(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5058492号
(P5058492)
(45)発行日　平成24年10月24日(2012.10.24)　(24)登録日　平成24年8月10日(2012.8.10)

(51)Int.Cl.　　F I
H04L 9/32 (2006.01)　　H04L 9/00 675B
G06F 12/00 (2006.01)　　G06F 12/00 546K

請求項の数 9　(全 18 頁)

(21)出願番号　特願2006-32953(P2006-32953)
(22)出願日　平成18年2月9日(2006.2.9)
(65)公開番号　特開2006-279933(P2006-279933A)
(43)公開日　平成18年10月12日(2006.10.12)
審査請求日　平成21年1月19日(2009.1.19)
(31)優先権主張番号　11/090,681
(32)優先日　平成17年3月25日(2005.3.25)
(33)優先権主張国　米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　ユージン　ダブリュ．ホッジス
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　ワン　マイクロソフト　ウェイ　マイクロソフト　コーポレーション内

最終頁に続く

(54)【発明の名称】分散情報管理方法および分散情報管理装置

(57)【特許請求の範囲】
【請求項１】
複数のアーティファクトへのアクセスを有し、第１のサイトにあるサーバであって、前記第１のサイトにあるプロキシサーバに接続されたサーバと、第２のサイトにあるクライアントであって、前記第１のサイトと前記第２のサイトとを接続するネットワークを介した前記プロキシサーバへのアクセスを有するクライアントと、の間の通信方法であって、前記プロキシサーバは、前記サーバと前記ネットワークとの間に接続され、前記プロキシサーバのキャッシュと関連付けられ、該キャッシュは、前記サーバによって暗号化されたアーティファクトのコピーを格納し、前記方法は、
前記クライアントが、セキュアなコネクションを介して、前記アーティファクトに対する第１の要求を前記サーバに送信するステップと、
前記クライアントが、前記第１の要求に応答して、前記セキュアなコネクションを介して前記アーティファクトについての情報を前記サーバから受信するステップであって、前記情報は、前記暗号化されたアーティファクトに関する暗号化情報と、前記暗号化されたアーティファクトに関する識別子とを含む、ステップと、
前記クライアントが、前記識別子を用いて、セキュアでないコネクションを介して前記アーティファクトに関する第２の要求を前記プロキシサーバに送信するステップと、
前記暗号化されたアーティファクトのコピーが前記キャッシュに格納されている場合、前記クライアントが、前記第２の要求に応答して、前記セキュアでないコネクションを介して、前記プロキシサーバの前記キャッシュから前記暗号化されたアーティファクトのコ

ピーを受信するステップと
前記クライアントが、前記暗号化情報を用いて前記暗号化されたアーティファクトのコピーを復号化するステップと
を備えることを特徴とする方法。
【請求項２】
前記アーティファクトのコピーが前記キャッシュに格納されていない場合、選択的に前記第２の要求を前記サーバに提供するステップをさらに備えることを特徴とする請求項１に記載の方法。
【請求項３】
前記第１の要求を送信するステップは、前記クライアントと前記サーバとの間の前記セキュアなコネクションを確立するステップを備えることを特徴とする請求項１に記載の方法。
【請求項４】
前記セキュアなコネクションを確立するステップは、ＳＳＬコネクションを確立するステップを備えることを特徴とする請求項３に記載の方法。
【請求項５】
第１のサイトにある装置と、第２のサイトにある装置と、前記第１のサイトにある装置及び前記第２のサイトにある装置を接続するネットワークと、を有するソースコードコントロールシステムであって、
前記第１のサイトにある装置は、
複数のソースコードファイルの複数のバージョンを格納するメモリストレージデバイスと、
前記メモリストレージデバイスにアクセスして、ソースコードファイルのバージョンを暗号化し、暗号化されたアーティファクトを生成し、前記ネットワークを介して前記暗号化されたアーティファクトを送信し、前記暗号化されたアーティファクトに関する識別子と前記暗号化されたアーティファクトに関する暗号化情報とを含む情報をセキュアなコネクションを介して送信するための第１のサーバと
前記第１のサーバと前記ネットワークとの間に接続された第２のサーバであって、前記第１のサーバから前記セキュアなコネクションを介して前記ソースコードファイルに関する第１の要求を受信したことに応答して前記第１のサーバから前記暗号化されたアーティファクトを受信し、前記第２のサーバのキャッシュに前記暗号化されたアーティファクトを格納し、前記第２のサイトにある装置から前記暗号化されたアーティファクトに関する第２の要求を受信したことに応答して、前記暗号化されたアーティファクトを送信する、第２のサーバと
を備え、
前記第２のサイトにある装置は、前記セキュアなコネクションを介してソースコードファイルに関する前記第１の要求を送信し、前記情報を前記セキュアなコネクションを介して受信し、前記識別子を用いて、前記ネットワークを介して前記アーティファクトに関する前記第２の要求を送信し、前記ネットワークを介して前記暗号化されたアーティファクトを受信し、前記暗号化情報を使用して前記暗号化されたアーティファクトを復号化し、前記ソースコードファイルのバージョンを生成するクライアントコンピュータを備えることを特徴とするソースコードコントロールシステム。
【請求項６】
前記第１のサーバは、複数のソースコードファイルのバージョンのそれぞれを異なる暗号化キーで暗号化することを特徴とする請求項５に記載のソースコードコントロールシステム。
【請求項７】
前記メモリストレージデバイスは、第１のタイプの識別子に関連付けて複数のソースコードファイルの複数のバージョンのそれぞれを格納し、
前記第１のサーバは、前記暗号化されたアーティファクトに関する第２のタイプの識別

子を前記ネットワークを介して通信し、
前記キャッシュは、前記第２のタイプの識別子と関連付けて、前記複数の暗号化されたソースコードファイルを格納することを特徴とする請求項５に記載のソースコードコントロールシステム。
【請求項８】
前記ネットワークを介したページのダウンロードに適応されるブラウザをそれぞれ保持する複数のクライアントコンピュータをさらに備え、
前記キャッシュは、前記ネットワークを介してダウンロードされた前記ページをキャッシュすることを特徴とする請求項５に記載のソースコードコントロールシステム。
【請求項９】
前記ネットワークはインターネットを備えることを特徴とする請求項８に記載のソースコードコントロールシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般的には情報管理システムに関し、より詳細には分散情報管理システムに関する。
【背景技術】
【０００２】
情報管理システムは広く使用されている。このようなシステムは、しばしば「アーティファクト」を蓄積するデータベースを含んでいる。アーティファクトは、情報管理システムによって処理された、編成された形式（organized form）におけるデータの集合である。アーティファクトはしばしばコンピュータファイルである。情報管理システムの一般的な種類は、ソフトウェアを開発している企業においてコンピュータソースコードの管理ファイルとして使用されることができるような、バージョンコントロールシステムである。
【０００３】
情報管理システムは、しばしばネットワーク接続されて、企業内の複数の人々がアーティファクトを扱う仕事ができるようにする。ソースコード管理システムの例では、プログラム開発者、プロジェクト管理者、テスト技術者、その他企業内の他の人々全てがソースコードファイルにアクセスすることができる。これらのファイルを中央のデータベースに格納することができる。企業内を通じたファイルの利用を促進するために、情報管理システムはしばしばデータベースとリンクしたサーバを含んでいる。サーバは、ネットワークに接続され、情報管理システムにおける情報の個々のユーザのワークステーションがデータベースからアーティファクトのコピーを取得することを許可する。ユーザがアーティファクトを要求すると、ワークステーションはサーバに要求を送信し、サーバはネットワークを介してアーティファクトのコピーを提供する。
【０００４】
情報管理システムのネットワークアーキテクチャは、企業が比較的広い地域に分布した複数のワークサイトを有している場合にさえも使用されることがありえる。場合によりインターネットを含む広域ネットワークは、分散した位置にある個々のワークステーションが、アーティファクトのデータベースにアクセスするサーバと通信を行うことができるようにする。
【０００５】
クライアントとサーバとの間の通信回線のいずれかの部分がアンセキュアな場合、たとえネットワークがアンセキュアであっても、セキュアなチャネルがネットワーク上に生成される可能性がある。インターネットはアンセキュアな通信回線の例である。アンセキュアなネットワーク上に生成されるセキュアなチャネルの例は、ＳＳＬ（Ｓｅｃｕｒｅ Ｓｏｃｋｅｔ Ｌａｙｅｒ）接続またはＶＰＮ（Ｖｉｒｔｕａｌ Ｐｒｉｖａｔｅ Ｎｅｔｗｏｒｋ）である。
【０００６】

