(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0037025
(43) 공개일자 2008년04월29일

(51) Int. Cl.
H04N 7/173 (2006.01)
(21) 출원번호 10-2008-7003597
(22) 출원일자 2008년02월14일
심사청구일자 없음
번역문제출일자 2008년02월14일
(86) 국제출원번호 PCT/GB2006/002619
국제출원일자 2006년07월14일
(87) 국제공개번호 WO 2007/007112
국제공개일자 2007년01월18일
(30) 우선권주장
0514492.8 2005년07월14일 영국(GB)

(71) 출원인
세큐스트림 테크놀로지스 에이에스
노르웨이, 7013 트론하임, 죠프만스가타 5
(72) 발명자
그리멘, 지슬레
노르웨이, 7052 트론하임, 타이홀트 알레 14에이
몽치, 크리스티앙
노르웨이, 7010 트론하임, 태럴즈가즈베이타 10
(74) 대리인
이경란

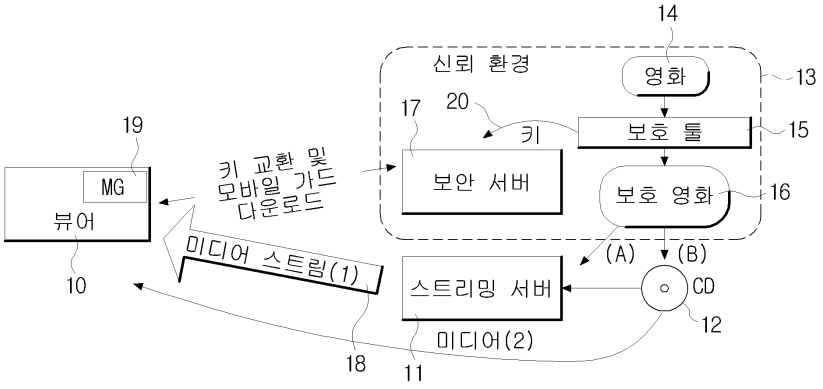
전체 청구항 수 : 총 39 항

(54) 멀티미디어 데이터 보호

(57) 요 약

본 발명은 (a) 상기 작품의 시간적으로 분리된 각 세그먼트들에 대응되는 일련의 서로 다른 키들을 이용하여 상기 작품을 암호화하는 단계, (b) 보안 서버에서 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-알고리즘은 클라이언트 상태의 함수인 결과를 포함함-, (c) 클라이언트에서 코드를 실행하고, 보안 서버에 결과를 반환하는 단계, (d) 결과가 변경되지 않은 뷰어를 나타내는지 여부를 결정하는 단계를 포함하며, (e) 서버에서 뷰어로 세그먼트를 전송하는 단계, (f) 전송된 세그먼트에 대응되는 키를 보안 원격 서버에서 뷰어로 보안 스트리밍하는 단계, (g) 획득된 미디어 키를 이용하여 세그먼트를 해독하는 단계, (h) 단계 (d)가 변조된 뷰어임을 나타내면, 다음 키들이 전송되는 것을 금지하고, 그렇지 않으면 단계 (e) 내지 단계 (g)를 반복하며 단계 (b) 내지 단계(d)를 반복하는 단계를 더 포함하는 클라이언트에 미디어 작품을 전송하는 방법을 제공한다.

대표도 - 도2



특허청구의 범위

청구항 1

미디어 작품을 클라이언트로 전송하는 방법에 있어서,

(a) 상기 작품의 시간적으로 구분된 각 세그먼트에 대응되는 일련의 다른 키들을 이용하여 상기 작품을 암호화하는 단계;

(b) 보안 서버에서 상기 클라이언트로 첫 번째 키를 보안 전송하고, 서버에서 상기 클라이언트로 상기 대응되는 세그먼트를 전송하는 단계;

(c) 상기 클라이언트에서, 상기 상응하는 세그먼트를 해독하기 위해 상기 첫 번째 키를 이용하는 단계;

(d) 상기 뷰어에서, 상기 해독된 부분을 제공하는 단계; 및

(f) 상기 단계 (b) 내지 (d)를 다음 세그먼트 및 키에 대하여 반복하는 단계를 포함하는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 2

제1항에 있어서, 키가 하나 이상의 세그먼트를 해독할 수 없도록, 상기 키들은 암호적으로 서로 독립되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 3

제1항 또는 제2항에 있어서, 상기 클라이언트가 상기 문서를 수신할 권한이 있다는 검사에 따라 키들이 제공되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 4

제1항, 제2항 또는 제3항에 있어서, 상기 키들은 상기 보안 서버와 상기 클라이언트 사이에 협업을 강제하기 위해 이용되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 5

항에 있어서, 각 키는 미리 설정된 길이의 세그먼트에 대응하는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 6

전술한 항 중 어느 한 항에 있어서, 상기 보안 서버는 상기 클라이언트로부터 원격에 있는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 7

전술한 항 중 어느 한 항에 있어서, 상기 키들은 랜덤 데이터 생성기 및 클라이언트에게 알려진 보안 서버의 공개키를 이용하는 키 교환 프로토콜을 이용하여 전송되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 8

전술한 항 중 어느 한 항에 있어서, 각 키는 상기 클라이언트에 의해 개별적으로 요청되어야 하는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 9

전술한 항 중 어느 한 항에 있어서, 상기 클라이언트가 변경되지 않았음을 보증하는 상기 클라이언트의 무결성 검사 단계를 더 포함하는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 10

제9항에 있어서, 상기 클라이언트 변경이 탐지되거나 및/또는 상기 클라이언트의 무결성 검사가 성공적이지 않은 경우, 보안 서버는 키 제공을 중지하는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 11

제9항 또는 제10항에 있어서, 상기 클라이언트의 무결성은 (여기서 정의된) 모바일 가드에 의하여 검사되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 12

제11항에 있어서, 상기 각 키는 상기 클라이언트의 무결성을 성공적으로 검증한 모바일 가드의 신뢰 간격 동안만 전송되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 13

제9항, 제10항 또는 제11항에 있어서, 상기 클라이언트가 변형되지 않았으면 정상 결과를 반환하게 되는 무작위로 생성된 알고리즘의 이용을 포함하는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 14

전술한 항 중 어느 한 항에 있어서, 키들의 공급이 중단되더라도 전송이 지속되도록 상기 세그먼트들은 대응하는 키들과 독립적으로 전송되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 15

전술한 항 중 어느 한 항에 있어서, 상기 미디어 작품은 기록물인 미디어 작품을 클라이언트로 전송하는 방법.

청구항 16

전술한 항 중 어느 한 항에 있어서, 상기 미디어 작품은 라이브 공연인 미디어 작품을 클라이언트로 전송하는 방법.

청구항 17

전술한 항 중 어느 한 항에 있어서, 상기 미디어 작품은 원격 서버에서 상기 클라이언트로 스트리밍되는 미디어 작품을 클라이언트로 전송하는 방법.

청구항 18

데이터를 클라이언트로 전송하는 방법에 있어서,

(a) 상기 데이터를 상기 클라이언트로 전송하는 단계;

(b) 보안 서버에서 상기 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-상기 알고리즘은 상기 클라이언트의 상태의 함수인 결과를 포함함-;

(c) 상기 클라이언트에서 상기 코드를 실행하고, 상기 보안 서버에 상기 결과를 반환하는 단계; 및

(d) 상기 결과가 변경되지 않은 뷰어임을 나타내는지를 결정하는 단계를 포함하는 데이터를 클라이언트로 전송하는 방법.

청구항 19

제19항에 있어서, 상기 데이터는 미디어 작품인 데이터를 클라이언트로 전송하는 방법.

청구항 20

제18항 또는 제19항에 있어서, 상기 클라이언트는 단계 (d)가 수행될 때까지 상기 데이터의 사용이 금지되는 데이터를 클라이언트로 전송하는 방법.

청구항 21

제18항, 제19항 또는 제20항에 있어서, (e) 결과가 변경되지 않은 클라이언트임을 나타내지 않는 경우 상기 데이터 및/또는 상기 데이터 해독에 필요한 키의 전송을 중지하는 단계를 더 포함하는 데이터를 클라이언트로 전송하는 방법.

청구항 22

제18항 내지 제21항 중 어느 한 항에 있어서, 상기 코드는 상기 클라이언트 프로그램 코드 및/또는 이미지가 입력되는 체크섬 연산을 포함하는 데이터를 클라이언트로 전송하는 방법.

청구항 23

제22항에 있어서, 상기 체크섬 연산은 그 입력으로 랜덤 넘버를 포함하는 데이터를 클라이언트로 전송하는 방법.

청구항 24

제23항에 있어서, 상기 클라이언트는 상기 보안 서버의 공개키로 랜덤 넘버를 해독하고, 해독된 랜덤 넘버는 미디어 키에 대한 요청 및 연산된 체크섬과 함께 상기 보안 서버로 전송되며, 상기 보안 서버는 상기 랜덤 넘버를 해독하여 상기 미디어 키를 암호화하기 위해 이용하며, 상기 보안 서버는 상기 체크섬의 자체 연산을 업데이트하기 위하여 상기 랜덤 넘버를 이용하고, 상기 보안 서버는 체크섬의 두 값을 비교하는 데이터를 클라이언트로 전송하는 방법.

청구항 25

제24항에 있어서, 상기 두 값이 동일하고 동일해야만, 상기 클라이언트가 미디어 키를 해독할 수 있도록 암호화된 미디어 키를 클라이언트로 전송하는 데이터를 클라이언트로 전송하는 방법.

청구항 26

제25항에 있어서, 상기 두 값이 동일하지 않으면, 상기 클라이언트 뷰어는 변경된 것으로 결정하는 데이터를 클라이언트로 전송하는 방법.

청구항 27

제18항 내지 제26항 중 어느 한 항에 있어서, 상기 코드는 뷰어 상에서 모호화 작업(들)을 수행하는 데이터를 클라이언트로 전송하는 방법.

청구항 28

제27항에 있어서, 상기 모호화 작업(들)은 실행 중인 뷰어의 메모리 이미지를 무작위화하는 단계를 포함하는 데이터를 클라이언트로 전송하는 방법.

청구항 29

제27항에 있어서, 상기 모호화 작업들은 여기에 정의된 하나 또는 이상의 코드 재배치, 코드 변형, 데이터 재배치, 데이터 숨김을 포함하는 데이터를 클라이언트로 전송하는 방법.

청구항 30

제18항 내지 제29항 중 어느 한 항에 있어서, 상기 모바일 가드는 모호화되는 데이터를 클라이언트로 전송하는 방법.

청구항 31

실행 중인 뷰어의 메모리 이미지를 무작위화하는 단계를 포함하는 실행 중인 뷰어의 모호화 방법.

