(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2016-526342
(P2016-526342A)
(43)公表日　平成28年9月1日(2016.9.1)

(51)Int.Cl.		F I			テーマコード（参考）
H04L 9/32	(2006.01)	H04L	9/00	675C	5J104

審査請求 未請求　予備審査請求 未請求　（全 91 頁）

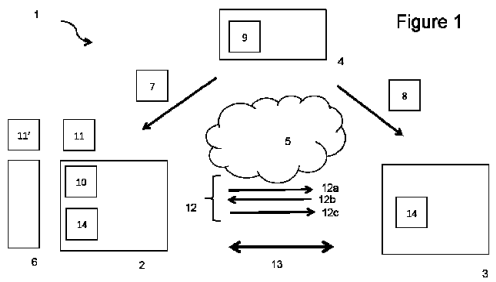
(21)出願番号	特願2016-516248 (P2016-516248)
(86)(22)出願日	平成26年5月30日 (2014.5.30)
(85)翻訳文提出日	平成28年1月26日 (2016.1.26)
(86)国際出願番号	PCT/GB2014/051666
(87)国際公開番号	WO2014/191768
(87)国際公開日	平成26年12月4日 (2014.12.4)
(31)優先権主張番号	1309702.7
(32)優先日	平成25年5月30日 (2013.5.30)
(33)優先権主張国	英国 (GB)

(71)出願人　514187730
ミラクル　リミテッド
イギリス国　イーシー２エイ　３エイワイ　ロンドン，　リビングトン　ストリート　８１
８１　Ｒｉｖｉｎｇｔｏｎ　Ｓｔｒｅｅｔ，　Ｌｏｎｄｏｎ　ＥＣ２Ａ　３ＡＹ　Ｕｎｉｔｅｄ　ｋｉｎｇｄｏｍ
(74)代理人　230115864
弁護士　永島　孝明
(74)代理人　100149168
弁理士　若山　俊輔

最終頁に続く

(54)【発明の名称】ペアリングを使用した多因子ゼロ知識認証

(57)【要約】
第１のエンティティが完全なシークレットを所有していることを、第２のエンティティに完全なシークレットを送信することなく第２のエンティティに証明することによって、第２のエンティティに対して第１のエンティティ自体を認証するための第１のエンティティでの方法であって、方法は、第１のエンティティで、ユーザーからの入力、少なくとも１つの第１の因子及び少なくとも１つの第２の因子に分割された完全なシークレットを受信することであって、入力が完全なシークレットの第２の因子に関係する、受信することと、第１のエンティティで、少なくとも１つの第１の因子及び入力から完全なシークレットを再構築することと、第１のエンティティで、再構築された完全なシークレットを使用して計算を実行し、第２のエンティティに計算の結果を送信することを含み、結果は第２のエンティティでのペアリング計算に対する入力を提供する。第２のエンティティは、クライアントがシークレットを所有しているかどうかを判断するためにペアリング計算を実行する。第１のエンティティはクライアントであってよく、第２のエンティティ



【特許請求の範囲】
【請求項１】
第１のエンティティが完全なシークレットを所有していることを、前記完全なシークレットを第２のエンティティに送信することなく前記第２のエンティティに証明することによって、前記第１のエンティティでの前記第１のエンティティ自体の前記第２のエンティティに対する認証方法であって、
前記第１のエンティティでユーザーからの入力を受信することであって、前記完全なシークレットが少なくとも第１の因子及び第２の因子に分割され、前記入力が前記シークレットの前記第２の因子に関係する、受信することと、
前記第１のエンティティで少なくとも前記第１の因子及び前記入力から前記完全なシークレットを再構築することと、
前記第１のエンティティで前記再構築された完全なシークレットを使用して計算を実行し、前記計算の前記結果を前記第２のエンティティに送信することであって、前記結果が前記第２のエンティティでペアリング計算に対する入力を提供する、計算を実行し、前記計算の前記結果を送信することと、
を含む方法。
【請求項２】
前記完全なシークレットが、前記第１の因子及び前記第２の因子を備えた２つの因子に分割され、前記入力が前記第２の因子の予想値を含み、前記完全なシークレットが前記第１の因子及び前記入力から再構築される、請求項１に記載の方法。
【請求項３】
前記シークレットが代数曲線上の点に相当し、前記ペアリング計算が前記代数曲線上のペアリングを備える、請求項２に記載の方法。
【請求項４】
前記第１のエンティティで前記第２のエンティティからチャレンジを受信することをさらに含み、前記取り戻されたシークレットを使用する前記計算が、前記第２のエンティティに送信される前記結果を得るために前記チャレンジも使用する、請求項３に記載の方法。
【請求項５】
前記第１のエンティティでの前記計算が、前記代数曲線上で別の点を得るために、前記第１のエンティティで、前記シークレットに相当する点、または前記シークレットに相当する少なくとも１つの点から導出される点を前記代数曲線上で乗算することをさらに含み、前記計算の前記結果を前記第２のエンティティに送信することが、その別の点の前記座標を送信することを含む、請求項３または４に記載の方法。
【請求項６】
前記第１のエンティティの前記シークレットが信頼機関によって発行され、前記第１のエンティティの前記アイデンティティ及び信頼機関によって記憶されるマスターシークレットに基づく、請求項３から５のいずれか１つに記載の方法。
【請求項７】
第１のエンティティで乱数値を生成することであって、ｘがｑよりも小さい、生成することと、
前記第１のエンティティで$A = H_1(ID_\alpha)$を計算することであって、上式でID_αは前記第１のエンティティと関連付けられる前記アイデンティティであり、H_1は前記代数曲線上の点に前記アイデンティティをハッシュするハッシュ関数である、計算することと、
前記第１のエンティティで前記曲線上で別の点$U = xA$を計算し、ID_α及びＵを前記第２のエンティティに送信することと、
前記第２のエンティティで生成される乱数値ｙを受信することであって、ｙがｑよりも小さく、前記再構築された完全なシークレットを使用して前記第１のエンティティで前記計算を実行し、前記結果を前記第２のエンティティに送信することが、新しい点Ｖ＝－（

ｘ＋ｙ）（（ｓ－α）Ａ＋αＡ）を計算し、前記第２のエンティティにＶを送信することを含み、前記第１の因子が（ｓ－α）Ａを備え、前記入力が値αを備え、前記クライアントシークレットｓＡが信頼機関によって発行され、前記クライアントアイデンティティに相当する前記点Ａをマスターシークレットｓで乗算することによって前記信頼機関によって得られ、前記ペアリング計算がマッピング

$$G_1 \times G_2 \rightarrow G_T$$

を備え、上式でG_1及びG_2が別個であり、ｑが群G_1、G_2、及びG_Tの位数である、受信すること、

をさらに含む、請求項３に記載の方法。

【請求項８】

前記第１のエンティティで時間許可証を受信することをさらに含み、前記再構築された完全なシークレットを使用する前記計算が、前記第２のエンティティに送信するための前記結果を得るために前記時間許可証も使用し、前記時間許可証が前記信頼機関によって発行され、前記時間許可証が、前記第１のエンティティが前記プロトコルを完了する資格のある期間から導出される、請求項６に記載の方法。

【請求項９】

前記第１のエンティティで時間許可証を受信することをさらに含み、前記再構築されたシークレットを使用する前記計算が、前記第２のエンティティに送信するための前記結果を得るために前記時間許可証も使用し、前記時間許可証が前記信頼機関によって発行され、前記時間許可証が、前記第１のエンティティが前記プロトコルを完了する資格のある期間及び追加データから導出される、請求項６に記載の方法。

【請求項１０】

前記ペアリング計算が前記第１のエンティティを認証するための前記第２のエンティティでの計算の部分を形成し、前記方法が、前記入力が前記第２の因子に一致しなかったことを示す応答を前記第２のエンティティから受信すること、及び前記第１のエンティティに再び認証を試すように要請することをさらに含む、請求項３から９のいずれか１つに記載の方法。

【請求項１１】

前記ペアリング計算が前記第１のエンティティを認証するための前記第２のエンティティでの計算の部分を形成し、前記方法が、前記第１のエンティティが前記シークレットを所有していると前記第２のエンティティが判断する場合に、前記第１のエンティティが前記認証が成功した旨の表示を受信することをさらに含む、請求項３から９のいずれか１つに記載の方法。

【請求項１２】

前記方法がさらに、前記第２のエンティティが前記第１のエンティティを認証できるようにするために前記計算の前記結果を送信することに応えて、前記第２のエンティティから応答を受信し、前記応答のデータ値から前記第１のエンティティで暗号鍵を導出することを含み、前記応答の前記データ値が前記ペアリング計算のペアリングの前記結果から得られる、請求項３から９のいずれか１つに記載の方法。

【請求項１３】

前記ペアリングがマッピング

$$G_1 \times G_2 \rightarrow G_T$$

であり、上式でG_1及びG_2は別個の群であり、ｑは該群G_1、G_2、及びG_Tの位数であり、前記計算の前記結果は前記代数曲線上の点Ｖの前記座標であり、G_1で、前記マッピングは第１の入力として前記点Ｖを採り、第２の入力としてG_2で前記代数曲線上で第２の点を採り、前記第２の点が前記第２のエンティティと関連付けられる固定点に相当する、請求項１２に記載の方法。

【請求項１４】

鍵を導出することが、前記データ値を前記第２のエンティティにとって未知の値乗し、前記結果をハッシュして前記鍵を得ることを含み、前記方法がさらに、前記鍵を使用して

、前記第２のエンティティへのメッセージを暗号化し、前記第２のエンティティから受信されるメッセージを解読することを含む、請求項１２または１３に記載の方法。
【請求項１５】
前記代数曲線が楕円曲線である、請求項３から１４のいずれか１つに記載の方法。
【請求項１６】
前記少なくとも第２の因子がＰＩＮを備える、請求項１から１５のいずれか１つに記載の方法。
【請求項１７】
第１のエンティティがシークレットを所有しているに違いないと、第２のエンティティが前記シークレット自体を受信することなく判断することによる、前記第２のエンティティでの前記第１のエンティティの認証方法であって、
前記第１のエンティティで実施される計算の結果を、前記第２のエンティティで受信することであって、前記計算が、少なくとも前記シークレットの第１の因子及び前記シークレットの第２の因子から前記第１のエンティティで再構築されるシークレットを使用する、受信することと、
前記クライアントが前記シークレットを所有していると判断するために前記第２のエンティティで計算を実行することであって、前記計算が前記第１のエンティティからの前記結果に基づくペアリング計算を備える、計算を実行することと、
を含む、方法。
【請求項１８】
前記シークレットが、前記第１の因子及び前記第２の因子を備える２つの因子に分割され、前記シークレットが前記２つの因子から再構築される、請求項１７に記載の方法。
【請求項１９】
前記第１のエンティティのシークレットが代数曲線上の点に相当し、前記ペアリング計算が前記代数曲線上のペアリングを含む、請求項１７または１８に記載の方法。
【請求項２０】
前記第２のエンティティでの前記計算が、前記クライアントから受信される前記結果に基づく第１のペアリング、及び前記第２のエンティティのシークレットに基づく第２のペアリングを含む、請求項１９に記載の方法。
【請求項２１】
前記第２のエンティティでの前記計算が、前記第１のエンティティが前記シークレットを所有しているか同化を判断するためにペアリングの積を計算することを含み、ペアリングの前記積が前記第１のペアリング及び前記第２のペアリングを含み、前記方法がさらに、前記第２のエンティティが、ペアリングの前記積が所定値に等しいかどうかを判断し、前記積が前記所定値に等しい場合前記第１のエンティティを認証することをさらに含む、請求項２０に記載の方法。
【請求項２２】
前記第１のエンティティの前記シークレットが信頼機関によって発行され、前記第１のエンティティの前記アイデンティティ及び信頼機関によって記憶されるマスターシークレットに基づいており、前記第２のエンティティの前記シークレットも前記信頼機関によって発行され、前記曲線上の固定点及び前記マスターシークレットに基づく、請求項２０または２１に記載の方法。
【請求項２３】
前記第１のエンティティから、前記第１のエンティティの前記アイデンティティ、及び前記代数曲線上の点Ｕの前記座標を受信することと、
前記第２のエンティティでｑよりも小さい乱数値ｙを生成し、前記値を前記第１のエンティティに送信することと
前記第２のエンティティで$A = H_1(ID_\alpha)$を計算することであって、上式でID_αは前記第１のエンティティと関連付けられるアイデンティティであり、H_1は前記アイデンティティを前記代数曲線上の点にハッシュするハッシュ関数であり、計算の結果を前記

エンティティで受信することが、前記第１のエンティティから点Ｖの前記座標を受信することを含み、Ｖ＝－（ｘ＋ｙ）（（ｓ－α）Ａ＋αＡ）であり、ｘがｑよりも小さい別の乱数値であり、前記第１のエンティティの前記シークレットが信頼機関によって発行され、点ｓＡに相当し、ｓがマスターシークレットであり、前記少なくとも第１の因子が前記楕円曲線上の点（ｓ－α）Ａの前記座標を含み、前記第２の因子が値αを含み、前記ペアリング計算を計算することが、ｇ＝ｅ（Ｖ，Ｑ）．ｅ（Ｕ＋ｙＡ，ｓＱ）を得るために第１のペアリングｅ（Ｖ，Ｑ）及び第２のペアリングｅ（Ｕ＋ｙＡ，ｓＱ）を含んだペアリングの積を計算することを含み、Ｑが前記曲線上の固定点であり、ｓＱが前記第２のエンティティのシークレットに相当する前記曲線上の別の点であり、前記第１のペアリング及び前記第２のペアリングがマッピング

$$G_1 \times G_2 \rightarrow G_T$$

であり、上式でG_1及びG_2は別個であり、ｑは群G_1、G_2、及びG_Tの位数であり、前記方法はさらに、ｇ≠１の場合接続を拒否することを含む、計算することと、
をさらに含む、請求項１９に記載の方法。

【請求項２４】

前記第１のエンティティが前記プロトコルを完了する資格がある期間を得て、前記期間及び前記第１のエンティティのアイデンティティを使用して、前記第２のペアリングに対する入力を導出することをさらに含む、請求項２０から２２のいずれか１つに記載の方法。

【請求項２５】

前記第１のエンティティが前記プロトコルを完了する資格がある期間及び追加データを得て、前記期間、前記追加データ、及び前記第１のエンティティのアイデンティティを使用して、前記第２のペアリングに対する入力を導出することをさらに含む、請求項２０から２２のいずれか１つに記載の方法。

【請求項２６】

前記方法がさらに、前記第１のペアリングの結果から導出される値を前記第１のエンティティに送信して、前記第１のエンティティが暗号鍵を導出できるようにすることを含む、請求項２０から２２、２４または２５のいずれか１つに記載の方法。

【請求項２７】

前記ペアリングがタイプ‐３マッピング

$$G_1 \times G_2 \rightarrow G_T$$

であり、上式でG_1及びG_2は別個であり、前記ペアリングは第１の入力として、前記第１のエンティティのシークレットに相当する少なくとも前記点から導出される点の前記代数曲線上の少なくとも１つの乗算によって導出されるG_1での前記代数曲線上の第１の点を採り、第２の入力としてG_2での前記代数曲線上の第２の点を採り、前記第２の点が前記第２のエンティティと関連付けられる固定点に相当する、請求項２６に記載の方法。

【請求項２８】

前記第２のエンティティが、前記第１のエンティティと前記第２のエンティティとの間の追加の通信のために使用される暗号鍵を計算することをさらに含み、前記暗号鍵の前記計算が、前記第１のエンティティのアイデンティティ及び前記第２のエンティティのシークレットに基づいてペアリングを計算することを含む、請求項２０から２７のいずれか１つに記載の方法。

