(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4591897号
(P4591897)

(45)発行日　平成22年12月1日(2010.12.1)　(24)登録日　平成22年9月24日(2010.9.24)

(51)Int.Cl.　F I
H04L 9/08 (2006.01)　H04L 9/00 601C
H04L 9/14 (2006.01)　H04L 9/00 601E
H04L 9/00 641

請求項の数 11　(全 13 頁)

(21)出願番号　特願2007-502329 (P2007-502329)
(86)(22)出願日　平成17年3月1日(2005.3.1)
(65)公表番号　特表2007-528172 (P2007-528172A)
(43)公表日　平成19年10月4日(2007.10.4)
(86)国際出願番号　PCT/EP2005/050895
(87)国際公開番号　WO2005/086452
(87)国際公開日　平成17年9月15日(2005.9.15)
審査請求日　平成20年2月7日(2008.2.7)
(31)優先権主張番号　0405245.2
(32)優先日　平成16年3月9日(2004.3.9)
(33)優先権主張国　英国(GB)

早期審査対象出願

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】鍵ベースの暗号化

(57)【特許請求の範囲】
【請求項１】
通信リンクによって相互に接続されたクライアントとサーバとの間でデータを、秘密鍵を用いて暗号化して送信するための方法であって、
前記クライアントが、
前記通信リンクが所定時間、アイドルであったと判断する第１のステップと、
以前アイドルであった前記通信リンクにより流すべきデータが存在すると判断する第２のステップと、
所定時間、前記通信リンクがアイドルであったと判断するとハートビートにより、当該クライアントが依然として存在することを前記サーバに通知する第３のステップと、
前記サーバから前記ハートビートの受信を確認する応答を受信する第４のステップと、
以前アイドルであった前記通信リンクにより流すべきデータが存在すると判断した際、所定時間内にハートビートの受信の確認を受信しないと前記サーバと前記通信リンクによって送信されたデータを暗号化するための新しい秘密鍵の生成を開始する第５のステップとを実行する方法。
【請求項２】
所定時間内にハートビートの受信の確認を受信しないと、前記第５のステップに代えて、前記クライアントは前記サーバとの通信を終了する第６のステップを実行する請求項１に記載の方法。
【請求項３】

前記第１のステップでは、前記クライアントから前記サーバに前記ハートビートを送信するのに十分なほど前記通信リンクがアイドルであったかを判断する請求項１に記載の方法。

【請求項４】

前記第５のステップでは、前記クライアントから前記サーバにハートビートを送信させるのに十分なほど前記通信リンクがアイドルであったと判断すると、新しい秘密鍵の生成を開始する請求項３に記載の方法。

【請求項５】

前記クライアントは、新しい秘密鍵の生成の開始前に、少なくとも前記サーバの認証を開始する第７のステップを実行する請求項１ないし４のいずれかに記載の方法。

【請求項６】

前記通信リンクがアイドルであったと判断すると、前記クライアントは前記秘密鍵で暗号化されたデータを無視する第８のステップを実行する請求項１に記載の方法。

【請求項７】

前記クライアントは新たに生成された秘密鍵で暗号化された後続データのみを受け入れる第９のステップを実行する請求項６に記載の方法。

【請求項８】

通信リンクによって相互に接続されたクライアントとサーバとの間でデータを、秘密鍵を用いて暗号化して送信するため前記クライアントに備えられた装置であって、

前記通信リンクが所定時間、アイドルであったと判断する第１の手段と、

以前アイドルであった前記通信リンクにより流すべきデータが存在すると判断する第２の手段と、

所定時間、前記通信リンクがアイドルであったと判断するとハートビートにより、当該クライアントが依然として存在することを前記サーバに通知する第３の手段と、

前記サーバから前記ハートビートの受信を確認する応答を受信する第４の手段と、

以前アイドルであった前記通信リンクにより流すべきデータが存在すると判断した際、所定時間内にハートビートの受信の確認を受信しないと前記サーバと前記通信リンクによって送信されたデータを暗号化するための新しい秘密鍵の生成を開始する第５の手段とを有する装置。

【請求項９】

前記通信リンクがアイドルであったと判断すると、前記秘密鍵で暗号化されたデータを無視する第６の手段を有する請求項８に記載の装置。

【請求項１０】

前記クライアントに備えられたコンピュータによって請求項１ないし７のいずれかに記載の前記方法を実行させるコンピュータ・プログラム。

【請求項１１】

請求項１０に記載のコンピュータ・プログラムが記録されたコンピュータに読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、暗号化に関し、詳細には、暗号化に使用された鍵の再交渉（renegotiation）に関する。

【背景技術】

【０００２】

個人も企業も同様に、様々なデータを送受信するためにコンピュータを使用している。このようなデータの妥当な部分は、機密のものになる可能性があり、したがって、データ・プライバシを保証することは重要である。

