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

特許第5345675号
(P5345675)

(45)発行日　平成25年11月20日(2013.11.20)　(24)登録日　平成25年8月23日(2013.8.23)

(51)Int.Cl.　　F I
H04L 9/32 (2006.01)　　H04L 9/00 675A
G09C 1/00 (2006.01)　　G09C 1/00 640E

請求項の数 57　(全 40 頁)

(21)出願番号　特願2011-508643 (P2011-508643)
(86)(22)出願日　平成21年5月6日 (2009.5.6)
(65)公表番号　特表2011-521548 (P2011-521548A)
(43)公表日　平成23年7月21日 (2011.7.21)
(86)国際出願番号　PCT/US2009/043040
(87)国際公開番号　WO2009/137621
(87)国際公開日　平成21年11月12日 (2009.11.12)
審査請求日　平成23年1月11日 (2011.1.11)
(31)優先権主張番号　12/118,593
(32)優先日　平成20年5月9日 (2008.5.9)
(33)優先権主張国　米国 (US)

(73)特許権者　595020643
クゥアルコム・インコーポレイテッド
ＱＵＡＬＣＯＭＭ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100091351
弁理士　河野　哲
(74)代理人　100088683
弁理士　中村　誠
(74)代理人　100109830
弁理士　福原　淑弘

最終頁に続く

(54)【発明の名称】トークンとベリファイアとの間の認証のためのネットワーク・ヘルパー

(57)【特許請求の範囲】
【請求項１】
トークンを認証するためのベリファイア上で動作可能な方法であって、
トークンから認証要求及びトークン識別子を受信することと、
ネットワーク化されたヘルパーから前記トークンに対応する複数のパズルを取得することと、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くことと、
前記パズル秘密に基づいて検証鍵を生成することと、
前記検証鍵に基づいて質問メッセージを生成することと、
前記パズル識別子及び前記質問メッセージを前記トークンに送信することと、
を備える方法。
【請求項２】
前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信すること、
をさらに備える請求項１に記載の方法。
【請求項３】
前記検証鍵を記憶することと、

前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けることと、
をさらに備える請求項１に記載の方法。
【請求項４】
前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得することをさらに備える請求項１に記載の方法。
【請求項５】
パズルがパズル識別子とパズル秘密とを含む符号化メッセージである請求項１に記載の方法。
【請求項６】
前記受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するために前記受信したパズルのうちのサブセット複数を解くことであって、前記検証鍵が前記パズル秘密の順序セットにも基づく、解くことと、
パズル秘密の前記順序セットに対応する前記パズル識別子の順序セットを前記質問メッセージとともに前記トークンに送信することと、
をさらに備える請求項１に記載の方法。
【請求項７】
前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記トークンによって維持されるカウンタにローカル・カウンタを同期させることであって、事後認証質問メッセージが現在のカウンタ値にも基づく、同期させること、
をさらに備える請求項１に記載の方法。
【請求項８】
前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信すること、
をさらに備える請求項７に記載の方法。
【請求項９】
タイムスタンプを生成するためのタイマを維持することであって、事後認証質問メッセージが現在のタイムスタンプにも基づく、維持することをさらに備える請求項１に記載の方法。
【請求項１０】
前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することをさらに備える請求項９に記載の方法。
【請求項１１】
前記選択されたパズルを解くことが、前記パズルを復号するための鍵を発見するためにブルート・フォース・アタックを実行することを含む請求項１に記載の方法。
【請求項１２】
前記質問メッセージがメッセージ認証コードである請求項１に記載の方法。
【請求項１３】
擬似ランダム・ナンスを生成することと、
前記トークンに対応するより多数のパズルの中から前記複数のパズルを取得する際に使用すべき前記ヘルパーに前記擬似ランダム・ナンスを送信することと、
をさらに備える請求項１に記載の方法。
【請求項１４】
トークンを認証するための検証デバイスであって、

ネットワークに対する高帯域幅を有する第１の通信インターフェースと、
トークンと通信するための低帯域幅を有する第２の通信インターフェースと、
前記第１の通信インターフェース及び前記第２の通信インターフェースに結合された処理回路であって、
前記第２の通信インターフェースを介してトークンから認証要求及びトークン識別子を受信すること、
前記第１の通信インターフェースを介してヘルパーから前記トークンに対応する複数のパズルを取得すること、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くこと、
前記パズル秘密に基づいて検証鍵を生成すること、
前記検証鍵に基づいて質問メッセージを生成すること、ならびに
前記第２の通信インターフェースを介して前記パズル識別子及び前記質問メッセージを前記トークンに送信すること
を行うように構成された処理回路と、
を備える検証デバイス。
【請求項１５】
前記処理回路が、
前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記検証デバイスが前記トークンを認証する、受信することを行うようにさらに構成された請求項１４に記載の検証デバイス。
【請求項１６】
前記処理回路に結合された、前記検証鍵を記憶するための記憶デバイスをさらに備え、
前記処理回路が、前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けるようにさらに構成された請求項１４に記載の検証デバイス。
【請求項１７】
前記処理回路が、前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得するようにさらに構成された請求項１４に記載の検証デバイス。
【請求項１８】
前記処理回路が、
前記受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するために前記受信したパズルのうちのサブセット複数を解くことであって、前記検証鍵が前記パズル秘密の順序セットにも基づく、解くことと、
パズル秘密の前記順序セットに対応する前記パズル識別子の順序セットを前記質問メッセージとともに前記トークンに送信することと、
を行うようにさらに構成された請求項１４に記載の検証デバイス。
【請求項１９】
前記処理回路が、
前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記トークンによって維持されるカウンタにローカル・カウンタを同期させることであって、事後認証質問メッセージが現在のカウンタ値にも基づく、同期させることと、
前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のカウンタ値を独立して知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することと、
を行うようにさらに構成された請求項１４に記載の検証デバイス。
【請求項２０】

前記処理回路が、
タイムスタンプを生成するタイマを維持することであって、事後認証質問メッセージが現在のタイムスタンプにも基づく、維持することと、
前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のタイムスタンプを独立して知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することと、
を行うようにさらに構成された請求項１４に記載の検証デバイス。
【請求項２１】
トークンを認証するための検証デバイスであって、
トークンから認証要求及びトークン識別子を受信するための手段と、
ネットワーク化されたヘルパーから前記トークンに対応する複数のパズルを取得するための手段と、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くための手段と、
前記パズル秘密に基づいて検証鍵を生成するための手段と、
前記検証鍵に基づいて質問メッセージを生成するための手段と、
前記パズル識別子及び前記質問メッセージを前記トークンに送信するための手段と、
を備える検証デバイス。
【請求項２２】
前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信するための手段であって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信するための手段をさらに備える請求項２１に記載の検証デバイス。
【請求項２３】
前記検証鍵を記憶するための手段と、
前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けるための手段と、
をさらに備える請求項２１に記載の検証デバイス。
【請求項２４】
前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得するための手段
をさらに備える請求項２１に記載の検証デバイス。
【請求項２５】
前記受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するために前記受信したパズルのうちのサブセット複数を解くための手段であって、前記検証鍵が前記パズル秘密の順序セットにも基づく、解くための手段と、
パズル秘密の前記順序セットに対応する前記パズル識別子の順序セットを前記質問メッセージとともに前記トークンに送信するための手段と、
をさらに備える請求項２１に記載の検証デバイス。
【請求項２６】
前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記トークンによって維持されるカウンタにローカル・カウンタを同期させるための手段であって、事後認証質問メッセージが現在のカウンタ値にも基づく、同期させるための手段と、
前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを証明する応答メッセージを前記トークンから受信するための手段であって、前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信するための手段と、
をさらに備える請求項２１に記載の検証デバイス。

【請求項２７】
タイムスタンプを生成するタイマを維持するための手段であって、事後認証質問メッセージが現在のタイムスタンプにも基づく、維持するための手段と、
前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを証明する応答メッセージを前記トークンから受信するための手段であって、前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信するための手段と、
をさらに備える請求項２１に記載の検証デバイス。
【請求項２８】
第１の通信インターフェースを介してトークンから認証要求及びトークン識別子を受信することと、
第２の通信インターフェースを介してヘルパーから前記トークンに対応する複数のパズルを取得することと、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くことと、
前記パズル秘密に基づいて検証鍵を生成することと、
前記検証鍵に基づいて質問メッセージを生成することと、
前記第１の通信インターフェースを介して前記パズル識別子及び前記質問メッセージを前記トークンに送信することと、
を行うように構成された処理回路
を備える処理デバイス。
【請求項２９】
前記処理回路に結合された、前記検証鍵を記憶するための記憶デバイス
をさらに備え、
前記処理回路が、前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けるようにさらに構成された請求項２８に記載の処理デバイス。
【請求項３０】
前記処理回路が、前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得するようにさらに構成された請求項２８に記載の処理デバイス。
【請求項３１】
ベリファイアに対してトークンを認証するための１つまたは複数の命令であって、プロセッサによって実行されたとき、前記プロセッサに、
ネットワーク化されたヘルパーから前記トークンに対応する複数のパズルを取得することと、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くことと、
前記パズル秘密に基づいて検証鍵を生成することと、
前記検証鍵に基づいて質問メッセージを生成することと、
前記パズル識別子及び前記質問メッセージを前記トークンに送信することと、
を行わせる１つまたは複数の命令を有するプログラムを格納するコンピュータ可読記録媒体。
【請求項３２】
プロセッサによって実行されたとき、前記プロセッサに、
前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することをさらに行わせる１つまたは複数の命令を有する請求項３１に記載のコンピュータ可読記録媒体。
【請求項３３】

プロセッサによって実行されたとき、前記プロセッサに、
前記検証鍵を記憶することと、
前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けることと、
をさらに行わせる１つまたは複数の命令を有する請求項３１に記載のコンピュータ可読記録媒体。
【請求項３４】
ベリファイアに対してトークンを認証するための前記トークン上で動作可能な方法であって、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングすることと、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始することと、
第１の検証鍵に基づく１つまたは複数のパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記１つまたは複数のパズル識別子によって識別される１つまたは複数のパズルに関連する１つまたは複数のパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信した１つまたは複数のパズル識別子、及び前記秘密鍵に基づいて前記１つまたは複数のパズル秘密を独立して取得することと、
前記１つまたは複数のパズル秘密に基づいて第２の検証鍵を生成することと、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証することと、
を備える方法。
【請求項３５】
前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けることをさらに備える請求項３４に記載の方法。
【請求項３６】
前記第２の検証鍵に基づいて、前記トークンが前記第１の検証鍵を知っていることを示す、前記ベリファイアへの応答メッセージを生成することをさらに備える請求項３４に記載の方法。
【請求項３７】
前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給することをさらに備える請求項３４に記載の方法。
【請求項３８】
前記ベリファイアから複数の順序付きパズル識別子を受信することであって、前記第１の検証鍵が、前記複数の順序付きパズル識別子に関連する対応する複数の順序付きパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記複数の順序付きパズル秘密を取得することであって、前記第１の検証鍵及び前記第２の検証鍵が前記複数の順序付きパズル秘密にも基づく、取得することと、
をさらに備える請求項３４に記載の方法。
【請求項３９】
前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記ベリファイアによって維持されるカウンタにローカル・カウンタを同期させることであって、後続の受信したメッセージ認証コードが現在のカウンタ値にも基づく、同期させること、

をさらに備える請求項３４に記載の方法。
【請求項４０】
前記トークンが前記検証鍵と前記現在のカウンタ値とを知っていることを証明する応答メッセージを前記ベリファイアに送信すること、
をさらに備える請求項３９に記載の方法。
【請求項４１】
タイムスタンプを生成するためのタイマを維持することであって、後続の受信したメッセージ認証コードが現在のタイムスタンプにも基づく、維持すること、
をさらに備える請求項３４に記載の方法。
【請求項４２】
前記トークンが前記検証鍵と前記現在のタイムスタンプとを知っていることを証明する応答メッセージを前記ベリファイアに送信することをさらに備える請求項４１に記載の方法。
【請求項４３】
異なるベリファイアを用いて複数のセキュアな検証鍵を確立することと、
前記検証鍵を記憶し、前記トークンと前記異なるベリファイアとの間の対称鍵認証として使用するために前記検証鍵の各々を対応するベリファイアに関連付けることと、
をさらに備える請求項３４に記載の方法。
【請求項４４】
ベリファイアと通信するための低帯域幅を有する第１の通信インターフェースと、
前記第１の通信インターフェースに結合された処理回路であって、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムを受信すること、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始すること、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信すること、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得すること、
前記パズル秘密に基づいて第２の検証鍵を生成すること、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証すること、
を行うように構成された処理回路と、
を備えるトークン。
【請求項４５】
前記処理回路に結合され、前記第２の検証鍵を記憶するように構成された記憶デバイスであって、前記処理回路が、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けるように構成された、記憶デバイス
をさらに備える請求項４４に記載のトークン。
【請求項４６】
前記処理回路が、
異なるベリファイアを用いて複数のセキュアな検証鍵を確立することと、
前記検証鍵を記憶し、前記トークンと前記異なるベリファイアとの間の対称鍵認証として使用するために前記検証鍵の各々を対応するベリファイアに関連付けることと
を行うようにさらに構成された請求項４４に記載のトークン。
【請求項４７】
前記処理回路が、
前記第２の検証鍵に基づいて、前記トークンが前記第１の検証鍵を知っていることを示す、前記ベリファイアへの応答メッセージを生成すること

を行うようにさらに構成された請求項４４に記載のトークン。
【請求項４８】
前記処理回路が、
前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給すること
を行うようにさらに構成された請求項４４に記載のトークン。
【請求項４９】
ベリファイアで認証するためのトークンであって、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングするための手段と、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始するための手段と、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信するための手段であって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信するための手段と、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得するための手段と、
前記パズル秘密に基づいて第２の検証鍵を生成するための手段と、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証するための手段と、
を備えるトークン。
【請求項５０】
前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けるための手段をさらに備える請求項４９に記載のトークン。
【請求項５１】
前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給するための手段をさらに備える請求項４９に記載のトークン。
【請求項５２】
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムを受信することと、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始することと、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得することと、
前記パズル秘密に基づいて第２の検証鍵を生成することと、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証することと、
を行うように構成された処理回路
を備える処理デバイス。
【請求項５３】
前記処理回路が、
前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けること
を行うようにさらに構成された請求項５２に記載の処理デバイス。

