(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년06월25일
(11) 등록번호 10-1159368
(24) 등록일자 2012년06월18일

(51) 국제특허분류(Int. Cl.)
G06F 15/16 (2006.01)
(21) 출원번호 10-2006-0010599
(22) 출원일자 2006년02월03일
심사청구일자 2011년01월27일
(65) 공개번호 10-2006-0103095
(43) 공개일자 2006년09월28일
(30) 우선권주장
11/090,681 2005년03월25일 미국(US)
(56) 선행기술조사문헌
US20050039034 A1

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
호드게스, 에우게네 더블유
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
크리스티에, 조셉
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이마이크로소프트 코포레이션 내
(74) 대리인
제일특허법인

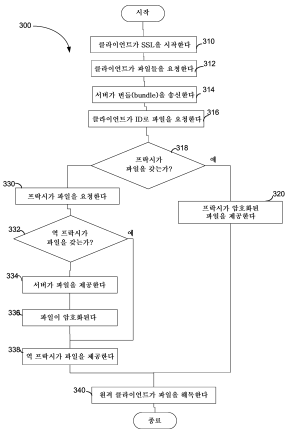
전체 청구항 수 : 총 20 항 심사관 : 이상헌

(54) 발명의 명칭 **분산 정보 관리를 위한 방법 및 장치**

(57) 요 약

소스 코드 관리 시스템을 위해 사용될 될 수 있는 것과 같은 정보 관리 시스템. 정보 관리 시스템은 기업 전반에 분산되고, 비보안 네트워크(unsecured network)를 통해 중앙 사이트로 접속된 원격 사이트들과 보안 통신을 하도록 하기 위해 동작한다. 보안 채널은 비보안 네트워크를 통해 생성되어 한 개 이상의 아티팩트들(artifacts)에 대한 암호화와 식별자 정보를 전송한다. 이 정보는 원격 사이트로부터의 아티팩트를 액세스하고 그것의 암호를 해독하기 위해 사용된다. 암호화되었지만, 아티팩트들이 식별가능한 형태를 유지하므로, 정보 관리 시스템을 통해 캐시(cache)될 수 있다. 아티팩트들을 캐시하는 능력은 원격 사이트들로부터 정보 관리 시스템을 동작시키기 위해 요구되는 대역폭을 많이 감소시킨다.

대 표 도 - 도3



특허청구의 범위

청구항 1

복수의 아티팩트(artifacts)로의 액세스를 갖는 서버와 프락시 서버(proxy server)로의 액세스를 갖는 클라이언트 간의 통신 방법으로서, 상기 프락시 서버는 연관된 캐시(cache)를 갖고, 상기 캐시는 아티팩트들의 복사본들을 저장하며, 상기 방법은,

a) 아티팩트에 대한 제1 요청을 상기 클라이언트로부터 상기 서버로 보안 연결을 통해 통신하는 단계;

b) 상기 제1 요청에 대한 응답으로, 상기 아티팩트에 대한 정보를 상기 서버로부터 보안 연결을 통해 수신하는 단계 - 상기 정보는 상기 아티팩트에 대한 암호화 정보(encryption information)를 포함함 - ;

c) 상기 정보를 사용하여 상기 아티팩트에 대한 제2 요청을 생성하는 단계 - 상기 제2 요청은 상기 서버로 향하는 통신이며, 상기 서버에 대한 식별자를 포함함 - ;

d) 상기 제2 요청을 상기 클라이언트로부터 비보안 연결을 통해 전송하는 단계; 및

e) 상기 아티팩트의 복사본이 상기 프락시 서버의 캐시에 저장되어 있는 경우, 상기 제2 요청에 대한 응답으로 상기 캐시로부터 상기 아티팩트의 복사본을 상기 비보안 연결을 통해 수신하는 단계

를 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 2

제1항에 있어서,

상기 아티팩트의 복사본이 상기 캐시에 저장되어 있지 않은 경우, 상기 아티팩트의 상기 복사본을 상기 서버로부터 상기 비보안 연결을 통해 수신하는 단계를 더 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 3

제1항에 있어서,

상기 아티팩트에 대한 정보를 수신하는 단계는, 상기 아티팩트에 대한 인코딩된 식별자를 수신하는 단계를 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 4

제1항에 있어서,

상기 캐시로부터 상기 아티팩트의 복사본을 수신하는 단계는, 상기 아티팩트의 암호화된 복사본을 수신하는 단계를 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 5

제4항에 있어서,

상기 암호화 정보를 사용하여 상기 아티팩트의 상기 암호화된 복사본을 해독하는 단계를 더 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 6

제1항에 있어서,

상기 보안 연결은 SSL 연결을 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 7

제4항에 있어서,

상기 서버는 제2 프락시 서버에 연결되고, 상기 제2 프락시 서버는 상기 아티팩트의 상기 암호화된 복사본을 저장하는 제2 캐시를 가지며,

상기 아티팩트의 암호화된 복사본을 상기 비보안 연결을 통해 수신하는 단계는, 상기 제2 캐시로부터 상기 아티팩트의 상기 암호화된 복사본을 수신하는 단계를 포함하는, 서버와 클라이언트 간의 통신 방법.

청구항 8

아티팩트에 액세스하도록 컴퓨팅 디바이스를 동작시키는 방법으로서, 상기 디바이스는 보안 채널 및 비보안 채널이 형성될 수 있는 적어도 하나의 네트워크에 연결되며, 상기 보안 채널은 제1 연결로서, 상기 제1 연결을 통해 전송될 입력 콘텐트(input content)에 대해 전송에 앞서 암호화 처리를 수행하며, 상기 비보안 채널은 제2 연결로서, 입력 콘텐트에 대해 암호화 처리를 수행하지 않고 상기 입력 콘텐트를 상기 제2 연결을 통해 전송하며, 상기 방법은,

a) 제1 아티팩트에 대한 제1 요청을 상기 보안 채널을 통해 통신하는 단계 - 상기 제1 요청은 상기 제1 아티팩트에 대한 제1 식별자를 포함함 - ;

b) 상기 제1 요청에 대한 응답으로, 상기 제1 아티팩트에 관한 정보를 상기 보안 채널을 통하여 수신하는 단계 - 상기 제1 아티팩트에 관한 상기 정보는 제2 아티팩트에 대한 제2 식별자 및 상기 제2 아티팩트를 암호화하는 암호화 키를 포함하고, 상기 제2 아티팩트는 상기 제1 아티팩트의 암호화된 복사본임 - ;

c) 상기 제2 아티팩트에 대한 상기 제2 식별자를 사용하여, 상기 제2 아티팩트의 복사본에 대한 제2 요청을 전송하는 단계 - 상기 제2 요청은 상기 비보안 채널을 통해 전송되며, 상기 제2 요청은 서버로 향하는 통신이고 상기 서버에 대한 식별자를 포함함 - ;

d) 상기 아티팩트의 암호화된 복사본이 프록시 서버의 캐시에 저장되어 있는 경우, 상기 제2 아티팩트의 복사본을 상기 캐시로부터 상기 비보안 채널을 통해 수신하는 단계 - 상기 제2 아티팩트의 암호화에 사용되는 상기 암호화 키는 상기 비보안 채널에 의해 제공되지 않음 - ; 및

e) 상기 암호화 키를 사용하여, 상기 제2 아티팩트의 상기 복사본을 암호화된 형태로부터 해독된 형태로 변환하는 단계

를 포함하는, 컴퓨팅 디바이스 동작 방법.

청구항 9

제8항에 있어서,

상기 제1 아티팩트는 소스 코드 파일인, 컴퓨팅 디바이스 동작 방법.

청구항 10

제9항에 있어서,

상기 제1 식별자는 버전 제어 시스템에 의해 유지되는 상기 소스 코드 파일의 버전을 식별하는 버전 식별자인, 컴퓨팅 디바이스 동작 방법.

청구항 11

제8항에 있어서,

상기 제1 아티팩트에 관한 상기 정보는 에러 감지 코드를 포함하는, 컴퓨팅 디바이스 동작 방법.

청구항 12

제8항에 있어서,

보안 채널을 사용하여 상기 제1 아티팩트에 관한 정보를 수신하는 단계는, 보안 채널을 사용하여 복수의 아티팩트에 관한 정보를 수신하는 단계를 포함하는, 컴퓨팅 디바이스 동작 방법.

청구항 13

제8항에 있어서,

상기 컴퓨팅 디바이스가 상기 제2 아티팩트의 복사본을 수신했음을 나타내는 통신을 상기 네트워크를 통해 송

신하는 단계를 더 포함하는, 컴퓨팅 디바이스 동작 방법.

청구항 14

제1 사이트의 장치, 제2 사이트의 장치 및 상기 제1 사이트의 장치와 상기 제2 사이트의 장치를 상호연결하는 네트워크를 갖는 유형의 소스 코드 제어 시스템으로서,

a) 상기 제1 사이트의 장치는,

i) 복수의 소스 코드 파일의 복수의 암호화되지 않은 버전을 저장하는 메모리 저장 디바이스;

ii) 서버로서,

A) 암호화된 아티팩트를 생성하기 위해 소스 코드 파일의 버전을 암호화하기 위한 컴퓨터 실행가능 명령들;

B) 상기 암호화된 아티팩트를 비보안 연결을 사용하여 상기 네트워크를 통해 통신하기 위한 컴퓨터 실행가능 명령들; 및

C) 상기 암호화된 아티팩트에 관한 암호화 정보를, 보안 연결을 사용하여, 상기 네트워크를 통해 통신하기 위한 컴퓨터 실행가능 명령들

을 저장하는 컴퓨터-판독가능 매체를 포함하는 서버; 및

iii) 상기 서버와 상기 네트워크 사이에 연결되는 제2 서버로서,

A) 상기 암호화된 아티팩트가 상기 서버로부터 상기 네트워크를 통해 통신될 때, 상기 서버로부터 상기 암호화된 아티팩트를 수신하기 위한 컴퓨터 실행가능 명령들;

B) 캐시에 상기 암호화된 아티팩트를 저장하기 위한 컴퓨터 실행가능 명령들; 및

C) 상기 캐시에 상기 암호화된 아티팩트를 저장한 이후에, 상기 제2 사이트로부터 수신되는 상기 암호화된 아티팩트에 대한 요청에 대한 응답으로, 상기 캐시로부터 상기 암호화된 아티팩트를 전송함으로써, 상기 요청을 상기 서버로 전달하지 않고 미래 요청에 응답하기 위한 컴퓨터 실행가능 명령들