【請求項２９】

前記第２のエンティティで前記暗号鍵を得るための前記ペアリングがタイプ‐３マッピング

$$G_1 \times G_2 \rightarrow G_T$$

であり、上式でG_1及びG_2は別個であり、前記ペアリングは第１の入力として、前記第１のエンティティのアイデンティティに相当する少なくとも１つの点から導出される点の前記代数曲線上の少なくとも１つの乗算によって導出されるG_1での前記代数曲線上の第１の点を採り、第２の入力として前記第２のエンティティのシークレットに相当するG_2

での前記代数曲線上の第２の点を採る、請求項２８に記載の方法。
【請求項３０】
前記第２のエンティティが、それぞれが前記第２のエンティティの前記シークレットの一部を記憶する少なくとも２つの構成要素を備える、請求項２０から２９のいずれか１つに記載の方法。
【請求項３１】
前記第２のエンティティのシークレットが、それぞれ前記２つの構成要素の記憶される前記２つの部分を追加することによって得ることができ、前記第２のペアリングを実行することが各構成要素でペアリングを実行し、入力としてそのそれぞれの部分を採り、次いで前記２つの構成要素で実施される前記ペアリングの積を実施することを含む、請求項３０に記載の方法。
【請求項３２】
前記第１のエンティティが前記シークレットを所有していないと判断することに応えて、前記第２の因子のエラーの前記範囲を決定すること、請求項１９から３１のいずれか１つに記載の方法。
【請求項３３】
前記方法がさらに、前記第２の因子の前記誤差の前記決定された範囲に基づいて再び認証するように前記クライアントに要請するかどうかを判断することを含む、請求項３２に記載の方法。
【請求項３４】
前記第１のエンティティが前記シークレットを所有していないと判断することに応えて、再び認証を試みるように前記第１のエンティティに要請することと、合計エラースコアの前記値を計算することとをさらに含み、前記値が認証試行のたびに可変量増加し、前記量が認証試行ごとの前記第２の因子のエラーの前記範囲に応じて変化する、請求項３２または３３に記載の方法。
【請求項３５】
前記合計エラースコアの前記値が所定の最大エラースコアを超える場合に、前記第１のエンティティに対する前記接続を拒否することをさらに含む、請求項３４に記載の方法。
【請求項３６】
再び認証を試行するように前記第１のエンティティに要請するかどうかの前記判断において、前記第１のエンティティの前記場所、前記アイデンティティ、前記ＩＰアドレス、及び前記認証の時間の中から少なくとも１つを含んだ追加情報を検討することをさらに含む、請求項３４または３５に記載の方法。
【請求項３７】
前記エラーの前記範囲が所定のタイプであると判断することに応えて、前記第１のエンティティがメッセージを送信しようとしていると判断し、前記エラーからそのメッセージを決定すること、請求項３２から３６のいずれか１つに記載の方法。
【請求項３８】
前記少なくとも第２の因子がＰＩＮを備える、請求項１９から３７のいずれか１つに記載の方法。
【請求項３９】
前記代数曲線が楕円曲線である、請求項１９から３８のいずれか１つに記載の方法。
【請求項４０】
前記第２の因子がソフトバイオメトリック
を備える、請求項１から３９のいずれか１つに記載の方法。
【請求項４１】
前記曲線が、（$p^4 - p^2 + 1$）／ｑが素数となるようにＢａｒｒｅｔｏ－Ｎａｅｈｒｉｇ　ＢＮ曲線パラメータを有する楕円曲線であり、上式でｐは素数係数であり、ｑは曲線上のマッピング
$G_1 \times G_2 \rightarrow G_T$

の群G_1、G_2、及びG_Tの位数である、請求項３から１５又は請求項１９から３９のいずれか１つに記載の方法。

【請求項４２】

クライアントが請求項１から１６のいずれか１つに係る方法を実施することと、サーバが請求項１９から３９のいずれか１つの方法を実施することを含む、サーバに対するクライアント認証方法。

【請求項４３】

装置が完全なシークレットを所有していることを、他のエンティティに該シークレットを明らかにすることなく、前記他のエンティティに提供することによって前記装置自体を前記他のエンティティに認証するための前記装置であって、

メモリ上に命令を記憶させる少なくとも１つのメモリと、

前記装置で、ユーザーから入力を受信することであって、前記完全なシークレットが少なくとも第１の因子及び第２の因子に分割され、前記入力が前記シークレットの前記第２の因子に関係する、受信する動作と、

前記装置で、すくなくとも前記第１の因子及び前記第２の因子からシークレットを再構築する動作と、

前記再構築されたシークレットを使用して前記装置で計算を実行し、前記他のエンティティに前記計算の前記結果を送信することであって、前記結果がペアリング計算への入力を提供する、計算を実行し、結果を送信する動作と、

を実行するための前記命令を実行するようにプログラミングされた少なくとも１台のプロセッサと、

を備える装置。

【請求項４４】

完全なシークレットが楕円曲線の点に相当し、前記少なくとも１つのメモリがさらに、前記プロセッサによる実行時に、

ｘがｑよりも小さい乱数値ｘを前記装置で生成する動作と、

前記装置で、$A=H_1(ID_\alpha)$を計算することであって、ID_αが前記装置と関連付けられた前記アイデンティティであり、H_1が前記アイデンティティを楕円曲線上の点にハッシュするハッシュ関数である、計算する動作と、

前記装置で前記楕円曲線上の別の点$U=xA$を計算し、ID_α及びＵを前記他のエンティティに送信する動作と、

前記他のエンティティで生成される、ｙがｑよりも小さい乱数値ｙを受信する動作と、

を実行する命令をさらに備え、

前記再構築された鍵を使用して前記装置で前記計算を実行し、前記他のエンティティに前記結果を送信するための前記命令が、新しい点$V=-(x+y)((s-\alpha)A+\alpha A)$を計算し、前記他のエンティティにＶを送信するための命令を含み、前記第１の因子が楕円曲線上の点$(s-\alpha)A$の前記座標を含み、前記入力が値αを含み、前記シークレットｓＡが信頼機関によって発行され、前記装置と関連付けられた前記アイデンティティに相当する前記点Ａをマスターシークレットｓで乗算することによって前記信頼機関によって得られ、前記ペアリング計算がマッピング

$$G_1 \times G_2 \rightarrow G_T$$

を含み、上式でG_1及びG_2は別個であり、ｑは群G_1、G_2、及びG_Tの位数である、請求項４３に記載の装置。

【請求項４５】

前記再構築されたシークレットを使用する前記計算が、前記他のエンティティに送信するための前記結果を得るために前記装置で受信される時間許可証も使用し、前記時間許可証が信頼機関によって発行され、前記時間許可証が、前記装置が前記プロトコルを完了する資格のある期間から導出される、請求項４３に記載の装置。

【請求項４６】

前記少なくとも１つのメモリが、前記計算の前記結果を前記他のエンティティに送信することに応えて、前記装置で、前記他のエンティティから受信されるメッセージのデータ値から暗号鍵を導出するための命令をさらに含み、前記応答の前記データ値が前記ペアリング計算のペアリングの前記結果から得られる、請求項４３、４４、または４５に記載の装置。

【請求項４７】

装置がシークレットを所有していることを、他のエンティティに該シークレットを明らかにすることなく、前記他のエンティティに証明することによって前記装置自体を前記他のエンティティに認証するための前記装置であって、

入力を受信するための手段であって、前記シークレットが少なくとも第１の因子及び第２の因子に分割され、前記入力が前記シークレットの前記第２の因子に関係する、入力を受信するための手段と、

前記装置で、少なくとも前記第１の因子及び前記第２の因子から前記完全なシークレットを再構築するための手段と、

前記再構築されたシークレットを使用して前記装置で計算を実行し、前記計算の前記結果を前記他のエンティティに送信するための手段であって、前記結果が前記他のエンティティで暗号ペアリングに入力を提供する、計算を実行し、送信するための手段と、

を備える、装置。

【請求項４８】

他のエンティティを、前記エンティティがシークレットを所有しているに違いないと判断することによって、前記シークレット自体を受信することなく認証するための装置であって、

メモリに命令を記憶させる少なくとも１つのメモリと、

前記装置で、前記他のエンティティで実施される計算の結果を受信する動作であって、前記計算が前記シークレットの少なくとも１つの第１の因子及び前記シークレットの少なくとも１つの第２の因子から再構築される前記シークレットを使用する、受信する動作と、

前記他のエンティティが前記シークレットを所有していると判断するために、前記装置で計算を実行する動作であって、前記計算が前記他のエンティティからの前記結果に基づく第１のペアリング、及び前記装置と関連付けられるシークレットに基づく第２のペアリングを含む、計算を実行する動作と、

を実行するための前記命令を実行するようにプログラミングされる少なくとも１台のプロセッサと、

を備える装置。

【請求項４９】

前記他のエンティティの前記シークレットが、楕円曲線上の点に相当し、前記少なくとも１つのメモリがさらに、

前記他のエンティティから、前記他のエンティティの前記アイデンティティ、及び前記代数曲線上の点Ｕの前記座標を受信するための命令と、

前記装置で、ｑより低い乱数値ｙを生成し、前記他のエンティティに送信するための命令と、

前記装置でＡ＝Ｈ$_1$（ＩＤ$_\alpha$）を計算するための命令であって、上式でＩＤαは前記他のエンティティと関連付けられる前記アイデンティティであり、Ｈ$_1$は前記アイデンティティを前記代数曲線上の点にハッシュするハッシュ関数であり、前記他のエンティティで実施される計算の結果を受信することが、前記他のエンティティから点Ｖの前記座標を受信することを含み、Ｖ＝－（ｘ＋ｙ）（（ｓ－α）Ａ＋αＡ）であり、ｘがｑよりも小さい別の乱数値であり、前記他のエンティティの前記シークレットが信頼機関によって発行され、点ｓＡに相当し、ｓがマスターシークレットであり、前記少なくとも第１の因子が前記代数曲線上の点（ｓ－α）Ａの前記座標を含み、前記第２の因子が値αを含み、前記ペアリング計算を計算することが、ｇ＝ｅ（Ｖ，Ｑ）．ｅ（Ｕ＋ｙＡ，ｓＱ）を得るため

に第１のペアリングｅ（Ｖ，Ｑ）及び第２のペアリングｅ（Ｕ＋ｙＡ，ｓＱ）を含んだペアリングの積を計算することを含み、上式でＱが前記代数曲線上の固定点であり、ｓＱが前記装置と関連付けられるシークレットに相当する前記楕円曲線上の別の点であり、前記第１のペアリング及び前記第２のペアリングがマッピング

$$G_1 \times G_2 \rightarrow G_T$$

であり、上式でG_1及びG_2は別個の群であり、ｑは群G_1、G_2、及びG_Tの位数であり、前記少なくとも１つのメモリがさらに、ｇ≠１の場合に前記接続を拒否するための命令を備える、請求項４８に記載の装置。

【請求項５０】

前記少なくとも１つのメモリがさらに、前記他のエンティティが暗号鍵を導出するために、前記他のエンティティから受信される前記結果に基づく入力を採る、前記ペアリング計算のペアリングの前記結果から導出される値を前記他のエンティティに送信するための命令、及び前記装置で別のペアリングから鍵を導出し、前記装置と関連付けられる前記シークレットに相当する入力を採るための命令を含む、請求項４８または４９に記載の装置。

【請求項５１】

前記装置が、それぞれが命令を記憶する前記少なくとも１つのメモリの内の別個のメモリを備え、それぞれが前記命令を実行するための前記１台または複数のプロセッサの内の別個のプロセッサを備える２つの構成要素を備え、前記装置と関連付けられる前記シークレットが２つの部分を含み、前記第２のペアリングを実行することが各構成要素でペアリングを実行し、前記装置と関連付けられる前記シークレットのそのそれぞれの部分を入力として採り、前記２つのペアリングの積を実施することを含む、請求項４８から５０のいずれか１つに記載の装置。

【請求項５２】

クライアントデバイス自体を認証側エンティティに対して、前記クライアントデバイスがシークレットを所有していることを、前記エンティティに前記シークレットを送信することなく証明することによって認証するための前記クライアントデバイスのためのコンピュータプログラムであって、前記クライアントデバイスの１台または複数のプロセッサによって実行されるときに、前記１台または複数のプロセッサに請求項１から１６のいずれか１つに記載の方法を実行させる命令を備える、コンピュータプログラム。

【請求項５３】

前記認証装置がシークレット自体を受信することなく、クライアントデバイスが前記シークレットを所有しているに違いないと判断することによって、前記クライアントデバイスを認証するためのコンピューティングシステム用のコンピュータプログラムであって、前記コンピューティングシステムの少なくとも１台のプロセッサによって実行されるときに、前記少なくとも１台のプロセッサに請求項１７から３９のいずれか１つに記載の前記方法を実施させる命令を備える、コンピュータプログラム。

【請求項５４】

第１のエンティティの第２のエンティティに対する多因子ゼロ知識証明認証を実行するコンピュータによって実装される方法。

【請求項５５】

前記方法が前記認証を実行するためにペアリングベースの暗号法を使用する、請求項５４に記載のコンピュータによって実装される方法。

【請求項５６】

前記第１のエンティティが別個のエンティティによって発行され、楕円曲線上の点に相当し、前記第１のエンティティで少なくとも１つの第１の因子及び少なくとも１つの第２の因子に分割されたシークレットと関連付けられ、前記方法がさらに、前記第１のエンティティに記憶される少なくとも前記第１の因子、及びユーザーから受信される少なくとも前記第２の因子から前記完全なシークレットを、前記第１のエンティティで再構築することと、前記再構築された完全なシークレットを使用して計算を実行し、前記第２のエンテ

ィティに前記計算の結果を送信することと、前記第１のエンティティを認証するために計算で前記第１のエンティティから受信される前記結果を前記第２のエンティティで使用することとを含み、前記第１のエンティティでの前記計算が非ペアリング計算であり、前記第２のエンティティでの前記計算がペアリング計算である、請求項５５に記載の方法。

【請求項５７】

前記第２のエンティティが、前記別個のエンティティによって発行される独自のシークレットも記憶し、前記第２のエンティティのシークレットが前記楕円曲線上の点に相当し、前記方法がさらに、

前記第１のエンティティから受信される前記結果をその入力の１つとして採る第１のペアリング、及び前記第２のエンティティのシークレットに相当する前記曲線上の前記点をその入力の１つとして採る第２のペアリングを含むペアリングの積を前記第２のエンティティで実行することと、

前記ペアリングの前記積の前記結果に基づいて前記第１のエンティティを認証するかどうかを判断することと、

を含む、請求項５６に記載の方法。

【請求項５８】

請求項５４から５７のいずれか１つに記載の方法を使用してクライアントを認証することを含み、前記第１のエンティティの前記認証中に計算されるペアリングの前記結果からセッション暗号鍵を導出することをさらに含む、認証鍵共有実行方法。

【請求項５９】

前記第２のエンティティが、前記ペアリングの前記結果から導出される値を前記第１のエンティティに送信することと、前記第１のエンティティが前記受信された値から前記鍵を導出することとをさらに含む、請求項５８に記載の方法。

【請求項６０】

コンピュータによって実装される、Ｂａｒｒｅｔｏ－Ｎａｅｈｒｉｇ（ＢＮ）楕円曲線を使用するペアリングベースの鍵共有実施方法であって、前記ＢＮ曲線パラメータが、（p^4-p^2+1）／ｑが素数となるように選択され、上式でｐは素数係数であり、ｑは前記楕円曲線上のマッピング

$$G_1 \times G_2 \rightarrow G_T$$

の群G_1、G_2、及びG_Tの前記位数である、方法。

【請求項６１】

装置の１台または複数のプロセッサによって実行されるときに、前記プロセッサに、他のエンティティと安全に通信するための鍵を導出するために装置で鍵共有プロトコルのステップを実施させる命令をコンピュータ可読媒体に記憶させるコンピュータ可読媒体であって、前記鍵共有プロトコルがＢａｒｒｅｔｏ－Ｎａｅｈｒｉｇ（ＢＮ）楕円曲線を使用し、前記ＢＮ曲線パラメータが、（p^4-p^2+1）／ｑが素数となるように選択され、上式でｐは素数係数であり、ｑは前記楕円曲線上のマッピング

$$G_1 \times G_2 \rightarrow G_T$$

の群G_1、G_2、及びG_Tの前記位数である、コンピュータ可読媒体。

【請求項６２】

装置で、クライアントからデータを受信することであって、前記データが前記装置に対して前記クライアントを認証する認証試行の部分として提供され前記クライアントが複数の因子に分割された実際のシークレットと関連付けられ、前記データが前記実際のシークレットの再構築を試行するために前記クライアントで使用される複数の因子から導出された、受信することと、