청구항 32

클라이언트로 미디어 작품을 전송하는 방법에 있어서,

(a) 상기 작품의 시간적으로 분리된 각 세그먼트들에 대응되는 일련의 서로 다른 키들을 이용하여 상기 작품을 암호화하는 단계;

(b) 보안 서버에서 상기 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-상기 알고리즘은 상기 클라이언트의 상태의 함수인 결과를 포함함-;

(c) 상기 클라이언트에서 상기 코드를 실행하고, 상기 보안 서버에 상기 결과를 반환하는 단계;

(d) 상기 결과가 변경되지 않은 뷰어를 나타내는지 여부를 결정하는 단계;

(e) 상기 서버에서 상기 뷰어로 세그먼트를 전송하는 단계;

(f) 상기 결과가 변경되지 않은 뷰어를 나타내는 경우, 상기 전송된 세그먼트에 대응되는 키를 보안 원격 서버에서 상기 뷰어로 보안 스트리밍하는 단계;

(g) 키를 이용하여 상기 세그먼트를 해독하는 단계를 포함하는 클라이언트로 미디어 작품을 전송하는 방법.

청구항 33

제32항에 있어서, 각 알고리즘-포함 소프트웨어 코드는 연관된 신뢰 간격을 가지며 상기 신뢰 간격 동안에 복수의 키가 상기 클라이언트로 스트리밍되는 클라이언트로 미디어 작품을 전송하는 방법.

청구항 34

제32항 또는 제33항에 있어서, (h) 단계 (b) 내지 단계 (g)를 반복하는 단계를 더 포함하는 클라이언트로 미디어 작품을 전송하는 방법.

청구항 35

클라이언트로 미디어 작품을 전송하는 방법에 있어서,

(a) 상기 문서의 시간적으로 분리된 각 세그먼트들에 대응되는 일련의 서로 다른 키들을 이용하여 상기 작품을 암호화하는 단계;

(b) 보안 서버에서 상기 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-상기 알고리즘은 상기 클라이언트의 상태의 함수인 결과를 포함함-;

(c) 상기 클라이언트에서 상기 코드를 실행하고, 상기 보안 서버에 상기 결과를 반환하는 단계;

(d) 상기 결과가 변경되지 않은 클라이언트를 나타내는지 여부를 결정하는 단계를 포함하며,

(e) 서버에서 상기 클라이언트로 세그먼트를 전송하는 단계;

(f) 상기 전송된 세그먼트에 대응되는 키를 보안 원격 서버에서 상기 클라이언트로 보안 스트리밍하는 단계;

(g) 상기 획득된 미디어 키를 이용하여 세그먼트를 해독하는 단계;

(h) 단계 (d)가 변조된 클라이언트임을 나타내면, 다음 키들이 전송되는 것을 금지하고, 그렇지 않으면 단계 (e) 내지 단계 (g)를 반복하는 단계를 더 포함하는 클라이언트로 미디어 작품을 전송하는 방법.

청구항 36

제35항에 있어서, (i) 단계 (b) 내지 단계 (d)를 반복하는 단계를 더 포함하는 클라이언트로 미디어 작품을 전송하는 방법.

청구항 37

보안 소스에서 클라이언트 프로그램을 구동하는 클라이언트 컴퓨터로 소프트웨어 코드를 전송하는 단계-상기 소프트웨어 코드는 상기 클라이언트 프로그램의 상태에 종속하는 알고리즘을 포함함-, 상기 소프트웨어 코드를 실행하고 상기 소스에게 그 결과값을 반환하는 단계를 포함하되, 상기 소스는 클라이언트 프로그램의 무결성을 결정할 수 있는 클라이언트 프로그램의 무결성을 검사하는 방법.

청구항 38

제37항에 있어서, 상기 클라이언트 프로그램은 인터넷 뱅킹, 온라인 게임 및 배포된 연산 중 하나에서 이용되는 클라이언트 프로그램의 무결성을 검사하는 방법.

청구항 39

전술한 항 중 어느 하나의 방법에 따라 동작하는 장치.

명 세 서

발명의 상세한 설명

<1> 본 발명은 영화, TV 쇼, 오디오 문서 등과 같은 시간적 차원을 갖는 멀티미디어 작품의 보안 분배에 관한 것이다. 특히, 본 발명은 사용자가 작품의 불법 복제물을 획득하지 못하도록 방지하면서 사용자에게 상기 작품들을 안전하게 분배하는 시스템에 관한 것이다. 본 발명의 측면들은 온라인 뱅킹, 게임 등 다른 서버 클라이언트 환경에서 또한 응용된다.

<2> 예술적 작품의 불법 복제는 지속적인 문제점이었다. 영화 산업 초기, 완성된 필름을 불법 복제하는 것이 가능하였지만, 이는 전문 장비에 접근 가능한 사람들 외에는 비용이 많이 소용되고 실용적이지 않았다. 홈 비디오 레코더의 출현으로 인하여, 영화 및 기타 기록 프로그램들에 대한 새로운 시장은 제작자에게 가능하게 되었지만, 동시에 상기 기록물을 불법 복제하거나 불법 배분하는 것이 가능해졌다.

<3> 오늘날, DVD 포맷은, 고 품질 재생과 더욱 편리하고 간편한 데이터 기록매체를 제공하며, 빠르게 비디오를 대체하고 있다. 또한, 저렴한 광대역 인터넷 접속의 출현으로, 현재 원격 서버로부터 가정용 컴퓨터로 영화 및 기타 미디어를 다운로딩 또는 스트리밍하는 부상하는 시장이 있다.

<4> 미디어 작품을 다운로드하는 경우, 그 복제물이 컴퓨터의 하드 드라이브 상에 저장되고, 비디오 기록물을 시청하는 것과 유사하게 미디어 작품은 통상적으로 사용자에 의해 반복적으로 시청될 수 있다. 스트리밍 컨텐츠는 생중계이든 기록물이든 거의 실시간으로(일부 버퍼링을 수행하는데 필요한 약간의 시간이 지연됨) 컴퓨터에 전송되어(통상적인 TV 프로그램과 유사함) 시청하게 된다. 라디오 및 일부 TV 방송국들이 이러한 방식으로 그들의 컨텐츠를 제공한다는 것은 잘 알려진 사실이다.

<5> 이러한 기술의 진보는 미디어 회사에게 유망하고 새로운 시장의 발전을 허용하지만, 이에 따라 작품들의 불법 복제물의 생산 및 분배를 방지하여야 하는 문제점이 또한 존재한다. 저렴한 가정용 컴퓨터도 컨텐츠를 DVD에 기록하는 능력을 가지는 것이 현재 일반적이다.

<6> 따라서, 기술은 이러한 복제를 방지하는 방향으로 발전되었다. 일반적인 방법으로, 미디어 제공자(여기서는 "컨텐츠 제공자"로 칭함)는, 예를 들어, "미디어 작품"이라고 총칭되는 영화와 같은 부호화된 미디어 작품을 소유한다. 이들은 사용자가 암호화된 미디어 작품의 복제물을 생성할 수 없도록 사용자의 클라이언트 프로그램/뷰어에 분배되고 제공되어야 한다. 전달은 네트워크를 통한 스트리밍이나 예를 들어, DVD와 같은 물리적 매체를 클라이언트에 전달함으로써 이루어질 수 있다.

<7> 작품이 네트워크를 통하여 전송되는 경우, 통상적으로 작품이 제3자에 의하여 복제되거나 인터셉트되는 것을 방지하기 위하여 암호화 수단으로 작품을 보호한다. 우리는 이러한 암호화 수단을 "전송 암호화"라 칭한다. (보안 수단인 암호화는 작품이 변환되고, 통상적으로 쉽고 효율적으로 전송될 수 있는 형태로 압축된다는 점에서, 부호화하고는 구별됨) 암호화 기술은 컴퓨터 네트워크를 통한 통신이 적절한 방법으로 보호될 수 있을 만큼 충분히 잘 개발되어 있고 안전하다

<8> 미디어 작품이 클라이언트에 전달되기 전에, 컨텐츠 소유자는 암호화 수단을 이용하여 부호화된 미디어 문서를 보호한다. 안전한 "제공자 환경"에서, 암호화 툴은 암호화하고 부호화된 미디어 작품인 "암호화된 작품"을 생성하기 위하여 "미디어 키"를 작품을 암호화하는데 이용된다

<9> 이는 클라이언트가 작품을 해독하도록 허가된 미디어 키를 소지한 경우에만, 작품을 이용할 수 있게 하기 위함이다. 이는 예를 들어 DVD 플레이어 및 DVD등의 클라이언트 프로그램/뷰어/플레이어 및/또는 미디어 내에 삽입된다. (클라이언트 프로그램/뷰어/플레이어는 독립된 장치이거나 컴퓨터 상의 뷰어 소프트웨어 프로그램일 수 있다.)

<10> 또 다른 방법은, 도 1에 개략적으로 도시되어 있으며, 요청에 따라 미디어 키가 라이센스 서버(1)로부터 획득된다. 이는 미디어 작품을 스트리밍할 수 있게 한다. 이러한 모델을 뒷받침하기 위하여, 암호화 툴(2)은 부가 정보와 함께 미디어 키를 라이센스(3)에 포장하고, 이를 라이센스 서버(1)에 전송한다. 이후, 클라이언트는 암호화되고 부호화된 미디어 스트림(4)을 스트리밍 서버(5)로부터 수신하고, 클라이언트에 제공되기 전에 뷰어(6)에서 해독된다. 암호화된 영화(7)를 시청하기 위하여, 뷰어는 라이센스 서버(도 1의 "개시 단계" 참조)로부터 미디어 키를 포함하는 라이센스를 요청한다.

<11> 뷰어가 라이센스(3)(및 따라서 미디어 키)를 수신하면, 스트리밍 서버(5)에 접속하여 암호화되고 부호화된 미디어 스트림(4)을 수신 받는다. 뷰어는 암호화되고 부호화된 미디어 스트림을 해독하고 이를 클라이언트에 제공하기 위해 미디어 키를 이용한다(도 1의 "스트리밍 단계" 참조).

<12> 전술한 시나리오의 큰 문제점은 뷰어가 호스트 상에서 실행되고, 클라이언트에 의하여 제어된다는 것이다. 따라서, 뷰어는 신뢰 환경(8)(영화(9)가 최초 암호화된 환경)에서 실행되지 않는다. 그러므로, 클라이언트가 뷰어를 변경할 위험이 있다. 뷰어가 통상적으로 미디어 스트림의 일부만 해독하고 복호하는 경우에도, 전 제공 과정 동안 부호화된 미디어 스트림의 모든 부분이 어느 시점에는 뷰어 메모리-이미지에 존재할 것이다. 다른 위험은, 뷰어의 메모리 이미지가 키를 포함해야 하기 때문에, 사용자는 미디어 키를 추출할 수도 있으며, 이 경우 사용자는 암호화되지 않은 부호화된 미디어의 복제물을 생성할 수 있다.

<13> 순수한 소프트웨어 기반 뷰어 뿐만 아니라 예를 들어, 전용 DVD 플레이어 등을 위한 하드웨어 기반 뷰어에서도 변경의 문제가 있다. 소프트웨어 기반 뷰어를 변경하는 것 보다 하드웨어 기반 뷰어를 변경하는 것이 더 어렵다 하더라도, 이것이 불가능하지 않다. 따라서, 이러한 문제점을 해결할 시스템의 필요성이 대두된다.