을 포함하는 제2 서버;

를 포함하며,

b) 상기 제2 사이트의 장치는 클라이언트 컴퓨터 및 캐시를 포함하고, 상기 클라이언트 컴퓨터는,

i) 소스 코드 파일에 대한 제1 요청을 상기 보안 연결을 통해 전송하기 위한 컴퓨터 실행가능 명령들;

ii) 상기 암호화 정보를 상기 보안 연결을 통해 획득하기 위한 컴퓨터 실행가능 명령들 - 상기 암호화 정보는 상기 소스 코드 파일의 암호화된 복사본에 대한 식별자를 포함함 - ;

iii) 상기 소스 코드 파일의 암호화된 복사본에 대한 제2 요청을 전송하기 위한 컴퓨터 실행가능 명령들 - 상기 제2 요청은 상기 서버로 향하는 통신이며, 상기 서버에 대한 식별자를 포함함 - ;

iv) 상기 네트워크를 통하여, 상기 암호화된 아티팩트를 상기 비보안 연결을 통해 수신하기 위한 컴퓨터 실행가능 명령들; 및

v) 상기 소스 코드 파일의 버전을 생성하기 위하여, 상기 암호화 정보를 사용하여 상기 암호화된 아티팩트를 해독하기 위한 컴퓨터 실행가능 명령들

을 저장하는 컴퓨터-판독가능 매체를 포함하며,

상기 캐시는 제2의 복수의 소스 코드 파일의 복수의 암호화된 버전을 포함하고, 상기 제2의 복수의 소스 코드 파일은 적어도 상기 제1 사이트의 상기 메모리 저장 디바이스에 저장된 상기 복수의 소스 코드 파일의 부분 집합인, 소스 코드 제어 시스템.

청구항 15

제14항에 있어서,

상기 서버의 상기 컴퓨터-판독가능 매체는 상이한 암호화 키로 소스 코드 파일의 복수의 버전의 각각을 암호화하기 위한 컴퓨터 실행가능 명령들을 더 포함하는, 소스 코드 제어 시스템.

청구항 16

제14항에 있어서,

a) 상기 메모리 저장 디바이스는 복수의 소스 코드 파일의 상기 복수의 버전들의 각각을 제1 유형 식별자와 관련하여 저장하고,

b) 상기 서버의 상기 컴퓨터-판독가능 매체는 상기 암호화된 아티팩트에 관한 제2 유형 식별자를 상기 네트워크를 통해 통신하기 위한 컴퓨터 실행가능 명령들을 더 포함하며,

c) 상기 캐시는 상기 복수의 암호화된 소스 코드 파일을 상기 제2 유형 식별자와 관련하여 저장하는, 소스 코드 제어 시스템.

청구항 17

제14항에 있어서,

a) 상기 네트워크를 통해 페이지들을 다운로드하도록 구성되는 브라우저를 각각 구비하는 복수의 클라이언트 컴퓨터를 더 포함하고;

b) 상기 캐시는 상기 네트워크를 통해 다운로드된 상기 페이지들을 캐싱하기 위한 컴퓨터 실행가능 명령들을 추가로 저장하는, 소스 코드 제어 시스템.

청구항 18

제14항에 있어서,

상기 제2 사이트는 제2 클라이언트 컴퓨터를 더 포함하고,

상기 제2 클라이언트 컴퓨터는,

ⅰ) 제2 보안 연결 및 제2 비보안 연결을 설정하기 위한 컴퓨터 실행가능 명령들;

ii) 상기 암호화 정보를 상기 제2 보안 연결을 통해 획득하기 위한 컴퓨터 실행가능 명령들;

iii) 상기 네트워크를 통하여, 상기 암호화된 아티팩트를 상기 제2 비보안 연결을 통해 수신하기 위한 컴퓨터 실행가능 명령들; 및

iv) 상기 소스 코드 파일의 상기 버전을 생성하기 위하여, 상기 암호화 정보를 사용하여 상기 암호화된 아티팩트를 해독하기 위한 컴퓨터 실행가능 명령들

을 저장하는 제2 컴퓨터-판독가능 매체를 포함하는, 소스 코드 제어 시스템.

청구항 19

제1항에 있어서,

상기 보안 연결은 제1 연결로서, 전송에 앞서, 상기 제1 연결을 통해 전송될 입력 콘텐트 상에 암호화 처리를 수행하며,

상기 비보안 연결은 제2 연결로서, 입력 콘텐트 상에 암호화 처리를 수행하지 않고 상기 입력 콘텐트를 상기 제2 연결을 통해 전송하는, 서버와 클라이언트 간의 통신 방법.

청구항 20

제14항에 있어서,

상기 제2 사이트의 장치는 상기 제2 사이트의 장치에 의해 전송된 상기 소스 코드 파일에 대한 상기 제1 요청에 대한 응답으로, 상기 암호화 정보를 수신함으로써 상기 암호화 정보를 획득하도록 구성되는, 소스 코드 제어 시스템.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0015] 본 발명은 일반적으로 정보 관리 시스템에 관한 것으로서, 특히 분산 정보 관리 시스템에 관한 것이다.

[0016] 정보 관리 시스템은 널리 사용된다. 그런 시스템은 종종 "아티팩트들(artifacts)"을 저장하는 데이터베이스를 포함한다. 아티팩트는 정보 관리 시스템에 의해 조작되는 조직화된 형태의 데이터의 콜렉션이다. 아티팩트들은 종종 컴퓨터 파일들이다. 정보 관리 시스템의 한 일반 유형은, 기업에서 소프트웨어를 개발하는 컴퓨터 소스 코드의 파일들을 관리하기 위해 사용될 수 있는 것과 같은, 버전 제어 시스템(version control system)이다.

[0017] 정보 관리 시스템은 종종 네트워크화되어 있어서 기업 내의 복수 명의 사람들이 그 아티팩트들과 작업할 수 있다. 소스 코드 관리 시스템의 예에서, 프로그램 개발자, 프로젝트 관리자, 테스트 엔지니어, 및 기업 내의 다른 이들이 모두 그 소스 코드 파일들을 액세스할 수 있다. 파일들은 중앙 데이터베이스에 저장될 수 있다. 기업 전반에서 파일들의 사용을 용이하게 하기 위해, 정보 관리 시스템은 종종 데이터베이스에 링크된 서버를 포함한다. 서버는 네트워크에 접속되어, 정보 관리 시스템의 정보의 개별 사용자들을 위한 워크스테이션들이 데이터베이스로부터 아티팩트들의 복사본들을 검색하도록 한다. 사용자들이 아티팩트들을 요청하면, 그들의 워크스테이션들은 서버로 요청들을 송신하고, 그 다음, 서버는 네트워크를 통해 아티팩트들의 복사본들을 제공한다.

[0018] 정보 관리 시스템을 위한 네트워크 아키텍처는 기업이 비교적 넓은 지리적 지역에 분산된 복수 개의 작업 사이트들을 가질 때도 사용될 수 있다. 인터넷을 포함할 수 있는, 광역 네트워크(WAN)는 분산된 위치의 개별 워크스테이션들이 아티팩트들의 데이터베이스로의 액세스를 갖는 서버와 통신하도록 한다.

[0019] 클라이언트와 서버 간의 통신 경로의 어떤 부분이 비보안되는 경우, 네트워크가 비보안되더라도, 보안 채널이 네트워크를 통해 생성될 수 있다. 인터넷은 비보안된 통신 경로의 일 예이다. 비보안 네트워크를 통해 생성될 수 있는 보안 채널들의 예들로는 SSL(secure socket layer) 접속 또는 VPN(virtual private network)이 있다.

[0020] 보안 채널에서, 통신 프로토콜은 전송이 인터셉트되더라도, 의도되지 않은 수취인이 네트워크를 통해 전송되는 정보의 내용을 알아내는 것을 어렵게 한다. 예를 들어, SSL 채널을 통해 전송하는 디바이스는 정보가 전송될 때 정보를 암호화한다. 정보는 또한 단편(piece)들로 쪼개질 수 있어서 전송되는 정보를 쉽게 탐지할 수 있는 상관관계가 없도록 한다. 결과적으로, 의도되지 않은 수취인이 암호화 메커니즘을 "크랙(crack)"하는 노력을 더욱 어렵게 하여, 파일들이나 또는 다른 논리적으로 관련된 데이터의 블록들을 나타내는 전송의 부분들을 식별할 수도 없을 것이다.

[0021] 보안 채널 사용의 단점은 보안 채널을 통해 정보를 다운로드하기 위해 사용자에 의해 요구되는 시간의 양 및 클라이언트-서버 환경에 부가되는 부하이다. SSL은 각 사용자의 세션을 위해 특정한 비대칭 키 쌍을 채택한다. 특정한 대칭 키는 데이터가 채널에서 전송될 때 그 데이터의 암호화를 위해 사용된다. 서버의 정보가 복수 개의 클라이언트들에게 분산되는 경우에, SSL 채널들의 사용은 서버에 높은 부하를 부가한다. 기업에 의해 비보안 채널들로 상호접속되는 지리적으로 분산된 사이트들과 동작할 수 있는 정보 관리 시스템을 위한 개선된 방법 및 장치를 제공하는 것은 바람직할 것이다.

발명이 이루고자 하는 기술적 과제

[0022] 본 발명은 컴퓨팅 디바이스가 보안 채널을 통해 아티팩트에 대한 정보를 수신할 수 있는 정보 관리 시스템에 관한 것이다. 이 정보는 비보안 채널을 통해 아티팩트의 암호화된 버전을 검색하고 해독하기 위해 사용된다.

[0023] 하나의 양상에서, 본 발명은 복수 개의 아티팩트들을 액세스하는 서버와, 연관된 캐시를 갖는 -캐시는 아티팩트들의 복사본들을 저장함- 프락시 서버를 액세스하는 클라이언트 간의 통신 방법에 관한 것이다. 방법은 클라이언트에서 서버로의 제1 요청을 통신하는 단계; 그 요청에 응답하여, 서버에서 클라이언트로 아티팩트에 대한 인코딩된 정보를 통신하는 단계; 클라이언트에서 프락시 서버로, 그 인코딩된 정보를 사용하여 제2 요청

을 생성하는 단계; 및 제2 요청에 응답하여, 아티팩트의 복사본이 캐시에 저장되었을 때, 캐시로부터 아티팩트의 복사본을 제공하는 단계를 포함한다.