前記データを導出するために使用される前記複数の因子の内の前記因子の内の１つが、前記実際のシークレットの前記複数の因子の内の対応する因子に対して異なると判断し、前記差異の前記範囲を決定することと、

前記差異と関連付けられるエラー値を決定することと、

前記クライアントのいくつかの認証試行について結合されたエラー値の値が所定の最大

エラー値を超えないと判断することに応えて、前記クライアントに再び認証を試行するように要請することと、
を含む、コンピュータによって実装される方法。
【請求項６３】
メモリの上に命令を記憶させる少なくとも１つのメモリと、
装置に対してクライアントを認証するための認証試行の部分として前記クライアントからデータを受信することであって、前記クライアントが複数の因子に分割されたシークレットと関連付けられ、前記受信されたデータが前記実際のシークレットの再構築を試行するために前記クライアントで使用される複数の因子から導出された、データを受信する動作と、
前記装置で、前記データを導出するために使用される前記複数の因子の前記因子の内の１つが前記実際のシークレットの前記複数の因子の対応する因子に対して異なっていると判断し、前記差異の前記範囲を決定する動作と、
前記差異と関連付けられるエラー値を決定する動作と、
前記クライアントのいくつかの認証試行について結合されたエラー値が所定の最大エラー値を超えないと判断することに応えて、前記クライアントに再び認証を試行するように要請する動作と、
を実行するための前記命令を実行するようにプログラミングされる少なくとも１台のプロセッサと、
を備える装置。
【請求項６４】
１台または複数のプロセッサによって実行されるときに、前記プロセッサに請求項６２に記載の方法を実施させる命令を備えるコンピュータプログラム。
【請求項６５】
クライアント及びサーバに対する、暗号化または認証を実行するための、コンピュータによって実装される、シークレット発行方法であって、前記クライアントがシークレットｓＡを発行され、前記サーバがシークレットｓＱを発行され、Ａが前記クライアントと関連付けられる、代数曲線上の点であり、Ｑが前記代数曲線上の固定点であり、ｓがマスターシークレットであり、ｓＡ及びｓＱが、前記代数曲線上で前記点Ａ及び前記点Ｑのそれぞれを前記マスターシークレットｓで乗算することによって得られる点を表す、方法。
【請求項６６】
１台または複数のプロセッサによって実行されるときに、前記プロセッサに請求項６５に記載の前記方法を実施させる命令を備えるコンピュータプログラム。
【請求項６７】
コンピュータによって実装される、第１のエンティティでの第２のエンティティへの情報の、前記第２のエンティティに前記実際の情報を送信することのない通信方法であって、前記情報が少なくとも１つの第１の因子及び少なくとも１つの第２の因子に分割されるシークレットと関連付けられ、前記方法が、
少なくとも前記第１の因子及びダミーの第２の因子をダミーシークレットに結合することであって、前記ダミーの第２の因子は、前記第２のエンティティに通信される前記情報に相当する値分、前記第２の因子に対して異なる、結合することと、
前記再構築されたダミーシークレットを使用して前記第１のエンティティで計算を実行し、前記第２のエンティティに前記計算の前記結果を送信することであって、前記結果が前記第２のエンティティで計算に使用されて、前記ダミーの第２の因子と前記第２の因子の前記差異を決定する、計算を実行し、結果を送信することと、
を含む、方法。
【請求項６８】
前記計算が楕円曲線上の暗号ペアリングを含み、前記シークレットが曲線上の点である、請求項６７に記載の方法。
【請求項６９】

前記情報がメッセージの一部を形成し、前記メッセージの各部が少なくとも２つの因子に分割されるシークレットと関連付けられ、前記方法がさらに、少なくとも１つの第１の因子及び少なくとも１つのダミー因子を、前記メッセージの各部のためのダミーシークレットに結合し、前記第２のエンティティが前記メッセージをアセンブルするために、前記ダミーシークレットに基づく計算の前記結果を前記第２のエンティティに送信することをさらに含む、請求項６７または６８に記載の方法。

【請求項７０】

第２のエンティティでの第１のエンティティからの、コンピュータによって実装される、情報入手方法であって、前記情報が、すくなくとも１つの第１の因子及び少なくとも１つの第２の因子に分割されるシークレットと関連付けられ、前記方法が、

前記第２のエンティティで、前記第１のエンティティで実施される計算の結果を受信することであって、前記計算が、前記第１のエンティティで前記シークレットの少なくとも１つの第１の因子及びダミーの第２の因子から作成されるダミーシークレットを使用し、前記ダミーの第２の因子が、前記第２のエンティティで得られる前記情報に対応する値分、前記第２の因子に対して異なる、受信することと、

前記ダミーの第２の因子と前記第２の因子の前記差異を決定するために前記第２のエンティティで計算を実行することであって、前記計算が前記第１のエンティティからの前記結果を使用する、実施することと、

を含む、方法。

【請求項７１】

前記第１のエンティティから複数の計算の前記結果を受信することをさらに含み、それぞれの計算がメッセージの異なる部分と関連付けられるダミーシークレットに基づき、各ダミーシークレットが実際の第２の因子に対して異なるダミーの第２の因子から導出される、請求項７０に記載の方法。

【請求項７２】

前記メッセージがクレジットカード情報を含む、請求項６７から７１のいずれか１つに記載の方法。

【請求項７３】

コンピュータによって実装される、第２のエンティティに対する第１のエンティティの認証方法であって、

前記第１のエンティティが、複数の因子に分割される第１のエンティティのシークレットと関連付けられ、前記方法が、前記複数の因子から再構築される前記第１のエンティティのシークレットから導出されるデータに基づいて、バイリニアマッピングを使用して計算を実行し、前記第１のエンティティが前記計算の結果に基づいて前記第１のエンティティのシークレットを所有していたに違いないかどうかを、前記第２のエンティティで判断することをさらに含む、方法。

【請求項７４】

前記シークレットの前記複数の因子から前記第１のエンティティで前記シークレットを再構築することと、

バイリニアマッピングに対する入力を導出するために前記再構築されたシークレットを使用することであって、前記計算を実行することが、前記第２のエンティティと関連付けられるデータに基づいて第１の入力及び第２の有力として前記入力を採る第１のバイリニアマッピングを計算すること、及び前記第１のエンティティと関連付けられるデータに基づく第１の入力、及び第２のエンティティのシークレットから導出される第２の入力データを採る第２のバイリニアマッピングを計算することを含み、前記第１のエンティティのシークレットが前記第１のエンティティと関連付けられる前記データから構築され、前記第２のエンティティのシークレットが前記第２のエンティティと関連付けられる前記データから構築され、したがって前記マッピングの前記結果、つまり前記マッピングの前記結果から導出される値が、前記第２のエンティティによって、前記第１のエンティティがそのシークレットを所有しているかどうかを判断するために使用できる、使用することと、

をさらに含む、請求項７３に記載の方法。
【請求項７５】
前記計算が、前記第１のバイリニアマッピング及び前記第２のバイリニアマッピングに相当する２つのペアリングを実現するマルチペアリングを実行し、前記マルチペアリングの前記結果が所定値に等しいかどうかを判断することを含む、請求項７４に記載の方法。
【請求項７６】
メモリの上に命令を記憶させる少なくとも１つのメモリと、
クライアントを認証するための前記命令を実行するようにプログラミングされる少なくとも１台のプロセッサであって、前記クライアントが複数の因子に分割されたシークレットと関連付けられ、前記命令が、実行時、前記少なくとも１台のプロセッサに、前記複数の因子から再構築される前記シークレットから導出されるデータに基づき、バイリニアマッピングを使用して計算を実行させ、前記計算に基づき、前記第１のエンティティが前記シークレットを所有していたに違いないと判断させる、少なくとも１台のプロセッサと、
を備える装置。
【請求項７７】
装置の１台または複数のプロセッサによって実行されるときに、前記装置の前記１台または複数のプロセッサに、クライアントを認証するための認証プロセスを実施させる命令を、非一過性のコンピュータ可読媒体に記憶させる非一過性のコンピュータ可読媒体であって、前記クライアントは複数の因子に分割されたシークレットに関連付けられ、前記命令が、実行時、前記１台または複数のプロセッサに、前記複数の因子から再構築される前記シークレットから導出されるデータに基づき、バイリニアマッピングを使用して計算を実行させ、前記計算に基づき、前記第１のエンティティが前記シークレットを所有していたに違いないかどうかを判断させる、非一過性のコンピュータ可読媒体。

【発明の詳細な説明】
【技術分野】
【０００１】
本願は、セキュリティに関する。より詳細には本願は第２のエンティティに対する第１のエンティティの認証に関するが、これに限らない。
【背景技術】
【０００２】
公開鍵暗号法の標準的な方法を使用してサーバが自体をクライアントに認証することは、基本的には解決済みの問題である。公開鍵基盤（ＰＫＩ）は、同様にしてこの機能性を可能にするセキュアソケットレイヤ（ＳＳＬ）プロトコルをサポートしている。ＰＫＩでの単一障害点、したがって攻撃の焦点は、認証機関である。ただし、このエンティティは一般的にオフラインであり、十分に防御されており、簡単に手が届かない。クライアントがクライアント自体をサーバに対して認証することの方がはるかに問題である。最も単純且つ最も一般的な機構はユーザー名／パスワードである。まったく満足の行くものではないが、クライアントでの唯一の義務はパスワードを生成し、覚えることである－そして、現実には、クライアントがより大きなシークレットを保護するほど十分に洗練されている、またはうまく組織化されていることは期待できない。ただし、機構としてのユーザー名／パスワードは、崩壊しつつある。いわゆるサーバに対するゼロデイアタックは、一般的にパスワードに関係する情報を含んだファイルを取り戻し、パスワードが十分に高エントロピーではない限り、パスワードは検出される。一般的に適用されるパッチは、クライアントが長く複雑で覚えにくいパスワードを採用することを主張することである。これは、本来は、認証サーバのハッキングが成功した（ますますありそうな）場合に、パスワードを保護するためにクライアントに課される副次的な防御手段である。理想的な世界では、サーバはクライアントがクライアント自体を認証するために行うことができる試行の数を制限できるため、クライアントは低エントロピーパスワードを使用できるはずであることに留意されたい。

【０００３】
提案されている代替策は多因子認証を採用することである。最も単純なケースでは、クライアントはトークンとパスワードの両方を所有していることを証明しなければならない。銀行は係る方法を採用する第一線にいるが、トークンはつねにある種の物理デバイスである。暗号法の厄介な解決の鍵は、今日までソフトウェアで完全に二因子認証を実装するために完全に満足の行く手段が発見されていない点である。
【０００４】
本特許明細書は、多様な考え方及び機能だけではなく、それらの創作的な表現も説明している。したがって、本特許文書の開示の一部は、著作権に対する請求がなされている資料を含み、これによって告知される。（合衆国法律集１７、４０１に準じ）Ｃｅｒｔｉｖｏｘ　Ｌｉｍｉｔｅｄ。著作権保護に対する請求は、本特許明細書に示され、説明される本発明の実施形態と関連付けられるすべての保護可能な表現に対して行われる。
【０００５】
特許文書または特許開示は特許商標庁の特許ファイルまたは記録に表示されているので、著作権者は、だれかによる該特許文書または該特許開示のファクシミリの複製に対して異議を唱えないが、何であれすべての他の著作権を留保する。したがって、何であれいかなる著作権下の明示的または暗示的なライセンスも認められない。
【０００６】
本発明の一態様に従って、第１のエンティティが完全なシークレットを所有していることを、第２のエンティティに完全なシークレットを送信することなく、第２のエンティティに証明することによって、第１のエンティティが第１のエンティティ自体を第２のエンティティに認証する方法が提供され、方法は、第１のエンティティでユーザーから入力を受信することであって、完全なシークレットが少なくとも第１の因子及び第２の因子に分割されており、入力がシークレットの第２の因子に関係する、受信することと、第１のエンティティにおいて、少なくとも第１の因子及び入力から完全なシークレットを再構築することと、再構築された完全なシークレットを使用して第１のエンティティで計算を実行し、計算の結果を第２のエンティティに送信することであって、結果が第２のエンティティでペアリング計算に入力を提供する、計算を実行し計算の結果を送信することとを含む。
【０００７】
結果は、第１のエンティティがシークレットを所有しているかどうかを判断するために第２のエンティティによって使用されるために第２のエンティティに送信されてよい。第２のエンティティは、第１のエンティティが鍵を所有しているかどうかを判断するために計算を実行し、ペアリング計算はその計算の部分を形成してよい。
【０００８】
方法は、コンピュータによって実装される方法であってよい。第１のエンティティはクライアントであってよく、第２のエンティティはバックエンドサーバであってよい。クライアントは、コンピューティング装置のブラウザで実行されるプログラムであってよい。入力はユーザーから受信されてよい。ユーザーは人またはコンピューティング装置であってよい。
【０００９】
結果的に、クライアントがクライアントのシークレットをサーバに送信しなくても、サーバはクライアントを認証できる。さらに、ペアリング計算に係る大量の処理をサーバで実行できる。
【００１０】
シークレットは、当該第１の因子及び当該第２の因子を含んだ２つの因子に分割された可能性があり、入力は第２の因子の予想値を含むことがあり、シークレットは第１の因子及び第２の因子の予想値から再構築される。
【００１１】
シークレットは代数曲線上の点に相当し、ペアリング計算は代数曲線での暗号ペアリング

を含んでよい。代数曲線は楕円曲線であってよい。代わりに、代数曲線は超楕円曲線となるだろう。

【００１２】

方法はさらに、第１のエンティティで第２のエンティティからのチャレンジを受け取ることを含んでよく、取り戻されたシークレットを使用する計算は、第２のエンティティに送信される結果を得るためにチャレンジを使用してもよい。

【００１３】

第１のエンティティでの計算は、第１のエンティティで、シークレットに相当する点からまたはシークレットに相当する少なくとも１つの点から導出される点から代数曲線上の点の座標を計算することを含んでよく、計算の結果を第２のエンティティに送信することは、該点の座標を送信することを含んでよい。

【００１４】

第１のエンティティでの計算は、代数曲線上で別の点を得るために、第１のエンティティでシークレットに相当する点またはシークレットに相当する少なくとも１つの点から導出される点を代数曲線上で乗算することを含んでよく、第２のエンティティに計算の結果を送信することは、その別の点の座標を送信することを含んでよい。

【００１５】

第１のエンティティのシークレットは、信頼機関によって発行された可能性があり、第１のエンティティのアイデンティティ、及び信頼機関によって記憶されているマスターシークレットに基づくことがある。信頼機関は独立したエンティティであってよい。また、信頼機関は第２のエンティティに別個のシークレットを発行した可能性があり、第２のエンティティは第１のエンティティを認証するために該シークレットを必要とすることがある。第２のエンティティは、第１のエンティティを認証するためにその計算で第２のエンティティのシークレットを使用してよい。