セキュアなチャネルにおいて、通信プロトコルは、たとえ伝送路が傍受されたとしても、意図しない受取人がネットワークを介して送信された情報の内容を判断することが困難なプロトコルである。例えば、ＳＳＬチャネルを介して送信するデバイスは、情報が送信される際に情報を暗号化する。また、情報は、送信される情報の関連性が容易に検出できないように断片化される場合がある。結果として、意図しない受信者は、ファイルや他の論理的に関連するデータブロックを表す伝達の一部を識別することさえもできない可能性があり、暗号化のメカニズムの「解読」を図ることをより難しくしている。

【発明の開示】

【発明が解決しようとする課題】

【０００７】

セキュアチャネルを使用する欠点は、セキュアチャネルを介してユーザが情報をダウンロードするのにかかる時間の量と、クライアント－サーバ環境において課されるオーバーヘッドである。ＳＳＬは、それぞれのユーザのセッションに対して、固有の非対称キーペアを使用する。固有の非対称キーは、チャネルを介して送信される際のデータの暗号化に使用される。サーバ上の情報が複数のクライアントに配信される場合に、ＳＳＬチャネルはサーバに高いオーバーヘッドを与える。アンセキュアなチャネルで接続された地理的に分布したサイトをもつ企業が操作できる等の、情報管理システムのための改良された方法および装置を提供することが望まれる。

【課題を解決するための手段】

【０００８】

本発明は、コンピューティングデバイスがセキュアチャネルを介してアーティファクトに関する情報を得ることができる情報管理システムに関する。この情報を使用して、アンセキュアなチャネルを介して、暗号化されたアーティファクトのバージョンを取得し復号化する。

【０００９】

１つの態様では、本発明は、複数のアーティファクトにアクセスしているサーバと、プロキシサーバに関連付けられたキャッシュであって、アーティファクトのコピーを格納するキャッシュを有するプロキシサーバにアクセスしているクライアントと、の間の通信方法に関する。この方法は、クライアントからサーバへの第１の要求を通信すること、要求に応答して、アーティファクトに関する暗号化された情報をサーバからクライアントに通信すること、暗号化された情報を用いてクライアントからプロキシサーバへの第２の要求を生成すること、及び、アーティファクトのコピーがキャッシュに格納された時に、第２の要求に応答してキャッシュからアーティファクトのコピーを提供することを備える。

【００１０】

さらなる態様として、本発明は、アーティファクトにアクセスするためのコンピューティングデバイスの動作方法に関する。セキュアなチャネル及びアンセキュアなチャネルを通じてネットワークに接続されているデバイスを形成することができる。この方法は、セキュアチャネルを使用してアーティファクトに関する情報を受信すること、アーティファクトに関する情報を使用してアンセキュアなチャネルを使用し、アーティファクトのコピーを要求すること、暗号化された形式でアーティファクトを受信すること、及び、アーティファクトに関する情報を使用して、アーティファクトを暗号化された形式から復号化された形式に変換することを備える。

【００１１】

さらなる態様として、本発明は、第１のサイトにある装置と、第２のサイトにある装置と、第１のサイト及び第２のサイトを相互に接続するネットワークと、を有する種類のソースコードコントロールシステムに関する。第１のサイトにある装置は、複数のソースコードファイルの複数のバージョンを格納するメモリストレージデバイス、暗号化されたアーティファクトを生成するために、ソースコードファイルのバージョンを暗号化するためのコンピュータ実行可能命令を格納するコンピュータ読取可能な媒体を備えるサーバ、暗号化されたアーティファクトをネットワークを介して通信すること、及び、暗号化された

アーティファクトに関する暗号化情報をネットワークを介して通信すること、を備える。第２のサイトにある装置はクライアントコンピュータを備える。このクライアントコンピュータは、暗号化情報を取得するためのコンピュータ実行可能な命令を格納するコンピュータ読取可能な媒体、ネットワークを介して暗号化されたアーティファクトを受信すること、及び、暗号化情報を使用して暗号化されたアーティファクトを復号し、ソースコードファイルのバージョンを生成すること、を備えている。

【発明を実施するための最良の形態】

【００１２】

添付図面は実物大であることを意図していない。図において、様々な形で描かれている各々の同一の部分又はほとんど同一の部分は同等の符号で表されている。簡明にするために、各図の全ての部分に符号が付されていない場合がある。

【００１３】

改良された情報管理システムは、アンセキュアなチャネルを介してアーティファクトのセキュアな通信を可能とすることによって提供される。アーティファクトは暗号化形式で通信され、アーティファクトを受信するワークステーションのある地域のプロキシサーバに格納される可能性がある。セキュアチャネルを使用して、それぞれのアーティファクトについての比較的少量の情報を送信する。このようなシステムを使用して、例えば、企業のリモートサイトを、インターネットのようなアンセキュアなチャネルを使用しているセントラルサイトと接続することができる。ソースコード管理システムは、ここでは情報管理システムの例として使用される。

【００１４】

図１は、本発明の実施形態に係る情報管理システムを示している。情報管理システムは、セントラルサイト１１０とリモートサイト１１２とを含んでいる。セントラルサイト１１０とリモートサイト１１２とはネットワークを介して接続されている。このネットワークはインターネット１１４であってもよい。

【００１５】

セントラルサイト１１０はデータベース１２０を含む。データベース１２０は、コンピュータ読み書き可能な記録媒体から構成されている。データベース１２０は、アーティファクトの格納と検索を執り行うコントローラを含む。記載の実施形態では、それぞれのアーティファクトはソースコードを含むファイルであり、このソースコードファイルはセントラルサイト１１０及びリモートサイト１１２を運営する企業によって遂行されている開発プロジェクトの一部である。この例では、データベース１２０の中のそれぞれのファイルはファイル名で記述され、それぞれのファイルの複数バージョンをデータベース１２０に格納することができる。データベース１２０は公知技術のデータベースであってもよいが、任意の適切な様式のデータベースを使用することができる。

【００１６】

セントラルサイト１１０は、複数のクライアントワークステーション126_1、126_2、・・・、126_4を含む。使用の際は、セントラルサイト１１０及びリモートサイト１１２を管理する企業内の人が、それぞれのクライアントワークステーションを使用して、データベース１２０からのアーティファクトを処理することができる。それぞれのワークステーションは、例えば、データベース１２０に格納されているソースコードファイルを作成するコード開発者によって使用されてもよい。あるいは、クライアントワークステーションは、データベース１２０からソースコードファイルを取得しそれをテストするコードテスターによって利用されてもよい。それぞれのクライアントワークステーションは、例えば、パーソナルコンピュータか同等のコンピューティングデバイスであってもよい。