[0024] 다른 양상에서, 본 발명은 아티팩트를 액세스하기 위해 컴퓨팅 디바이스를 동작하는 방법에 관한 것이고, 보안과 비보안 채널을 통해 네트워크에 결합된 디바이스가 형성될 수 있다. 방법은, 아티팩트에 관한 정보를 수신하기 위해 보안 채널을 사용하는 단계; 아티팩트의 복사본을 요청하기 위한 비보안 채널을 사용하는 아티팩트에 관한 정보를 사용하는 단계; 암호화된 형태로 아티팩트를 수신하는 단계; 및 아티팩트에 관한 정보를 사용하여 암호화된 형태에서 복호화된 형태로 아티팩트를 변환하는 단계를 포함한다.

[0025] 또 다른 양상에서, 본 발명은 제1 사이트의 장치, 제2 사이트의 장치, 및 제1 사이트의 장치와 제2 사이트의 장치를 상호접속하는 네트워크를 갖는 유형의 소스 코드 제어 시스템에 관한 것이다. 제1 사이트의 장치는 복수 개의 소스 코드 파일들의 복수 개의 버전들을 저장하는 메모리 저장 디바이스, 및 암호화된 아티팩트를 생성하기 위해 소스 코드 파일의 한 개의 버전으로 암호화하고, 네트워크를 통해 암호화된 아티팩트를 통신하며, 및 네트워크를 통해 암호화된 아티팩트에 관한 네트워크 암호화 정보를 통신하기 위한 컴퓨터-실행가능 명령들을 저장하는 컴퓨터-판독가능 매체를 포함하는 서버를 포함한다. 제2 사이트의 장치는 클라이언트 컴퓨터를 포함하고, 클라이언트 컴퓨터는 암호화 정보를 얻고, 네트워크를 통해 암호화된 아티팩트를 수신하며, 및 암호화 정보를 사용하여 암호화된 아티팩트를 해독하여 소스 코드 파일의 버전을 생성하기 위한 컴퓨터-실행가능한 명령들을 저장하는 컴퓨터-판독가능 매체를 포함한다.

발명의 구성 및 작용

[0026] 첨부한 도면들은 스케일링하기 위한 의도로 도시되지는 않았다. 도면들에서, 다양한 도면들에서 도시되는 각각의 동일한 또는 거의 동일한 컴포넌트는 유사한 부호로 표현된다. 명료성을 위해, 모든 컴포넌트가 모든 도면에서 레이블링되지는 않을 것이다.

[0027] 개선된 정보 관리 시스템이 비보안 채널을 통한 아티팩트들의 보안 통신을 허용하여 제공된다. 아티팩트들은 암호화된 형태로 통신되고, 아티팩트들을 수신하는 워크스테이션에 로컬인 프락시 서버에 저장될 수 있다. 보안 채널은 각 아티팩트에 대한 비교적 소량의 정보를 전송하기 위해 사용된다. 그런 시스템은, 예를 들어, 인터넷과 같은 비보안 채널을 사용하여 중앙 사이트로 기업의 원격 사이트를 접속하기 위해 사용될 수 있다. 소스 코드 관리 시스템은 본 명세서에서 정보 관리 시스템의 일 예로서 사용된다.

[0028] 도 1은 본 발명의 실시예에 따른 정보 관리 시스템을 도시한다. 정보 관리 시스템은 중앙 사이트(110)와 원격 사이트(112)를 포함한다. 중앙 사이트(110)와 원격 사이트(112)는, 인터넷(114)일 수 있는, 네트워크를 통해 접속된다.

[0029] 중앙 사이트(110)는 데이터베이스(120)를 포함한다. 데이터베이스(120)는 컴퓨터-판독가능 및 컴퓨터-기록가능 저장 매체로부터 형성된다. 그것은 아티팩트들의 저장 및 검색을 조직화하는 제어기들을 포함한다. 기재된 실시예에서, 각 아티팩트는 중앙 사이트(110)와 원격 사이트(112)를 동작하는 기업에 의해 수행되는 개발 프로젝트의 일부인 소스 코드를 포함하는 파일이다. 이 예에서, 데이터베이스(120)의 각 파일은 파일 이름으로 기재되고, 각 파일의 복수 개의 버전들은 데이터베이스(120)에 저장될 수 있다. 데이터베이스(120)는 이 분야에서 알려진 것처럼 데이터베이스일 수 있지만, 데이터베이스의 어떤 적절한 형태가 사용될 수 있다.

[0030] 중앙 사이트(110)는 복수 개의 클라이언트 워크스테이션들(126_1, 126_2,...126_4)을 포함한다. 사용에서, 각 클라이언트 워크스테이션은 데이터베이스(120)로부터의 아티팩트들을 프로세스하기 위해 중앙 사이트(110)와 원격 사이트(112)를 관리하는 기업 내의 사람에 의해 사용될 수 있다. 각 워크스테이션은, 예를 들어, 데이터베이스(120)에 저장되는 소스 코드 파일들을 생성하는 코드 개발자에 의해 사용될 수 있다. 다른 경우에, 클라이언트 워크스테이션은 데이터베이스(120)로부터 소스 코드 파일을 검색하고 그것을 테스트하는 코드 테스터에 의해 사용될 수 있다. 각 클라이언트 워크스테이션은, 예를 들어, 개인용 컴퓨터 또는 유사한 컴퓨팅 디바이스일 것이다.

[0031] 중앙 사이트(110)는 서버(122)를 포함한다. 서버(122)는 데이터베이스(120)에 접속된다. 서버(122)는 WAN(124)을 통해 클라이언트 워크스테이션들(126_1, 126_2, 126_3, 및 126_4)의 각각에 액세스할 수 있다. 서버(122)는 데이터베이스(120)에 저장된 아티팩트가 클라이언트 워크스테이션으로 전달되는 것을 요청하는 클라이언트 워크스테이션들(126_1, 126_2,...126_4)로부터의 통신을 수신하는 하드웨어와 소프트웨어 엘리먼트들의 조합일 수 있다. 서버(122)는 그런 요청들을 수신하고 데이터베이스(120)를 액세스하여 워크스테이션으로 요청

된 아티팩트의 복사본을 제공하는 하드웨어와 소프트웨어 엘리먼트들을 포함한다. 서버(122)는 이 분야에서 알려진 것처럼 HTTP 메시지들을 사용하여 통신하는 파일 서버일 수 있지만, 어떤 적절한 구현이 사용될 수 있다.

[0032] 원격 사이트(112)는 한 개 이상의 원격 클라이언트 워크스테이션들(156)을 포함한다. 본 명세서에서, 한 개의 원격 클라이언트 워크스테이션(156)이 단순성을 위해 보여지지만, 본 발명은 복수 개의 클라이언트 워크스테이션들이 원격 사이트들로부터 데이터베이스(120)의 아티팩트들을 액세스할 때 가장 유용할 것이다. 원격 클라이언트 워크스테이션(156)은 클라이언트 워크스테이션들(126_1,...126_4)과 동일한 유형의 워크스테이션일 것이고, 동일한 목적을 위해 사용될 것이다. 그러므로, 원격 클라이언트 워크스테이션(156)은 클라이언트 워크스테이션들(126_1... 126_4)과 유사한 데이터베이스(120)에 저장된 아티팩트들에 액세스를 할 수 있어야 한다. 그러나, 원격 클라이언트 워크스테이션(156)과 데이터베이스(120) 간의 정보 흐름 경로는, 비보안 네트워크인, 인터넷(114)을 통해 전달된다.

[0033] 보안 채널은 이 분야에서 알려진 것처럼 원격 클라이언트 워크스테이션(156)과 서버(122) 간에 형성될 수 있다. 그러나 보안 채널은 원격 클라이언트 워크스테이션(156)으로 데이터베이스(120)의 아티팩트들을 직접 전송하기 위해 사용될 필요가 없다. 데이터베이스(120)에 저장된 아티팩트들로의 더 신속한 액세스를 하도록 하기 위해, 원격 클라이언트 워크스테이션(156)과 중앙 사이트(110)의 서버 간의 보안 채널은 비교적 적은 양의 정보를 전송하기 위해 사용된다. 이 정보는 인터넷(114)을 통해 비보안 채널에서 전송되는 암호화된 아티팩트를 액세스하고 사용하도록 하기 위해 사용된다. 아티팩트는 네트워크를 통해 정보를 전송하기 위해 사용되는 프로토콜의 외부에서 암호화될 수 있고, 이것은 더욱 효과적인 아티팩트들의 전송을 산출할 수 있다.

[0034] 비보안된 채널을 통해 아티팩트들의 전송의 효율성이 증가하는 한 방식은 암호화된 아티팩트들이, 비보안된 위치들에서도, 캐시될 수 있다는 것이다. 어떤 적절한 하드웨어와 소프트웨어가 아티팩트들을 캐시하기 위해 사용될 수 있다. 도시된 실시예에서, 원격 사이트(112)는, 아티팩트들을 캐시할 수 있는 디바이스의 일 예인, 프락시 서버(150)를 포함한다. 프락시 서버(150)는 이 분야에서 알려진 것과 같은 프락시 서버일 것이다. 프락시 서버(150)는 원격 클라이언트(156)와 인터넷(114) 간에 접속된다. 원격 클라이언트 워크스테이션(156)이 인터넷(114)을 통해, 파일들이나 또는 웹 페이지들과 같은, 아티팩트들이 다운로드되도록 요청하면, 프락시 서버(150)는 프락시 서버(150)와 연관된 컴퓨터-판독가능 및 컴퓨터-기록가능 메모리의 아티팩트들의 복사본들을 수신하여 저장할 수 있다. 저장된 정보는 아티팩트들의 캐시를 형성한다. 아티팩트들에 대한 후속적 요청들은 캐시로부터 수행될 수 있고, 네트워크를 통한 소통량을 감소시킨다.

[0035] 원격 클라이언트 워크스테이션(156)은 정보에 대한 추가 요청들을 생성하고, 이들 요청들은 프락시 서버(150)에게 먼저 전달될 것이다. 프락시 서버(150)가 그것의 캐시에 요청된 아티팩트를 저장했다면, 프락시 서버(150)는 그것의 캐시로부터 원격 클라이언트 워크스테이션(156)에게 그 아티팩트를 제공한다. 그 다음, 그 요청은 인터넷(114)으로 전송되지 않는다. 프락시 서버(150)와 연관된 캐시로부터 정보를 제공하는 것은 원격 클라이언트 워크스테이션(156)으로 아티팩트들이 제공될 수 있는 속도를 증가시킬 수 있다. 한 개의 원격 클라이언트 워크스테이션 또는 여러 개의 상이한 원격 클라이언트 워크스테이션들에 의해 액세스가 되는지에 무관하게, 속도 증가는 동일한 아티팩트가 빈번하게 액세스될 때 가장 크다. 소스 코드 관리 시스템에서, 현재 개발중인 소스 코드의 부분들을 포함하고 있는 아티팩트들은 종종 빈번하게 액세스된다.