【００１６】

方法はさらに第１のエンティティにおいて、ｘがｑより小さい乱数値ｘを生成することと、第１のエンティティでＡ＝Ｈ$_1$（ＩＤ$_\alpha$）を計算することであって、上式でＩＤ$_\alpha$は第１のエンティティと関連付けられるアイデンティティであり、Ｈ$_1$は代数曲線上の点にアイデンティティをハッシュするハッシュ関数である計算することと、第１のエンティティで曲線上の別の点Ｕ＝ｘＡを計算し、第２のエンティティにＩＤ$_\alpha$及びＵを送信することと、第２のエンティティで生成される乱数値ｙを受信することを含んでよく、ｙはｑよりも小さく、再構築された鍵を使用して第１のエンティティで該計算を実行し、第２のエンティティに結果を送信することは、新しい点Ｖ＝－（ｘ＋ｙ）（（ｓ－α）Ａ＋αＡ）を計算し、第２のエンティティにＶを送信することを含む。第１の因子は点（ｓ－α）Ａの座標に相当し、入力は値αを含んでよく、クライアントシークレットは信頼機関によって発行されてよく、クライアントアイデンティティに相当する点Ａをマスターシークレットｓで乗算することによって、信頼機関によって得られてよい。ペアリング計算はマッピングＧ$_1$×Ｇ$_2$→Ｇ$_T$を含んでよく、上式でＧ$_1$及びＧ$_2$は別個であり、ｑは群Ｇ$_1$、Ｇ$_2$、及びＧ$_T$の位数である。

【００１７】

方法はさらに、第１エンティティで時間許可証を受信することを含んでよく、再構築されたシークレットを使用する計算も、第２のエンティティに送信するための結果を得るために時間許可証を使用してよい。時間許可証は信頼機関によって発行されてよく、時間許可証は、第１のエンティティがプロトコルを完了する資格がある期間から導出されてよい。

【００１８】

代わりに、方法はさらに、第１のエンティティで時間許可証を受信することを含んでよく、再構築されたシークレットを使用する計算も、第２のエンティティに送信するための結果を得るために時間許可証を使用してよい。時間許可証は信頼機関によって発行されてよく、時間許可証は、第１のエンティティがプロトコルを完了する資格がある期間及び追

加データから導出されてよい。

【００１９】

ペアリング計算は、第１のエンティティを認証するための第２のエンティティでの計算の部分を形成してよく、方法はさらに、第２のエンティティから、入力が第２の因子に一致しなかったことを示す応答を受信し、再び認証を試みるように第１のエンティティに要請することを含んでよい。

【００２０】

ペアリング計算は、第１のエンティティを認証するための第２のエンティティでの計算の部分を形成してよく、方法はさらに、第２エンティティが、第１のエンティティがシークレットを所有していると判断することに応えて、第１のエンティティが、認証が成功した旨の表示を受信することをさらに含んでよい。

【００２１】

方法は、さらに第２のエンティティから応答を受信し、第１のエンティティで該応答のデータ値から暗号鍵を導出することを含み、応答のデータ値は当該ペアリング計算のペアリングの結果から得られる。

【００２２】

ペアリング計算はマッピング$G_1 \times G_2 \rightarrow G_T$を含んでよく、上式で$G_1$及び$G_2$は別個の群であり、ｑは群$G_1$、$G_2$、及び$G_T$の位数であり、計算の結果は代数曲線上の点Ｖの座標であってよく、G_1において、マッピングは第１の入力として点Ｖを採り、第２の入力としてG_2の代数曲線上で第２の点を採ってよく、第２の点は第２のエンティティと関連付けられる固定点に相当する。

【００２３】

鍵を導出することは、データ値を第２のエンティティにとって未知の値乗し、結果をハッシュして鍵を得ることを含んでよく、方法はさらに、鍵を使用して第２のエンティティへのメッセージを暗号化し、第２のエンティティから受信されるメッセージを解読することを含んでよい。

【００２４】

方法はさらに、第１のエンティティが完全なシークレットから第２の因子を抽出して、第１の因子及び第２の因子を作成することを含んでよい。

【００２５】

第２の因子は暗証番号（ＰＩＮ）を含んでよい。第２の因子は、代わりにパスワードまたはソフトバイオメトリックを含んでよい。

【００２６】

本発明の別の態様に従って、クライアントデバイスがシークレットを所持していることを、エンティティにシークレットを送信することなくエンティティに証明することによって認証側エンティティに対してクライアントデバイス自体を認証するためのクライアントデバイス用のコンピュータプログラムも提供され、コンピュータプログラムは、クライアントデバイスの少なくとも１台のプロセッサによって実行されるときに、該少なくとも１台のプロセッサに上述された方法を実行させる命令を含む。

【００２７】

コンピュータプログラムは、クライアントデバイスのブラウザの一部であってよい、またはクライアントデバイスのブラウザで実行されてよい。認証側エンティティはコンピューティングシステムであってよい。コンピューティングシステムは認証側サーバであってよい。

【００２８】

本発明の別の態様に従って、コンピュータプログラムを記憶する非一過性の有形のコンピュータ可読媒体も提供される。

【００２９】

本発明の別の態様に従って、第１のエンティティがシークレットを所有しているに違いないと、第２のエンティティがシークレット自体を受信することなく判断することによっ

て第２のエンティティに対して第１のエンティティを認証する方法も提供され。方法は、第１のエンティティで実行される計算の結果を第２のエンティティで受信することであって、計算がシークレットの少なくとも１つの第１の因子及び１つの第２の因子から第１のエンティティで再構築されるシークレットを使用する、受信することと、第１のエンティティがシークレットを所有していると判断するために第２のエンティティで計算を実行することであって、計算が第１のエンティティから受信される結果に基づくペアリング計算を含む、実行すすることとを含む。

【００３０】

方法はコンピュータによって実装される方法であってよい。第１のエンティティはクライアントであってよい。第２のエンティティはバックエンドサーバであってよい。クライアントはデバイスのブラウザで実行されるプログラムであってよい。

【００３１】

シークレットは、当該第１の因子及び当該第２の因子を含んだ２つの因子に分割された可能性があり、シークレットは当該２つの因子から再構築されてよい。

【００３２】

第１のエンティティのシークレットは代数曲線上の点に相当してよく、ペアリング計算は代数曲線上のペアリングを含んでよい。曲線は楕円曲線または超楕円曲線であってよい。

【００３３】

ペアリング計算は、入力として直接的に第１のエンティティから結果を取り出してよい。代わりに、ペアリング計算は、ペアリング計算に対する入力を導出するために第１のエンティティからの結果を使用してよい。

【００３４】

第１のエンティティが第１のエンティティのシークレットを所有していると判断するための計算は、第２のエンティティと関連付けられる第２のエンティティのシークレットを必要とすることもある。

【００３５】

第２のエンティティでの計算は、第１のエンティティがシークレットを所有しているかどうかを判断するためにペアリングの積を計算することを含んでよく、ペアリングの積はクライアントから受信される結果に基づく第１のペアリング及び第２のエンティティのシークレットに基づく第２のペアリングを含む。ペアリングの積はマルチペアリングとして計算されてよい。代わりに、ペアリングの積は、次いで互いに乗算される２つの別々のペアリングとして計算されてよい。

【００３６】

第２のエンティティは、ペアリングの積が所定の値に等しいかどうかを判断してよく、積が所定の値に等しい場合、第１のエンティティを認証してよい。

【００３７】

第１のエンティティのシークレットは信頼機関によって発行された可能性があり、第１のエンティティのアイデンティティ及び信頼機関によって記憶されているマスターシークレットに基づいてよく、第２のエンティティのシークレットも信頼機関によって発行された可能性があり、曲線上の固定点及び該マスターシークレットから導出されてよい。

【００３８】

方法はさらに、第１のエンティティから、第１のエンティティのアイデンティティ及び代数曲線上の点Ｕの座標を受信することと、第２のエンティティでｑよりも小さい乱数値ｙを生成し、第１のエンティティに送信することと、（ID_{α}）が第１のエンティティと関連付けられるアイデンティティであり、H_1が代数曲線上の点にアイデンティティをハッシュするハッシュ関数である、第２のエンティティで$A = H_1(ID_{\alpha})$を計算することをさらに含んでよい。第１のエンティティから計算の結果を受信することは、第１のエンティティから点Ｖの座標を受信することを含んでよく、$V = -(x+y)((s-\alpha)A + \alpha A)$であり、ｘはｑより小さい別の乱数値であり、第１のエンティティのシークレ

ットは信頼機関によって発行され、点ｓＡに相当し、ｓはマスターシークレットであり、該少なくとも第１の因子は代数曲線上の点（ｓ－α）Ａに相当し、第２の因子は値αを含む。当該ペアリング計算を計算することは、第１のペアリングｅ（Ｖ，Ｑ）及び第２のペアリングｅ（Ｕ＋ｙＡ，ｓＱ）を含んだペアリングの積を計算して、ｇ＝ｅ（Ｖ，Ｑ）．ｅ（Ｕ＋ｙＡ，ｓＱ）を得ることを含んでよく、上式で、Ｑは代数曲線上の固定点であり、ｓＱは第２のエンティティのシークレットに相当する曲線上の別の点であり、第１のペアリング及び第２のペアリングはマッピング$G_1 \times G_2 \rightarrow G_T$であり、上式で$G_1$及び$G_2$は別個であり、ｑは群$G_1$、$G_2$、及び$G_T$の位数である。方法はさらに、ｇ≠１の場合、接続を拒否することを含んでよい。

【００３９】

方法はさらに、第１のエンティティがプロトコルを完了する資格のある期間を得ることと、期間及び第１のエンティティのアイデンティティを使用して第２のペアリングに対する入力を導出することとを含んでよい。

【００４０】

代わりに、方法はさらに、第１のエンティティがプロトコルを完了する資格がある期間及び追加のデータを得て、期間、追加のデータ、及び第１のエンティティのアイデンティティを使用して第２のペアリングに対する入力を導出することを含んでよい。

【００４１】

方法はさらに、第１のペアリングの結果から導出される値を第１のエンティティに送信して、第１のエンティティが暗号鍵を導出できるようにすることをさらに含んでよい。

【００４２】

ペアリング計算は、タイプ‐３マッピング$G_1 \times G_2 \rightarrow G_T$であるペアリングを含んでよく、上式で$G_1$及び$G_2$は別個であり、ペアリングは第１の入力として、第１のエンティティのシークレットに相当する点または少なくとも第１のエンティティのシークレットに相当する点から導出される点の、代数曲線上での少なくとも１つの乗算によって導出されるG_1の代数曲線上の第１の点を採り、第２の入力として、G_2の代数曲線上の第２の点を採り、第２の点は第２のエンティティと関連付けられる固定点に相当する。

【００４３】

方法はさらに、第２のエンティティが第１のエンティティと第２のエンティティとの間の追加の通信のために使用される暗号鍵を計算することを含んでよく、暗号鍵の計算は第１のエンティティのアイデンティティ及び第２のエンティティのシークレットに基づくペアリングを計算することを含む。ペアリングは、入力として、クライアントのアイデンティティから導出される点及び他のデータを採ってよい。

【００４４】

第２のエンティティで確立される暗号鍵は、第１のエンティティで確立される暗号鍵と同じである。新しい鍵は、エンティティが新しい通信セッションを開始するたびに確立されてよい。

【００４５】

第２のエンティティで暗号鍵を得るためのペアリングは、タイプ‐３マッピング$G_1 \times G_2 \rightarrow G_T$であってよく、上式で$G_1$及び$G_2$は別個であり、ペアリングは第１の入力として、第１のエンティティのアイデンティティに相当する少なくとも少なくとも１つの点から導出される点の代数曲線上での少なくとも１つの乗算によって導出されるG_1の代数曲線上の点を採り、第２の入力として、第２のエンティティのシークレットに相当するG_2の代数曲線上の点を採る。

【００４６】

第２のエンティティは、それぞれが第２のエンティティのシークレットの一部を記憶する少なくとも２つの構成要素を含んでよい。第２のエンティティのシークレットは、該２つの部分を追加することによって得ることができる。第２のペアリングを実行すすることは、各構成要素でペアリングを計算することと、そのそれぞれの部分を入力として採ることと、次いで２つのペアリングの結果の積を計算することを含んでよい。２つの構成要素

の内の１つ、または第２のエンティティの別個の構成要素は、２つのペアリングの積を実行できる。第２のエンティティのシークレットは２つを超える部分に分割されてよく、第２のエンティティはシークレットの部分を記憶するための２つを超える構成要素を含んでよい。

【００４７】

方法は、第１のエンティティがシークレットを所有していないと判断することに応えて、第２の因子の任意のエラーの範囲を決定することを含んでよい。

【００４８】

任意のエラーの範囲は、ｇの値及び方程式ｇ＝（ｅ（Ｕ＋ｙＡ，Ｑ）$^{\delta}$を使用して計算され、上式でδは第２の因子のエラーの範囲である。代わりに、時間許可証が使用される場合、時間許可証はｇ＝ｅ（Ｒ＋ｙＡ，Ｑ）$^{\delta}$を使用して計算されてよい。

【００４９】

方法はさらに、第１のエンティティがシークレットを所有していないと判断することに応えて、第１のエンティティに再び認証を試行するように要請すること、及び計算合計エラースコアの値を含んでよく、値は認証の試行ごとに可変量増加し、量は認証の試行のたびの第２の因子のエラーの範囲に応じて変化する。方法はさらに、合計エラースコアの値が所定の最大エラースコアを超える場合、第１のエンティティへの接続を拒否することを含んでよい。いくつかのタイプのエラーの場合、可変量はゼロであってよい。

【００５０】

方法はさらに、第１のエンティティに再び認証を試行するように要請するかどうかを第２のエンティティで決定するときに、第１のエンティティの場所、第１のエンティティのアイデンティティ、ＩＰアドレス、及び認証の時間の中から少なくとも１つを含んだ追加の情報を検討することを含んでよい。

【００５１】

方法はさらに、エラーが所定のタイプであると判断することに応えて、第１のエンティティがメッセージの送信を試行していると判断すること、及び第２の因子のエラーからそのメッセージを決定することを含んでよい。

【００５２】

第２の因子はＰＩＮを含んでよい。代わりに、第２の因子はパスワードまたはソフトバイオメトリックを含んでよい。

【００５３】

代数曲線は楕円曲線であってよい。曲線は、（ｐ^{4}－ｐ^{2}＋１）／ｑが素数となるようにＢＮ曲線パラメータを有する楕円曲線であってよく、上式でｐは素数係数であり、ｑは曲線上のマッピング$G_1 \times G_2 \rightarrow G_T$の群$G_1$、$G_2$、及び$G_T$の位数である。

【００５４】

本発明の別の態様に従って、認証装置がシークレット自体を受信することなく、クライアントデバイスがシークレットを所有しているに違いないと判断することによって、クライアントデバイスを認証するための認証装置用のコンピュータプログラムも提供され、コンピュータプログラムは、少なくとも１台のプロセッサによって実行されるときに少なくとも１台のプロセッサに上述された第２のエンティティでの方法を実行させる命令を含む。

【００５５】

本発明の別の態様に従って、コンピュータプログラムがその上に記録され、記憶される非一過性のコンピュータ可読媒体が提供されてよい。

【００５６】

本発明の別の態様に従って、クライアントが上記第１のエンティティで実施されるために説明された方法を実施すること、及びサーバが上記第２のエンティティで実施されるために説明された方法を実施することとを含んだ、サーバに対してクライアント認証する方法も提供される。

【００５７】

本発明の別の態様に従って、装置がシークレットを所有していることを、他のエンティティにシークレットを明らかにすることなく、他のエンティティに対して提供することによって、装置自体を別のエンティティに対して認証するための装置も提供され、装置は、命令がその上に記憶される少なくとも１つのメモリ、及び命令を実行して以下の動作、つまりユーザーからの入力を装置で受信する動作であって、シークレットが少なくとも１つの第１の因子及び第２の因子に分割され、入力がシークレットの第２の因子に関係する、受信する動作と、少なくとも第１の因子及び第２の因子から装置でシークレットを再構築する動作と、再構築されたシークレットを使用して装置で計算を実行し、計算の結果を他のエンティティに送信する動作とを実行するようにプログラミングされる１台または複数のプロセッサを含み、結果はペアリング計算に対する入力を提供する。