【００１７】

セントラルサイト１１０はサーバ１２２を含む。サーバ１２２はデータベース１２０に接続されている。サーバ１２２は、広域ネットワーク１２４を介してクライアントワークステーション126_1、126_2、・・・、126_4それぞれにアクセスすることが可能である。サーバ１２２は、クライアントワークステーション126_1、126_2、・・・、１

２６$_4$から、データベース１２０に格納されているアーティファクトをクライアントワークステーションに渡すよう要求する伝達（ｃｏｍｍｕｎｉｃａｔｉｏｎ）を受信する、ハードウェア及びソフトウェア要素の組み合わせとすることができる。サーバ１２２は、このような要求を受信し、データベース１２０にアクセスして要求されたアーティファクトのコピーをワークステーションに供給する、ハードウェア及びソフトウェア要素を含む。サーバ１２２は公知技術であるＨＴＴＰメッセージを利用して通信を行うファイルサーバであってもよいが、任意の適切な実装を用いることができる。

【００１８】

リモートサイト１１２は、１又は複数のリモートクライアントワークステーション１５６を含む。ここでは、簡単のために１つのリモートクライアントワークステーション１５６を図示しているが、複数のクライアントワークステーションがリモートサイトからデータベース１２０の中のアーティファクトにアクセスした場合に、発明は最も有用になる。リモートクライアントワークステーション１５６は、クライアントワークステーション１２６$_1$、１２６$_2$、・・・、１２６$_4$と同じ種類のワークステーションであり、同じ目的で使用することができる。このため、リモートクライアントワークステーション１５６は、クライアントワークステーション１２６$_1$、１２６$_2$、・・・、１２６$_4$と同じ様にデータベース１２０に格納されたアーティファクトにアクセスすべきである。しかしながら、リモートクライアントワークステーション１５６とデータベース１２０との間の情報が流れる経路はインターネット１１４を経由しており、このインターネット１１４がアンセキュアなネットワークである。

【００１９】

セキュアチャネルは、公知技術によりリモートクライアントワークステーション１５６とサーバ１２２との間に形成することができる。しかしながら、セキュアチャネルを使用する必要なしに、データベース１２０内のアーティファクトをリモートクライアントワークステーション１５６に直接転送する。データベース１２０に格納されているアーティファクトへのより速いアクセスを可能にするために、リモートクライアントワークステーション１５６とセントラルサイト１１０にあるサーバとの間のセキュアチャネルを使用して、比較的少量の情報を転送する。この情報は、インターネット１１４を介してアンセキュアなチャネル上を送信される暗号化されたアーティファクトのアクセスに使用され、かつ、その暗号化されたアーティファクトの使用を可能とする。アーティファクトは、ネットワークを介して情報を転送するために使用されるプロトコルの外側で暗号化されてもよい。これによりアーティファクトをより効率的に転送することが可能となる。

【００２０】

たとえアンセキュアな位置であったとしても、アンセキュアなチャネルを介したアーティファクトの送信効率が増加する方向においては、暗号化されたアーティファクトはキャッシュされる可能性がある。任意の適切なハードウェア及びソフトウェアを使用して、アーティファクトをキャッシュすることができる。図示した実施形態のように、リモートサイト１１２はプロキシサーバ１５０を含むが、このプロキシサーバ１５０はアーティファクトをキャッシュすることができるデバイスの一例である。プロキシサーバ１５０は公知技術のプロキシサーバであってもよい。プロキシサーバ１５０はリモートクライアント１５６とインターネット１１４との間を接続している。リモートクライアントワークステーション１５６がインターネット１１４を介してファイルやウェブページのようなアーティファクトのダウンロードを要求すると、プロキシサーバ１５０はアーティファクトのコピーを受信し、プロキシサーバ１５０に関連付けられたコンピュータ読み書き可能なメモリに格納することができる。格納された情報はアーティファクトのキャッシュを形成する。それに続くアーティファクトについての要求はキャッシュから実行することが可能であり、ネットワーク上のトラヒックを減少させることができる。

【００２１】

リモートクライアントワークステーション１５６がさらなる要求を生成すると、これらの要求はまずプロキシサーバ１５０を経由してもよい。プロキシサーバ１５０が、要求さ

れたアーティファクトを自サーバのキャッシュに格納した場合に、プロキシサーバ１５０はそのアーティファクトを自サーバのキャッシュからリモートクライアントワークステーション１５６に提供する。そして、その要求はインターネット１１４には転送されない。プロキシサーバ１５０に関連付けられたキャッシュから情報を提供することで、リモートクライアントワークステーション１５６に提供可能なアーティファクトの速度を増加させることができる。アクセスが１つのリモートクライアントワークステーションから行われるか、複数の異なるリモートクライアントワークステーションから行われるに関わらず、同じアーティファクトが頻繁にアクセスされた時にスピードの増加は最も高くなる。ソースコードマネジメントシステムにおいては、現在開発中のソースコードの一部を含むアーティファクトがしばしば頻繁にアクセスされる。

【００２２】

従来においては、インターネット１１４はセキュアなチャネルを提供するものと考えられていなかった。プロキシサーバは通常アンセキュアなチャネルを介して取得された情報を格納していたため、プロキシサーバはしばしばセキュアでなかった。リモートクライアントワークステーション１５６とサーバ１２２との間のセキュアな通信を提供するために、送信されるべきアーティファクトが暗号化される。プロキシサーバ１５０がセキュアでない場合には、アーティファクトを暗号化された形式でプロキシサーバ１５０にキャッシュすることができる。

【００２３】

さらに、暗号化されたアーティファクトをセントラルサイト１１０に格納することによって効果が得られる場合がある。図示した実施形態では、セントラルサイト１１０は、暗号化されたアーティファクトのコピーをも格納しているリバースプロキシサーバ１４０を含む。リバースプロキシサーバ１４０は、アーティファクトをキャッシュするのに使用される可能性のあるデバイスの一例である。リバースプロキシサーバ１４０は公知技術である規定に従ったキャシュの作動を含む公知技術のプロキシサーバであってもよい。リバースプロキシサーバ１４０はアーティファクトを暗号化してもよいし、又は、暗号化された形式でアーティファクトを受信してもよい。

【００２４】

図示した実施形態において、インターネット１１４上でアーティファクトについての要求が送信されると、これらの要求はリバースプロキシサーバ１４０に到達する。もし、リバースプロキシサーバ１４０が自サーバのキャッシュに要求されたアーティファクトの暗号化されたバージョンを格納している場合には、サーバ１４０は暗号化されたアーティファクトのコピーを提供する。リバースプロキシサーバ１４０がまだ自サーバのキャッシュに暗号化されたアーティファクトのコピーを格納していない場合には、サーバ１４０はサーバ１２２からアーティファクトを要求することができる。サーバ１２２は、暗号化されたアーティファクトを広域ネットワーク１２４を介して提供することができる。リバースプロキシサーバ１４０は自サーバのキャッシュに暗号化されたアーティファクトを格納し、インターネット１１４を介してその暗号化されたアーティファクトを送信することができる。

【００２５】

図２Ａは、図１で図示した情報管理システムの要素間における通信シーケンスを示している。情報の交換は、リモートクライアントワークステーション１５６がサーバ１２２とセキュアチャネルを確立することから始まる。図２Ａの実施形態においては、通信はセキュアチャネル２１０において初期化される。セキュアチャネル２１０は、現在知られているか今後開発されるかに関わらず、従来のセキュアプロトコルを用いて形成することができる。記載の実施形態においては、セキュアチャネル２１０はＳＳＬプロトコルを用いて生成される。要求２１２はセキュアチャネルを介してサーバ１２２に送信されるため、プロキシサーバ１５０及びリバースプロキシサーバ１４０のいずれも、要求の内容にはアクセスしない。本実施形態においては、要求２１２はサーバ１２２に直接送信される。