[0036] 종래에, 인터넷(114)은 보안 채널을 제공하는 것으로서 간주되지 않았다. 프락시 서버들이 보통 비보안 채널을 통해 얻어진 정보를 저장하므로, 프락시 서버들은 종종 보안되지 않는다. 원격 클라이언트 워크스테이션(156)과 서버(122) 간의 보안 통신을 제공하기 위해, 전송되는 아티팩트들은 암호화된다. 프락시 서버(150)가 보안되지 않는 경우에, 아티팩트들은 암호화된 형태로 프락시 서버(150)에 캐시될 수 있다.

[0037] 추가적 효율성들은 중앙 사이트(110)에서 암호화된 아티팩트들을 캐시하여 얻어질 수 있다. 도시된 실시예에서, 중앙 사이트(110)는 또한 암호화된 아티팩트들의 복사본들을 저장하는 역 프락시 서버(reverse proxy server)(140)를 포함한다. 역 프락시 서버(140)는 아티팩트들을 캐시하기 위해 사용될 수 있는 디바이스의 일 예이다. 역 프락시 서버(140)는 이 분야에서 공지된 것일 수 있는 정책들에 따라 동작하는 캐시를 포함하는, 이 분야에서 알려진 프락시 서버일 수 있다. 역 프락시 서버(140)는 아티팩트들을 암호화할 수 있거나, 또는 암호화된 형태로 아티팩트들을 수신할 수 있다.

[0038] 도시된 실시예에서, 아티팩트들에 대한 요청들이 인터넷(114)을 통해 전송됨에 따라, 그들은 역 프락시 서버(140)에 도달한다. 역 프락시 서버(140)가 그것의 캐시에 요청된 아티팩트의 암호화된 버전을 저장하면, 그것은 암호화된 아티팩트의 복사본을 제공할 수 있다. 역 프락시 서버(140)가 그것의 캐시에 아티팩트의 암호

화된 복사본을 이미 저장하지 않았을 때, 그것은 서버(122)로부터 아티팩트를 요청할 수 있다. 그 다음, 서버(122)는 WAN(124)을 통해 암호화된 아티팩트를 제공할 수 있다. 그 다음, 역 프락시 서버(140)는 그것의 캐시에 암호화된 아티팩트를 저장할 수 있고, 인터넷(114)을 통해 암호화된 아티팩트를 전송할 수 있다.

[0039] 도 2a는 도 1에 도시된 정보 관리 시스템의 엘리먼트들 간의 통신의 시퀀스를 도시한다. 교환은 서버(122)에 보안 채널을 수립하는 원격 클라이언트 워크스테이션(156)에서 시작한다. 도 2a의 실시예에서, 통신은 보안 채널(210)에서 시작된다. 현재 알려졌거나 또는 이후에 개발되는지에 무관하게, 보안 채널(210)은 종래 보안 프로토콜들을 사용하여 형성될 수 있다. 기재된 실시예에서, 보안 채널(210)은 SSL 프로토콜을 사용하여 생성된다. 요청(212)이 서버(122)로 보안 채널을 통해 전송되므로, 프락시 서버(150) 또는 역 프락시 서버(140)가 요청의 내용으로 액세스를 하지 않는다. 이 실시예에서, 요청(212)은 서버(122)로 직접 전달된다.

[0040] 요청(112)은 원격 클라이언트 워크스테이션(156)에 제공되는 한 개 이상의 아티팩트들을 식별한다. 이 실시예에서, 각 아티팩트는 그것의 파일 이름에 의해 식별된다. 데이터베이스(120)가 버전 제어 시스템의 일부로서 파일들을 저장하는 경우에, 파일 이름은 파일의 특정 버전을 식별할 수 있다. 중앙 사이트(110)의 서버(122)는 요청(212)에 번들(bundle)(214)로 응답한다.

[0041] 번들(214)은 또한 보안 채널(210)을 통해 전송된다. 번들(214)은 정보를 제공하여 원격 클라이언트 워크스테이션(156)이 요청된 아티팩트를 얻어서 사용하도록 한다. 이 실시예에서, 번들(214)은 아티팩트의 암호화된 버전을 위한 식별자를 포함한다. 번들(214)은 아티팩트의 암호화된 버전을 해독하기 위해 사용될 수 있는 암호화 키를 포함한다. 또한, 번들(214)은 요청되는 아티팩트를 위해 준비되는, 해쉬 코드와 같은, 오류 탐지 코드를 포함할 수 있다.

[0042] 이 실시예에서, 식별자는 아티팩트에 할당된 코드이다. 데이터베이스(120)의 각 아티팩트는 서버(122)에 의해 할당되는 특정한 식별자를 가진다. 식별자는 아티팩트의 기능이나 또는 구조에 대한 정보를 드러내지 않는 것이 선호된다. 이에 비하여, 파일 이름들은 종종 아티팩트의 기능을 설명하도록 선택된다. 기재된 실시예에서, 비보안 채널들을 통해 전송된 통신에서 아티팩트를 참조하기 위한 식별자가 사용된다. 파일 이름 대신에 비설명적 식별자를 사용하는 것은 보안을 강화할 것이다. 비보안 채널을 통한 전송의 비승인된 수취인들은 아티팩트의 암호화를 "크랙"하기 위해 사용될 수 있는 감소된 정보를 수신한다. 각 식별자는 어떤 적절한 방식으로 할당될 수 있다. 예를 들어, 식별자들은 랜덤하게 할당될 수 있거나, 또는 아티팩트들이 데이터베이스(120)에 추가되는 순서로 할당될 수 있다. 데이터베이스(120)가 파일들의 복수 개의 버전들을 저장하는 경우에, 각 버전은 그것 자신의 식별자를 가질 것이다.

[0043] 번들(214)과 연관된 암호화 키는 아티팩트의 암호화된 버전을 해독하기 위해 사용될 수 있는 키이다. 다수의 암호화의 형태들이 알려졌고, 어떤 적절한 암호화의 형태가 사용될 수 있다. 기재된 실시예에서, 대칭적 암호화 알고리즘은 아티팩트를 암호화하기 위해 사용되는 키가 그 아티팩트를 해독하기 위해 사용되는 동일한 키가 되도록 사용된다. 기재된 실시예에서, 적어도 64 비트의 키들을 갖는 암호화 알고리즘이 사용된다. 적절한 알고리즘들의 예들로는 AES 128과 AES 256 암호화 알고리즘들이 있다. 각 아티팩트는 특정한 암호화 키를 가질 수 있다. 데이터베이스(120)가 파일들의 복수 버전들을 저장하는 경우에, 각 버전은 그 자신의 암호화 키를 가질 수 있다.

[0044] 번들(214)과 연관된 오류 검사 코드는 보안의 추가 방책을 제공한다. 오류검사 코드는 중앙 사이트(110)에서 아티팩트에 동작을 수행하여 생성된다. 오류 검사 코드에 대해 생성된 값은 아티팩트를 나타내는 파일의 내용들에 따른다. 원격 클라이언트 워크스테이션(156)은 그것이 수신하는 아티팩트에 동일한 동작을 수행할 수 있다. 번들(214)과 연관된 오류 검사 코드가 원격 클라이언트 워크스테이션(156)에 의해 생성되는 오류 검사 코드와 매치하지 않으면, 원격 클라이언트 워크스테이션(156)은 파일이 탬퍼링(tampering)의 결과로서 손상되거나 또는 변경되었음을 식별할 수 있다. 기재된 실시예에서, 오류 검사 코드는 해싱(hashing) 알고리즘을 통해 생성된다. 사용될 수 있는 해쉬 알고리즘의 일 예는 SHA1 해싱 알고리즘이지만, 오류 검사 코드를 생성하는 어떤 적절한 방법이 사용될 수 있다.

[0045] 번들(214)이 원격 클라이언트 워크스테이션(156)에서 수신될 시에, 원격 클라이언트 워크스테이션(156)은 암호화된 아티팩트의 복사본을 위한 요청(216)을 생성할 수 있다. 일 실시예에서, 번들(214)의 일부로서 제공되는 식별자는 아티팩트를 위한 페이지 주소로서 역할을 한다. 원격 사이트(112)가 인터넷(114)을 통해 중앙 사이트(110)에 접속되는 경우의 예에서, 통신은 인터넷을 통해 종래에 사용되는 것과 같은 HTTP 메시지들의 형태일 수 있다. 요청(216)은 HTTP GET 요청일 수 있다. 이 예에서, 번들(214)에 전송되는 식별자는 아티팩트를 포함하는 파일에 대한 URL의 일부일 수 있다. 따라서, 요청(216)은 HTTP://server/identifier의 형태일

수 있다. HTTP://server로서 표현되는 URL의 부분은 서버(122)를 위한 웹 주소를 식별한다. "식별자"로서 식별되는 URL의 부분은, 데이터베이스(120)에 저장되는 파일과 같은, 서버(122)를 액세스할 수 있는 특정 파일을 나타낸다.

[0046] 도 1에 도시된 정보 관리 시스템이 동작을 시작할 때, 프락시 서버(150)는 그것의 캐시에 아무런 아티팩트를 포함하지 않는다. 이 시나리오는 도 2a에 도시된다. 따라서, 요청(216)은 요청(218)으로서 프락시 서버(150)를 통과한다.

[0047] 요청(218)은 인터넷(114)을 통해 역 프락시 서버(140)로 전달된다. 도 1의 정보 관리 서버가 동작을 시작함에 따라, 역 프락시 서버(140)는 또한 아티팩트들에 관한 아무런 정보도 포함하지 않는다. 요청(218)은 요청(220)으로서 역 프락시 서버(140)를 통과한다.

