다. 한 예로서, 무결성 정보는 제2 소프트웨어 컴포넌트가 분배 준비된 바로 직후에, 제2 소프트웨어 컴포넌트가 그것의 생성 환경, 또는 제2 소프트웨어 컴포넌트가 양호한 것으로 알려진 소프트웨어 컴포넌트로 남아 있는 다른 환경 밖으로 분배되기 전에 제2 소프트웨어 컴포넌트에 대해 발생될 수 있다. 무결성 정보는, 예를 들어, 제2 소프트웨어 컴포넌트가 그것의 양호한 것으로 알려진 형태로부터 수정되지 않았다는 것을 결정하는데 사용될 수 있다. 어떤 시나리오에서, 무결성 정보는 제2 소프트웨어 컴포넌트에 대한 해시 동작(예를 들어, 일방향 해시 또는 다른 해시 동작)을 수행함으로써 발생될 수 있다.

[0049] 어떤 구현에서, 컴포넌트 분석 서브시스템(116)은 제1 소프트웨어 컴포넌트가 교체된 후에 제1 소프트웨어 컴포넌트를 분석하도록 프로그램될 수 있다. 컴포넌트 분석 서브시스템(116)은 이 분석에 기초하여 제1 소프트웨어 컴포넌트가 손상되었는지 및/또는 얼마나 손상되었는지를 결정하도록 프로그램될 수 있다. 예를 들어, 제1 소프트웨어 컴포넌트가 활성 경로 밖으로 이동된 후에, 컴포넌트 관리자 서브시스템(112)은 제1 소프트웨어 컴포넌트를 분석 환경(102e)으로 이동시킬 수 있다. 표시된 바와 같이, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 제2 소프트웨어 컴포넌트로 교체될 수 있다. 그럼에도 불구하고, 컴포넌트 분석 서브시스템(116)은 제1 소프트웨어 컴포넌트를 분석하여 제1 소프트웨어 컴포넌트가 손상되었는지, 만약 그렇다면, 제1 소프트웨어 컴포넌트가 얼마나 손상되었는지를 결정할 수 있다. 이 방식으로, 다른 이점들 중에서, 손상에 보다 강한 소프트웨어 컴포넌트의 새로운 버전이 덜 강한 대응하는 소프트웨어 컴포넌트를 교체하도록 개발될 수 있다.

[0050] 어떤 구현에서, 도 2와 관련하여, 컴포넌트(202a)는 대응하는 컴포넌트(202b)가 런타임 환경(102a)에서 컴포넌트(202a)를 교체한 후에 동작(212)에서 컴포넌트 관리자 서브시스템(112)에 의해 런타임 환경(102)으로부터 분석 환경(102e)으로 이동될 수 있다. 다른 사용의 경우에, 컴포넌트(202b)는 또한 또 하나의 대응하는 컴포넌트가 런타임 환경(102b)에서 컴포넌트(202b)를 교체한 후에 동작(214)에서 컴포넌트 관리자 서브시스템(112)에 의해 런타임 환경(102a)으로부터 분석 환경(102e)으로 이동될 수 있다.

[0051] 논의된 바와 같이, 환경(102)은 하나 이상의 분리된 환경(102a, 102b, 102c, 102d, 102e), 또는 다른 환경을 포함할 수 있다. 한 환경에서의 서브시스템 또는 컴포넌트는 다른 환경에 접근하는 것이 제한될 수 있다(예를 들어, 판독 접근 제한, 기입 접근 제한, 실행 접근 제한, 삭제 접근 제한, 또는 다른 접근 제한).

[0052] 예를 들어, 컴포넌트 관리자 서브시스템(112)(예를 들어, 환경(102b)에 위치함)은 컴포넌트 저장소(114)로부터 제2 소프트웨어 컴포넌트를 얻기 위해 컴포넌트 저장소(114)(예를 들어, 환경(102c)에 위치함)에 판독 접근할 수 있다. 그러나, 환경(102b), 환경(102c), 컴포넌트 저장소(114), 또는 컴포넌트 관리자 서브시스템(112)은 컴포넌트 관리자 서브시스템(112)이 컴포넌트 저장소(114) 또는 환경(102c)에 기입 접근, 실행 접근, 또는 삭제 접근하지 못하도록 설정될 수 있다. 그러므로, 컴포넌트 관리자 서브시스템(112)은 컴포넌트 저장소(114)에 저장된 양호한 것으로 알려진 소프트웨어 컴포넌트를 의도적으로 또는 우연히 손상시키는데 이용될 수 없다.

[0053] 다른 예를 들면, 컴포넌트 생성 서브시스템(118)(예를 들어, 환경(102d)에 위치함)은 컴포넌트 저장소(114)에 제2 소프트웨어 컴포넌트 또는 다른 소프트웨어 컴포넌트를 부가하기 위해 컴포넌트 저장소(114)(예를 들어, 환경(102c)에 위치함)에 기입 접근할 수 있다. 그러나, 환경(102c), 환경(102d), 컴포넌트 생성 서브시스템(118), 또는 컴포넌트 저장소(114)는 컴포넌트 생성 서브시스템(118)가 컴포넌트 저장소(114) 또는 환경(102c)에 판독 접근, 실행 접근, 또는 삭제 접근하지 못하도록 설정될 수 있다. 어떤 시나리오에서, 환경(102b, 102a, 102c, 또는 102e)에서의 서브시스템 또는 다른 컴포넌트는 컴포넌트 생성 서브시스템(118) 또는 환경(102d)에 기입 접근, 판독 접근, 실행 접근, 또는 삭제 접근하지 못한다. 예를 들어, 컴포넌트 생성 서브시스

템(118)은 컴포넌트 생성 서브시스템(118)이 런타임 환경(102a)으로부터 손상될 수 없도록 런타임 환경(102a)으로부터 접근하지 못한다. 이 방식으로, 컴포넌트 생성 서브시스템(118)은 생성된 소프트웨어 컴포넌트가 컴포넌트 저장소(114)에 놓일 때 소프트웨어 컴포넌트가 양호한 것으로 알려진 소프트웨어 컴포넌트이도록 손상 없는 환경에서 소프트웨어 컴포넌트를 생성할 수 있다.

[0054] 어떤 구현에서, 보안 및/또는 무결성이 컴포넌트 생성 환경과 분리된 컴포넌트 저장소에 소프트웨어 컴포넌트를 놓기 전에 컴포넌트 생성 환경에서 생성되는 소프트웨어 컴포넌트에 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한함으로써 가능하게 될 수 있다. 예를 들어, 컴포넌트 생성 서브시스템(118)은 컴포넌트 생성 환경(102d)에서 소프트웨어 컴포넌트를 생성하고, 소프트웨어 컴포넌트가 컴포넌트 생성 환경(102d) 밖으로 이동되기 전에 소프트웨어 컴포넌트에서 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하도록 프로그램될 수 있다(예를 들어, 소프트웨어 컴포넌트가 컴포넌트 저장소(114)에 놓이기 전에, 소프트웨어 컴포넌트는 런타임 환경(102a) 등에서 인스턴스화된다).

[0055] **인터페이스 및 다른 서브 컴포넌트를 제한/무력화**

[0056] 한 구현에서, 도 3에 도시한 바와 같이, 컴포넌트 생성 서브시스템(118)은 컴포넌트 작성 서브시스템(302), 컴포넌트 실링 서브시스템(304), 컴포넌트 엑스포팅 서브시스템(306), 및/또는 다른 서브시스템을 포함할 수 있다.

[0057] 어떤 구현에서, 컴포넌트 실링 서브시스템(304)은 컴포넌트 생성 환경(102d)에서 적어도 제1 소프트웨어 컴포넌트를 식별할 수 있다. 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트, 제2 서브 컴포넌트, 또는 다른 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 기능을 제공하는 제1 서브 컴포넌트를 포함할 수 있다. 컴포넌트 실링 서브시스템(304)은 제1 서브 컴포넌트에 의해 제공된 기능이 제1 소프트웨어 컴포넌트가 실행될 때 제1 소프트웨어 컴포넌트를 통해 사용가능하지 않도록 제1 서브 컴포넌트를 식별 및 무력화할 수 있다. 컴포넌트 엑스포팅 서브시스템(306)은 제1 소프트웨어 컴포넌트가 제1 서브 컴포넌트에 의해 제공된 기능이 사용될 가능성 없이 컴포넌트 저장소(114)에 놓이도록 제1 서브 컴포넌트가 무력화된 이후에 제1 소프트웨어 컴포넌트를 컴포넌트 저장소(114)에 놓을 수 있다. 이 방식으로, 공격 가능하거나 잠재적으로 공격 가능한 기능의 사용가능성을 제1 소프트웨어 컴포넌트로부터 제거함으로써, 제1 소프트웨어 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 기능에 접근하는데 사용될 수 없다. 다른 이점들 중에서, 제1 소프트웨어 컴포넌트가 손상되는 것과 관련된 위험이 감소될 수 있다.

[0058] 한 구현에서, 제1 서브 컴포넌트의 무력화는 컴포넌트 실링 서브시스템(304)이 제1 소프트웨어 컴포넌트로부터 제1 서브 컴포넌트를 제거하는 것을 포함할 수 있다. 이와 같이, 제1 소프트웨어 컴포넌트가 컴포넌트 저장소(114)에 놓일 때, 제1 소프트웨어 컴포넌트는 제1 서브 컴포넌트를 포함하지 않을 수 있다. 제1 소프트웨어 컴포넌트로부터의 제1 서브 컴포넌트의 제거는 제1 소프트웨어 컴포넌트가 제1 서브 컴포넌트와 관련된 소스 코드 또는 실행가능한 코드를 더 이상 포함하지 않도록 제1 소프트웨어 컴포넌트로부터 제1 서브 컴포넌트와 관련된 소스 코드 또는 실행가능한 코드를 제거하는 것을 포함할 수 있다. 어떤 구현에서, 제1 소프트웨어 컴포넌트는 가상 머신일 수 있고, 제1 서브 컴포넌트는 보안 셸(SSH) 프로그램일 수 있다. 제1 소프트웨어 컴포넌트로부터의 제1 서브 컴포넌트의 제거는 가상 머신으로부터 SSH 프로그램을 형성하는 실행가능한 코드를 제거하는 것을 포함할 수 있다.

[0059] 다른 구현에서, 제1 서브 컴포넌트의 무력화는 제1 소프트웨어 컴포넌트로부터 제1 서브 컴포넌트를 제거하지 않고서 제1 서브 컴포넌트에 의해 제공된 기능을 턴 오프하는 것을 포함할 수 있다. 그러므로, 컴포넌트 저장소(114) 내의 제1 소프트웨어 컴포넌트의 제1 서브 컴포넌트가 무력화될 수 있지만, 제1 소프트웨어 컴포넌트는 여전히 제1 서브 컴포넌트를 포함할 수 있다. 한 예로서, 제1 서브 컴포넌트는 제1 소프트웨어 컴포넌트에 대한 불안정성을 피하기 위해서 제1 소프트웨어 컴포넌트로부터 제1 서브 컴포넌트를 제거하지 않고서 제1 소프트웨어 컴포넌트에 대해 무력화될 수 있다.

[0060] 어떤 구현에서, 제1 서브 컴포넌트를 무력화하는 것은 제1 소프트웨어 컴포넌트를 무력화 명령과 관련시키는 것을 포함할 수 있다. 한 구현에서, 제1 소프트웨어 컴포넌트는 제1 서브 컴포넌트가 무력화 명령에 기초하여 제1 소프트웨어 컴포넌트로부터 제거되도록 무력화 명령과 관련될 수 있다. 한 예로서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트를 런타임 환경에서 인스턴스화하도록 실행될 스크립트(예를 들어, 배시(Bash) 스크립트 또는 다른 스크립트)와 관련될 수 있다. 제1 소프트웨어 컴포넌트가 인스턴스화될 때, 스크립트는 제1 서브 컴포넌트와 관련된 코드를 제1 소프트웨어 컴포넌트로부터 제거할 수 있다. 이와 같이, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트가 실행될 때 제1 서브 컴포넌트와 관련된 코드를 포함하지 않을 수 있다.

다른 예로서, 무력화 명령은 제1 소프트웨어 컴포넌트가 컴포넌트 저장소에 놓이거나 또는 런타임 환경에서 사용가능하게 되기 전에 제1 소프트웨어 컴포넌트 내로 주입될 수 있다. 제1 소프트웨어 컴포넌트의 실행 시에, 무력화 명령은 제1 서브 컴포넌트와 관련된 코드가 제1 소프트웨어 컴포넌트로부터 제거되게 할 수 있다.

[0061] 다른 구현에서, 제1 소프트웨어 컴포넌트는 제1 서브 컴포넌트에 의해 제공된 기능이 무력화 명령에 기초하여 턴 오프되도록 무력화 명령과 관련될 수 있다. 한 예로서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트를 런타임 환경에 인스턴트화하도록 실행될 스크립트(예를 들어, 배시 스크립트 또는 다른 스트립트)와 관련될 수 있다. 제1 소프트웨어 컴포넌트가 인스턴트화될 때, 스크립트는 제1 서브 컴포넌트에 의해 제공된 기능이 턴 오프되게 할 수 있다. 그러므로, 제1 서브 컴포넌트에 의해 제공된 기능은 제1 소프트웨어 컴포넌트가 실행될 때 제1 소프트웨어 컴포넌트를 통해 사용가능하지 않을 것이다. 다른 예로서, 무력화 명령은 제1 소프트웨어 컴포넌트가 컴포넌트 저장소에 놓이거나 또는 런타임 환경에서 사용가능하게 되기 전에 제1 소프트웨어 컴포넌트 내로 주입될 수 있다. 제1 소프트웨어 컴포넌트의 실행 시에, 무력화 명령은 제1 서브 컴포넌트에 의해 제공된 기능이 턴 오프되게 할 수 있다.

[0062] 다양한 구현에서, 제1 서브 컴포넌트에 의해 제공된 기능은 제1 소프트웨어 컴포넌트를 유지하는데 사용되는 유지 기능 또는 관리 기능을 포함할 수 있다. 제1 서브 컴포넌트는 제1 서브 컴포넌트에 의해 제공된 유지 기능 또는 관리 기능에 기초하여 식별 또는 무력화될 수 있다. 예를 들어, 가상 머신을 유지하는 유지 기능 또는 관리 기능을 포함하는 가상 머신의 서브 컴포넌트는 텔넷 인터페이스, SSH 인터페이스, 리모트 데스크탑 포로토콜(RDP) 인터페이스, 관리 어플리케이션 프로그래머블 인터페이스(API), 관리 웹사이트, 또는 다른 서브 컴포넌트를 포함할 수 있다.

[0063] 어떤 구현에서, 제1 소프트웨어 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 제2 서브 컴포넌트를 더 포함할 수 있다. 제1 소프트웨어 컴포넌트의 제1 서브 컴포넌트가 무력화될 수 있는 반면, 제2 서브 컴포넌트는 무력화될 수 없다. 예를 들어, 제1 서브 컴포넌트는 제1 서브 컴포넌트가 제1 소프트웨어 컴포넌트의 기능을 위해 필요하지 않다는 결정에 기초하여 컴포넌트 실링 서브시스템(304)에 의해 무력화될 수 있다. 제2 서브 컴포넌트는 제2 서브 컴포넌트가 제1 소프트웨어 컴포넌트의 기능을 위해 필요하다는 결정에 기초하여 컴포넌트 실링 서브시스템(304)에 의해 무력화될 수 없다.

[0064] 소정의 구현에서, 제2 서브 컴포넌트가 제1 소프트웨어 컴포넌트로부터 제거될 수 없거나 혹은 무력화될 수 없지만, 컴포넌트 실링 서브시스템(304)은 제2 서브 컴포넌트와 관련한 위험을 없애거나 혹은 감소시키기 위해 제2 서브 컴포넌트와 관련된 하나 이상의 코드 세트를 제2 서브 컴포넌트로부터 제거하도록 프로그램될 수 있다. 예를 들어, 컴포넌트 실링 서브시스템은 제2 서브 컴포넌트가 공격당하거나 잠재적으로 공격당하게 하는 제2 서브 컴포넌트와 관련된 코드 세트를 식별하고, 제2 서브 컴포넌트가 코드 세트를 더 이상 포함하지 않도록 코드 세트를 제2 서브 컴포넌트로부터 제거하도록 프로그램될 수 있다. 제1 소프트웨어 컴포넌트가 제2 환경에 놓일 때, 제1 소프트웨어 컴포넌트는 코드 세트 없이 제2 서브 컴포넌트를 포함할 수 있다.

[0065] 한 시나리오에서, 소프트웨어 컴포넌트는 가상 머신일 수 있다. 가상 머신의 서브 컴포넌트는 배시일 수 있다. 배시가 가상 머신의 운영 체제가 적절히 동작하는데 요구되는 셸이라는 결과로서 배시는 가상 머신으로부터 제거될 수 없거나 혹은 무력화될 수 없다. 그럼에도 불구하고, "ls", "sed", "vi" 등과 같은 많은 커맨드는 그들 커맨드가 가상 머신의 운영 체제가 적절히 동작하는데 요구되지 않는다는 결과로서 제거될 수 있거나 혹은 무력화될 수 있다. 예를 들어, 이들 요구되지 않는 커맨드와 관련된 코드 세트는 가상 머신이 이들 코드 세트 없이 배시를 포함하도록 배시로부터 제거될 수 있다.

[0066] 어떤 구현에서, 컴포넌트 작성 서브시스템(302)은 컴포넌트 생성 환경(102d)에서 제1 소프트웨어 컴포넌트를 발생하도록 프로그램될 수 있다. 컴포넌트 작성 서브시스템(302)에 의해 발생된 제1 소프트웨어 컴포넌트는 (제1 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 기능을 제공하는) 제1 서브 컴포넌트, 제2 서브 컴포넌트, 또는 다른 서브 컴포넌트를 포함할 수 있다.

[0067] 한 구현에서, 컴포넌트 작성 서브시스템(302)은 제1 프로필 정보를 제1 소프트웨어 컴포넌트와 관련시킬 수 있다. 예를 들어, 제1 프로필 정보는 특정한 소프트웨어 컴포넌트의 특정한 서브 컴포넌트가 무력화되어야 하는지를 결정하는데 사용된다. 제1 서브 컴포넌트는 제1 프로필 정보에 기초하여 식별되거나 무력화될 수 있다. 한 예로서, 제1 프로필 정보는 제거를 통해 무력화될 서브 컴포넌트를 지정하는 정보, 제거되지 않고 무력화될 서브 컴포넌트를 지정하는 정보, 무력화되지 않을 서브 컴포넌트를 지정하는 정보, 또는 다른 정보를 포함할 수 있다. 한 시나리오에서, 제1 프로필 정보는 소프트웨어 컴포넌트 내의 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하기 위해 일반적으로 적용될 수 있다. 다른 시나리오에서, 제1 프로필 정보는 하나

이상의 특정 타입의 소프트웨어 컴포넌트, 특정한 식별자와 관련된 소프트웨어 컴포넌트 등에서 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하기 위한 것일 수 있다. 예를 들어, 제1 프로필 정보는 제1 소프트웨어 컴포넌트의 소프트웨어 컴포넌트 타입을 갖는 소프트웨어 컴포넌트에 특정되거나, 또는 제1 소프트웨어 컴포넌트의 식별자에 대응하는 식별자를 갖는 소프트웨어 컴포넌트에 특정될 수 있다.

[0068] 다양한 구현에서, 컴포넌트 작성 서브시스템(302)은 제1 소프트웨어 컴포넌트가 제1 서브 컴포넌트 및 제2 서브 컴포넌트를 포함하도록 컴포넌트 생성 환경(102d)에서 제1 소프트웨어 컴포넌트를 발생하도록 프로그램될 수 있다. 제1 컴포넌트는 제1 프로필 정보에 기초하여 컴포넌트 실링 서브시스템(304)에 의해 무력화될 수 있다. 제2 컴포넌트는 제1 프로필 정보에 기초하여 컴포넌트 실링 서브시스템(304)에 의해 무력화될 수 없다.