【請求項５４】
前記処理回路が、
前記第２の検証鍵に基づいて、前記トークンが前記第１の検証鍵を知っていることを示す、前記ベリファイアへの応答メッセージを生成することを行うようにさらに構成された請求項５２に記載の処理デバイス。
【請求項５５】
ベリファイアに対してトークンを認証するための１つまたは複数の命令であって、プロセッサによって実行されたとき、前記プロセッサに、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングすることと、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始することと、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得することと、
前記パズル秘密に基づいて第２の検証鍵を生成することと、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証することと、
を行わせる１つまたは複数の命令を有するプログラムを格納するコンピュータ可読記録媒体。
【請求項５６】
プロセッサによって実行されたとき、前記プロセッサに、
前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けることをさらに行わせる１つまたは複数の命令を有する請求項５５に記載のコンピュータ可読記録媒体。
【請求項５７】
プロセッサによって実行されたとき、前記プロセッサに、
前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給することをさらに行わせる１つまたは複数の命令を有する請求項５５に記載のコンピュータ可読記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、セキュアな認証に関し、より詳細には、限られたリソースのデバイスの認証に関する。
【背景技術】
【０００２】
電子認証は、一般に、それぞれ公開鍵（非対称鍵）暗号方法または対称鍵暗号方法に基づく何らかの形式のデジタル署名またはメッセージ認証コードを必要とする。公開鍵方法と対称鍵方法は異なる特徴を有するので、異なる状況において適用可能である。概して、公開鍵システムは、鍵の管理及び配信に関して大きい利点を有するが、計算量的に困難であり、生成されるデジタル署名が大きくなる傾向がある。対称鍵方法は、極めて効率的であり、小出力を生成するが、同時に複数の検証パーティとともにセキュアに使用できないという欠点を有する。
【０００３】
たいていの小型の認証トークンは、限られた計算リソースと小出力の必要性の両方により、対称鍵暗号法を使用せざるを得なかった。これは、トークンが、ただ１つの検証パーティとの認証にしか使用できないことを意味する。たとえば、２つの異なるエンティティ

が異なる対称鍵を用いて互換性のあるトークンを発行することはできるが、それらを１つのトークンに結合することは、対称鍵を共有することを必要とし、パーティ同士が互いに信用できないので、不可能である。米国の銀行が２ファクタ認証方法を実装するという最近の発表に伴って、ユーザは、どこへ行く場合にも複数の認証トークンを持ち歩かなければならない。

【０００４】

公開鍵システムに基づく認証方法が提案されているが、１つの欠点は、多くのトークンで利用可能な計算リソースよりも大きい計算リソースが要求されることである。即ち、トークンは、廉価になり、及び／または限られた処理リソースを有する傾向がある。そのようなトークンにより強力なプロセッサを追加すれば、コストが増大するだけでなく、バッテリ寿命がより短くなる。公開鍵システムを使用することの別の欠点は、デジタル署名のサイズが長くなると、数字の短配列のタイピングまたは読出しと比較して、実装が煩雑になることである。

【０００５】

別のタイプの認証システムは、認証を仲介するための信頼できるサード・パーティを使用する。そのようなシステムの一例はＭＩＴのケルベロス（Kerberos）である。しかしながら、信頼できるサード・パーティへの依存は、セキュリティが損なわれ得る別のポイントを追加するので、銀行などの多くの機関にとってディール・ブレイカーとなる。

【０００６】

したがって、単一のトークンが複数のベリファイアに対して使用できるように、公開鍵システムの利点と対称鍵システムの利点を組み合わせて、小型で効率的なデジタル署名を提供する方法が必要である。

【発明の概要】

【０００７】

トークンを認証するためのベリファイア上で動作可能な方法を提供する。トークンから認証要求及びトークン識別子を受信する。トークンに関連するヘルパーのヘルパー・アドレスを取得する。ネットワーク化されたヘルパーからトークンに対応する複数のパズルを取得する。パズルは、パズル識別子とパズル秘密とを含む符号化メッセージとすることができる。パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するためにそのパズルのうちの１つを解く。たとえば、選択されたパズルは、パズルを復号するための鍵を発見するために、ブルート・フォース・アタック(brute force attack)を実行することによって解くことができる。パズル秘密に基づいて検証鍵を生成する。次いで、検証鍵に基づいて質問メッセージを生成する。パズル識別子及び質問メッセージをトークンに送信する。トークンが検証鍵を知っていることを証明する応答メッセージをトークンから受信し、トークンが検証鍵を知っていることを応答メッセージが首尾よく証明するならば、ベリファイアがトークンを認証する。検証鍵を記憶し、トークンとベリファイアとの間の事後認証における対称鍵として使用するために検証鍵をトークン識別子に関連付ける。

【０００８】

一例では、受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するためにその受信したパズルのうちのサブセット複数を解く。検証鍵はパズル秘密の順序セットにも基づく。パズル秘密の順序セットに対応するパズル識別子の順序セットを質問メッセージとともにトークンに送信する。

【０００９】

別の特徴によれば、ベリファイアは、擬似ランダム・ナンス(pseudorandom nonce)を生成し、トークンに対応するより多数のパズルの中から複数のパズルを取得する際に使用すべきヘルパーにその擬似ランダム・ナンスを送信する。

【００１０】

一実装形態では、トークンとベリファイアとの間の認証が実行されるたびにトラッキングするためにトークンによって維持されるカウンタにローカル・カウンタを同期させ、事

後認証質問メッセージは現在のカウンタ値にも基づく。したがって、トークンが検証鍵と現在のカウンタ値とを知っていることを証明する応答メッセージをトークンから受信する。トークンが検証鍵及び現在のカウンタ値を知っていることを応答メッセージが首尾よく証明するならば、ベリファイアはトークンを認証する。

【００１１】

別の実装形態では、タイムスタンプを生成するタイマを維持し、事後認証質問メッセージは現在のタイムスタンプにも基づく。トークンが検証鍵と現在のタイムスタンプとを知っていることを証明する応答メッセージをトークンから受信する。トークンが検証鍵及び現在のタイムスタンプを知っていることを応答メッセージが首尾よく証明するならば、ベリファイアはトークンを認証する。

【００１２】

トークンを認証するための検証デバイスは、処理回路に結合された第１の通信インターフェースと第２の通信インターフェースとを備える。第１の通信インターフェースは、ネットワークに対する高帯域幅を有する。第２の通信インターフェースは、トークンと通信するための低帯域幅を有する。処理回路は、（ａ）第２の通信インターフェースを介してトークンから認証要求及びトークン識別子を受信すること、（ｂ）トークンに関連するヘルパーのヘルパー・アドレスを取得すること、（ｃ）第１の通信インターフェースを介してヘルパーからトークンに対応する複数のパズルを取得すること、（ｄ）パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するためにそのパズルのうちの１つを解くこと、（ｅ）パズル秘密に基づいて検証鍵を生成すること、（ｆ）検証鍵に基づいて質問メッセージを生成すること、（ｇ）第２の通信インターフェースを介してパズル識別子及び質問メッセージをトークンに送信すること、及び／または（ｈ）トークンが検証鍵を知っていることを証明する応答メッセージをトークンから受信することを行うように構成される。トークンが検証鍵を知っていることを応答メッセージが首尾よく証明するならば、検証デバイスはトークンを認証する。

【００１３】

検証デバイスはまた、処理回路に結合された、検証鍵を記憶するための記憶デバイスを含むことができる。処理回路は、さらに、トークンとベリファイアとの間の事後認証における対称鍵として使用するために検証鍵をトークン識別子に関連付けるように構成できる。

【００１４】

いくつかの実装形態では、処理回路は、さらに、（ａ）受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するためにその受信したパズルのうちのサブセット複数を解くことと（検証鍵はパズル秘密の順序セットにも基づく）、（ｂ）パズル秘密の順序セットに対応するパズル識別子の順序セットを質問メッセージとともにトークンに送信することとを行うように構成できる。

【００１５】

一実装形態では、トークンとベリファイアとの間の認証が実行されるたびにトラッキングするためにトークンによって維持されるカウンタにローカル・カウンタを同期させる。事後認証質問メッセージは現在のカウンタ値にも基づく。したがって、トークンが検証鍵と現在のカウンタ値とを知っていることを証明する応答メッセージをトークンから受信する。トークンが検証鍵及び現在のカウンタ値を独立して知っていることを応答メッセージが首尾よく証明するならば、ベリファイアはトークンを認証する。

【００１６】

別の実装形態では、タイムスタンプを生成するタイマを維持し／事後認証質問メッセージは現在のタイムスタンプにも基づく。したがって、トークンが検証鍵と現在のタイムスタンプとを知っていることを証明する応答メッセージをトークンから受信する。トークンが検証鍵及び現在のタイムスタンプを独立して知っていることを応答メッセージが首尾よく証明するならば、ベリファイアはトークンを認証する。

【００１７】

したがって、トークンを認証するための検証デバイスであって、（ａ）トークンから認証要求及びトークン識別子を受信するための手段、（ｂ）トークンに関連するヘルパーのヘルパー・アドレスを取得するための手段、（ｃ）ネットワーク化されたヘルパーからトークンに対応する複数のパズルを取得するための手段、（ｄ）パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するためにそのパズルのうちの１つを解くための手段、（ｅ）パズル秘密に基づいて検証鍵を生成するための手段、（ｆ）検証鍵に基づいて質問メッセージを生成するための手段、（ｇ）パズル識別子及び質問メッセージをトークンに送信するための手段、及び／または（ｈ）トークンが検証鍵を知っていることを証明する応答メッセージをトークンから受信するための手段であって、トークンが検証鍵を知っていることを応答メッセージが首尾よく証明するならば、ベリファイアがトークンを認証する、受信するための手段を備える、検証デバイスを提供する。さらに、検証デバイスは、（ｉ）検証鍵を記憶するための手段、（ｊ）トークンとベリファイアとの間の事後認証における対称鍵として使用するために検証鍵をトークン識別子に関連付けるための手段、（ｋ）受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するためにその受信したパズルのうちのサブセット複数を解くための手段（検証鍵はパズル秘密の順序セットにも基づく）、及び／または（ｌ）パズル秘密の順序セットに対応するパズル識別子の順序セットを質問メッセージとともにトークンに送信するための手段をも含むことができる。

【００１８】

また、（ａ）第１の通信インターフェースを介してトークンから認証要求及びトークン識別子を受信すること、（ｂ）第２の通信インターフェースを介してヘルパーからトークンに対応する複数のパズルを取得すること、（ｃ）パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するためにそのパズルのうちの１つを解くこと、（ｄ）パズル秘密に基づいて検証鍵を生成すること、（ｅ）検証鍵に基づいて質問メッセージを生成すること、（ｆ）第１の通信インターフェースを介してパズル識別子及び質問メッセージをトークンに送信すること、及び／または（ｇ）トークンとベリファイアとの間の事後認証における対称鍵として使用するために検証鍵をトークン識別子に関連付けることを行うように構成された処理回路を備える処理デバイスを提供する。

【００１９】

また、ベリファイアに対してトークンを認証するための１つまたは複数の命令であって、プロセッサによって実行されたとき、プロセッサに、（ａ）ネットワーク化されたヘルパーからトークンに対応する複数のパズルを取得すること、（ｂ）パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するためにそのパズルのうちの１つを解くこと、（ｃ）パズル秘密に基づいて検証鍵を生成すること、（ｄ）検証鍵に基づいて質問メッセージを生成すること、（ｅ）パズル識別子及び質問メッセージをトークンに送信すること、（ｆ）トークンが検証鍵を知っていることを証明する応答メッセージをトークンから受信すること（トークンが検証鍵を知っていることを応答メッセージが首尾よく証明するならば、ベリファイアはトークンを認証する）、及び／または（ｇ）トークンとベリファイアとの間の事後認証における対称鍵として使用するために検証鍵をトークン識別子に関連付けることを行わせる１つまたは複数の命令を有する機械可読媒体を提供する。

【００２０】

また、ベリファイアに対してトークンを認証するためのトークン上で動作可能な方法を提供する。秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングする。トークン識別子をベリファイアに供給することによってトークンの認証を開始する。第１の検証鍵に基づく１つまたは複数のパズル識別子とメッセージ認証コードとを含む質問メッセージをベリファイアから受信する。第１の検証鍵は、１つまたは複数のパズル識別子によって識別される１つまたは複数のパズルに関連する１つまたは複数のパズル秘密の関数とすることができる。トークンは、１つまたは複数のパズル生成アルゴリズム、受信した１つまたは複数のパズル識別子、及び秘密鍵に基づい

て１つまたは複数のパズル秘密を独立して取得する。１つまたは複数のパズル秘密に基づいて第２の検証鍵をトークンによって生成する。トークンは、第１の検証鍵と第２の検証鍵が同じであるかどうか判断するために受信したメッセージ認証コードを検証する。第２の検証鍵を記憶し、第１の検証鍵と第２の検証鍵が同じである場合、第２の検証鍵をベリファイアに関連付ける。トークンは、第２の検証鍵に基づいて、トークンが第１の検証鍵を知っていることを示す、ベリファイアへの応答メッセージを生成する。トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスをトークンがベリファイアに供給する。トークンは、ベリファイアから複数の順序付きパズル識別子を受信し、第１の検証鍵は、複数の順序付きパズル識別子に関連する対応する複数の順序付きパズル秘密の関数である。１つまたは複数のパズル生成アルゴリズム、受信したパズル識別子、及び秘密鍵に基づいて複数の順序付きパズル秘密を取得する。第１の検証鍵及び第２の検証鍵は複数の順序付きパズル秘密にも基づく。このようにして、異なるベリファイアを用いて複数のセキュアな検証鍵を確立することができる。トークンと異なるベリファイアとの間の対称鍵認証として使用するために検証鍵の各々を対応するベリファイアに関連付ける。

【００２１】

一実装形態では、トークンとベリファイアとの間の認証が実行されるたびにトラッキングするためにベリファイアによって維持されるカウンタにローカル・カウンタを同期させる。後続の受信したメッセージ認証コードは現在のカウンタ値にも基づく。トークンが検証鍵と現在のカウンタ値とを知っていることを証明する応答メッセージをベリファイアに送信する。

【００２２】

別の実装形態では、タイムスタンプを生成するためのタイマを維持する。したがって、後続の受信したメッセージ認証コードは現在のタイムスタンプにも基づく。トークンが検証鍵と現在のタイムスタンプとを知っていることを証明する応答メッセージをベリファイアに送信する。

【００２３】

第１の通信インターフェースと処理回路とを備えるトークンを提供する。第１の通信インターフェースは、ベリファイアと通信するための低帯域幅を有する。処理回路は、（ａ）秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムを受信すること、（ｂ）トークン識別子をベリファイアに供給することによってトークンの認証を開始すること、（ｃ）第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージをベリファイアから受信すること（第１の検証鍵は、パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である）、（ｄ）１つまたは複数のパズル生成アルゴリズム、受信したパズル識別子、及び秘密鍵に基づいて第１のパズル秘密を独立して取得すること、（ｅ）パズル秘密に基づいて第２の検証鍵を生成すること、及び／または（ｆ）第１の検証鍵と第２の検証鍵が同じであるかどうか判断するために受信したメッセージ認証コードを検証することを行うように構成される。処理回路は、第１の検証鍵と第２の検証鍵が同じである場合、第２の検証鍵をベリファイアに関連付ける。第２の検証鍵に基づいて、トークンが第１の検証鍵を知っていることを示す、ベリファイアへの応答メッセージを生成する。