【００２６】

要求２１２は、リモートクライアントワークステーション１５６に提供されるべき１又は複数のアーティファクトを識別する。本実施形態では、それぞれのアーティファクトはファイル名で識別される。データベース１２０がファイルをバージョンコントロールシステムの一部として格納する場合には、ファイル名はファイルの特定のバージョンを特定することができる。セントラルサイト１１０にあるサーバ１２２は、要求２１２にバンドル（ｂｕｎｄｌｅ）２１４で応答する。

【００２７】

バンドル２１４も、セキュアチャネル２１０を介して送信される。バンドル２１４は、リモートクライアントワークステーション１５６が、要求されたアーティファクトを取得し利用できるように、情報を提供する。本実施形態においては、バンドル２１４はアーティファクトの暗号化されたバージョンに対する識別子を含む。バンドル２１４は、アーティファクトの暗号化されたバージョンの復号化に用いることのできる暗号化キーを含む。また、バンドル２１４は、ハッシュコードなどの、要求されたアーティファクトのために用意されたエラー検知コードを含むことができる。

【００２８】

本実施形態では、識別子はアーティファクトに割り当てられたコードである。データベース１２０の中のそれぞれのアーティファクトは、サーバ１２２に割り当てられた一意の識別子を有している。望ましくは、識別子はアーティファクトの機能又は構造についての情報を示していない。対照的に、ファイル名はしばしばアーティファクトの機能を説明するものが選択される。記載の実施形態では、識別子を使用して、アンセキュアなチャネルを介して送信される通信において、アーティファクトを引用する（ｒｅｆｅｒ）。ファイル名の代わりに説明的でない識別子を使用することで、セキュリティを増加させることができる。アンセキュアなチャネルを介した伝送の権限の無い受信者は、アーティファクトの暗号化を「解読」するために使用可能な、減少された（ｒｅｄｕｃｅｄ）情報を受信する。各々の識別子をなんらかの適切な方法で割り当てることができる。例えば、識別子はランダムに割り当てられてもよいし、又は、アーティファクトがデータベース１２０に追加された順に割り当てられてもよい。データベース１２０が複数のファイルのバージョンを格納した場合、それぞれのバージョンは固有の識別子を有することとなる。

【００２９】

バンドル２１４に関連付けられている暗号化キーは、アーティファクトの暗号化されたバージョンの復号化に使用されるキーである。多くの暗号化様式が知られており、任意の適切な暗号化様式を使用することができる。記載の実施形態では、アーティファクトの暗号化に使用されるキーがアーティファクトの復号化に使用されるキーと同一である対称暗号化アルゴリズムが使用される。記載の実施形態では、少なくとも６４ビットのキーを有する暗号化アルゴリズムが使用される。適切なアルゴリズムの例はＡＥＳ１２８及びＡＥＳ２５６暗号化アルゴリズムである。それぞれのアーティファクトは固有の暗号化キーを保持していてもよい。データベース１２０がファイルの複数のバージョンを格納している場合、それぞれのバージョンは固有の暗号化キーを有することができる。

【００３０】

バンドル２１４と関連付けられているエラーチェックコードは、セキュリティのさらなる対策（ｍｅａｓｕｒｅ）を提供する。エラーチェックコードは、セントラルサイト１１０でアーティファクトへの操作が実行されることによって生成される。エラーチェックコードに対して生成される値は、アーティファクトを表すファイルの内容に依存する。リモートクライアントワークステーション１５６は、受信したアーティファクトに同じ操作を実行してもよい。バンドル２１４に関連付けられたエラーチェックコードがリモートクライアントワークステーション１５６によって生成されたエラーチェックコードと一致しない場合には、リモートクライアントワークステーション１５６は、改ざんの結果としてファイルが破損又は変更されたと同定することができる。記載の実施形態においては、エラーチェックコードはハッシュアルゴリズムを通じて生成される。使用できるハッシュアルゴリズムの一例はＳＨＡ１ハッシュアルゴリズムであるが、エラーチェックコードを生成

する任意の適切な方法を使用することができる。

【００３１】

一旦、バンドル２１４がリモートクライアントワークステーション１５６で受信されると、リモートクライアントワークステーション１５６は暗号化されたアーティファクトのコピーに対する要求２１６を生成することができる。実施形態の例では、バンドル２１４の一部として提供される識別子は、アーティファクトのページアドレスとして供給される。リモートサイト１１２がインターネット１１４上でリモートサイト１１０と接続されている場合の例では、インターネット上で従来から使用されていたように、通信はＨＴＴＰメッセージ形式とすることができる。要求２１６は、ＨＴＴＰ　ＧＥＴ要求とすることができる。この例では、バンドル２１４で送信される識別子は、アーティファクトを含むファイルに対するＵＲＬの一部であってもよい。従って、要求２１６は「HTTP://server/identifier.」という形式とすることができる。「HTTP://server」として表されるＵＲＬの一部は、サーバ１２２に対するｗｅｂアドレスを識別する。「identifier」として識別されるＵＲＬの一部は、データベース１２０に格納されているファイルのような、サーバ１２２がアクセスできる特定のファイルを表している。

【００３２】

図１に示す情報管理システムがオペレーションを開始した時には、プロキシサーバ１５０はキャッシュにアーティファクトを含んでいない。このシナリオは図２Ａに示されている。従って、要求２１６は要求２１８としてプロキシサーバ１５０を通過する。

【００３３】

要求２１８は、インターネット１１４を通過してリバースプロキシサーバ１４０に送信される。図１の情報管理システムがオペレーションを開始した時には、リバースプロキシサーバ１４０もアーティファクトに関する情報を含んでいない。要求２１８は、要求２２０としてリバースプロキシサーバ１４０を通過する。

【００３４】

要求２２０は広域ネットワーク１２４を通過してサーバ１２２に到る。サーバ１２２は、リモートクライアントワークステーション１５６によって送信された要求の中の識別子を用いて、データベース１２０中の特定のアーティファクトを識別する。サーバ１２２は、データベース１２０からアーティファクトを取得する。アーティファクトは、暗号化アルゴリズムを実行するためにプログラムされた任意の適切なハードウェアの中で暗号化できる。このシナリオにおいては、サーバ１２２はバンドル２１４と共に送信されたキーを用いてアーティファクトを暗号化する。しかしながら、サーバ１２２が暗号化を実行する必要はない。別の例として、データベース１２０はそれぞれのアーティファクトの暗号化されたバージョンを格納し、サーバ１２２は、要求されたアーティファクトの暗号化されたバージョンを単に取得してもよい。

【００３５】

どのようにアーティファクトが格納され暗号化されたかに関わらず、サーバ１２２は、リモートクライアントワークステーション１５６により起動されたＧＥＴ要求への応答２２２を準備している。リモートクライアントワークステーション１５６とサーバ１２２との間の通信がＨＴＴＰプロトコルを使用したインターネット１１４上で行われる例では、ＨＴＴＰプロトコルによって規定された形式によれば、暗号化されたアーティファクトは応答２２２に含まれている。

【００３６】

応答２２２は、まずリバースプロキシサーバ１４０に渡される。リバースプロキシサーバ１４０は、ここでのポリシーに従って、応答２２２に含まれている暗号化されたアーティファクトのコピーをキャッシュすることができる。暗号化されたアーティファクトは、サーバ１２２からアーティファクトを要求するために使用されるＵＲＬにより索引がつけられたリバースプロキシサーバ１４０に関連付けられているキャッシュに格納することができる。後続の同じアーティファクトに対するどの要求も、ｇｅｔ要求においては同じＵＲＬを使用するであろう。従って、暗号化されていないアーティファクトがリバースプロ

