(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2004-515117
(P2004-515117A)
(43)公表日　平成16年5月20日(2004.5.20)

(51)Int.Cl.[7]	F I		テーマコード（参考）
H04L 9/32	H04L 9/00	673D	5B085
G06F 15/00	G06F 15/00	330B	5J104
H04L 9/08	H04L 9/00	601B	

審査請求 未請求　予備審査請求 有　（全 64 頁）

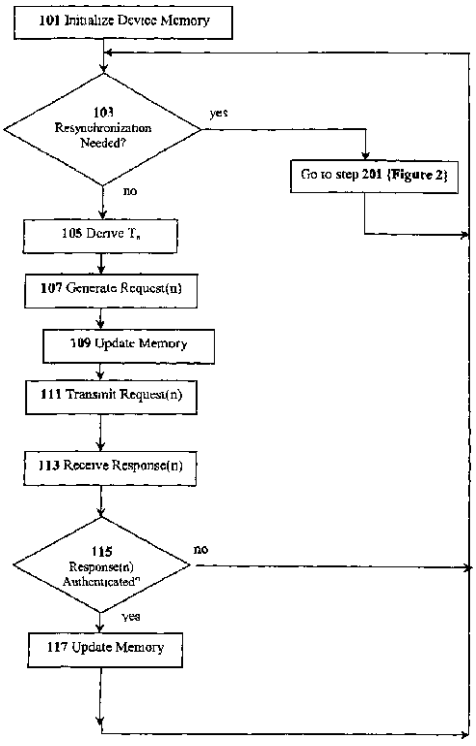
(21)出願番号　特願2002-544911(P2002-544911)
(86)(22)出願日　平成13年10月19日(2001.10.19)
(85)翻訳文提出日　平成15年4月21日(2003.4.21)
(86)国際出願番号　PCT/US2001/046290
(87)国際公開番号　WO2002/043309
(87)国際公開日　平成14年5月30日(2002.5.30)
(31)優先権主張番号　60/242,083
(32)優先日　平成12年10月20日(2000.10.20)
(33)優先権主張国　米国(US)
(31)優先権主張番号　60/246,843
(32)優先日　平成12年11月8日(2000.11.8)
(33)優先権主張国　米国(US)

(71)出願人　503147158
ウェイヴ・システムズ・コーポレーション
アメリカ合衆国・マサチューセッツ・０１２３８・リー・プレザント・ストリート・４８０・スィート・Ｂ２００
(74)代理人　100064908
弁理士　志賀　正武
(74)代理人　100108578
弁理士　高橋　詔男
(74)代理人　100089037
弁理士　渡邊　隆
(74)代理人　100101465
弁理士　青山　正和
(74)代理人　100094400
弁理士　鈴木　三義

最終頁に続く

(54)【発明の名称】暗号化データセキュリティシステムおよび方法

(57)【要約】
本発明において、コンピュータ装置と信頼できるサーバーとの間で通信するための方法が開示される。本発明の方法によれば、装置（１０５）からサーバーへの通信において用いるためのワンタイムパスワードが生成される。装置（１０５）は、装置（１０５）からの以前のメッセージに対する、サーバーからの以前の応答（１１３）の少なくとも一部の機能を含む少なくとも１つのワンタイム要請－認証データ（１０７）を生成する。次に、サーバーは、少なくとも１つのワンタイムパスワードの少なくとも一部の機能を含む少なくとも１つのワンタイム応答－認証データ（１１３）を生成する。



【特許請求の範囲】
【請求項１】
クライアント装置と信頼できるサーバーとの間の通信における信頼性を高めるための方法であって、
（ａ）装置からサーバーへの通信において用いるためのワンタイムパスワードを生成する段階と、
（ｂ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階と、
（ｃ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階と
を具備することを特徴とする方法。
【請求項２】
前記ワンタイム要請－認証データは、暗号化鍵の機能を具備することを特徴とする請求項１に記載の方法。
【請求項３】
前記ワンタイム応答－認証データは、暗号化鍵の機能を具備することを特徴とする請求項１に記載の方法。
【請求項４】
クライアント装置からのデータ要請を通信することにおける信頼性を高めるための方法であって、
（ａ）ワンタイムパスワードを生成する段階と、
（ｂ）装置からの以前の要請に対する、信頼できるサーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階と
を具備することを特徴とする方法。
【請求項５】
前記ワンタイム要請－認証データは、暗号化鍵の機能を具備することを特徴とする請求項４に記載の方法。
【請求項６】
クライアント装置から信頼できるサーバーへの要請からの応答を通信することにおける信頼性を高めるための方法であって、
（ａ）装置と前記サーバーとの間で共有される少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する要請を受信する段階と、
（ｂ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階と
を具備することを特徴とする方法。
【請求項７】
前記ワンタイム応答－認証データは、暗号化鍵の機能を具備することを特徴とする請求項６に記載の方法。
【請求項８】
前記要請は、暗号化された秘密データを具備し、
前記サーバーは、前記秘密データを回復するために、前記暗号化された秘密データを解読することを特徴とする請求項６に記載の方法。
【請求項９】
その後の要請は、少なくとも１つの秘密データの少なくとも一部を具備する少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備することを特徴とする請求項８に記載の方法。
【請求項１０】
前記要請内に具備されるワンタイムパスワードは、サーバーにより、そのデータベース内において、特定のクライアント装置に対応する入力を見つけるために用いられることを特徴とする請求項６に記載の方法。

【請求項１１】
クライアント装置と信頼できるサーバーとの間の通信を再同期化するための方法であって、
（ａ）装置からサーバーへの通信において用いるためのワンタイムパスワードを供給する段階と、
（ｂ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを供給する段階と、
（ｃ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを供給する段階と
を具備することを特徴とする方法。
【請求項１２】
クライアント装置からの再同期化要請の通信における信頼性を高めるための方法であって、
（ａ）ワンタイムパスワードを供給する段階と、
（ｂ）装置からの要請に対する、信頼できるサーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを供給する段階と
を具備することを特徴とする方法。
【請求項１３】
前記再同期化要請は、以前の再同期化データに取って代わる暗号化された再同期化データを具備することを特徴とする請求項１２に記載の方法。
【請求項１４】
信頼できるサーバーからの再同期化応答の送信における信頼性を高めるための方法であって、
（ａ）クライアント装置と関連したワンタイムパスワードを具備する要請を受信する段階と、
（ｂ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを供給する段階と
を具備することを特徴とする方法。
【請求項１５】
前記再同期化応答は、以前の再同期化データに取って代わる暗号化された再同期化データを具備することを特徴とする請求項１４に記載の方法。
【請求項１６】
クライアント装置と信頼できるサーバーとの間の通信における信頼性を高めるためのシステムであって、
（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、
（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段と
を具備し、
前記装置とサーバーとの間の通信は、
（ｉ）装置からサーバーへの通信において用いるためのワンタイムパスワードを生成する段階と、
（ｉｉ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階と、
（ｉｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階と
とを具備する方法にしたがって実施されることを特徴とするシステム。
【請求項１７】
（ａ）暗号化アルゴリズムと、
（ｂ）前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段と
をさらに具備し、

前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、
前記装置とサーバーとの間の通信は、暗号化ベースで実施されることを特徴とする請求項１６に記載のシステム。
【請求項１８】
前記ワンタイム要請－認証データは、暗号化鍵の機能を具備することを特徴とする請求項１６に記載のシステム。
【請求項１９】
前記ワンタイム応答－認証データは、暗号化鍵の機能を具備することを特徴とする請求項１６に記載のシステム。
【請求項２０】
クライアント装置からのデータ要請を通信することにおける信頼性を高めるためのシステムであって、
（ａ）クライアント装置と信頼できるサーバーとの間のネットワーク接続を確立するための手段と、
（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、
前記装置とサーバーとの間の通信は、
（ｉ）ワンタイムパスワードを生成する段階と、
（ｉｉ）装置からの以前の要請に対する、信頼できるサーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階と
を具備する方法にしたがって実施されることを特徴とするシステム。
【請求項２１】
（ａ）暗号化アルゴリズムと、
（ｂ）前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段と
をさらに具備し、
前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、
前記装置とサーバーとの間の通信は、暗号化ベースで実施されることを特徴とする請求項２０に記載のシステム。
【請求項２２】
前記ワンタイム要請－認証データは、暗号化鍵の機能を具備することを特徴とする請求項２０に記載のシステム。
【請求項２３】
クライアント装置から信頼できるサーバーへの要請からの応答を通信することにおける信頼性を高めるためのシステムであって、
（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、
（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、
前記装置とサーバーとの間の通信は、
（ｉ）装置と前記サーバーとの間で共有される少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する要請を受信する段階と、
（ｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階と
を具備する方法にしたがって実施されることを特徴とするシステム。
【請求項２４】
（ａ）暗号化アルゴリズムと、
（ｂ）前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピ

ュータへダウンロードするための手段と
をさらに具備し、
前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、
前記装置とサーバーとの間の通信は、暗号化ベースで実施されることを特徴とする請求項２３に記載のシステム。
【請求項２５】
前記ワンタイム応答－認証データは、暗号化鍵の機能を具備することを特徴とする請求項２３に記載のシステム。
【請求項２６】
前記要請は、暗号化された秘密データを具備し、
前記サーバーは、前記秘密データを回復するために、前記暗号化された秘密データを解読することを特徴とする請求項２３に記載のシステム。
【請求項２７】
その後の要請は、少なくとも１つの秘密データの少なくとも一部を具備する少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備することを特徴とする請求項２６に記載のシステム。
【請求項２８】
前記要請内に具備されるワンタイムパスワードは、サーバーにより、そのデータベース内において、特定のクライアント装置に対応する入力を見つけるために用いられることを特徴とする請求項２３に記載のシステム。
【請求項２９】
クライアント装置と信頼できるサーバーとの間の通信を再同期化するためのシステムであって、
（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、
（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段と
を具備し、
前記装置とサーバーとの間の通信は、
（ｉ）装置からサーバーへの通信において用いるためのワンタイムパスワードを供給する段階と、
（ｉｉ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを供給する段階と、
（ｉｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを供給する段階と
を具備する方法にしたがって実施されることを特徴とするシステム。
【請求項３０】
（ａ）暗号化アルゴリズムと、
（ｂ）前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段と
をさらに具備し、
前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、
前記装置とサーバーとの間の通信は、暗号化ベースで実施されることを特徴とする請求項２９に記載のシステム。
【請求項３１】
クライアント装置からの再同期化要請の通信における信頼性を高めるためのシステムであって、
（ａ）クライアント装置と信頼できるサーバーとの間のネットワーク接続を確立するための手段と、
（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段と

を具備し、
前記装置とサーバーとの間の通信は、
（ｉ）ワンタイムパスワードを供給する段階と、
（ｉｉ）装置からの要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを供給する段階と
を具備する方法にしたがって実施されることを特徴とするシステム。
【請求項３２】
（ａ）暗号化アルゴリズムと、
（ｂ）前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段と
をさらに具備し、
前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、
前記装置とサーバーとの間の通信は、暗号化ベースで実施されることを特徴とする請求項３１に記載のシステム。
【請求項３３】
前記再同期化要請は、以前の再同期化データに取って代わる暗号化された再同期化データを具備することを特徴とする請求項３１に記載のシステム。
【請求項３４】
信頼できるサーバーからの再同期化応答の送信における信頼性を高めるためのシステムであって、
（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、
（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段と
を具備し、
前記装置とサーバーとの間の通信は、
（ｉ）クライアント装置と関連したワンタイムパスワードを具備する要請を受信する段階と、
（ｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを供給する段階と
を具備する方法にしたがって実施されることを特徴とするシステム。
【請求項３５】
（ａ）暗号化アルゴリズムと、
（ｂ）前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段と
をさらに具備し、
前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、
前記装置とサーバーとの間の通信は、暗号化ベースで実施されることを特徴とする請求項３４に記載のシステム。
【請求項３６】
前記再同期化応答は、以前の再同期化データに取って代わる暗号化された再同期化データを具備することを特徴とする請求項３４に記載のシステム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、データ通信システムにおけるセキュリティの改善に関し、より詳細には、インターネットのような、安全ではない、または、安全か否かが疑わしいネットワークを介して送信され得るデータについての機密性と、信頼と、対攻撃抵抗力（ａｔｔａｃｋ－ｒｅｓｉｓｔａｎｃｅ）とを提供するためのシステムおよび方法に関する。
【０００２】

【従来の技術】
データ通信（明確には、例えば、分散型データネットワークを介しての複数のコンピュータユーザー間における通信）は、通信当事者により認可されていないか、または、データ受信者として意図されていない人物（以後、“インサイダー”または“傍受者（ｉｎｔｅｒｃｅｐｔｏｒｓ）”と称する）による多数の種類の攻撃にさらされているものと認識されている。このような攻撃は、プライベート情報を閲覧したいという欲求や、金銭上または他の詐欺行為を働きたいという欲求や、または、理由は何であれ単に通信の完全性を損なわせたいという欲求により動機づけられ得る。

【０００３】
本明細書およびクレームにおける“ワンタイム（ｏｎｅ－ｔｉｍｅ）”という語の使用は、或る手段を指定しかつ或るパスワードおよびデータの動的な更新または置換の結果に順応するための使用可能な（ｅｎｅｂｌｅｄ）能力を反映するように意図される。装置またはサーバーの観点からの、このような“ワンタイム”値の受容可能な再使用の程度は、特定の実施手段により決定され、かつ、本明細書内では規定されない。

【０００４】
サーバーコンピュータと、前記サーバーからのデータへのアクセス権を有する１つ以上のクライアントコンピュータとを含むネットワークという状況において（例えば、ワールドワイドウェブベースのウェブブラウザという状況において）、１９９９年２月３日にカリフォルニア州サンディエゴでのＮｅｔｗｏｒｋ ａｎｄ Ｄｉｓｔｒｉｂｕｔｅｄ Ｓｙｓｔｅｍ Ｓｅｃｕｒｉｔｙ Ｓｙｍｐｏｓｉｕｍにおいて最初に提示された、Ａ．ジュエルズ（Ｊｕｅｌｓ，Ａ．）およびＪ．ブレイナード（Ｂｒａｉｎａｒｄ，Ｊ．）のＣｌｉｅｎｔ Ｐｕｚｚｌｅｓ：Ａ Ｃｒｙｐｔｏｇｒａｐｈｉｃ Ｃｏｕｎｔｅｒｍｅａｓｕｒｅ ａｇａｉｎｓｔ Ｃｏｎｎｅｃｔｉｏｎ Ｄｅｐｌｅｔｉｏｎ Ａｔｔａｃｋ，ｈｔｔｐ：／／ｗｗｗ．ｒｓａｓｅｃｕｒｉｔｙ．ｃｏｍ／ｒｓａｌａｂｓ／ｓｔａｆｆ／ａｊｕｅｌｓ，１９９９（本明細書内で、以後“ジュエルズおよびブレイナード”と称する）（この開示内容は、参照により本明細書に編入される）において定義されるような接続消耗攻撃（ｃｏｎｎｅｃｔｉｏｎ ｄｅｐｌｅｔｉｏｎ ａｔｔａｃｋ）は、攻撃者が、サーバーに対する多数の接続（または、サービス）を開始しかつこれらを解決されないままにし、そのリソースを使い果たしかつ正規の要請にサービスを提供することを不可能にさせることを求めるものである。

【０００５】
接続消耗攻撃に対する抵抗力を高めるために、従来技術において種々の試みが行われてきた。

【０００６】
ジュエルズおよびブレイナードは、クライアントの部類を区別せずに、このタイプのサービス拒否（ｄｅｎｉａｌ－ｏｆ－ｓｅｒｖｉｃｅ）問題に取り組む。ジュエルズおよびブレイナードは、部外者の侵入を阻止するために動的に変更される暗号化“パズル”を用いる。

【０００７】
ｈｔｔｐ：／／ｗｗｗ．ｒｓａｓｅｃｕｒｉｔｙ．ｃｏｍ／ｐｒｏｄｕｃｔｓ／ｓｅｃｕｒｉｄ／ｄａｔａｓｈｅｅｔｓ／ｄｓａｕｔｈｅｎｔｉｃａｔｏｒｓ．ｈｔｍｌ（本明細書内で、以後“Ｄｓ認証子（Ｄｓａｕｔｈｅｎｔｉｃａｔｏｒｓ）”と称する）に公開されている他のアプローチは、ＳｅｃｕｒＩＤ認証子（ａｕｔｈｅｎｔｉｃａｔｏｒｓ）を用いる。これらは、専有的（ｐｒｏｐｒｉｅｔａｒｙ）アルゴリズムという状況において連続的に適用されるトークン独自の（ｔｏｋｅｎ－ｕｎｉｑｕｅ）鍵に基づいて一連のワンタイムパスワードを各々が提供するハードウェアまたはソフトウェアトークン（ｔｏｋｅｎ）である。クライアント側のホストは、現在のワンタイムパスワードと、一定のＰＩＮまたはパスフレーズ（ｐａｓｓｐｈｒａｓｅ）を、該ホストが自身の身元確認を行うことを希望する対象であるサーバーへ送信する。トークン独自の鍵についての知識（ｋｎｏｗｌｅｄｇｅ）を有するサーバーは、クライアントのトークンと同期することができ、こ