<14> 효율적인 보호 메커니즘에 대한 일반적인 요구 사항은 다음을 포함한다. 해독 비용이 적어도 작품의 가치와 동일한 수준일 수 있도록, 해독하는 것이 자원-집중적이어야 한다. 성공한 공격이 다른 곳에서도 적용될 수 있도록 일반화되서는 안 된다. 또한, 바람직하게는 탐지를 쉽게 할 수 있어야 한다. 이하에서 설명될 본 발명의 다양한 측면들은 이러한 요구 사항을 개시하며 본 발명의 바람직한 형태들은 이들을 모두 만족하는 시스템을 제공한다.

<15> 이하의 설명에서, 미디어 작품은 시간적 측면을 가지는 작품, 즉 적합한 순서로 실행되어야 하는 복수의 개념적인 단계들을 포함하는 작품이다. 상기 단계들은 일반적으로 연산적으로 서로 독립적이고, 독립적으로 처리될 수 있다. 대부분의 경우, 완전한 표현은 상당한 양의 시간을 소요하며, 영화의 경우 몇 분 또는 몇 시간이 소요된다.

<16> 본 발명의 일 측면에 따르면,

<17> (a) 상기 작품의 시간적으로 구분된 각 세그먼트에 대응되는 일련의 다른 키들을 이용하여 상기 작품을 암호화하는 단계;

<18> (b) 보안 서버에서 상기 클라이언트로 첫 번째 키를 보안 전송하고, 서버에서 상기 클라이언트로 상기 대응되는 세그먼트를 전송하는 단계;

<19> (c) 상기 클라이언트에서, 상기 상응하는 세그먼트를 해독하기 위해 상기 첫 번째 키를 이용하는 단계 ;

<20> (d) 상기 뷰어에서, 상기 해독된 부분을 제공하는 단계; 및

<21> (e) 상기 단계 (b) 내지 (d)를 다음 세그먼트 및 키에 대하여 반복하는 단계를 포함하는 미디어 작품을 클라이언트로 전송하는 방법이 제공된다.

<22> 본 발명은 앞서 정의된 바와 같이 시간적 측면을 가지는 어떠한 종류의 미디어 작품에도 적용될 수 있고, 영화들을 분배, 예를 들어 인터넷을 통하여 영화들을 스트리밍하는데 특히 유용하다.

<23> 문서를 일련의 세그먼트로 분할함으로써, 각 키가 하나의 세그먼트만을 해독할 수 있기 때문에, 즉 상기 키들이 기능적으로 독립되어 있기 때문에, 문서의 작은 일부분 이상을 복제하는 것은 비실용적이다. 따라서, 한번에 영화의 한 세그먼트만이 복제될 수 있다. 더욱이, 다른 키를 해제할 수 있는 마스터 키가 존재해서는 안되며, 바람직하게 상기 키들은 구조적으로 독립되어야 한다. 바람직하게, 통상적인 길이의 필름에는 수천 개의 서로 다른 키들이 이용되므로, 각 키는 미리 설정된 길이의 세그먼트, 예를 들어, 몇 초, 2초 또는 3초 이하, 가장 바람직하게는 1초 이하 길이의 세그먼트에 대응되는 것이 바람직하다. 대부분의 미디어 작품들은 실질적으로 완전한 경우에 중요한 가치를 가진다. 예를 들어, 마지막 몇 분이 손실된 영화는 일반적으로 가치가 거의 없을 것이다. 따라서, 영화를 불법으로 복제하려는 자는 각 세그먼트들 해독하여야 한다.

<24> 데이터 해독의 연속적 흐름을 유지하기 위하여, 본 발명의 일 실시예에서는 클라이언트가 현재 키 및 다음 키(들)을 요청할 수 있고, 메모리에 몇 개의 키들(예를 들어, 2, 3, 4 등)을 임시 저장할 수 있다.

<25> 통상적으로, 보안을 위한 서버는 뷰어로부터 떨어져 위치하는데, 본 명세서에서는 이를 "보안 서버"라 칭하기로 한다.

<26> 영화는 일반적으로 신뢰 환경에서 암호화된다. 바람직하게, 암호화 과정 동안 생성되는 키들은 보안 서버에 제공되며, 보안 서버는 신뢰 제공자 환경 내에 위치한다. 그러나, 키들이 보안 서버에서 뷰어로 전송되더라도, 영화 또는 다른 작품은 그 밖의 장소로부터 전송될 수 있다. 예를 들어, 영화 또는 기타 작품은 신뢰 환경 밖에 있는 개별 서버로부터 스트리밍될 수 있다. 그러므로, 본 발명의 바람직한 일 실시예에서, 영화가 신뢰 제공자 환경에서 암호화되면, 비보안 스트리밍 서버로 제공된다.

<27> 따라서, 이 구성에서, 클라이언트는, 원격 컴퓨터(예를 들어, 사용자의 PC) 상에서 구동되는 소프트웨어 뷰어 프로그램일 수 있으며, 키(미디어 키라 칭함)를 수신하기 위하여 보안 서버와 통신하고 개별 스트리밍 서버와 통신한다.

<28> 미디어 키는 클라이언트의 요청에 따라 클라이언트로 바람직하게는 전송되고, 이것은 랜덤 데이터 생성기 및 뷰어에게 알려진 보안 서버의 공개키를 이용하는 키 교환 프로토콜을 이용하여 바람직하게 이루어진다.

<29> 본 발명의 일 실시예에서, 다음 미디어 키를 획득하는 것이 필요하면, 클라이언트는 랜덤 데이터를 생성하고, 보안 서버의 공개키를 이용하여 랜덤 데이터를 암호화한다. 이후 암호화된 데이터는 다음 미디어 키에 대한 요청에 포함될 수 있으며, 바람직하게는 클라이언트를 식별하는 데이터와 함께, 보안 서버로 전송된다. 요청을 수신하면, 보안 서버는 클라이언트가 미디어 작품을 수신할 권한이 있는지 확인하고, 랜덤 데이터를 이용하여 키를 암호화하기 위해, 랜덤 데이터를 해독 및 추출하고, 랜덤 데이터 및 요청된 키를 이용하는 함수를 실행한다. 일 실시예에서, 랜덤 데이터와 요청된 키는 배타적 논리합될 수 있다. 이후, 결과는 클라이언트로 반환된다. 클라이언트가 결과를 수신하면, 이후 클라이언트는 대응 함수를 실행함으로써, 예를 들어, 결과를 키에 대한 최초 요청에 포함된 랜덤 데이터와 배타적 논리합함으로써, 요청된 키를 결과에서 추출할 수 있다. 이 방식으로, 암호화되고 부호화된 미디어 스트림은 뷰어의 소스 코드에 숨겨진 어떠한 비밀 키 없이도 해독될 수 있다.

<30> 바람직하게, 공개 키는 공개 키가 교환되는 "중간자" 공격을 방지하기 위하여 체크섬 연산에 포함되는 것이 바람직하다.

<31> 바람직한 형태의 프로토콜에서, 모바일 가드에 의해 검사된 클라이언트가 랜덤 데이터를 생성하는 클라이언트와 동일함을 보증하는 단계들이 적용된다. 이는 보증하는 단계는 미디어 키를 요청하기 위하여 이용된 랜덤 데이터를 포함하도록 입력을 체크섬까지 확장함으로써 실행될 수 있다. 따라서, 체크섬에 대한 입력은 클라이언트의 코드, 보안 서버의 공개 키, 및 키 요청과 함께 전송된 랜덤 데이터를 포함할 수 있다.

<32> 외부 엔트로피 소스는 감시될 수 있으므로, 랜덤 넘버를 생성하기 위해 사용되는 엔트로피 소스는 태스크가 스케줄 되고 인터럽트 되는 형태로 실행 환경 자체에 의해 생성된 것일 수 있다. 따라서, 랜덤 생성 프로세스는 뷰어의 현재 상태로부터의 데이터 및 실행중인 모바일 가드와 함께 시큐어 해쉬 알고리즘에 입력될 수 있는 서로 다른 연산 태스크들에 적용되는 복수의 스레드 생성을 포함할 수 있다.

<33> 클라이언트에 의해 수신될 연속적인 일련의 키들에 대한 필요는 사용자 협업을 강제하는데 이용될 수 있다. 따라서, 제공자가 요청한 일정 단계가 클라이언트에 의하여 수행되지 않는다면, 키 제공이 중지될 수 있다. 후술하는 바와 같이, 이 단계는 클라이언트의 무결성 검사일 수 있고, 바람직하게는 소위 "모바일 가드"가 뷰어가 변경되지 않았음을 나타내는 경우에만 새로운 키 요청에 응답한다.

<34> 모바일 가드를 적용하는 경우, 실제 클라이언트 뷰어/플레이어 프로그램 보다, 전술한 바람직한 키 교환 프로토콜에서 이용되는 랜덤 넘버를 생성할 수 있다.

<35> 이러한 협업 강제는 컨텐츠 제공자가 미디어 작품을 제어할 수 있기 때문에 가능하고, 작품의 시간적 특성으로 인하여 상기 작품은 후속 부분을 수신하기 위해서는 협업을 요구 받는 사용자에게 작은 부분들로 제공될 수 있음을 이해할 수 있다. 이는 라이센스가 전체 문서를 해제하여 사실상 시간적 특성을 무력화시키는 종래 시스템과 대비된다.

<36> 본 발명의 다른 실시예에서, 신뢰 환경은 확대되어 스트리밍 서버가 그 안에 포함될 수 있다. 이렇게 되면, 스트리밍 서버가 미디어 키를 생성하고, 전송 중인 미디어 스트림을 암호화할 수 있다. 이것은 각 미디어 스트림이 고유한 미디어 키 세트에 의해 암호화됨을 보증한다. 그것은 노출된 미디어 키는 동일한 영화의 서로 다른 복제물을 해독하는데 이용될 수 없음을 의미이다. 미디어 키의 분배를 촉진하기 위하여 스트리밍 서버는 보안 서버로 미디어 키를 전송하며, 전술한 바와 같이 뷰어에 미디어 키를 분배하는 할 수 있다. 그러나, 또 하나의 엔티티가 신뢰되어야 할 필요가 있으며, 전송 중 암호화는 복잡한 연산이라는 단점이 있다. 따라서, 일측면에서의 매우 높은 수준의 보안성과 다른 측면에서의 신뢰 환경의 복잡도 및 복잡한 연산간의 타협이 있다.

<37> 본 발명은 문서를 원격 서버로부터 스트리밍하는 구성들에 국한되지 아니한다. 문서가 암호화되었기 때문에, 문서는 어떠한 일반적인 방식으로도 분배될 수 있다. 따라서, 암호화된 문서는 로컬 서버 또는 물리적 매체(예를 들어, DVD)로부터 클라이언트로 제공될 수 있다. 이때, 문서는 로컬 서버 또는 물리적 매체로부터 뷰어로 전송될 수 있고, 전술한 방식으로 해독될 수 있다.