[0048] 요청(220)은 WAN(124)을 통해 서버(122)로 전달된다. 서버(122)는 원격 클라이언트 워크스테이션(156)에 의해 전송되는 요청의 식별자를 사용하여 데이터베이스(120)의 특정 아티팩트를 식별한다. 서버(122)는 데이터베이스(120)로부터 아티팩트를 검색한다. 아티팩트는 암호화 알고리즘을 실행하기 위해 프로그램된 어떤 적절한 하드웨어에서 암호화될 수 있다. 이 시나리오에서, 서버(122)는 번들(214)로 전송되는 키를 사용하여 아티팩트를 암호화한다. 그러나, 암호화가 서버(122)에 의해 수행될 필요가 있는 것은 아니다. 다른 경우의 일 예로서, 데이터베이스(120)는 각 아티팩트의 암호화된 버전을 저장할 수 있고, 서버(122)는 단순히 요청된 아티팩트의 암호화된 버전을 검색할 수 있다.

[0049] 어떻게 아티팩트가 저장되거나 또는 암호화되는지에 무관하게, 서버(122)는 원격 클라이언트 워크스테이션(156)에 의해 시작되는 GET 요청에의 응답(222)을 준비한다. 원격 클라이언트 워크스테이션(156)과 서버(122) 간의 통신이 인터넷(114)을 통해 HTTP 프로토콜을 사용하고 있을 경우의 예에서, 암호화된 아티팩트는 HTTP 프로토콜에 의해 규정된 형식에 따라 응답(222)에 포함된다.

[0050] 응답(222)은 역 프락시 서버(140)로 먼저 전달한다. 역 프락시 서버(140)는 그것의 정책들에 따라 응답(222)에 포함된 암호화된 아티팩트의 복사본을 캐시할 수 있다. 암호화된 아티팩트는 서버(122)로부터 아티팩트를 요청하기 위해 사용되는 URL에 의해 인덱스되는 역 프락시 서버와 연관된 캐시에 저장될 수 있다. 동일한 아티팩트를 위한 어떤 후속 요청은 get 요청에 동일한 URL을 사용할 것이다. 따라서, 암호화되지 않은 아티팩트가 역 프락시 서버(140)에 캐시되는 동안, 역 프락시 서버(140)는 그 아티팩트에 대한 후속적 요청들을 식별하여 응답할 수 있다.

[0051] 암호화된 아티팩트를 포함하는 응답(224)은 인터넷(114)을 통해 역 프락시 서버(140)에서 프락시(150)로 송신된다. 프락시(150)는 또한 그것의 정책들에 따라 그것의 캐시에 암호화된 아티팩트를 저장할 수 있다. 암호화된 아티팩트는 또한 아티팩트를 요청하기 위해 사용되는 URL에 의해 또한 인덱스되는 프락시 서버(150)와 연관된 캐시에 저장될 수 있다. 원격 클라이언트 워크스테이션(156)이 동일한 아티팩트의 복사본을 프락시 서버(150)와 연관된 캐시에 저장되는 동안에 후속적으로 요청하면, 프락시 서버(150)는 그것의 캐시로부터 아티팩트의 암호화된 복사본을 제공하여 응답할 수 있다.

[0052] 암호화된 아티팩트는 프락시 서버(150)에서 원격 클라이언트 워크스테이션(156)으로의 응답(226)에 송신된다. 원격 클라이언트 워크스테이션(156)은 번들(214)에 포함되는 암호화 키를 사용하는 복호화 소프트웨어로 프로그램될 수 있다. 그러므로, 응답(226)에 포함되는 암호화된 아티팩트의 복사본은 원격 클라이언트 워크스테이션(156)에서 해독될 수 있다. 그 다음, 번들(214)에 전송되는 오류 검사 코드는 아티팩트의 해독된 코드에 적용되어 그 아티팩트의 적절한 전송을 확인할 수 있다.

[0053] 원격 클라이언트 워크스테이션(156)에서 실행하는 소프트웨어 프로그램들이 원격 클라이언트 워크스테이션(156)이 요청된 아티팩트의 유효한 복사본을 수신했다고 결정하면, 수신확인이 송신될 수 있다. 이 실시예에서, 수신확인(230)은 보안 채널(228)을 통해 전송된다. 그러나, 수신확인을 통신하기 위한 어떤 적절한 수단이 사용될 수 있다.

[0054] 수신확인(230)은 서버(122)에 의해 사용되어 원격 클라이언트 워크스테이션(156)으로 통신되는 아티팩트들의 수를 감소시킬 수 있다. 예를 들어, 원격 클라이언트 워크스테이션(156)이 관련된 파일들의 그룹을 요청하면, 이전 수신확인들로부터의 정보가 서버(122)에 의해 사용되어 원격 클라이언트 워크스테이션(156)에게 이미 그 그룹의 파일들의 부분 집합이 제공되었다고 결정할 수 있다. 따라서, 서버(122)는 파일들의 그룹에 대한 요청에의 적절한 응답이 그룹에서 모든 파일들보다 적은 수를 필요로한다고 결정할 수 있다. 그러나, 수신확인(230)은 모든 실시예들에 포함되지는 않을 것이다.

[0055] 도 2b는 발생할 수 있는 다른 정보의 교환을 도시한다. 도 2a에 도시된 정보의 교환과 같이, 인터액션(interaction)은 원격 클라이언트 워크스테이션(156)에 의해 송신되는 요청(252)에서 시작한다. 요청(252)은 보안 채널(250)을 통해 전송된다. 서버(122)는 요청(252)에 번들(254)으로 응답한다. 번들(254)은 요청(252)에서 식별되는 파일 또는 파일들에 대한 식별자를 포함할 수 있다. 번들(254)은 또한 각 요청된 파일에 대해 연관된 암호화 키와 오류 검사 코드를 포함할 수 있다. 이 정보는 보안 채널(250)을 통해 원격 클라이언트 워크스테이션(156)으로 리턴된다.

[0056] 원격 클라이언트 워크스테이션(156)은 번들(254)에 포함된 식별자들을 사용하여 비보안 채널을 통해 아티팩트들에 대한 요청(256)을 발생시킨다. 요청(256)은 프락시 서버(150)로 전달된다. 도 2b에 의해 도시되는 교환에서, 프락시 서버(150)는 요청된 아티팩트가 캐시되도록 한다. 아티팩트의 복사본은 원격 클라이언트 워크스테이션(156) 또는 프락시 서버(150)를 통해 접속된 원격 사이트(112)의 어떤 다른 워크스테이션과의 이전 인터액션의 결과로서 캐시될 수 있다.

[0057] 프락시 서버(150)는 서버(122)로 요청을 하지 않고 암호화된 형태로 요청되는 아티팩트를 제공한다. 프락시 서버(150)는 암호화된 형태로 아티팩트의 복사본을 포함하는 응답(258)을 생성한다.

[0058] 도 2a와 연결하여 위에 논의된 것처럼, 원격 클라이언트 워크스테이션(156)은 번들(254)에 포함된 암호화 키를 사용하여 아티팩트를 해독한다. 그 다음, 원격 클라이언트 워크스테이션(156)은 번들(245)에 포함된 오류 검사 코드를 적용하여 그것이 요청된 아티팩트의 손상이 안 된 복사본을 정확히 수신했음을 확인할 수 있다. 응답에서, 원격 클라이언트 워크스테이션(156)은 수신확인(262)을 생성할 수 있다. 이 예에서, 이 수신확인(262)은 보안 채널(260)을 사용하여 서버(122)로 송신된다. 이 방식으로, 서버(122)가 원격 클라이언트 워크스테이션(156)에 요청된 아티팩트의 복사본을 직접 공급하지는 않지만, 서버(122)는 원격 클라이언트 워크스테이션(156)이 요청된 아티팩트의 복사본을 가짐을 확실하게 할 수 있다.

[0059] 도 2b에 도시된 이 시나리오는 원격 사이트(112)가 더 큰 기업의 개발 사무실일 때 발생하는 전형적인 상호교환일 것이다. 원격 사이트(112)는 특정 제품을 위한 개발하의 소스 코드 파일들의 최신 버전을 모두가 액세스하는 복수 명의 개발자들을 포함할 수 있다. 따라서, 각 개발자는 매일 소스 파일들 각각의 복사본들로 로드되는 원격 클라이언트 워크스테이션(156)과 같은 원격 클라이언트 워크스테이션을 사용할 수 있다. HTTP와 같은, 비보안된 프로토콜을 사용하여, 파일들을 전송하기 위해, 프락시 서버(150)는 특정 파일들이 원격 클라이언트 워크스테이션들로 전송되어 그들의 복사본들을 캐시함으로서 그 특정 파일들을 식별할 수 있다. 프락시 서버(150)가 비보안된 서버일지라도, 아티팩트들로의 비승인된 액세스는 그들이 암호화되므로 아티팩트들에 대해 정보를 드러내지 않는다. 그러나, 프락시 서버(150)가 아티팩트들이 전송되면서 아티팩트들을 식별할 수 있으므로, 그것은 그들을 캐시하여 그 아티팩트들에 대한 후속적 요청들에 응답할 수 있다. 이 방식으로, 인터넷(114) 또는 원격 사이트(112)와 중앙 사이트(110) 간의 어떤 다른 접속을 통해 전송된 정보의 양은 크게 감소될 수 있다.

[0060] 도 2c는 원격 클라이언트 워크스테이션(156)으로부터 송신된 아티팩트에 대한 요청에 응답하여 발생할 수 있는 다른 트랜잭션(transaction)을 도시한다. 이 도면에서, 원격 클라이언트 워크스테이션(156)으로부터의 요청(272)이 송신된다. 요청(272)은 인터넷(114)을 통해 형성될 수 있는 보안 채널(270)을 통해 송신된다. 요청(272)은 서버(122)로 전달되어 원격 클라이언트 워크스테이션(156)에 제공될 아티팩트 또는 아티팩트들을 식별한다.

[0061] 서버(122)는 번들(274)을 송신하여 요청(272)에 응답한다. 번들(274)은 원격 클라이언트 워크스테이션(156)이 적절한 아티팩트를 위한 요청을 형성할 수 있는 식별자를 포함한다. 번들(274)은 또한 암호화 키 및, 오류 검사 코드와 같은, 아티팩트에 관련된 다른 정보를 포함할 수 있다.

[0062] 원격 클라이언트 워크스테이션(156)은 번들(274)에 포함된 아티팩트에 대한 정보를 사용하여 요청(276)을 생성한다. 요청(276)은 번들(274)의 식별자를 사용하여 아티팩트에 대한 요청을 형식화한다. 요청(276)은 본 명세서에서 프락시 서버(150)에 전달되기 위해 도시된다.

[0063] 이 예에서, 프락시 서버(150)는 그것의 캐시에 요청되는 아티팩트의 복사본을 갖지 않는다. 따라서, 요청(278)은 프락시 서버(150)로부터 생성된다. 요청(278)은 인터넷(114)을 통해 역 프락시 서버(140)로 전달된다.