れにより、ワンタイムパスワードおよびＰＩＮの受信に基づいて特定のクライアントの（遠隔的）存在を認識することができる。これは、自己同期的（ｓｅｌｆ－ｓｙｎｃｈｒｏｎｉｚｉｎｇ）システムであり、この場合に、クライアントのトークンは、トランザクション毎の（ｐｅｒ－ｔｒａｎｓａｃｔｉｏｎ ｂａｓｉｓ）サーバーからの入力に基づいて、自身の動きを適応させない。さらに、このシステムは、エンティティの認証を提供するように設計されているが、これは、出所（ｏｒｉｇｉｎ）または完全性（ｉｎｔｅｇｒｉｔｙ）の認証ではなく、また、その後の任意の通信の“新鮮度（ｆｒｅｓｈｎｅｓｓ）”の認証でもない。

【０００８】

Ｒ．リベスト（Ｒｉｖｅｓｔ，Ｒ．）、Ａ．シャミア（Ｓｈａｍｉｒ，Ａ．）、Ｌ．エイドルマン（Ａｄｌｅｍａｎ，Ｌ．）のＡ Ｍｅｔｈｏｄ ｆｏｒ Ｏｂｔａｉｎｉｎｇ Ｄｉｇｉｔａｌ Ｓｉｇｎａｔｕｒｅｓ ａｎｄ Ｐｕｂｌｉｃ－ｋｅｙ Ｃｒｙｐｔｏｓｙｓｔｅｍｓ；Ｃｏｍｍｕｎｉｃａｔｉｏｎｓ ｏｆ ｔｈｅ Ａ．Ｃ．Ｍ．１９７８，２１，１２０－２６（本明細書内で、以後“リベスト、シャミア、および、エイドルマン”と称する）に記載され、かつ、Ｍ．ベラーレ（Ｂｅｌｌａｒｅ，Ｍ．）およびＰ．ロガウェイ（Ｒｏｇａｗａｙ，Ｐ．）のＯｐｔｉｍａｌ Ａｓｙｍｍｅｔｒｉｃ Ｅｎｃｒｙｐｔｉｏｎ － Ｈｏｗ ｔｏ Ｅｎｃｒｙｐｔ ｗｉｔｈ ＲＳＡ（１９９５年１１月１９日）（Ｏｐｔｉｍａｌ Ａｓｙｍｍｅｔｒｉｃ Ｅｎｃｒｙｐｔｉｏｎ Ｐａｄｄｉｎｇの論文ｈｔｔｐ：／／ｗｗｗ．ｃｓｅ．ｕｃｓｄ．ｅｄｕ／ｕｓｅｒｓ／ｍｉｈｉｒ／ｐａｐｅｒｓ／ｏａｅｐ．ｈｔｍｌ、および、Ａｄｖａｎｃｅｓ ｉｎ Ｃｒｙｐｔｏｌｏｇｙ － Ｅｕｒｏｃｒｙｐｔ ９４，Ｌｅｃｔｕｒｅｓ ｉｎ Ｃｏｍｐｕｔｅｒ Ｓｃｉｅｎｃｅ，Ａ．ＤｅＳａｎｔｉｓ Ｅｄ．，Ｓｐｒｉｎｇｅｒ Ｖｅｒｌａｇ，１９９４，９５０，９２－１１１内で公開された旧版の改訂版）（本明細書内で、以後“ベラーレおよびロガウェイ”と称する）に基づいて強化され、Ｄ．Ｂ．ジョンソン（Ｊｏｈｎｓｏｎ，Ｄ．Ｂ．）およびＳ．Ｍ．マチャーシュ（Ｍａｔｙａｓ，Ｓ．Ｍ．）のＡｓｙｍｍｅｔｒｉｃ Ｅｎｃｒｙｐｔｉｏｎ：Ｅｖｏｌｕｔｉｏｎ ａｎｄ Ｅｎｈａｎｃｅｍｅｎｔｓ，ＣｒｙｐｔｏＢｙｔｅｓ，Ｓｐｒｉｎｇ １９９６，Ｖｏｌｕｍｅ ２，Ｎｏ．１（ｈｔｔｐ：／／ｗｗｗ．ｒｓａｌａｂｓ．ｃｏｍ／ｃｒｙｐｔｏｂｙｔｅｓをさらに参照）（本明細書内で、以後“ジョンソンおよびマチャーシュ”と称する）においてさらに説明されるような方法は、２人の当事者が自分たちの通信の機密性を保証するための手段を提供し、この場合に、送信側は、暗号化の目的のために受信側の公開鍵（ｐｕｂｌｉｃ ｋｅｙ）を用い、かつ、受信側は、解読（プレーンテキスト（ｐｌａｉｎｔｅｘｔ）の回復）の目的のために、その対応する秘密鍵（ｐｒｉｖａｔｅ ｋｅｙ）を用いる。この方法は、２人の当事者は、アルゴリズム上は関連づけられたりまたは一組にされたりしているが互いに別個の鍵を用いるという点で非対称である。リベスト、シャミア、および、エイドルマンの方法については、ディジタル署名能力を例示化するために用いることもでき、この場合には、署名側は、この方法にしたがって、署名すべきメッセージに自分の秘密鍵を適用し、かつ、検証側は、該メッセージの出所および完全性の信憑性を検証する（ｖｅｒｉｆｙ）ために、この方法にしたがって、対応する公開鍵を適用する。ディジタル署名は、それ自体は、新鮮度の証拠を提供しない（すなわち、以前に用いられたメッセージについては、“古い（ｓｔａｌｅ）”メッセージとして検出されずに再生（ｒｅｐｌａｙ）することができる）。

【０００９】

２人の当事者は、ＤＥＳのような対称鍵暗号化アルゴリズムを用いて通信することができる。この場合に、ＤＥＳについては、メッセージ認証コード（ｍｅｓｓａｇｅ ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｃｏｄｅ：ＭＡＣ）能力を提供するために用いることもできる。これにより、秘密鍵の知識を有する受信側は、メッセージの発信者も秘密鍵についての知識を有しておりかつメッセージが送信中に変更されていないと判断することができる。

【００１０】

メッセージまたはその一部については、クライアントの身元をサーバー以外の当事者から

隠すために、かつ、クライアントの身元を同じクライアントから生じたことがあるトランザクションと関連づけることをより困難にするために暗号化することができる。この場合に、サーバーは、当事者の身元についての知識を必要とするあらゆる処理を実行する前に、解読アルゴリズムを適用する必要がある。ディジタル署名がメッセージに適用される場合に、不正確な公開鍵を用いて検証されたメッセージは検証を失敗するので、敵対者は、署名者による通信トランザクションをグループ化するために公開鍵のリストを用い得る。署名が暗号化される場合に、または、メッセージが暗号化された形式で送信されるプレーンテキストメッセージを介して署名が計算される場合に、署名の検証は、予備的な解読を必要とする。

【００１１】

【発明が解決しようとする課題】

したがって、署名の検証を必要としない安全な通信方法の必要性が存在する。サーバーが、トークン独自の鍵、および、クライアントトークンと同期するためのタイマーまたはカウンターのようなワンタイムパスワード更新アルゴリズムへの自動制御的（ｓｅｌｆ－ｒｅｇｕｌａｔｅｄ）入力についての知識を必要としないように、自己同期化しない安全な方法の必要性がさらに存在する。クライアント識別情報への認可されていないアクセスを可能にすることによりプライバシーを弱めることのない方法の必要性がさらに存在する。最後に、プロトコルによるパターンを送信できる登録されたクライアントの装置を利用する方法の必要性が存在し、この場合に、サーバーは、このようなパターンを、他の入力してくるインターネットトラフィックと区別することができる。従来技術は、これらの必要性を満たすものとは考えられない。

【００１２】

【課題を解決するための手段】

本発明は、コンピュータ装置と信頼できるサーバーとの間で通信するための方法に向けられる。前記方法は、（ａ）装置からサーバーへの通信において用いるためのワンタイムパスワードを生成する段階と、（ｂ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を含む少なくとも１つのワンタイム要請－認証データを生成する段階と、（ｃ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を含む少なくとも１つのワンタイム応答－認証データを生成する段階とを含む。前記ワンタイム要請－認証データまたは前記ワンタイム応答－認証データ、または、これらの両方は、暗号化鍵の機能を具備することが好ましい。装置からサーバーへの通信においてワンタイムパスワードが要請と関連したものとして“用いられる”時点において、ワンタイムパスワードは傍受にさらされ得るが、その一方で、ワンタイムパスワードに対する応答－認証データの依存関係は、まだそのように用いられたことがないワンタイムパスワードに関するものである。したがって、応答メッセージの送信については、その後の要請中にワンタイムパスワードを実際に使用することに先立つワンタイムパスワードの（安全な）ネゴシエーションまたは交換の一部と見なすことができる。ワンタイムパスワードの情報についての知識を、対応する解読を実行する能力を備えたサーバーへ伝達する目的のために、要請メッセージ内においてワンタイムパスワードまたはその構成要素を暗号化して送信することは、ワンタイムパスワードの使用とは見なされない。サーバーから装置への応答の傍受は、ワンタイム要請－認証データの生成または検証の成功を可能にしない。装置からサーバーへの要請の傍受は、ワンタイム応答－認証データの生成または検証の成功を可能にしない。

【００１３】

本発明の他の目的は、クライアント装置からのデータ要請を送信するための方法であって、（ａ）ワンタイムパスワードを生成する段階と、（ｂ）装置からの以前の要請に対する、信頼できるサーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階とを具備する方法を提供することである。前記ワンタイム要請－認証データは、暗号化鍵の機能を具備することが好ましい。

【００１４】

本発明の他の目的は、クライアント装置からの要請に対する、信頼できるサーバーからの応答を送信するための方法であって、（ａ）装置と前記サーバーとの間で共有される少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する要請を受信する段階と、（ｂ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階とを具備する方法を提供することである。前記ワンタイム応答－認証データは、暗号化鍵の機能を具備することが好ましい。

【００１５】

本発明の他の目的は、クライアント装置と信頼できるサーバーとの間の通信における信頼性を高めるためのシステムであって、（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、前記装置とサーバーとの間の通信は、（ｉ）装置からサーバーへの通信において用いるためのワンタイムパスワードを生成する段階と、（ｉｉ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階と、（ｉｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階ととを具備する方法にしたがって実施されるシステムを提供することである。前記システムは、暗号化アルゴリズムと、前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段とをさらに具備し、前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、前記装置とサーバーとの間の通信は、暗号化ベースで実施されることが好ましい。

【００１６】

本発明の他の目的は、クライアント装置からのデータ要請を通信することにおける信頼性を高めるためのシステムであって、（ａ）クライアント装置と信頼できるサーバーとの間のネットワーク接続を確立するための手段と、（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、前記装置とサーバーとの間の通信は、（ｉ）ワンタイムパスワードを生成する段階と、（ｉｉ）装置からの以前の要請に対する、信頼できるサーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成する段階とを具備する方法にしたがって実施されるシステムを提供することである。前記システムは、暗号化アルゴリズムと、前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段とをさらに具備し、前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、前記装置とサーバーとの間の通信は、暗号化ベースで実施されることが好ましい。

【００１７】

本発明の他の目的は、クライアント装置から信頼できるサーバーへの要請からの応答を通信することにおける信頼性を高めるためのシステムであって、（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、前記装置とサーバーとの間の通信は、（ｉ）装置と前記サーバーとの間で共有される少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する要請を受信する段階と、（ｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成する段階とを具備する方法にしたがって実施されるシステムを提供することである。前記システムは、暗号化アルゴリズムと、前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段とをさらに具備し、前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、前記装置とサーバーとの間の通信は、暗号化ベースで実施されることが好ましい。

【００１８】
本発明の他の目的は、クライアント装置と信頼できるサーバーとの間の通信を再同期化するための方法であって、（ａ）装置からサーバーへの通信において用いるためのワンタイムパスワードを生成または回収する段階と、（ｂ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を含む少なくとも１つのワンタイム要請－認証データを生成または回収する段階と、（ｃ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を含む少なくとも１つのワンタイム応答－認証データを生成または回収する段階とを具備する方法にも向けられる。好ましい実施形態において、前記ワンタイム要請－認証データは、Ａｌｌ ＮＵＬＬメッセージ暗号化鍵を具備する。他の好ましい実施形態において、前記ワンタイム応答－認証データは、Ａｌｌ ＮＵＬＬメッセージ暗号化鍵を具備する。前記方法については、サーバーにより現在のものとして認識されないワンタイムパスワードを用いるサーバーにより受信された要請が、以前に生成された応答がもしあれば、該応答の送信という結果となるように構成設定することができる。再同期化要請メッセージは、要請メッセージ（の１タイプ）と見なされる。再同期化応答メッセージは、応答メッセージ（の１タイプ）と見なされる。
【００１９】
本発明の他の目的は、クライアント装置からの再同期化要請を送信するための方法であって、（ａ）ワンタイムパスワードを生成または回収する段階と、（ｂ）装置からの要請に対する、信頼できるサーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを生成または回収する段階とを具備する方法を提供することである。好ましい実施形態において、前記ワンタイム要請－認証データは、Ａｌｌ ＮＵＬＬメッセージ暗号化鍵を具備する。他の好ましい実施形態において、前記再同期化要請は、以前の再同期化データに取って代わる暗号化された再同期化データを具備する。
【００２０】
本発明の他の目的は、信頼できるサーバーからの再同期化応答を送信するための方法であって、（ａ）クライアント装置と関連した少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する要請を受信する段階と、（ｂ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを生成または回収する段階とを具備する方法を提供することである。好ましい実施形態において、前記ワンタイム応答－認証データは、Ａｌｌ ＮＵＬＬメッセージ暗号化鍵を具備する。他の好ましい実施形態において、前記再同期化応答は、以前の再同期化データに取って代わる暗号化された再同期化データを具備する。
【００２１】
本発明の他の目的は、クライアント装置と信頼できるサーバーとの間の通信を再同期化するためのシステムであって、（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、前記装置とサーバーとの間の通信は、（ｉ）装置からサーバーへの通信において用いるためのワンタイムパスワードを供給する段階と、（ｉｉ）装置からの以前の要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを供給する段階と、（ｉｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを供給する段階とを具備する方法にしたがって実施されるシステムを提供することである。前記システムは、暗号化アルゴリズムと、前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段とをさらに具備し、前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、前記装置とサーバーとの間の通信は、暗号化ベースで実施される。
【００２２】
本発明の他の目的は、クライアント装置からの再同期化要請の通信における信頼性を高め

るためのシステムであって、（ａ）クライアント装置と信頼できるサーバーとの間のネットワーク接続を確立するための手段と、（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、前記装置とサーバーとの間の通信は、（ｉ）ワンタイムパスワードを供給する段階と、（ｉｉ）装置からの要請に対する、サーバーからの以前の応答の少なくとも一部の機能を具備する少なくとも１つのワンタイム要請－認証データを供給する段階とを具備する方法にしたがって実施されるシステムを提供することである。前記システムは、暗号化アルゴリズムと、前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段とをさらに具備し、前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、前記装置とサーバーとの間の通信は、暗号化ベースで実施される。

【００２３】

本発明の他の目的は、信頼できるサーバーからの再同期化応答の送信における信頼性を高めるためのシステムであって、（ａ）クライアント装置とサーバーとの間のネットワーク接続を確立するための手段と、（ｂ）クライアント装置とネットワーク接続を介してデータ通信を実施するための手段とを具備し、前記装置とサーバーとの間の通信は、（ｉ）クライアント装置と関連したワンタイムパスワードを具備する要請を受信する段階と、（ｉｉ）少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備する少なくとも１つのワンタイム応答－認証データを供給する段階とを具備する方法にしたがって実施されるシステムを提供することである。前記システムは、暗号化アルゴリズムと、前記暗号化アルゴリズムを、前記ネットワーク接続を介して、クライアントコンピュータへダウンロードするための手段とをさらに具備し、前記クライアントコンピュータとネットワーク接続を介してデータ通信を実施するための手段は、前記暗号化アルゴリズムにしたがい、前記装置とサーバーとの間の通信は、暗号化ベースで実施される。