[0069] 어떤 구현에서, 컴포넌트 엑스포팅 서브시스템(306)은 제1 소프트웨어 컴포넌트에 대한 무결성 정보를 발생하도록 프로그램될 수 있다. 예를 들면, 제1 소프트웨어 컴포넌트가 양호한 것으로 알려진 소프트웨어 컴포넌트인 동안에 무결성 정보는 제1 소프트웨어 컴포넌트에 대해 발생될 수 있다. 한 예로서, 무결성 정보는 제1 소프트웨어 컴포넌트가 분배 준비된 직후에, 제1 소프트웨어 컴포넌트가 그것의 생성 환경(예를 들어, 환경(102d)), 또는 제1 소프트웨어 컴포넌트가 양호한 것으로 알려진 소프트웨어 컴포넌트로 남는 다른 환경 밖으로 분배되기 전에, 제2 소프트웨어 컴포넌트에 대해 발생될 수 있다.

[0070] 어떤 구현에서, 무결성 정보는 제1 서브 컴포넌트가 무력화된 후에 그리고 제1 소프트웨어 컴포넌트가 컴포넌트 저장소(114)에 놓이기 전에 제1 소프트웨어 컴포넌트에 대해 발생될 수 있다. 무결성 정보는 무결정 정보가 제1 소프트웨어 컴포넌트에 대해 발생된 후에 제1 소프트웨어 컴포넌트가 수정되었는지를 결정하는데 사용될 수 있다. 어떤 구현에서, 무결성 정보는 제1 소프트웨어 컴포넌트에 대한 해시 동작(예를 들어, 일방향 해시 또는 다른 해시 동작)을 수행함으로써 발생될 수 있다.

[0071] **네트워크를 통해 송신될 정보를 인터리빙 및 난독화**

[0072] 도 4a를 참조하면, 어떤 구현에서, 송신 컴퓨터(104)는 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고/하거나 정보의 안전한 송신을 위해 각기 다른 암호 체계를 사용하여 슬라이스를 암호화하도록 프로그램될 수 있다. 예를 들어, 송신 컴퓨터(104)는 송신될 제1 정보, 송신될 제2 정보, 또는 송신될 다른 정보를 수신할 수 있다. 제1 정보는 제1 송신 컴포넌트와 관련될 수 있다. 제2 정보는 제1 송신 컴포넌트와 다른 제2 송신 컴포넌트와 관련될 수 있다. 송신 컴퓨터(108)는, 예를 들어, 송신 컴퓨터(104) 외부에 있는 컴포넌트, 송신 컴퓨터(104) 내에서 실행하고 있는 컴포넌트, 또는 다른 컴포넌트를 포함할 수 있다.

[0073] 송신 컴퓨터(104)는 제1 정보의 적어도 일부를 나타내는 제1 데이터 슬라이스, 제2 정보의 적어도 일부를 나타내는 제2 데이터 슬라이스, 또는 다른 데이터 슬라이스를 발생하도록 프로그램될 수 있다. 제1 데이터 슬라이스는 제1 암호 체계에 기초하여 발생될 수 있다. 제2 데이터 슬라이스는 제1 암호 체계와 다른 제2 암호 체계에 기초하여 발생될 수 있다. 한 예로서, 제1 암호 체계에 기초한 제1 데이터 슬라이스의 발생은 제1 암호화 타입 또는 제1 암호화 키를 사용하여 제1 정보의 부분을 암호화하는 것을 포함할 수 있다. 제2 암호 체계에 기초한 제2 데이터 슬라이스의 발생은 각각 제1 암호화 타입 또는 제1 암호화 키와 다른 제2 암호화 타입 또는 제2 암호화 키를 사용하여 제2 정보의 부분을 암호화하는 것을 포함할 수 있다. 암호화 타입의 예는 원타임 패드(OTP), 데이터 암호화 표준(DES), 진보된 암호화 표준(AES), 또는 다른 암호화 타입을 포함할 수 있다.

[0074] 송신 컴퓨터(104)는 제1 데이터 슬라이스용 제1 암호 체계, 제2 데이터 슬라이스용 제2 암호 체계, 또는 다른 정보를 지정하는 제1 헤더를 발생하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제1 헤더, 제1 데이터 슬라이스, 제2 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함하는 제1 데이터 패킷을 발생하도록 프로그램될 수 있다. 제1 데이터 패킷은 다음에 적절한 수신 컴퓨터(예를 들어, 수신 컴퓨터(106) 또는 다른 수신 컴퓨터)에 송신될 수 있다.

[0075] 어떤 구현에서, 도 4a와 관련하여, 송신 컴퓨터(104)는 송신 컴포넌트(108a-108n), 정보 송신 서브시스템(402), 또는 다른 컴포넌트를 포함할 수 있다. 다른 사용의 경우에, 도 4b와 관련하여, 수신 컴퓨터(106)는 수신 컴포넌트(110a-110n), 정보 수신 서브시스템(404), 또는 다른 컴포넌트를 포함할 수 있다. 상황에 따라, 특정한 송신 컴포넌트(108)는 또한 수신 컴포넌트일 수 있고, 특정한 수신 컴포넌트(110)는 또한 송신 컴포넌트일 수 있다는 점에 주목하여야 한다. 상황에 따라, 송신 컴퓨터(104)는 수신 컴퓨터일 수 있고, 수신 컴퓨터(106)는 송신 컴퓨터일 수 있다는 점에 또한 주목하여야 한다. 한 예로서, 송신 컴퓨터(104)는 정보 수신 서브시스템(404) 또는 수신된 데이터 패킷의 각각이 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 포함하는 수신된 데이터 패킷을 처리하는 다른 컴포넌트를 더 포함할 수 있다. 다른

예로서, 수신 컴퓨터(106)는 정보 송신 서브시스템(402) 또는 송신될 각기 다른 정보를 데이터 패킷의 각기 다른 슬라이스로 조합하고/하거나 정보의 안전한 송신을 위해 각기 다른 암호 체계를 사용하여 슬라이스를 암호화하는 다른 컴포넌트를 포함할 수 있다.

[0076] 도시된 바와 같이, 어떤 구현에서, 송신 컴퓨터(104) 및 수신 컴퓨터(106)는 데이터 패킷을 통해 서로 통신하도록 사전 프로그램될 수 있고 여기서 각 데이터 패킷은 각기 다른 정보와 관련되고 각기 다른 암호 체계를 사용하여 암호화된 데이터 슬라이스를 포함한다. 송신 컴퓨터(104) 및 수신 컴퓨터(106)의 사전 설정은, 예를 들어, 2개의 컴퓨터(104 및 106) 간의 유일한 관계를 설정할 수 있다. 2개의 컴퓨터 간의 관계는, 예를 들어: (i) 2개의 컴퓨터에 사용가능한 정책; (ii) 패드(예를 들어, OTP)의 집합 또는 헤더, 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 암호화하는데 사용되는 다른 암호 체계; (iii) 관계의 상태(예를 들어, 다음의 데이터 패킷 내의 헤더가 위치하여야 하는 곳, 어느 패드가 현재 사용되어야 하는지, 또는 기타 상태 정보); 또는 (iv) 관계의 다른 특성을 정의할 수 있다. 송신 컴퓨터(104)가 다른 컴퓨터와 다른 관계를 가질 수 있지만, 다른 컴퓨터는 다른 정책, 암호 체계, 관계 상태, 또는 다른 관계 특성을 정의하는 다른 관계들의 결과로서 송신 컴퓨터(104)로부터 수신 컴퓨터(106)로 향하는 데이터 패킷을 처리할 수 없다. 한편, 수신 컴퓨터(106)는 각기 다른 송신 컴포넌트(108)와 관련된 정보의 부분들을 얻고 이 얻어진 부분들을 조합하여 이후에 적절한 수신 컴포넌트(110)에 제공되는 정보를 형성하기 위해 송신 컴퓨터(104)로부터의 데이터 패킷을 처리할 수 있다.

[0077] 어떤 구현에서, 도 5를 참조하면, 송신 컴퓨터(104)는 각각 송신 컴포넌트(108a, 108b, 및 108c)로부터 정보(502, 504, 및 506)를 수신할 수 있다. 어떤 구현에서, 송신 컴포넌트(108a, 108b, 및 108c)는 가상 머신, 어플리케이션, 또는 송신 컴퓨터(104) 내의 다른 컴포넌트를 포함할 수 있다. 정보(502, 504, 및 506)는 송신 컴포넌트(108a, 108b, 및 108c)로부터, 각각 대응하는 수신 컴포넌트(110a, 110b, 및 110c)로 송신될 통신을 포함할 수 있다.

[0078] 송신 컴퓨터(104)는 정보(502)의 적어도 개개의 부분(502a-502n)을 나타내는 데이터 슬라이스, 정보(504)의 적어도 개개의 부분(504a-504n)을 나타내는 데이터 슬라이스, 및 정보(506)의 적어도 개개의 부분(506a-506n)을 나타내는 데이터 슬라이스를 발생할 수 있다. 데이터 패킷(508a-508n)은 (i) 데이터 패킷(508a)이 헤더(510a), 부분(502a, 504a, 및 506a), 또는 다른 데이터 패킷 요소를 포함하고; (ii) 데이터 패킷(508b)이 헤더(510b), 부분(502b, 504b, 및 506b), 또는 다른 데이터 패킷 요소를 포함하고; (iii) 등등이도록 발생될 수 있다. 헤더(510a)는 부분(502a, 504a, 및 506a)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계를 지정할 수 있다. 헤더(510b)는 부분(502b, 504b, 및 506b)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계를 지정할 수 있다. 한 예로서, 헤더(510)는 헤더(510)를 해석하는데 사용되어야 하는 정책을 지정하고 암호 체계에 대응하는 정책의 암호 체계 인덱스를 지정함으로써 암호 체계를 지정할 수 있다. 정책과 암호 체계 인덱스에 관한 더 이상의 상세가 이후 제공된다.

[0079] 이제 도 5를 참조하면, 데이터 패킷(508a-508n)은 수신 컴퓨터(106)로 송신될 수 있다. 수신 컴퓨터(106)는 정보(502, 504, 및 506)를 형성하기 위해 데이터 패킷(508a-508n)을 처리하고, 정보(502, 504, 및 506)를 그들 각각의 수신 컴포넌트(110a, 110b, 및 110c)에 제공할 수 있다. 한 예로서, 수신 컴퓨터(106)는 부분(502a-502n)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계, 부분(504a-504n)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계, 및 부분(506a-506n)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계를 식별하기 위해 헤더(510a-510n)를 처리할 수 있다. 데이터 슬라이스는 부분(502a-502n, 504a-504n, 및 506a-506n)을 얻기 위해 식별된 암호 체계를 사용하여 해독될 수 있다. 대응하는 부분은 다음에 정보(502, 504, 및 506)를 형성하도록 조합될 수 있다.

[0080] 어떤 구현에서, 송신 컴퓨터(104)는 랜덤하게 발생된 필러 데이터를 포함하는 제3 데이터 슬라이스를 발생하도록 프로그램될 수 있다. 제3 데이터 슬라이스는 제1 데이터 슬라이스 또는 제2 데이터 슬라이스의 크기와 다른 크기를 가질 수 있다. 제1 데이터 패킷은 제1 데이터 패킷이 제3 데이터 슬라이스를 포함할 수 있도록 발생될 수 있다. 이 방식으로, 다른 이점들 중에서, 랜덤하게 발생된 필러 데이터는 정보의 부분을 실제로 나타내는 데이터를 위한 위장의 다른 층을 제공하기 위해 데이터 패킷에서 인터리브될 수 있다. 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스는 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스가 정보의 부분을 나타내지 않는다는 사실을 의도하지 않은 수신인에게 더 감추기 위해서 암호 체계를 사용하여 암호화될 수 있다. 부가적으로, 또는 대안적으로, 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스의 크기는 정보의 부분을 나타내는 데이터 슬라이스의 시작과 끝을 숨기기 위해 데이터 패킷 내에서 변화될 수 있다.

[0081] 제한되지 않는 예에서, 도 5를 참조하면, 데이터 패킷(508a-508n)의 각각은 랜덤하게 발생된 필러 데이터(512)

를 포함하는 데이터 슬라이스를 포함할 수 있다. 데이터 패킷(508a-508n)이 수신 컴퓨터(106)에 의해 수신될 때, (정보의 부분과 반대로) 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스가 무시될 수 있다. 예를 들어, 정보의 부분을 나타내는 데이터 슬라이스의 시작과 끝 위치가 식별될 수 있다. 정보의 부분을 나타내는 데이터 슬라이스는 다음에 추출되어, 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스를 남길 수 있다. 후속하여, 추출된 데이터 슬라이스를 해독함으로써 정보의 부분이 얻어질 수 있다.

[0082] 한 구현에서, 제1 헤더는 제1 데이터 패킷 내의 제1 데이터 슬라이스의 위치, 제1 데이터 패킷 내의 제2 데이터 슬라이스의 위치, 제1 데이터 패킷 내의 제3 데이터 슬라이스의 위치, 또는 다른 위치 정보를 지정할 수 있다. 예를 들어, 데이터 패킷 헤더는 데이터 패킷 헤더를 해석하는데 사용되어야 하는 정책을 지정함으로써 데이터 슬라이스의 시작 위치, 데이터 슬라이스의 끝 위치, 데이터 슬라이스의 크기, 어느 데이터 슬라이스가 랜덤하게 발생된 필러 데이터를 포함하는지, 또는 다른 데이터 패킷 관련 정보를 지정할 수 있다.

[0083] 어떤 구현에서, 특정한 정책을 위한 예의 정책 정의는 {"Policy": {"Index": 0, "Name";"Standard ODEP", "SliceSizes": 128, 512, 32, 32, 128, 384, 268, 24, "TrafficModel": 2, "DataTypes": {0:"rand, 1: "otp", 2:"des", 3:"aes256"}}일 수 있다. 데이터 패킷 헤더는 정책 인덱스 0를 지정함으로써 데이터 패킷을 해석하는데 예의 정책이 사용되어야 한다는 것을 지정할 수 있다. 정책 정의에 기초하여, 트래픽 모델 인덱스 2에 대응하는 트래픽 모델은 데이터 패킷 내의 데이터 슬라이스의 위치, 어느 데이터 슬라이스가 랜덤하게 발생된 필러 데이터를 포함하는지, 또는 다른 데이터 패킷 관련 정보를 결정하는데 이용될 수 있다. 논의된 바와 같이, 어떤 구현에서, 정책 또는 다른 정책 관련 정보는 특정한 송신 컴퓨터와 특정한 수신 컴퓨터 간에 설정된 유일한 관계에 의해 정의될 수 있다. 한 예로서, 데이터 패킷 헤더에 의해 지정된 정책 인덱스에 대응하는 정책은 유일한 관계를 공유하는 송신 컴퓨터와 수신 컴퓨터에만 알려질 수 있다. 다른 예로서, 특정한 송신 컴퓨터는 송신 컴퓨터로부터 수신 컴퓨터로의 정보의 안전한 송신을 위한 특정한 수신 컴퓨터와의 제1 관계(예를 들어, 송신 채널), 및 수신 컴퓨터로부터 송신 컴퓨터에서 수신된 데이터 패킷을 처리하는 제2 관계(예를 들어, 수신 채널)를 가질 수 있다. 각각의 관계는, 예를 들어, 관계 또는 다른 데이터 패킷 관련 정보에 가용한 각기 다른 정책을 정의할 수 있다.

[0084] 어떤 구현에서, 도 6을 참조하면, 데이터 패킷 헤더(600)는 제1 비트 세트 내에 데이터 패킷 헤더(600)를 포함하는 데이터 패킷을 해석하는데 사용되어야 하는 정책을 나타내는 정책 인덱스, 데이터 패킷 내의 데이터 슬라이스를 암호화하는데 사용된 암호 체계를 나타내는 암호 체계 인덱스, 다음의 데이터 패킷 내의 헤더의 위치를 표시하는 헤더 오프셋, 역 접속을 위한 포트, 또는 다른 데이터 패킷 관련 정보를 지정할 수 있다.

[0085] 어떤 구현에서, 송신 컴퓨터(104)는 제3 암호 체계에 기초하여 제1 헤더를 발생하도록 프로그램될 수 있다. 제3 암호 체계는 제1 및 제2 암호 체계 중 하나와 동일할 수 있거나, 제1 및 제2 암호 체계 둘 다와 다를 수 있다. 한 구현에서, 도 6을 참조하면, 특정한 패드가 데이터 패킷 헤더(600)를 암호화하는데 사용될 수 있다. 데이터 패킷 헤더(600)를 암호화하는데 사용된 패드는, 예를 들어, 특정한 송신 컴퓨터와 특정한 수신 컴퓨터 간의 유일한 관계의 일부로서 사전 설정된 패드의 집합 중 하나일 수 있다.

[0086] 다양한 구현에서, 송신 컴퓨터(104)는 제3 암호 체계에 기초하여 제1 핸드셰이크 요구를 발생하도록 프로그램될 수 있다. 제1 핸드셰이크 요구는 수신 컴퓨터(106)에 송신될 수 있다. 송신 컴퓨터(104)는 수신 컴퓨터(106)로부터 제1 응답을 수신할 수 있다. 제1 헤더는 헤더를 암호화하기 위한 제3 암호 체계의 사용을 확인하는 제1 응답에 기초하여 제3 암호 체계를 사용하여 암호화될 수 있다.

[0087] 어떤 구현에서, 처음에 송신 컴퓨터(104)가 수신 컴퓨터(106)에 정보를 안전하게 송신하기 위해 수신 컴퓨터(106)와의 통신 세션을 개시할 때, 송신 컴퓨터(104)는 핸드셰이크 헤더를 갖는 핸드셰이크 패킷을 송신할 수 있다. 핸드셰이크 헤더는 바디가 없고 헤더의 모든 비트가 0으로 설정된 표준 24 비트 헤더를 포함할 수 있고, 2개의 컴퓨터 간에 설정된 관계가 인덱스 0에서 정의하는 제1 패드를 사용하여 암호화될 수 있다. 핸드셰이크 헤더가 수신 컴퓨터(106)에 의해 이해되면(예를 들어, 송신 컴퓨터(104)와 수신 컴퓨터(106)의 패드가 라인업), 수신 컴퓨터(106)는 정보를 안전하게 송신하기 위해 데이터 패킷을 발생하고/하거나 데이터 패킷으로부터 정보의 부분을 얻기 위해 데이터 패킷을 처리하는데 사용되어야 하는 설정된 관계에 의해 정의된 사전 설정된 정책 중 하나를 지정하는 메시지를 응신으로서 송신할 수 있다.

[0088] 어떤 구현에서, 송신 컴퓨터(104)는 제1 데이터 패킷이 송신되기 전에 수신 컴퓨터(106)에 송신된 데이터 패킷을 수신 컴퓨터(106)가 처리하지 못한 것에 기초하여 리셋 요구를 수신할 수 있다. 제1 핸드셰이크 요구는, 예를 들어, 리셋 요구에 기초하여 발생될 수 있다. 수신 컴퓨터(106)로부터 수신된 제1 응답은 제1 헤더용 제1 데이터 패킷 내의 제1 위치를 지정할 수 있다. 제1 데이터 패킷은 제1 데이터 패킷이 제1 데이터 패킷 내의 제

1 위치에 제1 헤더를 포함할 수 있도록 발생될 수 있다. 한 예로서, 수신 컴퓨터(106)가 후속하는 헤더의 위치를 놓쳐서 헤더를 해독할 수 없거나, 혹은 송신 컴퓨터(104)로부터 송신된 데이터 패킷을 적절히 처리하지 못한 경우에, 수신 컴퓨터(106)는 후속하는 데이터 패킷 헤더를 암호화하는데 사용되는 암호 체계, 다음의 데이터 패킷 헤더의 초기 위치, 데이터 패킷을 발생/처리하는데 사용될 정책, 또는 다른 정보를 결정하기 위해 2개의 컴퓨터 간의 해드셰이킹을 트리거하도록 리셋 요구를 송신 컴퓨터(104)에 송신할 수 있다.

