(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5334320号
(P5334320)
(45)発行日　平成25年11月6日(2013.11.6)　(24)登録日　平成25年8月9日(2013.8.9)

(51)Int.Cl.　　F I
H04L 9/32 (2006.01)　H04L 9/00 675A
G09C 1/00 (2006.01)　G09C 1/00 640E
G06F 21/41 (2013.01)　G06F 21/20 141

請求項の数 17　(全 17 頁)

(21)出願番号　特願2009-539518 (P2009-539518)
(86)(22)出願日　平成19年11月30日 (2007.11.30)
(65)公表番号　特表2010-512069 (P2010-512069A)
(43)公表日　平成22年4月15日 (2010.4.15)
(86)国際出願番号　PCT/US2007/086122
(87)国際公開番号　WO2008/127447
(87)国際公開日　平成20年10月23日 (2008.10.23)
審査請求日　平成22年10月27日 (2010.10.27)
(31)優先権主張番号　11/607,720
(32)優先日　平成18年12月1日 (2006.12.1)
(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　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100101373
弁理士　竹内　茂雄
(74)代理人　100118902
弁理士　山本　修
(74)代理人　100153028
弁理士　上田　忠

最終頁に続く

(54)【発明の名称】暗号証拠の再検証に基づく認証委任

(57)【特許請求の範囲】
【請求項１】
ゲートウェイを介してサービスプロバイダーにアクセスするクライアント／ユーザー間の認証委任の方法であって、
前記クライアント／ユーザーと前記ゲートウェイ間のクライアント認証を有するＴＬＳハンドシェイクを実行するステップであって、前記ＴＬＳハンドシェイクは、複数のメッセージ交換を特定するプロトコルによって定義される、ステップと、
前記クライアント／ユーザーが前記ゲートウェイに認証されたことを証明する認証証拠として、前記ＴＬＳハンドシェイクにおいて交換されたメッセージを記録するステップであって、前記ＴＬＳハンドシェイクにおいて交換されたメッセージは、証明書検証メッセージまでの、前記プロトコルにおいて特定されたすべてのメッセージを含み、前記証明書検証メッセージは、前記ＴＬＳハンドシェイクの以前のすべてのメッセージに対する署名から成る、ステップと、
前記記録を前記ゲートウェイから前記サービスプロバイダーに直接提供するステップと
を備え、前記サービスプロバイダーは、前記クライアント／ユーザーと前記ゲートウェイ間の認証に関与しないことを特徴とする方法。
【請求項２】
時間に関連するデータを前記ＴＬＳハンドシェイクのメッセージに組み込む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】

前記クライアント／ユーザーは、前記時間に関連するデータを組み込むことを特徴とする請求項２に記載の方法。
【請求項４】
前記ゲートウェイは、前記時間に関連するデータを組み込むことを特徴とする請求項２に記載の方法。
【請求項５】
前記サービスプロバイダーによって提供されたノンスを、前記ＴＬＳハンドシェイクの一部として、前記ゲートウェイから前記クライアント／ユーザーまでのメッセージに組み込む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６】
前記サービスプロバイダーは、受信したすべての記録のメモリを保持する、および同じ記録が２回以上使用されていないことを確認することを特徴とする請求項１に記載の方法。
【請求項７】
ＴＬＳハンドシェイクを実行する前記ステップは、
クライアント認証を有しない第１のハンドシェイクを実行するステップと、
前記第１のハンドシェイクの実行の成功した完了後、クライアント認証を有する第２のハンドシェイクを実行するステップと
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８】
前記クライアント／ユーザーとゲートウェイ間の前記第２のハンドシェイクは、前記第１のハンドシェイクから得たセッション鍵によって暗号化されることを特徴とする請求項７に記載の方法。
【請求項９】
前記サービスプロバイダーに提供された前記記録は、暗号化されないことを特徴とする請求項８に記載の方法。
【請求項１０】
認証委任を使用して、エンドサーバー上のサービスへのアクセスを許可する方法であって、
前記エンドサーバーが、ユーザーによって要求されたサービスにアクセスするために、要求を受信するステップと、
前記エンドサーバーが、前記ユーザーと前記ゲートウェイ／中間サーバー間で実行されるクライアント認証を有するＴＬＳハンドシェイクにおいて交換されたメッセージの記録を、前記ユーザーが前記ゲートウェイ／中間サーバーに認証されたことを証明する認証証拠として、前記ゲートウェイ／中間サーバーから直接受信するステップであって、前記ＴＬＳハンドシェイクは、複数のメッセージ交換を特定するプロトコルによって定義され、前記ＴＬＳハンドシェイクにおいて交換されたメッセージは、証明書検証メッセージまでの、前記プロトコルにおいて特定されたすべてのメッセージを含み、前記証明書検証メッセージは、前記ＴＬＳハンドシェイクの以前のすべてのメッセージに対する署名から成る、ステップと、
前記エンドサーバーが、前記記録を利用して、前記ＴＬＳハンドシェイクを再検証する、および前記ユーザーのアイデンティティを確認するステップと
を備え、前記エンドサーバーは、前記ユーザーと前記ゲートウェイ／中間サーバー間の認証に関与しないことを特徴とする方法。
【請求項１１】
ゲートウェイを介してサービスプロバイダーにアクセスするクライアント／ユーザー間の認証委任の方法であって、前記ゲートウェイは、
前記クライアント／ユーザーと前記ゲートウェイ間のクライアント認証を有するＴＬＳハンドシェイクを実行するステップであって、前記ＴＬＳハンドシェイクは、複数のメッセージ交換を特定するプロトコルによって定義される、ステップと、

前記クライアント／ユーザーが前記ゲートウェイに認証されたことを証明する認証証拠として、前記ＴＬＳハンドシェイクにおいて交換されたメッセージを記録するステップであって、前記ＴＬＳハンドシェイクにおいて交換されたメッセージは、証明書検証メッセージまでの、前記プロトコルにおいて特定されたすべてのメッセージを含み、前記証明書検証メッセージは、前記ＴＬＳハンドシェイクの以前のすべてのメッセージに対する署名から成る、ステップと、

前記記録を前記ゲートウェイから第三者エンティティに直接提供するステップと、

前記第三者エンティティによる前記記録の有効性の確認後、ユーザー資格を前記第三者エンティティから受信するステップと、

前記ユーザー資格を用いて、前記ユーザーを前記サービスプロバイダーに対して認証するステップと

を備える方法を実行し、

前記サービスプロバイダーは、前記クライアント／ユーザーと前記ゲートウェイ間の認証に関与しないことを特徴とする方法。

【請求項１２】

前記第三者エンティティは、認証局（Ｃｅｒｔｉｆｉｃａｔｅ　Ａｕｔｈｏｒｉｔｙ）であることを特徴とする請求項１１に記載の方法。

【請求項１３】

ユーザー資格を受信する前記ステップは、一時的な証明書およびそれに関連する秘密鍵を受信することを備えることを特徴とする請求項１２に記載の方法。

【請求項１４】

前記ユーザーを前記サービスプロバイダーに対して認証する前記ステップは、ＰＫＩＮＩＴを用いたＫｅｒｂｅｒｏｓを使用して実行されることを特徴とする請求項１２に記載の方法。

【請求項１５】

前記第三者エンティティはＫＤＣであり、ユーザー資格を受信する前記ステップは前記ＫＤＣからＫｅｒｂｅｒｏｓのサービスチケットを受信することを備えることを特徴とする請求項１１に記載の方法。