【００２４】

本発明は、クライアント－信頼できるサーバー間の効率的な両方向通信のセキュリティの種々の特徴についての同時的な補償範囲（ｃｏｖｅｒａｇｅ）を提供するために、緊密に統合されたアプローチを用いる。ジュエルズおよびブレイナードとは違い、本発明は、登録されたクライアント装置が、サーバーにより他の入力してくるインターネットトラフィックと区別することができるプロトコルにしたがってパターンを送信できるクライアントからなる顕著な部類を形成する、という事実を利用する。本発明は、（ｈｔｔｐ：／／ｃｓｒｃ．ｎｉｓｔ．ｇｏｖ／ｃｒｙｐｔｖａｌ／ｄｅｓ．ｈｔｍにおいて公開されているＦＩＰＳ ４６－２ Ｄａｔａ Ｅｎｃｒｙｐｔｉｏｎ Ｓｔａｎｄａｒｄ、および、ＦＩＰＳ ８１ ＤＥＳ Ｍｏｄｅｓ ｏｆ Ｏｐｅｒａｔｉｏｎ（ＭＡＣｉｎｇ）（本明細書内で、以後“ＦＩＰＳ”と称する）にしたがう）暗号化およびＭＡＣの使用と関連して、クライアントメッセージのリンク不可能性（ｕｎｌｉｎｋａｂｉｌｉｔｙ）の他に、両方向メッセージの出所、完全性、および、新鮮度のためのその後のメッセージにおいて用いられるワンタイムパスワードの構成要素およびワンタイム使用の（ｏｎ－ｔｉｍｅ－ｕｓｅ）ＭＡＣ鍵の構成要素を安全に送信するために、リベスト、シャミア、および、エイドルマンの方法を（ベラーレおよびロガウェイに基づいて強化されたものとして）用いる。この結果、Ｄｓ認証子（Ｄｓａｕｔｈｅｎｔｉｃａｔｏｒｓ）とは違い、前記方法は自己同期的ではなく、再同期化は、再送信によりサーバー端部上で効率的に処理される。サーバーは、（現在に対して）以前に有効であったワンタイムパスワードを用いて通信される正規の要請と詐欺的な要請とを追跡または識別する必要がない。その理由は、以前に生成された応答の回収および（再）送信を、さらなる計算やデータベースの更新を必要とせずに行えるような場合には、いかなる（潜在的にはリソースインテンシブ（ｒｅｓｏｕｒｃｅ－ｉｎｔｅｎｓｉｖｅ）の）暗号化処理もサーバーにより行われないためである。

【００２５】

サーバー端部上におけるメッセージ処理は、サービス拒否に対して抵抗力のある段階的なアプローチにより処理され、現時点で正規のワンタイムパスワードを添付されていない（

さらなる新たな処理のための候補のような）要請メッセージを最初に一掃する。現時点で正規のワンタイムパスワードを含めることは、サーバーのデータベース上での“ヒット”という結果となり、この場合に、ワンタイムパスワードは、１つのクライアント装置に関する情報についてデータベースを検索（ｌｏｏｋｕｐ）するために用いられる。要請メッセージの適切なデータフィールドに適用された場合のデータベースの入力（ｄａｔａｂａｓｅ ｅｎｔｒｙ）内のワンタイム使用のＭＡＣ鍵がメッセージの準拠性（ｍｅｓｓａｇｅ ｃｏｍｐｌｉａｎｃｅ）を示せば、ＲＳＡ解読が、サーバーの秘密鍵（この秘密鍵については、サーバーの暗号化モジュール内またはハードウェアセキュリティモジュール（ＨＳＭ）内で保護することができる）を用いて行われる。ＲＳＡ解読は、次のワンタイムパスワードと、メッセージ鍵が存在する場合には該メッセージ鍵とに関する情報を明らかにし、該メッセージ鍵は、バルク暗号化アルゴリズム（例えば、ＤＥＳの変形）を用いて送信された要請メッセージが存在する場合には該要請メッセージのうちのこの部分を解読するために用いられる。サーバーは、得られた現在のＭＡＣ鍵を用いて計算されたメッセージ認証コード（ＭＡＣ）を少なくとも部分的に組み込む応答メッセージを、最も最近に受信された要請メッセージ内で移送された次のワンタイムパスワードまたはその構成要素についての知識を用いることにより計算する。応答メッセージは、新たに生成されたメッセージ鍵と、クライアントによる次の要請のための次のワンタイム使用のＭＡＣ鍵の構成要素とを伝達することもできる。この伝達手段は、サーバーのデータベース内で指し示されたようなクライアントの公開鍵の下での暗号化であってもよい。応答メッセージは、バルク暗号化されたデータを含むこともでき、この場合に、対応するプレーンテキストを、（応答）メッセージ鍵を用いて回復することができる。本発明の（暗号化が可能な）装置とは、クライアント装置を指し、サーバーや該サーバーにおけるハードウェアセキュリティモジュール（ＨＳＭ）を指すものではない。

【００２６】

前記装置および装置サーバー（すなわち、信頼できるサーバー）の両方の公開鍵／秘密鍵の組は、安全な通信をトランザクションベースで取り決めるするために必要な共有される秘密を更新するために用いられる。このことは、暗号化通信に署名したり、または、署名された通信をプライバシーや計算上のオーバーヘッドやサービス拒否攻撃に関して暗号化したりする標準的な技術に対し、幾つかの利点を提供する。純粋な対称鍵によるアプローチは、装置サーバーのデータベース内の値の静的スナップショットに基づく攻撃の可能性につながる。装置が不完全なトランザクションにより装置サーバーとの暗号同期化（ｃｒｙｐｔｏｓｙｎｃｈｒｏｎｉｚａｔｉｏｎ）を失えば、中断されたトランザクションとその後のトランザクションとの間のプライバシーを脅かすようなリンクを提供せずに、かつ、期限切れの情報または望ましくない情報を装置に受理させずに、同期（ｓｙｎｃｈ）が再確立される。装置の公開鍵のリストが与えられても、どの装置が関与していたのかにしたがってトランザクションを区分化することはできない。

【００２７】

ＲＳＡとともにＯＡＥＰ（Ｏｐｔｉｍａｌ Ａｓｙｍｍｅｔｒｉｃ Ｅｎｃｒｙｐｔｉｏｎ Ｐａｄｄｉｎｇ）を用いることは、プロトコル中にさらされるデータを暗号化しかつ暗号文（ｃｉｐｈｅｒｔｅｘｔ）を以前のトランザクションにマッチングしようとすることによりトランザクションをリンクする試みを阻止する。要請メッセージおよび応答メッセージは、独立的に生成され、この結果、装置サーバーのデータベースのスナップショットを得ることは、要請において用いられた既知のメッセージ鍵において暗号化された応答メッセージという結果となる要請メッセージを、エミュレートされた装置に提起する機会を与えない。サービス拒否に関して、前記システムは、登録された装置が、自身の出力を、サーバーにおいて他の入力してくるインターネットトラフィックと区別することができるという点で顕著な部類を形成する、という事実を利用する。入力してくる要請メッセージ内でワンタイムパスワードを用いることが、装置サーバーのデータベース内における新たな（すなわち、現在の）ヒットという結果となれば、サーバーは、メッセージ鍵を回復するためのＲＳＡ解読と、プレーンテキストを回復するためのメッセージ鍵を伴う対称アル

ゴリズム解読との前にＭＡＣをチェックするために“ヒット”装置入力を用いる。入力してくる要請メッセージ内でワンタイムパスワードを用いることが、装置サーバーがたった今処理したばかりのトランザクションを参照すれば、サーバーは、さらなる処理またはデータベースの更新を招くことなく、以前の応答を再送信する。安全な通信プロトコルは、重大な工程が装置サーバーにおける安全な暗号化モジュール（または、ハードウェアセキュリティモジュール）内で実行される場合に、データベースへの認可されていないアクセスが、それ自体で、システムの完全性を損なわないように企画される。

【００２８】

後述する安全な通信プロトコルは、公開鍵の暗号化の使用を、暗号化（および解読）のためには必要とするが、ディジタル署名目的のためには必要としない。効率という観点から見て、安全な通信の検証の成功が、適切に機能する登録された装置の表示として機能する際に、“プレーンテキスト”内で装置により生成されかつ送信されるあらゆるディジタル署名を達成でき、かつ、後で、帯域外で、トランザクション処理からオフラインで検証できることに留意することは重要である。プレーンテキストは、メッセージ鍵の下で暗号化され、この場合に、暗号化されたメッセージ鍵および暗号化されたプレーンテキストは、ＭＡＣにより安全な通信の下で認証される。包含されるあらゆる署名が、署名されたテキストを添付されているという点で完全な署名であれば、安全な通信プロトコルは、署名の包含により可能にされる否認防止（ｎｏｎ－ｒｅｐｕｄｉａｔｉｏｎ）能力から独立してテキストを認証するように機能する。

【００２９】

適切に機能する装置は、安全な通信の要請内における偽の署名を受理しない。その理由は、これらの署名の生成および処理は装置により制御されるためである。この結果、本発明の方法については、サーバーによりメッセージと同時に署名を検証することを必要としないように実施することができる。署名については、後で記憶できかつ検証できるという点で、“ｂｌｏｂ”として扱うことができる。リアルタイム認証を処理するために、メッセージ認証コード（ＭＡＣ）をクライアントにより生成すること、および、サーバーにより検証することが、代わりに用いられる。この目的のために、対称鍵が用いられる。本発明の方法は、メッセージの新鮮度（すなわち、メッセージが以前に用いられたか、または、現在の受信の前に傍受されたか否か）を確立する手順をさらに具備することができる。

【００３０】

本発明の方法の１つの利点は、メッセージをクライアントにリンクすることができないことである。したがって、例えば、メッセージが傍受されても、本発明の方法は、該メッセージがどこから生じたのかを傍受者が見分けることを可能にしない。

【００３１】

本発明の方法は、自己同期的ではない。したがって、前記方法を適切に機能させることは、クライアント装置およびサーバーがこのような再整列（ｒｅａｌｉｇｎｍｅｎｔｓ）間における同期を独立的に維持できるように、サーバー入力に基づいて、クライアント装置が周期的に自身の動きを適応させることを必要としない。代わりに、同期化は、サーバーからの再送信を通して、または、サーバーによる同期化メッセージ処理を通して、トランザクションベースで回復される。

【００３２】

したがって、本発明は、署名の検証を必要としない安全な通信方法を含む。本発明は、自己同期的ではない安全な方法をさらに提供する。最後に、クライアント装置は信頼できるサーバーに登録されるので、クライアント装置は、サーバーにより他の入力してくるインターネットトラフィックと区別することができるプロトコルにしたがってパターンを送信する装置からなる顕著な部類を形成する。

【００３３】

【発明の実施の形態】

以下の内容は、本発明についての特に有用な幾つかの実施形態についての論考である。本発明による信頼できるサーバー（ｔｒｕｓｔｅｄ ｓｅｒｖｅｒ）は、２つの構成要素を

具備することが好ましい。第１構成要素は、状態変更を追跡することが可能なホストプロセッサおよびデータベースである。第２構成要素は、暗号化処理能力と固定値の安全な記憶装置とを備えたハードウェアセキュリティモジュール（ＨＳＭ）である。

【００３４】

本発明によるクライアント装置は、同様に、２つの構成要素を具備することが好ましい。第１構成要素はプロセッサである。第２構成要素は、安全な環境内のコプロセッサである。データについては、クライアント装置のコプロセッサにより暗号化することができ、かつ、信頼できるサーバーのＨＳＭのために意図することができる。これにより、暗号化されたデータについては、ＨＳＭによってのみ解読することができ、例えば、信頼できるサーバーにおいてインサイダーにより解読することはできない。信頼できるサーバーのＨＳＭおよびクライアント装置のコプロセッサは、各々が秘密鍵を有する。さらに、特定の各クライアント装置のための公開鍵は、ＨＳＭにより認識される。各装置のための公開鍵および秘密鍵の組については、生成段階または登録段階で定義することができる。

【００３５】

前記ＨＳＭは、信頼できるサーバー上のデータベース内への記憶のために暗号化されたデータがリリースされるか否かの決定をさらに行う。次のメッセージがクライアント装置への送信のためにＨＳＭにより準備される場合に、ＨＳＭは、その特定のクライアント装置に特有のデータを求めて、信頼できるサーバー上のデータベースにアクセスする必要がある。装置に特有でありかつ現時点のデータについての知識をＨＳＭが実証しなければ、装置はメッセージを有効なものとして受理しない。クライアント装置により信頼できるサーバーへ送信されたメッセージについても同じことが適用される。クライアント装置が、信頼できるサーバーのデータベース内における、該クライアント装置についての、受信されたメッセージに正確に対応するデータにアクセスできなければ、ＨＳＭは応答処理のためのメッセージを受理しない。装置またはこれに対応するデータベースの入力に対する現在のアクセス権をメッセージ作成者が有していなければ、ＨＳＭはメッセージを受理しない。

【００３６】

本発明の方法は、クライアント装置から信頼できるサーバーへ送信される要請メッセージ内に組み込まれるワンタイムパスワード（ｏｎｅ－ｔｉｍｅ ｐａｓｓｗｏｒｄ）を用いる。要請－認証データ内のワンタイムパスワードについては、サーバーにより、そのデータベース内において、クライアント装置に対応する入力を見つけるために用いることができる。“現在の”ワンタイムパスワードがこうして包含される一方で、この認証された要請メッセージ、および、サーバーから結果的に生じる確認の応答メッセージは、“次の”ワンタイムパスワードを、その値を公開したり開示したりせずに、装置およびサーバーの両方において設定するデータ交換を含む。要請－認証データは、暗号化された秘密データを具備することが好ましく、この場合に、サーバーは、暗号化された秘密データを解読して秘密データを回復する。次の、または、その後の要請は、少なくとも１つの秘密データの少なくとも一部を具備する少なくとも１つのワンタイムパスワードの少なくとも一部の機能を具備することが好ましい。インサイダーが信頼できるサーバーのデータベースの“スナップショット”を得た場合であっても、いったん実際のクライアントが信頼できるサーバーとのトランザクションの成功によりデータベースを更新させると、該インサイダーが、ＨＳＭを欺いて自分をクライアントであると思い込ませるためにこの獲得された知識を単独では使用することができない旨を、データの秘密交換（ｐｒｉｖａｔｅ ｅｘｃｈａｎｇｅ）は保証する。さらに、インサイダーは、現在のデータベースアクセス権に基づいて要請を提出した場合であっても、ＨＳＭからの応答を傍受することができない。その理由は、インサイダーが装置の秘密鍵を有していないためであり、かつ、応答内のプレーンテキストを解読するために用いられるメッセージ鍵が、要請内のメッセージ鍵についての知識のみに基づいて導き出せないためである。インサイダーは、ＨＳＭ秘密鍵を有していないので、クライアントからの入力してくるメッセージを傍受することもできない。

【００３７】

好ましい実施形態において、使用されるプロトコルが、以下のように定義される。

【００３８】

〈用語および表記〉

｛ｘ｝ＥｎｔｉｔｙＰｕｂＫ は、エンティティ（Ｅｎｔｉｔｙ）のＲＳＡ公開鍵の下でのメッセージｘのＲＳＡ－ＯＡＥＰ（Ｏｐｔｉｍａｌ Ａｓｙｍｍｅｔｒｉｃ Ｅｎｃｒｙｐｔｉｏｎ Ｐａｄｄｉｎｇ）暗号化を表す。

【００３９】

｛ＰＴ｝ＭｓｇＫ は、メッセージ鍵ＭｓｇＫの下での（プレーンテキスト）ＰＴの対称アルゴリズム（例えば、トリプルＤＥＳ）暗号化を表す。

【００４０】

ＭＡＣ（ｄａｔａ）Ｋｅｙ は、鍵Ｋｅｙの下でのデータのＭＡＣに基づく対称アルゴリズムを表す。

【００４１】

Ｐｒｏｔｏｃｏｌ Ｈｅａｄｅｒ は、クリアテキストデータ（すなわち、暗号化されずに送信されたデータ）を具備し、該クリアテキストデータは、プロトコルバージョン、または、さらなる処理の前に受信するために有用でありかつ開示による影響を受けづらい他のデータに関する項目を含むことができる。Ｐｒｏｔｏｃｏｌ Ｈｅａｄｅｒのデータフィールドは、固定長でなければ、その長さを指定する固定長の前提部分（ｐｒｅａｍｂｌｅ）を含むことができる。

【００４２】

引数（ａｒｇｕｍｅｎｔｓ）間またはデータフィールド間におけるコンマ（”，　”）の使用は、連結（ｃｏｎｃａｔｅｎａｔｉｏｎ）を示す。［ａ，　ｂ］は、ａの後にｂが続く連結を示す。

【００４３】

“．ＸＯＲ．”は、ビット単位（ｂｉｔ－ｗｉｓｅ）の排他的ＯＲ（すなわち、同様の長さのベクトル（ｌｉｋｅ－ｌｅｎｇｔｈ ｖｅｃｔｏｒｓ）の成分単位（ｃｏｍｐｏｎｅｎｔ－ｗｉｓｅ）の加算モジュロ－２）を示す。ＭＡＣ（ｄａｔａ）K_1．ＸＯＲ．K_2 は、K_1．ＸＯＲ．K_2 を鍵として用いた“ｄａｔａ”を介しての演算の結果として生じるＭＡＣ値である。