[0089] 소정의 구현에서, 송신 컴퓨터(104)는 제3 암호 체계(제1 헤더의 발생에 기초가 됨)와 다른 제4 암호 체계에 기초하여 제1 핸드셰이크 요구를 발생하도록 프로그램될 수 있다. 제1 핸드셰이크 요구는 수신 컴퓨터(106)에 송신될 수 있다. 송신 컴퓨터(104)는 수신 컴퓨터(106)로부터 제1 응답을 수신할 수 있다. 제1 응답은 제3 암호 체계의 사용을 지정할 수 있다. 제1 응답은, 예를 들어, 헤더 인덱스를 지정함으로써 제3 암호 체계의 사용을 지정할 수 있다. 제3 암호 체계에 대응하는 헤더 인덱스에 기초하여, 송신 컴퓨터(104)는 제3 암호 체계를 사용하여 제2 핸드셰이크 요구를 발생하도록 프로그램될 수 있다. 제2 핸드셰이크 요구는 수신 컴퓨터(106)에 송신될 수 있다. 제2 응답은 제2 컴퓨터 시스템으로부터 수신될 수 있다. 제2 응답은, 예를 들어, 헤더를 암호화하기 위한 제3 암호 체계의 사용을 확인할 수 있다. 제3 암호 체계는 제3 암호 체계의 사용을 확인하는 제2 응답에 기초하여 제1 헤더를 암호화하는데 사용될 수 있다.

[0090] 논의된 바와 같이, 어떤 구현에서, 처음에 송신 컴퓨터(104)가 수신 컴퓨터(106)에 정보를 안전하게 송신하기 위해 수신 컴퓨터(106)와의 통신 세션을 개시할 때, 송신 컴퓨터(104)는 핸드셰이크 헤더를 갖는 핸드셰이크 패킷을 송신할 수 있다. 핸드셰이크 헤더는 2개의 컴퓨터 간에 설정된 관계가 인덱스 0에서 정의하는 제1 패드를 사용하여 암호화될 수 있다. 핸드셰이크 헤더가 수신 컴퓨터(106)에 의해 이해되면(예를 들어, 패드가 라인업), 수신 컴퓨터(106)는 정보를 안전하게 송신하기 위해 데이터 패킷을 발생하고/하거나 데이더 패킷으로부터 정보의 부분을 얻기 위해 데이터 패킷을 처리하는데 사용되어야 하는 설정된 관계에 의해 정의된 사전 설정된 정책 중 하나를 지정하는 메시지를 응신으로서 송신할 수 있다.

[0091] 한편, 수신 컴퓨터(106)가 (예를 들어, 패드 오배열 및 다른 이유로 인해) 헤더를 판독할 수 없는 경우에, 수신 컴퓨터(106)는 사용할 다음의 패드 파일의 인덱스를 송신 컴퓨터(104)에 송신할 수 있다. 송신 컴퓨터(104)는 다음에 인덱스에 대응하는 패드를 사용하여 암호화된 핸드셰이크 헤더를 갖는 핸드셰이크 패킷을 송신함으로써 통신 세션을 재개시할 수 있다. 어떤 구현에서, 2개의 컴퓨터 간의 관계는 소정 수의 해드셰이킹 시도가 실패한 경우에 앞으로의 접속 시도가 되지 않도록 무효한 것으로서 마크될 수 있다.

[0092] 다양한 구현에서, 송신 컴퓨터(104)는 제1 정보의 적어도 제2 부분을 나타내는 제3 데이터 슬라이스, 제2 정보의 적어도 제2 부분을 나타내는 제4 데이터 슬라이스, 또는 다른 데이터 슬라이스를 발생하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제2 헤더, 제3 데이터 슬라이스, 제4 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함하는 제2 데이터 패킷을 발생하도록 프로그램될 수 있다. 제2 데이터 패킷은 다음에 적절한 수신 컴퓨터(예를 들어, 수신 컴퓨터(106) 또는 다른 수신 컴퓨터)에 송신될 수 있다.

[0093] 한 구현에서, 제3 데이터 슬라이스는 제1 암호 체계에 기초하여 발생될 수 있다. 제4 데이터 슬라이스는 제2 암호 체계에 기초하여 발생될 수 있다. 송신 컴퓨터(104)는 제2 헤더가 제3 데이터 슬라이스용 제1 암호 체계, 제4 데이터 슬라이스용 제2 암호 체계, 또는 다른 정보를 지정할 수 있도록 제2 헤더를 발생하도록 프로그램될 수 있다.

[0094] 어떤 구현에서, 제3 데이터 슬라이스는 제1 암호 체계와 다른 암호 체계에 기초하여 발생될 수 있다. 제4 데이터 슬라이스는 제2 암호 체계와 다른 암호 체계에 기초하여 발생될 수 있다. 송신 컴퓨터(104)는 제2 헤더가 제3 데이터 슬라이스용 제1 암호 체계와 다른 암호 체계, 제4 데이터 슬라이스용 제2 암호 체계와 다른 암호 체계, 또는 다른 정보를 지정할 수 있도록 제2 헤더를 발생하도록 프로그램될 수 있다.

[0095] 어떤 구현에서, 제1 헤더는 제1 데이터 패킷 내의 제1 위치에 위치할 수 있다. 제1 헤더는 제2 헤더용 제2 데이터 패킷 내의 제2 위치를 지정할 수 있다. 제2 데이터 패킷은 제2 데이터 패킷이 제1 헤더에 의해 지정된 제2 데이터 패킷 내의 제2 위치에 제2 헤더를 포함할 수 있도록 발생될 수 있다. 제2 헤더는 제2 데이터 패킷 내의 제3 데이터 슬라이스의 위치, 제2 데이터 패킷 내의 제4 데이터 슬라이스의 위치, 또는 다른 정보를 지정할 수 있다. 이 방식으로, 헤더 위치는 한 데이터 패킷과 다음의 데이터 패킷 간에 서로 다를 수 있기 때문에, 데이터 슬라이스 및 암호 체계가 적용되었던 대응하는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계의 식별은 송신 컴퓨터(104)와 수신 컴퓨터(106) 간의 설정된 관계 밖에 있는 어떤 컴포넌트에 대해, 불가능하지는 않더라도, 극도로 어려워질 수 있다.

[0096] 소정의 구현에서, 제1 데이터 패킷 또는 제2 데이터 패킷은 제1 세션 동안에 발생 또는 송신될 수 있다. 제2 헤더는 제3 헤더용 제3 데이터 패킷 내의 제3 위치를 지정할 수 있다. 송신 컴퓨터(104)는 제1 세션과 다른 제2 세션 동안에 제1 송신 컴포넌트와 관련되어 송신될 제3 정보를 수신하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제3 정보의 적어도 일부를 나타내는 제5 데이터 슬라이스를 발생하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제3 데이터 패킷이 제3 헤더 및 제5 데이터 슬라이스를 포함할 수 있도록 제3 데이터 패킷을 발생하도록 프로그램될 수 있다. 제3 데이터 패킷은 다음에 제2 세션 동안에 적절한 수신 컴퓨터(예를 들어, 수신 컴퓨터(106) 또는 다른 수신 컴퓨터)에 송신될 수 있다. 이와 같이, 어떤 시나리오에서, 데이터 패킷 헤더는 다음의 데이터 패킷이 후속하는 통신 세션까지 발생되지 않는 경우에도 설정된 관계를 통해 송신된 다음의(혹은 후속하는) 데이터 패킷 내의 헤더의 위치를 지정할 수 있다.

[0097] 한 구현에서, 제5 데이터 슬라이스는 제1 암호 체계에 기초하여 발생될 수 있다. 송신 컴퓨터(104)는 제3 헤더가 제5 데이더 슬라이스용 제1 암호 체계를 지정할 수 있도록 제3 헤더를 발생하도록 프로그램될 수 있다.

[0098] 어떤 구현에서, 제5 데이터 슬라이스는 제1 암호 체계와 다른 암호 체계에 기초하여 발생될 수 있다. 송신 컴퓨터(104)는 제3 헤더가 제1 암호 체계와 다른 암호 체계를 지정할 수 있도록 제3 헤더를 발생하도록 프로그램될 수 있다.

[0099] 다양한 구현에서, 제1 정책, 제2 정책, 또는 다른 정책이 송신 컴퓨터(104)에서 또는 송신 컴퓨터(104)와 관련하여 저장될 수 있다. 제1 헤더는 제1 정책을 지정함으로써 제1 데이터 슬라이스용 제1 암호 체계 및 제2 데이터 슬라이스용 제2 암호 체계를 지정할 수 있다.

[0100] 논의된 바와 같이, 어떤 구현에서, 특정한 정책을 위한 예의 정책 정의는 {"Policy": {"Index": 0, "Name";"Standard ODEP", "SliceSizes": 128, 512, 32, 32, 128, 384, 268, 24, "TrafficModel": 2, "DataTypes": {0:"rand", 1: "otp", 2:"des", 3:"aes256"}}일 수 있다. 데이터 패킷 헤더는 정책 인덱스 0를 지정함으로써 데이터 패킷을 해석하는데 예의 정책이 사용되어야 한다는 것을 지정할 수 있다. 정책 인덱스 0이 지정될 때, 암호 체계는 정책 인덱스 0에 대응하는 정책의 암호 체계 인덱스(예를 들어, OTP에 인덱스 1, DES에 인덱스 2, AES256에 인덱스 3 등)를 사용하여 데이터 슬라이스에 대해 지정될 수 있다.

[0101] 어떤 구현에서, 송신 컴퓨터(104)는 수신 컴퓨터(106)로부터 제2 데이터 패킷을 수신하도록 프로그램될 수 있다. 제2 데이터 패킷은 제2 헤더, 제3 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함할 수 있다. 제3 데이터 슬라이스는 제3 정보의 일부를 나타낼 수 있다. 제2 헤더는 제2 정책을 지정할 수 있다. 송신 컴퓨터(104)는 제2 정책을 지정하는 제2 헤더에 기초하여 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계를 식별하도록 설정될 수 있다. 송신 컴퓨터(104)는 제3 정보의 일부를 얻기 위해 제1 암호 체계에 기초하여 제3 데이터 슬라이스를 처리하도록 프로그램될 수 있다. 이와 같이, 어떤 구현에서, 다른 정책에 의해 지정된 암호 체계가 그래도 중첩할 수 있다.

[0102] 논의된 바와 같이, 어떤 구현에서, 정책 또는 다른 정책 관련 정보가 특정한 송신 컴퓨터와 특정한 수신 컴퓨터 간에 설정된 유일한 관계에 의해 정의될 수 있다. 한 예로서, 데이터 패킷 헤더에 의해 지정된 정책 인덱스에 대응하는 정책은 유일한 관계를 공유하는 송신 컴퓨터와 수신 컴퓨터에만 알려질 수 있다. 다른 예로서, 특정한 송신 컴퓨터는 송신 컴퓨터로부터 수신 컴퓨터로의 정보의 안전한 송신을 위한 특정한 수신 컴퓨터와의 제1 관계(예를 들어, 송신 채널), 및 수신 컴퓨터로부터 송신 컴퓨터에서 수신된 데이터 패킷을 처리하는 제2 관계(예를 들어, 수신 채널)를 가질 수 있다. 각각의 관계는, 예를 들어, 관계 또는 다른 데이터 패킷 관련 정보에 가용한 각기 다른 정책을 정의할 수 있다.

[0103] 한 구현에서, 제1 정책은 제1 암호 체계와의 제1 암호 체계 인덱스의 관련, 제2 암호 체계와의 제2 암호 체계 인덱스의 관련, 또는 다른 관련을 지정할 수 있다. 제2 정책은 제1 암호 체계와의 제3 암호 체계 인덱스의 관련 또는 다른 관련을 지정할 수 있다. 제1 헤더는 제1 정책, 제1 데이터 슬라이스용 제1 암호 체계, 및 제2 데이터 슬라이스용 제2 암호 체계를 지정함으로써 제1 데이터 슬라이스용 제1 암호 체계 및 제2 데이터 슬라이스용 제2 암호 체계를 지정할 수 있다. 제2 헤더는 제2 정책 및 제3 데이터 슬라이스용 제3 암호 체계를 지정함으로써 제3 데이터 슬라이스용 제1 암호 체계를 지정할 수 있다. 제1 암호 체계는 제2 정책 및 제3 데이터 슬라이스용 제3 암호 체계를 지정하는 제2 헤더에 기초하여 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 식별될 수 있다.

[0104] 소정의 구현에서, 수신 컴퓨터(106)는 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하도록 프로그램될 수 있다. 예를 들면, 수신 컴퓨터(106)는 제1 헤더,

제1 정보의 적어도 일부를 나타내는 제1 데이터 슬라이스, 제2 정보의 적어도 일부를 나타내는 제2 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함하는 제1 데이터 패킷을 수신하도록 프로그램될 수 있다. 제1 통신은 제1 송신 컴포넌트와 관련될 수 있다. 제2 통신은 제1 송신 컴포넌트와 다른 제2 송신 컴포넌트와 관련될 수 있다. 수신 컴포넌트는, 예를 들어, 수신 컴퓨터(106) 외부에 있는 컴포넌트, 수신 컴퓨터(106) 내에서 실행하고 있는 컴포넌트(예를 들어, 가상 머신, 어플리케이션, 또는 송신 컴포넌트(104) 내에서 실행하고 있는 다른 컴포넌트), 또는 다른 컴포넌트를 포함할 수 있다.

[0105] 수신 컴퓨터(106)는 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계, 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계, 또는 다른 정보를 제1 헤더에 기초하여 식별하도록 프로그램될 수 있다. 한 예로서, 제1 암호 체계를 사용하는 제1 데이터 슬라이스의 암호화는 제1 암호화 타입 또는 제1 암호화 키를 사용하여 제1 정보의 부분을 암호화하는 것을 포함할 수 있다. 제2 암호화 체계를 사용하는 제2 데이터 슬라이스의 암호화는 제1 암호화 타입 또는 제1 암호화 키와 각각 다른 제2 암호화 타입 또는 제2 암호화 키를 사용하여 제2 정보의 부분을 암호화하는 것을 포함할 수 있다. 논의된 바와 같이, 암호화 타입의 예는 OTP, DES, AES, 또는 다른 암호화 타입을 포함할 수 있다.

[0106] 수신 컴퓨터(106)는 제1 정보의 부분을 얻기 위해 제1 암호 체계에 기초하여 제1 데이터 슬라이스를 처리하고, 제2 정보의 부분을 얻기 위해 제2 암호 체계에 기초하여 제2 데이터 슬라이스를 처리하도록 프로그램될 수 있다.

[0107] 어떤 구현에서, 수신 컴퓨터(106)는 제2 헤더, 제1 정보의 적어도 제2 부분을 나타내는 제3 데이터 슬라이스, 제2 정보의 적어도 제2 부분을 나타내는 제4 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함하는 제2 데이터 패킷를 수신하도록 프로그램될 수 있다. 수신 컴퓨터(106)는 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계, 제4 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계, 또는 다른 정보를 제2 헤더에 기초하여 식별하도록 프로그램될 수 있다. 수신 컴퓨터(106)는 제1 정보의 제2 부분을 얻기 위해 제1 암호 체계에 기초하여 제3 데이터 슬라이스를 처리하고, 제2 정보의 제2 부분을 얻기 위해 제2 암호 체계에 기초하여 제4 데이터 슬라이스를 처리하도록 프로그램될 수 있다.

[0108] 수신 컴퓨터(106)는 제1 수신 컴포넌트를 위한 제1 정보를 형성하기 위해 제1 정보의 부분과 제1 정보의 제2 부분을 조합하고 제2 수신 컴포넌트를 위한 제2 정보를 형성하기 위해 제2 정보의 부분과 제2 정보의 제2 부분을 조합하도록 프로그램될 수 있다.

[0109] 예를 들어, 도 5와 관련하여, 수신 컴퓨터(106)에 의해 수신된 데이터 패킷(508a-508n)은 (i) 헤더(510a), 부분(502a, 504a, 및 506a), 또는 다른 데이터 패킷 요소를 갖는 데이터 패킷(508a); (ii) 헤더(510b), 부분(502b, 504b, 및 506b), 또는 다른 데이터 패킷 요소를 갖는 데이터 패킷(508b); 및 (iii) 등등을 포함할 수 있다. 헤더(510a)는 부분(502a, 504a, 및 506a)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계를 지정할 수 있다. 헤더(510b)는 부분(502b, 504b, 및 506b)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계를 지정할 수 있다.

[0110] 수신 컴퓨터(106)는 정보(502, 504, 및 506)를 형성하도록 데이터 패킷(508a-508n)을 처리하고, 정보(502, 504, 및 506)를 그들 각각의 수신 컴포넌트(110a, 110b, 및 110c)에 제공할 수 있다. 한 예로서, 수신 컴퓨터(106)는 부분(502a-502n)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계, 부분(504a-504n)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계, 및 부분(506a-506n)을 나타내는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계를 식별하기 위해 헤더(510a-510n)를 처리할 수 있다. 데이터 슬라이스는 부분(502a-502n, 504a-504n, 및 506a-506n)을 얻기 위해 식별된 암호 체계를 사용하여 해독될 수 있다. 대응하는 부분이 다음에 부분(502, 504, 및 506)을 형성하도록 조합될 수 있다.

[0111] 한 구현에서, 제1 헤더는 제1 데이터 패킷 내의 제1 위치에 위치할 수 있다. 제1 헤더는 제2 헤더용 제2 데이터 패킷 내의 제2 위치를 지정할 수 있다. 제2 데이터 패킷은 제1 헤더에 의해 지정된 제2 데이터 패킷 내의 제2 위치에 제2 헤더를 포함할 수 있다. 이 방식으로, 헤더 위치는 한 데이터 패킷과 다음의 데이터 패킷 간에 서로 다를 수 있기 때문에, 데이터 슬라이스 및 암호 체계가 적용되었던 대응하는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계의 식별은 송신 컴퓨터(104)와 수신 컴퓨터(106) 간의 설정된 관계 밖에 있는 어떤 컴포넌트에 대해, 불가능하지는 않더라도, 극도로 어려워질 수 있다.

[0112] 어떤 구현에서, 제1 데이터 패킷 및/또는 제2 데이터 패킷은 랜덤하게 발생된 필러 데이터를 포함할 수 있다. 논의된 바와 같이, 랜덤하게 발생된 필러 데이터는 정보의 부분을 실제로 나타내는 데이터를 위한 위장의 다른

층을 제공하기 위해 데이터 패킷에서 인터리브될 수 있다. 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스는 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스가 정보의 부분을 나타내지 않는다는 사실을 의도하지 않은 수신인에게 더 감추기 위해서 암호 체계를 사용하여 암호화될 수 있다. 부가적으로, 또는 대안적으로, 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스의 크기는 정보의 부분을 나타내는 데이터 슬라이스의 시작과 끝 위치를 숨기기 위해 데이터 패킷 내에서 변화될 수 있다.

[0113] 표시된 바와 같이, 도 5와 관련하여, 데이터 패킷(508a-508n)의 각각은 랜덤하게 발생된 필러 데이터(512)를 포함하는 데이터 슬라이스를 포함할 수 있다. 데이터 패킷(508a-508n)이 수신 컴퓨터(106)에 의해 수신될 때, (정보의 부분과 반대로) 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스가 무시될 수 있다. 예를 들어, 정보의 부분을 나타내는 데이터 슬라이스의 시작과 끝 위치가 식별될 수 있다. 정보의 부분을 나타내는 데이터 슬라이스는 다음에 추출되어, 랜덤하게 발생된 필러 데이터를 포함하는 데이터 슬라이스를 남길 수 있다. 후속하여, 추출된 데이터 슬라이스를 해독함으로써 정보의 부분이 얻어질 수 있다.