【００２４】

したがって、ベリファイアで認証するためのトークンであって、（ａ）秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングするための手段、（ｂ）トークン識別子をベリファイアに供給することによってトークンの認証を開始するための手段、（ｃ）第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージをベリファイアから受信するための手段（第１の検証鍵は、パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である）、（ｄ）１つまたは複数のパズル生成アルゴリズム、受信したパズル識別子、及び秘密鍵に基づいて第１のパズル秘密を独立して取得するための手段、（ｅ）パズル秘密に基づい

て第２の検証鍵を生成するための手段、（ｆ）第１の検証鍵と第２の検証鍵が同じであるかどうか判断するために受信したメッセージ認証コードを検証するための手段、（ｇ）第２の検証鍵を記憶し、第１の検証鍵と第２の検証鍵が同じである場合、第２の検証鍵をベリファイアに関連付けるための手段、及び／または（ｈ）トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスをトークンからベリファイアに供給するための手段を備えるトークンを提供する。

【００２５】

また、（ａ）秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムを受信すること、（ｂ）トークン識別子をベリファイアに供給することによってトークンの認証を開始すること、（ｃ）第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージをベリファイアから受信すること（第１の検証鍵は、パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である）、（ｄ）１つまたは複数のパズル生成アルゴリズム、受信したパズル識別子、及び秘密鍵に基づいて第１のパズル秘密を独立して取得すること、（ｅ）パズル秘密に基づいて第２の検証鍵を生成すること、（ｆ）第１の検証鍵と第２の検証鍵が同じであるかどうか判断するために受信したメッセージ認証コードを検証すること、及び／または（ｇ）第２の検証鍵に基づいて、トークンが第１の検証鍵を知っていることを示す、ベリファイアへの応答メッセージを生成することを行うように構成された処理回路を備える処理デバイスを提供する。

【００２６】

ベリファイアに対してトークンを認証するための１つまたは複数の命令であって、プロセッサによって実行されたとき、プロセッサに、（ａ）秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングすること、（ｂ）トークン識別子をベリファイアに供給することによってトークンの認証を開始すること、（ｃ）第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージをベリファイアから受信すること（第１の検証鍵は、パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である）、（ｄ）１つまたは複数のパズル生成アルゴリズム、受信したパズル識別子、及び秘密鍵に基づいて第１のパズル秘密を独立して取得すること、（ｅ）パズル秘密に基づいて第２の検証鍵を生成すること、（ｆ）第１の検証鍵と第２の検証鍵が同じであるかどうか判断するために受信したメッセージ認証コードを検証すること、及び／または（ｇ）第２の検証鍵を記憶し、第１の検証鍵と第２の検証鍵が同じである場合、第２の検証鍵をベリファイアに関連付けることを行わせる１つまたは複数の命令を有する機械可読媒体を提供する。

【図面の簡単な説明】

【００２７】

【図１】一例における、どのようにトークンに秘密鍵をプロビジョニングし、どのようにそのトークンに関連するパズルを生成するかを示すブロック図。

【図２】一例による、秘密鍵を生成し、トークンに配信し、その秘密鍵に関連する複数のパズルを生成するようにプロビジョニング・デバイス上で動作可能な方法を示す図。

【図３】トークン、検証デバイス、及びヘルパー・デバイスがトークンの初期認証中にどのように動作するかを示す図。

【図４】ヘルパーの支援を受けたトークンとベリファイアとの間の初期認証のための方法を示す図。

【図５】トークンとベリファイアとの間の事後認証を実行するための方法を示す図。

【図６】カウンタを使用したトークンとベリファイアとの間の認証を実行するための方法を示す図。

【図７】タイマを使用したトークンとベリファイアとの間の認証を実行するための方法を示す図。

【図８】複数の異なるベリファイアで認証するためにトークンが複数の検証鍵をどのように記憶するかを示すブロック図。

【図９】複数の異なるベリファイアで対称鍵を使用してセキュアな認証を実行するように

トークン上で動作可能な方法を示す図。
【図１０】トークンを認証するためにヘルパーの支援を受けて対称鍵を確立するように構成されたベリファイアを示すブロック図。
【図１１】ヘルパーの支援を受けてトークンのセキュアな認証を実行するベリファイア上で動作可能な方法を示す図。
【図１２】トークンを認証するための対称鍵を確立する際にベリファイアを支援するように構成されたヘルパーを示すブロック図。
【図１３】トークンを認証する際にベリファイアを支援するようにヘルパー上で動作可能な方法を示す図。
【発明を実施するための形態】
【００２８】
以下の説明では、実施形態の十分な理解が得られるように具体的な詳細を与える。ただし、これらの実施形態は、これらの具体的な詳細なしに実施できることを、当業者なら理解されよう。たとえば、これらの実施形態を不必要な詳細において不明瞭にしないために、回路をブロック図で示すか、またはまったく示さないことがある。他の場合では、これらの実施形態を不明瞭にしないために、よく知られている回路、構造、及び技法を詳細に図示しないことがある。
【００２９】
また、実施形態を、フローチャート、流れ図、構造図、またはブロック図として示されるプロセスとして説明することがあることに留意されたい。フローチャートは動作を逐次プロセスとして説明することがあるが、動作の多くは並行してまたは同時に実行できる。さらに、動作の順序を並び替えることができる。プロセスは、その動作が完了すると終了する。プロセスは、メソッド、関数、プロシージャ、サブルーチン、サブプログラムなどに対応することができる。プロセスが関数に対応する場合、その終了は呼出し側関数またはメイン関数への関数の復帰に対応する。
【００３０】
さらに、記憶媒体は、読取り専用メモリ（ＲＯＭ）、ランダム・アクセス・メモリ（ＲＡＭ）、磁気ディスク記憶媒体、光記憶媒体、フラッシュ・メモリデバイス、及び／または情報を記憶するための他のコンピュータ可読媒体を含む、データを記憶するための１つまたは複数のデバイスを表すことができる。「機械可読媒体」という用語は、携帯または固定記憶デバイス、光記憶デバイスならびに（１つまたは複数の）命令及び／またはデータを記憶することが可能な様々な他の媒体を含むが、これらに限定されない。
【００３１】
さらに、実施形態は、ハードウェア、ソフトウェア、ファームウェア、ミドルウェア、マイクロコード、またはその組合せによって実装できる。ソフトウェア、ファームウェア、ミドルウェア、またはマイクロコードで実装される場合、必要なタスクを実行するためのプログラムコードまたはコードセグメントを記憶媒体あるいは他の記憶手段などの機械可読媒体に記憶することができる。プロセッサは必要なタスクを実行することができる。コード・セグメントは、プロシージャ、関数、サブプログラム、プログラム、ルーチン、サブルーチン、モジュール、ソフトウェアパッケージ、クラス、または命令、データ構造もしくはプログラムステートメントの組合せを表すことができる。コードセグメントは、情報、データ、引数、パラメータ、またはメモリ内容をパス及び／または受信することによって、別のコードセグメントまたはハードウェア回路に結合できる。情報、引数、パラメータ、データなどは、特にメモリ共有、メッセージパッシング、トークンパッシング、及びネットワーク送信などを含む適切な手段を介してパス、転送、または送信できる。
【００３２】
「トークン(token)」という用語は、認証を助けるために許可されたユーザに関連付けられた、限定された処理リソース及び／または通信リソースをもつ物理デバイスを指す。「ベリファイア(verifier)」という用語は、トークンの認証を実行するデバイス、エンティティ、及び／または仮想オブジェクト（たとえば、ソフトウェア・アプリケーションな

ど）を指す。「ヘルパー(helper)」という用語は、トークンを認証する際にベリファイアを支援するデバイス、エンティティ、及び／または仮想オブジェクト（たとえば、ソフトウェア・アプリケーションなど）を指す。

【００３３】

一態様は、トークンとベリファイアとの間の認証のための対称鍵をセットアップすることを可能にするネットワーク化されたヘルパーを提供する。ベリファイアは、対称鍵を取得する際にベリファイアを支援するためにヘルパーを利用するが、ヘルパーはそれ自体で対称鍵を取得することができない。本方法は、プロビジョニング、初期認証、及び事後認証の３つの段階を含むことができる。

【００３４】

別の態様は、トークン及びベリファイアが、トークンとベリファイアとの間の認証のためのセキュアな対称鍵に関して同意できるようにするパズルベースのプロトコルを提供する。

【００３５】

プロビジョニング段階(provisioning stage)では、低帯域幅インターフェースを有する小型、低出力、及びポータブルなトークンに、複数の異なるベリファイアとともに関連するユーザを認証するために使用できる秘密鍵が与えられる。プロビジョニング段階は、トークンに割り当てられた秘密鍵に関連する複数のパズルを生成することと、それらのパズルをサード・パーティ・ヘルパーに配信することとを含むこともできる。

【００３６】

初期認証段階では、トークンとベリファイアが共有された対称鍵をネゴシエートできるようにすることによって、トークンがベリファイアに紹介される。ベリファイアは、そのトークンに関連するランダムな複数のパズルをヘルパーから取得するために使用できるトークン識別子を受信する。次いでベリファイアは、ブルート・フォース(brute force)によってパズルの少なくとも１つを解くかまたは解読して、関連するパズル秘密及びパズル識別子を取得する。ベリファイアは、解かれたパズルから取得されたパズル秘密及び他の情報に少なくとも部分的に基づいて、検証鍵を生成する。ベリファイアはパズル識別子と検証鍵のメッセージ認証コードとをトークンに供給する。トークンは、受信したパズル識別子を使用して、関連するパズル秘密を取り出し、検証鍵のローカルバージョンを生成する。次いでトークンは、（パズル識別子及びローカル検証鍵に少なくとも部分的に基づいて）メッセージ認証コードのローカルバージョンを生成し、そのローカルバージョンが受信したメッセージ認証コードに一致するかどうかを判断することができる。ローカルコードと受信したメッセージ認証コードとが一致する場合、その検証鍵をトークンとベリファイアとの間の対称鍵として使用することができる。

【００３７】

その後、事後認証段階において、再認証するために、より短い完全対称鍵ベースのプロセスを使用することができる。たとえば、（初期認証段階中に取得された）以前に生成された検証鍵は、認証のためにトークン及びベリファイアによって使用できる。このようにして、トークン、ベリファイア、及びヘルパーは、対称鍵方法と公開鍵方法とを組み合わせ、ベリファイアとヘルパーとの間で公開鍵デジタル署名の実装を分割する。ヘルパーは、信頼できるサード・パーティの機能の一部を実行するが、秘密対称鍵を知らず、ヘルパーが脅かされても、認証のスプーフィングは不可能である。この認証システムは完全公開鍵ベースのシステムの機能の大部分を提供するが、実際のトークンはプロビジョニング後に少量の処理リソース及び帯域幅のみを使用する。

【００３８】

鍵のプロビジョニング及びパズル・データベースの生成

図１は、一例における、どのようにトークンに秘密鍵をプロビジョニングし、どのようにそのトークンに関連するパズルを生成するかを示すブロック図である。プロビジョニング・デバイス１０２は、トークンを構成するように製造業者または配信者によって動作させることができる。プロビジョニング・デバイス１０２は、秘密鍵を生成する擬似ランダ

ム鍵生成器１０６を含む。トークン１１０の製造、セットアップ、または初期化中に、プロビジョニング・デバイス１０２は秘密鍵１０８を生成し、トークン１１０に配信することができ、トークン１１０において、秘密鍵１０８は鍵記憶デバイス１１２（たとえば、メモリ）に記憶される。トークン１１０はまた、その秘密鍵を使用してパズルを再生するために使用できる１つまたは複数のパズル生成アルゴリズムを記憶することができる。さらに、秘密鍵１０８は、パズル生成器１１４をシードするために使用される。パズル生成器１１４は、秘密鍵１０８を使用して、複数のパズル（たとえば、メッセージ）を生成し、そのようなパズルを（サード・パーティ・ヘルパーなどの）パズル・データベース１１６に供給するように構成される。一例では、各パズルはパズル秘密及び一意のパズル識別子を含むことができる。パズルは、パズル・データベース１１６に記憶する前に暗号化できる。秘密鍵１０８は、トークン１１０を脅かすために使用できないように、プロビジョニング・デバイス１０２によって破壊できる。

【００３９】

代替実装形態では、プロビジョニング・デバイス１０２は秘密鍵を生成することができない。代わりに、トークン１１０には他の手段によってあらかじめ秘密鍵がプロビジョニングされ、トークン１１０は、そのトークンに関連する複数のパズルを生成すべき秘密鍵を一時的に供給するのに十分長い間、プロビジョニング・デバイスに結合される。

【００４０】

図２に、一例による、秘密鍵を生成し、トークンに配信し、その秘密鍵に関連する複数のパズルを生成するようにプロビジョニング・デバイス上で動作可能な方法を示す。２０２において、擬似ランダム秘密鍵を取得する。この秘密鍵は、トークンによって供給されるか、またはプロビジョニング・デバイスによって生成されるかのいずれかが可能である。たとえば、秘密鍵とナンスとを入力として受け取る擬似乱数生成器（ＰＲＧ）を使用して、ランダムに見えるビットの任意に長い文字列として擬似ランダム秘密鍵を生成することができる。多くの代替策があるが、非線形ＳＯＢＥＲ（ＮＬＳ）などのセキュアなストリーム暗号を使用してＰＲＧをインスタンス化することができる。

【００４１】

２０４において、秘密鍵に関連する複数のパズル（たとえば、暗号化メッセージ）を生成する。２０６において、秘密鍵を外部トークン・デバイスに供給する。また、２０８において、複数のパズルを外部ヘルパー・デバイスに送信し、その外部ヘルパー・デバイスに記憶する。次いで２１０において、プロビジョニング・デバイスは秘密鍵を廃棄する。

【００４２】

生成され、ヘルパーに供給されるパズルの数は極めて大きく（たとえば、数千、数百万などに）なることがある。プロビジョニング・デバイス１０２は、トークン１１０に供給される秘密鍵に基づいてアルゴリズム的にこれらのパズルを生成することができる。したがって、トークンは要求に応じて特定のパズルを再生することができるので、トークン自体がすべてのパズルを記憶する必要はない。しかしながら、ヘルパーには秘密鍵が与えられず、したがって、ヘルパーは、トークンに関連するパズルのすべてを記憶する必要がある。これが実現される少なくとも２つの異なる方法が存在する。