【００４４】

Ｈ（ｍ） は、一方向ハッシュ関数（例えば、ＳＨＡ－１）をメッセージｍに適用した結果を示す。

【００４５】

〈装置側の基本的フロー〉

（公知の方法による）クライアント装置の登録の成功という結論において、装置および信頼できるサーバーが、T_0 およびT_{0TS} により示される２つの秘密値を共有し、かつ、各々が他方の公開鍵の確実なコピーを維持すると仮定する。この実施形態について、T_0 およびT_{0TS} の生成は、T_{0TS}．ＸＯＲ．T_0 が（２鍵式トリプルＤＥＳ（２－ｋｅｙ ｔｒｉｐｌｅ－ＤＥＳ））鍵であるようなものであってもよい。以下、図１を参照する。概略的に、装置および信頼できるサーバーが（暗号化）同期化状態（（ｃｒｙｐｔｏ－）ｓｙｎｃｈｒｏｎｉｚａｔｉｏｎ）にあれば、Ｒｅｑｕｅｓｔ（ｎ）の処理を開始する前の装置の持続性メモリ（ｐｅｒｓｉｓｔｅｎｔ ｍｅｍｏｒｙ）は、以下の通りである：

【数１】

T_{n-1}，T_{n-1TS}，Blank，

ここで、T_{n-1}は、ワンタイムパスワードである。

【００４６】

このデータ位置内の“Ｂｌａｎｋ”以外の値により明示されるように、信頼できるサーバーとの再同期化（ｒｅｓｙｎｃｈｒｏｎｉｚａｔｉｏｎ）が必要とされれば（１０３）、装置は、さらに後述するように、再同期化要請を生成する。再同期化が必要とされなければ、装置がＲｅｑｕｅｓｔ（ｎ）の開始を希望する場合に、装置は、新たな２鍵式トリプルＤＥＳ鍵Ｘを得て、かつ、$T_n = X.XOR.T_{n-1TS}$とする（１０５）。ここで、T_{n-1TS}は、信頼できるサーバーにより、以前の応答内に生成される。したがって、Ｒｅｑｕｅｓｔ（ｎ）は、以前の応答の少なくとも一部の機能を具備する。装置は、要請メッセージＲｅｑｕｅｓｔ（ｎ）（下記の説明を参照）を生成する（１０７）。（ここで、ＰＴ（プレーンテキスト）は、バルク暗号化形式で配信すべきクライアント側ユーザーのメッセージの内容の一部である。）２４バイトのトリプルＤＥＳ鍵（ＭｓｇＫ）が生成される。ＰＴは、ＭｓｇＫによってトリプルＤＥＳ暗号化される。次に、T_nとＭｓｇＫとの連続は、信頼できるサーバー（ＴＳ）の公開鍵によって、ＯＡＥＰパディングされ（ＯＡＥＰ－ｐａｄｄｅｄ）、かつ、ＲＳＡ暗号化される。ＣＢＣ（暗号ブロック連鎖（ｃｉｐｈｅｒ－ｂｌｏｃｋ－ｃｈａｉｎｉｎｇ））ＭＡＣは、“ｄａｔａ”［｛T_n，ＭｓｇＫ｝ＴＳＰｕｂＫ，｛ＰＴ｝ＭｓｇＫ］と連結されるプロトコルヘッダーを介して生成される。ＭＡＣは、$T_{n-1TS}.XOR.T_{n-1}$を用いて生成される。$T_{n-1TS}.XOR.T_{n-1}$が１６バイトであるので、ＭＡＣを計算する場合にトリプルＤＥＳアルゴリズムを実行するために二重鍵（ｄｏｕｂｌｅ ｋｅｙ）が用いられることに留意されたい。プロトコルヘッダーおよびT_{n-1}は、ＭＡＣに対してプリペンド（ｐｒｅｐｅｎｄ）される。データは、ＭＡＣの後に付加される。

【００４７】
この要請において、T_nおよびＭｓｇＫは、新たに生成された値である。

【００４８】
したがって、T_nおよびメッセージ鍵ＭｓｇＫは、サーバーへの移送目的のために、サーバーの公開鍵を用いて暗号化される。装置が全ての要請のために新たなメッセージ鍵を生成するので、メッセージ鍵を記憶するためのメモリは全く必要とされない。

【数２】

$$Request(n) = Protocol\ Header,\ T_{n-1},$$

$$MAC(Protocol\ Header,\ data)T_{n-1TS}.XOR.T_{n-1},\ data,$$

ここで、$data = \{T_n,\ MsgK\}TSPubK,\ \{PT\}MsgK$

【００４９】
Ｒｅｑｕｅｓｔ（ｎ）を送信する前に、装置は、以下の持続性メモリの状態へ進む（１０９）：

【数３】

$$T_{n-1},\ T_{n-1TS},\ T_n$$

次に、Ｒｅｑｕｅｓｔ（ｎ）は送信され（１１１）、かつ、Ｒｅｓｐｏｎｓｅ（ｎ）がサーバーから装置へ送信される（１１３）。サーバー側のフローについて、以下に説明する。

【００５０】
Ｒｅｓｐｏｎｓｅ（ｎ）を受信すると、装置は応答を完全に処理する。その理由は、ランダムに（または、疑似ランダムに）生成されたＭｓｇＫが、装置が基本的フローモードではなく再同期化モード（後述する）にあることを示すＡＬＬ ＮＵＬＬベクトルではないためである。信頼できるサーバーからのＲｅｓｐｏｎｓｅ（ｎ）についての満足を与える検証（１１５）という結論において、装置は、以下の持続性メモリの状態へ進む（１１７）：

【数４】

T_n, T_{nTS}, Blank

【００５１】

〈装置側のフロー － 暗号同期化の再確立〉

以下、図２を参照する。例えば、クライアント装置が時間切れになったか、または、クライアント側のプロセッサが故障し、この結果、装置が持続性メモリの状態T_{n-1}，T_{n-1TS}，T_nにあり、かつ、Ｒｅｓｐｏｎｓｅ（ｎ）を待機していないと仮定する。クライアント装置の動作が再開されると、クライアント装置は、装置が再同期化モードにあることを示すメッセージ鍵を生成する。クライアント装置は、ＮＵＬＬ ＭｓｇＫを生成し、かつ、ＮＵＬＬ ＭｓｇＫを用いて特殊な再同期化要請（２０１）を送信することが好ましい。ＰＴは存在しない：

【数５】

Request(n)＝Protocol Header, T_{n-1},

MAC(Protocol Header, data)T_{n-1TS}.XOR.T_{n-1}, data,

ここで、data＝{T_n, MsgK}TSPubK、および、暗号化鍵 MsgK＝ALL NULLs

【００５２】

前記装置は、自身が再同期化モードにあり、かつ、通常の送信モードにはないことを認識する：揮発性メモリＭｓｇＫ＝ＡＬＬ ＮＵＬＬｓという事実は、受信されたＲｅｓｐｏｎｓｅ（ｎ）を検証する場合にはあらゆる｛ＰＴ｝ＭｓｇＫ’フィールドを捨てるようにクライアント装置に通知する。この応答は、実際には、再同期化要請にではなく基本的フローに応答して最初に生成された記憶された応答である場合には、このような暗号化データフィールドを含むことができる。応答内のＭＡＣが、この場合には、プレーンテキストＰＴではなく暗号文｛ＰＴ｝ＭｓｇＫ’を介して計算されるので、装置は、ＭＡＣを検証するために解読を行う必要がない。

【００５３】

前記信頼できるサーバーからのＲｅｓｐｏｎｓｅ（ｎ）についての満足を与える検証（２０５）という結論において、装置は、以下の持続性メモリの状態へメモリを更新する（２０７）：

【数６】

T_n, T_{nTS}, Blank

【００５４】

〈サーバー側の基本的フロー〉

最初に、所定の値ｎについてのＲｅｑｕｅｓｔ（ｎ）を装置から受信する前に、この装置についての、信頼できるサーバーのデータベース値は、以下の通りである：

【数７】

T_{n-2}, T_{n-1}, T_{n-1TS}, Response(n-1)

【００５５】

Ｒｅｓｐｏｎｓｅ（ｎ）についての満足を与える検証に基づいて、信頼できるサーバーは、T_{nTS}とＲｅｓｐｏｎｓｅ（ｎ）とを生成し、かつ、そのデータベース値は、以下の通りである：

【数８】

T_{n-1}, T_n, T_{nTS}, Response(n)

【００５６】
以下、図３を参照する。クライアント装置からの要請を受信すると（３０３）、信頼できるサーバーは、Ｐｒｏｔｏｃｏｌ Ｈｅａｄｅｒのバージョンフィールドに基づいてどのメッセージングプロトコルを用いるべきかを確立する。信頼できるサーバーは、T_{n-1}に基づいてクライアント装置の身元を確立し、かつ、T_{n-1}を用いてデータベースの入力からT_{n-1TS}を回収する（３０５）。サーバーは、T_{n-1TS}．ＸＯＲ．T_{n-1}を用いてＭＡＣを有効化（ｖａｌｉｄａｔｅ）する。その秘密鍵を用いて、信頼できるサーバーは、｛T_n，ＭｓｇＫ｝ＴＳＰｕｂＫを解読しかつＯＡＥＰ復号化し、かつ、T_nをセーブする。回復されたＭｓｇＫの値は、ＰＴを回復すべく｛ＰＴ｝ＭｓｇＫを解読するために用いられる。次に、これに応じて、信頼できるサーバーはＰＴを処理する。
【００５７】
本発明の一実施形態において、入力してくる値T_{n-1}に対応する現在のワンタイムパスワード入力がサーバーのデータベース内に存在しなければ、サーバーは、入力してくる値を、以前に生成された応答に対応するT_xに対してマッチングしようとする（３０７）。検索が成功すれば、サーバーは、入力してくる要請のT_{n-1}に対応する応答を再送信する（３０９）。より明確には、所定の装置についての信頼できるサーバーの値が現時点でT_{n-1}，T_n，T_{nTS}，Ｒｅｓｐｏｎｓｅ（ｎ）であれば、T_{n-1}は、信頼できるサーバーにより、要請を新たに処理するためには用いられない。代わりに、信頼できるサーバーがそのデータベースの入力を装置のために更新したが装置がその状態をまだ更新していない場合には、対応するＲｅｓｐｏｎｓｅ（ｎ）が、信頼できるサーバーと装置との間の（暗号）同期化を再確立するために用いられる。
【００５８】
入力してくる要請内のT_{n-1}が（現在のワンタイムパスワードとマッチングするという点で）予想され、かつ、要請が認証される場合には、サーバーは、該要請が再同期化のための要請ではないことを検証する（３１３）。再同期化が必要とされれば、再同期化応答（さらに後述する）が生成される。再同期化が必要とされなければ、信頼できるサーバーは、新たな２鍵式トリプルＤＥＳ鍵Ｙを生成し、かつ、T_{nTS}＝Ｙ．ＸＯＲ．T_nとし、かつ、自身のＰＴと新たに生成されたＭｓｇＫ’とによってＲｅｓｐｏｎｓｅ（ｎ）を生成する（３１５）。応答メッセージは、ＭＡＣがT_n．ＸＯＲ．T_{n-1TS}を用いて計算されることを除けば、要請メッセージと同じフォーマットで生成され、かつ、新たなT_{nTS}が、装置の公開鍵ＤｅｖｉｃｅＰｕｂＫの下で暗号化された引数として現れる：
【数９】

$$\text{Response}(n) = \text{Protocol Header}, T_{n-1},$$

$$\text{MAC}(\text{Protocol Header}, \text{data})T_n.\text{XOR}.T_{n-1TS}, \text{data},$$

ここで、$\text{data} = \{T_{nTS}, \text{MsgK'}\}\text{DevicePubK}, \{\text{PT}\}\text{MsgK'}$

【００５９】
生成されたＲｅｓｐｏｎｓｅ（ｎ）は、応答ＭＡＣを計算するために用いられる鍵を経て次のワンタイムパスワードT_nを確認する機能を含む。
【００６０】
この応答において、T_{nTS}およびＭｓｇＫ’は、新たに生成された値であり、この場合に、戻されるメッセージ鍵（ＭｓｇＫ’）は、装置により生成されたメッセージ鍵（ＭｓｇＫ）とは異なる。
【００６１】
前記要請メッセージ鍵ＭｓｇＫおよび応答メッセージ鍵ＭｓｇＫ’のいずれも、データベ

ース内にセーブされない。ＨＳＭが要請を解読しかつ応答を生成した後に、データベース値T_n（新規），T_{n-1TS}（以前）への適時のアクセスは、この応答を、準拠的なクライアントプラットフォームにとって受理可能である別の応答に置き換えることを可能にする。クライアント装置がクライアントプラットフォームの公開鍵を用いて（ＭｓｇＫ’の代わりに）ＭｓｇＫ．ＸＯＲ．ＭｓｇＫ’を暗号化状態で送信することを期待するように、プロトコルが修正されれば、ＭｓｇＫ’、および、ＨＳＭにより生成されるようなＭｓｇＫ’のいずれもＨＳＭから離れないので、応答の置き換えは受理されない。さらに、クライアント装置は、そのＭｓｇＫを不揮発性メモリに記憶することも必要としない。その理由は、クライアント装置は、暗号同期化を再確立するのみであり、したがって、最初の応答が予想時に受信されなければ応答メッセージのあらゆるバルク暗号化された内容を無視するためである。

【００６２】

前記持続性メモリにおいて、信頼できるサーバーは、T_{n-2}，Ｒｅｓｐｏｎｓｅ（ｎ－１），T_{n-1}，T_{n-1TS}を以前に（或る形式のアクセス可能な記憶装置内に）有していた。このことは、今では、T_{n-1}，Ｒｅｓｐｏｎｓｅ（ｎ），T_n，T_{nTS}に置き換えられている（３１７）。いったん、Ｒｅｓｐｏｎｓｅ（ｎ）が生成されかつセーブされれば、T_{n-1TS}についての知識はもはや必要とされない。

【００６３】

前記信頼できるサーバーは、Ｒｅｓｐｏｎｓｅ（ｎ）をクライアント装置へ送信する（３１９）。メッセージの受信に基づいて、装置は、Ｐｒｏｔｏｃｏｌ Ｈｅａｄｅｒのバージョンを検証し、かつ、T_{n-1}を無視する。ＭＡＣは、T_n．ＸＯＲ．T_{n-1TS}を用いて検証される。その秘密鍵を用いて、クライアント装置は、{T_{nTS}，ＭｓｇＫ’}ＤｅｖｉｃｅＰｕｂＫを解読しかつＯＡＥＰ復号化する。装置は、ＰＴを回復するためにＭｓｇＫ’を用いる。装置は、これに応じて、ＰＴを処理する。

【００６４】

持続性メモリまたは不揮発性メモリにおいて、装置は、T_{n-1}，T_{n-1TS}，T_nを以前に有していた。このことは、今では、T_n，T_{nTS}，Ｂｌａｎｋに置き換えられている。

【００６５】

〈サーバー側のフロー － 暗号同期化の再確立〉

図４を参照すると、サーバーは、メッセージを解読しかつ処理する。信頼できるサーバーがＲｅｑｕｅｓｔ（ｎ）を受信した時にこの装置についての該サーバーのデータベース値がT_{n-2}，T_{n-1}，T_{n-1TS}，Ｒｅｓｐｏｎｓｅ（ｎ－１）であれば、該サーバーは、要請を処理し、かつ、装置が再同期化モードにあることを示すメッセージ鍵を用いてT_{nTS}およびＲｅｓｐｏｎｓｅ（ｎ）を生成する（４０１）。クライアント装置は、ＮＵＬＬ ＭｓｇＫ’を生成することが好ましい。ＰＴは存在しない。次に、サーバーは、Ｒｅｓｐｏｎｓｅ（ｎ）を送信する（４０５）：

【数１０】

$$\text{Response}(n) = \text{Protocol Header}, T_{n-1},$$

$$\text{MAC}(\text{Protocol Header}, \text{data})\, T_n.\text{XOR}.T_{n-1TS}, \text{data},$$

ここで、$\text{data} = \{T_{nTS}, \text{MsgK'}\}\text{DevicePubK}$、および、暗号化鍵 $\text{MsgK'} = \text{ALL NULLs}$

【００６６】

そのデータベース値は、T_{n-1}，T_n，T_{nTS}，Ｒｅｓｐｏｎｓｅ（ｎ）を含むように更新される（４０３）。

【００６７】

この装置について装置サーバーのデータベース値が、或るＲｅｓｐｏｎｓｅ（ｎ）に対して、T_{n-1}，T_n，T_{nTS}，Ｒｅｓｐｏｎｓｅ（ｎ）であれば、装置サーバーが