[0114] 다양한 구현에서, 수신 컴퓨터(106)는 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계, 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계, 또는 다른 헤더 정보를 식별하기 위해 제3 암호 체계를 사용하여 제1 헤더를 처리하도록 프로그램될 수 있다. 제3 암호 체계는 제1 및 제2 암호 체계 중 하나와 동일할 수 있거나, 제1 및 제2 암호 체계 둘 다와 다를 수 있다. 한 시나리오에서, 특정한 패드가 (도 6에 도시된) 데이터 패킷 헤더(600)를 암호화하는데 사용될 수 있다. 데이터 패킷 헤더(600)를 암호화하는데 사용된 패드는, 예를 들어, 특정한 송신 컴퓨터와 특정한 수신 컴퓨터 간의 유일한 관계의 일부로서 사전 설정된 패드의 집합 중 하나일 수 있다.

[0115] 도 5를 다시 참조하면, 어떤 구현에서, 제1 데이터 패킷은 송신 컴퓨터(104)로부터 수신될 수 있다. 수신 컴퓨터(106)는 송신 컴퓨터(104)로부터 제1 핸드셰이크 요구를 수신하도록 프로그램될 수 있다. 제1 핸드셰이크 요구는 제3 암호 체계에 기초하여 발생될 수 있다. 수신 컴퓨터(106)는 헤더를 발생하기 위한 제3 암호 체계의 사용을 확인하는 제1 응답을 발생하도록 프로그램될 수 있다. 제1 응답은 송신 컴퓨터(104)에 송신될 수 있다. 제1 헤더는, 예를 들어, 제1 응답에 기초하여 제3 암호 체계를 사용하여 발생될 수 있다.

[0116] 논의된 바와 같이, 어떤 구현에서, 처음에 송신 컴퓨터(104)가 수신 컴퓨터(106)에 정보를 안전하게 송신하기 위해 수신 컴퓨터(106)와의 통신 세션을 개시할 때, 송신 컴퓨터(104)는 핸드셰이크 헤더를 갖는 핸드셰이크 패킷을 송신할 수 있다. 핸드셰이크 헤더는 2개의 컴퓨터 간에 설정된 관계가 인덱스 0에서 정의하는 제1 패드를 사용하여 암호화될 수 있다. 핸드셰이크 헤더가 수신 컴퓨터(106)에 의해 이해되면(예를 들어, 패드가 라인업), 수신 컴퓨터(106)는 정보를 안전하게 송신하기 위해 데이터 패킷을 발생하고/하거나 데이터 패킷으로부터 정보의 부분을 얻기 위해 데이터 패킷을 처리하는데 사용되어야 하는 설정된 관계에 의해 정의된 사전 설정된 정책 중 하나를 지정하는 메시지를 응신으로서 송신할 수 있다.

[0117] 한 구현에서, 수신 컴퓨터(106)는 제1 데이터 패킷이 수신되기 전에 수신된 데이터 패킷을 처리하지 못한 것에 기초하여 리셋 요구를 발생하도록 프로그램될 수 있다. 리셋 요구는 송신 컴퓨터(104)에 송신될 수 있다. 제1 핸드셰이크 요구는 리셋 요구의 송신에 기초하여 송신 컴퓨터(104)로부터 수신될 수 있다. 송신 컴퓨터(104)에 송신된 제1 응답은 제1 헤더용 제1 데이터 패킷 내의 제1 위치를 지정할 수 있다. 제1 데이터 패킷은 제1 데이터 패킷이 제1 데이터 패킷 내의 제1 위치에 제1 헤더를 포함할 수 있도록 발생될 수 있다. 예를 들어, 수신 컴퓨터(106)가 후속하는 헤더의 위치를 놓쳐서 헤더를 해독할 수 없거나, 혹은 송신 컴퓨터(104)로부터 송신된 데이터 패킷을 적절히 처리하지 못한 경우에, 수신 컴퓨터(106)는 후속하는 데이터 패킷 헤더를 암호화하는데 사용되는 암호 체계, 다음의 데이터 패킷 헤더의 초기 위치, 데이터 패킷을 발생/처리하는데 사용될 정책, 또는 다른 정보를 결정하기 위해 2개의 컴퓨터 간의 해드셰이킹을 트리거하도록 리셋 요구를 송신 컴퓨터(104)에 송신할 수 있다.

[0118] 소정의 구현에서, 송신 컴퓨터(104)는 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 사용하여 정보를 안전하게 하도록 프로그램될 수 있다. 예를 들어, 송신 컴퓨터(104)는 송신될 제1 정보를 수신하도록 프로그램될 수 있다. 제1 정보는 제1 송신 컴포넌트와 관련될 수 있다.

[0119] 송신 컴퓨터(104)는 제1 정보의 적어도 제1 부분을 나타내는 제1 데이터 슬라이스, 제1 정보의 적어도 제2 부분을 나타내는 제2 데이터 슬라이스, 또는 다른 데이터 슬라이스를 발생하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제1 헤더, 제1 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함하는 제1 데이터 패킷을 발생하도록 프로그램될 수 있다. 제1 데이터 패킷은 제1 헤더가 제1 데이터 패킷 내의 제1 위치에 위치할 수 있도록 발생될 수 있다. 제1 데이터 패킷은 제1 헤더가 제2 헤더용 제2 데이터 패킷 내의 제2 위치를 지정할 수 있도록

발생될 수 있다. 송신 컴퓨터(104)는 제2 데이터 패킷이 제2 헤더, 제2 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함할 수 있도록 제2 데이터 패킷을 발생하도록 프로그램될 수 있다. 제2 데이터 패킷은 제2 헤더가 제1 헤더에 의해 지정된 제2 데이터 패킷 내의 제2 위치에 위치할 수 있도록 발생될 수 있다. 제1 데이터 패킷 및 제2 데이터 패킷은 적절한 수신 컴퓨터(예를 들어, 수신 컴퓨터(106) 또는 다른 수신 컴퓨터)에 송신될 수 있다. 이 방식으로, 헤더 위치는 한 데이터 패킷과 다음의 데이터 패킷 간에 서로 다를 수 있기 때문에, 데이터 슬라이스 및 암호 체계가 적용되었던 대응하는 데이터 슬라이스를 암호화하는데 사용되는 암호 체계의 식별은 송신 컴퓨터(104)와 수신 컴퓨터(106) 간의 설정된 관계 밖에 있는 어떤 컴포넌트에 대해, 불가능하지는 않더라도, 극도로 어려워질 수 있다.

[0120] 어떤 구현에서, 송신 컴퓨터(104)는 제1 암호 체계에 기초하여 제1 헤더 및 제1 암호 체계에 기초하여 제2 헤더를 발생하도록 프로그램될 수 있다. 제1 데이터 슬라이스와 제2 데이터 슬라이스는 각각 제1 암호 체계와 다른 제2 암호 체계에 기초하여 발생될 수 있다.

[0121] 한 구현에서, 제1 정보는 제1 세션 동안에 수신될 수 있다. 제2 헤더는 제3 헤더용 제3 데이터 패킷 내의 제3 위치를 지정할 수 있다. 송신 컴퓨터(104)는 제1 세션과 다른 제2 세션 동안에 제2 정보를 수신하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제2 정보의 적어도 일부를 나타내는 제3 데이터 슬라이스를 발생하도록 프로그램될 수 있다. 송신 컴퓨터(104)는 제3 데이터 패킷이 제3 헤더, 제3 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함할 수 있도록 제3 데이터 패킷을 발생하도록 프로그램될 수 있다. 제3 데이터 패킷은 제3 헤더가 제2 헤더에 의해 지정된 제3 데이터 패킷 내의 제3 위치에 위치할 수 있도록 발생될 수 있다. 제3 데이터 패킷은 적절한 수신 컴퓨터(예를 들어, 수신 컴퓨터(106) 또는 다른 수신 컴퓨터)에 송신될 수 있다. 이와 같이, 어떤 시나리오에서, 데이터 패킷 헤더는 다음의 데이터 패킷이 후속하는 통신 세션까지 발생되는 않는 경우에도 설정된 관계를 통해 송신된 다음의(혹은 후속하는) 데이터 패킷 내의 헤더의 위치를 지정할 수 있다.

[0122] 소정의 구현에서, 수신 컴퓨터(106)는 다른 데이터 패킷에 대해 데이터 패킷 내의 각기 다른 위치에서 시작하는 헤더를 갖는 데이터 패킷을 처리하도록 프로그램될 수 있다. 예를 들어, 수신 컴퓨터(106)는 제1 데이터 패킷, 제2 데이터 패킷, 또는 다른 데이터 패킷을 수신하도록 프로그램될 수 있다. 제1 데이터 패킷은 제1 헤더, 제1 정보의 제1 부분을 나타내는 제1 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함할 수 있다. 제2 데이터 패킷은 제2 헤더, 제1 정보의 제2 부분을 나타내는 제2 데이터 슬라이스, 또는 다른 데이터 패킷 요소를 포함할 수 있다.

[0123] 수신 컴퓨터(106)는 제1 데이터 패킷 내의 제1 위치의 제1 헤더를 식별하도록 프로그램될 수 있다. 수신 컴퓨터(106)는 제2 헤더를 위한 제2 위치를 지정하는 제1 헤더에 기초하여 제2 데이터 패킷 내의 제2 위치의 제2 헤더를 식별하도록 프로그램될 수 있다. 수신 컴퓨터(106)는 제1 헤더에 기초하여 제1 데이터 패킷 내의 제1 데이터 슬라이스, 및 제2 헤더에 기초하여 제2 데이터 패킷 내의 제2 데이터 슬라이스를 식별하도록 프로그램될 수 있다.

[0124] 수신 컴퓨터(106)는 제1 정보의 제1 부분을 얻기 위해 제1 데이터 슬라이스를 처리하고, 제1 정보의 제2 부분을 얻기 위해 제2 데이터 슬라이스를 처리하도록 프로그램될 수 있다. 수신 컴퓨터(106)는 제1 정보를 형성하기 위해 제1 정보의 제1 부분과 제1 정보의 제2 부분을 조합하도록 프로그램될 수 있다.

[0125] 어떤 구현에서, 수신 컴퓨터(106)는 제1 헤더를 암호화하는데 사용되는 제1 암호 체계를 식별하도록 프로그램될 수 있다. 수신 컴퓨터(106)는 제1 데이터 슬라이스를 암호화하는데 사용되는 제2 암호 체계를 식별하기 위해 제1 암호 체계를 사용하여 제1 헤더를 처리하도록 프로그램될 수 있다. 제2 암호 체계는 제1 암호 체계와 다를 수 있다. 제1 데이터 슬라이스는 제1 정보의 제1 부분을 얻기 위해 제2 암호 체계를 사용하여 처리될 수 있다. 제2 데이터 슬라이스는 제1 정보의 제2 부분을 얻기 위해 제2 암호 체계를 사용하여 처리될 수 있다.

[0126] 도 7은 본 발명의 양태에 따른, 런타임 환경에서 실행하고 있는 소프트웨어 컴포넌트를 대응하는 양호한 것으로 알려진 소프트웨어 컴포넌트로 교체하는 방법의 플로우차트의 예시도이다. 아래에 제시된 방법(700)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(700)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(700)의 동작이 도 7에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0127] 어떤 구현에서, 방법(700)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기

억된 명령에 응답하여 방법(700)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(700)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0128] 동작(702)에서, 런타임 환경에서 실행하고 있는 적어도 제1 소프트웨어 컴포넌트가 교체되어야 한다는 것을 나타내는 적어도 제1 이벤트가 결정될 수 있다. 제1 이벤트는, 예를 들어, 제1 소프트웨어 컴포넌트가 손상되었거나 잠재적으로 손상되었는지에 관계없이 결정될 수 있다. 어떤 구현에서, 제1 이벤트는 하나 이상의 교체 시간 간격의 경과, 제1 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과하는 것, 제1 소프트웨어 컴포넌트의 갱신된 버전의 사용가능성, 및/또는 다른 이벤트를 포함할 수 있다. 동작(702)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0129] 동작(704)에서, 제1 소프트웨어 컴포넌트에 대응하는 적어도 제2 소프트웨어 컴포넌트가 런타임 환경과 분리된 컴포넌트 저장소로부터 얻어질 수 있다. 동작(704)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0130] 동작(706)에서, 제2 소프트웨어 컴포넌트와 관련된 무결성 정보가 얻어질 수 있다. 동작(706)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0131] 동작(708)에서, 제2 소프트웨어 컴포넌트는 무결성 정보에 기초하여 유효화될 수 있다. 동작(708)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0132] 동작(710)에서, 제2 소프트웨어 컴포넌트는 런타임 환경에서 인스턴스화될 수 있다. 제2 소프트웨어 컴포넌트는 제2 소프트웨어 컴포넌트가 런타임 환경에서 사용가능하지 않도록 비활성 경로에서 인스턴스화될 수 있다. 어떤 구현에서, 제2 소프트웨어 컴포넌트는 제2 소프트웨어 컴포넌트가 무결성 정보를 통해 유효화되는 것에 기초하여 인스턴스화될 수 있다. 동작(710)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0133] 동작(712)에서, 제1 소프트웨어 컴포넌트는 (적어도 제1 소프트웨어 컴포넌트가 교체되어야 하는 것을 나타내는) 제1 이벤트에 기초하여 제2 소프트웨어 컴포넌트로 교체될 수 있다. 동작(712)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0134] 소정의 구현에서, 동작(712)과 관련하여, 제1 이벤트는 하나 이상의 교체 시간 간격의 경과, 제1 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과하는 것, 제1 소프트웨어 컴포넌트의 갱신된 버전의 사용가능성, 및/또는 다른 이벤트를 포함할 수 있다. 한 예로서, 제1 소프트웨어 컴포넌트는 하나 이상의 교체 시간 간격의 경과 이후에 교체될 수 있다. 제2 소프트웨어 컴포넌트는 하나 이상의 교체 시간 간격의 제2 경과 이후에 교체될 수 있다. 다른 예로서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트의 사용 횟수가 임계값에 도달하거나 초과하는 것에 기초하여 교체될 수 있다. 다른 예로서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트의 갱신된 버전이 사용가능함에 기초하여 교체될 수 있다. 제2 소프트웨어 컴포넌트는 갱신된 버전을 포함할 수 있다.

[0135] 다양한 구현에서, 동작(712)과 관련하여, 제1 소프트웨어 컴포넌트의 제2 소프트웨어 컴포넌트로의 교체는 제2 소프트웨어 컴포넌트가 런타임 환경에서 사용가능하도록 활성 경로에 제2 소프트웨어 컴포넌트를 놓는 것을 포함할 수 있다. 한 구현에서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 각각 가상 머신을 포함할 수 있고, 활성 경로에 제2 소프트웨어 컴포넌트를 놓는 것은 제2 소프트웨어 컴포넌트가 런타임 환경에서 사용가능하도록 제2 소프트웨어 컴포넌트와 관련된 네트워크 설정을 갱신하는 것을 포함할 수 있다. 다른 구현에서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 각각 운영 체제 처리를 포함할 수 있고, 활성 경로에 제2 소프트웨어 컴포넌트를 놓는 것은 제2 소프트웨어 컴포넌트를 처리 간 통신 설정을 통해 활성 경로 내로 이동시키는 것을 포함할 수 있다. 다른 구현에서, 제1 소프트웨어 컴포넌트 및 제2 소프트웨어 컴포넌트는 각각 실행가능한 코드를 포함할 수 있고, 활성 경로에 제2 소프트웨어 컴포넌트를 놓는 것은 제2 소프트웨어 컴포넌트와 관련된 메모리 어드레스 포인터 또는 룩업 테이블을 갱신하는 것을 포함할 수 있다.

[0136] 어떤 구현에서, 동작(712)과 관련하여, 제1 소프트웨어 컴포넌트의 제2 소프트웨어 컴포넌트로의 교체는 제1 소

프트웨어 컴포넌트를 활성 경로 밖으로 이동시키는 것을 더 포함할 수 있다. 한 구현에서, 제1 소프트웨어 컴포넌트는 제2 소프트웨어 컴포넌트가 활성 경로에 놓인 후에 활성 경로 밖으로 이동될 수 있다.

[0137] 동작(714)에서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트가 교체된 이후에 분석될 수 있다. 동작(714)은 하나 이상의 구현에 따라, 컴포넌트 분석 서브시스템(116)과 동일하거나 유사한 컴포넌트 분석 서브시스템에 의해 수행될 수 있다.

[0138] 동작(716)에서, 제1 소프트웨어 컴포넌트가 손상되었는지 및/또는 어떻게 손상되었는지의 결정이 이루어질 수 있다. 예를 들어, 제1 소프트웨어 컴포넌트가 손상되었는지 및/또는 어떻게 손상되었는지의 결정은 제1 소프트웨어 컴포넌트의 분석에 기초하여 이루어질 수 있다. 동작(716)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0139] 도 8은 본 발명의 양태에 따른, 소프트웨어 컴포넌트에서 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트를 제한하는 방법의 플로우차트의 예시도이다. 아래에 제시된 방법(800)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(800)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(800)의 동작이 도 8에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0140] 어떤 구현에서, 방법(800)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(800)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(800)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0141] 동작(802)에서, 컴포넌트 생성 환경에서의 적어도 제1 소프트웨어 컴포넌트가 식별될 수 있다. 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 기능을 제공하는 제1 서브 컴포넌트를 포함할 수 있다. 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 제2 서브 컴포넌트를 더 포함할 수 있다. 동작(802)은 하나 이상의 구현에 따라, 컴포넌트 실링 서브시스템(304)과 동일하거나 유사한 컴포넌트 실링 서브시스템에 의해 수행될 수 있다.

[0142] 동작(804)에서, 제1 소프트웨어 컴포넌트의 제1 서브 컴포넌트가 식별될 수 있다. 어떤 구현에서, 제1 서브 컴포넌트에 의해 제공된 기능은 제1 서브 컴포넌트를 유지하는데 사용되는 유지 기능 또는 관리 기능을 포함할 수 있다. 제1 서브 컴포넌트는 제1 서브 컴포넌트에 의해 제공된 유지 기능 또는 관리 기능에 기초하여 식별될 수 있다. 동작(804)은 하나 이상의 구현에 따라, 컴포넌트 실링 서브시스템(304)과 동일하거나 유사한 컴포넌트 실링 서브시스템에 의해 수행될 수 있다.

[0143] 동작(806)에서, 제1 서브 컴포넌트가 무력화될 수 있다. 예를 들어, 제1 서브 컴포넌트는 제1 서브 컴포넌트에 의해 제공된 기능이 제1 서브 컴포넌트가 실행될 때 사용가능하지 않도록 무력화될 수 있다. 소정의 구현에서, 제1 서브 컴포넌트를 무력화하는 것은 제1 소프트웨어 컴포넌트로부터 제1 서브 컴포넌트를 제거하는 것을 포함할 수 있다. 어떤 구현에서, 제1 서브 컴포넌트를 무력화하는 것은 제1 소프트웨어 컴포넌트로부터 제1 서브 컴포넌트를 제거하지 않고서 제1 서브 컴포넌트에 의해 제공된 기능을 턴 오프하는 것을 포함할 수 있다. 다양한 구현에서, 제1 서브 컴포넌트를 무력화하는 것은 제1 소프트웨어 컴포넌트를 무력화 명령과 관련시키는 것을 포함할 수 있다. 한 예로서, 무력화 명령은 제1 서브 컴포넌트가 무력화 명령에 기초하여 제1 소프트웨어 컴포넌트로부터 제거되도록 제1 소프트웨어 컴포넌트와 관련될 수 있다. 다른 예로서, 무력화 명령은 제1 서브 컴포넌트에 의해 제공된 기능이 무력화 명령에 기초하여 턴 오프되도록 제1 소프트웨어 컴포넌트와 관련될 수 있다. 동작(806)은 하나 이상의 구현에 따라, 컴포넌트 실링 서브시스템(304)과 동일하거나 유사한 컴포넌트 실링 서브시스템에 의해 수행될 수 있다.