【請求項１６】

前記第三者エンティティは、単一デバイスのゲートウェイと一緒に常駐することを特徴とする請求項１１に記載の方法。

【請求項１７】

ＴＬＳハンドシェイクを実行する前記ステップは、

クライアント認証を有しない第１のハンドシェイクを実行するステップと、

前記第１のハンドシェイクを実行するステップの成功した完了後、クライアント認証を有する第２のハンドシェイクを実行するステップと

をさらに備えることを特徴とする請求項１１に記載の方法。

【発明の詳細な説明】

【背景技術】

【０００１】

組織は、あるサービスをユーザーに一緒に提供する一連のエンティティを有することができる。例えば、データ、ウェブページ、機能的なソフトウェアオペレーションなどのリソースへのアクセスは、習熟かつ権限を与えられた一定のユーザーに限定する必要がある。権限のないユーザーや悪意のある攻撃者が、ウェブリソースにアクセスしようとしているユーザーのアイデンティティを認証するのに使用される機構を含むコンピューターリソースにアクセスすることを阻止するために、さまざまなアクセス制御方式が開発されている。特に、個人情報盗難の攻撃（例えば、フィッシング、ファーミング）の増加により、２要素認証が一般的になりつつある。

【０００２】

Ｔ－ＦＡ(Two-factor authentication)とは、アイデンティティと権限を確立するため

の異なる２つの方法を必要とする任意の認証プロトコルである。Ｔ－ＦＡの一般的な実装は、要因の一つとして「あなたが知るもの」（例えば、パスワードまたは個人識別番号）を使用し、他の要因として「あなたが有するもの」（例えば、クレジットカードまたはハードウェアトークン）か、あるいは「あなたであるもの」（例えば、指紋または網膜のパターン）を使用する。例えば、スマートカードは、Ｔ－ＦＡを提供する方法の１つである。スマートカードは、ハードウェアトークンの一例であり、典型的には、所与のデータ上の暗号化機能を実行するなどのさまざまなセキュリティ動作が可能であるマイクロプロセッサーを含む。スマートカードは通常、証明書ベースの認証を要求するプロトコルに使用できる１つまたは複数のＩＴＵ－Ｔ(International Telecommunication Union)Ｘ．５０９証明書（およびそれらに関連する秘密鍵）を保持する。ＳＳＬ(Secure Socket Layer)、ＴＬＳ(Transport Layer Security)、およびＫｅｒｂｅｒｏｓ（「Public Key Cryptography for Initial Authentication in Kerberos」の省略である、ＰＫＩＮＩＴを用いる）は、そのようなプロトコルのすべての例である。

【０００３】

証明書を使用するのにスマートカードは必要ない。多くのデバイス(例えば、コンピューター、携帯電話)は、証明書（およびそれらに関連する秘密鍵）を格納し、使用することが可能である。例えば、Ｗｉｎｄｏｗｓ（登録商標） Ｍｏｂｉｌｅ ５．０ は、（ＳＳＬまたはＴＬＳの最上部で起動する Ｅｘｃｈａｎｇｅ ＡｃｔｉｖｅＳｙｎｃｈ プロトコルを経由して）Ｅメールやカレンダー情報を同期化するために、証明書を使用して、Ｅｘｃｈａｎｇｅ ２００３ ＳＰ２ サーバー に対して認証することができる。

【０００４】

例えば、ウェブベースのアクセスを組織内部のネットワーク（イントラネット）上に置かれるウェブサーバーに提供するネットワーク端部のゲートウェイデバイスを含む、組織のネットワークの一連のエンティティに対する個人情報盗難の攻撃を阻止することが、重要である。

【０００５】

認証委任は、クライアントがサーバーに対する認証を委任するか、より具体的には、第三者認証サービス（またはゲートウェイ）が、ユーザーに代わって（実質的には、そのユーザーに成り済ますことによって）リソース（またはサーバー）にアクセスして認証することができる場合として広く定義されている。アクセスされたサーバーは、認証サービスのアカウントに基づいてというよりむしろ、ユーザーのアイデンティティ上に権限付与判定の基礎を置く。

【０００６】

本背景技術は、以下の発明の概要および詳細な説明に対して簡潔なコンテキストを導入するために提供される。本背景技術は、特許請求の範囲に記載された主題の範囲を決定する際の補助となることを意図しておらず、特許請求の範囲に記載された主題を上記に提示された任意のまたはすべての欠点もしくは問題を解決することができるこれらの実装のみに限定すると見なすことも意図していない。

【発明の概要】

【課題を解決するための手段】

【０００７】

再検証または暗号証拠に基づいた認証委任は、ユーザーが一連のエンティティ内の特定のサーバーにアクセスを望むとき、ゲートウェイデバイスとユーザー間のＴＬＳハンドシェイクの少なくとも一部の記録を利用する。暗号証拠は、ＴＬＳハンドシェイクの記録された部分を（１）サーバーが記録された部分を再検証して認証を確認する場合の所望のサーバーか、または（２）第三者エンティティが、記録された部分を再検証し、ユーザー資格をゲートウェイに提供することによって認証を確認し、今度はその資格を使用して、ユーザーとしてサーバーに対して認証する場合の第三者エンティティのいずれかに転送することによって提供される。いずれの場合にも、サーバーおよび第三者エンティティは、ユーザーとゲートウェイデバイス間の認証に関与せずに、ＴＬＳハンドシェイクの記録され

た部分を使用してユーザーアクセスを許可するかどうかの判定をする。

【０００８】

さまざまな説明的な例において、暗号証拠は、ＴＬＳハンドシェイクが時宜にかなう（すなわち、「フレッシュ(fresh)」）ように保証することによって付加的なセキュリティ対策を提供するために、タイムスタンプを含む。加えて、有効なＴＬＳハンドシェイクを確認した後、第三者エンティティは、ゲートウェイが、ユーザーに代わって、例えば、ＰＫＩＮＩＴを用いたＫｅｒｂｅｒｏｓを使用して、所望のサーバーに対して認証できる一時的な（すなわち、期限付き）ユーザー資格を発行するように設定することができる。

【０００９】

本概要は、簡易形式によって概念の選択を導入するために提供される。この概念は、詳細な説明の節においてさらに説明される。本概要において説明されたもの以外の要素またはステップが実現可能であり、要素またはステップは必ずしも必要とされない。本概要は、特許請求の範囲に記載された主題の主要な特徴または本質的な特徴を明らかにすることを意図しておらず、特許請求の範囲に記載された主題の範囲を決定する際の補助としての使用も意図しない。特許請求の範囲に記載された主題は、本開示のいずれの部分に述べた任意またはすべての欠点を解決する実装に限定されない。

【図面の簡単な説明】

【００１０】

【図１】ＴＬＳハンドシェイクメッセージの再検証に基づく認証委任のための説明的なアーキテクチャの簡易化した機能的なブロック図である。

【図２】図１の例示的なアーキテクチャを利用して、認証プロセスのステップを示す説明的なフロー図である。

【図３】ユーザー資格が、ＴＬＳハンドシェイクの少なくとも一部の記録を受信する第三者エンティティによって提供される認証委任のための説明的なアーキテクチャの簡易化した機能的なブロック図である。