このＲｅｑｕｅｓｔ（ｎ）を受信した時に、該装置サーバーは、Ｒｅｓｐｏｎｓｅ（ｎ）を（再）送信する。この場合に、受信されたＲｅｑｕｅｓｔ（ｎ）内のT_{n-1}は、データベースの入力（すなわち、以前に送信されかつ記憶されたＲｅｓｐｏｎｓｅ（ｎ）の値）にアクセスするために用いられる。要請が“新鮮”であったならば、データベースの入力（すなわち、T_{nTS}、および、装置の公開鍵ＤｅｖｉｃｅＰｕｂＫ）にアクセスするためにT_nが用いられたことになる。

【００６８】

前記クライアント装置は、図１および図２と関連して前述したように、応答メッセージを処理し、かつ、その持続性メモリを更新する。

【００６９】

前記サーバーにおける状態損失の可能性が重要なものであれば、本発明の拡張部分を、稀にしかアクセスされないフェイルセーフ（ｆａｉｌ－ｓａｆｅ）バックアップの可用性を利用するために展開することができる。例えば、“強要（Ｄｕｒｅｓｓ）モード”（後述する）に関する例外処理の場合に、サーバーは、バックアップ要請メッセージの受信を損なわれていない状態で認識する遠隔のバックアップサービスまたは設備にアクセスすることができる。サーバーにおける状態損失が検出されたりまたは該状態損失の疑いがある場合に、サーバーは、自身がバックアップ設備に堆積したデータのコピーを回収する。この状態回復方法の実施形態において、装置およびサーバーが、登録または他の初期化の一部としての初期値T_0，T_{0TS}について同意する場合に、装置およびサーバーは、初期の強要値の組Ｄｕｒｅｓｓ－T_0およびＤｕｒｅｓｓ－T_{0TS}についても同意する。これまでに説明したような再同期化が、公知の方法を利用して装置（または装置のユーザー）により追跡できるような規定された数の試行または規定された時間経過（または他の測定基準）の後に、（（暗号）同期化の再獲得または再確立という）望ましい効果の達成を失敗すれば、再同期化の例外処理版を用いることができる。装置側の再同期化という語は、装置側の処理の例外処理、または、強要モード版を具備することが理解される。サーバー側の再同期化という語は、サーバー側の処理の例外処理、または、強要モード版を具備することが理解される。Ｄｕｒｅｓｓ Ｒｅｑｕｅｓｔメッセージは、要請メッセージ（の一タイプ）と見なされる。Ｄｕｒｅｓｓ Ｒｅｓｐｏｎｓｅメッセージは、応答メッセージ（の一タイプ）と見なされる。装置は、Ｄｕｒｅｓｓ Ｒｅｑｕｅｓｔメッセージを生成しかつ送信する。Ｄｕｒｅｓｓ Ｒｅｑｕｅｓｔメッセージは、或る必要条件によって、標準的な要請メッセージのフォーマットに従い、また、結果的に生じるＤｕｒｅｓｓ Ｒｅｓｐｏｎｓｅメッセージも標準的な応答メッセージのフォーマットに従う。すなわち、現在の（標準的な）Ｔ値ではない現在のＤｕｒｅｓｓ－Ｔ値は、Ｄｕｒｅｓｓ ＲｅｑｕｅｓｔおよびＤｕｒｅｓｓ Ｒｅｓｐｏｎｓｅ内で用いられる。Ｄｕｒｅｓｓ ＲｅｑｕｅｓｔおよびＤｕｒｅｓｓ Ｒｅｓｐｏｎｓｅ内で新たに生成されたＴ値は、それぞれ、新たな“登録されたばかりの”開始地点へリセットするために用いられ、これにより、ここでは、T_0，T_{0TS}としてそれぞれ指定される（しかし、元のT_0，T_{0TS}値には関連づけられない）。Ｄｕｒｅｓｓ Ｒｅｑｕｅｓｔ（ｍ）のＰＴフィールドは、（少なくとも）Ｄｕｒｅｓｓ－T_mを含み、かつ、Ｄｕｒｅｓｓ Ｒｅｓｐｏｎｓｅ（ｍ）のＰＴフィールドは、（少なくとも）Ｄｕｒｅｓｓ－T_{mTS}を含む。

【数１１】

$$\text{Duress Request}(m) = \text{Protocol Header, Duress-}T_{m-1},$$

$$\text{MAC(Protocol Header, data)Duress-}T_{m-1TS}.\text{XOR.Duress-}T_{m-1}\text{, data,}$$

ここで、$\text{data} = \{T_0, \text{MsgK}\}\text{TSPubK, } \{\text{Duress-}T_m\}\text{MsgK}$

$$\text{Duress Response}(m) = \text{Protocol Header, Duress-}T_{m\ 1},$$

$$\text{MAC(Protocol Header, data)Duress-}T_m\text{.XOR.Duress-}T_{m-1TS}\text{, data,}$$

ここで、$\text{data} = \{T_{0TS}, \text{MsgK'}\}\text{DevicePubK, } \{\text{Duress-}T_{mTS}\}\text{MsgK'}$

【００７０】
前記装置による標準的な要請処理とは違い、失敗したＤｕｒｅｓｓ Ｒｅｑｕｅｓｔメッセージの再試行は、以前の（失敗した）試行の正確なコピーである。標準的なサーバーのデータベースの更新とは違い、サーバーが、
Ｄｕｒｅｓｓ－T_{m-2}，Ｄｕｒｅｓｓ－T_{m-1}，Ｄｕｒｅｓｓ－T_{m-1TS}，Ｄｕｒｅｓｓ Ｒｅｓｐｏｎｓｅ（ｍ－１） から、
Ｄｕｒｅｓｓ－T_{m-1}，Ｄｕｒｅｓｓ－T_{m}，Ｄｕｒｅｓｓ－T_{mTS}，Ｄｕｒｅｓｓ Ｒｅｓｐｏｎｓｅ（ｍ） へ、
ローカルに更新する場合に、この変更もまた、フェイルセーフ通信または他の非常に確実な（ｕｌｔｒａ－ｒｅｌｉａｂｌｅ）手段を用いてバックアップされる。

【００７１】
本発明の代替的実施形態は、認証段階と公開鍵暗号化段階とを組み合わせ、これにより、ＭＡＣの使用を不要にする。これは、入力してくる要請が現時点のものであることを示すサーバーのデータベース上のヒットが、要請メッセージデータの信憑性を検証する前に、サーバーにおいて要請メッセージをＲＳＡ－ＯＡＥＰ処理することを促進するという点で、より少ない段階のアプローチである。ＭＡＣベースの実施形態においては、ＭＡＣによる検証の失敗は、サーバーにおけるメッセージ処理の中断を促進する。ＭＡＣ無しでのアプローチにおける要請メッセージおよび応答メッセージの実施形態は、ＳＨＡ－１のような一方向ハッシュ関数Ｈを用いることができる：
【数１２】

$$\mathrm{Request}(n) = \mathrm{Protocol\ Header},\ T_{n-1},\ \{T_{n-1TS},\ T_n,\ H(\mathrm{Protocol\ Header},\ PT),\ MsgK\}TSPubK,\ \{PT\}MsgK$$

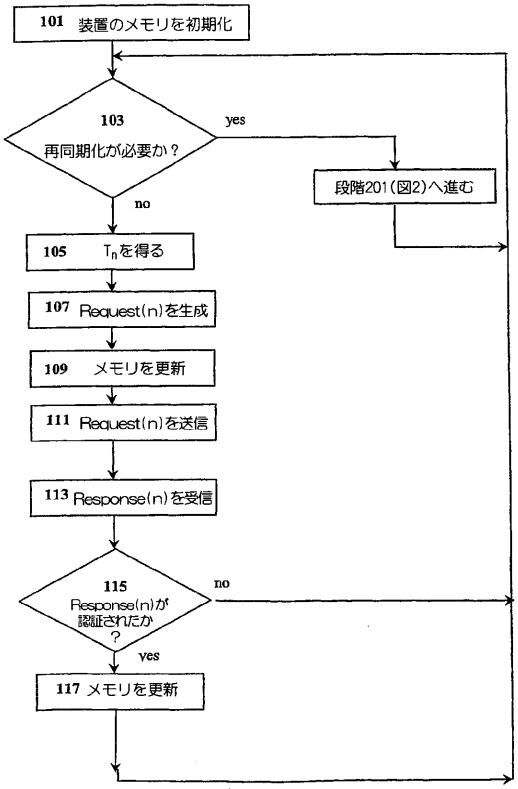
$$\mathrm{Response}(n) = \mathrm{Protocol\ Header},\ T_{n-1},\ \{T_{nTS},\ T_n,\ H(\mathrm{Protocol\ Header},\ PT),\ MsgK'\}DevicePubK,\ \{PT\}MsgK'$$

【００７２】
本明細書内で説明した実施形態に対する種々の変更および修正が当業者には明らかとなることを理解すべきである。このような変更および修正については、本発明の真意および範囲から逸脱することなく、かつ、本発明に付随する利点を縮小することなく行うことができる。したがって、このような変更および修正は、添付の請求項内に包含されるように意図される。

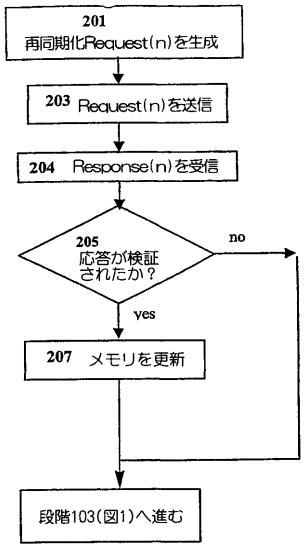
【図面の簡単な説明】
【図１】クライアント装置が、要請を信頼できるサーバーへ送信するために実行する工程のシーケンスのフローチャートを示す図である。
【図２】クライアント装置が、再同期化要請を信頼できるサーバーへ送信するために実行する工程のシーケンスのフローチャートを示す図である。
【図３】信頼できるサーバーが、クライアント装置からの要請に対する応答を送信するために実行する工程のシーケンスを示す図である。
【図４】信頼できるサーバーが、クライアント装置からの再同期化要請に対する再同期化応答を送信するために実行する工程のシーケンスを示す図である。

【符号の説明】
ＭｓｇＫ，ＭｓｇＫ’ メッセージ鍵
Ｒｅｑｕｅｓｔ（ｎ） 要請メッセージ
Ｒｅｓｐｏｎｓｅ（ｎ） 応答メッセージ
T_n ワンタイムパスワード

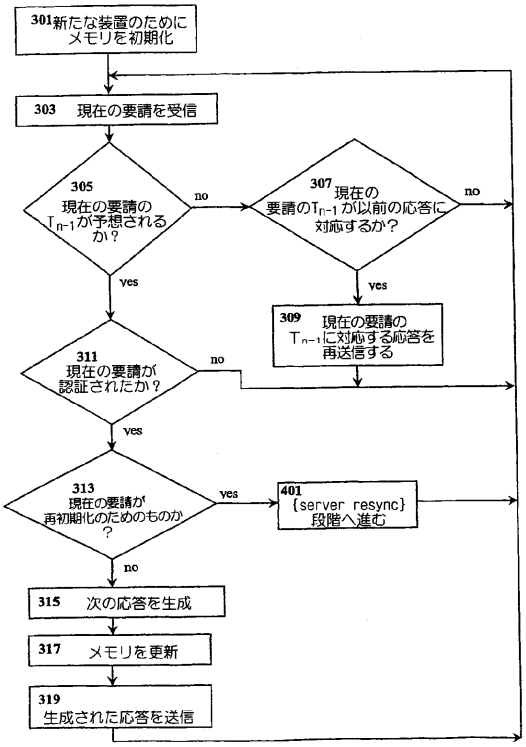
【図１】


101 装置のメモリを初期化
103 再同期化が必要か？
yes
段階201(図2)へ進む
no
105 T_nを得る
107 Request(n)を生成
109 メモリを更新
111 Request(n)を送信
113 Response(n)を受信
115 Response(n)が認証されたか？
no
yes
117 メモリを更新

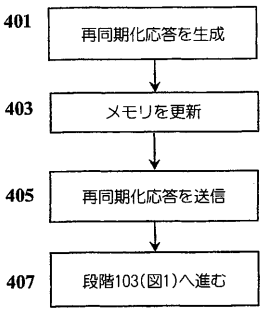
【図２】


201 再同期化Request(n)を生成
203 Request(n)を送信
204 Response(n)を受信
205 応答が検証されたか？
no
yes
207 メモリを更新
段階103(図1)へ進む

【図３】


301新たな装置のためにメモリを初期化
303 現在の要請を受信
305 現在の要請のT_{n-1}が予想されるか？
no
307 現在の要請のT_{n-1}が以前の応答に対応するか？
no
yes
yes
309 現在の要請のT_{n-1}に対応する応答を再送信する
311 現在の要請が認証されたか？
no
yes
313 現在の要請が再初期化のためのものか？
yes
401 {server resync}段階へ進む
no
315 次の応答を生成
317 メモリを更新
319 生成された応答を送信

【図４】


401 再同期化応答を生成
403 メモリを更新
405 再同期化応答を送信
407 段階103(図1)へ進む

【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 May 2002 (30.05.2002)

PCT

(10) International Publication Number
WO 02/43309 A2

(51) International Patent Classification[7]: H04L

(21) International Application Number: PCT/US01/46290

(22) International Filing Date: 19 October 2001 (19.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/242,083 20 October 2000 (20.10.2000) US
60/246,843 8 November 2000 (08.11.2000) US

(71) Applicant: WAVE SYSTEMS CORPORATION [US/US]; 480 Pleasant Street, Suite B200, Lee, MA 01238 (US).

(72) Inventor: KRAVITZ, David, W.; 3910 Ridgelea Drive, Fairfax, VA 22031 (US).

(74) Agents: BUTTER, Gary, M. et al.; Baker Botts, LLP, 30 Rockefeller Plaza, New York, NY 10112-0228 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/43309 A2

(54) Title: CRYPTOGRAPHIC DATA SECURITY SYSTEM AND METHOD

(57) Abstract: A method for communicating between a computer device and a trusted server is disclosed. According to the method of the invention, a one-time password for use in communication from the device to the server is generated. The device generates at least one one-time request-authentication datum that includes a function of at least a portion of a previous response from the server to a previous message from the device. The server then generates at least one one-time rseponse authentication datum that includes a function of at least a portion of at least one one-time password.

CRYPTOGRAPHIC DATA SECURITY SYSTEM AND METHOD

SPECIFICATION

BACKGROUND OF THE INVENTION

The present invention relates to improving security in data communications systems and in particular to systems and methods for providing confidentiality, trust, and attack-resistance for data that may be transmitted over insecure or dubiously-secure networks, such as the Internet.

Data communications, specifically communications between a plurality of computer users over a distributed data network, for instance, are known to be subject to multiple varieties of attacks by persons (henceforth referred to as "insiders" or "interceptors") not authorized by the communication parties or intended data recipients. Such attacks may be motivated by a desire to view private information, to commit financial or other fraud, or simply to corrupt communications integrity for whatever reason.

The use of the term "one-time" in the specification and claims is intended to reflect an enabled capability to specify a means and accommodate the results of dynamic update or replacement of certain passwords and datums. The degree of acceptable reuse of such "one-time" values from the perspective of a device or server is determined by the particular implementation and is not prescribed herein.

In the context of a network including a server computer and one or more client computers having access to data from said server (as, for instance, in a World Wide Web-based webserver context), a connection depletion attack, as defined in Juels, A. and Brainard, J., *Client Puzzles: A Cryptographic Countermeasure against Connection Depletion Attacks*, http://www.rsasecurity.com/rsalabs/staff/ajuels, 1999, first presented at the Network and Distributed System Security Symposium, San Diego, California, February 3, 1999 (hereinafter "Juels and Brainard") (herein incorporated by reference) is one in which the attacker seeks to initiate and leave unresolved a large number of connection (or

service) requests to a server, exhausting its resources and rendering it incapable of servicing legitimate requests.

A variety of attempts have been made in the art to increase resistance to connection depletion attacks.

Juels and Brainard addresses this type of denial-of-service problem without distinguishing between classes of clients. Juels and Brainard uses cryptographic "puzzles" which are dynamically changed to discourage an outsider break-in.

Another approach, published at http://www.rsasecurity.com/products/securid/datasheets/dsauthenticators.html (hereinafter "Dsauthenticators"), uses SecurID authenticators. These are hardware or software tokens each providing a sequence of one-time passwords based on a token-unique key applied successively in the context of a proprietary algorithm. The client-side host transmits the current one-time password and a constant PIN or passphrase to a server to which it wants to identify itself. A server that possesses knowledge of the token-unique keys can synchronize with the client tokens, and thereby recognize the (remote) presence of the particular client upon receipt of the one-time password and PIN. This is a self-synchronizing system, in which the client token does not adjust its behavior based on inputs from the server on a per-transaction basis. Furthermore, the system is designed to provide entity authentication, but not authentication of the origin or integrity or the "freshness" of any ensuing communications.