[0144] 소정의 구현에서, 제1 소프트웨어 컴포넌트는 제1 소프트웨어 컴포넌트가 제1 서브 컴포넌트, 제2 서브 컴포넌트, 또는 다른 서브 컴포넌트를 포함하도록 컴포넌트 생성 환경에서 발생될 수 있다. 제1 프로필 정보는 제1 소프트웨어 컴포넌트와 관련될 수 있다. 제1 프로필 정보는, 예를 들어, 특정한 소프트웨어 컴포넌트의 특정한 서브 컴포넌트가 무력화되어야 하는지를 결정하는데 사용될 수 있다. 제1 서브 컴포넌트는 제1 프로필 정보에 기초하여 무력화될 수 있다. 제2 서브 컴포넌트는 제1 프로필 정보에 기초하여 무력화될 수 없다. 제1 소프트

웨어 컴포넌트의 발생 및 제1 소프트웨어 컴포넌트와의 제1 프로필 정보의 관련은 하나 이상의 구현에 따라, 컴포넌트 작성 서브시스템(302)과 동일하거나 유사한 컴포넌트 작성 서브시스템에 의해 수행될 수 있다.

[0145] 동작(808)에서, 제1 소프트웨어 컴포넌트의 제2 서브 컴포넌트가 식별될 수 있다. 제2 서브 컴포넌트는 공격 가능하거나 잠재적으로 공격 가능한 서브 컴포넌트로서 식별될 수 있다. 어떤 구현에서, 제1 서브 컴포넌트는 제1 서브 컴포넌트가 제1 소프트웨어 컴포넌트의 기능을 위해 필요하지 않다는 결정에 기초하여 무력화될 수 있다. 제2 서브 컴포넌트는 제2 서브 컴포넌트가 제1 소프트웨어 컴포넌트의 기능을 위해 필요하다는 결정에 기초하여 무력화될 수 없다. 동작(808)은 하나 이상의 구현에 따라, 컴포넌트 실링 서브시스템(304)과 동일하거나 유사한 컴포넌트 실링 서브시스템에 의해 수행될 수 있다.

[0146] 동작(810)에서, 제2 서브 컴포넌트가 공격당하거나 잠재적으로 공격당하게 하는 제2 서브 컴포넌트와 관련된 코드 세트가 식별될 수 있다. 동작(810)은 하나 이상의 구현에 따라, 컴포넌트 실링 서브시스템(304)과 동일하거나 유사한 컴포넌트 실링 서브시스템에 의해 수행될 수 있다.

[0147] 동작(812)에서, 코드 세트가 제2 서브 컴포넌트로부터 제거될 수 있다. 예를 들어, 코드 세트는 제2 서브 컴포넌트가 코드 세트를 더 이상 포함하지 않도록 제2 서브 컴포넌트로부터 제거될 수 있다. 동작(812)은 하나 이상의 구현에 따라, 컴포넌트 실링 서브시스템(304)과 동일하거나 유사한 컴포넌트 실링 서브시스템에 의해 수행될 수 있다.

[0148] 동작(814)에서, 무결성 정보가 제1 소프트웨어 컴포넌트에 대해 발생될 수 있다. 예를 들어, 무결성 정보는 제1 서브 컴포넌트가 무력화된 후에, 코드 세트가 제2 서브 컴포넌트로부터 제거된 후에, 그리고 제1 소프트웨어 컴포넌트가 컴포넌트 생성 환경과 분리된 컴포넌트 저장소에 놓이기 전에 제1 소프트웨어 컴포넌트에 대해 발생될 수 있다. 무결성 정보는, 예를 들어, 무결성 정보가 제1 서브 컴포넌트에 대해 발생된 후에 제1 소프트웨어 컴포넌트가 수정되었는지를 결정하는데 사용될 수 있다. 동작(814)은 하나 이상의 구현에 따라, 컴포넌트 엑스포팅 서브시스템(306)과 동일하거나 유사한 컴포넌트 엑스포팅 서브시스템에 의해 수행될 수 있다.

[0149] 동작(816)에서, 제1 소프트웨어 컴포넌트 및 무결성 정보는 컴포넌트 저장소에 놓일 수 있다. 동작(816)은 하나 이상의 구현에 따라, 컴포넌트 엑스포팅 서브시스템(306)과 동일하거나 유사한 컴포넌트 엑스포팅 서브시스템에 의해 수행될 수 있다.

[0150] 도 9는 본 발명의 양태에 따른, 송신될 각기 다른 정보를 데이터 패킷의 슬라이스로 조합하고 정보의 안전한 송신을 위해 각기 다른 암호 체계를 사용하여 슬라이스를 암호화하는 방법을 도시한다. 아래에 제시된 방법(900)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(900)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(900)의 동작이 도 9에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0151] 어떤 구현에서, 방법(900)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(900)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(900)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0152] 동작(902)에서, 송신될 제1 정보 및 송신될 제2 정보가 수신될 수 있다. 제1 정보는 제1 송신 컴포넌트와 관련될 수 있다. 제2 정보는 제2 송신 컴포넌트와 관련될 수 있다. 동작(902)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0153] 동작(904)에서, 제1 데이터 슬라이스가 제1 암호 체계에 기초하여 발생될 수 있다. 제1 데이터 슬라이스는 제1 데이터 슬라이스가 제1 정보의 적어도 제1 부분을 나타내도록 발생될 수 있다. 제1 정보의 제1 부분은, 예를 들어, 제1 데이터 슬라이스를 발생하기 위해 제1 암호 체계를 사용하여 암호화될 수 있다. 동작(904)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0154] 동작(906)에서, 제2 데이터 슬라이스가 제2 암호 체계에 기초하여 발생될 수 있다. 제2 데이터 슬라이스는 제2 데이터 슬라이스가 제2 정보의 적어도 제1 부분을 나타내도록 발생될 수 있다. 제2 정보의 제1 부분은, 예를 들어, 제2 데이터 슬라이스를 발생하기 위해 제2 암호 체계를 사용하여 암호화될 수 있다. 동작(906)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0155] 동작(908)에서, 제1 헤더가 제3 암호 체계에 기초하여 발생될 수 있다. 제1 헤더는 제1 헤더가 제1 데이터 슬라이스용 제1 암호 체계 및 제2 데이터 슬라이스용 제2 암호 체계를 지정하도록 발생될 수 있다. 제1 헤더는, 예를 들어, 제3 암호 체계를 사용하여 암호화될 수 있다. 동작(908)은 하나 이상의 구현에 따라, 정보 송신 시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0156] 어떤 구현에서, 동작(908)과 관련하여, 제1 헤더는 제1 헤더가 데이터 패킷 내의 제1 데이터 슬라이스의 위치, 데이터 패킷 내의 제2 데이터 슬라이스의 위치, 또는 데이터 패킷 내의 다른 데이터 슬라이스의 위치를 지정하도록 발생될 수 있다.

[0157] 소정의 구현에서, 동작(908)과 관련하여, 제1 핸드셰이크 요구가 제3 암호 체계에 기초하여 (제1 헤더가 발생되는) 제1 컴퓨터 시스템에서 발생될 수 있다. 제1 핸드셰이크 요구는 제2 컴퓨터 시스템에 송신될 수 있다. 제1 응답이 제2 컴퓨터 시스템으로부터 수신될 수 있다. 제1 응답은, 예를 들어, 헤더를 암호화하기 위한 제3 암호 체계의 사용을 확인할 수 있다. 제3 암호 체계는 제3 암호 체계의 사용을 확인하는 제1 응답에 기초하여 제1 헤더를 암호화하는데 사용될 수 있다. 어떤 구현에서, (예를 들어, 제1 헤더의 발생 전에) 제2 컴퓨터 시스템에 송신된 데이터 패킷을 제2 컴퓨터 시스템이 처리하지 못한 것에 기초하여 제1 컴퓨터 시스템에서 리셋 요구가 수신될 수 있다. 제1 핸드셰이크 요구는, 예를 들어, 리셋 요구에 기초하여 발생될 수 있다. 이러한 동작은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 시스템에 의해 수행될 수 있다.

[0158] 다양한 구현에서, 동작(908)과 관련하여, 제1 핸드셰이크 요구가 제4 암호 체계에 기초하여 (제1 헤더가 발생되는) 제1 컴퓨터 시스템에서 발생될 수 있다. 제1 핸드셰이크 요구는 제2 컴퓨터 시스템에 송신될 수 있다. 제1 응답이 제2 컴퓨터 시스템으로부터 수신될 수 있다. 제1 응답은 제3 암호 체계의 사용을 지정할 수 있다. 제1 응답은, 예를 들어, 헤더 인덱스를 지정함으로써 제3 암호 체계의 사용을 지정할 수 있다. 제2 핸드셰이크 요구가 제3 암호 체계에 대응하는 헤더 인덱스에 기초하여 제3 암호 체계를 사용하여 발생될 수 있다. 제2 핸드셰이크 요구는 제2 컴퓨터 시스템에 송신될 수 있다. 제2 응답이 제2 컴퓨터 시스템으로부터 수신될 수 있다. 제2 응답은, 예를 들어, 헤더를 암호화하기 위한 제3 암호 체계의 사용을 확인할 수 있다. 제3 암호 체계는 제3 암호 체계의 사용을 확인하는 제2 응답에 기초하여 제1 헤더를 암호화하는데 사용될 수 있다. 이러한 동작은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 시스템에 의해 수행될 수 있다.

[0159] 동작(910)에서, 제1 데이터 패킷이 발생될 수 있다. 제1 데이터 패킷은 제1 데이터 패킷이 제1 헤더, 제1 데이터 슬라이스, 제2 데이터 슬라이스, 랜덤하게 발생된 필러 데이터, 또는 다른 데이터를 포함하도록 발생될 수 있다. 어떤 구현에서, 제1 데이터 패킷은 제1 데이터 패킷이 제1 데이터 슬라이스, 제2 데이터 슬라이스, 또는 다른 데이터 슬라이스용으로 제1 헤더에 의해 지정된 제1 데이터 패킷 내의 위치들에 1 데이터 슬라이스, 제2 데이터 슬라이스, 또는 다른 데이터 슬라이스를 포함하도록 발생될 수 있다. 동작(910)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0160] 동작(912)에서, 제3 데이터 슬라이스는 제1 암호 체계에 기초하여 발생될 수 있다. 제3 데이터 슬라이스는 제3 데이터 슬라이스가 제1 정보의 적어도 제2 부분을 나타내도록 발생될 수 있다. 제1 정보의 제2 부분은, 예를 들어, 제3 데이터 슬라이스를 발생하기 위해 제1 암호 체계를 사용하여 암호화될 수 있다. 동작(912)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0161] 동작(914)에서, 제4 데이터 슬라이스는 제2 암호 체계에 기초하여 발생될 수 있다. 제4 데이터 슬라이스는 제4 데이터 슬라이스가 제2 정보의 적어도 제2 부분을 나타내도록 발생될 수 있다. 제2 정보의 제2 부분은, 예를 들어, 제4 데이터 슬라이스를 발생하기 위해 제2 암호 체계를 사용하여 암호화될 수 있다. 동작(914)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0162] 동작(916)에서, 제2 헤더가 제3 암호 체계에 기초하여 발생될 수 있다. 제2 헤더는 제2 헤더가 제3 데이터 슬라이스용 제1 암호 체계 및 제4 데이터 슬라이스용 제2 암호 체계를 지정하도록 발생될 수 있다. 제2 헤더는, 예를 들어, 제3 암호 체계를 사용하여 암호화될 수 있다. 동작(916)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0163] 동작(918)에서, 제2 데이터 패킷이 발생될 수 있다. 제2 데이터 패킷은 제2 데이터 패킷이 제2 헤더, 제3 데이터 슬라이스, 제4 데이터 슬라이스, 랜덤하게 발생된 필러 데이터, 또는 다른 데이터를 포함하도록 발생될 수 있다. 어떤 구현에서, 제2 데이터 패킷은 제2 데이터 패킷이 제3 데이터 슬라이스, 제4 데이터 슬라이스, 또는

다른 데이터 슬라이스용으로 제2 헤더에 의해 지정된 제2 데이터 패킷 내의 위치들에 제3 데이터 슬라이스, 제4 데이터 슬라이스, 또는 다른 데이터 슬라이스를 포함하도록 발생될 수 있다. 동작(918)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0164] 소정의 구현에서, 동작(908, 910, 및 918)과 관련하여, 제1 데이터 패킷은 제1 데이터 패킷이 제1 데이터 패킷 내의 제1 위치에 제1 헤더를 포함하도록 발생될 수 있다. 제1 헤더는 제2 헤더가 위치하여야 할 제2 데이터 패킷 내의 제2 위치를 제1 헤더가 지정하도록 발생될 수 있다. 제2 데이터 패킷은 제2 데이터 패킷이 제1 헤더에 의해 지정된 제2 위치에 제2 헤더를 포함하도록 발생될 수 있다.

[0165] 동작(920)에서, 제1 데이터 패킷 및 제2 데이터 패킷이 송신될 수 있다. 제1 데이터 패킷 및 제2 데이터 패킷은, 예를 들어, (제1 데이터 패킷 및 제2 데이터 패킷이 발생되는) 제1 컴퓨터 시스템으로부터 제2 컴퓨터 시스템으로 송신될 수 있다. 동작(920)은 하나 이상의 구현에 따라, 정보 송신 서브시스템(402)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0166] 도 10은 본 발명의 양태에 따른, 각기 다른 암호 체계를 사용하여 암호화된 각기 다른 정보와 관련된 데이터 슬라이스를 갖는 데이터 패킷을 처리하는 방법을 도시한다. 아래에 제시된 방법(1000)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(1000)은 설명되지 않은 하나 이상의 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(1000)의 동작이 도 10에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0167] 어떤 구현에서, 방법(1000)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(1000)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(1000)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0168] 동작(1002)에서, 제1 데이터 패킷이 수신될 수 있다. 제1 데이터 패킷은 제1 헤더, 제1 데이터 슬라이스, 제2 데이터 슬라이스, 또는 다른 데이터를 포함할 수 있다. 제1 데이터 슬라이스는 제1 정보의 적어도 제1 부분을 나타낼 수 있다. 제2 데이터 슬라이스는 제2 정보의 적어도 제1 부분을 나타낼 수 있다. 제1 정보는 제1 송신 컴포넌트와 관련될 수 있다. 제2 정보는 제2 송신 컴포넌트와 관련될 수 있다. 제2 송신 컴포넌트는 제1 송신 컴포넌트와 다를 수 있다. 동작(1002)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0169] 동작(1004)에서, 제2 데이터 패킷이 수신될 수 있다. 제2 데이터 패킷은 제2 헤더, 제3 데이터 슬라이스, 제4 데이터 슬라이스, 또는 다른 데이터를 포함할 수 있다. 제3 데이터 슬라이스는 제1 정보의 적어도 제2 부분을 나타낼 수 있다. 제4 데이터 슬라이스는 제2 정보의 적어도 제2 부분을 나타낼 수 있다. 어떤 구현에서, 제1 헤더는 제1 데이터 패킷 내의 제1 위치에 위치할 수 있다. 제1 헤더는 제2 헤더가 위치하여야 할 제2 데이터 패킷 내의 제2 위치를 지정할 수 있다. 동작(1004)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0170] 동작(1006)에서, 제1 암호 체계가 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 식별될 수 있다. 제2 암호 체계가 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 식별될 수 있다. 제1 암호 체계의 식별 또는 제2 암호 체계의 식별은 제1 헤더에 기초할 수 있다. 동작(1006)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0171] 소정의 구현에서, 동작(1006)과 관련하여, 제1 헤더는 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계를 식별하고, 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계를 식별하고, 또는 다른 데이터를 암호화하는데 사용되는 다른 암호 체계를 식별하기 위해 제3 암호 체계를 사용하여 처리될 수 있다. 어떤 구현에서, 제1 데이터 패킷이 수신되기 전에 수신된 데이터 패킷을 처리하지 못한 것에 기초하여 (제1 헤더가 처리되는) 제1 컴퓨터 시스템에서 리셋 요구가 발생될 수 있다. 리셋 요구는 제1 데이터 패킷이 후속하여 수신되는 제2 컴퓨터 시스템에 송신될 수 있다. 리셋 요구의 송신에 기초하여 제2 컴퓨터 시스템으로부터 제1 컴퓨터 시스템에서 핸드셰이크 요구가 수신될 수 있다. 핸드셰이크 요구는 제3 암호 체계에 기초하여 발생될 수 있다. 헤더를 발생하기 위한 제3 암호 체계의 사용을 확인하는 응답이 발생될 수 있다. 응답이 제2 컴퓨터 시스템에 송신될 수 있다. 제1 헤더는, 예를 들어, 응답에 기초하여 제3 암호 체계를

사용하여 발생될 수 있다. 동작은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0172] 동작(1008)에서, 제1 데이터 슬라이스는 제1 정보의 제1 부분을 얻기 위해 제1 암호 체계에 기초하여 처리될 수 있다. 동작(1008)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0173] 동작(1010)에서, 제2 데이터 슬라이스는 제2 정보의 제1 부분을 얻기 위해 제2 암호 체계에 기초하여 처리될 수 있다. 동작(1010)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0174] 동작(1012)에서, 제1 암호 체계는 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 식별될 수 있다. 제2 암호 체계는 제4 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 식별될 수 있다. 제1 암호 체계의 식별 또는 제2 암호 체계의 식별은 제2 헤더에 기초할 수 있다. 동작(1012)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0175] 다양한 구현에서, 동작(1012)과 관련하여, 제2 헤더는 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계를 식별하고, 제4 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계를 식별하고, 또는 다른 데이터를 암호화하는데 사용되는 다른 암호화 체계를 식별하기 위해 제3 암호 체계를 사용하여 처리될 수 있다. 이러한 동작은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0176] 동작(1014)에서, 제3 데이터 슬라이스는 제1 정보의 제2 부분을 얻기 위해 제1 암호 체계에 기초하여 처리될 수 있다. 동작(1014)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0177] 동작(1016)에서, 제4 데이터 슬라이스는 제2 정보의 제2 부분을 얻기 위해 제2 암호 체계에 기초하여 처리될 수 있다. 동작(1016)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0178] 동작(1018)에서, 제1 정보의 제1 부분과 제1 정보의 제2 부분은 제1 수신 컴포넌트용 제1 정보를 형성하도록 조합될 수 있다. 동작(1018)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0179] 동작(1020)에서, 제2 정보의 제1 부분과 제2 정보의 제2 부분은 제2 수신 컴포넌트용 제2 정보를 형성하도록 조합될 수 있다. 동작(1020)은 하나 이상의 구현에 따라, 정보 수신 서브시스템(404)과 동일하거나 유사한 통신 서브시스템에 의해 수행될 수 있다.

[0180] 도 11은 본 발명의 양태에 따른, 컴포넌트의 서브 컴포넌트를 제한하고, 컴포넌트를 대응하는 양호한 것으로 알려진 컴포넌트로 교체하고, 네트워크를 통해 컴포넌트로 또는 컴포넌트로부터 송신되는 정보를 안전하게 하고 난독화함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도이다.

[0181] 아래에 제시된 방법(1100)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(1100)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(1100)의 동작이 도 11에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0182] 어떤 구현에서, 방법(1100)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(1100)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(1100)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0183] 동작(1102)에서, 소프트웨어 컴포넌트가 컴포넌트 생성 환경에서 발생될 수 있다. 동작(1102)은 하나 이상의 구현에 따라, 컴포넌트 작성 서브시스템(302)과 동일하거나 유사한 컴포넌트 작성 서브시스템에 의해 수행될 수 있다.

[0184] 동작(1104)에서, 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 소프트웨어 컴

포넌트의 컴포넌트가 무력화될 수 있다. 동작(1104)은 하나 이상의 구현에 따라, 도 8에 의해 도시된 방법(800)의 동작(802, 804, 806, 808, 810, 또는 812) 중 하나 이상을 포함할 수 있다.

[0185] 동작(1106)에서, 소프트웨어 컴포넌트는 런타임 환경과 분리된 컴포넌트 저장소에 저장될 수 있다. 런타임 환경은, 예를 들어, 대응하는 소프트웨어 컴포넌트가 실행하고 있는 생성 환경일 수 있다. 동작(1106)은 하나 이상의 구현에 따라, 도 8에 의해 도시된 방법(800)의 동작(814 및 816) 중 하나 이상을 포함할 수 있다.