【００５８】

完全なシークレットは楕円曲線上の点に相当してよく、非一過性のコンピュータ可読媒体はさらに、１台または複数のプロセッサによって実行されるときに、以下、つまりｘがｑより小さい乱数値ｘを装置で生成することと、$A = H_1(ID_\alpha)$を装置で計算することであって、上式でID_αは装置と関連付けられるアイデンティティであり、H_1が楕円曲線上の点にアイデンティティをハッシュするハッシュ関数である、計算することと、装置の楕円曲線で別の点$U = xA$を計算し、他のエンティティにID_α及びＵを送信することと、ｙがｑより小さい他のエンティティで生成される乱数値ｙを受信することとを実行する命令を有してよい。再構築された鍵を使用し、装置で当該計算を実行し、他のエンティティに結果を送信するための命令は、新しい点$V = -(x+y)((s-\alpha)A+\alpha A)$を計算し、他のエンティティにＶを送信するための命令を含んでよく、第１の因子は点$(s-\alpha)A$の座標であり、入力は値αを含み、シークレットｓＡは信頼機関によって発行され、装置と関連付けられるアイデンティティに相当する点をマスターシークレットｓで乗算することによって信頼機関によって得られ、ペアリング計算はマッピング

$$G_1 \times G_2 \rightarrow G_T$$

を含み、上式でG_1及びG_2は別個であり、ｑは群G_1、G_2、及びG_Tの位数である。

【００５９】

再構築されたシークレットを使用する計算は、他のエンティティに送信するための結果を得るために装置で受信される時間許可証を使用してもよく、時間許可証は信頼機関によって発行されてよく、時間許可証は装置がプロトコルを完了する資格がある期間から導出されてよい。

【００６０】

該少なくとも１つのメモリはさらに、計算の結果を他のエンティティに送信することに応えて、他のエンティティから受信されるメッセージのデータ値から暗号鍵を導出するための命令を含んでよく、応答のデータ値は当該ペアリング計算のペアリングの結果から得られる。

【００６１】

本発明の別の態様に従って、装置がシークレットを所有していることを、他のエンティティにシークレットを明らかにすることなく他のエンティティに対して証明することによって、別のエンティティに対して装置自体を認証するための装置も提供され、装置は、装置でユーザーからの入力を受信するための手段であって、シークレットが少なくとも１つの第１の因子及び１つの第２の因子に分割され、入力がシークレットの第２の因子に関係する、受信するための手段と、装置で少なくとも１つの第１の因子及び１つの第２の因子からシークレットを再構築するための手段と、再構築されたシークレットを使用し装置で計算を実行し、計算の結果を他のエンティティに送信するための手段とを含み、結果は他のエンティティでのペアリングに対する入力を提供する。

【００６２】

別のエンティティに対して装置自体を認証するための装置は、クライアントデバイスであってよい。該少なくとも１つのメモリは、クライアントのブラウザメモリの部分を形成するメモリを含んでよい。

【００６３】
本発明の別の態様に従って、エンティティがシークレット自体を受信することなくシークレットを所有しているに違いないと判断することによって、別のエンティティを認証するための装置も提供され、装置は、命令がその上に記憶される少なくとも１つのメモリと、他のエンティティで実行される計算の結果を装置で受信するための命令であって、計算が少なくともシークレットの第１の因子及びシークレットの第２の因子から再構築されるシークレットを使用する、受信するための命令と、他のエンティティがシークレットを所有していると判断するために装置で計算を実行するための命令とを実行するようにプログラミングされた少なくとも１台のプロセッサとを含み、計算は、他のエンティティからの結果に基づく入力を採る第１のペアリング、及び装置と関連付けられるシークレットに基づく第２のペアリングを含む。
【００６４】
装置はコンピューティングシステムであってよい。コンピューティングシステムは認証側サーバであってよい。
【００６５】
他のエンティティと関連付けられるシークレットは代数曲線上の点に相当してよく、非一過性のコンピュータ可読媒体はさらに、他のエンティティから、他のエンティティのアイデンティティ及び代数曲線上の点Ｕの座標を受信するための命令と、ｑより小さい乱数値ｙを生成し、他のエンティティに送信するための命令と、$A = H_1(ID_\alpha)$を計算するための命令であって、上式でID_αが第１のエンティティと関連付けられるアイデンティティであり、H_1が代数曲線上の点にアイデンティティをハッシュするハッシュ関数である計算するための命令とを含んでよい。計算の結果を受信することは、第１のエンティティから点Ｖの座標を受信することを含んでよく、$V = -(x+y)((s-\alpha)A+\alpha A)$であり、ｘはｑより小さい別の乱数値であり、他のエンティティのシークレットは信頼機関によって発行され、点ｓＡに相当し、ｓはマスターシークレットであり、因子は代数曲線上の点（ｓ－α）Ａの座標を含み、第２の因子は値αを含む。さらに、当該ペアリング計算を実行することは、第１のペアリングｅ（Ｖ，Ｑ）及び第２のペアリングｅ（Ｕ＋ｙＡ，ｓＱ）を含んだペアリングの積を計算して、ｇ＝ｅ（Ｖ，Ｑ）．ｅ（Ｕ＋ｙＡ，ｓＱ）を得ることを含んでよく、上式でＱは楕円曲線上の固定点であり、ｓＱは装置のシークレットに相当する楕円曲線上の別の点であり、第１のペアリング及び第２のペアリングはマッピング$G_1 \times G_2 \rightarrow G_T$であり、上式で$G_1$及び$G_2$は別個の群であり、ｑは群$G_1$、$G_2$２、及び$G_T$の位数である。命令は、ｇ≠１の場合に接続を拒否するための命令も含んでよい。
【００６６】
命令はさらに、他のエンティティがプロトコルを完了する資格がある期間を得るための命令、及び期間及び他のエンティティのアイデンティティを使用して第２のペアリングに対する入力を導出するための命令も含んでよい。
【００６７】
少なくとも１つのメモリはさらに、他のエンティティから受信される結果に基づいて入力を採る、ペアリング計算のペアリングの結果から導出される値を、他のエンティティに送信して、他のエンティティが暗号鍵を導出できるようにするための命令と、別のペアリングから装置で鍵を導出し、装置のシークレットに相当する入力を採るための命令とを含んでよい。
【００６８】
装置は、それぞれが該少なくとも１つのメモリの別個のメモリを含み、それぞれがそのそれぞれのメモリに記憶される命令を実行するための該少なくとも１台のプロセッサの別個のプロセッサを有し、それぞれが装置と関連付けられるシークレットの一部を記憶する２つの構成要素を含んでよく、第２のペアリングを実行するための命令は、各構成要素でペアリングを実行するための命令と、シークレットのそのそれぞれの部分を入力として採るための命令と、２つのペアリングの積を実行するための命令とを含む。２つの部分はと

もに追加されると完全なシークレットを形成する曲線上の２つの異なる点に相当する。

【００６９】

命令はさらに、他のエンティティが他のエンティティと関連付けられるシークレットを所有していないと判断することに応えて、第２の因子の任意のエラーの範囲を決定することを含んでよい。

【００７０】

命令はさらに、他のエンティティが他のエンティティと関連付けられるシークレットを所有していないと判断することに応えて、再び認証を試行するように他のエンティティに要請すること、及び認証を試行するたびに可変量、エラーカウントの値を増すことを含んでよく、該量は各認証試行でのエラーの範囲に依存して変わる。

【００７１】

命令はさらに、エラーカウントの値が所定の最大エラー値を超える場合、他のエンティティへの接続を拒否することを含んでよい。

【００７２】

代数曲線は、（p^4-p^2+1）／ｑが素数となるようにＢＮ曲線パラメータを有する楕円曲線であってよく、上式でｐは素数係数であり、ｑは曲線上のマッピング$G_1 \times G_2 \rightarrow G_T$の群$G_1$、$G_2$、及び$G_T$の位数である。

【００７３】

本発明の別の態様に従って、他のエンティティがシークレットを所有しているに違いないと、装置がシークレット自体を受信することなく判断することによって別のエンティティを認証するための装置も提供され、装置は、他のエンティティで実行される計算の結果を装置で受信するための手段であって、計算が、少なくとも１つのシークレットの第１の因子及び少なくとも１つのシークレットの第２の因子から他のエンティティで再構築されるシークレットを使用する、受信するための手段と、他のエンティティがシークレットを所有していると判断するために装置で計算を実行するための手段とを含み、計算は他のエンティティからの結果に基づいて入力を採るペアリング計算を含む。

【００７４】

他のエンティティを認証するための装置は、サーバであってよい。

【００７５】

本発明の別の態様に従って、上述されたようなクライアントデバイス、及び上述されたようなサーバを含んだシステムも提供される。また、システムは、信頼機関を提供する独立したエンティティを含んでもよい。信頼機関はクライアントにシークレットを発行し、サーバに別のシークレットを発行してよい。

【００７６】

本発明の別の態様に従って、第２のエンティティに対する第１のエンティティの多因子ゼロ知識証明認証を実行するコンピュータによって実装される方法も提供される。

【００７７】

方法は、認証を実行するためにペアリングベースの暗号法を使用してよい。

【００７８】

認証の後には、以後の安全なデータの交換を可能にするために認証鍵共有が続いてよい。

【００７９】

第１のエンティティは、楕円曲線上の点に相当し、第１のエンティティにおいて少なくとも１つの第１の因子及び少なくとも１つの第２の因子に分割される、別個のエンティティによって発行されるシークレットと関連付けられてよく、方法はさらに、第１のエンティティが、第１のエンティティに記憶される少なくとも１つの第１の因子、及びユーザーから受信される少なくとも１つの第２の因子から完全なシークレットを再構築し、再構築された完全なシークレットを使用して計算を実行し、計算の結果を第２のエンティティに送信することを含んでよい。第２のエンティティは第１のエンティティを認証するために計算の結果を使用してよい。第１のエンティティでの計算は非ペアリング計算であってよ

く、第２のエンティティでの計算はペアリング計算であってよい。

【００８０】

第２のエンティティは、やはり別のエンティティによって発行される独自のシークレットを記憶してよく、第２のエンティティのシークレットも楕円曲線上の点に相当し、方法はさらに、その入力の１つとして第１のエンティティから受信される結果を採る第１のペアリング、及びその入力の１つとして第２のエンティティのシークレットに相当する曲線上の点を採る第２のペアリングを含んだペアリングの積を第２のエンティティで実行することと、ペアリングの積の結果に基づいて第１のエンティティを認証するかどうかを判断することを含んでよい。

【００８１】

ペアリングの積は、マルチペアリングとして実行されてよい。代わりに、第２のエンティティは最初に第１のペアリング及び第２のペアリングを実行し、次いで２つのペアリングの結果を同時に乗算してよい。

【００８２】

本発明のさらに別の態様に従って、上記に定められた認証方法を使用してクライアントを認証することを含み、認証プロセス中に計算されるペアリングの結果からセッション暗号鍵を導出することをさらに含む、認証鍵共有を実行するコンピュータによって実装される方法も提供される。

【００８３】

方法はさらに、第２のエンティティがペアリングの結果から導出される値を第１のエンティティに送信すること、及び第１のエンティティが受信した値から鍵を導出することを含んでよい。

【００８４】

本発明のさらに別の態様に従って、Ｂａｒｒｅｔｏ－Ｎａｅｈｒｉｇ（ＢＮ）楕円曲線を使用しペアリングベースの鍵共有を実行する、コンピュータによって実装される方法が提供され、ＢＮ曲線パラメータは、（p^4-p^2+1）／ｑが素数となるように選択され、上式でｐは素数係数であり、ｑは楕円曲線上のマッピング$G_1 \times G_2 \rightarrow G_T$の群$G_1$、$G_2$、及び$G_T$の位数である。

【００８５】

さらに、本発明のさらに別の態様に従って、少なくとも１台のプロセッサによって実行されるときに、１台または複数のプロセッサに、別のエンティティと安全に通信するための鍵を導出するために鍵共有プロトコルのステップをエンティティで実施させる命令を含んだコンピュータプログラムが提供され、鍵共有プロトコルはＢａｒｒｅｔｏ－Ｎａｅｈｒｉｇ（ＢＮ）楕円曲線を使用し、ＢＮ曲線パラメータは、（p^4-p^2+1）／ｑが素数となるように選択され、上式でｐは素数係数であり、ｑは楕円曲線上のマッピング$G_1 \times G_2 \rightarrow G_T$の群$G_1$、$G_2$、及び$G_T$の位数である。

【００８６】

本発明のさらに別の態様に従って、上記に定義されるように、コンピュータプログラムがその上に記憶された非一過性のコンピュータ可読媒体も提供される。

【００８７】

本発明のさらに別の態様に従って、上記に定義されるような非一過性のコンピュータ可読媒体、及び非一過性のコンピュータ可読媒体上の命令を実行するための少なくとも１台のプロセッサを含んだ装置も提供される。

【００８８】

本発明の別の態様に従って、装置でクライアントからデータを受信することであって、データが装置にクライアントを認証するための認証試行の部分として提供される、受信することと、受信されたデータと関連付けられたエラーがあることを装置で判断することと、エラーの範囲に基づいてエラー値を決定することと、クライアントのいくつかの認証試行に対する組み合わせられたエラー値の値が所定の最大エラー値を超えないと判断することに応えて、クライアントに再度認証を試行するように要請することとを含んだ、コンピ

ュータによって実装される方法が提供される。

【００８９】

本発明の別の態様に従って、装置においてクライアントからデータを受信することであって、データが装置にクライアントを認証する認証試行の部分として提供され、クライアントが、複数の因子に分割された実際のシークレットと関連付けられ、データが実際のシークレットの再構築を試行するためにクライアントで使用される複数の因子から導出される、受信することと、データを導出するために使用される複数の因子の内の因子の１つが実際のシークレットの複数の因子の内の対応する因子に対して異なると判断し、当該差異の範囲を決定することと、差異と関連付けられたエラー値を決定することと、クライアントのいくつかの認証試行に対して結合されたエラー値の内の値が所定の最大エラー値を超えないと判断することに応えて、クライアントに再び認証を試行するように要請することを含んだ、コンピュータによって実装される方法も提供される。

【００９０】

実際のシークレットの複数の因子は、装置にとって未知であってよい。因子はＰＩＮであってよく、エラー値はＰＩＮの何桁が間違っているのかに依存してよい。

【００９１】

本発明のさらに別の態様に従って、装置の少なくとも１台のプロセッサによって実行されるときに、少なくとも１台のプロセッサにクライアントから受信されるデータを装置で受信させ、該データは装置にクライアントを認証するための認証試行の部分として提供され、クライアントは複数の因子に分割された実際のシークレットと関連付けられ、データは実際のシークレットの再構築を試行するためにクライアントで使用される複数の因子から導出され、少なくとも１台のプロセッサにデータを導出するために使用される複数の因子の内の１つが実際のシークレットの複数の因子の対応する因子に対して異なっていると判断させ、複数の因子の内の因子の当該１つと、対応する因子との差異と関連付けられるエラー値を決定させ、クライアントのいくつかの認証試行に対して結合されたエラー値が所定の最大エラー値を超えないと判断することに応えて、少なくとも１台のプロセッサにクライアントに再び認証を試行するように要請させる命令を含んだコンピュータプログラムも提供される。