【００４３】

パズル・データベース１１６は、作成された後にヘルパーに通信される。いくつかの実装形態では、プロビジョナとヘルパーは同じデバイスとすることができる。代替実装形態では、関連するトークンを用いて、パズル・データベースを含む記憶媒体（たとえば、ディスク）を分散することができる。次いで、エンドユーザは、適切なヘルパーを選択し、パズル・データベースを（記憶媒体から）そのヘルパーに配信する。ベリファイアからパズルの要求が受信されたときどのパズルを送信すべきかを識別できるように、ヘルパーには（受信したパズル・データベースに対応する）ユーザの識別情報及び／またはトークン識別子を教えることができる。

【００４４】

様々な適用例では、複数のヘルパーがトークンのパズル・データベースを受信し、それ

によって冗長性及びスケーリングを与えることができる。ヘルパーは、複数の異なるトークンのパズル・データベースを記憶することができる。

【００４５】

パズル生成

各々が中程度の困難さ量の多数のパズルを作成することによって、ベリファイアは、中程度の計算労力量でパズルを解くことができるようになる。パズルは、ベリファイアによってブルート・フォース・アタックができるようにするのに十分短い未知のパズル鍵（ＰＫ）を用いて符号化されたメッセージの形態とすることができる。

【００４６】

パズルはランダムな順序で生成できる。各パズルは、パズル識別子フィールド（ＰＩＤ）、パズル秘密鍵（ＰＳ）、及びパズル・メッセージ認証コード（ＰＭ）フィールドを含むことができる。パズル識別子（ＰＩＤ）は、可能な数のパズルの範囲内でランダムに生成できる。一例では、ＰＩＤは３２ビットとすることができる。パズル秘密（ＰＳ）は完全強度暗号鍵（たとえば、１２８ビット）とすることができる。一例では、ＰＳは、（トークン３０２またはプロビジョニング・デバイス１０２において、）パズル生成機能またはアルゴリズム（ＰＧＦ）によってトークン秘密鍵（ＳＫ）及び／またはパズル識別子（ＰＩＤ）に基づいて計算できる（すなわち、ＰＧＦ（ＳＫ，ＰＩＤ））。パズル・メッセージ認証コード（ＰＭ）はＰＳ及びＰＩＤのメッセージ認証コード（すなわち、ＰＭ（ＰＳ，ＰＩＤ））として定義でき、それによりベリファイアはいつパズルを首尾よく「クラック」または解読したかを知ることができる。すなわち、ベリファイアは、パズルが首尾よく解かれたことを示すＰＩＤ、または何らかの他のフラグ、マーカー、もしくはビット・シーケンスを識別することによって、パズルが首尾よく解読できたかどうかを知ることができる。

【００４７】

ヘルパーに配信する前に、トークンまたはプロビジョニング・デバイスのいずれかによってランダムに生成された、一意の限られた強度のパズル鍵（ＰＫ）を使用してパズルを暗号化することができる。この暗号化の強度、及び生成すべきパズルの個数については、ヘルパー・パズル・データベースが脅かされる可能性がどの程度あるかのみに依存する。ハッキングまたは不正アクセスの可能性がないとみなされる場合、パズル暗号化をまったくなしとし、潜在的なベリファイアごとにより少ないパズルを使用することができる。

【００４８】

パズルシステムを使用すると、ヘルパーが脅かされた場合に多少追加のセキュリティが与えられる。そのようなセキュリティは、特定のトークンに使用されるパズルの複雑さ及び数に基づくことができる。たとえば、特定のトークンに対して多数の容易に解読可能なパズルを使用することができ、その場合、セキュリティは、ベリファイアによって選択された１つのパズル、またはパズルの組合せによって与えられる。ベリファイアは、（ヘルパーから受信された多数のパズルから）１つまたは複数の擬似ランダムに選択されたパズルを解読するかまたは解くと、選択されたパズルごとにＰＩＤ及びＰＳを取得することができる。ベリファイアは、トークンに送信される（１つまたは複数の）ＰＩＤに対応するパズル秘密（ＰＳ）に基づいてセッション鍵または検証鍵を（ＶＫ）生成することができる。ベリファイアからＰＩＤを受信すると、トークンもセッション鍵または検証鍵（ＶＫ）を生成することができる。したがって、ベリファイア及びトークンの両方が共通の対称鍵に関してセキュアに同意することができる。盗聴者または脅かされたヘルパーは、ベリファイアによってどの（１つまたは複数の）パズルが解かれたかを知らないので、セッション鍵または検証鍵（ＶＫ）を判定することはより難しいタスクになる。せいぜい盗聴者またはヘルパーは、特定のトークンに関連するすべてのパズルを解こうと試みることができるが、極めて多数（たとえば、数百万）存在するので、これは大いにより計算量的に高コストになる。

【００４９】

初期認証段階

トークンに秘密鍵がプロビジョニングされ、ヘルパー・デバイスが関連するパズルを取得すると、ヘルパー・デバイスの支援を受けてトークンと検証デバイスとの間で認証を実行することができる。トークンと特定のベリファイアとの間で初期認証するためのプロセスは、ベリファイアに対して計算集約的であり、おそらくは従来の公開鍵システムよりも計算集約的である。したがって、一実装形態では、初期認証段階は、トークンをベリファイアに「紹介」して、トークンとベリファイアが共有された対称鍵をネゴシエートできるようにするために使用できる。その後、より短い完全対称鍵ベースのプロセスを事後認証（再認証）に使用することができる。

【００５０】

図３に、トークン３０２、検証デバイス３０４、及びヘルパー・デバイス３０６がトークン３０２の初期認証中にどのように動作するかを示す。トークン３０２は、限定を与えられた処理回路３０８と、秘密鍵（ＳＫ）、トークンＩＤ、検証鍵ＶＫ、及びパズル生成アルゴリズムもしくは機能を記憶するための記憶デバイス３１０と、低帯域幅で非セキュアな通信インターフェース３１２とを含むデバイスとすることができる。低帯域幅インターフェースは、ＵＳＢ、Ｂｌｕｅｔｏｏｔｈ(登録商標）、Ｎｅａｒ－Ｆｉｅｌｄ Ｃｏｍｍｕｎｉｃａｔｉｏｎ、及びセルラー／ＬＤＣなどにわたる範囲にある、数値キーパッド及び８桁ディスプレイほどの単純なものでよい。

【００５１】

ベリファイア３０４は、トークンよりも著しく多くの処理リソースを有する処理回路３１４と、検証鍵及び関連するトークンＩＤを記憶するための記憶デバイス３１５と、高帯域幅でセキュアな通信インターフェース３１６とを含むデバイス、エンティティ、及び／または仮想オブジェクトとすることができる。ヘルパー３０６は、処理回路３１８と、トークン３０２に関連する複数のパズルを記憶するための記憶デバイス３２０と、高帯域幅でセキュアな通信インターフェース３２２とを含むデバイス、エンティティ、及び／または仮想オブジェクトとすることができる。

【００５２】

ユーザがトークン３０２を使用して自分自身をベリファイア３０４で認証しようと試みると、トークンは認証要求を（トークンＩＤとともに）ベリファイア３０４に送信する。ベリファイア３０４及びヘルパー３０６は、データを通信及び／または交換することが可能である高帯域幅ネットワーク（たとえば、インターネット、プライベート・ネットワークなど）に結合できる。ベリファイア３０４は（トークンＩＤに関連する）複数のパズルをヘルパー３０６に要求する。ヘルパー３０６は応答して、（受信したトークンＩＤに関連する）パズルの擬似ランダムに選択されたセットをベリファイア３０４に送信する。いくつかの実装形態では、ヘルパー３０６とベリファイア３０４は、それらの相互作用をセキュアにするためにトランスポート層セキュリティ（ＴＬＳ）プロトコルまたはセキュリティ・インターネット・プロトコル（ＩＰｓｅｃ）などのよく知られているセキュリティ・プロトコルを使用して、オープンなインターネット上で通信することができる。

【００５３】

次いでベリファイア３０４は、受信したパズルのうちの少なくとも１つを選択し、ブルート・フォース・アタックによって解読またはクラックする。パズルを解読またはクラックすると、ベリファイアはパズルからパズル秘密（ＰＳ）及びパズルＩＤ（ＰＩＤ）を取得する。ベリファイア３０４により、パズル秘密（ＰＳ）に基づいて検証鍵（ＶＫｖ）が生成される。ベリファイア３０４は、パズルＩＤ（ＰＩＤ）と、検証鍵（ＶＫｖ）のメッセージ認証コード（ＭＡＣ）とをトークン３０２に送信する。トークン３０２は、受信したＰＩＤと、そのあらかじめ記憶された秘密鍵（ＳＫ）及びパズル生成アルゴリズムとを使用して、ＰＩＤのパズル秘密を取得する。次いでトークン３０２は、パズル秘密を使用して、それ自体のローカル検証鍵（ＶＫｔ）を再生することができる。検証鍵は一致するはずであり（ＶＫｔ＝ＶＫｖ）、それにより検証鍵ＶＫをトークン３０２とベリファイア３０４との間の認証のための対称鍵として使用することが可能になる。トークン３０２は、ベリファイア３０４との将来の認証のために検証鍵（ＶＫ）を記憶することができる。

【００５４】
いくつかの実装形態では、ベリファイア３０４は、複数のパズル秘密及びパズル識別子を取得するために複数のパズルを破る、クラックする、または解読することができることに留意されたい。複数のパズル秘密に基づいて検証鍵（ＶＫｖ）が生成でき、それによってさらなる強度が得られる。対応するパズル識別子がトークン３０２に送信され、次いでトークン３０２はそれらのパズル識別子を使用してそのローカル検証鍵（ＶＫｔ）を生成することができる。
【００５５】
図４に、ヘルパーの支援を受けたトークンとベリファイアとの間の初期認証のための方法を示す。トークン４０２に秘密鍵（ＳＫ１）をプロビジョニングする。ユーザは、認証を必要とするセキュアなトランザクションを実行したいと望むとき、トークン４０２を使用して、トークン識別子（トークンＩＤ）とともに認証要求４１２を送信することによってベリファイア４０４との認証を開始することができる。認証ヘルパー４０６は、トークン４０２（またはトークンのユーザ）によって識別されるか、またはデフォルトとして識別できる。たとえば、トークン４０２またはユーザのいずれかが、トークンのパズルが記憶されているヘルパー４０６のコンタクト情報をベリファイア４０４に供給することができる。
【００５６】
ベリファイア４０４は、ヘルパー４０６とのセキュアな及び／または認証された接続を確立する。ベリファイア４０４はまた、４１４において第１のランダム・ナンス（Ｎｏｎｃｅ１）を生成する。次いでトークンＩＤ及びＮｏｎｃｅ１をヘルパー４０６に送信する。
【００５７】
ヘルパー４０６には、以前にトークン４０２のパズル・データベース４１０がプロビジョニングされており、ヘルパー４０６はトークンＩＤを使用してパズル・データベース４１０からパズルを選択する。一実装形態では、ヘルパー４０６は、４１８において第１のナンス（Ｎｏｎｃｅ１）を使用して、トークン４０２に関連するＮ個のパズルの擬似ランダム選択（セット）を選択し（Ｎは、２と数百万の間の整数である）、４２０においてそれらをベリファイアに戻すことができる。Ｎｏｎｃｅ１がベリファイア４０４によって供給された場合、ヘルパー４０６は、パズルのセットが、受信したＮｏｎｃｅ１の影響を受けて選択されたという証明を与えることができる。次いで、ベリファイア４０４と４０６との間の接続を閉じるかまたは終了する。
【００５８】
４２２において、ベリファイア４０４は受信したパズルのうちの１つまたは複数を選択する。いくつかの実装形態では、ｔ個のパズルのランダムなシーケンスを選択し、ただし、ｔ≧１であり、ｔはすべての受信したパズルのサブセットである。ヘルパー４０６が脅かされた場合、より多くのパズルを使用するほどより多くのセキュリティが得られる。なぜならば、ヘルパー４０６及び攻撃者は、検証鍵を生成するためにどのｔ個のパズルが使用されるのか、またはどんな順序（シーケンス）で結合されるのかを知らないからである。
【００５９】
選択されたパズルが暗号化されている場合、４２４において、ベリファイア４０４は（たとえば、ブルート・フォース・アタック技法を使用して）その選択されたパズルを解いて（解読またはクラックして）、選択されたパズルの各々のパズル識別子（ＰＩＤ）及びパズル秘密（ＰＳ）を取得する。
【００６０】
パズル秘密（ＰＳ）と、場合によっては第２の擬似ランダム・ナンス（Ｎｏｎｃｅ２）４２６とに基づいて、４２８においてベリファイア４０４は検証鍵（ＶＫｖｅｒ）を計算する。たとえば、ハッシュ関数Ｈ（たとえば、セキュアなハッシュアルゴリズムＳＨＡ－２５６など）を使用して検証鍵ＶＫｖｅｒ＝Ｈ（Ｎｏｎｃｅ２，ＰＳ）を生成することが

できる。いくつかの実装形態では、追加のセキュリティのために、複数のパズルを使用して結果的にパズル秘密PS_1、PS_2．．．、PS_tが検証鍵ＶＫｖｅｒ＝Ｈ（Ｎｏｎｃｅ２，PS_1，．．．，PS_t）に組み合わせられるようにする。４３０において、ベリファイア４０６は、第２のナンス（Ｎｏｎｃｅ２）とｔ個のＰＩＤの順序付きリストとを含むことができるメッセージをトークン４０２に送信する。４３０において、ベリファイア４０４はまた、入力として秘密検証鍵（ＶＫｖｅｒ）及びメッセージを取り、タグＭＡＣ（ＶＫｖｅｒ，Ｎｏｎｃｅ２，ＰＩＤ）を出力するメッセージ認証コード（ＭＡＣ）を送信する。４３２において、ベリファイア４０４はまた、事後認証に使用するためにこのユーザに関連するベリファイアＶＫｖｅｒを記憶する。