[0186] 동작(1108)에서, 런타임 환경에서 실행하고 있는 대응하는 소프트웨어 컴포넌트가 제1 이벤트에 기초하여 저장된 소프트웨어 컴포넌트로 교체될 수 있다. 어떤 구현에서, 제1 이벤트는 하나 이상의 교체 시간 간격의 경과, 대응하는 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과하는 것, 대응하는 소프트웨어 컴포넌트의 갱신된 버전의 사용가능성, 및/또는 다른 이벤트를 포함할 수 있다. 동작(1108)은 하나 이상의 구현에 따라, 도 7에 의해 도시된 방법(700)의 동작(702, 704, 706, 708, 710, 712, 714, 또는 716) 중 하나 이상을 포함할 수 있다.

[0187] 동작(1110)에서, 데이터 패킷이 소프트웨어 컴포넌트에 의해 발생될 수 있다. 발생된 데이터 패킷의 각각은 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함할 수 있다. 어떤 구현에서, 소프트웨어 컴포넌트는 정보 송신 서브시스템(402) 및/또는 정보 수신 서브시스템(404)과 동일하거나 유사한 정보 송신 서브시스템 및/또는 정보 수신 서브시스템을 포함하는 소프트웨어 컴포넌트의 인스턴스를 포함할 수 있다. 동작(1110)은 하나 이상의 구현에 따라, 도 9에 의해 도시된 방법(900)의 동작(902, 904, 906, 908, 910, 912, 914, 916, 또는 918) 중 하나 이상을 포함할 수 있다.

[0188] 동작(1112)에서, 수신된 데이터 패킷이 소프트웨어 컴포넌트에 의해 처리될 수 있다. 수신된 데이터 패킷의 각각은 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함할 수 있다. 어떤 구현에서, 소프트웨어 컴포넌트는 정보 송신 서브시스템(402) 및/또는 정보 수신 서브시스템(404)과 동일하거나 유사한 정보 송신 서브시스템 및/또는 정보 수신 서브시스템을 포함하는 소프트웨어 컴포넌트의 인스턴스를 포함할 수 있다. 동작(1112)은 하나 이상의 구현에 따라, 도 10에 의해 도시된 방법(1000)의 동작(1002, 1004, 1006, 1008, 1010, 1012, 1014, 1016, 또는 1018) 중 하나 이상을 포함할 수 있다.

[0189] 도 12는 본 발명의 양태에 따른, 컴포넌트의 서브 컴포넌트를 제한하고 컴포넌트를 대응하는 양호한 것으로 알려진 컴포넌트로 교체함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도이다.

[0190] 아래에 제시된 방법(1200)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(1200)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(1200)의 동작이 도 12에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0191] 어떤 구현에서, 방법(1200)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(1200)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(1200)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0192] 동작(1202)에서, 소프트웨어 컴포넌트가 컴포넌트 생성 환경에서 발생될 수 있다. 동작(1202)은 하나 이상의 구현에 따라, 컴포넌트 작성 서브시스템(302)과 동일하거나 유사한 컴포넌트 작성 서브시스템에 의해 수행될 수 있다.

[0193] 동작(1204)에서, 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 소프트웨어 컴포넌트의 서브 컴포넌트가 무력화될 수 있다. 동작(1204)은 하나 이상의 구현에 따라, 도 8에 의해 도시된 방법(800)의 동작(802, 804, 806, 808, 810, 또는 812) 중 하나 이상을 포함할 수 있다.

[0194] 동작(1206)에서, 소프트웨어 컴포넌트는 런타임 환경과 분리된 컴포넌트 저장소에 저장될 수 있다. 런타임 환경은, 예를 들어, 대응하는 소프트웨어 컴포넌트가 실행하고 있는 생성 환경일 수 있다. 동작(1206)은 하나 이상의 구현에 따라, 도 8에 의해 도시된 방법(800)의 동작(814 및 816) 중 하나 이상을 포함할 수 있다.

[0195] 동작(1208)에서, 런타임 환경에서 실행하고 있는 대응하는 소프트웨어 컴포넌트가 제1 이벤트에 기초하여 저장된 소프트웨어 컴포넌트로 교체될 수 있다. 어떤 구현에서, 제1 이벤트는 하나 이상의 교체 시간 간격의 경과,

대응하는 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과하는 것, 대응하는 소프트웨어 컴포넌트의 갱신된 버전의 사용가능성, 및/또는 다른 이벤트를 포함할 수 있다. 동작(1208)은 하나 이상의 구현에 따라, 도 7에 의해 도시된 방법(700)의 동작(702, 704, 706, 708, 710, 712, 714, 또는 716) 중 하나 이상을 포함할 수 있다.

[0196] 도 13은 본 발명의 양태에 따른, 컴포넌트를 대응하는 양호한 것으로 알려진 컴포넌트로 교체하고 네트워크를 통해 컴포넌트로 또는 컴포넌트로부터 송신되는 정보를 안전하게 하고 난독화함으로써 컴포넌트 및 정보의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도이다.

[0197] 아래에 제시된 방법(1300)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(1300)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(1300)의 동작이 도 13에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0198] 어떤 구현에서, 방법(1300)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(1300)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(1300)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0199] 동작(1302)에서, 런타임 환경에서 실행하고 있는 대응하는 소프트웨어 컴포넌트가 런타임 환경과 분리된 컴포넌트 저장소에 저장된 소프트웨어 컴포넌트로 교체될 수 있다. 대응하는 소프트웨어 컴포넌트의 교체는 제1 이벤트에 기초할 수 있다. 어떤 구현에서, 제1 이벤트는 하나 이상의 교체 시간 간격의 경과, 대응하는 소프트웨어 컴포넌트의 사용 횟수가 사용 임계값에 도달하거나 초과하는 것, 대응하는 소프트웨어 컴포넌트의 갱신된 버전의 사용가능성, 및/또는 다른 이벤트를 포함할 수 있다. 동작(1302)은 하나 이상의 구현에 따라, 도 7에 의해 도시된 방법(700)의 동작(702, 704, 706, 708, 710, 712, 714, 또는 716) 중 하나 이상을 포함할 수 있다.

[0200] 동작(1304)에서, 데이터 패킷이 소프트웨어 컴포넌트에 의해 발생될 수 있다. 발생된 데이터 패킷의 각각은 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함할 수 있다. 어떤 구현에서, 소프트웨어 컴포넌트는 정보 송신 서브시스템(402) 및/또는 정보 수신 서브시스템(404)과 동일하거나 유사한 정보 송신 서브시스템 및/또는 정보 수신 서브시스템을 포함하는 소프트웨어 컴포넌트의 인스턴스를 포함할 수 있다. 동작(1304)은 하나 이상의 구현에 따라, 도 9에 의해 도시된 방법(900)의 동작(902, 904, 906, 908, 910, 912, 914, 916, 또는 918) 중 하나 이상을 포함할 수 있다.

[0201] 동작(1306)에서, 수신된 데이터 패킷은 소프트웨어 컴포넌트에 의해 처리될 수 있다. 수신된 데이터 패킷의 각각은 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함할 수 있다. 어떤 구현에서, 소프트웨어 컴포넌트는 정보 송신 서브시스템(402) 및/또는 정보 수신 서브시스템(404)과 동일하거나 유사한 정보 송신 서브시스템 및/또는 정보 수신 서브시스템을 포함하는 소프트웨어 컴포넌트의 인스턴스를 포함할 수 있다. 동작(1306)은 하나 이상의 구현에 따라, 도 10에 의해 도시된 방법(1000)의 동작(1002, 1004, 1006, 1008, 1010, 1012, 1014, 1016, 또는 1018) 중 하나 이상을 포함할 수 있다.

[0202] 도 14는 본 발명의 양태에 따른, 컴포넌트의 서브 컴포넌트를 제한하고 네트워크를 통해 컴포넌트로 또는 컴포넌트로부터 송신되는 정보를 안전하게 하고 난독화함으로써 컴포넌트의 보안 및 무결성을 가능하게 하는 방법의 플로우차트의 예시도이다.

[0203] 아래에 제시된 방법(1400)의 동작은 예시하고자 하는 것이다. 어떤 구현에서, 방법(1400)은 설명되지 않은 하나 이상의 추가 동작으로, 및/또는 논의된 동작 중 하나 이상의 동작 없이 달성될 수 있다. 부가적으로, 방법(1400)의 동작이 도 14에 도시되고 아래에 설명되는 순서는 제한하려는 의도는 아니다.

[0204] 어떤 구현에서, 방법(1400)은 하나 이상의 처리 장치(예를 들어, 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘)에서 구현될 수 있다. 하나 이상의 처리 장치는 전자 기억 매체에 전자적으로 기억된 명령에 응답하여 방법(1400)의 동작의 일부 또는 모두를 실행하는 하나 이상의 장치를 포함할 수 있다. 하나 이상의 처리 장치는 방법(1400)의 동작들 중 하나 이상의 실행을 위해 특별히 설계되도록 하드웨어, 펌웨어, 및/또는 소프트웨어를 통해 구성된 하나 이상의 장치를 포함할 수 있다.

[0205] 동작(1402)에서, 소프트웨어 컴포넌트가 컴포넌트 생성 환경에서 발생될 수 있다. 동작(1402)은 하나 이상의

구현에 따라, 컴포넌트 작성 서브시스템(302)과 동일하거나 유사한 컴포넌트 작성 서브시스템에 의해 수행될 수 있다.

[0206] 동작(1404)에서, 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 소프트웨어 컴포넌트의 서브 컴포넌트가 무력화될 수 있다. 동작(1404)은 하나 이상의 구현에 따라, 도 8에 의해 도시된 방법(800)의 동작(802, 804, 806, 808, 810, 또는 812) 중 하나 이상을 포함할 수 있다.

[0207] 동작(1406)에서, 소프트웨어 컴포넌트는 런타임 환경과 분리된 컴포넌트 저장소에 저장될 수 있다. 동작(1406)은 하나 이상의 구현에 따라, 도 8에 의해 도시된 방법(800)의 동작(814 및 816) 중 하나 이상을 포함할 수 있다.

[0208] 동작(1408)에서, 소프트웨어 컴포넌트는 런타임 환경에서 인스턴스화될 수 있다. 예를 들어, 소프트웨어 컴포넌트는 컴포넌트 저장소로부터 얻어진 다음에 런타임 환경에서 인스턴스화될 수 있다. 후속하여, 소프트웨어 컴포넌트는 런타임 환경에서 런타임 실행 서브시스템에 의해 실행될 수 있다. 동작(1408)은 하나 이상의 구현에 따라, 컴포넌트 관리자 서브시스템(112)과 동일하거나 유사한 컴포넌트 관리자 서브시스템에 의해 수행될 수 있다.

[0209] 동작(1410)에서, 데이터 패킷이 소프트웨어 컴포넌트에 의해 발생될 수 있다. 발생된 데이터 패킷의 각각은 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함할 수 있다. 어떤 구현에서, 소프트웨어 컴포넌트는 정보 송신 서브시스템(402) 및/또는 정보 수신 서브시스템(404)과 동일하거나 유사한 정보 송신 서브시스템 및/또는 정보 수신 서브시스템을 포함하는 소프트웨어 컴포넌트의 인스턴스를 포함할 수 있다. 동작(1410)은 하나 이상의 구현에 따라, 도 9에 의해 도시된 방법(900)의 동작(902, 904, 906, 908, 910, 912, 914, 916 또는 918) 중 하나 이상을 포함할 수 있다.

[0210] 동작(1412)에서, 수신된 데이터 패킷은 소프트웨어 컴포넌트에 의해 처리될 수 있다. 수신된 데이터 패킷의 각각은 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함할 수 있다. 어떤 구현에서, 소프트웨어 컴포넌트는 정보 송신 서브시스템(402) 및/또는 정보 수신 서브시스템(404)과 동일하거나 유사한 정보 송신 서브시스템 및/또는 정보 수신 서브시스템을 포함하는 소프트웨어 컴포넌트의 인스턴스를 포함할 수 있다. 동작(1412)은 하나 이상의 구현에 따라, 도 10에 의해 도시된 방법(1000)의 동작(1002, 1004, 1006, 1008, 1010, 1012, 1014, 1016, 또는 1018) 중 하나 이상을 포함할 수 있다.

[0211] 어떤 구현에서, 도면에 도시된 다양한 컴퓨터 및 서브시스템은 여기에 설명된 기능을 수행하도록 프로그램된 하나 이상의 컴퓨팅 장치를 포함할 수 있다. 컴퓨팅 장치는 하나 이상의 전자 기억 장치, 하나 이상의 컴퓨터 프로그램 명령으로 프로그램된 하나 이상의 물리적 프로세서, 및/또는 다른 컴포넌트를 포함할 수 있다. 컴퓨팅 장치는 네트워크 또는 다른 컴퓨팅 플랫폼과의 정보의 교환을 가능하게 하는 통신 선, 또는 포트를 포함할 수 있다. 컴퓨팅 장치는 본 명세서에서 서버에 속하는 기능을 제공하기 위해 함께 동작하는 복수의 하드웨어, 소프트웨어, 및/또는 펌웨어를 포함할 수 있다. 예를 들어, 컴퓨팅 장치는 컴퓨팅 장치로서 함께 동작하는 컴퓨터 플랫폼의 클라우드에 의해 구현될 수 있다. 컴퓨터 및 서브시스템의 각각은 특정한 보호 대책을 개별적으로 제공할 수 있거나 여기에 설명된 보호 대책의 조합을 함께 제공하기 위해 다른 컴퓨터 및 서브시스템과 조합될 수 있다.

[0212] 전자 기억 장치는 정보를 전자적으로 기억하는 비일시적 기억 매체를 포함할 수 있다. 전자 기억 장치의 전자 기억 매체는 서버와 일체로 제공되는(즉, 실질적으로 제거 불가능한) 시스템 기억 장치 또는 예를 들어, 포트(예를 들어, USB 포트, 파이어와이어 포트 등) 또는 드라이브(예를 들어, 디스크 드라이브 등)을 통해 서버와 제거 가능하게 접속될 수 있는 제거 가능 기억 장치 중 하나 또는 둘 다를 포함할 수 있다. 전자 기억 장치는 광학적으로 판독가능한 기억 매체(예를 들어, 광 디스크 등), 자기적으로 판독가능한 기억 매체(예를 들어, 자기 테이프, 자기 하드 드라이브, 플로피 드라이브 등), 전기 전하 기초 기억 매체(예를 들어, EEPROM, RAM 등), 솔리드-스테이트 기억 매체(예를 들어, 플래시 드라이브 등), 및/또는 다른 전자적으로 판독가능한 기억 매체 중 하나 이상을 포함할 수 있다. 전자 기억 장치는 하나 이상의 가상 기억 리소스(예를 들어, 클라우드 기억 장치, 가상 사설망, 및/또는 가상 기억 리소스)를 포함할 수 있다. 전자 기억 장치는 소프트웨어 알고리즘, 프로세서에 의해 결정된 정보, 서버로부터 수신된 정보, 클라이언트 컴퓨팅 플랫폼으로부터 수신된 정보, 또는 서버를 여기에 설명된 바와 같이 기능하게 하는 다른 정보를 저장할 수 있다.

[0213] 프로세서는 서버에서 정보 처리 능력을 제공하도록 프로그램될 수 있다. 이와 같이, 프로세서는 디지털 프로세서, 아날로그 프로세서, 정보를 처리하도록 설계된 디지털 회로, 정보를 처리하도록 설계된 아날로그 회로, 상

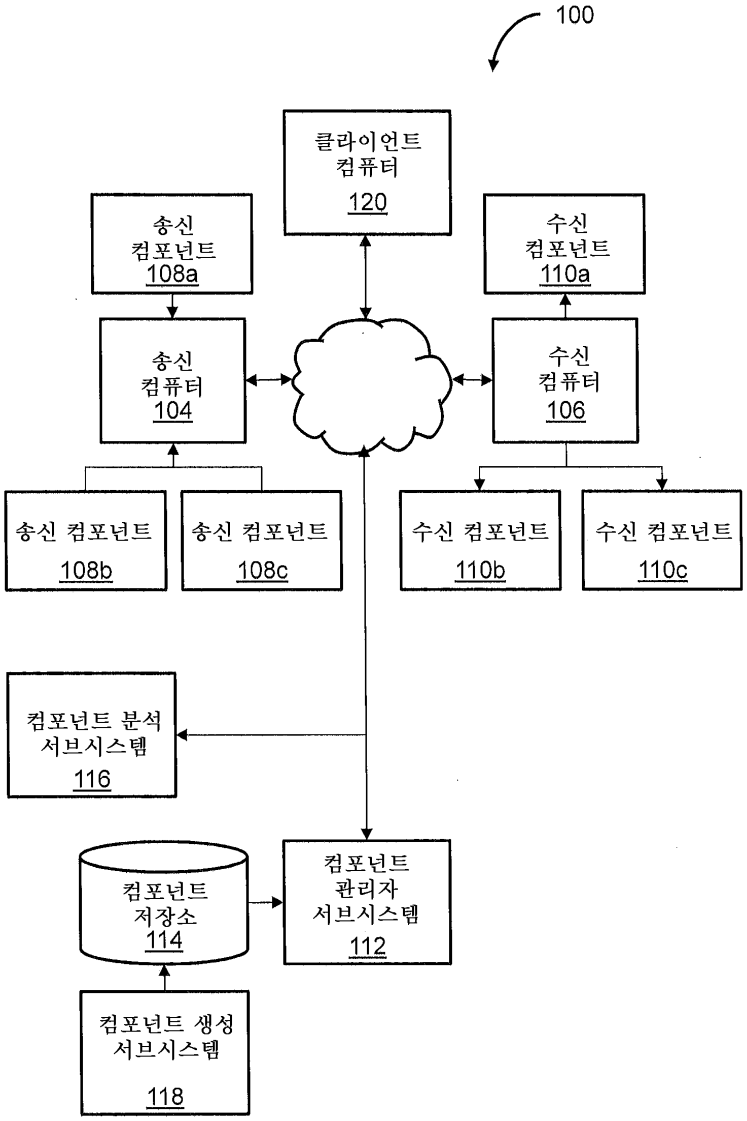
태 머신, 및/또는 정보를 전자적으로 처리하는 다른 메커니즘 중 하나 이상을 포함할 수 있다. 어떤 구현에서, 프로세서는 복수의 처리 유닛을 포함할 수 있다. 이들 처리 유닛은 동일한 장치 내에 물리적으로 위치할 수 있고, 또는 프로세서는 협동하여 동작하는 복수의 장치의 처리 기능을 나타낼 수 있다. 프로세서는 서브시스템(112, 116, 118, 302, 304, 306, 402, 404), 또는 다른 서브시스템의 여기에 설명된 기능을 수행하기 위해 컴퓨터 프로그램 명령을 실행하도록 프로그램될 수 있다. 프로세서는 소프트웨어; 하드웨어; 펌웨어; 소프트웨어, 하드웨어, 또는 펌웨어의 어떤 조합; 및/또는 프로세서 상에 처리 능력을 설정하는 다른 메커니즘에 의해 모듈을 실행하도록 프로그램될 수 있다.

[0214] 여기에 설명된 다른 서브시스템(112, 116, 118, 302, 304, 306, 402, 또는 404)에 의해 제공된 기능의 설명은 예시의 목적을 위한 것이고, 제한되지 않는다는 점을 이해하여야 하며, 그 이유는 서브시스템(112, 116, 118, 302, 304, 306, 402, 또는 404) 중 어느 것이든 설명된 것보다 많거나 적은 기능을 제공할 수 있기 때문이다. 예를 들어, 서브시스템(112, 116, 118, 302, 304, 306, 402, 또는 404) 중 하나 이상이 제거될 수 있고, 그 기능의 일부 또는 모두가 서브시스템(112, 116, 118, 302, 304, 306, 402, 또는 404) 중 다른 것들에 의해 제공될 수 있다. 다른 예로서, 부가적인 서브시스템이 본 명세서에서 서브시스템(112, 116, 118, 302, 304, 306, 402, 또는 404) 중 하나에 속하는 기능의 일부 또는 모두를 수행하도록 프로그램될 수 있다.