【０００３】

データ・プライバシを達成する方法として一般的なものは、暗号化アルゴリズムの使用

によるものである。このようなアルゴリズムは典型的には鍵ベースのものであり、対称または非対称のいずれかとして分類される。

【０００４】

対称暗号化アルゴリズムでは、問題のデータの送信側と受信側のみに知られている秘密鍵を使用する。送信側でデータを暗号化するために使用する秘密鍵は、受信側で受信されるときにデータを暗号化解除するために使用するものと同じである。

【０００５】

これに対して、非対称暗号化アルゴリズムでは、公開鍵と秘密鍵（秘密鍵）の両方を使用する。公開鍵は誰にでも知られている可能性があるが、秘密鍵は限られた数のエンティティのみに知られている。一方の鍵はデータを暗号化するために使用され、もう一方の鍵はデータの暗号化解除を可能にする。

【０００６】

セキュア・ソケット・レイヤー（ＳＳＬ：Ｓｅｃｕｒｅ　Ｓｏｃｋｅｔｓ　Ｌａｙｅｒ）は、インターネットによりセキュア・データ伝送を達成するためのプロトコルである。ＳＳＬでは、非対称暗号化技法と対称暗号化技法の両方を使用する。

【０００７】

２人の当事者（たとえば、アリスとボブ）間の初期認証ハンドシェークには、１対の非対称鍵が使用される。以下の例では、アリスがボブを認証したいと希望している（当然のことながら、ボブもアリスを認証したいと希望している可能性があり、これは賢明なことである）。ボブは公開鍵・秘密鍵の対を有する。ボブの公開鍵はアリスに開示されている。アリスはボブにメッセージを伝送し、その後、ボブがこれを自分の秘密鍵で暗号化してアリスに返す。アリスは、ボブが前もって彼女に開示した公開鍵を使用して、ボブからのメッセージを暗号化解除する。暗号化解除されたメッセージが、アリスが当初ボブに送信したメッセージと一致する場合、アリスは、ボブが自分で名乗っている通りの人であると想定することができる。しかし、ＳＳＬでは、第三者がアリスの元のメッセージを入手して、アリスを詐称するのを防止するために、デジタル署名も使用する。また、ＳＳＬでは、証明書も使用する。証明書は、公開鍵が本当に、たとえば、ボブのものであることを照明するために使用される。

【０００８】

ボブを認証すると、アリスはボブとデータを交換する用意ができている。しかし、データ交換を行えるようになる前に、アリスとボブは対称（秘密）鍵について合意していなければならない。交換すべきデータは、まず、この秘密鍵で暗号化される。両当事者が秘密鍵について合意しているので、アリスはこの鍵を使用して自分のデータを暗号化することができ、ボブはアリスから受信したデータを暗号化解除することができる。

【０００９】

この秘密鍵が無許可の第三者によって発見された場合、それを使用して、データを暗号化解除し、送信偽装（spoof）メッセージの暗号化／データの変更を行うことができることが分かるであろう。

【００１０】

データを交換するためにアリス（クライアント）とボブ（サーバ）が使用するＳＳＬ秘密鍵を定期的に再交渉することが好ましいのは、このためである。秘密鍵の再交渉は、クライアントとサーバの両方についてＣＰＵ集中ハンドシェークを実行することを必要とする。再交渉ごとに完全非対称認証とそれに続く対称秘密鍵の交渉とを必要とするときに、これは特にプロセッサ集中型のものになる。

【００１１】

ＳＳＬの詳細な概要については、ｈｔｔｐ：／／ｄｅｖｅｌｏｐｅｒ．ｎｅｔｓｃａｐｅ．ｃｏｍ／ｔｅｃｈ／ｓｅｃｕｒｉｔｙ／ｓｓｌ／ｈｏｗｉｔｗｏｒｋｓ．ｈｔｍｌに記載されている可能性があることに留意されたい。