キシサーバ１４０にキャッシュされている間、リバースプロキシサーバ１４０はこのアーティファクトに対する後続の要求を認識しかつ応答することができる。

【００３７】

暗号化されたアーティファクトを含む応答２２４は、インターネット１１４を介してリバースプロキシサーバ１４０からプロキシサーバ１５０に送信される。プロキシサーバ１５０もまた、そのポリシーに従って、暗号化されたアーティファクトを自サーバのキャッシュに格納することができる。暗号化されたアーティファクトは、アーティファクトを要求するために使用されるＵＲＬにより索引がつけられたプロキシサーバ１５０に関連付けられているキャシュに同様に格納することができる。もし、リモートクライアントワークステーション１５６が、プロキシサーバ１５０に関連付けられているキャッシュに格納されている間に同じアーティファクトのコピーを続いて要求した場合、プロキシサーバ１５０は自サーバのキャッシュから暗号化されたアーティファクトのコピーを提供することによって応答することができる。

【００３８】

暗号化されたアーティファクトは、応答２２６に含まれてプロキシサーバ１５０からリモートクライアントワークステーション１５６へ送信される。リモートクライアントワークステーション１５６は、バンドル２１４に含まれる暗号化キーを使用する復号化ソフトウェアによってプログラミングされていてもよい。このため、応答２２６に含まれる暗号化されたアーティファクトのコピーは、リモートクライアントワークステーション１５６において復号化できる。バンドル２１４で送信されたエラーチェックコードは、復号化されたアーティファクトのコピーに適用され、アーティファクトが適切に送信されたことを検証する。

【００３９】

リモートクライアントワークステーション１５６で実行されているソフトウェアプログラムは、リモートクライアントワークステーション１５６が、要求されたアーティファクトの妥当なコピーを自装置が受信したことを判断すると、肯定応答（ａｃｋｎｏｗｌｅｄｇｅｍｅｎｔ）２３０を送信することができる。本実施形態においては、肯定応答２３０はセキュアチャネル２２８を介して送信される。しかしながら、肯定応答を通信するための任意の適切な手段を使用することができる。

【００４０】

サーバ１２２は、肯定応答２３０を利用して、リモートクライアントワークステーション１５６に通信されるアーティファクトの数を減少させることができる。例えば、リモートクライアントワークステーション１５６が関連のあるグループファイルを要求した場合、サーバ１２２は、以前の肯定応答からの情報を使用して、グループ内のファイルの一部がリモートクライアントワークステーション１５６に既に提供されたかを決定することができる。従って、サーバ１２２は、グループファイルに対する要求への適切な応答によって、グループ内の全てのファイルよりも少ないファイルの送信が要求されていることを判断することができる。しかしながら、肯定応答２３０は全ての実施態様に含まれていなくてもよい。

【００４１】

図２Ｂは、起こりうる別の情報交換を図示している。図２Ａで図示した情報交換が起こると、リモートクライアントワークステーション１５６により送信された要求２５２と共に対話が始まる。要求２５２はセキュアチャネル２５０を介して送信される。サーバ１２２は要求２５２に対してバンドル２５４で応答する。バンドル２５４は、要求２５２において識別されるファイル又は複数のファイルに対する識別子を含むことができる。要求２５４はまた、それぞれの要求されたファイルに関連付けられている暗号化キー及びエラーチェックコードを含むことができる。この情報はセキュアチャネル２５０を介してリモートクライアントワークステーション１５６から返信される。

【００４２】

リモートクライアントワークステーション１５６は、バンドル２５４に含まれる識別子

を使用して、アンセキュアなチャネルを介してアーティファクトを要求するための要求２５６を発行する。要求２５６はプロキシサーバ１５０に送信される。図２Ｂに示される情報交換では、プロキシサーバ１５０は既に要求されたアーティファクトをキャッシュしている。アーティファクトのコピーは、以前のリモートクライアントワークステーション１５６との対話の結果として、又は、プロキシサーバ１５０を介して接続されているリモートサイト１１３に存在する他のいずれかのワークステーションとの対話の結果として、キャッシュされていてもよい。

【００４３】

プロキシサーバ１５０は、サーバ１２２への要求を生成することなく、暗号化形式で、要求されたアーティファクトを提供する。プロキシサーバ１５０は、暗号化形式のアーティファクトのコピーを含む応答２５８を生成する。

【００４４】

図２Ａに関連して上述したように、リモートクライアントワークステーション１５６は、バンドル２５４に含まれている暗号化キーを使用してアーティファクトを暗号化する。そして、リモートクライアントワークステーション１５６は、バンドル２５４に含まれているエラーチェックコードを適用して、自装置が、改ざんされていない要求されたアーティファクトのコピーを正確に受信したことを検証する。応答として、リモートクライアントワークステーション１５６は肯定応答２６２を生成することができる。この例では、肯定応答２６２はセキュアチャネル２６０を使用してサーバ１２２に送信される。このようにして、サーバ１２２が、要求されたアーティファクトのコピーを直接リモートクライアントワークステーション１５６に供給しないにもかかわらず、サーバ１２２はリモートクライアントワークステーション１５６が要求されたアーティファクトのコピーを有していることを確かめることができる。

【００４５】

図２Ｂに示されているこのシナリオは、リモートサイト１１２がより大きい企業の開発部門である場合に起きる典型的な情報交換であり得る。リモートサイト１１２は、特定製品のための開発下にあるソースコードファイルの最新バージョンにアクセスする複数の開発者全てを含むことができる。従って、それぞれの開発者は、毎日ソースファイル各々のコピーがロードされるリモートクライアントワークステーション１５６のようなリモートクライアントワークステーションを使用することができる。ファイルを送信するためにＨＴＴＰ等のアンセキュアなプロトコルを使用することによって、特定のファイルがリモートクライアントワークステーションに送信される時及びそれらのコピーをキャッシュする時に、プロキシサーバ１５０はそれらのファイルを識別することができる。たとえプロキシサーバ１５０がアンセキュアなサーバであったとしても、アーティファクトは暗号化されているため、アーティファクトに対する不正なアクセスでアーティファクトについての情報が漏洩することはない。しかし、プロキシサーバ１５０はアーティファクトが送信されたことを認識できるため、プロキシサーバ１５０はそれらのアーティファクトをキャッシュし、アーティファクトに対する後続の要求に対して応答する。このようにして、インターネット１１４上、又は、他のリモートサイト１１２とセントラルサイト１１０との間のなんらかの他の接続で送信された情報の量をかなり減少させることができる。

【００４６】

図２Ｃは、リモートクライアントワークステーション１５６から送信されたアーティファクトに対する要求に応答して発生する可能性のある別の処理である。この図において、要求２７２はリモートクライアントワークステーション１５６から送信される。要求２７２は、インターネット１１４を通じて形成できるセキュアチャネル２７０を介して送信される。要求２７２は、直接サーバ１２２に送信され、リモートクライアントワークステーション１５６に提供すべき１又は複数のアーティファクトを識別する。

【００４７】

サーバ１２２は、バンドル２７４を送信することによって要求２７２に応答する。バンドル２７４は、リモートクライアントワークステーション１５６が適切なアーティファク

トに対する要求を形成することができる識別子を含む。また、バンドル２７４は、暗号化キー及びエラーチェックコードなどの、アーティファクトに関係する他の情報も含む。

【００４８】

リモートクライアントワークステーション１５６は、バンドル２７４に含まれているアーティファクトについての情報を使用して、要求２７６を生成する。要求２７６は、バンドル２７４中の識別子を使用して、アーティファクトに対する要求をフォーマットする。要求２７６は、ここではプロキシサーバ１５０を通過するように示されている。