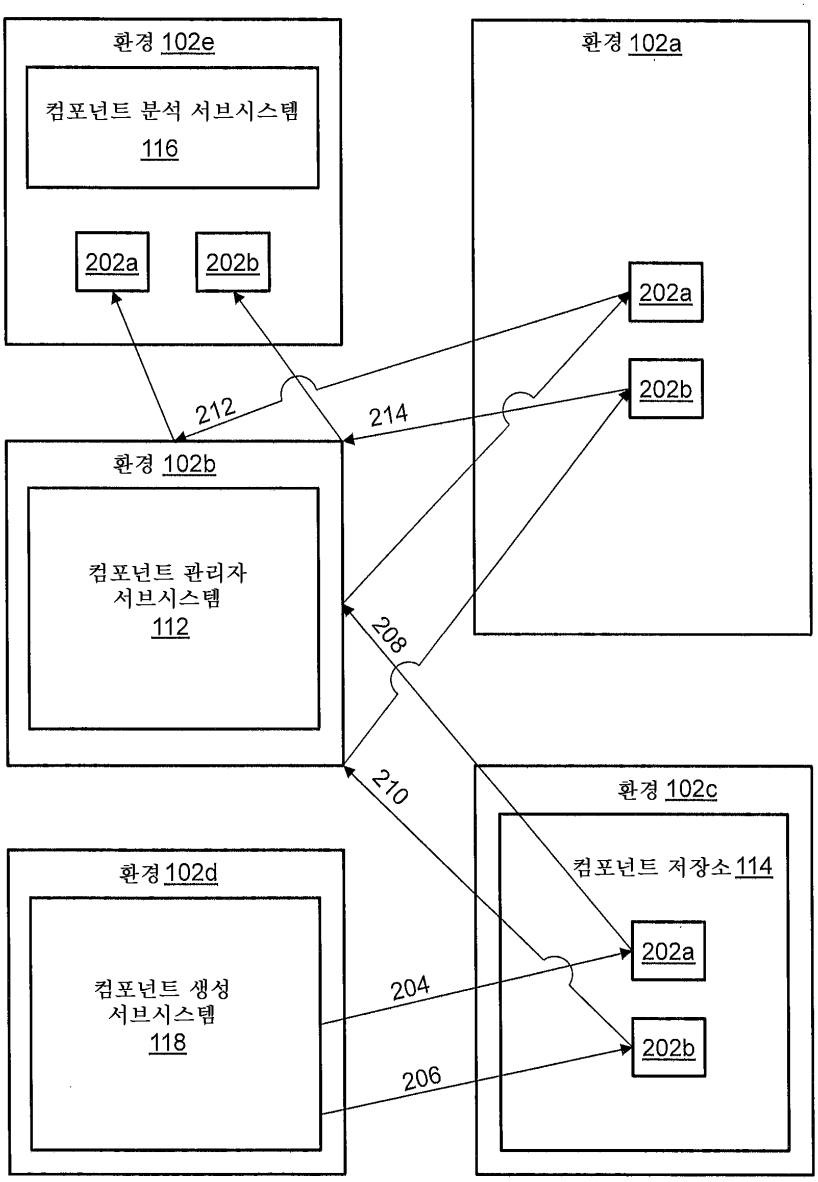
[0215] 본 발명이 가장 실제적이고 양호한 구현으로 현재 고려되는 것에 기초하여 예시의 목적을 위해 상세히 설명되었지만, 이러한 상세는 단지 그 목적을 위한 것이고, 본 발명은 개시된 구현에 제한되지 않고, 오히려 첨부된 청구 범위의 취지 및 범위 내에 있는 변형 및 균등한 구성을 포함하는 것으로 의도된다는 것을 이해하여야 한다. 예를 들어, 본 발명은 가능한 정도까지 어떤 구현의 하나 이상의 특징이 기타 구현의 하나 이상의 특징과 조합될 수 있다는 것을 숙고한다는 것을 이해하여야 한다.

도면

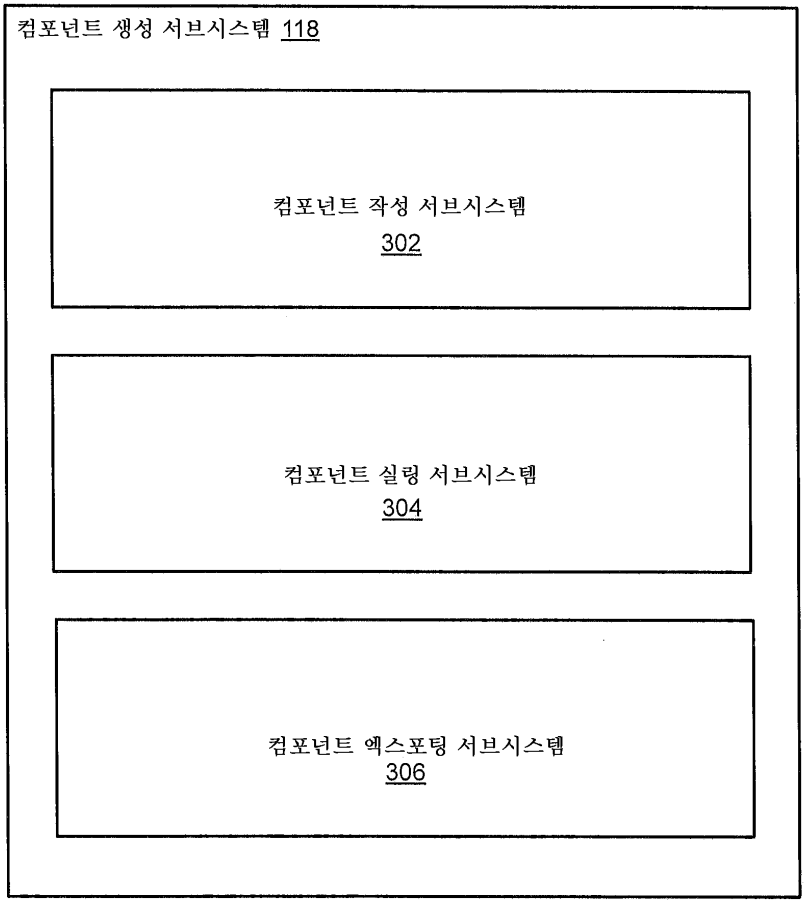
도면1


100
클라이언트 컴퓨터 120
송신 컴포넌트 108a
수신 컴포넌트 110a
송신 컴퓨터 104
수신 컴퓨터 106
송신 컴포넌트 108b
송신 컴포넌트 108c
수신 컴포넌트 110b
수신 컴포넌트 110c
컴포넌트 분석 서브시스템 116
컴포넌트 저장소 114
컴포넌트 관리자 서브시스템 112
컴포넌트 생성 서브시스템 118

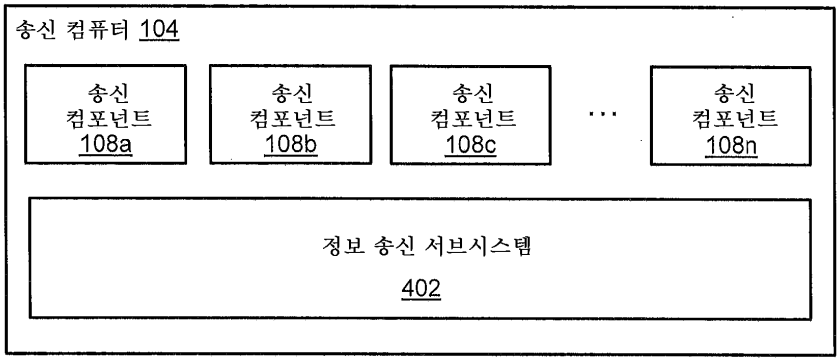
도면2


환경 102e
컴포넌트 분석 서브시스템
116
202a
202b
환경 102a
202a
202b
212
214
환경 102b
컴포넌트 관리자
서브시스템
112
208
210
환경 102c
컴포넌트 저장소 114
202a
202b
환경 102d
컴포넌트 생성
서브시스템
118
204
206

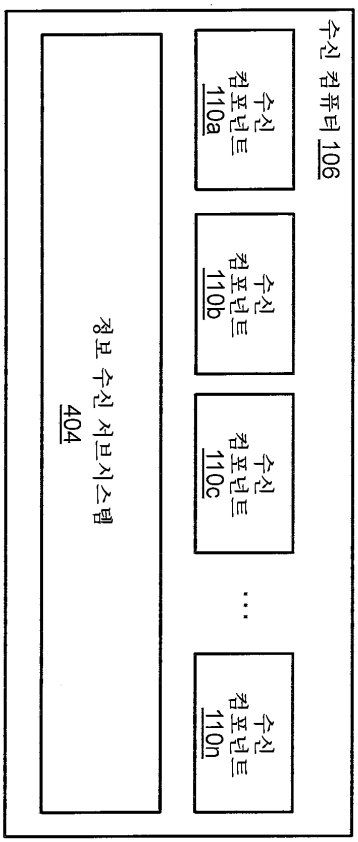
도면3


컴포넌트 생성 서브시스템 118
컴포넌트 작성 서브시스템
302
컴포넌트 실링 서브시스템
304
컴포넌트 엑스포팅 서브시스템
306

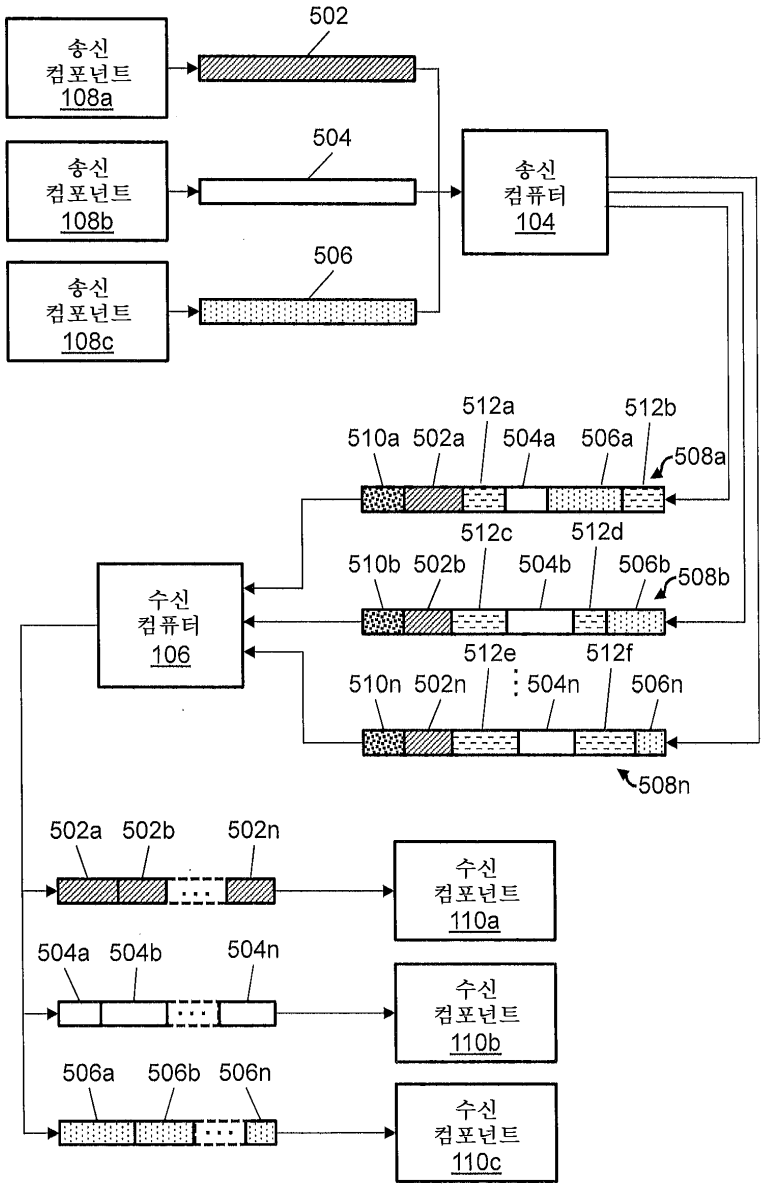
도면4a


송신 컴퓨터 104
송신
컴포넌트
108a
송신
컴포넌트
108b
송신
컴포넌트
108c
...
송신
컴포넌트
108n
정보 송신 서브시스템
402

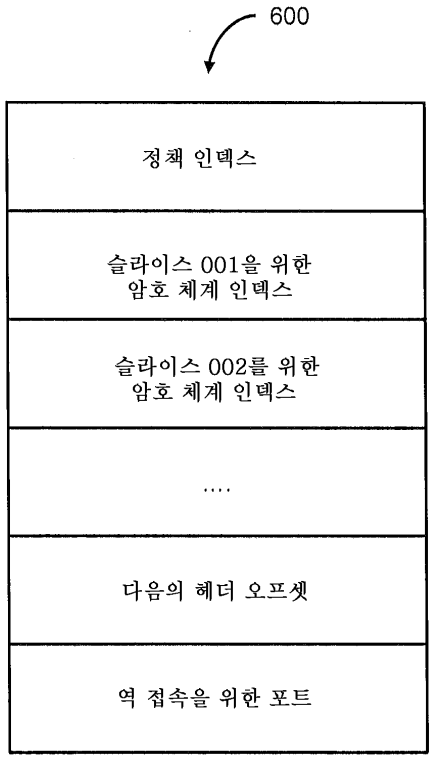
도면4b


수신 컴퓨터 106
수신 컴포넌트 110a
수신 컴포넌트 110b
수신 컴포넌트 110c
...
수신 컴포넌트 110n
정보 수신 서브시스템 404

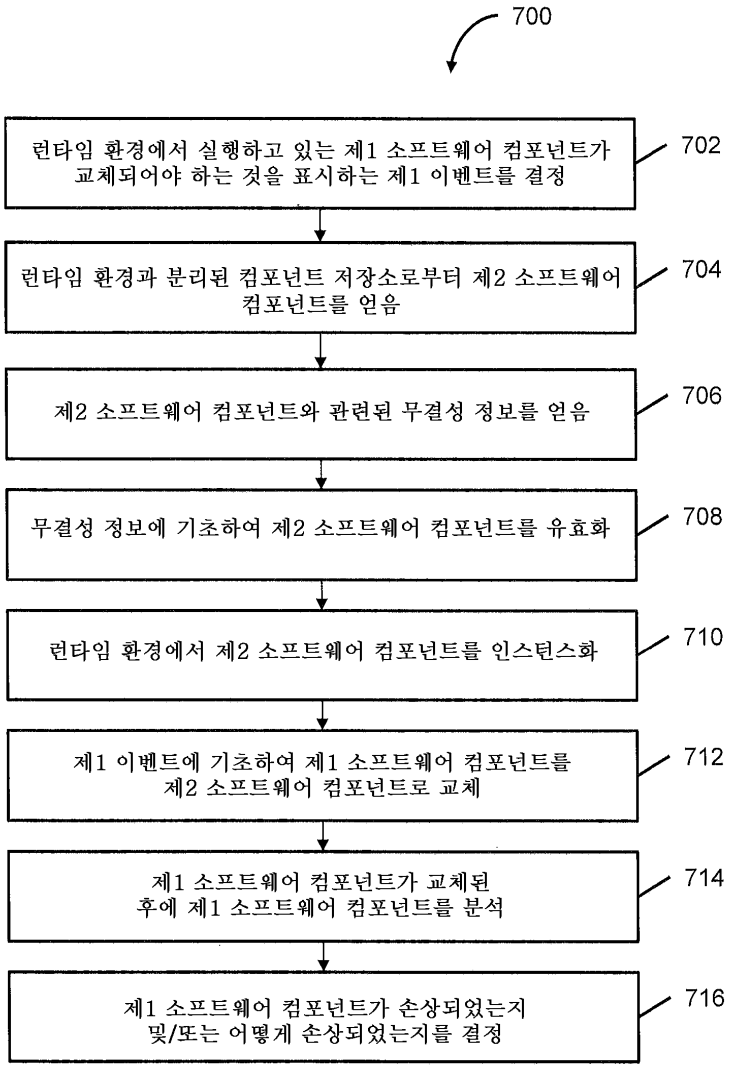
도면5


송신 컴포넌트 108a
송신 컴포넌트 108b
송신 컴포넌트 108c
502
504
506
송신 컴퓨터 104
수신 컴퓨터 106
512a
512b
510a
502a
504a
506a
508a
512c
512d
510b
502b
504b
506b
508b
512e
512f
510n
502n
504n
506n
508n
502a
502b
502n
수신 컴포넌트 110a
504a
504b
504n
수신 컴포넌트 110b
506a
506b
506n
수신 컴포넌트 110c

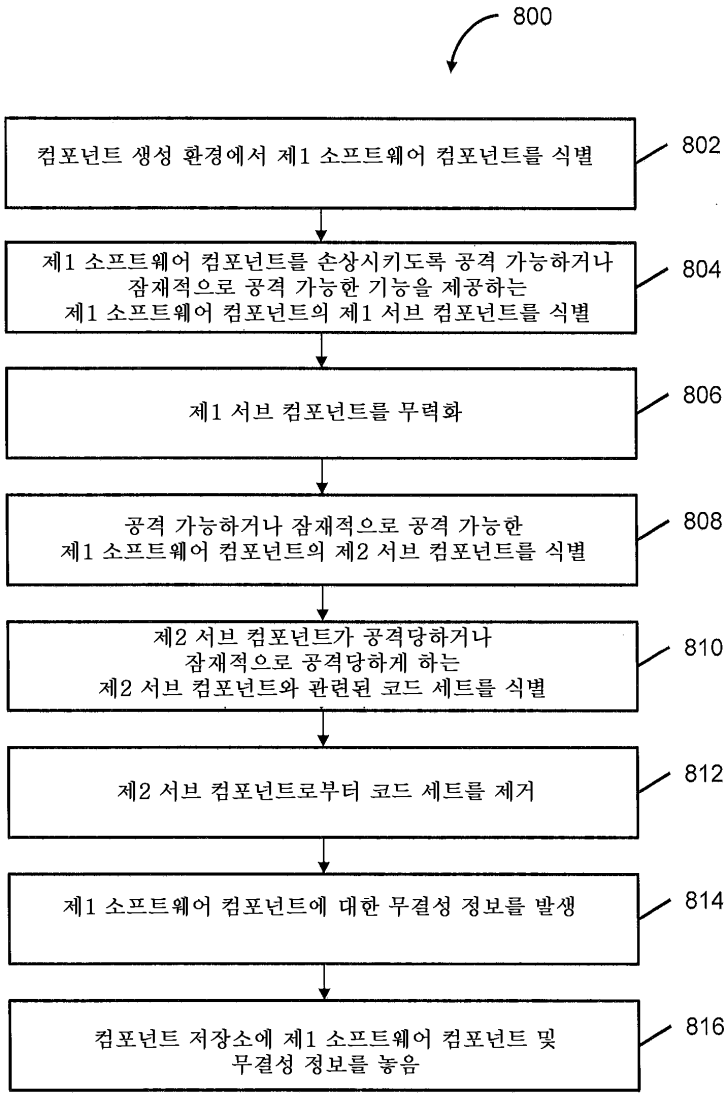
도면6


600
정책 인덱스
슬라이스 001을 위한
암호 체계 인덱스
슬라이스 002를 위한
암호 체계 인덱스
....
다음의 헤더 오프셋
역 접속을 위한 포트