[0064] 도 2c에 도시된 예에서, 원격 프락시 서버(140)는 그것의 캐시에 요청된 아티팩트의 복사본을 저장한다. 따라서, 역 프락시 서버(140)는 요청(278)에 응답하여 응답(280)을 생성한다. 응답(280)은 요청된 아티팩트의

복사본을 포함한다. 아티팩트는 어떤 적절한 프로토콜로 전송될 수 있지만, 본 명세서에 기재된 실시예들에서, HTTP 프로토콜이 사용된다. 아티팩트는 암호화된 형태로 전송된다.

[0065] 응답(280)은 인터넷(114)을 통해 프락시 서버(150)에 전달된다. 프락시 서버(150)가 그것의 캐시에 저장된 요청된 아티팩트의 복사본을 갖지 않으므로, 프락시 서버(150)는 응답(280)에 아티팩트의 복사본을 저장한다. 아티팩트는 요청(276)에 사용되는 URL에 의해 인덱스되는 프락시 서버와 연관되는 캐시에 저장될 수 있다. 요청(276)에 사용되는 URL은 실제 파일 이름보다는 번들(274)에 제공되는 식별자를 사용한다.

[0066] 프락시 서버(150)는 암호화된 아티팩트의 복사본을 응답(282)의 일부로서 전달한다. 원격 클라이언트 워크스테이션(156)은 응답(282)을 수신한다. 원격 클라이언트 워크스테이션(156)은 번들(274)의 일부로서 제공되는 암호화 키를 사용하여 응답(282)에 포함되는 아티팩트의 암호화된 복사본을 해독할 수 있다. 원격 클라이언트 워크스테이션(156)은 또한 오류 검사 코드를 해독된 파일에 적용하여, 파일이 원격 사이트(112)와 중앙 사이트(110) 간의 네트워크 접속의 비보안된 부분들을 통해 전송되는 동안에 그 파일이 손상되거나 탬퍼링되지 않았다고 결정할 수 있다.

[0067] 그 다음, 원격 클라이언트 워크스테이션(156)은 서버(122)에 수신확인(286)을 송신할 수 있다. 이 예에서, 수신확인(286)은 보안 채널(284)를 통해 전송된다.

[0068] 프락시 서버(150)와 역 프락시 서버(140)가 요구되지는 않지만, 도 2c는 역 프락시 서버(140)를 포함하는 이득을 기업의 정보 관리 시스템의 일부로서 도시한다. 역 프락시 서버(140)는 WAN(124)을 통해 전송되는 정보의 양을 감소시킨다. 그것은 또한 서버(122)가 데이터베이스(120)로부터의 아티팩트들을 암호화하면서 보내는 시간의 양을 감소시킨다.

[0069] 도 3은 정보 관리 시스템이 동작할 수 있는 프로세스를 도시한다. 프로세스는 클라이언트가 서버와 보안 접속을 시작하는 블록(310)에서 시작한다, 클라이언트는 도 1에 도시된 (156)과 같은 원격 클라이언트 워크스테이션일 수 있다. 그러나, 동일한 프로세스가 중앙 사이트에 위치된, 126_1, 126_2,... 126_4과 같은, 클라이언트들에 채택될 수 있다.

[0070] 블록(312)에서, 클라이언트는 서버로부터의 한 개 이상의 파일들을 요청한다. 블록(312)에 송신된 요청은 한 개 이상의 파일들을 식별할 수 있다. 요청이 보안 채널을 통해 송신되므로, 요청은 각 파일을 이름이나 또는 어떤 식별 형식으로, 그것이 기업 내에 보안적으로 보유되는 것이 선호되는 파일에 대한 정보를 드러내더라도, 각 파일을 식별할 수 있다. 어떤 적절한 형식은 한 개 또는 복수 개의 파일들을 요청하기 위해 채택될 수 있다. 예를 들어, 복수 개의 파일들에 대한 요청은 개별 파일들에 대한 일련의 요청들로서 형식화될 수 있다. 그러나 어떤 적절한 형식이 채택될 수 있다.

[0071] 블록(314)에서, 서버는 요청된 파일들에 관한 정보를 제공하는 번들으로 응답한다. 번들은 비보안된 채널을 통해 보안된 방식으로 파일들을 액세스하기 위해 요구되는 정보를 포함한다. 이 예에서, 번들은 파일을 요청하기 위한 네트워크 주소를 형성하기 위해 사용될 수 있는 식별자를 각 파일에 대해 포함한다. 번들은 또한 각 파일에 대해 암호화 키를 포함한다. 암호화 키는 파일들의 각각에 대해 상이한 것이 선호된다. 각 파일에 대해 개별적 암호화 키를 사용하는 것은, 한 개의 암호화 키가 손상되더라도, 정보 관리 시스템에 저장된 전체 정보의 단지 비교적 적은 퍼센트만이 손상되도록 하는 것을 확실하게 한다. 번들은 요청된 파일에 대한 다른 정보를 추가적으로 포함할 수 있다. 위에 기재된 예들에서, 추가적 정보가 오류 검사 코드에 포함되어, 파일의 전송이나 또는 탬퍼링에서의 오류들이 식별될 수 있다. 번들에 포함될 수 있는 다른 가능한 정보는 파일의 크기, 데이터베이스(120)에 그것이 저장된 날짜, 또는 파일을 요청하는 클라이언트에 유용한 다른 정보일 수 있다. 번들이 복수 개의 파일들에 대한 정보를 제공하면, 정보는 어떤 적절한 형식으로 제공될 수 있다. 예를 들어, 정보는, 각 파일 당 한 개의 집합으로, 데이터 집합들의 스트림으로서 형식화되어 제공될 수 있다.

[0072] 블록(316)에서, 클라이언트는 번들에 제공된 정보를 사용하여 비보안된 통신 채널을 통해 한 개 이상의 파일들을 요청한다.

[0073] 결정 블록(318)에서, 클라이언트가 액세스를 하는 프락시 서버로부터 파일을 이용할 수 있는지에 대한 결정이 되어진다. 만약 그렇다면, 프로세싱은 프락시가 파일을 제공하는 블록(320)으로 진행한다. 파일은 암호화된 형태로 제공된다.

[0074] 결정 블록(318)에서, 클라이언트에게 파일이 로컬에서 이용가능하지 않다는 것이 결정되면, 프로세싱은 블록

(330)으로 진행한다. 블록(330)에서, 프락시는 중앙 위치로 파일 요청을 전달한다.

[0075] 결정 블록(322)에서, 중앙 위치의 역 프락시는 그것이 요청된 파일의 복사본을 캐시했는지를 판정한다. 만약 그렇다면, 프로세싱은 역 프락시가 그 파일을 제공하는 블록(338)으로 진행한다.

[0076] 역 프락시가 그 파일을 캐시하지 않았을 경우에, 프로세싱은 블록(334)으로 진행한다. 블록(334)에서, 데이터베이스를 관리하는 서버는 요청된 파일을 검색한다. 암호화는, 서버(122)일 수도 있는, 어떤 적절한 컴퓨터 프로세서에서 수행될 수 있지만, 다른 서버들 또는 컴퓨터들이 암호화를 수행하기 위해 사용될 수 있다. 프로세스 블록(336)에서, 파일이 암호화된다.

[0077] 프로세스는 블록(338)에서 계속한다. 역 프락시 서버가 그것의 캐시로부터 파일을 얻거나 또는 중앙 위치에서 서버에 의해 제공되는 파일의 버전을 암호화하는지에 무관하게, 블록(338)에서, 역 프락시 서버는 클라이언트에게 그 파일을 제공한다.

[0078] 파일이 클라이언트에게 제공되면, 프로세싱은 블록(340)에서 계속된다. 암호화된 파일이 역 프락시에 의해 또는 클라이언트 근처 사이트의 프락시에 의해 제공되든지에 무관하게, 프로세싱은 블록(340)으로 진행한다. 블록(340)에서, 원격 클라이언트는 그 파일을 해독한다. 그 다음, 그 해독된 파일은 원격 클라이언트에서 실행하는 애플리케이션에 제공될 수 있다.

[0079] 도 3에 도시된 프로세스는 어떤 적절한 방식으로 구현될 수 있다. 예를 들어, 파일 관리 시스템과의 인터액션을 제어하는 원격 클라이언트 워크스테이션의 소프트웨어는 프로토콜 스택의 애플리케이션 층의 소프트웨어로서 구현될 수 있다.

[0080] 파일들이 HTTP와 같은 표준 프로토콜을 사용하여 전송되는 실시예들에서, 프락시 서버와 역 프락시 서버(140)는 현재 알려진 또는 이후에 개발되는 어떤 애플리케이션의 프락시 서버들을 위해 사용되는 것과 같은 종래의 하드웨어 및 소프트웨어 엘리먼트들일 수 있다. 현재 알려졌거나 또는 이후에 개발되는지에 무관하게, 유사하게, 서버(122)와 데이터베이스(120)는 종래 서버와 데이터베이스 하드웨어와 소프트웨어 액세스를 사용하여 구현될 수 있다. 서버(122) 또는 역 프락시 서버(140)는 소프트웨어로 프로그램되어 파일들을 암호화하고 원격 클라이언트들에 의해 발생된 요청들에 응답하여 번들들을 제공할 수 있다. 예를 들어, 그런 소프트웨어는 중앙 사이트에서 또는 중앙 사이트에 액세스할 수 있는 어떤 편리한 하드웨어 또는 소프트웨어에서 서버에 병합될 수 있다. 예를 들어, 그런 프로그램은 프로토콜 스택의 애플리케이션 레벨로 병합될 수 있다.

[0081] 다양하고 상이한 실시예들이 가능하다. 예를 들어, 아티팩트들은, 비보안 네트워크에 노출된 아티팩트들 중의 어떤 것에 대한 정보의 양을 감소시킬 수 있는, 코드화된 식별자를 사용하여 비보안 네트워크를 통해 요청된다고 기재된다. 아티팩트에 대해 이름 또는 다른 식별자를 사용하는 것이 바람직하지 않은 양의 정보를 누설하지 않는 경우에, 요청이 코드화된 식별자를 사용하는 것이 필요하지 않다.