【００１２】

現行の解決策

現行の秘密鍵再交渉実現例では、一般に、以下の２通りの方法のうちの１つを使用する。
（ｉ）ｘ分ごとにＳＳＬクライアントによって再交渉が開始される時限リセット（たとえば、Ｗｅｂブラウザは２分ごとに鍵の再交渉を開始することができる）または
（ｉｉ）特定のしきい値のバイト数が流れた後の開始。
【００１３】
しかし、メッセージング環境では通信リンクが典型的にはアイドルと使用中との間で変動するので、これらの解決策はこのような環境では効率よく機能しない。特に通信リンクが（メッセージング環境で起こり得るように）変動する時刻にアイドルまたは使用中になる場合、上述の解決策は特に効率の悪いものになる。
【発明の開示】
【発明が解決しようとする課題】
【００１４】
現行の解決策の問題
（ｉ）時限再交渉－アイドル通信リンク
長時間の間、通信リンクによりデータ（メッセージ）がまったく送信されていない場合、不必要な数の完全認証および再交渉が行われた可能性がある。換言すれば、クライアントは、それが通信したいサーバを不必要なほど頻繁に認証し、そのサーバと秘密鍵について再交渉する可能性がある。したがって、パフォーマンスは不必要に低下する。
（ｉｉ）バイトしきい値実現例－アイドル通信リンク
この解決策は、秘密鍵がアイドル・リンク上で有効である時間を増加し、検出されずに秘密鍵を攻撃し、「送信偽装」メッセージを送信するためのより多くの時間をハッカに与えるものである。
（ｉｉｉ）時限交渉－使用中通信リンク
使用中通信リンクは、同じ秘密鍵で暗号化された大量のデータを流すことになる。ハッカは、一般に時限再交渉に使用される類の時間に鍵を破れそうもないが、問題は、ハッカが暗号化されたデータを記録して、都合の良いときにそれを分析できることである。これは、大量のデータが同じ秘密鍵で暗号化されている場合、ハッカにとってより容易なことであり、したがって、通信リンクが使用中であるときにこの解決策を使用すると、大量のデータのセキュリティは損なわれる可能性がある。
（ｉｖ）バイトしきい値実現例－使用中通信リンク
使用中通信リンク上で同じ秘密鍵で暗号化されたデータの量は最小限になる。したがって、この解決策は、単一秘密鍵で暗号化されたデータの量を最小限にするものである。しかし、リンクが主としてアイドルである場合、この解決策は適切ではない（上記を参照）。
【００１５】
したがって、使用中とアイドルとの間で変動する環境での暗号化は、これまでは問題の多いものであった。
【課題を解決するための手段】
【００１６】
一態様によれば、通信リンクにより第１のエンティティと第２のエンティティとの間を流れるデータを暗号化するための秘密鍵を使用するセキュア・データ通信を容易にするための方法が提供され、この方法は、通信リンクがアイドルであったと判断するステップと、以前アイドルであった通信リンクにより流すべきデータが存在すると判断するステップと、以前アイドルであった通信リンクにより流すべきデータが存在すると判断したことに応答して、通信リンクにより第１のエンティティと第２のエンティティとの間で送信されたデータを暗号化するための新しい秘密鍵の生成を開始するステップとを含む。
【００１７】
このようにして、アイドル通信リンクによる伝送が再開しようとしている場合のみ、鍵生成が行われる。これは、鍵生成が時限ベースで行われる可能性のある従来技術とは反対

のものである。

【００１８】

好ましくは、事前構成された量のデータが通信リンクにより送信された時期を決定することも可能である。事前構成された量がリンクにより送信されると、好ましくは新しい秘密鍵の生成が開始される。

【００１９】

これは、通信リンクが主としてアイドルではない状況を考慮している。したがって、使用中リンク上でも、鍵生成は十分頻繁に行われる。

【００２０】

一実施形態では、リンクにより流すべきデータが存在するという判断の結果として新しい秘密鍵の生成が開始される前に、少なくとも所定時間、通信リンクがアイドルでなければならない。

【００２１】

このようにして、短時間のアイドル状態は、新しい秘密鍵を生成するためのプロセスを直ちに開始させるわけではない。

【００２２】

リンクが少なくともｘ秒間の間、アイドルであり、流すべきデータが存在する場合に、新しい秘密鍵を生成しなければならないことを伝える単純なタイムアウト・システムを使用できることに留意されたい。

【００２３】

好ましい一実施形態では、所定の時間の間、通信リンクがアイドルであったと判断されると、ハートビートにより、第１のエンティティが依然として存在することが第２のエンティティに通知される。

【００２４】

このようにして、第２のエンティティは、第２のエンティティが現在、通信リンクにより流すべきデータをまったく持っていなくても、第１のエンティティが依然として現存していることを承知している。第１のエンティティは、２つ以上のハートビートを第２のエンティティに送信することができる（すなわち、十分長い間、リンクがアイドルである場合）ことに留意されたい。

【００２５】

第２のエンティティは、好ましくは、第１のエンティティからのハートビートの受信を確認する。

【００２６】

一実施形態では、所定時間内にハートビートの受信の確認が第１のエンティティによってまったく受信されない場合、第１のエンティティによる第２のエンティティとの通信が終了する。これは、第２のエンティティに障害が発生したか、または第三者がハートビート／ハートビートに対する応答を消費しているためである。

【００２７】

他の実施形態では、所定時間内にハートビートの受信の確認が第１のエンティティによってまったく受信されない場合、第１のエンティティによって第２のエンティティにもう一度データを伝送することが許容される前に新しい秘密鍵の生成が開始される。当然のことながら、プロセスが認証を含まない場合（以下を参照）、第三者が第２のエンティティであるふりをし、したがって、鍵生成プロセスに関与することもできるであろう。