【００９２】

本発明の別の態様に従って、命令がその上に記憶される少なくとも１つのメモリと、以下の動作、つまり装置に対してクライアントを認証する認証試行の部分としてクライアントからデータを受信する動作であって、クライアントが複数の因子に分割されたシークレットと関連付けられ、受信されたデータが実際のシークレットの再構築を試みるためにクライアントで使用される複数の因子から導出される、受信する動作と、データを導出するために使用される複数の因子の因子の内の１つが実際のシークレットの複数の因子の対応する因子に対して異なると判断し、差異の範囲を決定する動作と、差異と関連付けられるエラー値を決定する動作と、クライアントのいくつかの認証試行に対して結合されたエラー値が所定の最大エラー値を超えないと判断することに応えて、クライアントに再び認証を試行するように要請する動作を実行するための命令を実行するようにプログラミングされる少なくとも１台のプロセッサとを含んだ装置も提供される。

【００９３】

本発明の別の実施形態に従って、クライアント及びサーバに暗号化または認証を実行するためのシークレットを発行するコンピュータによって実装される方法も提供され、クライアントは、シークレットｓＡを発行され、サーバはシークレットｓＱを発行され、Ａは代数曲線上の、クライアントと関連付けられる点であり、Ｑは代数曲線上の固定点であり、ｓはマスターシークレットであり、ｓＡ及びｓＱは、代数曲線上で、それぞれ点Ａ及び点Ｑのマスターシークレットｓでの乗算によって得られる点を表す。

【００９４】

方法は、クライアント及びサーバを含んだシステムを制御する信頼機関によって実行されてよい。固定点Ｑは、信頼機関によって制御されるシステムのすべてのサーバにとって

同じ点であってよい。マスターシークレットｓは特定のサーバと関連付けられてよく、異なるサーバに対して異なってよい。点Ａは、クライアントのアイデンティティに対応するデータをハッシュすることによって得られてよい。

【００９５】

方法は、コンピュータによって実装される方法であってよい。

【００９６】

本発明の別の態様に従って、１台または複数のプロセッサによって実行されるときに、１台または複数のプロセッサに上記に定義された方法を実行させる命令を含んだ非一過性のコンピュータ可読媒体も提供される。

【００９７】

本発明の別の態様に従って、当該非一過性のコンピュータ可読媒体、及び媒体上の命令を実行するための１台または複数のプロセッサを含んだ信頼機関も提供される。

【００９８】

本発明のさらに別の態様に従って、第２のエンティティに実際の情報を送信することなく、第２のエンティティに情報を通信するための第１のエンティティでのコンピュータによって実装される方法が提供され、情報は少なくとも１つの第１の因子及び１つの第２の因子に分割されるシークレットと関連付けられ、方法は、少なくとも１つの第１の因子及びダミーの第２の因子をダミーシークレットに結合することであって、ダミーの第２の因子は第２のエンティティに通信される情報に対応する値分、第２の因子に対して異なる、結合することと、再構築されたダミーシークレットを使用して第１のエンティティで計算を実行し、計算の結果を第２のエンティティに送信することを含み、結果は、ダミーの第２の因子と第２の因子との間の差異を決定するために第２のエンティティで計算に使用される。

【００９９】

計算は楕円曲線での暗号ペアリングを含んでよく、シークレットは曲線上の点であってよい。

【０１００】

情報はメッセージの部分を形成してよく、メッセージの各部分は少なくとも２つの因子に分割されるシークレットと関連付けられ、方法はさらに、少なくとも１つの第１の因子及びダミー因子を、メッセージの各部分のためにダミーシークレットに結合し、第２のエンティティがメッセージを組み立てるために、ダミーシークレットに基づく計算の結果を第２のエンティティに送信することをさらに含む。

【０１０１】

メッセージはクレジットカード情報を含んでよい。

【０１０２】

本発明の別の態様に従って、少なくとも１台のプロセッサによって実行されるときに、少なくとも１台のプロセッサに上記に定められた情報を通信する方法を実施させる命令を含んだコンピュータプログラムも提供される。コンピュータプログラムは、コンピュータ可読媒体上に記憶されてよい。

【０１０３】

本発明の別の態様に従って、第２のエンティティで第１のエンティティから情報を得るためのコンピュータによって実装される方法も提供され、該情報は少なくとも１つの第１の因子及び１つの第２の因子に分割されるシークレットと関連付けられ、方法は、第１のエンティティで実行される計算の結果を第２のエンティティで受信することであって、計算は少なくともシークレットの第１の因子及びダミーの第２の因子から、第１のエンティティで作成されるダミーシークレットを使用し、ダミーの第２の因子は第２のエンティティで得られる情報に対応する値分、第２の因子に対して異なる、受信することと、ダミーの第２の因子と第２の因子との差異を決定するために第２のエンティティで計算を実行することとを含み、計算は第１のエンティティからの結果を使用する。

【０１０４】

計算は、第１のエンティティからの結果に基づくペアリングを含んでよい。

【０１０５】

方法はさらに、第１のエンティティから複数の計算の結果を受信することであって、各計算はメッセージの異なる部分と関連付けられたダミーシークレットに基づいており、各ダミーシークレットは実際の因子に対して異なるダミー因子から導出される、受信することと、第２のエンティティでメッセージを組み立てることとを含んでよい。

【０１０６】

差異を決定するための計算は、値ｇを得るために計算を実行することと、ｇのために得られる値からδの値を決定し、関係性ｇ＝ｅ（Ｕ＋ｙＡ，Ｑ）$^{\delta}$を使用することを含んでよく、上式でＵ、ｙＡ、及びＱは楕円曲線上の点である。

【０１０７】

差異を計算するための計算は、値ｇを得るために計算を実行することと、ｇのために得られる値からδの値を決定し、関係性ｇ＝ｅ（Ｒ＋ｙＡ，Ｑ）$^{\delta}$を使用することを含んでよく、上式でＲ、ｙＡ及びＱは楕円曲線上の点である。

【０１０８】

メッセージは、金融取引を実行するための情報を含んでよい。メッセージは、クレジットカード番号またはデビットカード番号を含んでよい。

【０１０９】

本発明の別の態様に従って、１台または複数のプロセッサによって実行されるときに、プロセッサに、上記に定められた情報を得る方法を実施させる命令を含んだコンピュータプログラムも提供される。コンピュータプログラムは、コンピュータ可読媒体に記憶されてよい。

【０１１０】

本発明の別の態様に従って、メッセージの部分ごとにシークレットを生成し、メッセージの一部に対応する因子を各因子から抽出して、複数の多因子シークレットを作成するためのコンピュータによって実装される方法も提供される。

【０１１１】

さらに、本発明の別の態様に従って、プロセッサによって実行されるときに、プロセッサに上記に定められた方法を実施させる命令がその上に記憶される非一過性のコンピュータ可読媒体が提供される。

【０１１２】

本発明の別の態様に従って、当該コンピュータ可読媒体、及び命令を実行するための１台または複数のプロセッサを備える信頼機関も提供されてよい。

【０１１３】

本発明の別の態様に従って、第２のエンティティに対して第１のエンティティを認証するコンピュータによって実装される方法も提供され、第１のエンティティは、複数の因子に分けられたシークレットと関連付けられ、方法は、計算を実行することと、複数の因子から再構築されるシークレットから導出されるデータに基づいてバイリニアマッピングを使用することと、計算に基づいて第１のエンティティがシークレットを所有していたに違いないかどうかを判断することを含む。

【０１１４】

方法はさらに、シークレットの複数の因子から、第１のエンティティでシークレットを再構築することと、再構築されたシークレットを使用してバイリニアマッピングに対する入力を導出することを含んでよく、計算を実行することは、第２のエンティティと関連付けられたデータに基づいて当該入力を第１の入力及び第２の入力として採る第１のバイリニアマッピングを計算すること、並びに第１のエンティティと関連付けられるデータ及び第２のエンティティと関連付けられるシークレットから導出される第２の入力データに基づいて第１の入力を採る第２のバイリニアマッピングを計算することを含み、第１のエンティティのシークレットは第１のエンティティと関連付けられる当該データから構築され、第２のエンティティのシークレットは第２のエンティティと関連付けられるデータから

構築され、これによって双線型性を使用して、マッピングの結果またはマッピングの結果から導出される値は、第１のエンティティがそのシークレットを所有しているかどうかを第２のエンティティによって判断するために使用できる。

【０１１５】

方法は、第１のエンティティまたは第２のエンティティのどちらかで第１のバイリニアマッピングを計算することを含んでよい。

【０１１６】

計算は、第１のバイリニアマッピング及び第２のバイリニアマッピングに対応する２つのペアリングを実現するマルチペアリングを実行することと、第１のエンティティがシークレットを所有していたに違いないかどうかを判断することが、マルチペアリングの結果が所定値に等しいかどうかを判断することを含んでよい。

【０１１７】

代わりに、計算は、バイリニアマッピングの結果をハッシュすることを含んでよく、第１のエンティティがシークレットを所有していたに違いないかどうかを判断することは、結果または結果のハッシュを比較することを含んでよい。

【０１１８】

本発明のさらに別の態様に従って、命令がその上に記憶される少なくとも１つのメモリ、及びクライアントを認証するために命令を実行するようにプログラミングされる少なくとも１台のプロセッサを含んだ装置も提供され、クライアントは複数の因子に分割されるシークレットと関連付けられ、命令は実行時少なくとも１台のプロセッサに、複数の因子から再構築されたシークレットから導出されるデータに基づいてバイリニアマッピングを使用して計算を実行させ、計算に基づいて第１のエンティティがシークレットを所有していたのに違いないかどうかを判断させる。

【０１１９】

本発明のさらに別の態様に従って、装置の少なくとも１台のプロセッサによって実行されるときに、装置の少なくとも１台のプロセッサにクライアントを認証するための認証プロセスを実行させる命令がその上に記憶される非一過性のコンピュータ可読媒体も提供され、クライアントは複数の因子に分割されたシークレットと関連付けられ、命令は、実行時、複数の因子から再構築されたシークレットから導出されるデータに基づいて、バイリニアマッピングを使用して、１台または複数のプロセッサに計算を実行させ、第１のエンティティが計算に基づきシークレットを所有していたに違いないかどうかを判断させる。

【０１２０】

本発明の実施形態が、例として、添付図面を参照して説明される。

【図面の簡単な説明】

【０１２１】

【図１】クライアント、認証サーバ、及び信頼機関を含んだシステムの例を示す概略ブロック図である。

【図２】図１のシステムでのクライアントの概略ブロック図である。

【図３】図２のシステムでの認証サーバの概略ブロック図である。

【図４】図１のシステムでの信頼機関の概略ブロック図である。

【図５】代替システムの例の概略ブロック図である。

【図６】サーバに対してクライアントを認証するためのプロトコルを示す流れ図である。

【図７】サーバに対してクライアントを認証するためのプロトコルを示す流れ図である。

【図８】ＰＩＮエラーを処理するための認証サーバでのプロセスを示す流れ図である。

【図９】セッション鍵も確立するための図６及び図７のプロトコルの延長を示す流れ図である。

【図１０】セッション鍵も確立するための図６及び図７のプロトコルの延長を示す流れ図である。

【図１１】時間許可証も含むサーバに対してクライアントを認証するためのプロトコルを示す流れ図である。

【図１２】時間許可証も含むサーバに対してクライアントを認証するためのプロトコルを示す流れ図である。
【図１３】システムの当事者に追加の情報を送信するために時間許可証を使用するシステムの概略図である。
【図１４】システムのエンティティでのセットアッププロセスを示す流れ図である。
【図１５】システムのエンティティでのセットアッププロセスを示す流れ図である。
【図１６】システムのエンティティでのセットアッププロセスを示す流れ図である。
【図１７】情報を、システムによって提供される内密のチャネルでどのようにして送信できるのかの例を示す流れ図である。
【図１８】情報を、システムによって提供される内密のチャネルでどのようにして送信できるのかの例を示す流れ図である。
【図１９】情報を、システムによって提供される内密のチャネルでどのようにして送信できるのかの例を示す流れ図である。
【発明を実施するための形態】
【０１２２】
図１を参照すると、クライアント２、認証サーバ３、及び信頼機関（ＴＡ）４を含んだシステム１が示されている。クライアント、認証サーバ、及びＴＡはデータネットワーク５等の通信媒体上で通信してよい。クライアント２はユーザー６によって操作される。
【０１２３】
クライアント２は、例えば、データまたは別のリソースにアクセスする、トランザクションを実行する、またはサーバにデータを送信するためにサーバ３に対してクライアント自体を認証することを希望する。クライアントは、（図１に図示されていない）別のサーバへのアクセスを得ることを希望することがある。また、クライアント及びサーバは、認証に続いて安全な方法でデータを交換することも希望することがあるため、クライアント及びサーバはデータを交換するための暗号鍵を確立することを希望することがある。鍵はメッセージを暗号化するためにクライアントによって、及びメッセージを解読するためにサーバによって使用されてよく、逆もまた同様である。
【０１２４】
ＴＡ４は、クライアントにシークレット７を、及びサーバに別のシークレット８を発行する。シークレットはＴＡ４によって記憶されるマスターシークレット９から導出される。いくつかの実装は、クライアント２がサーバ３に対して、クライアントがそのシークレット７を所有していることを、サーバに対してシークレット自体またはシークレットについての何かを明らかにすることなく証明できるようにする。また、本明細書に説明されるプロトコルのいくつかの例は、クライアントが実際に独自のシークレット７を有していると判断するために、サーバ３がサーバシークレット８を所有していることも必要とする。クライアント２に対してシークレット７を発行する決定は、サーバへの接続を試行するクライアントにシークレットを発行することの、サーバ３から信頼機関４への要求時に実行されてよい。サーバは以前にシステムに登録し、信頼機関４から独自のシークレット８を受け取った可能性がある。いったんシークレットがクライアントに発行されると、クライアントは次いでクライアント自体をサーバに対して認証することができ、サーバ及びクライアントは通信リンクを確立できる。サーバ３は、クライアント２のアイデンティティをＴＡに提供して、ＴＡがクライアントシークレット７を導出し、安全な接続上でそれをクライアントに送信できるようにしてよい。他の例では、クライアントはそれ自体でそのアイデンティティをＴＡに送信し、ＴＡに、サーバ３に対して認証するためにクライアントによって使用されるクライアントシークレットをクライアントに発行することを要求してよい。クライアントのアイデンティティは、ユーザー及び／またはクライアントを識別する任意のデータ列を含んでよい。アイデンティティは名前、ｅメールアドレス、社会保障番号、役職、または免許証番号を含んでよいが、これに限定されない。
【０１２５】
クライアントは受信されたシークレット７を複数の因子１０及び１１に処理する。図１

では、シークレットは第１の因子１０及び第２の因子１１に分割されると示される。ただし、シークレットは任意の数の因子に分割できる。セットアップ中、クライアントはユーザー６から因子１１の内の１つを受信し、クライアントシークレット７からその因子を抽出し、クライアントに残っているもの、つまり第１の因子１０を記憶する。クライアントシークレットが２つを超える因子に分割される場合、クライアントはシークレットから複数の因子を抽出してよい。例えば、クライアントはユーザーから、または複数のユーザーから２つ以上の因子を受信し、シークレットからすべての因子を抽出してよい。

【０１２６】

図１のシステムでは、シークレットの第１の因子１０はトークンである。トークンはソフトウェアトークンであってよい。ソフトウェアトークンはデータ列であってよい。データ列は、例えば「クッキー」として記憶されてよい。第２の因子１１は、任意のデータ列であることがある。データ列は、例えば暗証番号（ＰＩＮ）、パスワード、またはソフトバイオメトリックであってよい。以後、第２の因子１１をＰＩＮとして参照するが、それが別のタイプのデータでもあり得ることが理解される。クライアントは、トークン１０だけを記憶してよい。ＰＩＮ１１は、ユーザーによって覚えられてよい。