[0082] 다른 예에서, 암호화된 파일들의 해독이 원격 클라이언트 워크스테이션에서 발생한다고 기재되었다. 복호화 프로세스는 어떤 적절한 프로세서에서 수행될 수 있다. 프락시 서버(150)로의 비승인된 액세스가 염려가 안 되는 경우에, 프락시 서버(150)는 복호화를 수행할 수 있고, 그것의 캐시에 아티팩트들의 해독된 복사본들을 저장할 수 있다. 다른 경우에, 원격 사이트(112)의 분리된 프로세서가 아티팩트들의 복호화를 수행하기 위해 사용될 수 있다.

[0083] 유사하게, 암호화가 발생하는 시간과 장소는 또한 변경될 수 있다. 예를 들어, 암호화된 아티팩트들은 데이터베이스(120)에 저장될 수 있다. 그런 실시예에서, 서버(122)는 보안 네트워크를 통해 클라이언트 워크스테이션들 126_1, ...126_4 또는 서버(122)에 접속된 다른 프로세서들에 파일들을 제공하기 전에 그 파일들을 해독할 수 있다. 다른 경우에, 클라이언트 워크스테이션들 126_1, ...126_4은 암호화된 아티팩트들을 수신할 수 있고, 사용하기 전에 그들을 해독할 수 있다. 예를 들어, 이 방식으로 프로세싱 부하를 재분산하는 것은, 원격 사이트들로 분산된 정보의 양이 중앙 사이트(110)에서 사용되는 정보의 양과 비교하여 큰 경우에 바람직할 것이다. 이 관점에서, "중앙"과 "원격"은 아티팩트들을 저장하는 데이터베이스와 사용하기 위해 아티팩트들을 수신하는 프로세서 간의 네트워크 접속들의 특성을 나타내는 용어들이다. 데이터베이스(120)는 정보 관리 시스템을 사용하는 기업에서 중앙의 위치에 저장될 필요는 없다.

[0084] 본 발명의 적어도 한 개의 실시예의 여러 양상들을 기재했으므로, 다양한 변경, 수정, 및 개선을 당업자들이 쉽게 할 수 있음을 이해할 것이다.

[0085] 그런 변경, 수정, 및 개선은 이 개시의 일부인 것으로서 의도되고, 본 발명의 사상 및 범위 내에 있는 것으로

서 의도된다. 따라서, 전술한 설명과 도면들은 단지 예일 뿐이다.

[0086] 본 발명의 상술한 실시예들은 다수의 방식들 중의 어떤 것으로 구현될 수 있다. 예를 들어, 실시예들은 하드웨어, 소프트웨어, 또는 그들의 조합을 사용하여 구현될 수 있다. 소프트웨어에 구현될 때, 소프트웨어 코드는, 한 개의 컴퓨터에 제공되거나 또는 복수 개의 컴퓨터들 간에 분산되는지에 무관하게, 어떤 적절한 프로세서 또는 프로세서들의 집합에서 실행될 수 있다.

[0087] 또한, 본 명세서에 기술된 다양한 방법들과 프로세스들은 다양한 운영 시스템들 또는 플랫폼들 중의 어떤 것을 채택하는 한 개 이상의 프로세서들에서 실행가능한 소프트웨어로서 코드화될 수 있다. 또한, 그런 소프트웨어는 다수의 적절한 프로그래밍 언어들 및/또는 종래의 프로그래밍이나 스크립팅 툴들 중의 어떤 것을 사용하여 작성될 수 있고, 또한 실행가능한 기계어 코드로서 컴파일될 수 있다.

[0088] 이 관점에서, 본 발명은, 한 개 이상의 컴퓨터들 또는 다른 프로세서들에서 실행될 때, 위에 논의된 본 발명의 다양한 실시예들을 구현하는 방법들을 수행하는 한 개 이상의 프로그램들로 인코드되는 컴퓨터 판독가능 매체(또는 복수 개의 컴퓨터 판독가능 매체)(예를 들어, 컴퓨터 메모리, 한 개 이상의 플로피 디스크, 컴팩트 디스크, 광 디스크, 자기 테이프 등)로서 구현될 수 있다. 컴퓨터 판독가능 매체 또는 매체들은, 거기에 저장된 프로그램이나 프로그램들이 한 개 이상의 다른 컴퓨터들 또는 다른 프로세서들에 로드되어 위에 논의된 것처럼 본 발명의 다양한 양상들을 구현할 수 있도록, 전송가능하다.

[0089] "프로그램"이라는 용어는 컴퓨터 또는 다른 프로세서를 프로그램하여 위에 논의된 것처럼 본 발명의 다양한 양상들을 구현하기 위해 채택될 수 있는 어떤 유형의 컴퓨터 코드 또는 명령들의 집합을 일컫기 위한 일반적인 의미로 본 명세서에서 사용된다. 추가하여, 이 실시예의 일 양상에 따라, 실행될 때 본 발명의 방법들을 수행하는 한 개 이상의 컴퓨터 프로그램들이 한 개의 컴퓨터 또는 프로세서에 존재할 필요가 없고, 다수의 상이한 컴퓨터들 또는 프로세서들 간에 모듈형(modular) 방식으로 분산되어 본 발명의 다양한 양상들을 구현할 수 있음을 이해해야 한다.

[0090] 본 발명의 다양한 양상들은 단독으로, 조합하여, 또는 전술에서의 실시예들에서 특정하게 논의되지 않은 다양한 배치들에서 사용될 수 있고, 그러므로 전술된 설명에 기재되거나 또는 도면들에 도시된 컴포넌트들의 세부사항들과 배치로만 그것의 응용에서 제한되지는 않는다. 예를 들어, 일 실시예에 기재된 양상들은 다른 실시예에서 기재된 양상들과 어떤 방식으로 조합될 수 있다.

[0091] 또한, 단계들의 타이밍과 순서는 변경될 수 있다. 예를 들어, 도 2a~2c에 의해 도시된 인터액션들은 요청을 발생시키고 특정 아티팩트에 관련된 식별자와 암호화 키를 수신하는 원격 클라이언트에서 시작한다. 한 개의 아티팩트에 대한 식별자와 암호화 키가 변경될 수 있음이 가능하다. 만약 그렇다면, 각 원격 클라이언트가 그것이 파일을 요구하는 각 시간에 식별자와 암호화 키를 요청하는 것이 필요할 것이다. 그러나, 클라이언트 워크스테이션이 파일에 대한 식별자와 암호화 키의 복사본을 저장할 수 있고, 그것이 이전에 얻어서 저장했던 암호화 키를 사용하여 (216, 256, 또는 276)과 같은 요청들을 생성할 수 있음이 가능하다.

[0092] 청구항 엘리먼트를 수정하기 위해 청구범위에서 "제1", "제2", "제3" 등과 같은 순서적 용어들의 사용은 어떤 우선순위, 선행관계, 또는 한 청구항 엘리먼트의 다른 엘리먼트에 대한 순서, 또는 방법의 단계들이 수행되는 임시 순서를 그 자체가 내포하지는 않고, 청구항 엘리먼트들을 구별하기 위해 특정 이름을 갖는 한 개의 청구항 엘리먼트를 동일한 이름을 갖는 다른 엘리먼트로부터 구별하기 위한 단지 레이블들로서 사용된다(그러나, 순서적 용어의 사용을 위해).

[0093] 또한, 본 명세서에 사용된 어법과 용어는 설명 목적일 뿐이고, 제한적인 것으로서 간주되어서는 안 된다. 본 명세서에서 "포함하는(including)", "포함하는(comprising)", 또는 "갖는(having)", "포함하는(containing)", "관련되는(involving)", 및 그들의 변형들의 사용은 이후에 리스트되는 항목들 및 그들의 동등물들 및 추가적 항목들을 포함하려고 의도된다.

발명의 효과

[0094] 중앙 사이트와 원격 사이트가 네트워크로 연결된 환경에서 소스 코드 관리 시스템으로도 사용될 수 있는 분산 정보 시스템이 개시된다. 클라이언트가 네트워크를 통해 서버로부터 아티팩트(예를 들어, 파일)에 대한 암호화 정보를 얻을 수 있고, 이 암호화 정보를 바탕으로 프락시의 캐시 또는 서버와 연결된 데이터베이스로부터 암호화된 아티팩트의 복사본을 수신하여 해독하여 사용할 수 있다. 수취된 아티팩트의 복사본은 로컬 프락시의 캐시에 저장되어, 추후 관련된 요청시에 꺼내서 사용되므로 서버와의 불필요한 통신의 양을 감소시킬 수

있다.

도면의 간단한 설명

[0001] 도 1은 본 발명의 실시예에 따른 정보 관리 시스템의 아키텍처를 나타내는 도면.

[0002] 도 2a는 도 1의 정보 관리 시스템에 도시된 장치 간의 통신을 나타내는 도면.

[0003] 도 2b는 다른 동작 상태에 따른 도 1의 정보 관리 시스템의 장치 간의 통신을 나타내는 도면.

[0004] 도 2c는 다른 동작 상태에 따른 도 1의 정보 관리 시스템의 장치 간의 통신을 나타내는 도면.

[0005] 도 3은 본 발명에 따른 정보 관리 시스템의 정보의 프로세싱을 나타내는 플로차트.