【００６１】

ベリファイア４０４からＰＩＤ及びＭＡＣを受信すると、４３４においてトークン４０２は、受信したＰＩＤに基づいて解かれたパズルのパズル秘密ＰＳを再生する。たとえば、トークン４０２は、その秘密鍵（ＳＫ１）及びパズル生成アルゴリズムを使用して、識別されたＰＩＤのパズル秘密ＰＳを取得することができる。次いで４３６において、トークン４０２は、パズル秘密ＰＳ及び第２のナンス（Ｎｏｎｃｅ２）に基づいてローカル検証鍵ＶＫｔｏｋｅｎを生成する。４３８において、トークン４０２は、ローカルＭＡＣ（ＶＫｔｏｋｅｎ，Ｎｏｎｃｅ２，ＰＩＤ）に対して受信したメッセージ認証コードＭＡＣ（ＶＫｖｅｒ，Ｎｏｎｃｅ２，ＰＩＤ）を検証する。ＭＡＣが等しくない場合、受信されたＭＡＣ（ＶＫｖｅｒ，Ｎｏｎｃｅ２，ＰＩＤ）及び／またはメッセージのコンテンツの改ざん（または偶発的破損）を示すエラーが示される。トークン４０２はまた、リプレイ・アタックを防ぐために、記憶しているのと同じ検証鍵ＶＫをすでに有してないかを確認する。上首尾であると仮定すると、４４０において、トークン４０２はＶＫを空きスロットに記憶する。すなわち、トークン４０２は（異なるアプリケーションまたはベリファイアの）検証鍵をスロット中に編成する。ユーザは、後の選択または表示のために、検証鍵ＶＫに関連するスロット番号の記録を取るか、または検証鍵ＶＫに関連するスロットに名前を割り当てることができる。

【００６２】

４４２において、トークン４０２は、（後述の事後認証段階の場合とほとんど同様に）その認証応答を生成し、ベリファイアに送信する。ベリファイア４０４は、適正であるはずの応答を受信し、アカウントを使用準備完了にマーキングする。

【００６３】

事後認証段階

トークンと特定のベリファイアとの間で検証鍵（ＶＫ）がセットアップされると、それらの間の事後認証要求においてその検証鍵（ＶＫ）を使用することができる。検証鍵（ＶＫ）は、トークンとベリファイアとの間で秘密またはセキュアに保たれる。その検証鍵（ＶＫ）は、トークンとベリファイアとの間の質問応答認証プロセスにおける対称鍵として使用できる。いくつかの実装形態では、通信のプライバシーのために検証鍵（ＶＫ）を使用して２つのデバイス間のメッセージを暗号化することもできる。

【００６４】

図５に、トークンとベリファイアとの間の事後認証を実行するための方法を示す。トークン５０２とベリファイア５０４との間で、セキュアな検証鍵ＶＫａ５０６及び５０８を以前に確立していることができる。この検証鍵ＶＫａは、トークン５０２とベリファイア５０４との間の関係に特有なことがあり、他のベリファイアには適用不可能である。５１０において、トークン５０２は、そのトークンＩＤとともに認証要求をベリファイア５０４に送信する（これは、たとえば、顧客がトランザクションを実行するように銀行出納係に依頼する場合など、間接的なことがある）。ベリファイア５０４はトークンＩＤを使用して、トークン５０２に対して以前に確立された検証鍵ＶＫａ５０８を識別する。５１２において、ベリファイア５０４は質問及びＭＡＣ（ＶＫａ，質問）を生成する。質問は、任意のもしくは擬似ランダムな数、文字列、及び／またはビットもしくはシンボルのシーケンスとすることができる。次いで５１４において、質問及びＭＡＣ（ＶＫａ，質問）を

トークンに送信する。受信された質問と、その以前に記憶された検証鍵ＶＫａ５０６とを使用して、５１６においてトークン５０２はＭＡＣ（ＶＫａ，質問）を検証する。すなわち、トークン５０２は、受信したＭＡＣ（ＶＫａ，質問）が、ローカルに計算されたＭＡＣ（ＶＫａ，質問）と同じであるかどうかを検証する。

【００６５】

一実装形態では、トークン５０２は、認証要求５１０の送信先であるベリファイアがどれであるかを知っていることが可能である。たとえば、ユーザは、トークン５０２によって使用されるべき（特定のベリファイアに関連する）検証鍵ＶＫａ５０６を識別することができる。そのような実装形態では、受信したＭＡＣ（ＶＫａ，質問）を検証するとき、トークン５０２は単に、使用するように言われた検証鍵ＶＫａ５０６を使用する。

【００６６】

他の実装形態では、トークン５０２は、どのベリファイアと通信しているのかを知ることができない。そのような場合、トークンは、単に、その記憶されている検証鍵の各々に対して受信したＭＡＣ（ＶＫａ，質問）を検証しようと試みることができる。記憶されている検証鍵のうちの１つが同じＭＡＣを生成する場合、それは適正な検証鍵であると仮定される。

【００６７】

検証鍵ＶＫａが認証されると、５１８においてトークン５０２は応答及びＭＡＣ（ＶＫａ，応答）をベリファイア５０４に送信する。この応答は、任意のもしくは擬似ランダムな数、文字列、及び／またはビットもしくはシンボルのシーケンスとすることができる。トークンとベリファイアとの間のメッセージが同じにならないように質問と応答は互いに異なるように選択される。概して、セキュリティについて、応答は最初に発された質問に関係することができるが、同じでないことがある。たとえば、応答は、受信された質問文字列よりも大きい文字列である文字列のことがある。そのような場合、応答自体を送信する必要がない。５２０において、ベリファイア５０４は、その知られている検証鍵ＶＫａ５０８と受信した応答とを使用することによって受信したＭＡＣ（ＶＫａ，応答）を検証する。受信したＭＡＣが、ベリファイア５０４によって計算されたローカルＭＡＣに一致する場合、認証は完了である。次いで５２２において、トークン５０２、ベリファイア５０４のいずれか、または両方によってセッション鍵を確立する。

【００６８】

盗聴者が以前に使用されたセッション鍵を再利用しようと試みるリプレイ・アタックを防ぐために、トークンはセキュリティ機構を実装することができる。たとえば、様々なトークンは、認証を証明するためにシーケンス・カウンタまたはタイムスタンプを実装することができる。カウンタまたはタイムスタンプは、検証鍵ＶＫを知っているパーティのみが予測できる形で変わる。したがって、トークンまたはセッション鍵から以前のＭＡＣ応答を取得した盗聴者は、それを再利用することが不可能である。

【００６９】

シーケンス・カウンタを用いた認証

図６に、カウンタを使用したトークンとベリファイアとの間の認証を実行するための方法を示す。トークン６０２とベリファイア６０４の両方の上で、検証鍵ＶＫａ６０６及び６０８、ならびに関連する順次カウンタｃｏｕｎｔｅｒ＿ＶＫａ６１０及び６１２をプロビジョニングする。検証鍵ＶＫａ６０６及び６０８、ならびに順次カウンタｃｏｕｎｔｅｒ＿ＶＫａ６１０及び６１２は、以前に構成しているか、及び／またはトークン６０２とベリファイア６０４との間で同期していることが可能である。たとえば、検証鍵ＶＫａがトークン６０２とベリファイア６０４との間で最初に確立されるとき、トークン６０２及びベリファイア６０４はそれらのカウンタ６１０とカウンタ６１２を同期させることができる。カウンタ６１０及び６１２は、無許可パーティがトークン６０２及び／またはベリファイア６０４として見せかけようと試みる場合のセキュリティ対策として使用される。（トークン６０２における）ｃｏｕｎｔｅｒ＿ＶＫａ６１０、及び（ベリファイア６０４における）ｃｏｕｎｔｅｒ＿ＶＫａ６１２は、無許可パーティがトークン６０２及び／ま

たはベリファイア６０４として見せかけようと試みる場合のセキュリティ対策として使用される。

【００７０】

トークンは、そのトークンＩＤとともに認証要求６１４を送信することによってトランザクションを開始する。認証要求を受信すると、ベリファイア６０４は、トークンＩＤを使用して、トークン６０２に関連する検証鍵ＶＫａ６０８を識別する。次いでベリファイア６０４は、検証鍵ＶＫａ６０８に対応するｃｏｕｎｔｅｒ＿ＶＫａ６１２を選択する。６１６において、ベリファイア６０４においてｃｏｕｎｔｅｒ＿ＶＫａを固定の増分だけ増分してｃｏｕｎｔｅｒ＿ＶＫａ’を取得する。６１８において、質問と、検証鍵ＶＫａ、ｃｏｕｎｔｅｒ＿ＶＫａ’、及び質問のメッセージ認証コード（ＭＡＣ（ＶＫａ，ｃｏｕｎｔｅｒ＿ＶＫａ’，質問））とを生成する。６２０において、この質問及びＭＡＣ（ＶＫａ，ｃｏｕｎｔｅｒ＿ＶＫａ’，質問）をベリファイアからトークンに送信する。トークン６０２によって記憶されている各検証鍵ＶＫｎについて、ａ）６２２において、関連するｃｏｕｎｔｅｒ＿ＶＫｎを取り出し（ただし、１≦ｎ≦Ｎであり、Ｎは、トークンによって記憶されている検証鍵の総数である）、ｂ）次いで検証鍵ＶＫｎ及びｃｏｕｎｔｅｒ＿ＶＫｎごとにＭＡＣ（ＶＫｎ，ｃｏｕｎｔｅｒ＿ＶＫｎ＋ｉ，質問）を計算し（たとえば、１≦ｉ≦５）、ｃ）計算されたＭＡＣが受信したＭＡＣに合致する場合、選択された検証鍵（すなわち、ＶＫａ）は適正であると仮定する。いくつかの実装形態では、応答をより迅速にするために、質問は、各検証鍵ＶＫｎのＭＡＣをあらかじめ計算し記憶できるような既知の数、文字列、またはビット・シーケンスとすることができる。

【００７１】

適正な検証鍵ＶＫａを発見すると、６２８においてトークン６０２は、そのｃｏｕｎｔｅｒ＿ＶＫａ’＝ｃｏｕｎｔｅｒ＿ＶＫａ＋増分と更新する。したがって、トークン６０２とベリファイア６０４の両方が、同期されたカウンタを維持している。次いでトークンは、６３０において応答及びＭＡＣ（ＶＫａ，ｃｏｕｎｔｅｒ＿ＶＫａ’，応答）を計算し、６３２においてベリファイアに送信する。

【００７２】

ベリファイア６０４は、それ自体のＶＫａ、及びｃｏｕｎｔｅｒ＿ＶＫａ’、ならびに受信した応答を使用して、受信したＭＡＣ（ＶＫａ，ｃｏｕｎｔｅｒ＿ＶＫａ’，応答）を検証する。受信したＭＡＣが首尾よく検証された場合、トークン６０２は認証されたとみなす。６３６及び６３８において、トークン６０２とベリファイア６０４の両方はセッション鍵＝ＭＡＣ（ＶＫ，ｃｏｕｎｔｅｒ＿ＶＫａ’，「Ｋ」）を計算し、ただし「Ｋ」は、既知であるかまたはトークン６０２とベリファイア６０４の両方が判断できる、秘密鍵またはあらかじめ構成された値とすることができる。

【００７３】

トークン６０２が攻撃者からランダムな質問を受信した場合、トークン６０２は対応する検証鍵ＶＫを発見することができず、それはエラーを示し、認証は失敗することに留意されたい。

【００７４】

タイムスタンプを用いた認証

図７に、タイマを使用したトークンとベリファイアとの間の認証を実行するための方法を示す。この方法は、ユーザに、トークンに対してベリファイアを識別させることによって認証プロトコルを単純化し、それによって（ベリファイアを識別するのに用いる）ベリファイア７０４からの質問の必要を回避する。この機構はまた、認証ステップの数を低減するために（上述のカウンタベースの認証プロトコルなどの）他のプロトコルにおいて実装できる。この方法では、トークン７０２及びベリファイア７０４が、同期されたタイマＴｔｏｋｅｎ７１０及びＴｖｅｒ７１２（たとえば、協定世界時ＵＴＣ）を妥当な精度で維持することができると仮定する。たとえば、トークン７０２は、プロビジョニング中に同期または初期化されるタイマＴｔｏｋｅｎ７１０を有することができる。ベリファイア７０４は、そのタイマＴｖｅｒ７１２が同期及び／または維持されるネットワーク接続を

有することができる。７１４において、トークン７０２は、そのトークンを認証しているベリファイア７０４を識別する指示を（ユーザなどから）受信する。たとえば、ユーザは、以前は特定のベリファイア７０４に関連付けられていた特定の名前または検証鍵ＶＫａ７０６を選択することができる。このようにして、トークン７０２は、以前はベリファイア７０４に関連付けられていた適切な検証鍵ＶＫａ７０６を取得することができる。トークンは、７１６において応答及びＭＡＣ（ＶＫａ，タイムスタンプ，応答）を生成し、７１８においてベリファイアに送信する。応答において使用されるタイムスタンプはタイマＴｏｋｅｎ７１０から取得する。７１８において、トークンの認証のために応答、ＭＡＣ（ＶＫａ，タイムスタンプ，応答）、及びトークンＩＤをベリファイアに送信する。

【００７５】

７２０において、ベリファイアは、（トークンＩＤに対応する）検証鍵ＶＫａ７０８を取り出し７０８、（タイムスタンプを取得するための）タイマＴｖｅｒ７１２と受信した応答とを使用してＭＡＣのローカルバージョンを計算することによって、受信したＭＡＣ（ＶＫ，タイムスタンプ，応答）を検証する。タイムスタンプの分解能は、タイマＴｏｋｅｎ７１０とタイマＴｖｅｒ７１２との間のわずかな相違、またはトークン７０２によってＭＡＣが生成された時刻と、それがベリファイア７０２に与えられた時刻との間の遅延に適応するように調整できる。たとえば、タイムスタンプは、＋／－３０秒または何らかの他のタイムウィンドウの分解能を有することができる。

【００７６】

いくつかの実装形態では、７２２及び７２４において、トークン７０２及び／またはベリファイア７０４は、特定のトランザクション中のセッション鍵として使用するためのＭＡＣ（ＶＫ，タイムスタンプ，セッション鍵）を計算することができ、ただし、セッション鍵は、既知であるかまたはトークン７０２とベリファイア７０４の両方が判断できる、擬似ランダムまたは任意の数、文字列、及び／またはビット・シーケンスである。

【００７７】

トークンは、ベリファイア７０４から質問を受信しないので、ベリファイアが本物であるかどうかを知らないことに留意されたい。したがって、トークンは、信頼できるベリファイアにトークンを提示しているユーザに依拠することができる。追加のセキュリティが所望される場合、このトークン７０２は、ベリファイアが有効または本物である（たとえば、ベリファイアが検証鍵ＶＫａを知っている）かどうかを確認することができる質問をベリファイア７０４に要求することができる。

【００７８】

使用のシナリオ

単一ファクタ認証システムでは、ユーザは、自分自身をベリファイアで認証するためにトークンを使用することができる。すなわち、適正な検証鍵を有するトークンを占有しているだけで、ユーザを認証するのに十分である。ただし、単一ファクタ認証は、物理トークンへのアクセス権を得た人は誰でも、ベリファイアによって守られているアカウント及び／または情報に不正にアクセスできるという欠点を有する。