【００２８】

好ましい一実施形態では、第１のエンティティにより第２のエンティティにハートビートを送信させるのに十分なほど通信リンクがアイドルであったと判断することは可能である。好ましくは、第１のエンティティにより第２のエンティティにハートビートを送信させるのに十分なほどリンクがアイドルであったと判断したことに応答して、新しい秘密鍵の生成が開始される。

【００２９】

好ましい一実施形態では、新しい秘密鍵の生成の開始前に、少なくとも第２のエンティティの認証が開始される。

【００３０】

新しい秘密鍵の生成は、好ましくは、第１のエンティティと第２のエンティティとの間で実行された交渉プロセスの結果として行われる。

【００３１】

他の態様によれば、通信リンクにより第１のエンティティと第２のエンティティとの間を流れるデータを暗号化するための秘密鍵を使用するセキュア・データ通信を容易にするための方法が提供され、この方法は、通信リンクがアイドルであったと判断するステップと、通信リンクがアイドルであったと判断したことに応答して、秘密鍵で暗号化されたデータを無視するステップとを含む。

【００３２】

好ましくは、新たに生成された秘密鍵で暗号化された後続データのみが受け入れられる。

【００３３】

好ましくは、少なくとも所定時間、通信リンクはアイドルでなければならない。好ましくは、これは、第１のエンティティからのハートビートの受信により示される。

【００３４】

一実施形態により、少なくとも所定時間、通信リンクがアイドルであり、ハートビートが第１のエンティティからまったく受信されていないと判断されると、第１のエンティティとの通信が終了する。

【００３５】

これは、第１のエンティティに障害が発生したか、または第三者がハートビートを消費していると想定されるためである。

【００３６】

他の実施形態により、少なくとも所定時間、通信リンクがアイドルであり、ハートビートが第１のエンティティからまったく受信されていないと判断したことに応答して、新たに生成された秘密鍵で暗号化された後続データのみが受け入れられる。

【００３７】

他の態様によれば、通信リンクにより第１のエンティティと第２のエンティティとの間を流れるデータを暗号化するための秘密鍵を使用するセキュア・データ通信を容易にするための装置が提供され、この装置は、通信リンクがアイドルであったと判断するための手段と、以前アイドルであった通信リンクにより流すべきデータが存在すると判断するための手段と、以前アイドルであった通信リンクにより流すべきデータが存在すると判断したことに応答して、通信リンクにより第１のエンティティと第２のエンティティとの間で送信されたデータを暗号化するための新しい秘密鍵の生成を開始するための手段とを含む。

【００３８】

他の態様によれば、通信リンクにより第１のエンティティと第２のエンティティとの間を流れるデータを暗号化するための秘密鍵を使用するセキュア・データ通信を容易にするための装置が提供され、この装置は、通信リンクがアイドルであったと判断するための手段と、通信リンクがアイドルであったと判断したことに応答して、秘密鍵で暗号化されたデータを無視するための手段とを含む。

【００３９】

好ましくは、データの保全性を信頼することが安全であるとは見なされないという意味で、秘密鍵で暗号化されたデータが無視される。したがって、好ましくは、新たに生成された秘密鍵で暗号化されたデータのみが信頼できるものとして受け入れられる。

【００４０】

本発明はコンピュータ・ソフトウェアで実現可能であることが分かるであろう。

【００４１】

次に、一例としてのみ、添付図面に関連して、本発明の好ましい一実施形態について説

明する。

【発明を実施するための最良の形態】

【００４２】

次に、図１および図２に関連して、本発明の好ましい一実施形態について説明する。２つの図は、互いに関連して読まなければならない。

【００４３】

ＳＳＬクライアント５は、ＳＳＬサーバ６にデータを伝送することを希望している。まず、ＳＳＬクライアントは、接続イニシエータ（connectioninitiator）５５を使用して、通信リンク９０によりサーバとの接続を開始する。次に、クライアント５は、サーバ上の同等のコンポーネント１０’と通信するオーセンティケータ１０を使用して、サーバ６を認証する（ステップ１００）。

【００４４】

サーバ６を認証すると、クライアントとサーバは、鍵ネゴシエータ（keynegotiator）・コンポーネント２０、２０’により対称秘密鍵について交渉する（ステップ１１０）。その後、この秘密鍵を使用して、クライアントが通信リンク９０を越えて流すメッセージを暗号化し、暗号化解除する。

【００４５】

クライアント上のデータ検出器（datadetector）７０は、クライアント５が通信リンク９０により流すべき任意のデータを持っているかどうかを検出するよう機能する（ステップ１２０）。リンクにより流すべきデータが存在する場合、結果的にクライアントがサーバにハートビートを送信することになるのに十分なほど以前リンクがアイドルであったかどうかがステップ１３０で検出される（以下を参照）。