<38> 이러한 구성이 종래 기술에 비하여 현저하게 개선된 것이라 하더라도, 뷰어가 변조되어 해독된 작품(영화 등)이 기록되고 복제될 수 있는 위험성이 여전히 존재하고 있다. 따라서, 바람직하게, 본 발명은 뷰어가 변조되지 않았다는 것을 보증하는, 뷰어의 무결성을 검사하는 수단을 더 포함한다. 이는 주기적으로 및/또는 키가 요청된 경우, 체크섬과 같은 신호를 보안 서버로 전송하도록 프로그래밍함으로써 이루어질 수 있다. 그러한 신호는 뷰어의 상태에 의존하도록 디자인될 수 있어서 뷰어에 대한 변경은 신호를 바꿀 수 있다.

<39> 그러나, 이러한 방법도 뷰어의 실제 상태에 상관없이 "정상" 신호를 전송하도록 변경된 뷰어를 프로그래밍함으로써 극복될 수 있다는 위험성이 있다. 바람직하게는, 그러므로, 본 방법은 복수의 서로 다른 테스트를 이용하여 보안 서버가 뷰어에게 문의하는 것이 더 요구하며, 테스트는 시간에 따라 달라진다. 특히 바람직한 형태로, 테스트들은 뷰어가 변경되지 않았으면 정상 결과를 반환하는 랜덤하게 생성된 알고리즘들의 사용을 포함한다. 또한, 응답이 없거나 응답의 지연은 뷰어가 변경된 것으로 취급될 수 있다.

<40> 따라서, 바람직하게는 뷰어 변경의 탐지 및/또는 상기 뷰어의 무결성 검사가 성공적이지 않은 경우, 보안 서버는 키 제공을 중지한다.

<41> 가장 바람직한 구성은 알고리즘이 소프트웨어 코드(예를 들어, 기계어 코드)의 형태로 보안 서버에서 클라이언트로 전송되는 것이다. 소프트웨어 코드는 "모바일 가드"라 칭할 수 있고, 이하 더욱 상세하게 설명하기로 한다.

<42> 무결성 검사 시스템은 그 자체만으로 진보된 개념이며, 따라서, 다른 측면에서 보면,

<43> (a) 데이터를 클라이언트로 전송하는 단계;

<44> (b) 보안 서버에서 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-알고리즘은 클라이언트 상태의 함수인 결과를 포함함-;

<45> (c) 클라이언트에서 코드를 실행하는 단계;

<46> (d) 보안 서버에 결과를 반환하는 단계; 및

<47> (e) 결과가 변경되지 않은 뷰어임을 나타내는지를 결정하는 단계를 포함하는 데이터를 클라이언트로 전송하는 방법이 제공된다.

<48> 데이터는 예를 들어, 인터넷을 통하여 클라이언트로 스트리밍되는 미디어 작품이거나, 전술한 바와 같이 로컬 서버, DVD 또는 기타 미디어로부터 제공될 수 있다. 그러나, 더욱 상세하게 후술하겠지만, 상기 데이터는 서버와 클라이언트 사이에 전송 가능한 모든 종류의 데이터일 수 있다. 클라이언트는 컴퓨터 또는 TV 셋탑 박스와 같은 하드웨어 장치에서 구동되는 프로그램일 수 있다. 단계 (b)에 언급된 알고리즘은 문서의 일 부분이 전송되기 전에 전송되거나, 알고리즘에 앞서 작품의 전부 또는 일부가 전송될 수 있다. 바람직하게, 문서는 단계 (d)가 수행될 때까지 보이지 않는다.

<49> 단계 (d)의 결과에 따라, 적절한 동작을 행할 수 있다. 작품이 스트리밍되는 경우에, 뷰어가 변경되지 않은 것이 발견되면, 작품 및 해독에 필요한 키들의 전송은 계속하도록 일반적으로 허용될 수 있다. 그러나, 결과가 변경되지 않은 뷰어임을 나타내지 않는 경우 작품 및/또는 작품 해독에 필요한 키의 전송을 중지하는 추가 단계 (e)가 있을 수 있다. 바람직하게, 모바일 가드로부터 아무런 결과도 반환되지 않는다면, 이는 클라이언트가 변경된 것을 표시하는 것으로 볼 수 있다.

<50> 작품이 로컬 서버, DVD 등과 같은 로컬 소스로부터 전송되고 있으면, 동작은 문서를 해독하는데 필요한 키들의 전송을 중지하는 것일 수 있다.

<51> 또한, 클라이언트가 변경된 경우, 다른 동작이 수행될 수 있다. 예를 들어, 전송은 유지되고 사용자를 식별하기 위한 증거가 수집될 수 있다. 예를 들어, 범죄 행위를 발견하거나 문서들의 추후 불법 복제를 방지하기 위하여, 법적 또는 조사 행위를 취하는 것이 바람직하면 이것이 적절할 수 있다.

<52> 전술한 바와 같이, 변경된 클라이언트 식별에 대한 응답으로 취해진 동작은 해독 키들의 전송을 중지할 수

있다. 그러므로, 상기 방법은 작품을 시간적으로 분리된 복수의 세그먼트들로 분할하는 단계-상기 세그먼트들은 서로 다른 키들을 이용하여 암호화됨-를 더 포함할 수 있다. 이 키들은 순차적으로, 그리고 바람직하게는 전술한 바와 같이 클라이언트에 분배될 수 있다. 따라서, 키들의 분배가 중지되면, 작품의 나머지 부분은 해독될 수 없다.

<53> 상기 방법은 전술한 소프트웨어 코드에서 랜덤하게 생성된 비밀 알고리즘을 이용하여 수행되는 것이 바람직하다. 이러한 소위 강제 알고리즘은 클라이언트(예를 들어, 뷰어 프로그램)의 상태에 종속적인 결과를 생성하지만, 랜덤한 특성 때문에 사용자가 정상 결과를 추측할 수 없다. 바람직하게, 강제 알고리즘은 뷰어 프로그램의 코드가 입력되는 체크섬 연산을 포함한다. 알고리즘은 전체적으로 비밀이지만, 체크섬 연산은 메시지 다이제스트 알고리즘 5(MD5) (RFC1321 www.fags.org/rfcs/rfc1321.html)와 같이 공지된 것일 수 있으며, 무작위화된 입력 변경과 조합하여 이용될 수 있다.

<54> 입력 변경은 체크섬에 입력될 데이터를 다른 순서로 바꾸는 변조자(modifier)의 랜덤 생성을 참조한다. 일 구현에서, 소프트웨어 코드(이하, "모바일 가드"라 함)가 생성되면, 랜덤 시퀀스가 결정된다. 알고리즘이 실행되면, 뷰어로부터의 입력 코드는 동일한 크기의 n개의 블록으로 분할된다. 입력 코드는 이후 전술한 랜덤 시퀀스로 섞어지고, 결과는 이후 체크섬 알고리즘에 입력된다. 이러한 구성에서 체크섬 알고리즘 자체가 공개된 것이라 하더라도, 그 결과는 n개의 블록들이 체크섬 알고리즘에 삽입된 순서의 함수이다. 그 순서는 보안 서버에 알려지므로, 보안 서버는 반환된 결과가 무결한 뷰어를 나타내는지를 결정할 수 있다.

<55> 입력-필터링 체크섬의 또 다른 방식은 공지된 체크섬 알고리즘을 분석하고, 주어진 순서로 입력을 읽도록 체크섬 알고리즘을 재조합하는 것이다.

<56> 입력 필터링을 이용하는 대신, 스크래치로부터 체크섬 함수를 생성할 수 있다. 따라서, 입력은 l 개의 워드들(32 비트)로 분할될 수 있고, 입력으로부터 l 워드들과 워드를 출력하는 가변 영역으로부터 m 워드들을 읽는 함수 f가 생성된다. 함수는 차례대로 수행되는 임의 개수의 할당들을 포함하고, 체크섬은 f를 적용한 모든 결과들의 2^{32} 합 모듈로일 수 있다.

<57> 합성 함수는 체크섬 알고리즘의 거의 모든 코드가 무작위적으로 생성되고, 코드에 더 구조적 다양성을 이끌어내는 장점을 가진다. 빌딩 블록들은 매우 작기 때문에, 다른 알고리즘과의 더 쉬운 인터리빙을 허용한다.

<58> 또한, 소프트웨어 코드는 바람직하게 부가 알고리즘을 포함하며, 이는 비밀이거나 또는 비밀이 아닐 수 있다. 부가 알고리즘은 기능적으로 및/또는 공간적으로 비밀 알고리즘과 연관되는 것이 바람직하다. 이 방식으로 부가 알고리즘을 실행하지 않는 경우 비밀 알고리즘이 구현되지 않기 때문에, 클라이언트의 컴퓨터/뷰어는 부가 알고리즘 수행하도록 강제될 수 있다. 부가 알고리즘은 예를 들어, 뷰어 하드웨어의 무결성을 검사하는데 이용될 수 있다.

<59> 모바일 가드가 뷰어와 동일한 환경에 존재하므로, 잠정적으로 공격들에 취약하다. 사용자는 모바일 가드가 구현한 보호 방법들을 회피하기 위하여, 모바일 가드를 변경하려 할 수 있다. 모바일 가드에 대한 자동 공격들은 전술한 바와 같이 모바일 가드들이 부분적으로 무작위로 생성되는 것을 보증함으로써 방지될 수 있다. 또한, 모호화(obfuscation) 변형들이 모바일 가드에 적용될 수 있다. 모바일 가드는 모바일 가드에 특화된 방식으로 체크섬에 인터리빙된 오페이크 데이터 구조에 상기 체크섬을 숨길 수 있다. 변수들은 모바일 가드의 메모리에 무작위로 위치될 수 있고, 또한 모바일 가드의 명령들은 블록들로 분할되며, 메모리 내에 또한 무작위로 위치될 수 있다. 이는 바람직하게 모바일 가드로의 엔트리 포인트를 포함한다. 실제로, 하나의 모바일 가드에 대한 엔트리 포인트는 그 이전의 모바일 가드에 의하여 제공될 수 있다.

<60> 이런 단계들이 실행된다면, 모든 자동화된 공격을 개시하기 전에 모호화를 극복하기 위한 인적 공격이 필요해진다. 이러한 방식은 불가피하게 상당 시간이 소요되고, 그래서 연속적인 모바일 가드들간의 "신뢰 간격"이 충분히 짧으면, 효과가 없을 것이다. 다시 말하면, 모바일 가드들이 자주 교체되므로, 이것이 가치가 있기 위해서는 시간이 충분하지 않다. 따라서, 모호화 프로세스는 모바일 가드가 다른 모바일 가드로 교체되지 전의 시간 간격 동안, 모바일 가드를 부당 변경으로부터 보호한다.