【００７９】

２ファクタ認証システムでは、より強いセキュリティを達成するために２つのセキュリティ対策を使用する。１つのそのようなファクタは、ユーザが認証の時点で供与する、セキュアなパスワード、鍵、識別子、写真識別、指紋、ボイスサンプルなどとすることができる。第２のファクタは、同じく認証の時点で提示される適正な（１つまたは複数の）検証鍵を記憶しているトークンとすることができる。

【００８０】

本明細書で説明するトークンとベリファイアとの間の認証プロトコルは、手動で、自動で、またはそれらの組合せのいずれかで実行できる。いくつかの例では、ユーザは、トークンとベリファイアの仲介者として働きをすることができる。たとえば、質問及び応答（たとえば、ＭＡＣ）は、ユーザがトークン及び／またはベリファイアに手動で入力することが可能な比較的短い数字、文字及び／または記号の列とすることができる。他の例では

、トークンを（たとえば、ワイヤレス、ユニバーサル・シリアルバス、音声などの）インターフェースによってベリファイアに直結し、それによってそのインターフェースなどを介して認証メッセージを交換することができる。トークンとベリファイアが直接互いに通信する実装形態では、認証プロトコルは、追加のセキュリティのためにより長い質問及び応答、より多くのパズルなどを使用することができる。

【００８１】

トークンの例

図８は、複数の異なるベリファイア８０４、８０６、及び８０８で認証するためにトークン８０２が複数の検証鍵をどのように記憶するかを示すブロック図である。トークン８０２は、限られた処理リソース及び限られた帯域幅通信インターフェースを有し、図１～図７に示すように動作するように構成できる。いくつかの従来技術のトークンは、異なるベリファイアとの認証のための複数の公開鍵を記憶することができるが、そのような公開鍵認証システムを実装するためにかなりの処理リソース及び通信帯域幅を要求することがある。他のトークンは、セキュリティの考慮事項により、ただ１つのベリファイアで利用できる対称鍵を記憶することができる。そのような従来技術のトークンとは対照的に、このトークン８０２は、限られた処理リソース及び限られた帯域幅インターフェースを利用しながら複数の対称鍵を取得し、セキュアに記憶するように構成される。すなわち、トークン８０２は、図１～図７に示す方法で複数の検証鍵（すなわち、対称鍵）を取得し、それによりトークン８０２を複数の異なるベリファイアで認証することが可能になる。詳細には、検証鍵は、トークン８０２、及びベリファイア８０４、８０６、８０８によってセキュアに生成され、トークン８０２とベリファイア８０４、８０６、８０８との間で送達または送信はされない。セキュアな方法で初期配備した後に追加の検証鍵（すなわち、対称鍵）をトークンに追加することができる。

【００８２】

図９に、複数の異なるベリファイアで対称鍵を使用してセキュアな認証を実行するようにトークン上で動作可能な方法を示す。プロビジョニング段階９０２中に、秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングする。９０４（初期認証段階）において、トークンは、そのトークン識別子をベリファイアに供給することによってベリファイアとの認証を開始する。ユーザまたはトークンは、トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスをベリファイアに供給することができる。

【００８３】

９０６において、第１の検証鍵に基づく１つまたは複数のパズル識別子とメッセージ認証コードとを含む質問メッセージをベリファイアから受信し、第１の検証鍵は、１つまたは複数のパズル識別子によって識別される１つまたは複数のパズルに関連する１つまたは複数のパズル秘密の関数である。９０８において、トークンは、１つまたは複数のパズル生成アルゴリズム、受信した１つまたは複数のパズル識別子、及び秘密鍵に基づいて１つまたは複数のパズル秘密を（独立して）取得する。たとえば、トークンは、パズル生成アルゴリズムへの入力として秘密鍵及び／または１つまたは複数のトークン識別子を使用して、トークン識別子に対応する１つまたは複数のパズル及び／または１つまたは複数のパズル秘密を生成することができる。９１０において、１つまたは複数のパズル秘密に基づいて第２の検証鍵をトークンによって生成する。９１２において、トークンは、第１の検証鍵と第２の検証鍵が同じであるかどうか判断するために受信したメッセージ認証コードを検証する。９１４において、トークンは第２の検証鍵に基づいてベリファイアへの応答を生成する。メッセージ認証コードが首尾よく検証された場合、９１６において、トークンは第２の検証鍵を記憶し、その第２の検証鍵をベリファイアに関連付ける。第１の検証鍵及び第２の検証鍵は、トークンとベリファイアとの間の事後認証のための対称鍵として使用される。同様の方法で、トークンは、他のベリファイアを用いて異なる対称鍵をセットアップすることができる。このようにして、単一のトークンを使用して、異なるベリファイアで使用される複数の検証鍵（すなわち、対称鍵）を記憶することができる。トーク

ンは、他の検証鍵のセキュリティを脅かすことなしに必要に応じて異なるベリファイアの追加の検証鍵を確立することができる。

【００８４】

ベリファイアの例

図１０は、トークンを認証するためにヘルパーの支援を受けて対称鍵を確立するように構成されたベリファイア１００２を示すブロック図である。ベリファイア１００２は、かなりの処理能力をもつ処理回路１００４と、トークンと通信するための低帯域幅通信インターフェース１００６（たとえば、キーパッド、ワイヤレス・トランスポンダなど）と、ヘルパーと通信するための高帯域幅通信インターフェース１００８と、ヘルパーから受信したパズルとトークンに関連する検証鍵とを記憶するための記憶デバイス１０１０とを含むことができる。ベリファイア１００２は、図１～図７に示すように動作するように構成できる。処理回路１００４は、低帯域幅通信インターフェース１００６を介してトークンからトークン識別子を受信し、高帯域幅通信インターフェース１００８を介してヘルパーに複数のパズルを要求するように構成される。ヘルパーのコンタクト情報は、トークン、トークンのユーザ、またはデフォルトのロケーション（たとえば、インターネットアドレス、Ｕｎｉｆｏｒｍ　Ｒｅｓｏｕｒｃｅ　Ｌｏｃａｔｏｒ（ＵＲＬ）など）によって与えることができる。次いで、処理回路１００４は、受信したパズルのうちの１つをランダムに選択し、その選択したパズルをブルート・フォース・アタックによって解読または復号する。そのようなブルート・フォース・アタックでは、処理回路１００４は、パズルを首尾よく解読または復号する鍵が発見されるまで様々な可能な鍵を試みることができる。パズルを暗号化する鍵の長さ（たとえば、３２ビット、６４ビット、１２８ビットなど）はベリファイア１００２に知られていてよく、それによって鍵の探索を制限することができる。

【００８５】

図１１に、ヘルパーの支援を受けてトークンのセキュアな認証を実行するベリファイア上で動作可能な方法を示す。１１０２において、（低帯域幅インターフェースを介して）トークンから認証要求をトークン識別子とともに受信する。１００４において、ベリファイアは、トークンに関連するヘルパーのヘルパー・アドレスを取得する。次いで１１０６において、ベリファイアは、トークンに対応する複数（たとえば、数千、数十万、または数百万）のパズルをヘルパーに要求する。いくつかの実装形態では、ヘルパーによるバイアスなしに擬似ランダムに複数のパズルが選択されるように、ベリファイアは第１の擬似ランダム・ナンスをヘルパーに送信することもできる。次いで１１０８において、ベリファイアは、受信したパズルのうちの１つを擬似ランダムに選択し、その選択したパズルをブルート・フォース・アタックによって解いて、パズル秘密及び対応するパズル識別子を取得する。たとえば、ベリファイアは、パズルを符号化した鍵の長さを知っている場合、パズルが解かれるまで（たとえば、メッセージが首尾よく復号されるまで）すべての可能な鍵を試みることができる。次いで１１１０において、ベリファイアはパズル秘密を使用して検証鍵を生成する。１１１２において、ベリファイアは検証鍵に基づいて質問メッセージ認証コードを生成する。１１１４において、パズル識別子及び質問メッセージをトークンに送信する。１１１６において、同じ検証鍵をトークンが知っていることを証明する応答メッセージ認証コードをトークンから受信する。１１１８において、ベリファイアは将来の認証のために検証鍵を記憶し、その検証鍵をトークンに関連付ける。

【００８６】

いくつかの実装形態では、検証鍵を生成するために（ただ１つの代わりに）複数のパズルを使用することによって追加のセキュリティを与えることができる。ベリファイアは、複数のパズルを擬似ランダムに選択して解読し、それによって複数のパズル秘密を取得することができる。次いで、複数のパズル秘密を組み合わせて検証鍵を生成することができる。

【００８７】

ヘルパーの例

図１２は、トークンを認証するための対称鍵を確立する際にベリファイアを支援するように構成されたヘルパー１２０２を示すブロック図である。ヘルパー１２０２は、高帯域幅通信インターフェース１２０６と記憶デバイス１２０８とに結合された処理回路１２０４を含むネットワーク化されたデバイスとすることができる。処理回路１２０４は、パズルプロビジョナからそのインターフェース１２０６を介して複数のパズルを受信し、その複数のパズルを記憶デバイス１２０８に記憶するように構成される。パズルは、パズルプロビジョナによって識別された特定のトークンに関連付けられる。処理回路１２０４はまた、インターフェース１２０６を介して１つまたは複数のベリファイアからパズルの要求を受信するように構成される。その要求は、識別されたトークンに関連する複数のパズルを取り出し、送信するために処理回路１２０４が使用するトークン識別子を含む。ヘルパー１２０２は、トークンを用いて対称鍵を確立する際にベリファイアを支援することができるが、各トークンを解読するために必要な多数のパズル及び処理リソースにより、ヘルパー１２０２が対称鍵を決定することは法外に重くなる。

【００８８】

図１３に、トークンを認証する際にベリファイアを支援するように（ネットワーク）ヘルパー上で動作可能な方法を示す。１３０２において、ヘルパーは、トークン識別子に関連する複数（たとえば、数千、数十万、及び／または数百万）の（符号化されたまたは符号化されない）パズルをトークン・プロビジョナから受信する。１３０４において、ヘルパーはそれらの複数のパズルを記憶する。１３０６において、ヘルパーは、トークン識別子に関連するパズルについての後続の要求をベリファイアから受信する。この要求は、ベリファイアとトークンとの間で対称鍵（すなわち、検証鍵）が確立される、ベリファイアとトークンとの間の初期認証段階中に発生することがある。１３０８において、ヘルパーは、トークン識別子に関連する記憶された複数のパズルのサブセットを（擬似ランダムに）選択する。次いで１３１０において、パズルの選択されたサブセットをベリファイアに送信する。いくつかの実装形態では、ヘルパーは、ベリファイアに送信される複数のパズルを擬似ランダムに選択するために使用するナンス（たとえば、文字または数のストリングなど）を要求元ベリファイアから受信することができる。ヘルパーはまた、複数のパズルを選択する際にナンスが利用されたという証明（たとえば、メッセージ認証コードなど）を与えることができる。これにより、ヘルパーがパズルの選択を改ざんすることが防止される。

【００８９】

ヘルパーは、トークンの認証を実行する際にベリファイアを支援することはできるが、ベリファイアになりすますことが可能であってはならない。ヘルパーがそれ自体で悪意のある場合、パズルの一部を解読することに時間を費やし、それらをベリファイアに配信することのみを保証することが可能である。しかしながら、ベリファイアに、ベリファイアに送信するパズルを選択するために使用されたことをヘルパーが証明する擬似ランダム・ナンスを送信させることによって、ヘルパーによるそのような操作を防止することができる。また、トークンとベリファイアとの間の鍵確立段階でヘルパーが盗聴することができる場合、ヘルパーはそれらの間で対称鍵が確立されたことを確認することも可能である。ただし、鍵確立段階中に首尾よく通信を傍受することは起こりそうにない。

【００９０】

ヘルパーが初期鍵確立を妨害しないで、代わりにいくつかの事後認証を傍受する場合、ヘルパーは、トークンとベリファイアとの間で確立された検証鍵（対称鍵）を最終的に発見するために十分な情報を取得することができる。しかしながら、これは、ヘルパーが、トークンに関連するすべてのパズルを解くことと、（検証鍵が順序付きパズル秘密のサブセットに基づく）パズル秘密の様々な組合せを試みることと、検証鍵を生成する際に使用され得た（ヘルパーに明らかにされない）ナンスを発見することとを必要とする。したがって、より強いセキュリティを達成するために、ベリファイアは、パズルのより大きいセットをヘルパーに要求し、それらのより大きいサブセットを使用して、トークンでその検証鍵を確立することができる。

【００９１】
図１、図２、図３、図４、図５、図６、図７、図８、図９、図１０、図１１、図１２、及び／または図１３で示した構成要素、ステップ、及び／または機能のうちの１つまたは複数を、トークンとベリファイアとの間の認証に影響を及ぼすことなく、単一の構成要素、ステップ、または機能に再編成し及び／または組み合わせることができ、あるいは、いくつかの構成要素、ステップ、または機能で実施することができる。また、本発明から逸脱することなく追加の要素、構成要素、ステップ、及び／または機能を追加することができる。図１、図３、図８、図１０、及び／または図１２に示した装置、デバイス、及び／または構成要素は、図２、図４、図５、図６、図７、図９、図１１、及び／または図１３に記載した方法、特徴、またはステップのうちの１つまたは複数を実行するように構成できる。本明細書に記載の新規のアルゴリズムは、ソフトウェア及び／または組込みハードウェアで効率的に実施できる。

【００９２】
さらに、本明細書で開示する実施形態に関連して説明した様々な例示的な論理ブロック、モジュール、回路、及びアルゴリズム・ステップは、電子ハードウェア、コンピュータ・ソフトウェア、または両方の組合せとして実装できることを、当業者は諒解されよう。ハードウェアとソフトウェアのこの互換性を明確に示すために、様々な例示的な構成要素、ブロック、モジュール、回路、及びステップを、上記では概してそれらの機能に関して説明した。そのような機能をハードウェアとして実装するか、ソフトウェアとして実装するかは、特定の適用例及び全体的なシステムに課される設計制約に依存する。

【００９３】
実施形態についての説明は、例示的なものであり、特許請求の範囲を限定するものではない。したがって、本教示は、他のタイプの装置、ならびに多くの代替形態、修正形態、及び変更形態に容易に適用できることが当業者には明らかであろう。

以下に本件出願当初の特許請求の範囲に記載された発明を付記する。

［１］　トークンを認証するためのベリファイア上で動作可能な方法であって、
トークンから認証要求及びトークン識別子を受信することと、
ネットワーク化されたヘルパーから前記トークンに対応する複数のパズルを取得することと、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くことと、
前記パズル秘密に基づいて検証鍵を生成することと、
前記検証鍵に基づいて質問メッセージを生成することと、
前記パズル識別子及び前記質問メッセージを前記トークンに送信することと、
を備える方法。