【図４】クライアントとＫｅｒｂｅｒｏｓ（バージョン５）プロトコルにおける鍵配布センター間の説明的なメッセージ交換を示す図である。

【図５】図３の例示的なアーキテクチャを利用して、認証プロセスのステップを示す説明的なフロー図である。

【図６】典型的なＴＬＳハンドシェイクフェーズのクライアントとサーバー間のメッセージの交換を示す概略メッセージフロー図である。

【発明を実施するための形態】

【００１１】

再検証または暗号証拠に基づいた本認証委任の説明的なコンテキストは、クライアント／ユーザーがゲートウェイを通じて１つまたは複数のサービスプロバイダーにアクセスするものである。しかしながら、このコンテキストは単に説明的であり、他のコンテキストおよび環境でも適用できることを強調しておく。例えば、本認証委任は、ウェブサーバーがバックエンドアプリケーションまたはデータベースに対してユーザーとして認証する必要があるとき、もしくは代替的に一連のエンティティがあり連鎖するエンティティ間で認証が必要とされる場合の任意の設定において使用することができる。

【００１２】

アクセスコンテキストにおいて、ゲートウェイデバイスは、ユーザーが要求を提出するウェブサーバーへのアクセスを提供し、これらの要求は、ゲートウェイに到達し、最終的には内部のウェブサーバーに到達する。しかしながら、ゲートウェイとウェブサーバーの両方は、典型的には、これらに接続しているユーザーが所望のリソースにアクセスできるかどうかを判定するために、ある認証形式を必要とする。

【００１３】

ゲートウェイを組み込んで、ＦＢＡ(form-based authentication)を使用する場合、ユーザーは、ログオン形式でユーザー名およびパスワードを入力する必要がある。次にユーザーは、この形式を提出し、ゲートウェイは、ユーザーのユーザー名およびパスワードを

受信する。次にゲートウェイは、それらの資格を使用して、ユーザーに代わって内部のウェブサーバーに対して認証することができる。ゲートウェイがパスワードを受信し、要望通りにパスワードを使用することができるため、これは非常に簡単であり実行可能である。しかしながら、ある認証方式と一緒にはこれは実行できない。例えば、ユーザーが、パスワードを提供しない認証方式を使用してゲートウェイに対して認証する場合、ゲートウェイは、ユーザーに代わって内部のウェブサーバーに対して認証するために再使用することができる何らの資格を有さない。

【００１４】

この問題に対するいくつかの解決方法が提案されている。例えば、一解決方法では、「信頼された第三者」が関与する。ここでは、ゲートウェイを「信頼する」のに先立って信頼された第三者を組み込んで、すべてのユーザーに代わって定義されたウェブサーバー（または一般的にサービス）のセットに対して認証する。この技術を、ゲートウェイ（またはフロントエンドサーバー）が、クライアントに代わって他のサーバーと使用するチケットを要求できるプロトコルとして実装することができる。信頼された第三者は、次に、ゲートウェイがその後任意のユーザーに成り済ますことができるように、任意のユーザーに代わってサービスチケットをゲートウェイに進んで提供する。

【００１５】

信頼された第三者を組み込んで、具体的な条件の下にサービスチケットを提供することもできる。例えば、Ｋｅｒｂｅｒｏｓプロトコルにおいて、クライアントは、サービスチケットを介してゲートウェイに対して認証し、Ｋｅｒｂｅｒｏｓに制約された委任は、信頼された第三者（鍵配送センター）を組み込んで、その条件を課すことができる方法を提供する。この場合では、ゲートウェイは、主張したユーザーが実際にゲートウェイに認証（サービスチケットを介して）された証拠を提供しなければならない。これは、システム全体のセキュリティを強化するのに重要である。例えば、このような証拠を要求する利点は、侵害されたゲートウェイは、最初にユーザーがそのゲートウェイに正しく認証されなければユーザーに代わってサーバーにアクセスすることができないことである。

【００１６】

これらの提案がパスワードなしの認証に対処するための方法を提供する一方で、場合によっては、ＫＤＣ(Key Distribution Center)または他の信頼された第三者エンティティと関与しない認証委任モデルを実装するのが望ましいかもしれない。このようなモデルにおいて、ゲートウェイは、ＫＤＣとの通信を全く行わずに、ユーザーに代わって内部のウェブサーバーに対して認証する。この種の機能性を提供する多くの解決方法がある。例えば、一部の製品は、いったんユーザーがゲートウェイに対して認証すると、そのゲートウェイは、内部のウェブサーバーによって信頼されたトークン（ＨＴＴＰ(Hypertext Transfer Protocol)　場合によってはクッキー(cookie)）を戻すことができるように、ゲートウェイおよび任意の数の内部のウェブサーバー上でインストール／組み込むことができる。他の提案と同様に、このモデルに対する１つの問題は、ゲートウェイは完全に信頼されたエンティティであるので、システムの全体のセキュリティを低下させることである。

【００１７】

本配置は、暗号証拠の再検証に基づいた認証委任を提供する。ゲートウェイ（またはフロントエンドサーバー）は、ウェブサーバー（またはバックエンドサーバー）へのアクセスを提供する。クライアント／ユーザーは、クライアント認証を有するＴＬＳハンドシェイクを使用してゲートウェイに対して認証する。ＴＬＳハンドシェイクの記録、または少なくともユーザーがゲートウェイに認証されたことを証明するのに十分なＴＬＳハンドシェイクの記録は、次いで、ウェブサーバー（そのハンドシェイクの有効性を再検証する）か、または第三者エンティティ（その記録を検証した後、ユーザー資格をゲートウェイに提供し、次にゲートウェイは、ウェブサーバーに対して認証する）のいずれかに提供される。

【００１８】

これより同様の参照番号が同様の要素を指す図を参照しながら、図１に本認証委任を利

用する例示的なネットワークアーキテクチャを示す。クライアント／ユーザーコンピューターシステム１０を、ゲートウェイ２０（認証サーバとも呼ばれる）に動作可能につなげることによって、クライアント／ユーザー１０とウェブサーバー３０（ネットワークサーバーとも呼ばれる）のネットワーク間の通信ができるようになる。ゲートウェイ２０は、ユーザーを認証するのに必要な情報を含むデータベース／ディレクトリ（図示せず）を含む（代替的に、ゲートウェイは、外部のユーザーデータベース／ディレクトリを用いてネットワーク上で通信することができる）。ログオンに応答して、ユーザー／クライアント１０は、参照数字３５に示すように、最初にクライアント認証を有するＴＬＳハンドシェイクを経由してゲートウェイ２０に対して認証する。ＴＬＳハンドシェイクプロトコルではクライアント認証が随意的であるため、クライアント認証は、ここで意図的に述べられていることに留意されたい。

【００１９】

ＴＬＳプロトコルは、インターネット上で通信プライバシーを提供し、クライアント／サーバーアプリケーションが、盗聴、改ざん、またはメッセージ偽造を阻止するように設計された方法によって通信することを可能にする。ＴＬＳハンドシェイクプロトコルによって、アプリケーションプロトコルがそのデータの最初のバイトを送信または受信する前に、サーバーとクライアントは、互いに認証し合い、暗号アルゴリズムと暗号鍵を交渉することができる。

【００２０】

ＴＬＳの１つの利点は、アプリケーションプロトコルインディペンデントがあることである。従って上位レベルプロトコルを、透過的にＴＬＳプロトコルの最上層に置くことができる。ＴＬＳハンドシェイクプロトコルを、以下のように要約することができる。すなわち、ユーザー／クライアントは、クライアントのハローメッセージを、サーバーのハローメッセージによって応答しなければならないサーバー（図１のゲートウェイ２０）に送信する。さもなければ、致命的なエラーが起こり、この接続が失敗する。クライアントのハローとサーバーのハローを使用して、クライアントとサーバー間のセキュリティ強化機能を確立する。