The method described in Rivest, R., Shamir, A., and Adleman, L., *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the A.C.M.* 1978, 21, 120-26 (hereinafter "Rivest, Shamir and Adleman") (and enhanced based on Bellare, M., and Rogaway, P., *Optimal Asymmetric Encryption – How to Encrypt with RSA*, November 19, 1995 (revised version of Optimal Asymmetric Encryption Padding paper: http://www-cse.ucsd.edu/users/mihir/papers/oaep.html; earlier version published in *Advances in Cryptology – Eurocrypt 94, Lectures in Computer Science,* A. DeSantis Ed., Springer Verlag, 1994, 950, 92-111 (hereinafter "Bellare and Rogaway") as explained further in Johnson, D. B., and Matyas, S. M., *Asymmetric Encryption: Evolution and*

Enhancements, Cryptobytes, Spring 1996, Volume 2, No. 1 (*see also* http://www.rsalabs.com/cryptobytes) (hereinafter "Johnson and Matyas") provides a means for two parties to secure the confidentiality of their communications, where the transmitting party employs the public key of the receiving party for the purpose of encryption and the receiving party employs its corresponding private key for the purpose of decryption (recovery of plaintext). The method is asymmetric in that the two parties use keys that are distinct from each other, although they are algorithmically related or paired. The method in Rivest, Shamir and Adleman can also be used to instantiate a digital signature capability, where the signing party applies its private key to the message to be signed in accordance with the method, and the verifying party applies the corresponding public key in accordance with the method in order to verify the authenticity of the origin and the integrity of the message. Digital signatures, in and of themselves, do not provide evidence of freshness; i.e., a previously used message may be replayed without being detected as a "stale" message.

Two parties can communicate using a symmetric-key encryption algorithm, such as DES. In this case, the same key is known to both parties. DES can also be used to provide a message authentication code (MAC) capability. Thus, a receiving party which possesses knowledge of the secret key can determine that the originator of the message also had knowledge of the secret key and that the message has not been altered in transit.

Messages or portions thereof can be encrypted to conceal the identity of the client from parties other than the server, and to make it more difficult to link transactions as having come from the same client. In this case, the server needs to apply the decryption algorithm before performing any processing which requires knowledge of the party's identity. If digital signatures are applied to messages, an adversary can use the list of public keys to group the communications transactions according to the signers, since messages verified using the incorrect public key should fail verification. If the signatures are encrypted, or the signature is computed over the plaintext message where the message is transmitted in encrypted form, then verification of the signature requires preliminary decryption.

Thus, there is a need for a secure communications method that does not require signature verification. There is also a need for a secure method that is not self-synchronizing, so that the server is not required to possess knowledge of token-unique keys as well as self-regulated input to the one-time password update algorithm, such as time or counters in order to synchronize with the client tokens. There is also a need for a method that does not abridge privacy by enabling unauthorized access to client-identifying information. Finally, there is a need for a secure method that takes advantage of registered client devices that can transmit patterns according to a protocol, where the server can differentiate such patterns from other incoming Internet traffic. The prior art is not believed to meet these needs.

SUMMARY OF THE INVENTION

The present invention is directed to a method for communicating between a computer device and a trusted server. The method includes the steps of: (a) generating a one-time password for use in communication from the device to the server; (b) generating at least one one-time request-authentication datum that includes a function of at least a portion of a previous response from the server to a previous request from the device; and (c) generating at least one one-time response-authentication datum that includes a function of at least a portion of at least one one-time password. Preferably, the one-time request-authentication datum or the one-time response-authentication datum or both comprise a function of an encryption key. At the point at which a one-time password is "used" in a communication from a device to a server as associated with a request, the one-time password may be exposed to interception, while the dependence of a response-authentication datum on a one-time password is with respect to a one-time password that has not yet been so used. Thus the transmission of a response message may be considered part of the (secure) negotiation or exchange of a one-time password which precedes its actual use during a later request. The encrypted transmission of a one-time password or a component thereof within a request message for the purpose of conveying knowledge of this information to a server equipped with the capability to execute the corresponding decryption is not considered use of the one-time password. Interception of a response from a server to a device does not enable successful generation or verification of a

one-time request-authentication datum. Interception of a request from a device to a server does not enable successful generation or verification of a one-time response-authentication datum.

Another object of the invention is to provide a method for transmitting a data request from a client device, comprising: (a) generating a one-time password; and (b) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from a trusted server to a previous request from the device. Preferably, the one-time request-authentication datum comprises a function of an encryption key.

Another object of the invention is to provide a method for transmitting a response from a trusted server to a request from a client device, comprising: (a) receiving a request comprising a function of at least a portion of at least one one-time password shared between the device and said server; and (b) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password. Preferably, the one-time response-authentication datum comprises a function of an encryption key.

Another object of the invention is to provide a system for enhancing trust in communications between a client device and a trusted server, comprising: (a) means for establishing a network connection between the client device and the server; and (b) means for conducting communications of data with the client device over the network connection, wherein the communications between the device and the server are conducted in accordance with a method comprising: (i) generating a one-time password for use in communication from the device to the server; (ii) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from the server to a previous request from the device; and (iii) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password. Preferably, the system further comprises an encryption algorithm and means for downloading the encryption algorithm to the client computer over the network connection, wherein the means for conducting communications of data with the client computer over the network connection is in accordance with the encryption algorithm and wherein the

communications between the device and the server are conducted on an encrypted basis.

Another object of the invention is to provide a system for enhancing trust in communicating a data request from a client device, comprising: (a) means for establishing a network connection between the client device and a trusted server; and (b) means for conducting communications of data with the client device over the network connection, wherein the communications between the device and the server are conducted in accordance with a method comprising: (i) generating a one-time password; and (ii) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from a trusted server to a previous request from the device. Preferably, the system further comprises an encryption algorithm and means for downloading the encryption algorithm to the client computer over the network connection, wherein the means for conducting communications of data with the client computer over the network connection is in accordance with the encryption algorithm and wherein the communications between the device and the server are conducted on an encrypted basis.

Another object of the invention is to provide a system for enhancing trust in communicating a response from a request from a client device to a trusted server, comprising: (a) means for establishing a network connection between the client device and the server; and (b) means for conducting communications of data with the client device over the network connection, wherein the communications between the device and the server are conducted in accordance with a method comprising: (i) receiving a request comprising a function of at least a portion of at least one one-time password shared between the device and the server; and (ii) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password. Preferably, the system comprises an encryption algorithm and means for downloading the encryption algorithm to the client computer over the network connection, wherein the means for conducting communications of data with the client computer over the network connection is in accordance with the encryption algorithm and wherein the

communications between the device and the server are conducted on an encrypted basis.

The present invention is also directed to a method for resynchronizing the communication between a client device and a trusted server, which includes the steps of: (a) generating or retrieving a one-time password for use in communication from the device to the server; (b) generating or retrieving at least one one-time request-authentication datum that includes a function of at least a portion of a previous response from the server to a previous request from the device; and (c) generating or retrieving at least one one-time response-authentication datum that includes a function of at least a portion of at least one one-time password. In one preferred embodiment, the one-time request-authentication datum comprises an All NULLs message encryption key. In another preferred embodiment, the one-time response-authentication datum comprises an All NULLs message encryption key. The method may be configured so that a request received by the server that uses a one-time password that is not recognized as current by the server may result in the transmission of a previously generated response if any at all. A resynchronization request message is considered to be (one type of) a request message. A resynchronization response message is considered to be (one type of) a response message.

Another object of the invention is to provide a method for transmitting a resynchronization request from a client device, comprising: (a) generating or retrieving a one-time password; and (b) generating or retrieving at least one one-time request authentication datum comprising a function of at least a portion of a previous response from a trusted server to a request from the device. In a preferred embodiment, the one-time request-authentication datum comprises an All NULLs message encryption key. In another preferred embodiment, the resynchronization request comprises an encrypted resynchronization datum that replaces a previous resynchronization datum.

Another object of the invention is to provide a method to transmit a resynchronization response from a trusted server, comprising: (a) receiving a request comprising a function of at least a portion of at least one one-time password

associated with a client device; and (b) generating or retrieving at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password. In a preferred embodiment, the one-time response-authentication datum comprises an All NULLs message encryption key. In another preferred embodiment, the resynchronization response comprises an encrypted resynchronization datum that replaces a previous resynchronization datum.

Another object of the invention is to provide a system for resynchronizing communication between a client device and a trusted server, comprising: (a) means for establishing a network connection between the client device and the server; and (b) means for conducting communications of data with the client device over the network connection, wherein the communications between the device and the server are conducted in accordance with a method comprising: (i) supplying a one-time password for use in communication from the device to the server; (ii) supplying at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from the server to a previous request from the device; and (iii) supplying at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password. Preferably, the system comprises an encryption algorithm and means for downloading the encryption algorithm to the client computer over the network connection, wherein the means for conducting communications of data with the client computer over the network connection is in accordance with the encryption algorithm and wherein the communications between the device and the server are conducted on an encrypted basis.

Another object of the invention is to provide a system for enhancing trust in transmission of a resynchronization request from a client device, comprising: (a) means for establishing a network connection between the client device and a trusted server; and (b) means for conducting communications of data with the client device over the network connection, wherein the communications between the device and the server are conducted in accordance with a method comprising: (i) supplying a one-time password; and (ii) supplying at least one one-time request authentication datum comprising a function of at least a portion of a previous response from the

server to a request from the device. Preferably, the system comprises an encryption algorithm and means for downloading the encryption algorithm to the client computer over the network connection, wherein the means for conducting communications of data with the client computer over the network connection is in accordance with the encryption algorithm and wherein the communications between the device and the server are conducted on an encrypted basis.

Another object of the invention is to provide a system for enhancing trust in transmission of a resynchronization response from a trusted server, comprising: (a) means for establishing a network connection between a client device and the server; and (b) means for conducting communications of data with the client device over the network connection, wherein the communications between the device and the server are conducted in accordance with a method comprising: (i) receiving a request comprising a one-time password associated with a client device; and (ii) supplying at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password. Preferably, the system of claim 34, further comprising an encryption algorithm and means for downloading the encryption algorithm to the client computer over the network connection, wherein the means for conducting communications of data with the client computer over the network connection is in accordance with the encryption algorithm and wherein the communications between the device and the server are conducted on an encrypted basis.

The present invention uses a tightly integrated approach in order to provide simultaneous coverage of various aspects of efficient client – trusted server bi-directional communications security. Unlike Juels and Brainard, the present invention takes advantage of the fact that registered client devices form a distinguished class of clients which can transmit patterns according to a protocol which can be differentiated from other incoming Internet traffic by the server. The invention uses the method in Rivest, Shamir and Adleman (as enhanced based on Bellare and Rogaway) in order to securely transmit components of one-time passwords and of one-time-use MAC keys that are used in subsequent messages for bi-directional message origin, integrity, and freshness, as well as unlinkability of

client messages, in association with the use of encryption and MACs (in accordance with FIPS 46-3 *Data Encryption Standard* and FIPS 81 *DES Modes of Operation* (MACing), published at http://csrc.nist.gov/cryptval/des.htm (hereinafter "FIPS")). Consequently, unlike DSauthenticators, the method is not self-synchronizing; resynchronization is handled efficiently on the server end by retransmission. The server does not need to track or distinguish between legitimate and fraudulent requests which are communicated using previously (versus currently) valid one-time passwords, because no (potentially resource-intensive) cryptographic processing is done by the server in such cases; a retrieval and (re-)transmission of a previously generated response may be done, without the need for further computation or database-updating.

Message processing on the server end is handled by a denial-of-service resistant phased approach, which first dispels request messages (as candidates for further new processing) that are not accompanied by a currently legitimate one-time password. The inclusion of a currently legitimate one-time password results in a "hit" on the server database, in which case the one-time password is used for database lookup of information pertaining to a single client device. If the one-time-use MAC key in that database entry when applied to the appropriate data fields of the request message indicates message compliance, RSA decryption is done using the server's private key (which can be secured within a crypto module or hardware security module (HSM) at the server). RSA decryption uncovers information pertaining to the next one-time password and the message key (if present) which is used to decrypt that portion of the request message, if any, which was transmitted using a bulk encryption algorithm (such as a variant of DES). The server computes a response message, that incorporates a message authentication code (MAC) computed using a current MAC key derived, at least in part, by using knowledge of the next one-time password or a component thereof that was transported within the most recently received request message. The response message may also convey a freshly generated message key and a component of the next one-time-use MAC key for the client's next request. The means of conveyance may be encryption under the client's public key as pointed to in the server database. The response message may also include bulk-encrypted data,

where the corresponding plaintext can be recovered using the (response-)message key. The (encryption-capable) device of the present invention refers to the client device, and not the server or a hardware security module (HSM) at the server.

The public/private key pairs of both the device and the device server (i.e., trusted server) are used to update the shared secrets necessary to negotiate secure communications on a transactional basis. This offers several advantages over the standard techniques of signing encrypted communications or encrypting signed communications with respect to privacy, computational overhead, and denial-of-service attacks. A purely symmetric key approach would lead to the possibility of attack based on a static snapshot of values in the device server database. If a device loses cryptosynchronization with the device server because of an incomplete transaction, synch is reestablished without providing privacy-threatening linkage between the aborted and subsequent transactions and without having the device accept outdated or unwanted information. Given the list of device public keys, one cannot partition transactions according to which devices were involved.

The use of Optimal Asymmetric Encryption Padding (OAEP) along with RSA thwarts attempts to link transactions by encrypting data which is revealed during the protocol and trying to match the ciphertext to previous transactions. Request- and Response- message keys are independently generated so that obtaining a snapshot of the device server database would not afford one the opportunity to put forth a Request message with an emulated device which results in a Response message encrypted in the known message key used within the Request. With respect to denial-of-service, the system takes advantage of the fact that registered devices form a distinguished class in that their output can be differentiated at the server from other incoming Internet traffic. If the use of the one-time password within the incoming request message results in a fresh (or current) hit in the device server database, the server uses the "hit" device entry to check the MAC, followed by the RSA decryption to recover the message key, and symmetric algorithm decryption with the message key to recover the plaintext. If the use of the one-time password within the incoming request message refers to the transaction that the device server has just-previously processed, the server retransmits the previous response without

incurring additional processing or database updates. The secure communications protocol is designed so that if critical operations are executed within a secure-crypto-(or hardware-security-) module at the device server, unauthorized database access would not in itself undermine the integrity of the system.

The secure communications protocol described below does not require the use of public-key cryptography for the purpose of digital signatures, but rather only for encryption (and decryption). From an efficiency point of view, it is important to note that as successful verification of secure communications serves as an indication of a properly registered functioning device, any digital signatures which are generated and transmitted by the device within the "plaintext" can be archived and later verified out-of-band, offline from the transaction processing. The plaintext is encrypted under the message key, where the encrypted message key and encrypted plaintext are authenticated by a MAC under secure communications. Provided that any included signatures are full signatures, in that they are accompanied by the text that is signed, the secure communications protocol serves to authenticate the text independently of the non-repudiation capability enabled by the inclusion of signatures.

A properly functioning device would not accept bogus signatures within secure communications requests, because the generation and handling of these signatures would be controlled by the device. Consequently, the method of the present invention may be implemented so as not to require verification, by the server, of signatures at the same time as the message. A signature can be treated as a "blob" in that it can be stored and verified afterwards. To handle real-time authentication, client generation and server verification of a message authentication code (MAC) is employed instead. For this purpose, a symmetric key is used. The method of the present invention may further comprise a procedure to establish the freshness of the message, that is, whether the message has been previously used or intercepted prior to current receipt.

One advantage of the method of the present invention is that a message cannot be linked to a client. Thus, for example, if a message is intercepted, the present method does not enable the interceptor to tell where the message came from.

The method of the present invention is not self-synchronizing. Thus, the proper functioning of the method does not require that the client device periodically adjust its behavior based on server input so that the device and server can independently maintain synch between such realignments. Instead, synchronization is recovered on a transactional basis through retransmission from, or synchronization message processing by, the server.

Accordingly, the present invention includes a secure communications method that does not require signature verification. It further provides a secure method that is not self-synchronizing. Finally, since the client devices are registered with the trusted server, they form a distinguished class of devices that transmit patterns, according to a protocol, that can be differentiated from other incoming Internet traffic by the server.

BRIEF DESCRIPTION OF THE FIGURES

Figure 1 shows a flow chart of the sequence of operations the client device performs to transmit a request to the trusted server.

Figure 2 shows a flow chart of the sequence of operations the client device performs to transmit a resynchronization request to the trusted server.

Figure 3 shows the sequence of operations the trusted server performs to transmit a response to a request from the client device.

Figure 4 shows the sequence of operations the trusted server performs to transmit a resynchronization response to a resynchronization request from the client device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS OF THE INVENTION

The following is a discussion of several particularly useful embodiments of the present invention. The trusted server according to the present invention preferably comprises two components. The first component is a host processor and database capable of tracking state changes. The second component is a hardware security module (HSM) equipped with cryptographic processing capability and secured storage of fixed values.