［２］　前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信すること、
をさらに備える［１］に記載の方法。

［３］　前記検証鍵を記憶することと、
前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けることと、
をさらに備える［１］に記載の方法。

［４］　前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得することをさらに備える［１］に記載の方法。

［５］　パズルがパズル識別子とパズル秘密とを含む符号化メッセージである［１］に記載の方法。

［６］　前記受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するために前記受信したパズルのうちのサブ

セット複数を解くことであって、前記検証鍵が前記パズル秘密の順序セットにも基づく、解くことと、
パズル秘密の前記順序セットに対応する前記パズル識別子の順序セットを前記質問メッセージとともに前記トークンに送信することと、
をさらに備える［１］に記載の方法。
［７］　前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記トークンによって維持されるカウンタにローカル・カウンタを同期させることであって、事後認証質問メッセージが現在のカウンタ値にも基づく、同期させることと、
をさらに備える［１］に記載の方法。
［８］　前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信すること、
をさらに備える［７］に記載の方法。
［９］　タイムスタンプを生成するためのタイマを維持することであって、事後認証質問メッセージが現在のタイムスタンプにも基づく、維持することをさらに備える［１］に記載の方法。
［１０］　前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することをさらに備える［９］に記載の方法。
［１１］　前記選択されたパズルを解くことが、前記パズルを復号するための鍵を発見するためにブルート・フォース・アタックを実行することを含む［１］に記載の方法。
［１２］　前記質問メッセージがメッセージ認証コードである［１］に記載の方法。
［１３］　擬似ランダム・ナンスを生成することと、
前記トークンに対応するより多数のパズルの中から前記複数のパズルを取得する際に使用すべき前記ヘルパーに前記擬似ランダム・ナンスを送信することと、
をさらに備える［１］に記載の方法。
［１４］　トークンを認証するための検証デバイスであって、
ネットワークに対する高帯域幅を有する第１の通信インターフェースと、
トークンと通信するための低帯域幅を有する第２の通信インターフェースと、
前記第１の通信インターフェース及び前記第２の通信インターフェースに結合された処理回路であって、
前記第２の通信インターフェースを介してトークンから認証要求及びトークン識別子を受信すること、
前記第１の通信インターフェースを介してヘルパーから前記トークンに対応する複数のパズルを取得すること、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くこと、
前記パズル秘密に基づいて検証鍵を生成すること、
前記検証鍵に基づいて質問メッセージを生成すること、ならびに
前記第２の通信インターフェースを介して前記パズル識別子及び前記質問メッセージを前記トークンに送信すること
を行うように構成された処理回路と、
を備える検証デバイス。
［１５］　前記処理回路が、
前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵を知っていることを前記応答メッ

セージが首尾よく証明するならば、前記検証デバイスが前記トークンを認証する、受信することを行うようにさらに構成された［１４］に記載の検証デバイス。
［１６］　前記処理回路に結合された、前記検証鍵を記憶するための記憶デバイスをさらに備え、
前記処理回路が、前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けるようにさらに構成された［１４］に記載の検証デバイス。
［１７］　前記処理回路が、前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得するようにさらに構成された［１４］に記載の検証デバイス。
［１８］　前記処理回路が、
前記受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するために前記受信したパズルのうちのサブセット複数を解くことであって、前記検証鍵が前記パズル秘密の順序セットにも基づく、解くことと、
パズル秘密の前記順序セットに対応する前記パズル識別子の順序セットを前記質問メッセージとともに前記トークンに送信することと、
を行うようにさらに構成された［１４］に記載の検証デバイス。
［１９］　前記処理回路が、
前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記トークンによって維持されるカウンタにローカル・カウンタを同期させることであって、事後認証質問メッセージが現在のカウンタ値にも基づく、同期させることと、
前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のカウンタ値を独立して知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することと、
を行うようにさらに構成された［１４］に記載の検証デバイス。
［２０］　前記処理回路が、
タイムスタンプを生成するタイマを維持することであって、事後認証質問メッセージが現在のタイムスタンプにも基づく、維持することと、
前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵及び前記現在のタイムスタンプを独立して知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することと、
を行うようにさらに構成された［１４］に記載の検証デバイス。
［２１］　トークンを認証するための検証デバイスであって、
トークンから認証要求及びトークン識別子を受信するための手段と、
ネットワーク化されたヘルパーから前記トークンに対応する複数のパズルを取得するための手段と、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くための手段と、
前記パズル秘密に基づいて検証鍵を生成するための手段と、
前記検証鍵に基づいて質問メッセージを生成するための手段と、
前記パズル識別子及び前記質問メッセージを前記トークンに送信するための手段と、
を備える検証デバイス。
［２２］　前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信するための手段であって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信するための手段をさらに備える［２１］に記載の検証デバイス。
［２３］　前記検証鍵を記憶するための手段と、
前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するため

に前記検証鍵を前記トークン識別子に関連付けるための手段と、
をさらに備える［２１］に記載の検証デバイス。
［２４］　前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得するための手段をさらに備える［２１］に記載の検証デバイス。
［２５］　前記受信したパズルのうちのサブセット複数を擬似ランダムに選択し、複数のパズル秘密及び対応するパズル識別子を取得するために前記受信したパズルのうちのサブセット複数を解くための手段であって、前記検証鍵が前記パズル秘密の順序セットにも基づく、解くための手段と、
パズル秘密の前記順序セットに対応する前記パズル識別子の順序セットを前記質問メッセージとともに前記トークンに送信するための手段と、
をさらに備える［２１］に記載の検証デバイス。
［２６］　前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記トークンによって維持されるカウンタにローカル・カウンタを同期させるための手段であって、事後認証質問メッセージが現在のカウンタ値にも基づく、同期させるための手段と、
前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを証明する応答メッセージを前記トークンから受信するための手段であって、前記トークンが前記検証鍵及び前記現在のカウンタ値を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信するための手段と、
をさらに備える［２１］に記載の検証デバイス。
［２７］　タイムスタンプを生成するタイマを維持するための手段であって、事後認証質問メッセージが現在のタイムスタンプにも基づく、維持するための手段と、
前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを証明する応答メッセージを前記トークンから受信するための手段であって、前記トークンが前記検証鍵及び前記現在のタイムスタンプを知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信するための手段と、
をさらに備える［２１］に記載の検証デバイス。
［２８］　第１の通信インターフェースを介してトークンから認証要求及びトークン識別子を受信することと、
第２の通信インターフェースを介してヘルパーから前記トークンに対応する複数のパズルを取得することと、
前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くことと、
前記パズル秘密に基づいて検証鍵を生成することと、
前記検証鍵に基づいて質問メッセージを生成することと、
前記第１の通信インターフェースを介して前記パズル識別子及び前記質問メッセージを前記トークンに送信することと、
を行うように構成された処理回路
を備える処理デバイス。
［２９］　前記処理回路に結合された、前記検証鍵を記憶するための記憶デバイス
をさらに備え、
前記処理回路が、前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けるようにさらに構成された［２８］に記載の処理デバイス。
［３０］　前記処理回路が、前記トークンに関連する前記ヘルパーのヘルパー・アドレスを取得するようにさらに構成された［２８］に記載の処理デバイス。
［３１］　ベリファイアに対してトークンを認証するための１つまたは複数の命令であって、プロセッサによって実行されたとき、前記プロセッサに、
ネットワーク化されたヘルパーから前記トークンに対応する複数のパズルを取得することと、

前記パズルのうちの１つを擬似ランダムに選択し、パズル秘密及び対応するパズル識別子を取得するために前記パズルのうちの１つを解くことと、
前記パズル秘密に基づいて検証鍵を生成することと、
前記検証鍵に基づいて質問メッセージを生成することと、
前記パズル識別子及び前記質問メッセージを前記トークンに送信することと、
を行わせる１つまたは複数の命令を有する機械可読媒体。
［３２］　プロセッサによって実行されたとき、前記プロセッサに、
前記トークンが前記検証鍵を知っていることを証明する応答メッセージを前記トークンから受信することであって、前記トークンが前記検証鍵を知っていることを前記応答メッセージが首尾よく証明するならば、前記ベリファイアが前記トークンを認証する、受信することをさらに行わせる１つまたは複数の命令を有する［３１］に記載の機械可読媒体。
［３３］　プロセッサによって実行されたとき、前記プロセッサに、
前記検証鍵を記憶することと、
前記トークンと前記ベリファイアとの間の事後認証における対称鍵として使用するために前記検証鍵を前記トークン識別子に関連付けることと、
をさらに行わせる１つまたは複数の命令を有する［３１］に記載の機械可読媒体。
［３４］　ベリファイアに対してトークンを認証するための前記トークン上で動作可能な方法であって、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングすることと、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始することと、
第１の検証鍵に基づく１つまたは複数のパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記１つまたは複数のパズル識別子によって識別される１つまたは複数のパズルに関連する１つまたは複数のパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信した１つまたは複数のパズル識別子、及び前記秘密鍵に基づいて前記１つまたは複数のパズル秘密を独立して取得することと、
前記１つまたは複数のパズル秘密に基づいて第２の検証鍵を生成することと、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証することと、
を備える方法。
［３５］　前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けることをさらに備える［３４］に記載の方法。
［３６］　前記第２の検証鍵に基づいて、前記トークンが前記第１の検証鍵を知っていることを示す、前記ベリファイアへの応答メッセージを生成することをさらに備える［３４］に記載の方法。
［３７］　前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給することをさらに備える［３４］に記載の方法。
［３８］　前記ベリファイアから複数の順序付きパズル識別子を受信することであって、前記第１の検証鍵が、前記複数の順序付きパズル識別子に関連する対応する複数の順序付きパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記複数の順序付きパズル秘密を取得することであって、前記第１の検証鍵及び前記第２の検証鍵が前記複数の順序付きパズル秘密にも基づく、取得することと、
をさらに備える［３４］に記載の方法。

［３９］　前記トークンと前記ベリファイアとの間の認証が実行されるたびにトラッキングするために前記ベリファイアによって維持されるカウンタにローカル・カウンタを同期させることであって、後続の受信したメッセージ認証コードが現在のカウンタ値にも基づく、同期させること、
をさらに備える［３４］に記載の方法。
［４０］　前記トークンが前記検証鍵と前記現在のカウンタ値とを知っていることを証明する応答メッセージを前記ベリファイアに送信すること、
をさらに備える［３９］に記載の方法。
［４１］　タイムスタンプを生成するためのタイマを維持することであって、後続の受信したメッセージ認証コードが現在のタイムスタンプにも基づく、維持すること、
をさらに備える［３４］に記載の方法。
［４２］　前記トークンが前記検証鍵と前記現在のタイムスタンプとを知っていることを証明する応答メッセージを前記ベリファイアに送信することをさらに備える［４１］に記載の方法。
［４３］　異なるベリファイアを用いて複数のセキュアな検証鍵を確立することと、
前記検証鍵を記憶し、前記トークンと前記異なるベリファイアとの間の対称鍵認証として使用するために前記検証鍵の各々を対応するベリファイアに関連付けることと、
をさらに備える［３４］に記載の方法。
［４４］　ベリファイアと通信するための低帯域幅を有する第１の通信インターフェースと、
前記第１の通信インターフェースに結合された処理回路であって、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムを受信すること、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始すること、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信すること、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得すること、
前記パズル秘密に基づいて第２の検証鍵を生成すること、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証すること、
を行うように構成された処理回路と、
を備えるトークン。
［４５］　前記処理回路に結合され、前記第２の検証鍵を記憶するように構成された記憶デバイスであって、前記処理回路が、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けるように構成された、記憶デバイス
をさらに備える［４４］に記載のトークン。
［４６］　前記処理回路が、
異なるベリファイアを用いて複数のセキュアな検証鍵を確立することと、
前記検証鍵を記憶し、前記トークンと前記異なるベリファイアとの間の対称鍵認証として使用するために前記検証鍵の各々を対応するベリファイアに関連付けることと
を行うようにさらに構成された［４４］に記載のトークン。
［４７］　前記処理回路が、
前記第２の検証鍵に基づいて、前記トークンが前記第１の検証鍵を知っていることを示す、前記ベリファイアへの応答メッセージを生成すること
を行うようにさらに構成された［４４］に記載のトークン。
［４８］　前記処理回路が、

前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給すること
を行うようにさらに構成された［４４］に記載のトークン。
［４９］　ベリファイアで認証するためのトークンであって、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングするための手段と、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始するための手段と、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信するための手段であって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信するための手段と、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得するための手段と、
前記パズル秘密に基づいて第２の検証鍵を生成するための手段と、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証するための手段と、
を備えるトークン。
［５０］　前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けるための手段をさらに備える［４９］に記載のトークン。
［５１］　前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給するための手段をさらに備える［４９］に記載のトークン。
［５２］　秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムを受信することと、
前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始することと、
第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信することと、
前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得することと、
前記パズル秘密に基づいて第２の検証鍵を生成することと、
前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証することと、
を行うように構成された処理回路
を備える処理デバイス。
［５３］　前記処理回路が、
前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けること
を行うようにさらに構成された［５２］に記載の処理デバイス。
［５４］　前記処理回路が、
前記第２の検証鍵に基づいて、前記トークンが前記第１の検証鍵を知っていることを示す、前記ベリファイアへの応答メッセージを生成することを行うようにさらに構成された［５２］に記載の処理デバイス。
［５５］　ベリファイアに対してトークンを認証するための１つまたは複数の命令であって、プロセッサによって実行されたとき、前記プロセッサに、
秘密鍵、トークン識別子、及び１つまたは複数のパズル生成アルゴリズムをトークンにプロビジョニングすることと、

前記トークン識別子をベリファイアに供給することによって前記トークンの認証を開始することと、

第１の検証鍵に基づくパズル識別子とメッセージ認証コードとを含む質問メッセージを前記ベリファイアから受信することであって、前記第１の検証鍵が、前記パズル識別子によって識別されるパズルに関連する第１のパズル秘密の関数である、受信することと、

前記１つまたは複数のパズル生成アルゴリズム、前記受信したパズル識別子、及び前記秘密鍵に基づいて前記第１のパズル秘密を独立して取得することと、

前記パズル秘密に基づいて第２の検証鍵を生成することと、

前記第１の検証鍵と前記第２の検証鍵が同じであるかどうか判断するために前記受信したメッセージ認証コードを検証することと、

を行わせる１つまたは複数の命令を有する機械可読媒体。

［５６］　プロセッサによって実行されたとき、前記プロセッサに、

前記第２の検証鍵を記憶し、前記第１の検証鍵と前記第２の検証鍵が同じである場合、前記第２の検証鍵を前記ベリファイアに関連付けることをさらに行わせる１つまたは複数の命令を有する［５５］に記載の機械可読媒体。