【００２１】

図６は、典型的なＴＬＳハンドシェイクフェーズのクライアントとサーバー間のメッセージ交換を示す概略メッセージフロー図である。ＴＬＳプロトコルは、ＲＦＣ２２４６のＴＬＳプロトコル　バージョン１．０において詳細に記載されており、この開示は、参照することによって本明細書に組み込まれる。クライアント／ユーザーは、クライアント－サーバー関係においてゲートウェイデバイスと通信する。

【００２２】

さらに具体的には、図６に示すように、実際の鍵交換は、メッセージを４つまで使用する。すなわち、サーバー証明書、サーバー鍵交換、クライアント証明書、およびクライアント鍵交換である。新しい鍵交換方法は、これらのメッセージに対しフォーマットを指定し、そのメッセージの使用を定義することによって作り出され、これによってクライアントとサーバーは、共有秘密に対して合意できる。ハローメッセージの後、サーバーは、そのメッセージが認証されるべきであるならば、メッセージの証明書を送信する。加えて、要求される場合は（例えば、サーバーが証明書を有しないか、または署名のみに対するものである場合）サーバー鍵交換メッセージを送信することができる。

【００２３】

サーバーが認証された場合、暗号スイートの選択がふさわしければ、サーバーは、クライアントから証明書を要求することができる。次にサーバーは、ハンドシェイクのハローメッセージフェーズが完了したことを表示する、サーバーのハロー完了メッセージを送信する。次にサーバーは、クライアントの応答を待つ。サーバーが、証明書要求メッセージ(certificate request message)を送信した場合、クライアントは、その証明書メッセージを送信しなければならない。クライアント鍵交換メッセージが送信され、そのメッセージ内容は、クライアントハローとサーバーハロー間で選択した公開鍵アルゴリズムに依存

する。クライアントが署名機能を有する証明書を送信した場合、デジタル署名された証明書検証メッセージを送信して、この証明書を明確に検証する。

【００２４】

図１に戻ると、この説明的な例において、ゲートウェイ２０は、このハンドシェイクの一部として交換されたデータの記録（図１では参照番号４５とともにＴＨＲとして示す）を作成する。より詳しくは、この記録は、ＴＬＳハンドシェイクの以前のすべてのメッセージに対する署名から成り、ユーザー／クライアント１０がその証明書と一致する秘密鍵を実際に所有することを証明する、証明書検証メッセージまでのデータを少なくとも含む。

【００２５】

ＴＬＳハンドシェイクの記録、またはＴＨＲは次に、ユーザー／クライアント１０がゲートウェイ２０に認証された認証証拠（すなわち、「プルーフ(proof)」）として、内部のウェブサーバー３０に直接提供される。

【００２６】

内部のウェブサーバー３０は、クライアント／ユーザー１０とゲートウェイ２０間の認証に関与せずに、単にクライアント／ユーザー１０とゲートウェイ２０間の認証の認証証拠（すなわち、ＴＨＲ）を提供されて、次いで所望のリソースへのアクセスを提供するかどうかに関する判定をするにすぎない。

【００２７】

提案された方式は、ＴＬＳハンドシェイクが証明書検証メッセージを含む場合しか使用できないことに留意されたい。このメッセージは、以下のいずれかの条件が真である場合は使用されない。すなわち

１）ＴＬＳハンドシェイクは、クライアント認証を含まない。

２）クライアントとゲートウェイは、（新しいセキュリティパラメータを交渉する代わりに）以前のＴＬＳセッションを再開するまたは現在のセッションを繰り返す決定をする。この場合、ＴＬＳハンドシェイクは、証明書検証メッセージを含まない（ＲＦＣ２２４６の３０－３１ページ参照）。

３）クライアント証明書は、署名機能（すなわち、固定ディフィー・ヘルマンパラメーターを含むものを除くすべての証明書）を有する。例えば、暗号スイート ＥＣＤＨ＿ＥＣＤＳＡとＥＣＤＨ＿ＲＳＡ（ＲＦＣ４４９２参照）は、クライアント認証をサポートするが、証明書検証メッセージは利用しない。

【００２８】

図２は、図１に示した例示的なアーキテクチャにおいて、クライアント／ユーザーが、ウェブサーバーへアクセスしようとするときに実行される認証プロセスの説明的なフロー図である。このプロセスは、クライアント／ユーザーが、ウェブサーバーリソースを所望し、ゲートウェイ２０にアクセスするときに開始する（ステップ２００）。クライアント／ユーザーが、ウェブサーバーにログインしていない場合、そのクライアント／ユーザーは、ウェブサーバーがアクセスを許可する前に、認証されなければならない。

【００２９】

クライアント／ユーザーは次に、所望のウェブサーバーリソースを要求する（ステップ２１０）。クライアント／ユーザーを認証するために、ゲートウェイ２０とクライアント／ユーザー１０は次に、上記に詳しく説明した方法によってクライアント認証を有するＴＬＳハンドシェイクを実行する（ステップ２２０）（当業者は当然、ゲートウェイ２０は、クライアント／ユーザーが所望のリソースを要求する前か、またはクライアント／ユーザーがリソースを要求した後に、即座にクライアント／ユーザー１０を認証できることを理解するだろう）。ＴＬＳハンドシェイクの少なくとも一部の記録が作成されて、要求されたウェブサーバーに提供される（ステップ２３０）。

【００３０】

ＴＨＲの受信後、ウェブサーバー３０は、クライアント／ユーザーが、ゲートウェイに認証されたことを検証する（証明書のクライアント／ユーザーの署名を有効にすることに

よって、ＴＨＲのメッセージおよびある実施形態におけるＴＨＲのタイムスタンプ（以下に論ずる）も検証する）（ステップ２４０）。クライアント／ユーザーがウェブサーバーにアクセスする権限を与えられたと仮定して、ＴＨＲが検証された場合、要求されたウェブサーバーへのアクセスが許可され（ステップ２５０）、ＴＨＲが検証されない場合、そのアクセスは拒否される（ステップ２６０）。

【００３１】

攻撃者がＴＨＲを取得できて、ＴＨＲを再使用してクライアント／ユーザーに成り済まそうとする場合に、このような「リプレイアタック」を阻止するか、または少なくとも抑制するために想定されるいくつかの技術および機構がある。まず、サーバーおよび／またはクライアント／ユーザーが、時間に関連するデータ（例えば、タイムスタンプ）をそれらのハンドシェイクメッセージに組み込んだと仮定して、サービスプロバイダー（例えば、内部のウェブサーバー）は、受信したＴＨＲを調べて、それが「フレッシュ」であることを確認することができる。この解決方法は、典型的には、ゲートウェイ２０およびウェブサーバー３０に（またはユーザー／クライアント１０およびウェブサーバー３０）、同期化されたクロックを有するよう要求する。とはいえ、当業者は当然、多くの実現可能な次善策があることがわかるだろう。

【００３２】

代替的に、ゲートウェイ２０は、サービスプロバイダー（ウェブサーバー３０）にノンスを求め、ＴＬＳハンドシェイクの一部としてユーザー／クライアント１０に送信するメッセージの１つの中にノンスを組み込むことが可能である。サービスプロバイダーは次に、受信したＴＨＲを調べて、それが以前に作成されてゲートウェイ２０を通過したノンスを含むことを確認することができる。