The client device according to the present invention likewise preferably comprises two components. The first component is a processor. The second component is a co-processor in a secure environment. Data may be encrypted by the co-processor of the client device and intended for the HSM of the trusted server. Thus, the encrypted data can only be decrypted by the HSM, and not, for example, by an insider at the trusted server. The HSM of the trusted server and the co-processor of the client device each have a private key. In addition, the public key for each particular client device is recognized by the HSM. The public-private key pair for each device may be defined at the production or registration stage.

The HSM also makes the decision whether the decrypted data is released for storage into a database on the trusted server. When a next message is prepared by the HSM for transmission to a client device, the HSM needs to access the database on the trusted server for data specific to that particular client device. The device will not accept the message as valid unless the HSM has demonstrated knowledge of data that is device-specific and current. The same holds true for a message sent by the client device to the trusted server – the HSM will not accept the message for response processing unless it can access data in the trusted server database for that client device that accurately corresponds to the received message. The HSM will not accept the message unless the creator of the message had current access to the device or the corresponding database entry.

The method of the invention uses a one-time password that is incorporated into the request message sent from the client device to the trusted server. The one-time password in the request-authentication datum may be used by the server to locate an entry in its database that corresponds to the client device. While the "current" one-time password is thus included, this authenticated request message and the resulting and confirming response message from the server include an exchange of data which sets the "next" one-time password at both the device and the server, without exposing or disclosing its value. Preferably, the request-authentication datum comprises an encrypted secret datum, wherein the server decrypts the encrypted secret datum to recover the secret datum. Preferably, a next or later request comprises a function of at least a portion of at least one one-time password comprising at least a

portion of at least one secret datum. The private exchange of data ensures that even if an insider gets a "snapshot" of the trusted server database, he would be unable to use this acquired knowledge alone in order to deceive the HSM into thinking that he is the client once the actual client has caused an update of the database by successfully transacting with the trusted server. Furthermore, an insider cannot interpret a response from the HSM, even if he has submitted the request based on current database access, because he lacks the device private key and because the message key used to encrypt plaintext within the response is not derivable based solely on knowledge of the message key in the request. An insider also cannot interpret an incoming message from the client, because he lacks the HSM private key.

In a preferred embodiment, the protocol used is defined as follows:

Terminology And Notation

{x}EntityPubK represent RSA-OAEP (optimal asymmetric encryption padding) encryption of a message x under the RSA public key of Entity.

{PT}MsgK represents the symmetric-algorithm (e.g., triple-DES) encryption of (plaintext) PT under message key MsgK.

MAC(data)Key represents the symmetric-algorithm based MAC of data under the key Key.

Protocol Header comprises cleartext data (i.e., data transmitted unencrypted), which may include items pertaining to protocol version or other data that is useful to receive prior to further processing and not sensitive to disclosure. The Protocol Header data field, if not of fixed length, may include a fixed length preamble which specifies its length.

The use of a comma (",") between arguments or data fields indicates concatenation. [a,b] indicates the concatenation of a followed by b.

".XOR." denotes bit-wise exclusive-or, i.e., component-wise addition modulo-2 of like-length vectors. MAC(data)K_1.XOR.K_2 is the MAC value that results from operating over "data" using K_1.XOR.K_2 as the key.

H(m) indicates the result of applying a one-way hash function (e.g., SHA-1) to a message m.

Device-Side Basic Flow

Suppose that at the conclusion of successful registration of the client device (in accordance with known methods), the device and the trusted server share two secret values denoted by T_0 and T_{0TS}, and each maintains a reliable copy of the other's public key. For this embodiment, the generation of T_0 and T_{0TS} may be such that T_{0TS} .XOR. T_0 is a (2-key triple-DES) key. Reference is now made to Figure 1. In general, if the device and the trusted server are in (crypto-)synchronization, the persistent memory of the device prior to beginning the process of Request(**n**) is:

T_{n-1}, T_{n-1TS}, Blank,

where T_{n-1} is a one-time password.

If resynchronization with the trusted server is needed (**103**) as evidenced by a value other than "Blank" in that data position, the device generates a resynchronization request as further discussed below. If resynchronization is not needed, when the device wants to initiate Request(**n**), it derives (**105**) a new 2-key triple-DES key X, and lets T_n = X .XOR. T_{n-1TS}. Here T_{n-1TS} is generated by the trusted server in a previous response. Accordingly, the Request(**n**) comprises a function of at least a portion of a previous response. The device generates a request message (**107**), Request(**n**) (see expression below), where PT (Plain Text) is that portion of the client-side user's message content to be delivered in bulk-encrypted form. A 24-byte triple-DES key (MsgK) is generated. PT is triple-DES encrypted with MsgK. The concatenation of T_n and MsgK is then OAEP-padded and RSA-encrypted with the trusted server's (TS) public key. A CBC (cipher-block-chaining) MAC is generated over the Protocol Header concatenated with the "data" [{T_n,MsgK}TSPubK, {PT}MsgK]. The MAC is generated using T_{n-1TS} .XOR. T_{n-1}. Note that T_{n-1TS} .XOR. T_{n-1} is 16 bytes, so a double key is used to run the triple-DES algorithm when calculating the MAC. The Protocol Header and T_{n-1} are prepended to the MAC. The data is appended after the MAC.

In this request, T_n and MsgK are freshly generated values.

Therefore, T_n and the message key MsgK are encrypted using the public key of the server for the purpose of transport to the server. Since the client

device generates a new message key for every request, no memory is required to store the message key.

Request(**n**) = Protocol Header, T_{n-1},

MAC(Protocol Header, data)T_{n-1TS}.XOR.T_{n-1}, data,

where data = { T_n,MsgK}TSPubK,{PT}MsgK.

Prior to transmitting Request(**n**), the device goes into the following persistent memory state (**109**):

T_{n-1}, T_{n-1TS}, T_n. The Request(**n**) is then transmitted (**111**), and a Response(**n**) is transmitted from the server to the device(**113**). The server-side flow is discussed below.

Upon receiving a Response(**n**), the device fully processes the response since MsgK, having been generated randomly (or pseudo-randomly) is not the ALL NULLS vector which would indicate that the device is in resynchronization mode (discussed below) rather than in basic-flow mode. At the conclusion of satisfactory verification (**115**) of Response(**n**) from the trusted server, the device goes into the following persistent memory state (**117**):

T_n, T_{nTS}, Blank.

Device-Side Flow - Re-establishing Cryptosynchronization

Reference is now made to Figure 2. Suppose, for example, the client device has timed out or the client-side processor has crashed, so that the device is in persistent memory state T_{n-1}, T_{n-1TS}, T_n and is not awaiting Response(**n**). When operation of the client device resumes, it generates a message key that indicates that the device is in resynchronization mode. Preferably, the client device generates a NULL MsgK and transmits a special resynchronization request (**201**) using the NULL MsgK. No PT is present:

Request(**n**) = Protocol Header, T_{n-1},

MAC(Protocol Header, data)T_{n-1TS}.XOR.T_{n-1}, data,

where data = {T_n, MsgK}TSPubK, and where the encryption key MsgK = All NULLs.

The device knows that it is in cryptosynchronization mode and not in normal transmit (basic-flow) mode: The fact that in volatile memory MsgK = All

NULLs informs the client device to disregard any {PT}MsgK' field when verifying a received Response(**n**). The response may include such an encrypted-data field in the event that it is actually a stored response first generated in response to a basic-flow- rather than a resynchronization- request. Since the MAC in the response, in this case, is computed over ciphertext {PT}MsgK' rather than plaintext PT, the device does not need to do the decryption in order to verify the MAC.

At the conclusion of satisfactory verification of Response(**n**) from the trusted server (**205**), the device updates memory to the following persistent memory state (**207**):

T_n, T_{nTS}, Blank

Server-Side Basic Flow

Prior to first receiving a Request(**n**) from the device for a given value of **n**, the trusted server database values for that device are:

T_{n-2}, T_{n-1}, T_{n-1TS}, Response(**n-1**)

Upon satisfactory verification of Request(**n**), the trusted server generates T_{nTS} and Response(**n**), and its database values are:

T_{n-1}, T_n, T_{nTS}, Response(**n**)

Reference is now made to Figure 3. Upon receipt of the request from the client device (**303**), the trusted server establishes which messaging protocol to use based on the Protocol Header's version field. The trusted server establishes the identity of the client device based on T_{n-1} and retrieves T_{n-1TS} from the database entry using T_{n-1} (**305**). The server validates the MAC using T_{n-1TS} .XOR. T_{n-1}. Using its private key, the trusted server decrypts and OAEP-decodes {T_n,MsgK}TSPubK, and saves T_n. The recovered value of MsgK is used to decrypt {PT}MsgK to recover PT. The trusted server then processes PT accordingly.

In one embodiment of the invention, if there is no current one-time password entry in the server database that corresponds to the incoming value T_{n-1}, the server attempts to match the incoming value against a T_x corresponding to a previously generated response **(307).** If the look-up is successful, the server retransmits the response corresponding to T_{n-1} of the incoming request (**309**). More specifically, if the trusted server's values for a given device are currently T_{n-1}, T_n,

T_{nTS}, Response(**n**), then T_{n-1} is not used by the trusted server to freshly process the request. Instead, the corresponding Response(**n**) is used to re-establish (crypto)synchronization between the trusted server and the device, in the event that the trusted server has updated its database entry for the device, but the device has not updated its state.

If T_{n-1} in the incoming request is expected (in that it matches a current one-time password) and the request is authenticated, then the server verifies that the request is not a request for resynchronization (**313**). If resynchronization is needed, a resynchronization response is generated as is further discussed below. If resynchronization is not needed, the trusted server generates a new 2-key triple-DES key Y, and lets T_{nTS} = Y .XOR. T_n, and generates a Response(**n**) (**315**) with its own PT and a freshly generated MsgK'. The response message is generated in the same format as the request message except that the MAC is calculated using T_n.XOR.T_{n-1TS} and the new T_{nTS} appears as an argument encrypted under the device's public key DevicePubK:

Response(**n**) = Protocol Header, T_{n-1},
MAC(Protocol Header, data)T_n.XOR.T_{n-1TS}, data,
where data = { T_{nTS} ,MsgK'}DevicePubK,{PT}MsgK'.

The generated Response(**n**) includes a confirming function of the next one-time password T_n via the key used to calculate the response MAC.

In this response, T_{nTS} and MsgK' are freshly generated values, wherein the message key that is returned (MsgK') is different from the message key generated by the device (MsgK).

Neither request message key MsgK nor response message key MsgK' is saved into the database. After the HSM decrypts the request and generates a response, timely access to the database values T_n (new) and T_{n-1TS} (previous) would enable a substitution of the response with a different one that would be acceptable to a compliant client platform. But if the protocol is modified so that the client device (or the inclusive client platform) expects MsgK .XOR. MsgK' (instead of MsgK') to be sent encrypted using the client platform's public key, then a substitution of response would not be accepted since neither MsgK, nor the MsgK' as generated by the HSM,

should leave the HSM. The client device still does not need to store its MsgK in non-volatile memory, since it only reestablishes cryptosynchronization and thus ignores any bulk-encrypted content of the response message if the first response is not received when expected.

In persistent memory, the trusted server previously had (in some form of accessible storage): T_{n-2}, Response(**n-1**), T_{n-1}, T_{n-1TS}. This is now replaced with: T_{n-1}, Response(**n**), T_n, T_{nTS} (**317**). Knowledge of T_{n-1TS} is no longer required once Response(**n**) has been generated and saved.

The trusted server sends Response(**n**) to the client device (**319**). Upon receipt of the message, the device verifies the Protocol Header's version and ignores T_{n-1}. The MAC is verified using T_n.XOR.T_{n-1TS} . Using its private key, the client device decrypts and OAEP decodes { T_{nTS} ,MsgK'}DevicePubK. The device uses MsgK' to decrypt PT. The device processes the PT accordingly.

In persistent or non-volatile memory, the device previously had: T_{n-1}, T_{n-1TS}, T_n. This is now replaced with: T_n, T_{nTS}, Blank.

Server-Side Flow - Re-establishing Cryptosynchronization

Referring now to Figure 4, the server decrypts and processes the message. If the trusted server's database values for that device are T_{n-2}, T_{n-1}, T_{n-1TS}, Response(**n-1**) when it receives this Request(**n**), it processes the request and generates T_{nTS} and Response(**n**) (**401**) using a message key that indicates that the device is in resynchronization mode. Preferably, the client device generates a NULL MsgK'. No PT is present. The server then transmits Response(**n**) (**405**):

Response(**n**) = Protocol Header, T_{n-1},

MAC(Protocol Header, data)T_n.XOR.T_{n-1TS}, data,

where data = {T_{nTS}, MsgK'}DevicePubK, and where the encryption key MsgK' = All NULLs

Its database values are updated to include T_{n-1}, T_n, T_{nTS}, and Response(**n**) (**403**).

If the device server's database values for that device are T_{n-1}, T_n, T_{nTS}, Response(**n**), for some Response(**n**), when it receives this Request(**n**), it (re)transmits Response(**n**). In this case T_{n-1} within the received Request(**n**) is used to access the

database entry, i.e., the previously transmitted and stored value of Response(n). If the request had been "fresh," T_n would have been used to access the database entry, i.e. T_{nTS}, and the device public key DevicePubK.

The client device processes the response message and updates its persistent memory as discussed above in connection with Figure 1 and Figure 2.

If the possibility of state loss at a server is of concern, an extension of the invention can be deployed to take advantage of availability of infrequently accessed fail-safe backup. For example, in the case of exception processing with respect to "Duress mode" (as exemplified below), the server may be able to access a remote backup service or facility which acknowledges uncorrupted receipt of backup-request messages. In the event of detected or suspected state loss at the server, the server would retrieve a copy of the data that it had deposited with the backup facility. In an embodiment of this state-recoverability method: When the device and server agree on initial values T_0 and T_{0TS} as part of registration or other initialization, they also agree on an initial pair of Duress values, Duress-T_0 and Duress-T_{0TS}. If resynchronization as described thus far, fails to achieve the desired effect (of regaining or re-establishing (crypto-)synchronization) after a prescribed number of attempts or a prescribed elapse of time (or other metric) as may be tracked by the device (or device user) utilizing known methods, an exception processing version of resynchronization may be employed. It is understood that the term device-side resynchronization comprises the exception processing or duress mode version of device-side processing. It is understood that the term server-side resynchronization comprises the exception processing or duress mode version of server-side processing. A duress request message is considered to be (one type of) a request message. A duress response message is considered to be (one type of) a response message. The device generates and transmits a Duress request message. This follows the format of a standard request message, as does the resulting Duress response message relative to a standard response message, with certain qualifications. Namely, the current Duress-T values rather than the current (standard) T values are used within the Duress Request and Duress Response; the newly generated T values in the Duress Request and Duress Response, respectively, are used to reset to a new "just-registered" starting

point and hence are designated here as T_0 and T_{0TS}, respectively (but are unrelated to the original T_0 and T_{0TS} values). The PT field of Duress Request(**m**) includes (at least) Duress-T_m, and the PT field of Duress Response(**m**) includes (at least) Duress-T_{mTS}.

Duress Request(**m**) = Protocol Header, Duress-T_{m-1},
MAC(Protocol Header, data)Duress-T_{m-1TS}.XOR.Duress-T_{m-1}, data,
where data = { T_0,MsgK}TSPubK,{Duress-T_m}MsgK.

Duress Response(**m**) = Protocol Header, Duress-T_{m-1},
MAC(Protocol Header, data)Duress-T_m.XOR.Duress-T_{m-1TS}, data,
where data = { T_{0TS} ,MsgK'}DevicePubK,{Duress-T_{mTS}}MsgK'.

Unlike standard request processing by the device, a retry of a failed Duress Request message is an exact copy of the previous (failed) attempt. Unlike standard server database updating, when the server locally updates from
Duress-T_{m-2}, Duress-T_{m-1}, Duress-T_{m-1TS}, Duress Response(**m-1**), to
Duress-T_{m-1}, Duress-T_m, Duress-T_{mTS}, Duress Response(**m**),
this change is also backed up using fail-safe communications or other ultra-reliable means.

An alternative embodiment of the basic invention would combine the authentication and public key encryption phases, thus eliminating the use of the MAC. This is a less-phased approach in that a hit on the server database indicating that an incoming request is current would precipitate RSA-OAEP processing of the request message at the server prior to verifying the authenticity of the request-message data. In the MAC-based embodiment, failure of the MAC to verify precipitates an abort to message processing at the server. An embodiment of the request and response messages in the MAC-less approach could employ a one-way hash function H, such as SHA-1:

Request(**n**) = Protocol Header, T_{n-1},
{T_{n-1TS}, T_n, H(Protocol Header, PT), MsgK}TSPubK,
{PT}MsgK; and

Response(n) = Protocol Header, T_{n-1},

{T_{nTS}, T_n, H(Protocol Header, PT),

MsgK'}DevicePubK, {PT}MsgK'.