【０１２７】

ユーザー６が、例えばサーバもしくは別のエンティティでデータにアクセスするために、またはサーバにデータを送信するために、後でサーバ３に対して認証することを希望するとき、それは、クライアント２がシークレットを取り戻すために記憶されているトークンとともに使用できるＰＩＮ予想値１１’を入力する。ＰＩＮ予想値１１’が実際のＰＩＮ１１に一致する場合、クライアントは実際のシークレット７を取り戻す。シークレットが２つを超える因子に分割される場合、クライアントは、クライアントがシークレットを取り戻すために記憶されているトークンとともにクライアントが使用する複数の因子を受信してよい。

【０１２８】

クライアント及びサーバは、複数のメッセージ１２を交換するために進む。メッセージは、コミットメント１２ａ、チャレンジ１２ｂ、及び応答１２ｃを含んでよい。コミットメント、チャレンジ、及び応答のそれぞれは、複数のメッセージを含んでよい。また、メッセージ１２は追加のメッセージを含んでもよい。メッセージに基づいて、以下にさらに詳細に説明されるように、サーバは次いでクライアントを認証できる。サーバに対してクライアントを認証し、クライアント及びサーバは、より詳細に説明されるように、複数の他のメッセージ１３を交換するために進んでよい。クライアント及びサーバは、メッセージの中のデータのいくらかを暗号化するために鍵１４を確立するために進んでよい。また以下により詳細に説明されるように、鍵は、認証メッセージ１２で交換される情報、クライアント２及びサーバ３のそれぞれのシークレット並びにプロセスで作成され、それぞれの当事者だけに既知である他のパラメータから導出されてよい。サーバのシークレットは鍵を計算するためにも必要とされるので、インポスターサーバは、それが本当のサーバであると偽り、プロトコルを完了してクライアントを認証し、セッション鍵を確立することはできない。代わりに、鍵は例えば他のメッセージ１３でＳＳＬを使用して確立されてよい。

【０１２９】

図１は、１つのクライアントと１つの認証サーバしか示していないが、システムが複数の認証サーバ及びクライアントをサポートすることができ、各認証サーバが複数のクライアントと安全な通信リンクを確立できることが理解される。

【０１３０】

クライアント、サーバ、及び信頼機関のそれぞれは、ここで図２、図３、及び図４に関してより詳細に説明される。

【０１３１】

クライアント、サーバ、及びＴＡは、ソフトウェアとしてまたはソフトウェア及びハードウェアの組合せとして実装されてよい。以下の説明では、クライアント、サーバ、及び

ＴＡは、ソフトウェアが実行するコンピューティング装置も含むとして説明される。ただし、いくつかの実装では、クライアント、サーバ、及びＴＡは、１台または複数のプロセッサによって実行されるときに、それぞれクライアント、サーバ、及びＴＡの機能性を提供する命令をコンピュータプログラムとして提供される。暗号機能は、例えば、プラグインソフトウェアモジュールによって提供されてよい。

【０１３２】

図２を参照すると、クライアント２はコンピュータ、サーバ、または他のコンピューティング装置を含む。クライアント２は、例えばパーソナルコンピュータ、ラップトップ、または携帯電話または他のパーソナルポータブルデバイスまたはモバイル機器であってよい。コンピューティング装置は、プロセッサ２１、コンピュータプログラム命令を記憶するためのメモリ２２、及びデータを記憶するためのストレージ２３を含む。メモリ２２は、クライアントがＰＩＮ１１を作成できるようにするための初期化プログラム２４を記憶し、トークン１０、及びクライアントがＰＩＮ１１及びトークン１０を使用してサーバに対してクライアント自体を認証できるようにするための認証プログラム２５を記憶してよい。クライアントは、安全な通信リンク上で、ＴＡ４から初期化プログラム２４、及び認証サーバ３から認証プログラム２５を受信した可能性がある。代わりに、クライアントはＴＡ、認証サーバ、または別のエンティティから両方のプログラムを受信する。いくつかの実装では、初期化プログラム及び認証プログラムは、同じコンピュータプログラムの部分を形成する。初期化プログラムはシステムに加わるための命令を記憶してよい。命令は、ユーザー６がＰＩＮを入力するためのグラフィックユーザーインタフェースを提供するための命令、ＰＩＮを入力するためにユーザーによって使用される入力装置からＰＩＮを受信するための命令、及びシークレットの複数の因子を作成するためにシークレット７からＰＩＮを抽出するための命令を記憶してよい。入力装置は、例えば、クライアントの部分を形成するまたはクライアントに接続可能であるキーパッドまたはキーボードであってよい。認証プログラム２５は、例えばＰＩＮ及びトークン等の複数の因子からクライアントシークレットを再構築するための命令を含み、サーバとの認証プロトコルに関与してよい。また、命令は、クライアントが認証サーバとの認証鍵共有プロセスに関与できるようにもする。いくつかの例では、初期化プログラム２４は、信頼機関４によってクライアントに対してウェブインタフェースを通して提供されてよい。同様に、いくつかの例では、認証プログラム２５は、認証サーバ３からクライアント２にウェブインタフェースを介して提供されてよい。クライアントはウェブインタフェースを通してプログラムを受信し、メモリ２２にプログラムを記憶してよい。いくつかの実装では、プログラムがウェブインタフェースを通して提供されるとき、プログラム、またはプログラムの一部はメモリ２２に記憶されないことがある。

【０１３３】

また、メモリはサーバ及び他のデバイスとの通信のための追加の命令を記憶してもよい。例えば、命令はブラウザを提供してよい。ブラウザは、ユーザーがワールドワイドウェブにアクセスするためのウェブブラウザであってよいが、ブラウザはプライベートネットワークでリソースにアクセスするためのブラウザであることもある。初期化プログラム２４及び認証プログラム２５は、ブラウザの一部を形成してよい、またはブラウザを通して提供されてよい。クライアントは、例えばプログラムをダウンロードし、クライアントのブラウザメモリに記憶してよい。いくつかの例では、クライアント２は、ブラウザによって提供されてよい。

【０１３４】

プロセッサ２１は命令を実行するように構成される。プロセッサは、メモリ２２での命令の実行中になんらかのデータを記憶し、迅速にアクセスするために、例えばキャッシュメモリ等の専用の内蔵一時メモリを有する。

【０１３５】

ストレージ２３はトークン１０を記憶してよい。ストレージはブラウザのストレージによって提供されてよい。代わりに、ストレージは、トークンを記憶するために別個のユニ

バーサルシリアルバス（ＵＳＢ）フラッシュドライブを含んでよい。ストレージは、クライアントのアイデンティティ２６も記憶してよい。例えば、クライアントのアイデンティティはトークン１０とともにメタデータとして記憶されてよい。他の例では、アイデンティティはクライアントのストレージに記憶されるのではなく、アイデンティティが必要とされるたびにユーザーによって入力される。時間許可証が使用されるいくつかの例では、ストレージは時間許可証２７も記憶してよい。時間許可証は、以下により詳細に説明される。ストレージは保護される必要はない。記憶されていないＰＩＮ１１がないと、クライアントシークレットを取り戻すことができない。結果的に、たとえ外部エンティティがトークン及びクライアントアイデンティティを手に入れても、外部エンティティは完全なシークレット７を取り戻すことはできない。

【０１３６】

初期化プログラム２４及び認証プログラム２５は、ＪａｖａＳｃｒｉｐｔ（登録商標）プログラムによって提供されてよい。（登録商標）プログラムは、１つまたは複数の別々のプログラムモジュールを含んでよく、初期化プログラム及び認証プログラムは別々のモジュールによって提供されてよい。ＪａｖａＳｃｒｉｐｔ（登録商標）プログラムは、ユーザーからＰＩＮを受信するために使用されてもよい。ＪａｖａＳｃｒｉｐｔ（登録商標）プログラムは、ブラウザのストレージに、または代わりの保護されていないストレージにトークンを記憶してよい。クライアントは、このＪａｖａＳｃｒｉｐｔ（登録商標）プログラムによって提供されてよい。プログラムは、汎用ユーザーデバイスで実行するように構成されてよい。ＪａｖａＳｃｒｉｐｔ（登録商標）プログラムは、認証サーバ３及び／またはＴＡ４からデバイスにウェブインタフェースを通して提供されてよい。いくつかの実装では、プログラムは、例えば認証サーバ３及び／またはＴＡ４から受信またはダウンロードされ、汎用ユーザーデバイスにインストールされるだろう。

【０１３７】

図１に関して言及されたように、クライアントはユーザー６によって操作されてよい。ユーザー６は、通常、人間であるだろう。ただし、ユーザー６は、代わりにコンピュータ等の機械であってよい。ユーザーが機械であるとき、ＰＩＮは機械によって生成されてよい。サーバ３に対する認証が必要とされるとき、機械は次いでクライアントがシークレット７を再構築できるようにするためにクライアント２にＰＩＮを提供する。

【０１３８】

図３を参照すると、認証サーバ３はコンピュータ、サーバまたは他のコンピューティング装置を含む。コンピューティング装置は、プロセッサ３１、コンピュータプログラム命令を記憶するためのメモリ３２、及びデータを記憶するためのストレージ３３を含む。メモリ３２は、登録プロセス及び認証プロセスのサーバ側で実行するためのコンピュータコードの形をとる命令を有するサーバ認証プログラム３４を含む。認証プログラムは、独自のシークレットを得るためにＴＡと登録するための命令を含む。また、プログラムはクライアントを認証するための命令も含む。いくつかの実装では、プログラムはセッション鍵を導出するためのコンピュータ命令を含んでよい。プログラムはさらに、クライアントが認証プロトコルのクライアント側を実行するための命令を含んだクライアント認証プログラム２５をクライアントに送信するための命令を含んでよい。受信時、汎用ユーザーデバイスは、システムに参加し、ＴＡから受信されるシークレットを使用し、汎用ユーザーデバイス自体をサーバに認証するためにプログラムを使用できる。クライアントはプログラムを記憶し、実行してよく、プログラムは、ＴＡから受信されるシークレットを使用するサーバに関して認証プロセスを実行するようにクライアントを構成してよい。さらに、認証プログラム３４は、クライアントがユーザーから間違ったＰＩＮを受信し、その結果サーバに対して認証する試行に失敗した場合にエラー処理プロセスを実行するための命令を含んでよい。認証プログラム３４は、ＴＡ４に登録するためのモジュール、及びクライアント２と通信するための別個のモジュールを含んでよい。また、認証プログラム３４がエラー処理プロセッサを実行するための命令も含むとき、認証プログラムは別個のエラー処理モジュールを含んでよい。サーバ３は、ＴＡ４から認証プログラム３４を得た可能性が

ある。

【０１３９】

プロセッサ３１は、メモリ内の命令を実行するように構成される。プロセッサは、メモリ３２内の命令の実行中に一時的にデータを記憶するための、キャッシュメモリ等の専用の内蔵一時メモリも備えてもよい。

【０１４０】

ストレージ３３は、サーバシークレット８を記憶するための安全なストレージを含んでよい。ストレージは、ストレージがそのコピーを、サーバに対して認証することを希望する新しいクライアントごとに送信するクライアント認証プログラムも含んでよい。

【０１４１】

上述されたように、いくつかの実装では、認証サーバはソフトウェアのみで実装され、プロセッサによって実行されるプログラム命令だけを含む。他の実装では、サーバは専用のハードウェアデバイスであってよい。

【０１４２】

図４を参照すると、ＴＡ４はコンピュータ、サーバ、または他のコンピューティング装置を含んでよい。ＴＡはクライアント及びサーバに対して別個のエンティティであり、システムのコントローラとして働く。コンピューティング装置は、プロセッサ４１、コンピュータプログラム命令を記憶するためのメモリ４２、及びデータを記憶するためのストレージ４３を含む。メモリは、マスターシークレット９を生成し、システムで参加者を登録する初期化プログラム４４を記憶してよい。また、メモリは、システムのエンティティに識別子を割り当ててもよい。また、プログラムは、クライアントがトークン及びＰＩＮを生成できるようにするためにクライアントに初期化プログラム２４を送信するための命令を含んでもよい。例えば、上述されたように、ＴＡはウェブインタフェースを通してクライアントに初期化プログラム２４を提供してよい。また、初期化プログラムは、以下により詳細に説明されるように、時間許可証を発行するための命令を含んでもよい。

【０１４３】

プロセッサ４１は、メモリ内の命令を実行するように構成される。プロセッサは、メモリ４２内の命令の実行中にデータを一時的に記憶するための、キャッシュメモリ等の専用の内蔵一時メモリも含む。

【０１４４】

ストレージ４３は、マスターシークレット９を記憶する安全なストレージを含んでよい。安全なストレージは不正操作不可または不正使用不可であってよい。また、安全なストレージは、システムで登録されるクライアント及びサーバのリスト４５も含んでよい。また、クライアント及びサーバのリスト４５は、以下により詳細に説明されるようにクライアントのための時間許可証も記憶してよい。さらに、信頼機関のストレージは、以下により詳細に説明されるように、プロトコルで使用される代数曲線及びハッシュ関数を定めるパラメータを含んだ、プロトコルで使用されるパラメータ４６も記憶してよい。ＴＡはクライアント及びサーバにこの情報を送信する、または必要とされるときにクライアント及びサーバが得るためにこの情報を公開してよい。情報は、クライアント２に提供される初期化プログラムに含まれてよい。

【０１４５】

上述されたように、いくつかの例では、ＴＡはプロセッサによって実行されるプログラム命令を含むにすぎない。他の例では、ＴＡは専用のハードウェアデバイスであってよい。

【０１４６】

図１に関して説明されるように、クライアント、サーバ、及びＴＡはデータネットワーク等の通信媒体５によって接続されてよい。例えば、エンティティはインターネットによって接続されてよい。ただし、クライアント、サーバ、及びＴＡは、さらにまたは代わりに他の通信媒体を使用して通信してよい。クライアント、サーバ、及びＴＡのプロセッサ、メモリ、及びストレージはそれぞれ１台または複数のプロセッサ、メモリ、及びストレ

ージモジュールを含んでよいことが理解される。

【０１４７】

システムは、クライアント２がクライアント自体を認証サーバ３に対して認証できるようにするためにバイリニアマッピングを使用してよい。システムは、クライアントが、そのシークレット情報を認証サーバに送信することなくクライアント自体をサーバに認証できるようにするためにバイリニアマッピングを使用してよい。これはゼロ知識証明認証として知られている。ゼロ知識証明暗号方式は、一方の当事者（証明者）が他方の関係者（検証者）に対して、一方の当事者がシークレット値を有していることをシークレットについて何も明らかにすることなく証明する方式である。結果的に、クライアントがサーバに対して何回認証しようとも、サーバはシークレットを決定することはできない。

【０１４８】

バイリニアマッピングは、ペアリングを含んでよい。ペアリングは特別なペアリングにフレンドリな楕円曲線に作用する。非対称ペアリングマップが使用されてよい。ペアリングの双線型性特性は、本明細書に説明されるゼロ知識証明暗号方式の例を実現するために使用されてよい。

【０１４９】

楕円曲線上のペアリングは当業者によって知られており、本明細書で詳細に説明されない。簡略に、非対称ペアリングは、別個の第１の群及び第２の群に属する曲線上の２つの点を入力として採り、点を第３の群での要素にマッピングする。ペアリングのバイリニア特性のため、クライアントシークレットｓＡ及びサーバと関連付けられたデータは、クライアントシークレット及びサーバシークレットが、それぞれ特殊な方法でクライアントと関連付けられるデータ及びサーバと関連付けられたデータから構築されるときに、クライアント及びサーバシークレットｓＱと関連付けられるデータとして第３の群内の同じ要素にマッピングできる。

$$e(sA, Q) = e(A, sQ) = e(A, Q)^{s}$$

【０１５０】

クライアントシークレット７は、クライアントアイデンティティ２６の形でクライアントと関連付けられた公用データを第１の群に属する曲線上の点Ａにハッシュし、次いで、曲線上で該点をマスターシークレットｓで乗算することによってＴＡで構築されてよい。サーバシークレット８は、曲線上で第２の群の固定点Ｑをマスターシークレットｓで乗算することによってＴＡで構築される。結果的に、サーバは、サーバで独自のシークレットデータに基づくペアリングを計算し、そのペアリングから導出されるクライアントシークレットまたは値に基づくペアリングの結果を分析することによって、クライアントシークレット７自体を受け取ることなく、クライアントがクライアントシークレット７を所有していたかどうかを判断できる。