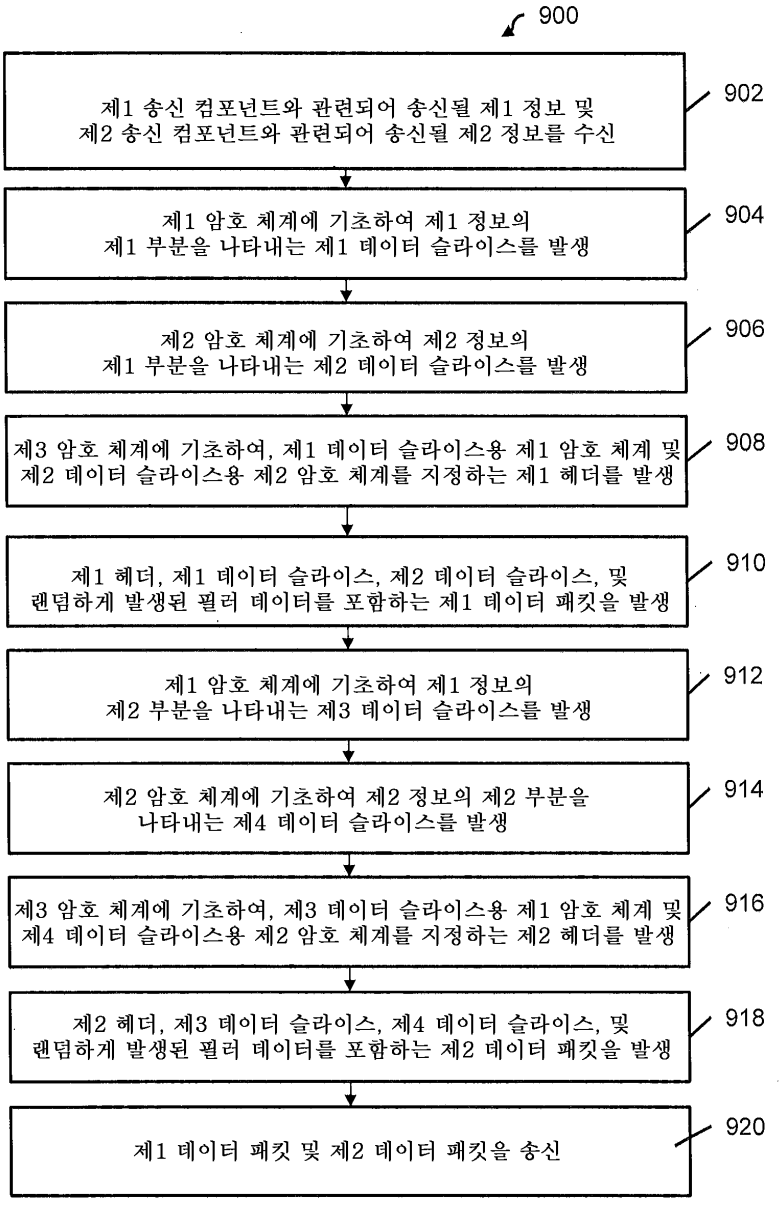
도면7


700
런타임 환경에서 실행하고 있는 제1 소프트웨어 컴포넌트가 교체되어야 하는 것을 표시하는 제1 이벤트를 결정
702
런타임 환경과 분리된 컴포넌트 저장소로부터 제2 소프트웨어 컴포넌트를 얻음
704
제2 소프트웨어 컴포넌트와 관련된 무결성 정보를 얻음
706
무결성 정보에 기초하여 제2 소프트웨어 컴포넌트를 유효화
708
런타임 환경에서 제2 소프트웨어 컴포넌트를 인스턴스화
710
제1 이벤트에 기초하여 제1 소프트웨어 컴포넌트를 제2 소프트웨어 컴포넌트로 교체
712
제1 소프트웨어 컴포넌트가 교체된 후에 제1 소프트웨어 컴포넌트를 분석
714
제1 소프트웨어 컴포넌트가 손상되었는지 및/또는 어떻게 손상되었는지를 결정
716

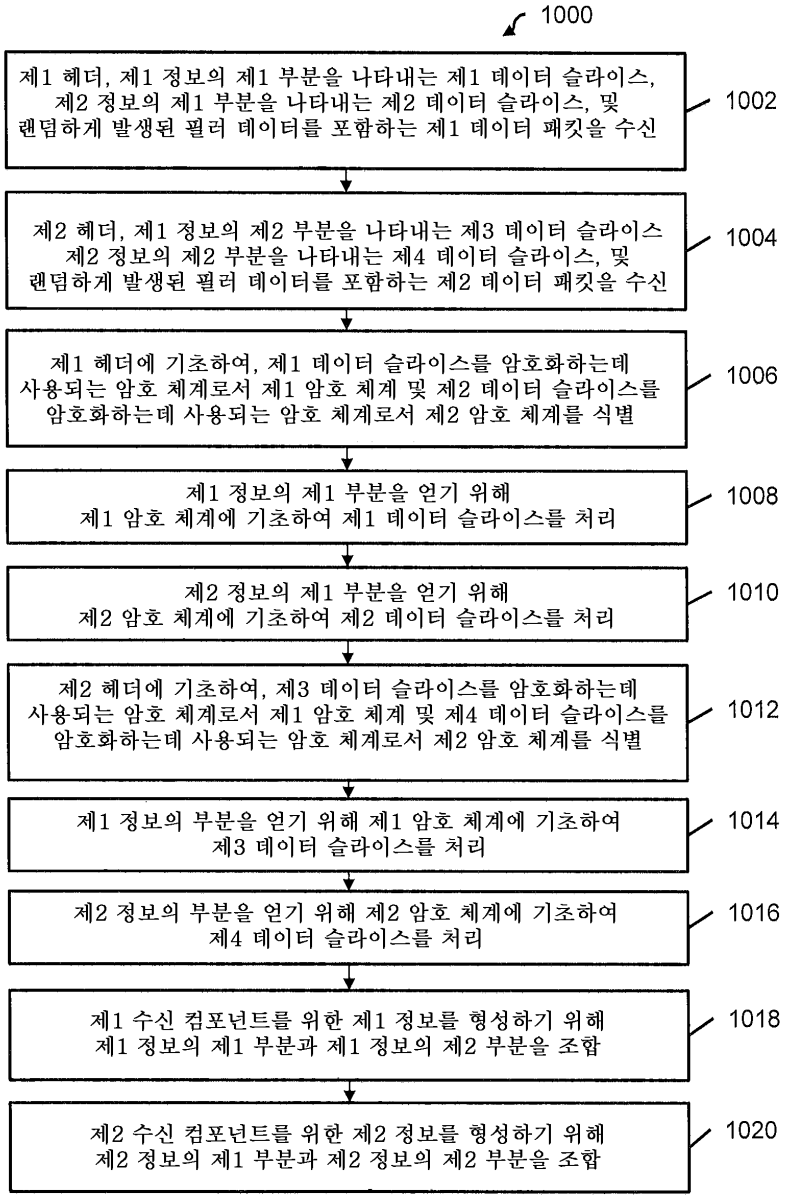
도면8


800
컴포넌트 생성 환경에서 제1 소프트웨어 컴포넌트를 식별
802
제1 소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 기능을 제공하는 제1 소프트웨어 컴포넌트의 제1 서브 컴포넌트를 식별
804
제1 서브 컴포넌트를 무력화
806
공격 가능하거나 잠재적으로 공격 가능한 제1 소프트웨어 컴포넌트의 제2 서브 컴포넌트를 식별
808
제2 서브 컴포넌트가 공격당하거나 잠재적으로 공격당하게 하는 제2 서브 컴포넌트와 관련된 코드 세트를 식별
810
제2 서브 컴포넌트로부터 코드 세트를 제거
812
제1 소프트웨어 컴포넌트에 대한 무결성 정보를 발생
814
컴포넌트 저장소에 제1 소프트웨어 컴포넌트 및 무결성 정보를 놓음
816

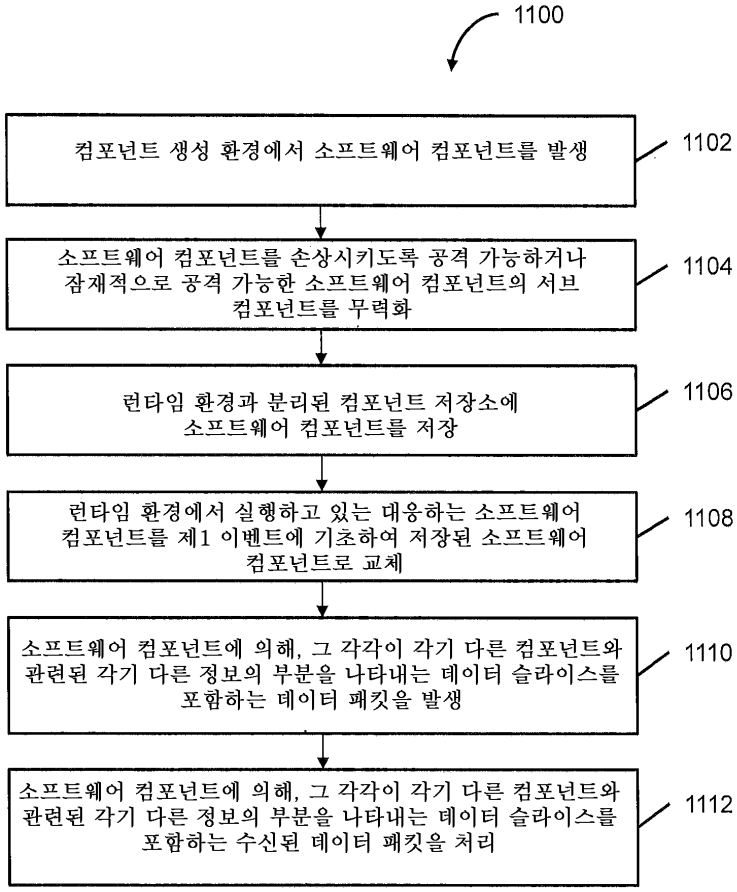
도면9


900
제1 송신 컴포넌트와 관련되어 송신될 제1 정보 및 제2 송신 컴포넌트와 관련되어 송신될 제2 정보를 수신
902
제1 암호 체계에 기초하여 제1 정보의 제1 부분을 나타내는 제1 데이터 슬라이스를 발생
904
제2 암호 체계에 기초하여 제2 정보의 제1 부분을 나타내는 제2 데이터 슬라이스를 발생
906
제3 암호 체계에 기초하여, 제1 데이터 슬라이스용 제1 암호 체계 및 제2 데이터 슬라이스용 제2 암호 체계를 지정하는 제1 헤더를 발생
908
제1 헤더, 제1 데이터 슬라이스, 제2 데이터 슬라이스, 및 랜덤하게 발생된 필러 데이터를 포함하는 제1 데이터 패킷을 발생
910
제1 암호 체계에 기초하여 제1 정보의 제2 부분을 나타내는 제3 데이터 슬라이스를 발생
912
제2 암호 체계에 기초하여 제2 정보의 제2 부분을 나타내는 제4 데이터 슬라이스를 발생
914
제3 암호 체계에 기초하여, 제3 데이터 슬라이스용 제1 암호 체계 및 제4 데이터 슬라이스용 제2 암호 체계를 지정하는 제2 헤더를 발생
916
제2 헤더, 제3 데이터 슬라이스, 제4 데이터 슬라이스, 및 랜덤하게 발생된 필러 데이터를 포함하는 제2 데이터 패킷을 발생
918
제1 데이터 패킷 및 제2 데이터 패킷을 송신
920

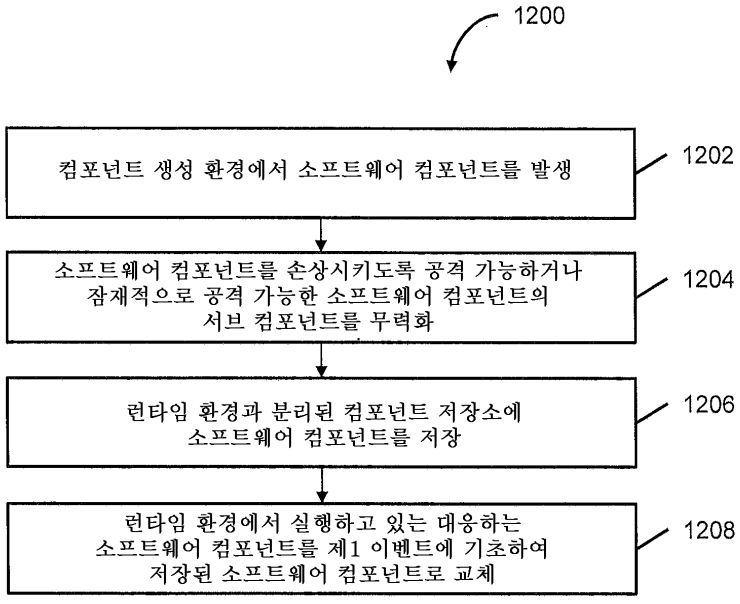
도면10


1000
제1 헤더, 제1 정보의 제1 부분을 나타내는 제1 데이터 슬라이스, 제2 정보의 제1 부분을 나타내는 제2 데이터 슬라이스, 및 랜덤하게 발생된 필러 데이터를 포함하는 제1 데이터 패킷을 수신
1002
제2 헤더, 제1 정보의 제2 부분을 나타내는 제3 데이터 슬라이스 제2 정보의 제2 부분을 나타내는 제4 데이터 슬라이스, 및 랜덤하게 발생된 필러 데이터를 포함하는 제2 데이터 패킷을 수신
1004
제1 헤더에 기초하여, 제1 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계 및 제2 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계를 식별
1006
제1 정보의 제1 부분을 얻기 위해 제1 암호 체계에 기초하여 제1 데이터 슬라이스를 처리
1008
제2 정보의 제1 부분을 얻기 위해 제2 암호 체계에 기초하여 제2 데이터 슬라이스를 처리
1010
제2 헤더에 기초하여, 제3 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제1 암호 체계 및 제4 데이터 슬라이스를 암호화하는데 사용되는 암호 체계로서 제2 암호 체계를 식별
1012
제1 정보의 부분을 얻기 위해 제1 암호 체계에 기초하여 제3 데이터 슬라이스를 처리
1014
제2 정보의 부분을 얻기 위해 제2 암호 체계에 기초하여 제4 데이터 슬라이스를 처리
1016
제1 수신 컴포넌트를 위한 제1 정보를 형성하기 위해 제1 정보의 제1 부분과 제1 정보의 제2 부분을 조합
1018
제2 수신 컴포넌트를 위한 제2 정보를 형성하기 위해 제2 정보의 제1 부분과 제2 정보의 제2 부분을 조합
1020

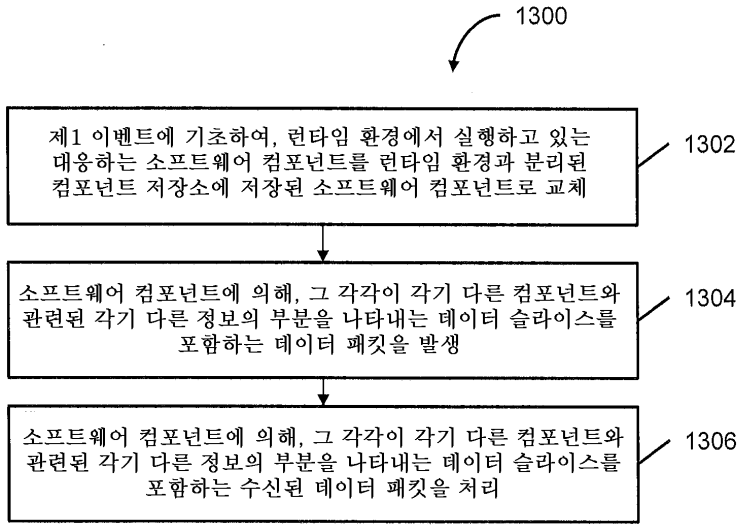
도면11


1100
컴포넌트 생성 환경에서 소프트웨어 컴포넌트를 발생
1102
소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 소프트웨어 컴포넌트의 서브 컴포넌트를 무력화
1104
런타임 환경과 분리된 컴포넌트 저장소에 소프트웨어 컴포넌트를 저장
1106
런타임 환경에서 실행하고 있는 대응하는 소프트웨어 컴포넌트를 제1 이벤트에 기초하여 저장된 소프트웨어 컴포넌트로 교체
1108
소프트웨어 컴포넌트에 의해, 그 각각이 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함하는 데이터 패킷을 발생
1110
소프트웨어 컴포넌트에 의해, 그 각각이 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함하는 수신된 데이터 패킷을 처리
1112

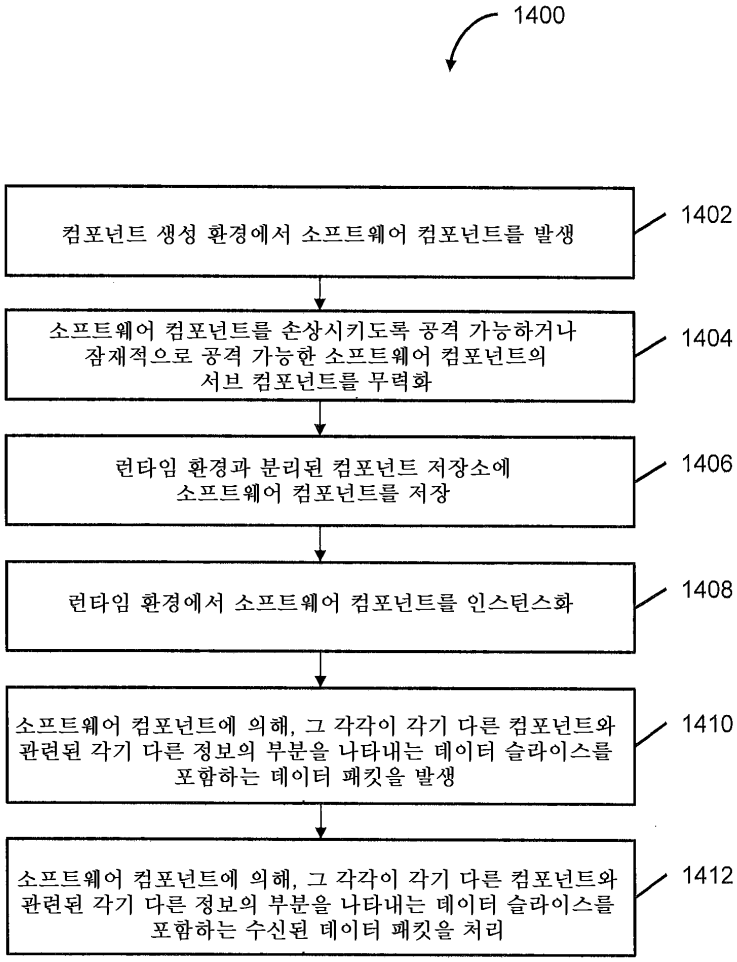
도면12


1200
컴포넌트 생성 환경에서 소프트웨어 컴포넌트를 발생
1202
소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 소프트웨어 컴포넌트의 서브 컴포넌트를 무력화
1204
런타임 환경과 분리된 컴포넌트 저장소에 소프트웨어 컴포넌트를 저장
1206
런타임 환경에서 실행하고 있는 대응하는 소프트웨어 컴포넌트를 제1 이벤트에 기초하여 저장된 소프트웨어 컴포넌트로 교체
1208

도면13


1300
제1 이벤트에 기초하여, 런타임 환경에서 실행하고 있는 대응하는 소프트웨어 컴포넌트를 런타임 환경과 분리된 컴포넌트 저장소에 저장된 소프트웨어 컴포넌트로 교체
1302
소프트웨어 컴포넌트에 의해, 그 각각이 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함하는 데이터 패킷을 발생
1304
소프트웨어 컴포넌트에 의해, 그 각각이 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함하는 수신된 데이터 패킷을 처리
1306

도면14


1400
컴포넌트 생성 환경에서 소프트웨어 컴포넌트를 발생
1402
소프트웨어 컴포넌트를 손상시키도록 공격 가능하거나 잠재적으로 공격 가능한 소프트웨어 컴포넌트의 서브 컴포넌트를 무력화
1404
런타임 환경과 분리된 컴포넌트 저장소에 소프트웨어 컴포넌트를 저장
1406
런타임 환경에서 소프트웨어 컴포넌트를 인스턴스화
1408
소프트웨어 컴포넌트에 의해, 그 각각이 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함하는 데이터 패킷을 발생
1410
소프트웨어 컴포넌트에 의해, 그 각각이 각기 다른 컴포넌트와 관련된 각기 다른 정보의 부분을 나타내는 데이터 슬라이스를 포함하는 수신된 데이터 패킷을 처리
1412