【００３３】

このような２つの可能性のいずれにおいても、ゲートウェイ２０（またはユーザー／クライアント１０）は、あるデータをＴＬＳハンドシェイクメッセージに組み込む（さらに、ハンドシェイクプロトコルは基本的に、データ転送セッションのセキュリティパラメータを交渉する一連の順序付けられたメッセージである）。このようなデータの組み込みは、典型的には、以下の方法の１つによって行われるものとする。すなわち

（１）サーバーは、タイムスタンプまたはノンスをサーバーのハローメッセージ内に、このメッセージのランダムフィールドの一部として、置く（ハンドシェイクプロトコルのこの態様の詳細は、ＲＦＣ２２４６の７．４．１．３節に見出すことができる）

（２）サーバーは、タイムスタンプまたはノンスをサーバーのハロー拡張子に置く（詳細は、ＲＦＣ３５４６の２．２節に見出すことができる）

（３）クライアント／ユーザーは、タイムスタンプをクライアントのハローメッセージ内に、このメッセージのランダムフィールドの一部として、置く（ハンドシェイクプロトコルのこの態様の詳細は、ＲＦＣ２２４６の７．４．１．２節に見出すことができる）

（４）クライアント／ユーザーは、タイムスタンプをクライアントのハロー拡張子内に置く（詳細は、ＲＦＣ３５４６の２．１節に見出すことができる）

【００３４】

最後に、上記の代替案の各々に加えて、サービスプロバイダ（ウェブサーバ３０）は、同じＴＨＲを２回以上使用しないことを保証するために、受信するすべてのＴＨＲを記憶することが可能である。この記憶は、ある共有された記憶装置または通信機構を経由して、サービスプロバイダー間で共有することも可能である。

【００３５】

別の説明的な実装において、ゲートウェイまたはクライアントとゲートウェイ間の通信チャネルを侵害する攻撃者に対するより一層の防御として、「２重の」ＴＬＳハンドシェイクを使用することができる。この場合、クライアント／ユーザー１０およびゲートウェイ２０は、クライアント認証を有しない第１のＴＬＳハンドシェイクを実行する。第１のＴＬＳハンドシェイクがうまく完了したとき、クライアント／ユーザー１０およびゲートウェイ２０は、クライアント認証を有する第２のＴＬＳハンドシェイクを実行する。後に

証拠（ＴＨＲ）として使用される第２のハンドシェイクは、クライアント／ユーザー１０およびゲートウェイ２０が第１のハンドシェイクから得たセッション鍵によって送信のために暗号化される。従ってＴＨＲは、暗号化されずに（例えば、暗号化されていない文章で）送信されないように防御されているので、攻撃者がたとえゲートウェイ２０を侵害できたとしても、ＴＨＲを取得するのはより困難になるだろう。

【００３６】

これまで論じられ、図１に示した実施形態は、ユーザー／クライアント１０、ゲートウェイ２０、およびサービスプロバイダー（ウェブサーバー３０）から成る。図３に示す、以下でより詳細に論じられる代替的実施形態では、第三者エンティティ４０を利用する。この場合、サービスプロバイダー（ウェブサーバー３０）は、第三者エンティティ４０を「信頼」して、ユーザーの実アイデンティティを提供する。このような第三者エンティティ４０を、ＫｅｒｂｅｒｏｓＫＤＣ(Ｓ４Ｕ２Ｓｅｌｆ＋Ｓ４Ｕ２Ｐｒｏｘｙなど)またはＣＡ(Certificate Authority、認証局)にすることができる。第三者エンティティ４０を、個別のエンティティとして図３に示しているが、ある構成においては、第三者エンティティ（ＫＤＣまたはＣＡ）は、ゲートウェイ２０として同じマシン上に常駐することに留意する。

【００３７】

図３に示した実施形態をさらに具体的に論じる前に、ＫＤＣとして知られる信頼された第三者の使用に関与するＫｅｒｂｅｒｏｓプロトコルが、クライアントとサービス間の共有セッション鍵を交渉し、クライアントとサービス間の相互認証を提供することを論ずる。

【００３８】

Ｋｅｒｂｅｒｏｓの基礎は、チケットと認証コードである。チケットは、特定のサービスを目的とするエンベロープ（公開メッセージ）の対称鍵（チケットセッション鍵－２つのエンドポイント間で共有されるただ１つの鍵である）をカプセル化する。チケットの内容は、サービスのプリンシパルと発行するＫＤＣ間で共用される対称鍵を用いて暗号化される。チケットの暗号化された部分は、他の項目の中で、クライアントのプリンシパル名を含む。認証コードは、関連するチケットのチケットセッション鍵を使用して、直近に作成されたことを示すことができる記録である。チケットセッション鍵は、そのチケットを要求したクライアントにより公知である。認証コードの内容は、関連するチケットセッション鍵を用いて暗号化される。認証コードの暗号化された部分は、他の項目の中で、タイムスタンプとクライアントのプリンシパル名を含む。

【００３９】

図４に示したように、Ｋｅｒｂｅｒｏｓ（バージョン５）プロトコルは、以下のクライアント４０５とＫＤＣ４１０間、およびクライアント４０５とアプリケーションサーバー４１５間のメッセージ交換から成る。すなわち

ＡＳ(Authentication Service、認証サービス)交換

クライアントは、典型的にはＴＧＴ(Ticket Granting Ticket、チケット保証チケット)である、ＫｅｒｂｅｒｏｓＡＳ(authentication server)から「最初の」チケットを取得する。ＡＳ－ＲＥＱメッセージ４２０とＡＳ－ＲＥＰ４２５メッセージはそれぞれ、クライアントとＡＳ間の要求メッセージと応答メッセージである。

ＴＧＳ(Ticket Granting Service、チケット保証サービス)交換

続いてクライアントは、認証するためにＴＧＴを使用して、ＫｅｒｂｅｒｏｓＴＧＳ(ticket-granting server)から特定のサービスのサービスチケットを要求する。ＴＧＳ－ＲＥＱメッセージ４３０とＴＧＳ－ＲＥＰ４３５メッセージはそれぞれ、クライアントとＴＧＳ間の要求メッセージと応答メッセージである。

クライアント／サーバーＡＰ(Authentication Protocol、認証プロトコル)交換

次にクライアントは、クライアントのチケットセッション鍵の所有を認証するサービスチケットと認証コードから成るＡＰ－ＲＥＱメッセージ４４０を用いて要求する。サーバーは、随意的に、ＡＰ－ＲＥＰメッセージ４４５を用いて応答することができる。ＡＰ交

換は、典型的には、セッション固有対称鍵を交渉する。

【００４０】

ＡＳおよびＴＧＳは、典型的には、ＫＣＤとしても知られる単一のデバイス内に組み込まれる。

【００４１】

ＡＳ交換において、ＫＤＣ応答は、他の項目の中で、チケットセッション鍵を含み、クライアントとＫＤＣ間で共有される鍵（ＡＳ応答鍵）を使用して暗号化される。ＡＳ応答鍵は、典型的には、人間ユーザー用のクライアントのパスワードから得られる。従って、人間ユーザーにとって、攻撃に対するＫｅｒｂｅｒｏｓプロトコルの抵抗力は、人間ユーザーのパスワードの効力ほど強くない。