It should be understood that various changes and modifications to the embodiments described herein will be apparent to those skilled in the art. Such changes and modifications can be made without departing from the spirit and scope of this invention and without diminishing its attendant advantages. It is therefore intended that such changes and modifications be covered in the appended claims.

CLAIMS

1. A method for enhancing trust in communications between a client device and a trusted server, comprising:

(a) generating a one-time password for use in communication from the device to the server;

(b) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from the server to a previous request from the device; and

(c) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

2. The method of claim 1, wherein said one-time request-authentication datum comprises a function of an encryption key.

3. The method of claim 1, wherein said one-time response-authentication datum comprises a function of an encryption key.

4. A method for enhancing trust in communicating a data request from a client device, comprising:

(a) generating a one-time password; and

(b) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from a trusted server to a previous request from the device.

5. The method of claim 4, wherein said one-time request-authentication datum comprises a function of an encryption key.

6. A method for enhancing trust in communicating a response from a request from a client device to a trusted server, comprising:

(a) receiving a request comprising a function of at least a portion of at least one one-time password shared between the device and said server; and

(b) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

7. The method of claim 6, wherein said one-time response-authentication datum comprises a function of an encryption key.

8. The method of claim 6, wherein said request comprises an encrypted secret datum, wherein said server decrypts said encrypted secret datum to recover said secret datum.

9. The method of claim 8, wherein a subsequent request comprises a function of at least a portion of at least one one-time password comprising at least a portion of at least one secret datum.

10. The method of claim 6, wherein the one-time password comprised within the request is used by the server to locate an entry in its database corresponding to the particular client device.

11. A method for resynchronizing communication between a client device and a trusted server, comprising:

(a) supplying a one-time password for use in communication from the device to the server;

(b) supplying at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from the server to a previous request from the device; and

(c) supplying at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

12. A method for enhancing trust in transmission of a resynchronization request from a client device, comprising:

(a) supplying a one-time password; and

(b) supplying at least one one-time request authentication datum comprising a function of at least a portion of a previous response from a trusted server to a request from the device.

13. The method of claim 12, wherein said resynchronization request comprises an encrypted resynchronization datum that replaces a previous resynchronization datum.

14. A method for enhancing trust in transmission of a resynchronization response from a trusted server, comprising:

(a) receiving a request comprising a one-time password associated with a client device; and

(b) supplying at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

15. The method of claim 14, wherein said resynchronization response comprises an encrypted resynchronization datum that replaces a previous resynchronization datum.

16. A system for enhancing trust in communications between a client device and a trusted server, comprising:

(a) means for establishing a network connection between the client device and the server; and

(b) means for conducting communications of data with the client device over the network connection,

wherein said communications between the device and the server are conducted in accordance with a method comprising:

(i) generating a one-time password for use in communication from the device to the server;

(ii) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from the server to a previous request from the device; and

(iii) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

17. The system of claim 16, further comprising

(a) an encryption algorithm, and

(b) means for downloading said encryption algorithm to the client computer over said network connection,

wherein said means for conducting communications of data with the client computer over the network connection is in accordance with said encryption algorithm and wherein said communications between the device and the server are conducted on an encrypted basis.

18. The system of claim 16, wherein said one-time request-authentication datum comprises a function of an encryption key.

19. The system of claim 16, wherein said one-time response-authentication datum comprises a function of an encryption key.

20. A system for enhancing trust in communicating a data request from a client device, comprising:

(a) means for establishing a network connection between the client device and a trusted server; and

(b) means for conducting communications of data with the client device over the network connection,

wherein said communications between the device and the server are conducted in accordance with a method comprising:

(i) generating a one-time password; and

(ii) generating at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from a trusted server to a previous request from the device.

21. The system of claim 20, further comprising

(a) an encryption algorithm, and

(b) means for downloading said encryption algorithm to the client computer over said network connection,

wherein said means for conducting communications of data with the client computer over the network connection is in accordance with said encryption

algorithm and wherein said communications between the device and the server are conducted on an encrypted basis.

22. The system of claim 20, wherein said one-time request-authentication datum comprises a function of an encryption key.

23. A system for enhancing trust in communicating a response from a request from a client device to a trusted server, comprising:

(a) means for establishing a network connection between the client device and the server; and

(b) means for conducting communications of data with the client device over the network connection,

wherein said communications between the device and the server are conducted in accordance with a method comprising:

(i) receiving a request comprising a function of at least a portion of at least one one-time password shared between the device and said server; and

(ii) generating at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

24. The system of claim 23, further comprising

(a) an encryption algorithm, and

(b) means for downloading said encryption algorithm to the client computer over said network connection,

wherein said means for conducting communications of data with the client computer over the network connection is in accordance with said encryption algorithm and wherein said communications between the device and the server are conducted on an encrypted basis.

25. The system of claim 23, wherein said one-time response-authentication datum comprises a function of an encryption key.

26. The system of claim 23, wherein said request comprises an encrypted secret datum, wherein said server decrypts said encrypted secret datum to recover said secret datum.

27. The system of claim 26, wherein a subsequent request comprises a function of at least a portion of at least one one-time password comprising at least a portion of at least one secret datum.

28. The method of claim 23, wherein the one-time password comprised within the request is used by the server to locate an entry in its database corresponding to the particular client device.

29. A system for resynchronizing communication between a client device and a trusted server, comprising:

(a) means for establishing a network connection between the client device and the server; and

(b) means for conducting communications of data with the client device over the network connection,

wherein said communications between the device and the server are conducted in accordance with a method comprising:

(i) supplying a one-time password for use in communication from the device to the server;

(ii) supplying at least one one-time request-authentication datum comprising a function of at least a portion of a previous response from the server to a previous request from the device; and

(iii) supplying at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

30. The system of claim 29, further comprising

(a) an encryption algorithm, and

(b) means for downloading said encryption algorithm to the client computer over said network connection,

wherein said means for conducting communications of data with the client computer over the network connection is in accordance with said encryption algorithm and wherein said communications between the device and the server are conducted on an encrypted basis.

31. A system for enhancing trust in transmission of a resynchronization request from a client device, comprising:

(a) means for establishing a network connection between the client device and a trusted server; and

(b) means for conducting communications of data with the client device over the network connection,

wherein said communications between the device and the server are conducted in accordance with a method comprising:

(i) supplying a one-time password; and

(ii) supplying at least one one-time request authentication datum comprising a function of at least a portion of a previous response from the server to a request from the device.

32. The system of claim 31, further comprising

(a) an encryption algorithm, and

(b) means for downloading said encryption algorithm to the client computer over said network connection,

wherein said means for conducting communications of data with the client computer over the network connection is in accordance with said encryption algorithm and wherein said communications between the device and the server are conducted on an encrypted basis.

33. The system of claim 31, wherein said resynchronization request comprises an encrypted resynchronization datum that replaces a previous resynchronization datum.

34. A system for enhancing trust in transmission of a resynchronization response from a trusted server, comprising:

(a) means for establishing a network connection between a client device and the server; and

(b) means for conducting communications of data with the client device over the network connection,

wherein said communications between the device and the server are conducted in accordance with a method comprising:

(i) receiving a request comprising a one-time password associated with a client device; and

(ii) supplying at least one one-time response-authentication datum comprising a function of at least a portion of at least one one-time password.

35. The system of claim 34, further comprising

(a) an encryption algorithm, and

(b) means for downloading said encryption algorithm to the client computer over said network connection,

wherein said means for conducting communications of data with the client computer over the network connection is in accordance with said encryption algorithm and wherein said communications between the device and the server are conducted on an encrypted basis.

36. The system of claim 34, wherein said resynchronization response comprises an encrypted resynchronization datum that replaces a previous resynchronization datum.

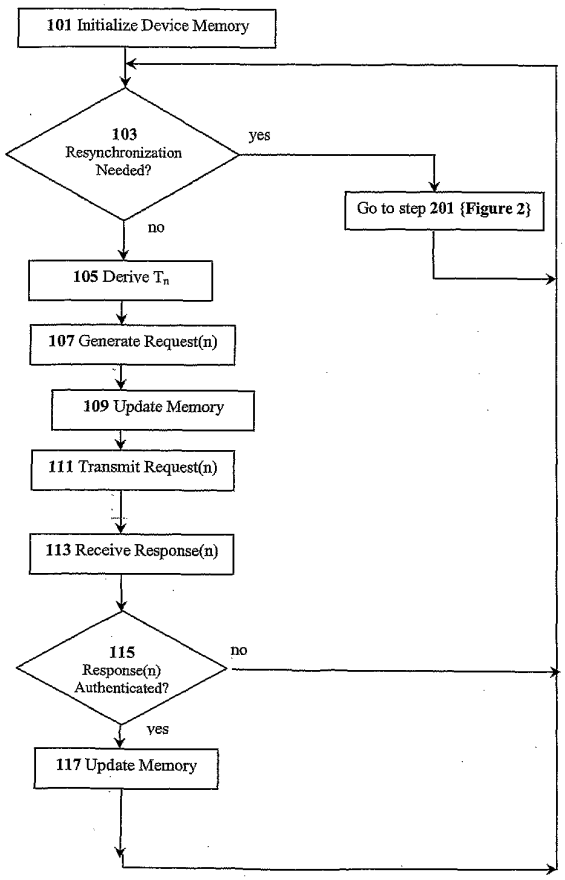

101 Initialize Device Memory
103 Resynchronization Needed?
yes
Go to step 201 {Figure 2}
no
105 Derive T_n
107 Generate Request(n)
109 Update Memory
111 Transmit Request(n)
113 Receive Response(n)
115 Response(n) Authenticated?
no
yes
117 Update Memory

FIGURE 1

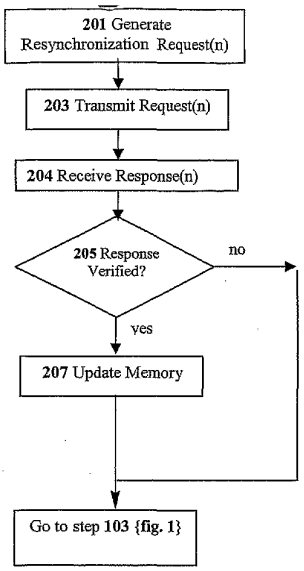

201 Generate Resynchronization Request(n)
203 Transmit Request(n)
204 Receive Response(n)
205 Response Verified?
no
yes
207 Update Memory
Go to step 103 {fig. 1}

FIGURE 2

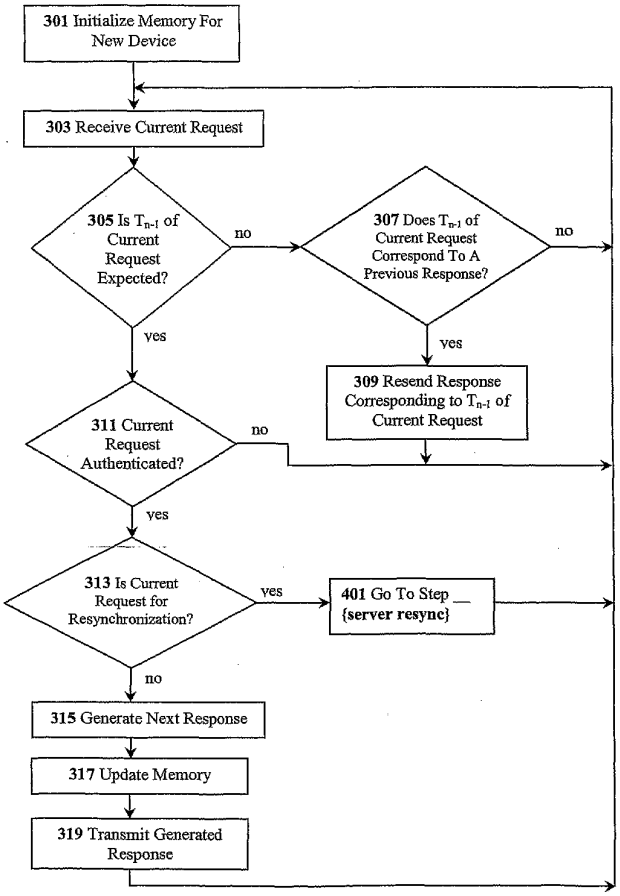

301 Initialize Memory For New Device
303 Receive Current Request
305 Is T_{n-1} of Current Request Expected?
no
307 Does T_{n-1} of Current Request Correspond To A Previous Response?
no
yes
yes
309 Resend Response Corresponding to T_{n-1} of Current Request
311 Current Request Authenticated?
no
yes
313 Is Current Request for Resynchronization?
yes
401 Go To Step __ {server resync}
no
315 Generate Next Response
317 Update Memory
319 Transmit Generated Response

FIGURE 3

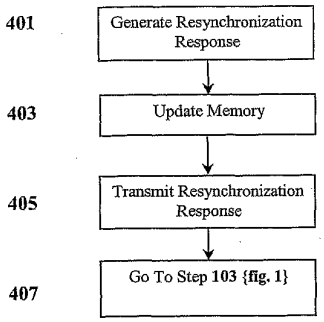

401
Generate Resynchronization Response
403
Update Memory
405
Transmit Resynchronization Response
407
Go To Step 103 {fig. 1}

FIGURE 4

【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 May 2002 (30.05.2002)

PCT

(10) International Publication Number
WO 02/043309 A3

(51) International Patent Classification⁷: H04L 9/32

(21) International Application Number: PCT/US01/46290

(22) International Filing Date: 19 October 2001 (19.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/242,083 20 October 2000 (20.10.2000) US
60/246,843 8 November 2000 (08.11.2000) US

(71) Applicant: WAVE SYSTEMS CORPORATION [US/US]; 480 Pleasant Street, Suite B200, Lee, MA 01238 (US).

(72) Inventor: KRAVITZ, David, W.; 3910 Ridgelea Drive, Fairfax, VA 22031 (US).

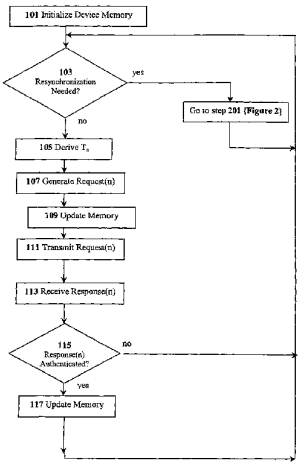
(74) Agents: BUTTER, Gary, M. et al.; Baker Botts, LLP, 30 Rockefeller Plaza, New York, NY 10112-0228 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: CRYPTOGRAPHIC DATA SECURITY SYSTEM AND METHOD



(57) Abstract: A method for communicating between a computer device and a trusted server is disclosed. According to the method of the invention, a one-time password for use in communication from the device (105) to the server is generated. The device (105) generates at least one on-time request-authentication datum (107) that includes a function of at least a portion of a previous response (113) from the server to previous message from the device (105). The server then generates at least one on-time response authentication datum (113) that includes a function of at least a portion of at least one-time password.

WO 02/043309 A3

WO 02/043309 A3

IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
6 February 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/46290

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : H04L 9/32
US CL : 713/202
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 380/21,23,25,37, 43, 49,50

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,841,871 B1 (PINKAS) 24 November 1998 (24.11.1998), abstract, column 4, lines 1-67, column 9, column 5, lines 9-20, column 9, lines 34-65, Fig. 1-2. Column 9, lines 13-34, column 12, line30-45	1-10 --- 11-36
Y, P	US 6,148,404 A (YATSUKAWA) 14 November 2000 (14.11.2000), abstract, column 15, lines 20-64, lines 65-67 through column 16, lines 1-2, lines 33-67, column 17, lines 1-35, lines 65-67 through column 18, lines 1-9, Fig.16, column 19, lines 46-67, column 23, lines 26-47, column 24, lines 38-65.	11-36
Y	US 5,661,807 A (GUSKI et al) 26 August 1997 (26.08.1997), the entire document.	1-36
Y	5,241,599 A (BELLOVIN et al.) 31 August 1993 (31.08.1993), the enitire document.	1-36

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
01 June 2002 (01.06.2002)

Date of mailing of the international search report
09 JUL 2002

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703)305-3230

Authorized officer
Gail O Hayes
Telephone No. (703) 305-4274

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/46290

Continuation of B. FIELDS SEARCHED Item 3:
WEST, Dialog, ProQuest, Dogpile. Search Terms: one time password and authentication, kerberos password and one time password, password and authentication, client/server session key.

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,PH,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TR,TT,TZ,UA,UG,UZ,VN,YU,ZA,ZW

(74)代理人 100107836
弁理士 西 和哉
(74)代理人 100108453
弁理士 村山 靖彦
(74)代理人 100110364
弁理士 実広 信哉
(72)発明者 デヴィッド・ダブリュ・クラヴィッツ
アメリカ合衆国・ヴァージニア・２２０３１・フェアファックス・リッジリー・ドライヴ・３９１０

Ｆターム(参考) 5B085 AE09 AE23 AE29 BC02 BE04 BG02 BG07
5J104 AA07 AA16 EA01 EA03 EA04 EA15 KA01 KA06 KA21 MA01
NA02 NA05 PA07