【００４９】

この例では、プロキシサーバ１５０は、要求されたアーティファクトのコピーを自サーバのキャッシュに保持していない。従って、要求２７８はプロキシサーバ１５０から生成される。要求２７８はインターネット１１４を介してリバースプロキシサーバ１４０へ送信される。

【００５０】

図２Ｃで図示した例では、リバースプロキシサーバ１４０は、要求されたアーティファクトのコピーを自サーバのキャッシュに格納している。従って、リバースプロキシサーバ１４０は、要求２７８に応答して応答２８０を生成する。応答２８０は、要求されたアーティファクトのコピーを含む。このアーティファクトは任意の適切なプロトコルにおいて送信できるが、ここで記載される実施形態においてはＨＴＴＰプロトコルが使用される。アーティファクトは暗号化形式で送信される。

【００５１】

応答２８０はインターネット１１４を介してプロキシサーバ１５０に送信される。プロキシサーバ１５０は、要求されたアーティファクトを自サーバのキャッシュに保持していないため、プロキシサーバ１５０は応答２８０の中のアーティファクトのコピーを格納する。アーティファクトは、要求２７６において使用されたＵＲＬによって索引がつけられたプロキシサーバ１５０に関連付けられているキャシュに格納できる。要求２７６で使用されるＵＲＬは、実際のファイル名というよりはむしろ、バンドル２７４から提供された識別子を使用する。

【００５２】

プロキシサーバ１５０は、暗号化されたアーティファクトのコピーを応答２８２の一部として転送する。リモートクライアントワークステーション１５６は、応答２８２を受信する。リモートクライアントワークステーション１５６は、バンドル２７４の一部として提供された暗号化キーを使用することによって、応答２８２に含まれている暗号化されたアーティファクトのコピーを復号化することができる。リモートクライアントワークステーション１５６は、また、復号化されたファイルにエラーチェックコードを適用して、ファイルがリモートサイト１１２とセントラルサイト１１０との間のアンセキュアなネットワーク接続部分を送信されている間に破損され又は改ざんされていないことを判断することもできる。

【００５３】

次に、リモートクライアントワークステーション１５６は、肯定応答２８６をサーバ１２２に送信する。この例では、肯定応答２８６はセキュアチャネル２８４上を転送される。

【００５４】

プロキシサーバ１５０及びリバースプロキシサーバ１４０は必要とされていないが（ｎｏｔ　Ｒｅｑｕｉｒｅｄ）、図２Ｃは、リバースプロキシサーバ１４０を、企業における情報管理システムの一部として含むことの利点を示している。リバースプロキシサーバ１４０は広域ネットワーク１２４を介して送信される情報量を減少させている。これは、また、サーバ１２２がデータベース１２０からのアーティファクトを暗号化するのに費やす時間をも減少させている。

【００５５】

図３は、情報管理システムが実行することのできる処理を図示している。処理は、クラ

イアントがサーバとのセキュアなコネクションを起動するブロック３１０において開始される。クライアントは、図１で図示した１５６のようなリモートクライアントワークステーションであってもよい。しかしながら、同じ処理は、セントラルサイトに位置している126_1、126_2、126_3、126_4のようなクライアントで使用できる。

【００５６】

ブロック３１２においては、クライアントは１又は複数のサーバを要求する。ブロック３１２において送信される要求は、１又は複数のファイルで識別することができる。要求はセキュアなチャネルを介して送信されるため、たとえ名称又は何らかの識別形式が、企業内で安全を維持するのに好ましいファイルについての情報を表しているとしても、要求は、名称又は何らかの識別形式によってそれぞれのファイルを識別することができる。１又は複数のファイルを要求するために任意の適切なフォーマットを用いることができる。例えば、複数のファイルに対する要求を、個別のファイルに対する要求の連続として形成（ｆｏｒｍａｔ）することができる。しかしながら、任意の適切な形式（ｆｏｒｍａｔ）を使用することができる。

【００５７】

ブロック３１４においては、サーバは、要求されたファイルに関する情報を提供するバンドルで応答する。バンドルは、アンセキュアなチャネル上において安全な方式でファイルへのアクセスを要求する情報を含む。この例では、バンドルは、ファイルを要求するためのネットワークアドレスを形成するために使用できる、それぞれのファイルに対する識別子を含む。バンドルはまた、それぞれのファイルに対する暗号化キーを含む。望ましくは、暗号化キーはそれぞれのファイルで異なっている。それぞれのファイルに別の暗号化キーを使用することによって、たとえ１つの暗号化キーが危険にさらされた（ｃｏｍｐｒｏｍｉｓｅ）としても、情報管理システムに格納されている全情報のうち比較的少ない割合のみが危険にさらされることを保障する。バンドルは、さらに、要求されたファイルについての他の情報を含むことができる。上述した例においては、追加情報がエラーチェックコードに含まれているため、通信エラー又はファイルの改ざんを識別することが可能である。バンドルに含むことができる他の情報は、ファイルのサイズ、ファイルがデータベース１２０に格納された日、又は、クライアントにとってファイルを要求するのに有用な他の情報があり得る。もし、バンドルが複数のファイルに関する情報を提供する場合、何らかの適切な形式でその情報を提供することができる。例えば、それぞれのファイルに対して１セットの状態で、データセットのストリームとして形成して、その情報を提供することが可能である。

【００５８】

ブロック３１６においては、クライアントはバンドルで提供された情報を使用して、アンセキュアな通信チャネルを介して１又は複数のファイルを要求する。

【００５９】

判定ブロック３１８においては、クライアントがアクセスするプロキシサーバから、ファイルが利用可能であるか否かの判定がなされる。もし利用可能である場合には、処理は、プロキシがファイルを提供するブロック３２０に進む。ファイルは暗号化された形式で提供される。

【００６０】

もし、判定ブロック３１８において、ファイルが場所的に（ｌｏｃａｌｌｙ）クライアントに利用可能でないと判定された場合には、処理はブロック３３０に進む。ブロック３３０では、プロキシはセントラルロケーションにファイル要求を送信する。

【００６１】

判定ブロック３２２においては、セントラルロケーションにあるリザーブプロキシは、要求されたファイルのコピーを自装置がキャッシュしているかを判定する。もしキャッシュしている場合には、処理はリザーブプロキシがファイルを提供するブロック３３８に進む。

【００６２】

リザーブプロキシがファイルをキャッシュしていない場合には、処理はブロック３３４に進む。ブロック３３４においては、データベース管理サーバは、要求されたファイルを取得する。暗号化は任意の適切なコンピュータプロセッサにおいて実行され、このコンピュータプロセッサはサーバ１２２であってもよいが、他のサーバ又はコンピュータが暗号化実行のために使用されてもよい。処理ブロック３３６においては、ファイルが暗号化される。

【００６３】

処理はブロック３３８に継続して進む。リザーブプロキシサーバが、そのキャッシュからファイルを取得するか、セントラルロケーションにあるサーバによって提供されたファイルのバージョンを暗号化するかに関わらず、ブロック３３８では、リザーブプロキシサーバはクライアントにファイルを提供する。

【００６４】

ファイルがクライアントに提供されると、処理はブロック３４０に継続して進む。暗号化されたファイルがリザーブプロキシによって提供されようと、クライアントの近くのプロキシから提供されようと、処理はブロック３４０へと進む。ブロック３４０においては、リモートクライアントはファイルを復号化する。そして、復号化ファイルをリモートクライアントで動作しているアプリケーションに提供することができる。