【００４２】

データ発信元認証および完全秘密を容易にするために、Ｘ．５０９証明書形式での非対称暗号の使用（「ＩＳＯＣ(Internet Society、インターネット協会)」によって管理された文書シリーズ「Ｒｅｑｕｅｓｔ　ｆｏｒ　Ｃｏｍｍｅｎｔｓ」の項目でＲＦＣ３２８０を参照）が普及している。確立されたＰＫＩ(Public Key Infrastructure、公開鍵基盤)は、認証および安全な通信を確立するために使用することができる鍵管理および鍵分散機構を提供する。公開鍵暗号をＫｅｒｂｅｒｏｓに付加することは、公開鍵プロトコルに上手い合同式を提供し、強力なパスワードを管理する人間ユーザーの負担を取り除き、ケルベライズ(Kerberized)されたアプリケーションが、現存する鍵サービスおよびアイデンティティ管理を利用できるようにする。

【００４３】

ＫｅｒｂｅｒｏｓＴＧＴにより与えられる利点は、クライアントが、長期間の秘密を１回しかさらさないことである。ＴＧＴおよびそれに関連するセッション鍵は次に、後に続く任意のサービスチケット要求に対して使用することができる。１つの結果では、すべての追加的な認証が、最初の認証が実行された方法から独立している。その結果、最初の認証は、公開鍵暗号をＫｅｒｂｅｒｏｓ認証に組み込むのに便利な場所を提供する。加えて、最初の交換の後の対称暗号の使用は、性能考察に適している。

【００４４】

ＲＦＣ４４５６は、クライアントおよびＫＤＣが、ＡＳ交換によって相互に認証する公開鍵と秘密鍵の組を使用して、クライアントとＫＤＣのみにより知られるＡＳ応答鍵を交渉し、ＫＤＣによって送信されたＡＳ－ＲＥＰを暗号化できる方法およびデータ形式について記載する。

【００４５】

図３を参照すると、ＴＬＳハンドシェイク（図１への参照とともに論じられたのと同様に、「２重の」ハンドシェイクにすることができる）の完了後、ゲートウェイ２０は、ＴＨＲ４５を第三者エンティティ４０に提供する。図１の場合のように、再度、「信頼するエンティティ」（この場合、第三者エンティティ４０）は、クライアント／ユーザー１０とゲートウェイ２０間の認証に関与せずに判定する。むしろ、信頼するエンティティは、ＴＨＲに依存して、ユーザー資格をゲートウェイに提供するかどうかを判定する。

【００４６】

さらに具体的には、有効なＴＬＳハンドシェイク（すなわち、ＴＨＲ）の交換において、第三者エンティティ４０は、参照番号５５に示すように、ＵＣ(user credential)の形式をゲートウェイ２０に戻す。ゲートウェイ２０は、今度は、参照番号６５に示すように、ＵＳを使用してウェブサーバー３０に対して認証する。ＫＤＣの場合、ユーザー資格は、ユーザー１０の名におけるＫｅｒｂｅｒｏｓのサービスチケットまたはＴＧＴとなる。ＣＡの場合、ユーザー資格は、ユーザーの名における証明書（通常、短い存続期間とともに）となる。

【００４７】

図５は、クライアント／ユーザーが、第三者エンティティを含むシステムのウェブサーバーにアクセスしようとするときに実行される認証プロセスのステップを示す説明的なフ

ロー図である。このシステムにおいて、そのプロセスは、クライアント／ユーザー１０がゲートウェイ２０にアクセスするときに開始する（ステップ５００）。クライアント／ユーザー１０は次に、所望のウェブサーバー３０のリソースを要求する（ステップ５１０）。クライアント／ユーザー１０がウェブサーバーにログインしていない場合、そのクライアント／ユーザー１０は、ウェブサーバー３０がアクセスを許可する前に認証されなければならない。

【００４８】

クライアント／ユーザー１０およびゲートウェイ２０は次に、上記に詳細に示した方法でクライアント認証を有するＴＬＳハンドシェイクを実行する（ステップ５２０）。ＴＬＳハンドシェイクの少なくとも一部の記録（ＴＨＲ）が作成され、第三者エンティティ（「信頼するエンティティ」）４０に提供される（ステップ５３０）。

【００４９】

ＴＨＲの受信後、第三者エンティティ４０は、ユーザーがゲートウェイに認証されたことを検証する（ＴＨＲの証明書検証メッセージを有効化することによって）（ステップ５４０）。ＴＨＲが有効かつフレッシュであると検証された場合（ステップ５５０）、ユーザー資格（例えば、ＣＡの場合における一時的な証明書、またはＫＤＣの場合におけるＫｅｒｂｅｒｏｓのサービスチケット）は、ゲートウェイ２０に提供される（ステップ５６０）。ゲートウェイ２０は次に、ユーザー資格を使用して、実クライアント／ユーザーとしてウェブサーバー３０に対して認証する（ステップ５７０）。しかしながら、ＴＨＲが有効かつフレッシュと検証できない場合（ステップ５５０において）、ユーザー資格は、ゲートウェイ２０に提供されず、アクセスは拒否される（ステップ５５５）。

【００５０】

クライアント／ユーザーがウェブサーバーにアクセスする権限を与えられたと仮定して、ユーザー資格（例えば、クライアントの証明書）がウェブサーバー３０によって認証された場合、要求されたウェブサーバーへのアクセスは、クライアント／ユーザーに許可され（ステップ５８０）、クライアント証明を検証できない場合、アクセスは、典型的には、拒否される（ステップ５９０）。

【００５１】

ユーザー資格は、サービスチケット（ＫＤＣベースのデプロイメントにおける）か、一時的な証明書（ＣＡベースのデプロイメントにおける）のいずれかから成るゲートウェイ２０に提供される。ＫＤＣベースのデプロイメントは、上記で論じたＫｅｒｂｅｒｏｓに制約された委任（Ｓ４Ｕ２Ｓｅｌｆ＋Ｓ４Ｕ２Ｐｒｏｘｙ）とほぼ同じなので、これ以上論じない。その代わりに、ＣＡベースのデプロイメントについて論じる。

【００５２】

ＣＡベースのデプロイメントにおいて、有効かつ「フレッシュ」なＴＨＲが提供されたとき、１つまたは複数のＣＡを使用してクライアント証明書を発行する。このシナリオにおいて、サービスプロバイダー（ウェブサーバー３０）は、ＣＡを信頼するように組み込まれて（これは典型的には、ＣＡ自体の証明書は、サービスプロバイダーのオペレーティングシステム上のある特定の場所にインストールされなければならないことを意味する）、ユーザーの実アイデンティティを提供しなければならない。

【００５３】

いったんユーザー（およびそれに関連する秘密鍵）の名においてクライアント証明書が与えられると、ゲートウェイ２０は次に、実際のユーザーとしてサービスプロバイダー３０に対して認証するこれらの資格を使用する。証明書および秘密鍵を使用してサービスプロバイダー（例えば、ウェブサーバー３０）に対して認証するために、ゲートウェイ２０およびサービスプロバイダーは、クライアント証明書をサポートする認証プロトコルを使用しなければならない。クライアント証明書をサポートする２つの公知の認証プロトコルは、ＴＬＳ（またはＳＳＬ）、またはＫｅｒｂｅｒｏｓ（ｗ／ＰＫＩＮＩＴ）である。上記により詳細に論じたように、ＴＬＳ（またはＳＳＬ）は、(証明書に基づいた)クライアント認証を含むことができる。「Ｐｕｂｌｉｃ Ｋｅｙ Ｃｒｙｐｔｏｇｒａｐｈｙ ｆ