<61> 해독된 영화 데이터가 저장된 컴퓨터 내의 기억 장소들을 염탐하는 관찰자의 위험이 있다. 공지된 기억 장소를 이용하면, 이후 데이터가 복제될 수 있다. 따라서, 임의의 기억 장소를 식별하여(장소-기반 식별) 코드를 위치하도록 실행하는 것은 바람직하지 않으며, 바람직하게는, 기억 장소들이 한 번 사용되면, 재사용되지 않아야 한다. 또한, 패턴 기반 식별(MPEG 헤더와 같은 시퀀스를 찾음으로써 코드가 획득될 수 있음) 또한 바람직하게는

피해야 한다.

<62> 따라서, 뷰어는 염탐으로 그 상태를 확인되지 않도록 모바일 가드에 의하여 보호되는 것이 바람직하다. 이를 하기 위하여, 모바일 가드는 상기 공격들을 방지하기 위한 하나 이상의 보호 알고리즘을 더 포함하는 것이 바람직하다. 클라이언트(예를 들어, 뷰어 프로그램)에서 모호화 작업을 수행(이하, "런타임 뷰어 모호화"라 함), 즉, 모호화가 실행되는 뷰어 상에서 수행될 수 있다. 이는 구동중인 뷰어의 메모리 이미지를 바꾼다.

<63> 상기 런타임 뷰어 모호화는 더 진보적 개념이라 사료되며, 따라서 또 다른 측면에 따르면, 본 발명은 실행 뷰어의 메모리 이미지를 무작위화하는 것을 포함하는 실행 뷰어의 모호화 방법을 제공한다.

<64> 런타임 모호화는 다음 기술들 중 하나 이상을 포함할 수 있다.

<65> 코드 재배치는 메모리 내에서 코드 블록들의 이동을 포함한다. 프로그램이 실행되면, 모바일 가드는 메모리의 다른 부분에 코드를 이동시키며, 이후 실행될 수 있다. 이러한 알고리즘은 체크섬 연산에 밀접하게 인터리빙되는 것이 바람직하다.

<66> 바람직하게, 코드 재배치는 (1) 프로그램 내의 모든 기본 빌딩 블록들을 식별하고, 재배치 가능한 작은 세그먼트들로 분할하고; (2) 모바일 가드를 실행하는 동안 상기 세그먼트들이 메모리 내의 무작위 기억 장소들에 재배열되며; (3) 새로운 코드 장소에 대응되는 모든 점프 명령들을 변경함으로써 구현된다. 결과적으로, 공격자는 모바일 가드가 실행되는 동안 변하는 메모리 이미지와 직면하게 될 것이다. 세그먼트들의 기억 장소가 보안 서버에서 제공되는 모바일 가드에 의하여 결정되기 때문에, 일정 기억 장소가 일정 데이터를 포함한다는 가정에 의존할 수 없는 공격자에게는 예측 불가능하게 된다.

<67> 데이터 재배치는 데이터를 이동하고 및 데이터에 접근하는 명령어들을 바꾸는 것을 포함한다. 다시, 새로운 기억 장소들은 무작위로 결정될 수 있다.

<68> 데이터 숨김은 장소 및 패턴 기반 식별의 문제를 야기한다. 한 가지 방식은, 데이터의 외관을 변조하는-효과적으로 데이터를 마스킹하는- 양방향 함수를 적용하는 것이다. 바람직하게는, 간단한 원-타임 패드 방식이 이용된다. 원-타임 패드는 랜덤 데이터의 어레이에 인덱스를 생성하는 새로이 생성된 모듈로 함수를 포함한다. 랜덤 데이터는 랜덤하고 민감한 부분들 사이에 배타적 논리합 연산자를 적용함으로써 민감 데이터를 바꾸는데 이용될 수 있다. 바람직하게는, 이것들과 랜덤 데이터 사이에 배타적 논리합 연산자를 적용한다.

<69> 하나의 방식은 데이터를 스크램블 및 언스크램블하여 민감 데이터를 스크램블 형태로 저장하고, 필요하면 언스크램블하며, 이후 리스크램블하거나 삭제한다. 그러나, 이는 데이터가 언스크램블된 경우, 짧은 윈도우를 남긴다.

<70> 그러나, 데이터가 프로세서의 레지스트리에 있을 때까지 언스크램블을 지연시키는 스트림 프로세싱을 이용할 수 있다.

<71> 따라서, 실제 컨텐츠 디코더는 마지막 해독 작용을 수행하기 위하여 변경될 수 있는데, 이는 새로운 데이터를 필요로 한다. 이는 메인 메모리에 어떠한 해독된 데이터도 존재하지 않을 것이라는 것을 의미한다. 이는 다음의 단계들을 이용하여 제공될 수 있다.

<72> a) 모바일 가드는 필요한 경우 마지막 해독 단계를 수행하는 디코더를 변경한다;

<73> b) 다음 암호화된 세그먼트를 획득한다;

<74> c) 암호화된 세그먼트에 대한 미디어 키를 획득한다;

<75> d) 해독 스트림이 생성되어 디코더의 변경 방식에 따라 메모리 내의 무작위한 기억 장소들에 위치된다;

<76> e) 이후, 디코더는 한번에 하나의 바이트 또는 하나의 워드를 읽고, 필요한 경우 이들을 해독한다.

<77> 코드 다양성은 실행 중에 클라이언트 프로그램 상의 모바일 가드에 의하여 수행되는 연산들을 포함한다. 수행된 연산들은 코드를 바꾸며 그 의미가 바뀌지 않으면서 서로 다른 명령들을 포함하게 된다. 이는 패턴 기반 식별을 방지하기 위한 것이다. 다음의 단계들 중 하나 이상이 수행될 수 있다.

<78> 컨텍스트-독립 명령들이 삽입될 수 있다. 컨텍스트-독립 명령들은 입력 컨텍스트는 프로그램 내의 컨텍스트들과 공유될 수 있으나, 출력 컨텍스트는 프로그램 내의 어느 입력 컨텍스트와도 상이한 명령이다. 컨텍스트-독립 명령들은 프로그램의 어떠한 입력 컨텍스트도 바꿀 수 없기 때문에, 문맥-독립 명령들이 무엇을 처리하는지는 중

요하지 않다.

<79> 컨텍스트 종속 명령들은 동일한 기능을 수행하는 명령들로 교체될 수 있다. 이는 달성하기 더욱 어려우나, 데이터 흐름 분석에 의하여 식별될 수 없기 때문에 더욱 효과적임을 이해할 수 있다.

<80> 생성될 수 있는 함수 독립 변화들은 명령들의 실행 순서 변경, 임시 변수들을 가지거나 가지지 않은 명령들의 삽입, 메모리 내 명령들의 재정렬 및 제어 흐름 변화들의 생성을 포함한다.

<81> 함수 종속 변화들은 기능 및 부작용을 원래대로 유지하기 위하여 주의를 요구한다. 함수 종속 변화들은 명령들을 기능적 균등물로 교체, 항등 함수들을 도입, 상수 값을 임의로 값을 초기화하고 최초 상수에 부합되는 값으로 정정하는 연산을 수행하는 명령들로의 교체를 포함한다. 또한, 목적지에 대한 복제들이 새로이 생성된 변수의 복제들로 교체될 수 있도록, 변수들이 도입될 수 있다.

<82> 예를 들어, TV 셋톱박스 같은 하드웨어 기반 뷰어 솔루션이 적용된 실시예에서, 뷰어의 분배자는 뷰어 소프트웨어뿐만 아니라 뷰어 환경, 즉 하드웨어 및 오퍼레이팅 시스템을 제어한다. 따라서, 하드웨어 기반 뷰어는 순수한 소프트웨어 기반 솔루션 보다 훨씬 더 완전한 방식으로 모바일 가드에 의하여 전반적으로 검사될 수 있다. 이 실시예에서, 모바일 가드의 체크섬 알고리즘은 뷰어 소프트웨어 검사에만 제한되지 않으며, 운영 시스템 및 하드웨어의 서로 다른 특성들을 검사할 수 있다.

<83> 따라서, 시스템은 하드웨어 기반 뷰어들에 관련해서 두 가지 방식으로 이용될 수 있다. 첫째, 시스템은 값비싼 변조 방지 하드웨어를 기반으로 하는 솔루션들을 대체할 수 있다. 둘째, 시스템은 변조 방지 하드웨어가 훼손된 경우 실행되기 시작하는 부가적인 보안 수단들을 제공할 수 있다.

<84> 본 발명은 바람직하게 문서의 개별적으로 암호화된 세그먼트들 및 "모바일 가드" 개념의 이용을 조합한 것이라 볼 수 있다. 따라서, 본 발명의 또 다른 측면에서 보면, 본 발명은,

<85> (a) 상기 작품의 시간적으로 분리된 각 세그먼트들에 대응되는 일련의 서로 다른 키들을 이용하여 상기 작품을 암호화하는 단계;

<86> (b) 보안 서버에서 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-알고리즘은 클라이언트 상태의 함수인 결과를 포함함-;

<87> (c) 클라이언트에서 코드를 실행하고, 보안 서버에 결과를 반환하는 단계;

<88> (d) 결과가 변경되지 않은 뷰어임을 나타내는지 여부를 결정하는 단계;

<89> (e) 서버에서 뷰어로 세그먼트를 전송하는 단계;

<90> (f) 결과가 변경되지 않은 뷰어임을 나타내는 경우, 전송된 세그먼트에 대응되는 키를 보안 원격 서버에서 뷰어로 보안 스트리밍하는 단계;

<91> (g) 키를 이용하여 세그먼트를 해독하는 단계를 포함하는 클라이언트로 미디어 작품을 전송하는 방법을 제공한다.

<92> 상기 단계들이 전술한 순서대로 실행될 수 있다 하더라도, 상기 단계들 중 적어도 일부는 다른 순서로 또는 동시에 실행될 수 있음을 이해할 수 있다. 예를 들어, 단계 (e)는 단계 (b), (c), (d) 또는 (f)와 동시에 수행되어, 세그먼트들이 키들과 함께 또는 키들의 전송 전후에 전송된다. 그러나, 키들은 세그먼트가 해독되기 전에 사용되어야 한다.

<93> 일 실시예에서, 상기 방법은 단계 (b) 내지 단계 (g)를 반복하는 단계 (d)를 더 포함한다.

<94> 그러나, 일반적으로, 전송된 소프트웨어 코드는 예를 들어 30초 이하인 일정한 "수명" 또는 "신뢰 간격"을 가진다. 한편, 일반적으로 세그먼트들은 소프트웨어 코드의 수명보다 더 자주, 예를 들어 초당 한번 전송된다. 이와 같이, 새로운 소프트웨어 코드는 세그먼트가 전송되는 각 시점마다 전송될 필요가 없으나, 현재의 소프트웨어 코드의 수명이 만료한 경우에는 새로운 소프트웨어 코드기 전송되어야 할 것이다. 따라서, 일반적으로 단계 (b)가 반복될 경우, 단계 (e) 내지 단계 (g)는 새로운 소프트웨어 코드가 필요할 때까지 반복될 것이다. 이 방식으로 한 피스의 소프트웨어 코드(모바일 가드)가 많은 키들의 전달을 보호한다.