【００６５】

図３により示した処理は、任意の適切な方法で実行できる。例えば、ファイル管理システムとのやりとりを制御するリモートクライアントワークステーション上のソフトウェアを、プロトコルスタックのアプリケーションレイヤーにおけるソフトウェアとして使用することができる。

【００６６】

ファイルがＨＴＴＰなどの標準的なプロトコルを用いて送信される実施形態においては、プロキシサーバ及びリザーブプロキシサーバ１４０は、何らかの現在知られている又はこれから開発されるアプリケーションにおいてプロキシサーバのために使用される等の、従来のハードウェア及びソフトウェア要素であることができる。同様に、サーバ１２２及びデータベース１２０は、現在知られている又はこれから開発される慣習的なサーバとデータベースハードウェアとソフトウェアアクセスとを使用して実装できる。サーバ１２２又はリザーブプロキシサーバ１４０は、ソフトウェアでプログラムを行うことで、リモートクライアントによって発行された要求に応答してファイルを暗号化しバンドルを提供することができる。このようなソフトウェアは、セントラルサイトにあるサーバ、又は、セントラルサイトにアクセス可能な、どのような使い勝手がよいハードウェア及びソフトウェアに組み込まれてもよい。このようなプログラムは、例えば、プロトコルスタックのアプリケーションレベルに組み込まれてもよい。

【００６７】

様々な別の実施形態が可能である。例えば、アーティファクトは、アンセキュアなネットワークを介してコード識別子を用いて要求されると説明したが、これはアンセキュアなネットワークにおいて危険にさらされる何れのアーティファクトについての情報量をも減らすことが可能である。名前又はアーティファクトに対する他の識別子を使用しても、望ましくない情報量を開示することがない場合、要求にコード識別子を利用することは必要でない。

【００６８】

他の例として、暗号化ファイルの復号化は、リモートクライアントワークステーションで行われるとして説明した。その復号化処理は、どのような適切なプロセッサにおいて実行されてもよい。プロキシサーバ１５０への不正なアクセスが懸念されない場合、プロキシサーバ１５０は復号化を実行し、復号化されたアーティファクトのコピーを自サーバのキャッシュに格納してもよい。代替として、リモートサイト１１２にある別個のプロセッサをアーティファクトの復号化の実行に使用してもよい。

【００６９】

同様に、暗号化が起こる時間と場所も変形可能である。例えば、暗号化されたアーティファクトはデータベース１２０に格納されてもよい。このような実施形態においては、サーバ１２２は、クライアントワークステーション126_1、・・・、126_4又はセキュアなネットワークを介してサーバ１２２に接続されている他のプロセッサにファイルを提供する前に、そのファイルを復号化することができる。代替として、クライアントワークステーション126_1、・・・、126_4は、使用する前に、暗号化されたアーティファクトを受信し復号化してもよい。例えば、リモートサイトに配信される情報量がセントラルサイト１１０で使用される情報量に比較して多い場合に、このように処理負荷を再配分することが望ましい。それに関連して、「セントラル」及び「リモート」は、アーティファクトを格納するデータベースと、使用するアーティファクトを受信するプロセッサとの間のネットワーク接続の性質を示す用語である。データベース１２０は、情報管理システムを使用する企業のセントラルの位置に格納される必要はない。

【００７０】

従って、本発明の少なくとも１つの実施形態で説明され、いくつかの態様において説明したように、様々な代替、変更及び改良は、当業者が容易に想到するであろうことは理解される。

【００７１】

このような代替、変更及び改良はこの開示の一部を意図しており、発明の精神及び範囲内であることを意図している。従って、上記の記述及び図面はほんの一例である。

【００７２】

上述した本発明の諸実施形態は、数々の方法のうちいずれを使用しても実施することが可能である。例えば、諸実施形態は、ハードウェア、ソフトウェア、又はこれらの組み合わせを使用して実現することが可能である。ソフトウェアによって実現する場合、プロセッサが単一のコンピュータで提供されるとしても又は複数のコンピュータの中に分布するとしても、ソフトウェアコードは任意の適切なプロセッサ又は一群のプロセッサの上で実行可能である。

【００７３】

また、ここで概説した様々な方法又は処理は、様々なオペレーティングシステム又はプラットホームのいずかを使用する１又は複数のプロセッサ上で実行可能なソフトウェアでコーディングすることができる。さらに、このようなソフトウェアは、適切なプログラミング言語、及び／又は、従来のプログラミングツール又はスクリプトツールのいずかを使用して記述することができ、また実行可能なマシン言語のコードとしてコンパイルすることもできる。

【００７４】

この点において、発明は、１又は複数のプログラムでエンコードされたコンピュータ読取可能な媒体（又は複数のコンピュータ読取可能な媒体）（例えば、コンピュータメモリ、１又は複数のフロッピー（登録商標）ディスク、コンパクトディスク、光ディスク、磁気テープ等）として実現することができる。このプログラムは、１又は複数のコンピュータ又は他のプロセッサ上で実行（ｅｘｅｃｕｔｅ）された場合に、上述した発明の様々な実施形態を提供する方法を実行する（ｐｅｒｆｏｒｍ）。コンピュータ読取可能な記録媒体は運搬可能であり、プログラム及びその記録媒体に記録されているプログラムは、上述した発明の様々な特徴を実現するために、１又は複数の異なるコンピュータ又はプロセッサにロード可能である。

【００７５】

用語「プログラム」は、ここでは、上述した本発明の様々な特徴を実現するために、コンピュータ又は他のプロセッサをプログラムするために使用することが可能な、任意の種類のコンピュータコード又は命令セットを指す一般的な意味で使用されている。

【００７６】

さらに、本実施形態の１つの態様によれば、本発明の様々な態様を実現するために、本発明の方法を実行する1又は複数のコンピュータプログラムは単一のコンピュータプロセ

ッサに存在する必要はなく、多数の異なるコンピュータ又はプロセッサの間でモジュールの形で配分されてもよいことも理解されるべきである。

【００７７】

本発明の様々な態様は、単独で使用されてもよいし、組み合わせで使用されてもよいし、上述した実施形態で具体的に論じなかった様々なアレンジを加えて使用してよく、従って、そのアプリケーションにおいて、上記で説明された又は図で示された要素の詳細及びアレンジに限定されることはない。例えば、１つの実施形態で説明した態様は、他の実施形態で説明した態様と任意の方法によって組み合わせることができる。

【００７８】

また、ステップのタイミング及び順序は変形可能である。例えば、図２Ａ、・・・、２Ｃに示したやり取りは、リモートクライアントが要求を発行し、特定のアーティファクトに関連する識別子及び暗号化キーを受信することで開始した。単一のアーティファクトに対する識別子及び暗号化キーは、変化する可能性がある。もしそうであれば、それぞれのリモートクライアントは、自装置がファイルを要求する度に識別子及び暗号化キーを要求することが必要となる可能性がある。しかしながら、クライアントワークステーションがファイルに対する識別子及び暗号化キーのコピーを格納し、以前に取得して格納しておいた暗号化キーを使用して、２１６、２５６及び２７６といった要求を生成することができる。

【００７９】

「第１の」、「第２の」、「第３の」などの、クレーム要素を変更するためにクレームにおいて順序用語を使用することは、それ自体は優先度、優位性、クレーム要素の他に対する順序、又は、方法の動作が実行される一時的な順序を暗示するものではなく、単に、クレーム要素を区別するために、ある名前を有する１つのクレーム要素を同じ名前（序数用語の使用を除いて）を有する他の要素と区別するためのラベルとして使用される。