ｏｒ　Ｉｎｉｔｉａｌ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｉｎ　Ｋｅｒｂｅｒｏｓ」と題したＰＫＩＮＩＴ機構（ＲＦＣ４５５６）は、Ｋｅｒｂｅｒｏｓを使用可能のクライアントが、公開鍵暗号を経由して（すなわち、証明書およびそれに関連する秘密鍵を経由して）ＴＧＴを取得できるＫｅｒｂｅｒｏｓプロトコルへのプロトコル拡張の機構である。さらに具体的には、このような拡張は、事前認証データフィールドの非対称鍵署名および／または暗号アルゴリズムを使用することによって、公開鍵暗号を最初の認証交換に組み込む方法を提供する。

【００５４】

クライアント／ユーザーがウェブサーバーにアクセスする権限を与えられたと仮定して、いったん資格（クライアント証明書）がウェブサーバー３０によって認証されると、要求されたウェブサーバーへのアクセスは、クライアント／ユーザーに許可される。もちろんクライアント証明書が検証できない場合（すなわちウェブサーバーがＣＡを信頼するように組み込まれてない場合）、ユーザーによるウェブサーバーへのアクセスは拒否される。

【００５５】

図１の実施形態にあるように、「２重の」ＴＬＳハンドシェイクを、ゲートウェイ２０を侵害する攻撃者に対するさらなる追加的な防御のために、図３の実施形態に実装することができる。

【００５６】

本明細書の発明の主題を構造上の特徴および／または方法論的動作に特有な言語によって説明してきたが、特許請求の範囲に定義された発明の主題は、上記の具体的な特徴または動作に必ずしも限定されないことも理解されたい。むしろ上記の具体的な特徴または動作は、特許請求の範囲を実装する例示的な形式として開示される。

【００５７】

例えば、この文書を通じて我々は、ゲートウェイを介してクライアント／ユーザーがアクセスするサービスプロバイダーから成る一連のエンティティを参照する。しかしながら、これは、実現可能なシナリオの１つにすぎず、便宜上この文書を通じて使用されているにすぎない。他の実現可能なシナリオは、バックエンドアプリケーションまたはデータベースに対するユーザーとして認証する必要があるウェブサーバーを含む。本開示の新規の態様は、連鎖の中のエンティティ間の認証を要求する一連のエンティティに適用できる。各エンティティが元のクライアントとして連鎖中の次のエンティティに対して認証しなければならない、連鎖中の任意の数のエンティティにすることができる。

【００５８】

１つの要素が別の要素に反応するものとして表示される場合、それらの要素を、直接的または間接的につなげられることがさらに理解される。本明細書に図示した接続を、要素間の結合または通信インタフェースを実現するために、実際には論理的または物理的にすることができる。接続は、とりわけ、ソフトウェアプロセス間のプロセス間通信、またはネットワークコンピューター間のマシン間通信として実装することができる。

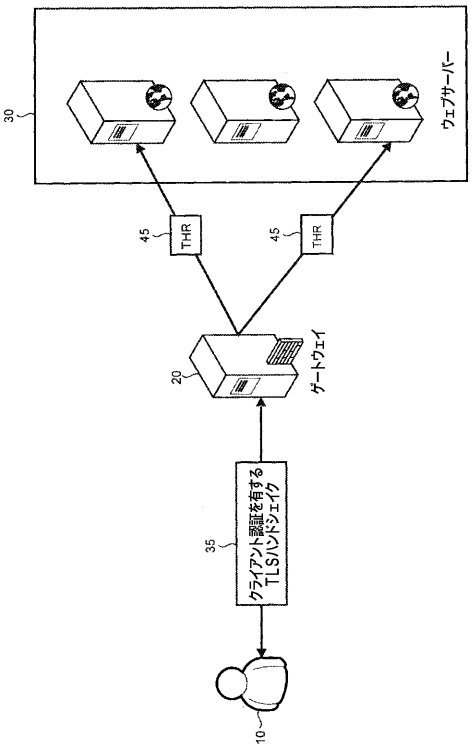
【００５９】

本明細書で使用される用語「例示的な」および「説明的な」は、実施例、事例、または説明としての役割を果たすことを意味する。本明細書で「例示的」または「説明的」として記述された任意の実装またはそれらの態様は、他の実装またはそれらの態様に対して必ずしも好適または有利であるように構成されていない。

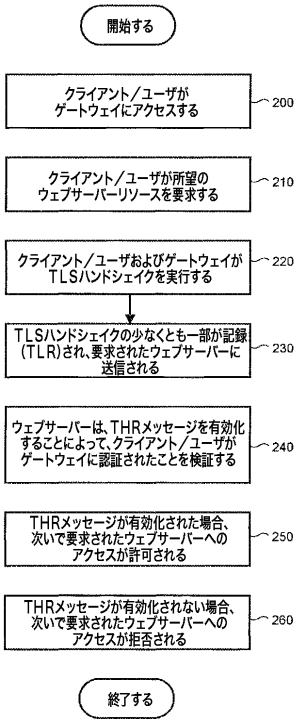
【００６０】

添付した特許請求の範囲の趣旨および範囲から逸脱しなければ、上記の具体的な実施形態以外の実施形態を考案できることが理解されているので、本明細書の主題の範囲は、次の特許請求の範囲に準拠することが意図される。

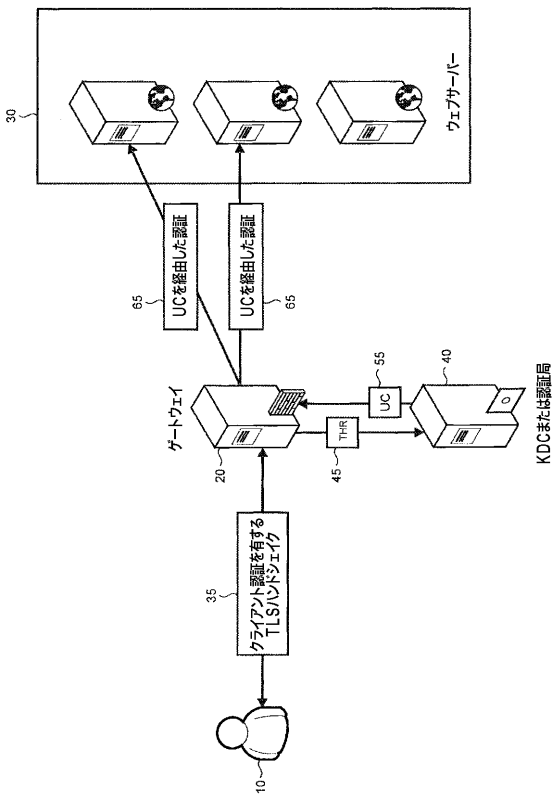
【図１】


30
ウェブサーバー
45
THR
45
THR
20
ゲートウェイ
35
クライアント認証を有するTLSハンドシェイク
10

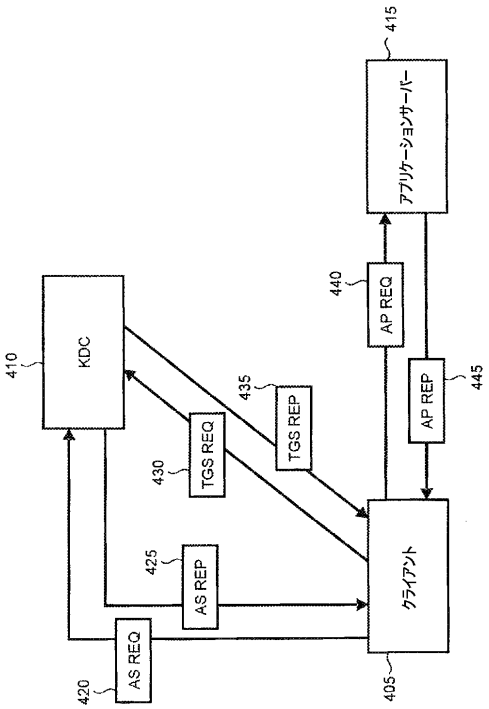
【図２】


開始する
クライアント／ユーザがゲートウェイにアクセスする
200
クライアント／ユーザが所望のウェブサーバーリソースを要求する
210
クライアント／ユーザおよびゲートウェイがTLSハンドシェイクを実行する
220
TLSハンドシェイクの少なくとも一部が記録(TLR)され、要求されたウェブサーバーに送信される
230
ウェブサーバーは、THRメッセージを有効化することによって、クライアント／ユーザがゲートウェイに認証されたことを検証する
240
THRメッセージが有効化された場合、次いで要求されたウェブサーバーへのアクセスが許可される
250
THRメッセージが有効化されない場合、次いで要求されたウェブサーバーへのアクセスが拒否される
260
終了する