<95> 코드의 실행 및 결과가 변경되지 않은 뷰어임을 나타내는지의 결정(단계 c 및 d)은 소프트웨어 코드의 각 피스에 대하여 한번 이상 수행될 수 있다 하더라도, 일반적으로, 이는 소프트웨어 코드의 수명 중 한번만 필요할 것

이다. 따라서, 일반적으로 단계 (b)가 반복된 후에만, 단계 (c) 및 단계 (d)가 반복될 것이다.

<96> 또 다른 측면에서 보면, 본 발명은,

<97> (a) 상기 작품의 시간적으로 분리된 각 세그먼트들에 대응되는 일련의 서로 다른 키들을 이용하여 상기 작품을 암호화하는 단계;

<98> (b) 보안 서버에서 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 단계-알고리즘은 클라이언트 상태의 함수인 결과를 포함함-;

<99> (c) 클라이언트에서 코드를 실행하고, 보안 서버에 결과를 반환하는 단계;

<100> (d) 결과가 변경되지 않은 뷰어임을 나타내는지 여부를 결정하는 단계를 포함하며,

<101> (e) 서버에서 뷰어로 세그먼트를 전송하는 단계;

<102> (f) 전송된 세그먼트에 대응되는 키를 보안 원격 서버에서 뷰어로 보안 스트리밍하는 단계;

<103> (g) 획득된 미디어 키를 이용하여 세그먼트를 해독하는 단계;

<104> (h) 단계 (d)가 변조된 뷰어를 표시하면, 다음 키들이 전송되는 것을 금지하고, 그렇지 않으면 단계 (e) 내지 단계 (g)를 반복하는 단계를 더 포함하는 클라이언트로 미디어 작품을 전송하는 방법을 포함한다.

<105> 바람직하게, 상기 방법은 단계 (b) 내지 단계 (d)를 반복하는 단계 (i)를 더 포함한다.

<106> 상기 단계들이 전술한 순서대로 실행될 수 있다 하더라도, 상기 단계들 중 적어도 일부는 다른 순서로 또는 동시에 실행될 수 있음을 이해할 수 있다. 사실, 일부 단계들은 다른 단계들보다 더 많은 회수로 수행될 수 있다.

<107> 단계 (b) 내지 (d)는 단계 (e) 내지 (h)와 독립적으로 수행될 수 있고, 바람직하게는 동시에 수행될 수 있다. 전술한 바와 같이, 소프트웨어 코드는 많은 세그먼트들 및 키들의 전송을 처리할 수 있는 수명을 일반적으로 가진다. 이와 같이, 단계 (b) 내지 (d)의 반복(단계 (i)에서 언급됨)은 일반적으로 단계 (e) 내지 (g) (단계 (h)에서 언급됨)의 반복보다는 덜 빈번하게 수행될 것이다. 바람직하게, 단계 (i)는 소프트웨어 코드의 수명이 만료된 경우에만 수행될 수 있다.

<108> 또한, 본 발명은 전술한 바와 같이 동작하는 장치로 확대되고, 스트리밍되는 미디어를 수신하는 클라이언트와 서버 장치의 조합 및 각각을 포함한다. 따라서, 또 다른 측면으로부터, 본 발명은

<109> (a) 미디어 작품을 상기 클라이언트로 전송하는 수단;

<110> (b) 보안 서버에서 클라이언트로 알고리즘을 포함하는 소프트웨어 코드를 전송하는 수단-알고리즘은 클라이언트 상태의 함수인 결과를 포함함-; 및

<111> (c) 결과를 수신하고, 결과가 변경되지 않은 클라이언트를 나타내는지를 결정하는 상기 보안 서버에 연관된 수단을 포함하는 클라이언트에게 미디어 작품을 배분하는 시스템을 제공할 수 있다.

<112> 또 다른 측면은 클라이언트, 예를 들어 영화 등의 작품을 재생하는 뷰어를 제공하며, 클라이언트는 작품을 수신하고 원격 소스로부터 알고리즘을 포함하는 소프트웨어 코드를 수신하고; 클라이언트 상에서 알고리즘을 실행하고; 및 상기 원격 소스에 상기 알고리즘의 결과를 반환하여, 상기 클라이언트의 무결성의 상기 원격 소스에게 입증하고, 상기 문서를 재생할 수 있도록 한다.

<113> 클라이언트는 작품을 해독하거나, 작품의 세그먼트를 원격 소스로부터 제공받은 키를 이용하여 해독함으로써 작품의 재생을 바람직하게는 가능하게 한다. 바람직하게 클라이언트는 일련의 키를 요청하고, 순서대로 키를 이용하여 작품의 연속되는 섹션을 해독하며, 이후 작품은 연속적인 제공으로 재생한다. 바람직하게, 전술한 바와 같이, 키들의 권한 설정은 소스에 무결성을 입증하는 클라이이언트에 종속적이다.

<114> 또한, 본 발명은 클라이언트와의 조합으로, 전술한 작품들을 전달하는 일 조합의 시스템으로 확장되며, 문서들이 클라이언트로 전달되며 뷰어가 소스에게 클라이언트의 무결성을 증명하는 경우에만 재생될 수 있다.

<115> 종래 소프트웨어 솔루션들과는 대조적으로, 본 발명은 프로그램 코드이든 미디어 문서이든, 사용자가 이용 가능한 데이터에 포함된 비밀들에 의존하지 않는다고 볼 수 있다. 본 발명은 복제 시도들의 초기 감지가 가능하고, 미디어 문서의 본질적인 부분이 복제되기 전에 컨텐츠 제공자가 대응할 수 있게 한다.

<116> 또한, 모바일 가드를 이용하여 시스템 무결성 검사하는 개념이 클라이언트에게 (정의된) 문서들을 전송하는

것 이상의 분야들에 응용된다는 것이 인정되었다. 본 발명은 입력 데이터의 연산을 수행하는 비제어 환경에서 구동되는 코드의 무결성 및 신뢰성을 검증하는데 일반적으로 이용될 수 있다. 본 발명은 탐지되지 않아도, 상대방이 데이터 처리 방식을 바꾸는 것을 방지하는데 이용될 수 있다. 따라서, 미디어 뷰어에 관련된 전술한 설명은 모든 클라이언트 프로그램에 적용될 수 있다. 응용 분야는 게임, 뱅킹, 오디오 등을 포함한다.

<117> 따라서, 또 다른 측면에서 보면, 본 발명은 보안 소스에서 클라이언트 프로그램을 구동하는 클라이언트 컴퓨터로 소프트웨어 코드(예컨데, 모바일 가드)를 전송하는 단계-소프트웨어 코드는 클라이언트 프로그램의 종속하는 결과를 가지는 알고리즘을 포함하는-, 소프트웨어 코드를 실행하고 소스로 그 결과를 반환하는 단계를 포함하되, 소스는 클라이언트 프로그램의 무결성을 결정할 수 있다. 또한, 본 발명은 상기 방법에 따라 운영되는 장치로 확장된다.

<118> 본 발명의 이러한 측면은 전술한 바람직한 특징, 특히 모바일 가드와 관련된 특징의 일부 또는 전부를 적용할 수 있다. 미디어 작품들과 관련한 레퍼런스들은 서버와 클라이언트 사이에서 전송되는 임시 페이로드 데이터에도 유사하게 적용된다. 따라서, 서비스 제공자는 동일한 방식으로 사용자의 클라이언트의 협업을 이끌어낼 수 있고, 협업이 중지되거나 변조가 탐지되면 다음 페이로드 데이터를 보류할 수 있다.

<119> 따라서, 서버와 통신하는 클라이언트는 통신이 진행 중인 상태에서 클라이언트의 무결성을 검사할 수 있음을 이해할 수 있다. 그러므로, 본 발명은 비제어 환경에서 동작하는 클라이언트를 신뢰할 수 있게 한다. 클라이언트의 무결성이 손상된 것으로 파악된 경우, 대응 동작을 수행할 수 있다. 예를 들어, 클라이언트와의 통신을 종료하고, (전술한 미디어 스트리밍 어플리케이션으로) 해독 키들의 권한 설정을 중지 및/또는 (예를 들어, 뱅킹 시스템에 부정한 공격이 의심되는 경우) 증거를 수집할 수 있다.

<120> 본 발명은, 비밀성 및 부정식성이 일반적으로 이슈들이 아니더라도, 정확한 실행은 이슈인, 분산 연산들과의 관계에 있어서 유용하다. 따라서, 모바일 가드는 클라이언트-소프트웨어 및, 필요한 경우 하드웨어 모두-를 의도적 또는 비의도적인 변경을 방지하는데 이용될 수 있다. 따라서, 분산 컴퓨팅 작업을 시작하는 경우는 연산을 수행하는 원격 노드에서 클라이언트들의 정상 운용을 검사하기 위해 모바일 가드를 이용할 수 있다.

<121> 온라임 게임의 경우, 클라이언트 프로그램들의 변경은 치팅을 가능하게 하고, 속임수를 제어하지 못한다면 고객 불만과 수입 손실을 야기한다. 관련된 데이터는 비밀이 아니고 (미디어 작품과 함께) 이를 기록하데 있어서 작은 포인트가 있으므로, 통상적으로 클라이언트 소프트웨어의 무결성을 검증하는 것만으로 충분하다. 게임이 클라이언트-서버 기반으로 운용되는 경우, 모바일 가드를 이미 전술한 바와 같이 적용시킬 수 있다. 사용자가 모바일 가드와의 협업을 허락하지 않는다면, 사용자는 전체적인 게임 상태에 대한 업데이트를 거부당할 수 있다.

<122> 홈 뱅킹의 경우, 모바일 가드는 기밀 데이터에 제3자가 접근할 수 없도록 하는데 이용될 수 있다. 일반 사용자가 클라이언트 프로그램을 수정하는데 보통 무관심한 동안, 중간자 공격의 희생자가 될 수 있다. 따라서, 뱅킹 서버는 홈 뱅킹 클라이언트의 무결성 및 신뢰성을 검증하기 위하여 모바일 가드를 이용할 수 있는데, 모바일 가드는 뱅킹 서버의 공개 키를 포함할 수 있다. 상기 공개키는 홈 뱅킹 클라이언트로부터 뱅킹 서버에 전송되는 모든 데이터를 암호화하는데 이용되고, 모바일 가드의 무결성이 보장되기 때문에 사용자는 그의 데이터가 기밀하게 유지된다고 확신할 수 있다.

<123> 도 2에 도시된 바와 같이, 클라이언트는 스트리밍 서버(11) 또는 선택적으로, 예를 들어, CD(12), 로컬 스토리지 미디어로부터 스트리밍되는 미디어(예를 들어, 영화)를 보는데 이용할 수 있는 뷰어(10)를 제공받는다. 시스템의 각 구성 요소들은 신뢰 환경(13) 밖에 위치한다. 신뢰 환경 내에는 암호화되지 않은 영화(14), 보호 영화(16)를 생성하기 위한 보호 툴(15), 및 보안 서버(17)가 위치한다.