【００８０】

また、ここで使用される表現及び専門用語は説明を目的としており、限定的に考えるべきでない。「含有する（ｉｎｃｌｕｄｉｎｇ）」、「構成する（ｃｏｍｐｒｉｓｉｎｇ）」、「有する（ｈａｖｉｎｇ）」、「含む（ｃｏｎｔａｉｎｉｎｇ）」、「含む（ｉｎｖｏｌｖｉｎｇ）」の使用、及び、そのバリエーションは、その後でリストされる項目、及び、追加の項目のみならず同等物をも含むことを意味している。

【図面の簡単な説明】

【００８１】

【図１】本発明の実施形態に係る情報管理システムの構造を示す概略図である。

【図２Ａ】図１の情報管理システムにおいて示された装置間の通信を示す概略図である。

【図２Ｂ】別の動作状態に従った図１の情報管理システムにおける装置間の通信を示す概略図である。

【図２Ｃ】別の動作状態に従った図１の情報管理システムにおける装置間の通信を示す概略図である。

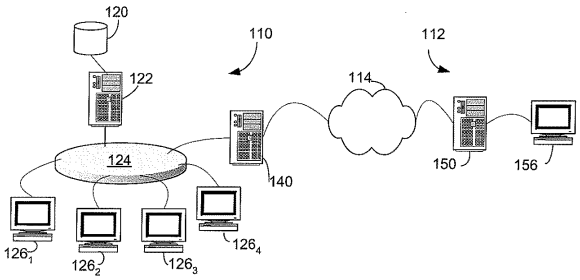
【図３】本発明に係る情報管理システムにおける情報処理を示すフローチャートである。

【符号の説明】

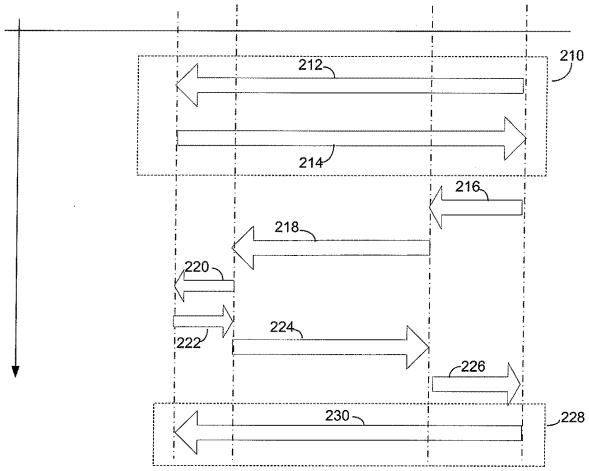
【００８２】

１１０　セントラルサイト
１１２　リモートサイト
１１４　インターネット
１２４　広域ネットワーク
126_1、126_2、126_3、126_4　クライアントワークステーション
１４０　リバースプロキシサーバ
１５０　プロキシサーバ
１５６　リモートクライアントワークステーション

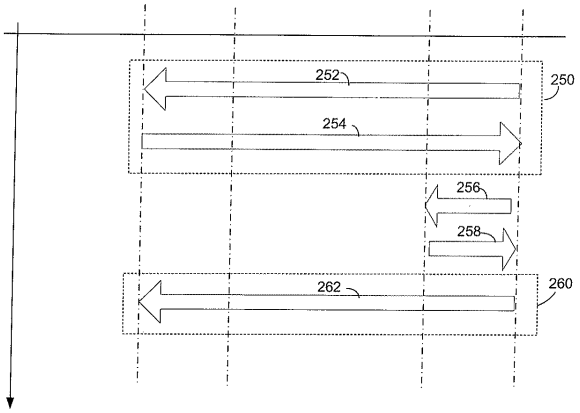
【図１】



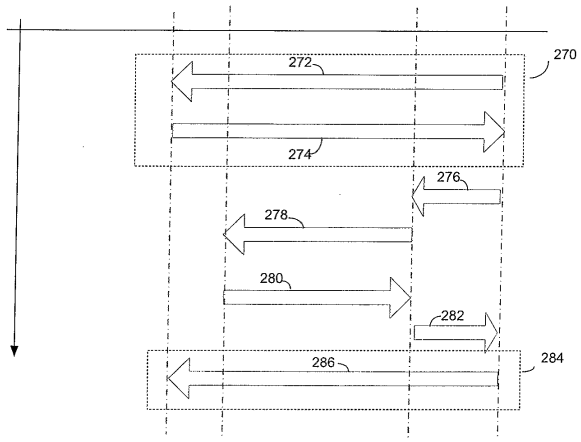
【図２Ａ】



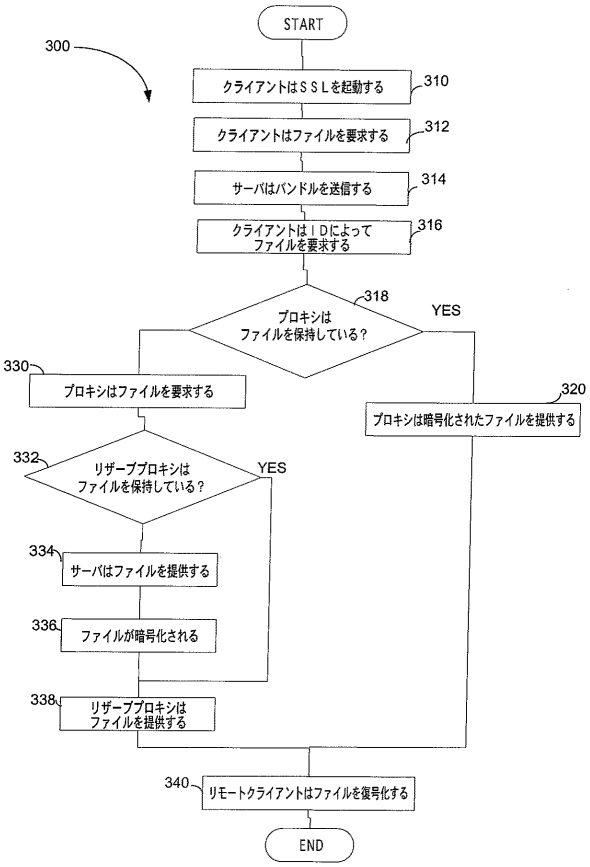
【図２Ｂ】



【図２Ｃ】



【図３】



フロントページの続き

(72)発明者　ジョセフ　クリスティー
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　ワン　マイクロソフト　ウェイ　マイクロソフト　コーポレーション内

審査官　青木　重徳

(56)参考文献　米国特許出願公開第２００３／０１６３５６９（ＵＳ，Ａ１）
特開２００４－０３８４３９（ＪＰ，Ａ）
国際公開第０３／０６７８０１（WO，Ａ１）
特開２００３－０６９６３９（ＪＰ，Ａ）
特開２００３－１２２６９４（ＪＰ，Ａ）
米国特許出願公開第２００４／００９８４６３（ＵＳ，Ａ１）
米国特許出願公開第２００３／００９３６９４（ＵＳ，Ａ１）
国際公開第０２／０９９７１６（WO，Ａ１）
Larry J. Hughes, Jr. 著／長原宏治　監訳，“インターネットセキュリティ”，日本，株式会社インプレス，１９９７年　2月２１日，初版，ｐ．８６－１０２
Jian Zhang, Yunzhang Pei, Dong xie, A Flexible Content Protection System for Media-on-Demand, Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering, ２００２年

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／３２
Ｇ０６Ｆ　１２／００