【００４６】

そうではないと想定すると、このデータは現行の秘密鍵で暗号化され、送信される（図示せず）。事前構成された数のバイトが送信されたかどうかがバイト測定器（bytemeasurer）４０により判断される（ステップ１５０）。答えがＮＯである場合、プロセスはステップ１２０にループして、流すべきデータがそれ以上存在するかどうかを確認する。

【００４７】

（バイト測定器４０によって検出された通り）事前構成された数のバイトが送信された場合、コンポーネント１０、１０’、２０、２０’を使用して再認証し、鍵について再交渉する時期である（ステップ１００、１１０）。この時点で、バイト測定器４０によって保持された送信バイト数の値はゼロにリセットされる。

【００４８】

秘密鍵はバイトしきい値が満足された結果として規則正しく再交渉されるので、構成可能なバイトしきい値は、同じ秘密鍵で使用中通信リンク上で送信されるデータの量が制限されることを保証する。したがって、同じ秘密鍵で暗号化されるデータの量は最小限になる。

【００４９】

適切なバイトしきい値の設定は１つのトレードオフであることに留意されたい。

【００５０】

しきい値が低いほど、再認証が実行され、秘密鍵が変更される頻度が高くなり、したがって、必要とされる処理能力が増加する。しかし、再認証が実行され、秘密鍵が変更される頻度が高くなるほど、通信リンクにより流れるデータの安全性が高くなる。

【００５１】

しきい値が高いほど、パフォーマンスがより良好になる（再認証および秘密鍵の再交渉の回数が減少するため）。当然のことながら、しきい値がより低い環境に比べ、通信リンクにより流れるデータの安全性は低くなる。

【００５２】

タイマ３０は、構成可能な時間の間、通信リンク９０がアイドルであった時期を決定するために、データ検出器コンポーネント７０によって使用される。そうである場合、これ

は、（ハートビート発行器（heartbeatissuer）５０により）特殊な「ハートビート」メッセージを発行して、それが依然として存在することをＳＬＬサーバ６に対して確認する（ステップ１６０）。（次に、タイマはゼロにリセットされるが、好ましくは、再認証が開始されるときにもタイマがゼロに設定されることに留意されたい。）クライアントは、サーバ（ハートビート受信器（heartbeatreceiver）７５、ハートビート応答発生器（heartbeatreply generator）８０）からハートビートに対する応答を待つ（ステップ１７０）。以下を参照されたい。

【００５３】

その結果、非常に多数のハートビートが発生する（すなわち、不要なトラフィックが多すぎる）可能性があるので、構成可能な時間は好ましくは短すぎない（たとえば、１０秒）ものであることに留意されたい。選択された時間は環境によって決まるものであり、たとえば、５分間という時間が適切である可能性がある。

【００５４】

ＳＳＬサーバは、１つまたは複数の「ハートビート」メッセージを受信した後、同じ秘密鍵で暗号化されたアプリケーション・データを含む任意のその他のメッセージを「送信偽装」として拒否することになる（データ・リジェクタ（datarejecter）・コンポーネント９５）。送信偽装データを検出すると、管理者によりこれをログ記録すること、およびクライアントとの接続をクローズすることなど、適切なアクション（複数も可）を実行しなければならない。

【００５５】

ＳＳＬクライアントがＳＳＬサーバ（通信リンクが以前アイドルであったことを示すハートビートを受信したもの）に新しいメッセージを送信するために、ＳＳＬクライアントは、ＳＳＬサーバがそれを「送信偽装」として拒否するのを回避するために、メッセージを送信する前にまず新しい秘密鍵について再交渉しなければならない（ステップ１２０、１３０、１００、および１１０）。したがって、アイドル状態の期間後に機密漏れは一切発生しないはずである。

【００５６】

前に述べた通り、アイドル状態の期間後に送信された送信偽装データにより、好ましくは、サーバはクライアントとの接続を終了する。次に、クライアントは、サーバとのその接続を再開することを選択することができ、それ以上の何らかのデータをサーバに送信する前に再認証および再交渉を行わなければならない。

【００５７】

ハートビートは有用なデータをいずれも含まないので、暗号化する必要はないことに留意されたい。

【００５８】

構成可能な時間（すなわち、クライアント５が使用するのと同じ期間）より長い間、リンクがアイドルであったことを（データ検出器コンポーネント７０’およびタイマ・コンポーネント３０’により）検出したときにＳＳＬサーバが「ハートビート」を受信しない場合、ＳＳＬサーバは接続ターミネータ（connectionterminator）８５によりその接続をクローズする。これは、秘密鍵再交渉を妨げて秘密鍵の存続時間を延長するためにハッカが「ハートビート」メッセージを消費するのを防止するものである。