<124> 도 1에서 예시한 종래 시스템에 따르면, 컨텐츠 소유자는 부호화된 미디어 문서를 클라이언트에게 전달하기 전에 부호화된 미디어 문서를 보호한다. 그러나, 단일 미디어 키를 이용하는 대신, 보호 툴(15)이 매우 많은(수천의) 미디어 키(20)들을 이용하여 영화를 암호화한다. 이러한 프로세스는 암호화되고 부호화된 미디어인 보호 영화(16)를 생성한다.

<125> 미디어 키(20)들은 적절한 시간에 배포될 수 있도록 분배되고, 미디어 자원의 제공 동안, 미디어 키(20)들은 후술하는 바와 같이 요청에 따라 시간 간격을 두고 한번에 하나씩 클라이언트에게 안전하게 스트리밍된다. 미디어 그 자체는 별도로 스트리밍된다. 각 키는 키들을 스트리밍하는데 필요한 자원이 매우 낮은 오버헤드를 생성하도록 하기 위해 몇 바이트(약 16바이트)만을 포함한다.

<126> 각 키는 영화의 약 1초 또는 최대 몇 초만을 해독하는데 이용될 수 있으므로 하나의 키만을 획득하는 것은 의미

가 없다.

<127> 본 발명의 제1 실시예에 따르면, 보호 영화는 경로 A, 스트리밍 서버(11), 및 미디어 스트림(18)을 통하여 데이터 스트림의 형태로 클라이언트로 전달된다. 또 다른 실시예에서는 실재 미디어, 예를 들어 CD 또는 DVD(12)가 이용된다.

<128> 뷰어(10)는 클라이언트의 호스트에서 실행되고, 미디어 스트림(18) (또는 다른 실시예에서는 CD/DVD로부터)을 통하여 스트리밍 서버(11)로부터 보호 영화(16)를 수신한다. 제공 프로세스 동안, 뷰어(10)는 보안 서버(17)와 통신하여, 보호 영화(16)를 해독하기 위하여 필요한 미디어 키(20)들을 다운로드 받는다.

<129> 또한, 뷰어(10)는 약 30초의 일정한 주기에 따라 모바일 가드(19)라 불리는 코드의 피스들을 다운로드 받는다. 이것들 각각은 보안 서버(17)에서 생성된 알고리즘 형태의 비밀 정보를 포함하고 있다. 그 알고리즘들의 실행이 스트리밍된 데이터(18)를 이용하기 위해서 필요하다. 각 모바일 가드(19)가 뷰어로 전송되면, 비밀 알고리즘에 의해 결정된 연산들을 수행하고, 그 결과를 보안 서버에 반환한다. 모바일 가드는 뷰어가 변조되지 않은 경우에만 연산의 결과가 정상이 되도록 설계되어 있다. 비밀 알고리즘의 결과는 뷰어의 무결성을 보안 서버에 입증하는 체크섬을 포함한다.

<130> 또한, 모바일 가드는 비밀 알고리즘과 기능적 및 공간적으로 연관된 다른 부가 알고리즘들을 포함할 수 있다. 이 방식으로, 부가 알고리즘을 실행하지 않으면 비밀 알고리즘이 구현되지 않으므로, 클라이언트의 컴퓨터/뷰어는 부가 알고리즘을 실행하도록 강제될 수 있다. 이 방식으로, 뷰어는 완전히 검사될 수 있다.

<131> 모바일 가드로부터 보안 서버(17)로 반환된 결과가 예상 결과와 일치하지 않으면, 보안 서버는 뷰어로의 미디어 키(20) 분배를 중지한다. 만일 뷰어(10)가 모바일 가드를 거부하거나, 정상 결과가 일정 시간 내에 도달하지 않은 경우에도 동일하게 발생한다. 키 교환 프로토콜은 이후 더 상세하게 설명하기로 한다.

<132> 비밀 알고리즘들은 검사된 데이터(즉, 뷰어 코드)의 변경을 탐지할 확률이 높은 체크섬 연산들을 기반으로 한다. 도 3에 도시된 바와 같이, 무작위로 생성된 체크섬 알고리즘(21)(모바일 가드에서 사용)은 무작위화된 입력 변경(22) 및 변경된 입력에 대하여 수행하는 공지된 체크섬 연산(23)의 두 단계로 구분된 체크섬 연산을 이용한다. 이러한 두 단계는 함께 무작위화된 그리고 비밀 체크섬 알고리즘을 생성한다.

<133> 입력 변경은 체크섬 연산(23)에 입력될 데이터를 다른 순서로 바꾸는 변조자의 랜덤 생성을 참조한다. 보안 서버에 의하여 모바일 가드가 생성되면, 랜덤 시퀀스가 결정된다. 체크섬 알고리즘(21)이 뷰어에 의하여 실행되면, 뷰어로부터의 입력 프로그램 코드는 동일한 크기의 n개의 블록으로 분할된다. 이후 입력 변경 단계(22)에서, 입력 프로그램 코드는 전술한 랜덤 시퀀스로 섞어진다. 이후, 이것의 결과는 체크섬 연산 단계(23)로 입력된다. 이것은 공지된 메시지 다이제스트 알고리즘(MD5)을 이용한다. 이후, 체크섬 연산이 실행되고, 그 결과가 보안 서버에 반환된다.

<134> 체크섬 알고리즘 그 자체는 공지된 것이라 하더라도, 그 결과는 n개의 블록들이 체크섬 알고리즘에 입력된 순서의 함수이다. 그 순서는 보안 서버에 알려지므로 보안 서버는 반환된 결과가 무결한 뷰어를 나타내는지를 결정할 수 있다.

<135> 모바일 가드는 변조로부터 그리고 내부 작업이 염탐으로부터 보호될 필요가 있다. 모바일 가드 보호의 첫 번째 측면은 뷰어를 검사할 필요가 있을 때마다 새로운 버전을 무작위로 생성하는 것이다. 두 번째로 뷰어 환경에서 사용중인 모바일 가드의 수명을 (30초 이하로) 짧게 한다. 모바일 가드에 대한 인간에 의한 공격들(즉, 자동화와 반대되는 지능적 공격)이 이론적으로 가능하다 하더라도, 상당한 시간을 소요할 것이다. 그러므로, 각 모바일 가드에 대한 몇 초의 시간이 만료된 경우, 인간이 지원하는 공격은 모바일 가드가 모든 공격을 완료하기 전에 만료되기 때문에 사실상 불가능해진다.

<136> 모바일 가드는 전술한 바와 같이, 자동화 공격을 막기 위하여 모호화된다.

<137> 모바일 가드는 여기서 '런타임 뷰어 모호화'라고 지칭되는, 구동중인 뷰어의 메모리 이미지를 무작위화한다. 뷰어의 코드 및 데이터 영역들은 교체되고, 스택은 스크램블된다. 이하에서 상세히 설명된다.

<138> 런타임 뷰어 모호화의 효과는 무작위화하고 이로 인해 해독되고 부호화된 스트림의 기억 장소들을 감추기 때문에 오직 지능형 공격만이 뷰어의 이미지가 구동되는 동안 수행될 수 있다는 것을 분명하게 한다.

<139> 메모리 억세스의 장소를 무작위화하기 위해, 모바일 가드는 뷰어 코드 및 데이터 영역의 구조를 변경한다. 코드 및 데이터 영역은 논리적 세그먼트로 분할된다. 세그먼트 보더들은 오피 코드들 내에 위치되지 않도록 주의

해야 한다.

<140> 새로 다운로드된 모바일 가드가 제어를 수신한 후 그리고 스트림의 해독을 시작하기 전에, 모바일 가드는 새로운 위치들로 세그먼트들을 재배치한다. 상기 프로세스는 다음을 분명하게 하는 코드 세그먼트의 변경-다이내믹 링커에 의하여 수행되는 재배치와 유사-을 포함한다.

<141> 1. 점프 및 브랜치 명령들은 위치를 재배치된 위치들로 제어를 전송한다.

<142> 2. 리드 및 라이트 명령은 재배치된 위치에서 데이터를 액세스한다.

<143> 세그먼트를 재배치한 후에, 모바일 가드는 다음 모바일 가드로 대체될 때까지 그 동작을 수행한다.

<144> 모바일 가드는 뷰어에서 일정 함수들의 엔트리 포인트들을 알 필요가 있다. 세그먼트들의 새로운 위치들은 보안 서버에 알려지고, 모바일 가드에 제공된다. 이 방식으로 클라이언트측에서 두 개의 모바일 가드간에 정보를 교환할 필요가 없다.

<145> 스택 스크램블링에 대해, 스택은 이전의 함수 호출에 대한 반환 주소를 포함한다. 이는 제어 흐름을 염탐하거나, 스택상의 반환 주소를 바꿈으로써 뷰어의 제어 흐름을 변경하기 위하여 이용될 수 있다. 상기 공격에서, 프로그램이 호출 함수로 되돌아가려고 하면, 가능한 적대 코드로 제어를 대신 전송할 수 있다.

<146> 스택을 공격으로부터 보호하기 위하여, 새로운 반환 주소가 스택에 부가되면 스택을 점진적으로 스크램블하는 방법이 이용된다. 검사된 코드는, 함수 호출 후에, 모바일 가드의 스크램블 함수로 제어를 전달하고, 호출하는 함수로 제어를 반환하기 전에 스택상의 새로운 반환 주소를 스크램블한다. 스택을 언스크램블하기 위하여, 모바일 가드 내의 대응되는 언스크램블 함수가 반환 주소를 사용하기 전에 호출된다.

<147> 스크램블 함수의 구현은 모바일 가드가 뷰어를 검사하기 위하여 필요한 경우 생성된다는 사실을 이용한다. 이는 고유한 스크램블 함수 및 언스크램블 함수가 각 모바일 가드에서 생성될 수 있도록 한다. 스크램블 함수는 기본적으로 보안 서버에서 생성되고 모바일 가드에 포함되는 랜덤 데이터 세트를 포함하는데, 랜덤 데이터 세트는 뷰어의 스택상의 반환 주소와 배타적 논리합(XOR)된다. 랜덤 데이터의 어느 부분을 이용하는지를 선택하기 위해, 간단한 수학적 함수가 랜덤 데이터 세트로의 인덱스를 계산하는데 적용된다.

<148> 따라서, 뷰어는 (전술한 바와 같이) 염탐에 의하여 결정되는 뷰어의 상태(다양한 컨텐츠 및 제어 흐름의 위치를 포함함)에 대해 모바일 가드에 의해 보호된다.