［５７］　プロセッサによって実行されたとき、前記プロセッサに、

前記トークンに関連する複数のパズルを記憶しているヘルパーのヘルパー・アドレスを前記トークンから前記ベリファイアに供給することをさらに行わせる１つまたは複数の命令を有する［５５］に記載の機械可読媒体。

【図１】

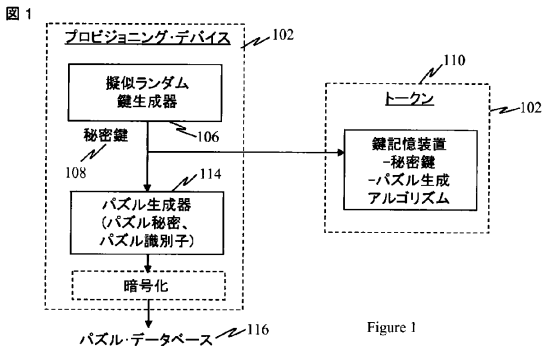

Figure 1

【図２】

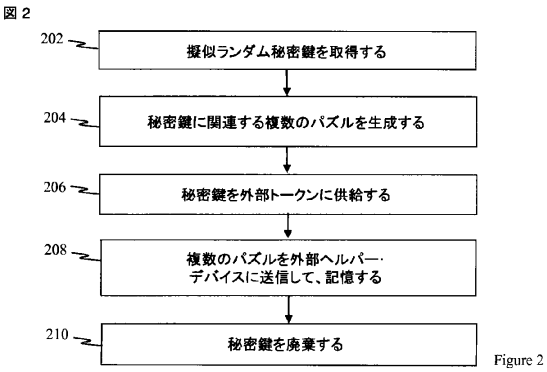

Figure 2

【図３】

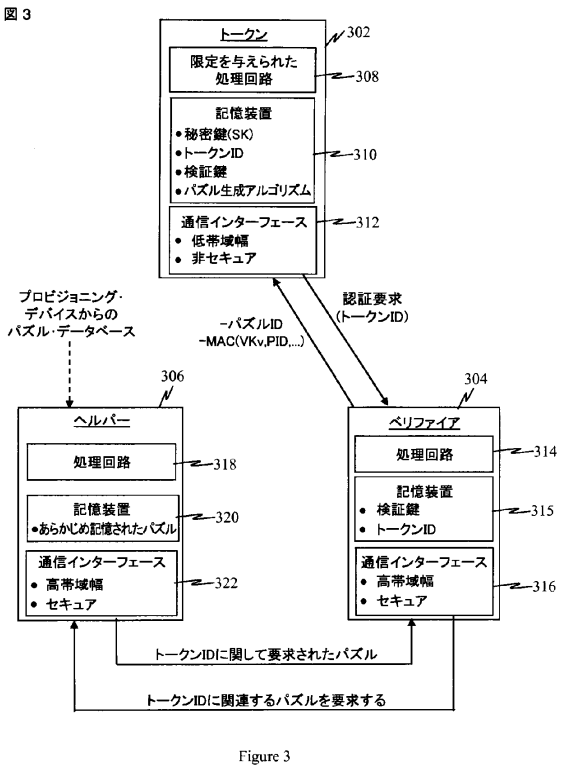

Figure 3

【図４】

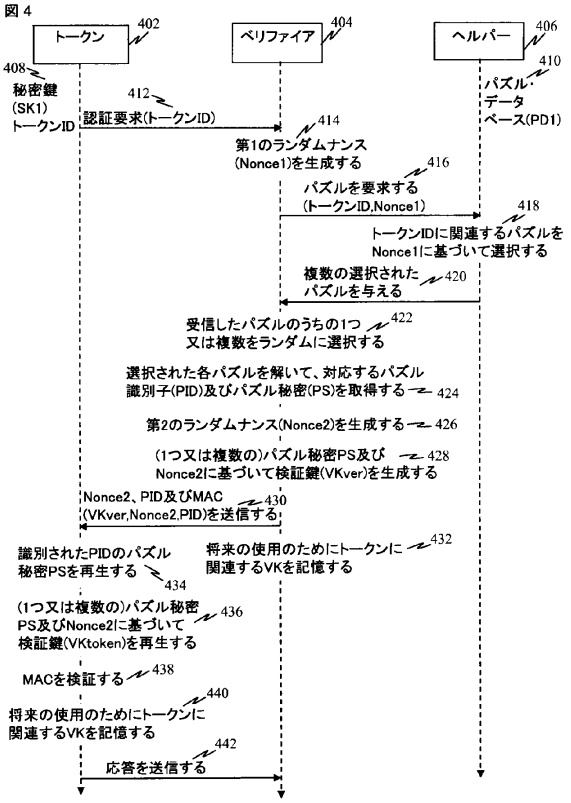

Figure 4

【図５】

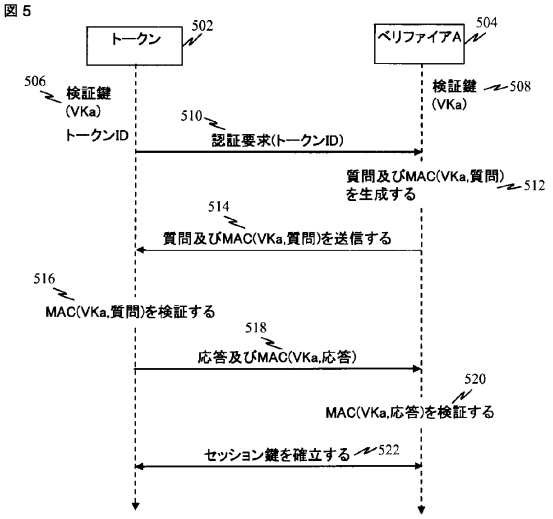

Figure 5

【図６】

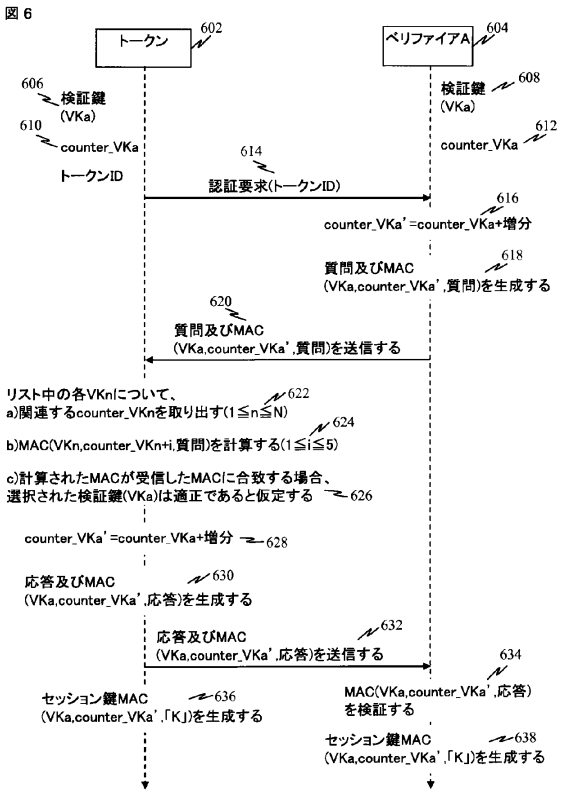

Figure 6

【図７】

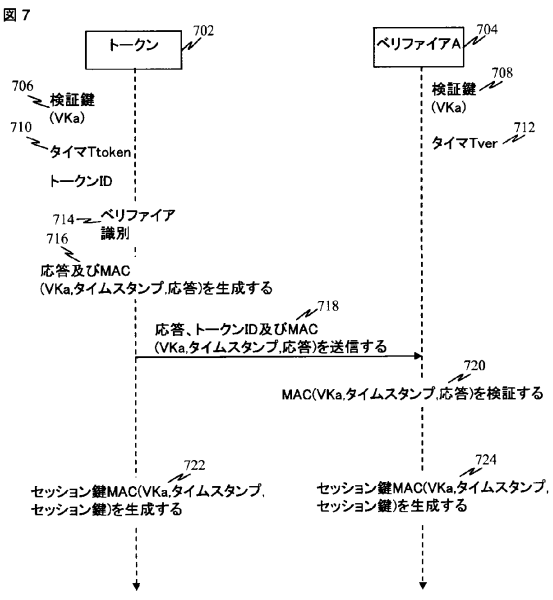

Figure 7

【図８】

図 8

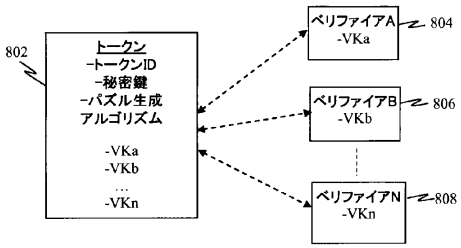

Figure 8

【図９】

図 9

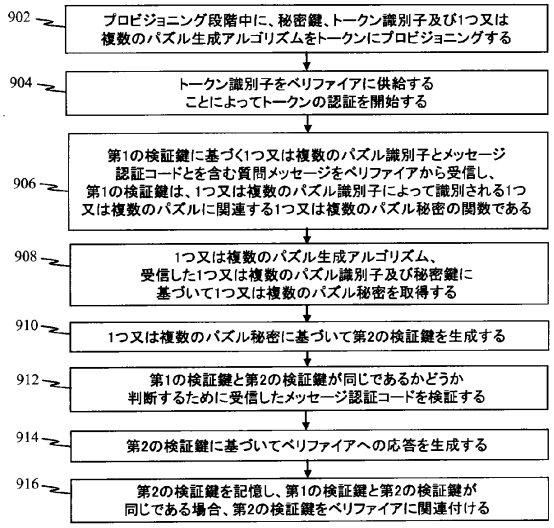

Figure 9

【図１０】

図 10

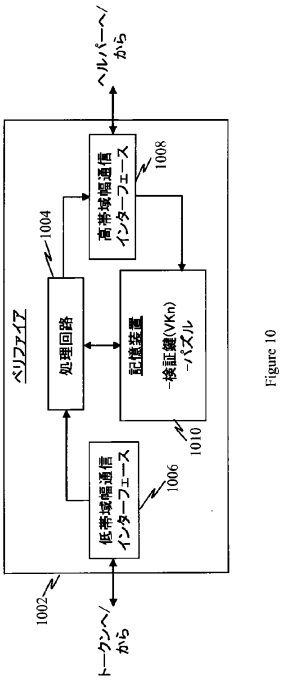

Figure 10

【図１１】

図 11

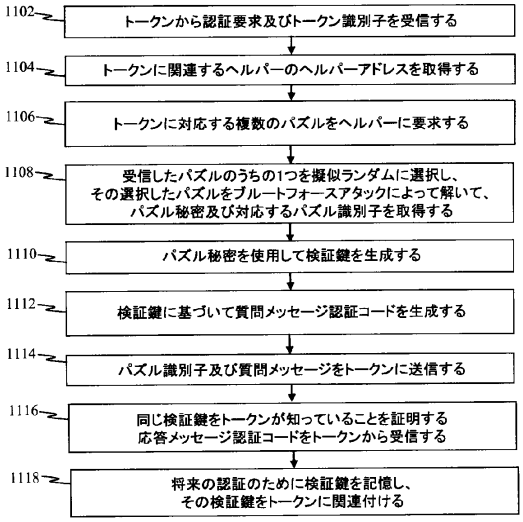

Figure 11

【図１２】

図 12

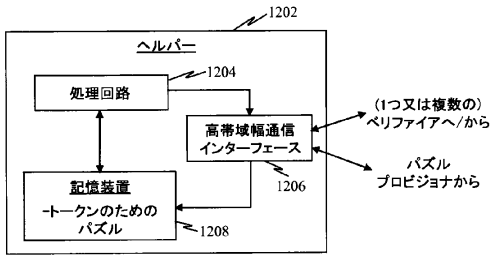

Figure 12

【図１３】

図 13

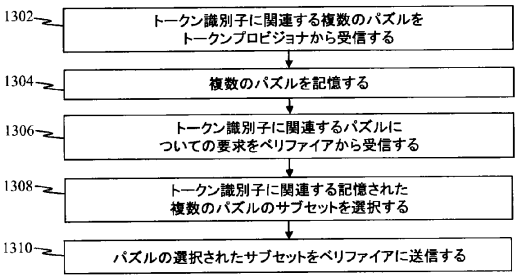

Figure 13

フロントページの続き

(74)代理人 100075672
弁理士 峰 隆司
(74)代理人 100095441
弁理士 白根 俊郎
(74)代理人 100084618
弁理士 村松 貞男
(74)代理人 100103034
弁理士 野河 信久
(74)代理人 100119976
弁理士 幸長 保次郎
(74)代理人 100153051
弁理士 河野 直樹
(74)代理人 100140176
弁理士 砂川 克
(74)代理人 100101812
弁理士 勝村 紘
(74)代理人 100124394
弁理士 佐藤 立志
(74)代理人 100112807
弁理士 岡田 貴志
(74)代理人 100111073
弁理士 堀内 美保子
(74)代理人 100134290
弁理士 竹内 将訓
(74)代理人 100127144
弁理士 市原 卓三
(74)代理人 100141933
弁理士 山下 元
(72)発明者 ローズ、グレゴリー・ゴードン
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５
(72)発明者 ガントマン、アレクサンダー
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５
(72)発明者 ウィガーズ・デ・ブリーズ、ミリアム
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５
(72)発明者 パッドン、マイケル
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５
(72)発明者 ホークス、フィリップ・マイケル
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５

審査官 松平 英

(56)参考文献 特開２００１－３５２３２３（ＪＰ，Ａ）
特開２００７－１９５１５５（ＪＰ，Ａ）

特表２００３－５２０４６７（ＪＰ，Ａ）
特表２０１０－５０６４８０（ＪＰ，Ａ）
国際公開第２００８／０４１０５２（WO，Ａ１）
米国特許出願公開第２００３／０２１７２６９（ＵＳ，Ａ１）
米国特許出願公開第２００７／０２５５０６０（ＵＳ，Ａ１）
Ronald L. Rivest et al, Time-lock puzzles and timed-release crypto, [online], １９９６年, ［平成２５年１月１０日検索］, インターネット<URL: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.5709>
Feng, W et al, Design and implementation of network puzzles, INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, ２００５年　3月, Vol.4, p.2372-2382

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０９Ｃ　　１／００
Ｈ０４Ｌ　　９／００
Ｇ０６Ｆ　２１／２０
Ｇ０６Ｆ　２１／２４