[0006] <도면의 주요 부분에 대한 부호의 설명>

[0007] 110 : 중앙 사이트

[0008] 112 : 원격 사이트

[0009] 120 : 데이터베이스

[0010] 122 : 서버

[0011] 126_1 ~ 126_4 : 클라이언트 워크스테이션

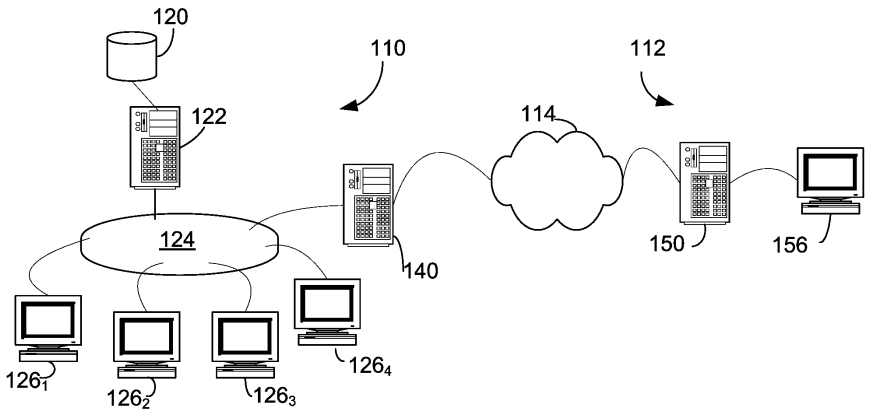
[0012] 140 : 역 프락시 서버

[0013] 150 : 프락시 서버

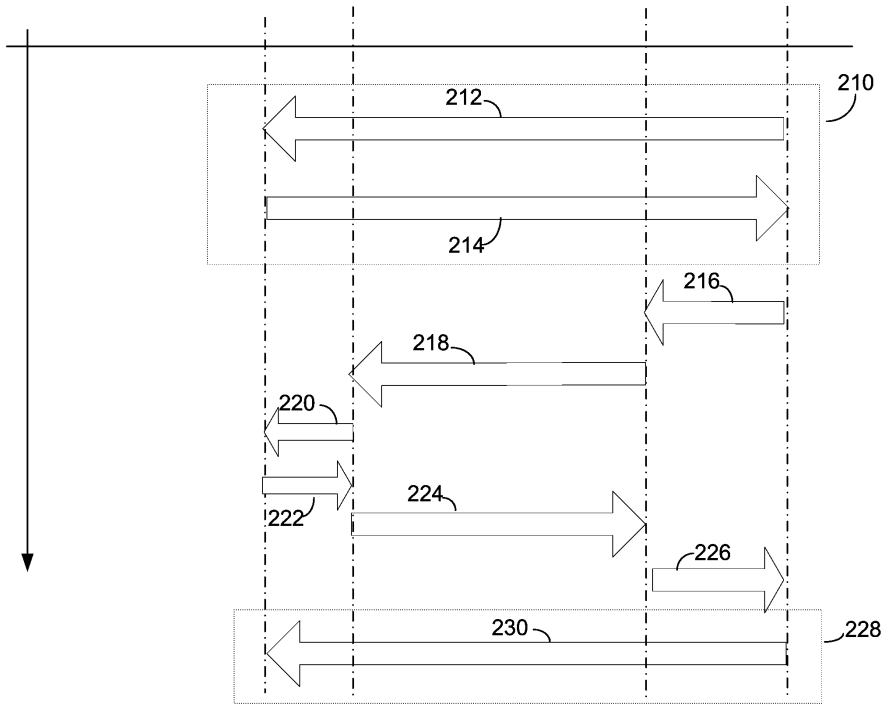
[0014] 156 : 원격 클라이언트 워크스테이션

도면

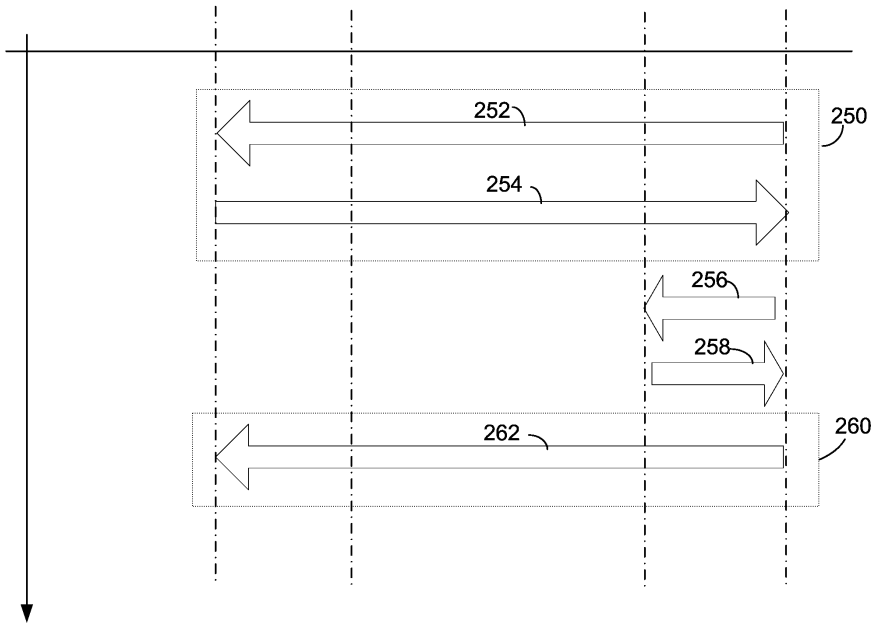
도면1



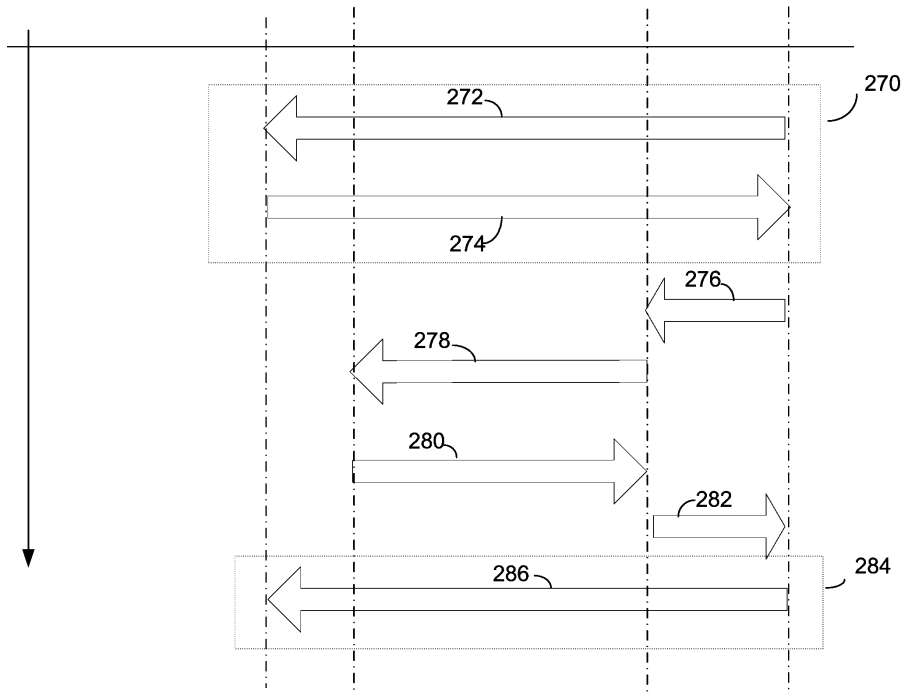
도면2a


210
212
214
216
218
220
222
224
226
228
230

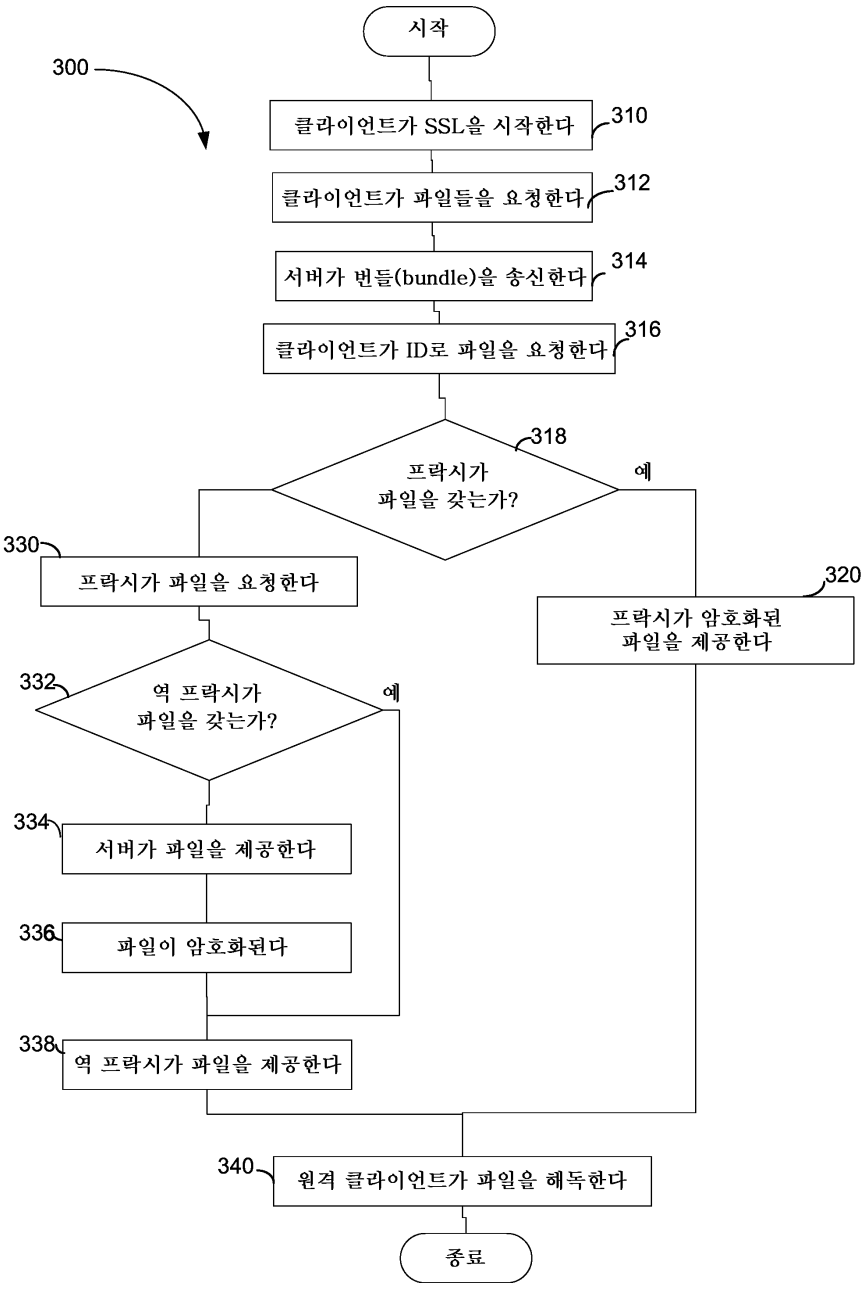
도면2b


250
252
254
256
258
260
262

도면2c



도면3


시작
300
클라이언트가 SSL을 시작한다
310
클라이언트가 파일들을 요청한다
312
서버가 번들(bundle)을 송신한다
314
클라이언트가 ID로 파일을 요청한다
316
318
프락시가
파일을 갖는가?
예
330
프락시가 파일을 요청한다
320
프락시가 암호화된
파일을 제공한다
332
역 프락시가
파일을 갖는가?
예
334
서버가 파일을 제공한다
336
파일이 암호화된다
338
역 프락시가 파일을 제공한다
340
원격 클라이언트가 파일을 해독한다
종료