<149> 미디어 키들은 초 당 약 1개의 비율로 뷰어에게 전송된다. 이는 랜덤 데이터 생성기 및 클라이언트에게 알려지는 보안 서버의 공개키를 사용하는 키 교환 프로토콜을 이용하여 실행된다. 다음 미디어 키를 획득하려면, 뷰어(10)는 16바이트의 랜덤 데이터를 생성하고, 이들을 보안 서버(17)의 공개키로 암호화한다. 암호화된 데이터는 이후 키에 대한 요청 내에 포함되며, 이는 보안 서버에 전송된다.

<150> 보안 서버는 요청을 검사하고, 뷰어가 정상임을 모바일 가드가 표시하는 경우에만 요청을 승인하다. 모바일 가드가 모두 정상임을 표시하는 경우, 보안 서버는 랜덤 데이터를 추출하고, 요청된 키로 배타적 논리합하여, 뷰어에게 그 결과를 반환한다.

<151> 뷰어가 결과를 수신하면, 결과를 최초 키를 요청하면서 제공된 랜덤 데이터와 배타적 논리합 함으로써, 결과로부터 요청된 키를 추출한다.

<152> 이 프로토콜은 뷰어의 소스 코드에 비밀키가 숨겨져 있지 않는 경우에도, 암호화되고, 부호화된 미디어 스트림을 해독하는 방법을 제공한다. 키의 수명은 단 몇 초이며, 이는 하나 또는 몇 개의 비밀키를 추출한 경우에 보안 스트리밍 프로세스가 실패하는 것을 방지한다.

<153> 클라이언트에 의하여 수행되는 효율적으로 2개로 분리된 스레드가 있고, 이들은 도 4의 흐름도에 요약되어 있음을 이해할 것이다.

<154> 첫 번째 스레드는 검증이다. 클라이언트는 모바일 가드를 수신하며, 이후 클라이언트 프로그램을 검증한다. 검증이 확인되면, 모바일 가드가 만료할 때까지 다음의 신뢰 간격 동안 n개의 키가 수신될 수 있다. 새로운 모바일 가드로 이후 상기 스레드를 반복한다.

<155> 이와 병행하여 제공 스레드가 구동된다. 각 키에 대해, 미디어 스트림의 세그먼트가 수신되고, 해독되고, 제공된다.

<156> 도 5는 서버의 동작을 요약한 것이다. 키 요청을 수신하면, 모바일 가드가 활동하면 그리고 활동하여야만(즉,

여전히 모바일 가드의 신뢰 간격 이내면), 키를 전송한다. 모바일 가드가 만료되면, 새로운 모바일 가드가 클라이언트로 전송되고, 이것이 클라이언트를 검증하는데 이용된다. 결과가 비정상하면, 클라이언트는 변조된 것으로 취급되어, 이후 키 전송은 중지된다. 결과가 정상이라면, 이후 새로운 신뢰 간격이 시작되고, 신뢰 간격 동안 클라이언트로 키가 전송된다.

도면의 간단한 설명

<157> 본 발명의 실시예들을, 예시로서, 첨부된 도면을 참조하여 설명하고자 한다.

<158> 도 1은 전술한 종래의 미디어 스트리밍 시스템을 개략적으로 나타낸 도면이다.

<159> 도 2는 본 발명의 일 실시예를 개략적으로 도면이다.

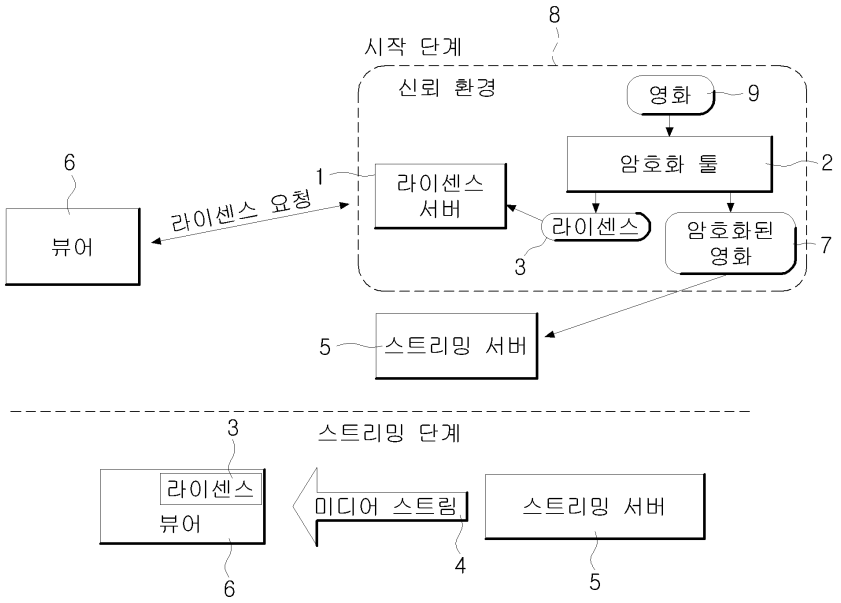
<160> 도 3은 일 실시예에서 사용되는 무작위로 생성된 체크섬 알고리즘의 컴포넌트들을 나타내는 개략적인 블록도이다.

<161> 도 4는 일 실시예의 동작을 나타내는 흐름도이다.

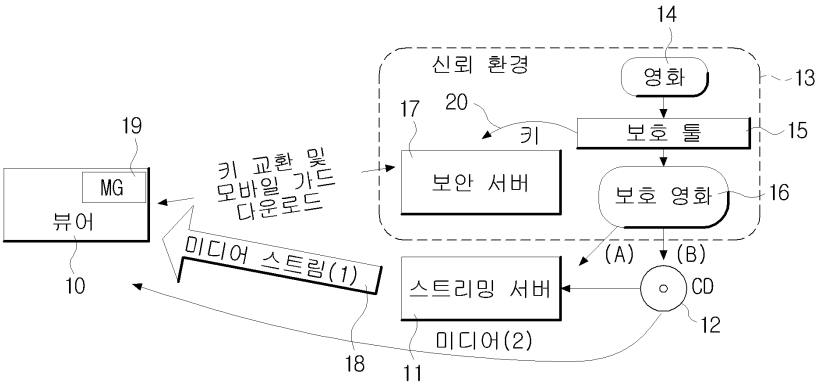
<162> 도 5는 일 실시예에 적용된 서버 알고리즘에 대한 흐름도이다.

도면

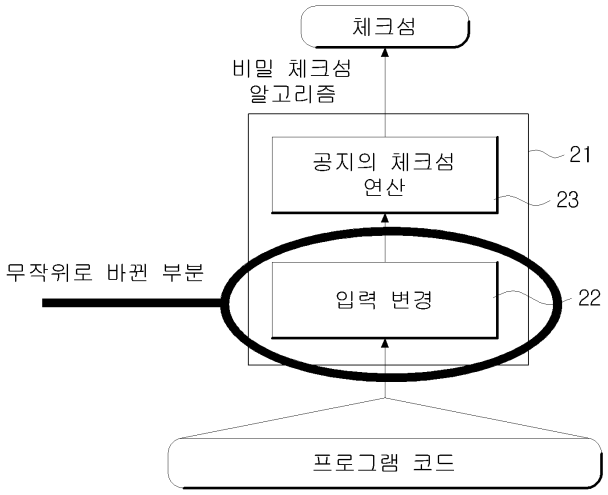
도면1



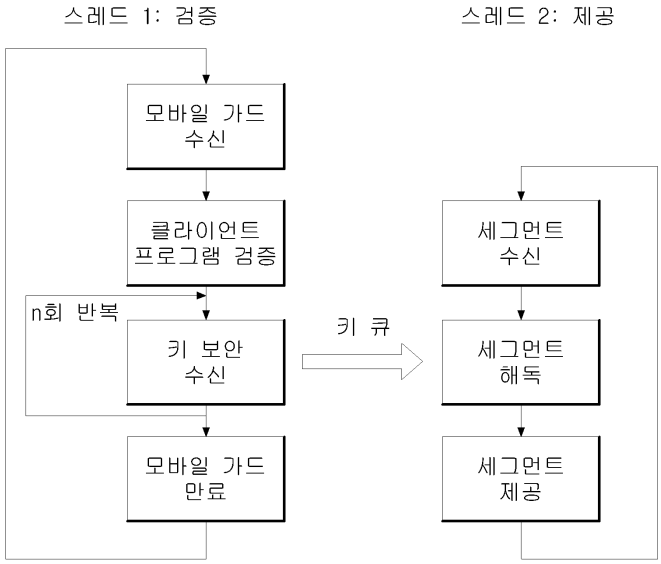
도면2


14
신뢰 환경
영화
13
17
20
키
보호 툴
15
19
MG
뷰어
키 교환 및
모바일 가드
다운로드
보안 서버
보호 영화
16
미디어 스트림(1)
(A)
(B)
스트리밍 서버
CD
12
미디어(2)
10
18
11

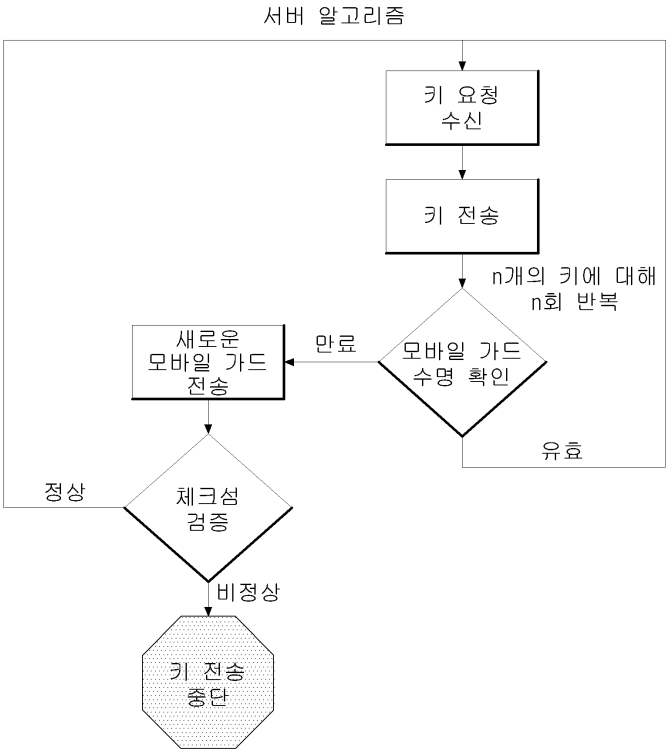
도면3


체크섬
비밀 체크섬
알고리즘
공지의 체크섬
연산
21
23
무작위로 바뀐 부분
입력 변경
22
프로그램 코드

도면4


스레드 1: 검증
스레드 2: 제공
모바일 가드 수신
클라이언트 프로그램 검증
n회 반복
키 보안 수신
모바일 가드 만료
키 큐
세그먼트 수신
세그먼트 해독
세그먼트 제공

도면5


서버 알고리즘
키 요청 수신
키 전송
n개의 키에 대해 n회 반복
모바일 가드 수명 확인
만료
새로운 모바일 가드 전송
유효
정상
체크섬 검증
비정상
키 전송 중단