【００５９】

いずれのアプリケーション・データも損なわないので、送信偽装ハートビートを検出する必要はまったくないことに留意されたい。

【００６０】

ＳＳＬサーバ６が存在し、クライアントからハートビートを受信した場合、ＳＳＬサーバ６は、特殊な「ハートビート」メッセージに応答し、ハートビートが通信リンクを越えて流れるのに十分なほど長い間、接続がアイドルであったことを記憶に留める。ＳＳＬクライアントは、それが依然として存在することを確認するために、任意の数の「ハートビート」メッセージを送信することができる。

【００６１】
「ハートビート」メッセージは、バイトしきい値によって秘密鍵をトリガしなければならない時期を計算する際に使用されるバイト合計に寄与しないことに留意されたい。
【００６２】
クライアントがサーバから応答を受信すると、図２のプロセスはステップ１２０にループする。リンクにより流すべきデータが存在すると判断された場合、リンクがハートビートの発生を引き起こすのに十分なほど、以前アイドルであったかどうかがもう一度、ステップ１３０でテストされる。答えがＹＥＳである場合、秘密鍵について再交渉しなければならない。したがって、再認証および鍵の交渉が実施されるまで、クライアントはそれ以上いずれのデータもサーバに送信しなくなる。
【００６３】
これは、通信リンクがアイドルであった時間が長時間であるためにハッカが何とか秘密鍵を破った場合に、その鍵がもはやそれらにとって役に立たないものであることを意味する。
【００６４】
応答がまったく受信されない場合、クライアントは接続ターミネータ５５を使用してその接続をクローズする（ステップ１８０）。これは、サーバがもはや存在しないかまたは誰か他の人がサーバの応答を消費していることを応答の失敗が示しているからである。
【００６５】
代替一実施形態では、クライアントは、接続を終了する前に追加の回数、サーバに連絡を取ろうと試みることができることに留意されたい。これは、サーバからの応答の欠如が一時的な問題に過ぎない可能性があるからである。大事を取るために、再認証／鍵の交渉を開始することができる。
【００６６】
ハートビート間のタイミング（アイドル・リンク上で２つ以上が送信される場合）は好ましくは一定であることに留意されたい。各ハートビート・メッセージ間でランダム・タイミングが使用される場合、ハートビートが期限切れになった（おそらくハッカによって消費された）時期を予測することは可能ではないであろう。
【００６７】
当然のことながら、ハートビートが最初に送信される前の時間（およびハートビート間の間隔）とバイトフロー・カウントは好ましくは、同じ秘密鍵が長時間の間、使用中のものとして存続しないように選択されることが分かるであろう。選択された値が、秘密鍵を収集し発見するための時間をハッカに提供するほど十分高いものである場合、「送信偽装」メッセージは依然として達成可能なものでありうることに留意されたい。しかし、ハートビート・トリガの秘密鍵再交渉が行われると、ハッカはもはやサーバを欺すことができなくなる。このため、データ送信側が新しい鍵の交渉を開始することが好ましい。さもなければ、正しく暗号化された送信偽装メッセージをサーバが受信している場合、サーバの見地から、再交渉は不要である。
【００６８】
これまでに述べた解決策を使用する場合の重要な利点が４つある。
（ｉ）この提案は、安全な状態で存続しながら最適パフォーマンスを達成するためにアイドル通信リンク上で絶対に必要であるときのみ、再認証および秘密鍵の再交渉が発行されることを保証するものである。
（ｉｉ）正しい秘密鍵で暗号化された場合でも「送信偽装」メッセージを検出できることは、「ハートビート」メッセージの使用によって提供されるものであり、これはデータ通信が再開されたときに秘密鍵が再交渉されるからである。
（ｉｉｉ）この提案は、損なわれた秘密鍵でハッカによって読み取られる可能性のあるアプリケーション・データの量を制限するために使用中通信リンク上で秘密鍵が規則正しく変更されることを保証するものである。
（ｉｖ）特殊な「ハートビート」メッセージは、アプリケーション・データを含まず、こ

のため、データを暗号化および暗号化解除するために使用された秘密鍵が暴力によって発見可能である場合でも、ハッカにとって無用なものである。

【００６９】

このプロトコルは、好ましくは、１つまたは複数のハートビートがリンクを越えて流れるように十分長い時間の間、通信リンクがアイドルであった後で、特定の数のバイトが通信リンクを越えて流れたときに、認証および鍵の交渉が必ず実行されることを保証するものである。

【００７０】

現行の合意を得た秘密鍵を発見した可能性がある場合でも、「送信偽装」メッセージを送信しているハッカは、（クライアントの）再認証および鍵の交渉（いずれについても再認証の証明書を持っていない）を開始するための非対称秘密鍵を所有していないので、合意を得たプロトコルに従うことができない。さらに、古い対称秘密鍵は、サーバがクライアントからのハートビートを確認した瞬間から無効になる。

【００７１】

この解決策は、再認証および鍵の交渉を不必要に実行する必要なしに、ハッカがアイドル通信リンク上で「送信偽装」メッセージを送信するのを効果的に防止する。また、この解決策は、使用中通信リンクにも対処するものである。