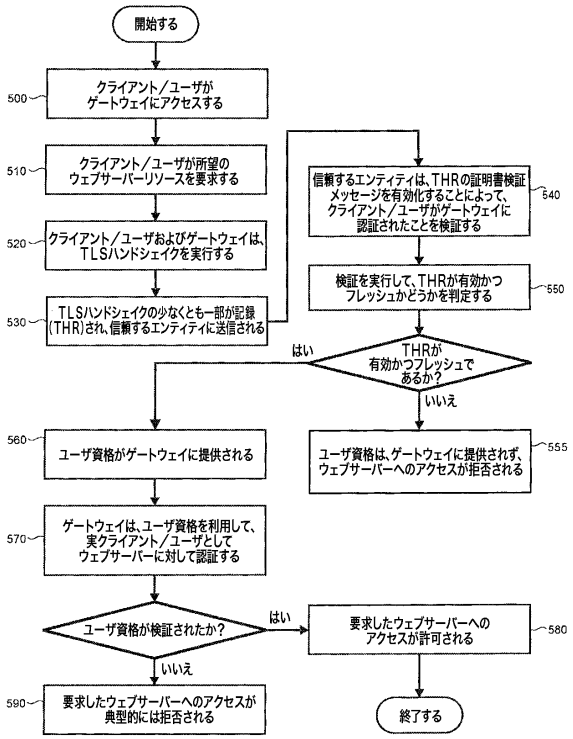
【図３】


30
ウェブサーバー
65
UCを経由した認証
65
UCを経由した認証
ゲートウェイ
20
55
UC
40
45
THR
KDCまたは認証局
35
クライアント認証を有するTLSハンドシェイク
10

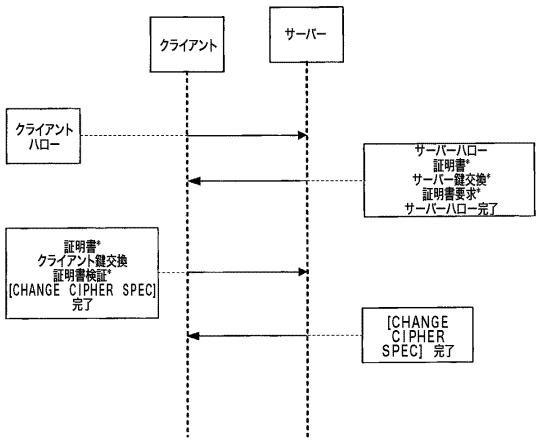
【図４】


415
アプリケーションサーバー
440
AP REQ
445
AP REP
410
KDC
435
TGS REP
430
TGS REQ
425
AS REP
420
AS REQ
クライアント
405

【図５】


開始する
500
クライアント／ユーザが
ゲートウェイにアクセスする
510
クライアント／ユーザが所望の
ウェブサーバーリソースを要求する
520
クライアント／ユーザおよびゲートウェイは、
TLSハンドシェイクを実行する
530
TLSハンドシェイクの少なくとも一部が記録
(THR)され、信頼するエンティティに送信される
信頼するエンティティは、THRの証明書検証
メッセージを有効化することによって、
クライアント／ユーザがゲートウェイに
認証されたことを検証する
540
検証を実行して、THRが有効かつ
フレッシュかどうかを判定する
550
THRが
有効かつフレッシュで
あるか？
はい
いいえ
555
ユーザ資格は、ゲートウェイに提供されず、
ウェブサーバーへのアクセスが拒否される
560
ユーザ資格がゲートウェイに提供される
570
ゲートウェイは、ユーザ資格を利用して、
実クライアント／ユーザとして
ウェブサーバーに対して認証する
ユーザ資格が検証されたか？
はい
580
要求したウェブサーバーへの
アクセスが許可される
いいえ
590
要求したウェブサーバーへのアクセスが
典型的には拒否される
終了する

【図６】


クライアント
サーバー
クライアント
ハロー
サーバーハロー
証明書*
サーバー鍵交換*
証明書要求*
サーバーハロー完了
証明書*
クライアント鍵交換
証明書検証*
[CHANGE CIPHER SPEC]
完了
[CHANGE
CIPHER
SPEC] 完了

フロントページの続き

(74)代理人 100120112
弁理士 中西 基晴
(74)代理人 100147991
弁理士 鳥居 健一
(74)代理人 100119781
弁理士 中村 彰吾
(74)代理人 100162846
弁理士 大牧 綾子
(74)代理人 100173565
弁理士 末松 亮太
(74)代理人 100138759
弁理士 大房 直樹
(74)代理人 100091063
弁理士 田中 英夫
(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 ジェナディ メドゥビンスキー
アメリカ合衆国 ９８０５２－６３９９ 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 インターナショナル パテンツ内
(72)発明者 ニール ナイス
アメリカ合衆国 ９８０５２－６３９９ 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 インターナショナル パテンツ内
(72)発明者 トマー シラン
アメリカ合衆国 ９８０５２－６３９９ 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 インターナショナル パテンツ内
(72)発明者 アレクサンダー テプリットスキー
アメリカ合衆国 ９８０５２－６３９９ 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 インターナショナル パテンツ内

審査官 青木 重徳

(56)参考文献 特開２００６－１６５６７８（ＪＰ，Ａ）
特表２００３－５０３９６３（ＪＰ，Ａ）
特開２００３－２２９８４９（ＪＰ，Ａ）
米国特許出願公開第２００７／０２９４７４９（ＵＳ，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.94-108,120-121,125-132
Matthew Hur, Joseph Salowey, Ari Medvinsky, “Kerberos Cipher Suites in Transport Layer Security (TLS)”, INTERNET-DRAFT Transport Layer Security Working Group draft-ietf-tls-kerb-01.txt, [online], ２００１年１１月 8日, Obsoletes: RFC 2712, [retrieved on 2012-10-30]. Retrieved from the Internet, ＵＲＬ, <http://tools.ietf.org/pdf/draft-ietf-tls-kerb-01.pdf>
K. Jackson, S. Tuecke, D. Engert, “TLS Delegation Protocol”, Internet Draft, [online], ２００２年 2月, draft-ietf-tls-delegation-01.txt, [retrieved on 2012-10-30]. Retrieved from the Internet, ＵＲＬ, <http://tools.ietf.org/pdf/draft-ietf-tls-delegation-01.pdf>

(58)調査した分野(Int.Cl., ＤＢ名)
Ｈ０４Ｌ　　９／３２
Ｇ０６Ｆ　２１／４１
Ｇ０９Ｃ　　１／００