【００７２】

正常な再認証を行うために、ＳＳＬクライアントがＳＳＬサーバに認証情報を提示する必要がないように、ＳＳＬを構成することが可能であり、この場合、サーバがクライアントに対して認証されるだけであり、逆は行われないことに留意されたい。しかし、これは、第三者がクライアントであるふりをし、そのようにサーバと通信することを可能にする恐れがあるので、安全なピアツーピア環境では賢明なことではない。

【００７３】

メッセージング環境に特に適用可能なものとして本発明を説明してきたが、このようなものに限定することが意図されているわけではないことに留意されたい。本発明は、アイドル時間と使用中時間との間で変動する環境であれば、どのような環境にも適用可能である。

【００７４】

さらに、ＳＳＬ暗号化プロトコルに関して本発明を説明してきたが、この場合も、このような限定はまったく意図されていない。しかし、本発明は、認証および鍵の交渉がプロセッサ集中型である環境であれば、どのような環境にも特に適用可能である。他の例はＴＬＳである。

【００７５】

この例示的な実施形態では、データはクライアントからサーバに流れていることに留意されたい。これは、そうでなければならないわけではなく、データは反対方向に流れることもできる。好ましくは、データを送信する人であれば誰でも、認証および鍵の交渉を開始し、ハートビートも送信する。

【００７６】

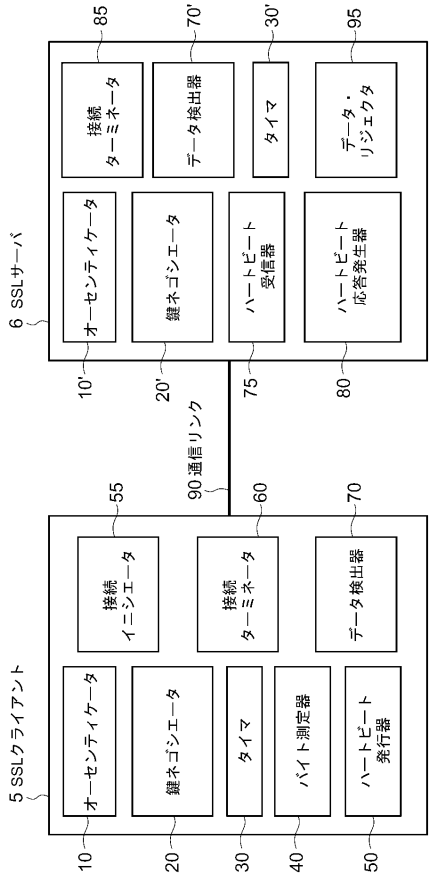
代替一実施形態では、認証および鍵の再交渉は必ずＳＳＬクライアントによって開始される。したがって、ＳＳＬサーバが送信すべきデータを有する場合、サーバは、まず認証し再交渉するよう、ＳＳＬクライアントに依頼する。反対も当てはまる可能性がある。

【００７７】

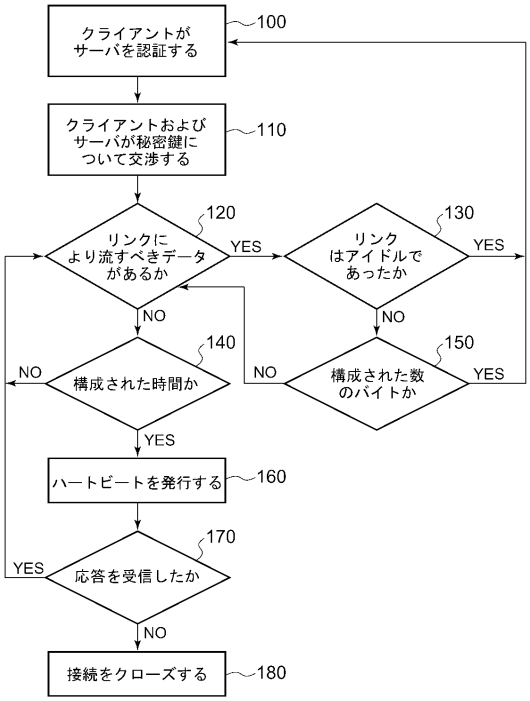
機会ごとに初期完全ハンドシェーク（非対称認証）を実行し、次に秘密鍵の交渉を実行することに関して好ましい実施形態を説明してきたが、これは、そうでなければならないわけではないことに留意されたい。認証に続いて鍵の交渉が行われることは特にプロセッサ集中型になるので、本発明はこの状況において特に適用可能である。しかし、本発明は、好ましくは、セッション・キャッシングを使用する環境（あまりプロセッサ集中ではない）でも適用可能である。これは、たとえば、ＳＳＬ　ｖ３．０およびＴＬＳにおいて使用可能な特徴である。

【００７８】
セッション・キャッシングは、初期ハンドシェーク中に実行することができる。クライアントとサーバは、共通セッションＩＤと、マスタ秘密鍵と、何らかの証明書チェーンを保管する。この情報は、通常、構成可能な時間の間、キャッシュに保持される。
【００７９】
後続ハンドシェークが要求され（すなわち、クライアントが新しい秘密鍵を要求した場合）、この情報がキャッシュから消失していない場合、両方がそれぞれのセッションＩＤを互いに提示する。そのセッションＩＤ同士が一致する場合、キャッシュされた情報を使用して、ハンドシェーク中に実行される処理を削減することになり、これは、完全ハンドシェークとは対照的に、一般に短縮ハンドシェークと呼ばれる。
【００８０】
セッション・キャッシングの使用に関する弱点は、ハッカがハンドシェークに応答するときに元のセッションＩＤを提示するだけでよい（いかなる証明書も交換されず、いかなる公開鍵操作も行われない）ことである。セッションＩＤはクライアントの「ハロー」フローに含まれ、したがって、電信から詮索することが可能である。
【００８１】
データは一方向のみに流れる必要はなく、データは両方向に流れることができることに留意されたい。このシナリオでは、好ましくは、秘密鍵の再交渉が必要になったときに送信すべきデータを有する人であれば誰でも、秘密鍵の再交渉を開始する。好ましくは、両端のうちの一方は、ハートビートの送信を担当するものとして指定される（すなわち、少なくとも所定時間、いかなるデータもいずれかの方向に流れなかった後）。したがって、ハートビートおよびそれに対する応答は、両端の存在を決定するために使用される。使用されるバイト・カウントは、好ましくは、特定の時間中に通信リンクにより送信されたすべてのデータの総計であり、すなわち、両端によって送信されたデータを含む。一実施形態では、一方は、バイト・カウントおよびリンクのアイドル状態を追跡し、２つのしきい値のいずれかを満足すると、もう一方に通知する。
【図面の簡単な説明】
【００８２】
【図１】本発明の好ましい一実施形態によるクライアント／サーバ・コンポーネント図である。
【図２】本発明の好ましい一実施形態によりクライアントによって実行される処理の流れ図である。[MSOffice1]正確には期間ですが、意味的に請求項３に合わせて「時間」と訳しました。他の請求項も同様。

【図１】


6 SSLサーバ
85
70'
30'
95
接続ターミネータ
データ検出器
タイマ
データ・リジェクタ
オーセンティケータ
鍵ネゴシエータ
ハートビート受信器
ハートビート応答発生器
10'
20'
75
80
90 通信リンク
5 SSLクライアント
55
60
70
接続イニシエータ
接続ターミネータ
データ検出器
オーセンティケータ
鍵ネゴシエータ
タイマ
バイト測定器
ハートビート発行器
10
20
30
40
50

【図２】


クライアントがサーバを認証する
100
クライアントおよびサーバが秘密鍵について交渉する
110
リンクにより流すべきデータがあるか
120
YES
NO
リンクはアイドルであったか
130
YES
NO
構成された時間か
140
NO
YES
構成された数のバイトか
150
NO
YES
ハートビートを発行する
160
応答を受信したか
170
YES
NO
接続をクローズする
180

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 ハラン、リチャード、マイケル、ウィン
イギリス国エス・オー１５ ２エイチ・ジェイ ハンプシャー州サザンプトン ウィルトン・アヴェニュー２ フラット＃３
(72)発明者 ホラン、マイケル
イギリス国エス・オー２２ ４ジェイ・エイ ハンプシャー州ウィンチェスター オリヴァーズ・バッテリー・ロード・ノース１２
(72)発明者 ラムゼイ、ジョナサン
イギリス国エス・オー２３ ９ディ・エイチ ハンプシャー州ウィンチェスター トラファルガー・ストリート ウェストゲイト・ハウス８

審査官 青木 重徳

(56)参考文献 特開平０２－１６４１５４（ＪＰ，Ａ）
特開平０５－１１０５７１（ＪＰ，Ａ）
特開平０９－２６９７２７（ＪＰ，Ａ）
特開平１１－３１３０７７（ＪＰ，Ａ）
特開平０７－１１５４１３（ＪＰ，Ａ）
特開平０５－００３４７８（ＪＰ，Ａ）
特開平０５－２０６８７５（ＪＰ，Ａ）
特開昭５８－０５６５５２（ＪＰ，Ａ）
米国特許第０６３６０２６９（ＵＳ，Ｂ１）
Ron Mraz, “Secure Blue: An Architecture for a Scalable, Reliable High Volume SSL Internet Server”, Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, ２００１年１２月１０日, p.391-398, ＵＲＬ, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=991556

(58)調査した分野(Int.Cl., ＤＢ名)
H04L 9/08
H04L 9/14