【０１５１】

当業者は、ペアリングを実行するために使用できる多数の適切な楕円曲線が存在することを理解する。さらに、ペアリングは本明細書で楕円曲線で実行されるために記載されているが、他の適切な曲線が使用されてよいことが理解される。

【０１５２】

楕円曲線代数は、曲線上の点の加算及び乗算を実行するための特定の規則を定義する。これらは技術で既知であり、本明細書では説明されない。簡略に、規則は、２つの点Ａ及びＢをともに加算して別の点Ｃを得ることを含む加算と呼ばれる演算を定義する。幾何学的には、これは、例えば、異なるｘ座標を有する点のために、点ＡとＢの間で直線を描画し、直線が曲線に交差する点Ｃを見つけることを含んでよい。ただし、規則は、どのようにして代数的に新しい点を得ることができるのかも定める。乗算と呼ばれる演算はｋＡを得るためにｋ回数、同じ点Ａの繰り返される加算によって定義される。減算と呼ばれる演算は、別の点Ａから減算される点Ｂの負数を得て、次いで点Ａ及び‐Ｂの加算を実行することを含む。Ｂの負数は同じｘ座標を有しているが、ｙ座標の負数、－Ｂを有する点である。

【０１５３】
当業者によって認識されるように、本明細書に説明される暗号プロトコルを実現するために多くの異なるタイプのペアリングを使用できる。例えば、ウェイルペアリング、テートペアリング、エイトペアリング、及びＲ－エイトペアリングは、使用され得る適切なペアリングの例である。
【０１５４】
プロトコルはさらに、クライアント２がペアリングを計算することを必要とされず、代わりに大量の処理が認証サーバ３で実行されるように構築されてよい。クライアントは、クライアントのシークレット７を使用して計算を実行し、サーバに結果を送信する。サーバは、サーバ自体のシークレット８を使用して、及びペアリング計算を計算することによって、クライアントが、クライアントシークレットを所有することなく、受信された結果を得ることができただろうかをチェックする。サーバが、クライアントがシークレットを所有していなければ計算の結果を得ることができなかっただろうと判断する場合、サーバはクライアントを認証する。サーバで実行される計算は、クライアントから受信される結果を使用するペアリング、及び専用のシークレット情報、並び該２つのペアリングの積に基づくペアリングを含んでよい。計算は、マルチペアリング計算として実行されてよい。いくつかの例では、製品が特定の値に等しい場合、サーバは、クライアントが事実上鍵を所有していると判断できる。いくつかの例では、サーバが、クライアントがシークレットを所有していないと判断する場合、サーバはクライアントによって受信されるＰＩＮのエラーの範囲を決定できることがある。クライアントに対するサーバの応答は、ＰＩＮでのエラーの範囲に依存してよい。
【０１５５】
いったんクライアントが認証されると、クライアント及びサーバは次いでセッション鍵を計算するために進んでよい。クライアント及びサーバは、すでに交換された情報及び追加の情報に基づいて同じ対称セッション鍵を確立するために進んでよい。代わりに、クライアント及びサーバは、トランスポート層セキュリティ／セキュアソケットレイヤ（ＴＬＳ／ＳＳＬ）プロトコルを使用してセッション鍵を確立してよい。いくつかの例では、使用される鍵はセッションごとに異なり、クライアント及びサーバはセッションごとに新しいセッション鍵を確立し、パーフェクトフォワードシークレット性（ＰＦＳ）を提供する必要があるだろう。
【０１５６】
図５に関して、別のシステムの例が示されている。図１及び図５では、類似する参照番号は類似する構成要素を示す。図５では、認証サーバ３は、２台のサーバ３ａ及び３ｂ、並びにプロキシ３ｃによって置き換えられる。上述されたように、図３の認証サーバ３のプロセッサ３１、メモリ３２、及びストレージ３３は、それぞれ複数のプロセッサ、メモリ、及びストレージを含んでよく、認証サーバ３は、それぞれが専用のプロセッサ、メモリ、及びストレージを有するいくつかのサーバモジュールと、分散型サーバとして実装されてよい。図５のシステムでは、２台のサーバ３ａ、３ｂ、及びプロキシ３ｃのそれぞれは、専用のプロセッサ、メモリ、及びストレージを有してよい。サーバ３ａ及び３ｂのそれぞれは、サーバシークレット８の部分８ａ、８ｂを発行され、サーバシークレット８の部分８ａ、８ｂを記憶する。それ自体が別個のサーバであってよい、プロキシ３ｃはクライアント２と通信し、関連する情報を各サーバ３ａ、３ｂに渡す。各サーバは独自の計算を実行し、プロキシ３ｃは、クライアント２を認証するために計算の結果を結合する。計算は、プロキシもサーバも完全なサーバシークレット８を得ることができないように実行される。結果的に、プロキシまたはどちらかのサーバに対する攻撃は、ハッカーがサーバシークレットを得て、クライアントに対しサーバのふりをすることを許さない。また、プロキシ３ｃはサーバ３ａ、３ｂから受信される結果からセッション鍵１４を確立してもよい。いくつかのシステムでは、別個のプロキシは存在しないことがあるが、サーバの内の１つが代わりにプロキシの機能を実行し、関連する情報を他のサーバに通信する。この分散型のサーバ配置は、以下により詳細に説明される。

【０１５７】
いくつかのシステムでは、ＴＡは２つの構成要素に分割されてよく、該構成要素のそれぞれはマスターシークレットの一部を記憶し、ハッカーがマスターシークレットを盗むことをより困難にする。
【０１５８】
以下に説明されるプロトコルの例のいくつかは、「Ｍ－Ｐｉｎ」と呼ばれる。「Ｍ－Ｐｉｎ」だけが、認証を実行するためのプロトコルを指す。「Ｍ－Ｐｉｎ　Ｆｕｌｌ」はやはり鍵共有を実行するための拡張プロトコルを指す。
【０１５９】
以下に段階的な手法での多様なプロトコル及び実装を説明する。多様な既知のプロトコル及び方式、並びにそれらのプロトコル及び方式に対して提案されている修正形態または変形形態を説明する。次いで、Ｍ－Ｐｉｎプロトコル及びＭ－Ｐｉｎプロトコルに基づくいくつかのプロトコルを説明する。また、代替プロトコルも説明する。さらに、プロトコルの実装及び変形の例、並びにプロトコルをどのようにして使用できるのかの例も説明する。
【０１６０】
既存の提案の検討
２００３年に、「ＩＤ－ｂａｓｅｄ　ｐａｓｓｗｏｒｄ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｓｃｈｅｍｅ　ｕｓｉｎｇ　Ｓｍａｒｔ　Ｃａｒｄｓ　ａｎｄ　Ｆｉｎｇｅｒｐｒｉｎｔｓ」方式がＫｉｍら［Ａ１３］によって提案された。Ｓｃｏｔｔ［Ａ２０］は、該方式が単一のトランザクションを受動的に盗聴したアタッカーによってどのようにして包括的に壊されるのかを示した。明確にするために、参考文献［Ａ１３］及び［Ａ２０］の詳細は、本明細書で参照される他の参考文献の詳細とともに、明細書末尾に一覧表示される。
【０１６１】
Ｍａｒｔｉｎｅｚ－Ｐｅｌａｅｚ及びＲｉｃｏ－Ｎｏｖｅｌｌａ［Ａ１５］によるはるかに最近の２０１２年の論文は、自らの論文「Ａｎ　Ｉｍｐｒｏｖｅｍｅｎｔ　ｏｆ　Ｌｉａｏ　ａｔ　ａｌ’ｓ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｓｃｈｅｍｅ　ｕｓｉｎｇ　Ｓｍａｒｔ　Ｃａｒｄｓ」［Ａ２３］に説明されるＳｏｏｄ、Ｓａｒｊｅ、及びＳｉｎｇｈによる方式の暗号解析に成功している。この方式は、基本的に２要因認証提案である。［Ａ１５］の要約「…我々は、Ｓｏｏｄらの方式が依然として悪意のあるユーザー攻撃、介入者攻撃、盗難スマートカード攻撃、オフラインＩＤ予想値攻撃、なりすまし攻撃、及びサーバスプーフィング攻撃に対して弱いことを示している…」。二因子識別に関する引用論文のリストを追いかけると、同様に壊された不毛なプロトコルを発見する。多くの他の方式は、多くの文献は生成するが、かなり信頼できない破損‐修復‐破損‐修復のサイクルに陥っている。暗号法の確立はこの研究分野を、誰も非常に長くは生き残ることがない一種の暗号法の西部の荒野として大方放棄してしまったようである。
【０１６２】
この問題は、「認証鍵交換の分野での過去３０年間の研究は、単一因子に基づく鍵交換（ＫＥ）方式でさえ正しく得ることが信じられないほど困難であることを証明してきた。多因子認証鍵交換（ＡＫＥ）プロトコルの設計は、さらに困難となるだけである。］と述べることによってこの問題を説明するＨａｏ及びＣｌａｒｋｅ［Ａ１２］によって認識された。しかしながら、詳しくない読者の反応は以下のようである可能性がある。つまり、暗号プロトコルの開発のこの段階で、使いやすく且つ安全な二因子認証の問題に対する満足が行く且つ広く受け入れられる解決策がどうしてないと思われるだろうか。
【０１６３】
パスワード認証鍵交換（ＰＡＫＥ）は十分に研究された分野であり、ＰＡＫＥに対して多くの証明された方法が提案されている［Ａ３０］。しかしながら、大部分は、それこそハッキングされたサーバでアタッカーが探し、アタッカーがサーバのセキュリティを完全にアンロックでいるようにする「パスワードファイル」、つまり「パスワード検証者ファ

イル」の保守を必要とする。通常、このファイルは、＜Ｕｓｅｒｎａｍｅ，Ｓａｌｔ，Ｈ（Ｓａｌｔ｜Ｐａｓｓｗｏｒｄ）＞のタプルから構成され、上式でＳａｌｔは乱数であり、Ｈ（．）は一方向ハッシュ関数であり、｜は連結を示す。十分に認識されるように、高エントロピーを有するようにパスワードが選ばれない限り、アタッカーは、オフライン辞書攻撃をもってコンピュータ速度で辞書をくまなく検索することによってパスワードを発見することに成功する。辞書の外部にあるパスワードだけがこの攻撃を切り抜ける。このことが、高エントロピーパスワードは事前にサーバと合意されていなければならないというトートロジーにつながる。ＰＡＫＥの全体的な目的が高エントロピー暗号化キーを相互に認証し、同意することであることを思い出されたい。

【０１６４】

１つの代替策は、各クライアントにスマートカードを発行し、サーバ上ではなくここにパスワード関連情報を記憶することである。これは、認証プロセスに第２の因子、つまりスマートカード自体を付加する追加の利点を有する。ただし、これが機能するには、スマートカード構成要素が完全なスマートカード機能性を維持しなければならないようである。そのセキュリティが壊されると、次いでシステムは故障する。この方式の実装の成功についてはＹａｎｇら［Ａ３１］を参照すること。ただし、スマートカードは高価である。また、係る方式は、サーバが長期のシークレットｓを維持することをつねに必要とする。そして再び、サーバがハッキングされると、ｓが明らかになり、システム全体のセキュリティをアンロックする可能性がある。

【０１６５】

多因子認証

多因子認証は、一般に（ａ）我々がもっている何か、（ｂ）我々が知っている何か、及び（ｃ）我々である何かから成り立っている。多因子認証が我々にとって最もなじみが深い形は二因子ＡＴＭ銀行カード及び４桁のＰＩＮ番号としてだろう。ここでは、我々の説明の大部分は二因子認証との関連となるが、第３の因子に対するサポートも幾分考慮される。

【０１６６】

「我々がもっている何か」の因子は、おそらく磁気ストリップに記録されるデータのフォームファクタに、もしくはＱＲコード（登録商標）に静的データを記憶する物理的なトークン、またはなじみのあるＵＳＢスティックであることがある。また、それはスマートカードであることがある。スマートカードが、独自の保護されているシークレット及び計算能力を有することがあり、クローン化できないという点でスマートカードが追加の機能性を有することに留意されたい。ただし、スマートカードははるかに高価であり、スマートカードを失うと高価な代替が必要になる。

【０１６７】

「我々が知っている何か」の因子はパスワードである。ただし、パスワードは２つの異なる特徴に分けられる。例えば、８文字以上を有し、大文字と小文字の両方、及び少なくとも１つの数字を含まなければならないパスワード等、いま我々によって一般的に求められている高エントロピーパスワード。次に、例えば４桁のＰＩＮ等低エントロピーパスワードがある。結局、我々は高エントロピーパスワードには言葉のパスワードだけを、低エントロピーパスワードにはワードＰＩＮを使用することになる。それらを区別する代替の方法は、願わくば前者はそうではないが、後者がオフライン辞書攻撃によって容易に検出されるのを観察することである。オフライン辞書攻撃は、アタッカーが（コンピュータ速度で）考えられるパスワード／ＰＩＮの辞書を通読し、正しいものをいつ発見したのかを容易に検出できるものである。

【０１６８】

「我々である何か」因子は、おそらく指紋または虹彩スキャン等のバイオメトリックとして捕捉される。ただし、それらは一般的には不正確であり、バイオメトリック測定におけるある範囲の値が、許容可能とみなされ得る。

【０１６９】

バイオメトリックは「ハード」または「ソフト」であることがある。ソフトバイオメトリックは、小さな群の帰属関係に個人を制限するために使用できる分類を返す。ハードバイオメトリックは、正確な識別を行うために記憶されているテンプレートと併せて機能する。これは、ＰＩＮとパスワードとの比較に類似する‐明らかに、多くの人が安全に同じＰＩＮを共有しているが、理想的にはあらゆるパスワードは一意であるべきである。例えばソフトバイオメトリック指紋スキャナはきわめて安価であることがあるが、ハードバイオメトリックは高価となる傾向がある。ハードバイオメトリック用のテンプレートは攻撃に対する貴重な標的となり、どこかに安全に記憶されなければならない。

【０１７０】

多くの方式がトークンとしてスマートカードを提案している。しかしながら、方式は多くの場合、スマートカードのセキュリティが破られ、そのシークレットが明らかにされるケースを考慮し、その場合、それは受動トークンも同然である［Ａ２９］、［Ａ２６］。いくつかのアプリケーションの場合、我々は自らをこの最も簡略で、最も容易で、最も安価な（したがって、もっとも実際的な）シナリオに制限する。

【０１７１】

以下が想定される。

１．静的なクローン化可能なトークン、

２．低エントロピー４桁ＰＩＮ番号

３．（任意選択で）ソフトバイオメトリック。

【０１７２】

しかしながら、本発明の実施形態が二因子認証または三因子認証に制限されるのではなく、２つの因子、３つの因子、及び３を超える因子を含んだ複数の因子に分割されるシークレットを使用できることが理解されるだろう。我々のトークンは静的であってよいので、我々はソフトウェア専用の解決策を提供する立場にあることに留意されたい。クライアントの場合、提案されている方式の内のいくつかは、事実上、（静的データが磁気ストリップ上に記録されている）試され、信頼されているＡＴＭカード、及び関連付けられたＰＩＮ番号にきわめて類似している。ただし、はるかに敵意をもったドメイン、つまりインターネット上で機能するもの。

【０１７３】

セキュリティ特徴

多くの著者は係るプロトコルの望ましい特徴の役に立つリストを作成してきた。Ｔｓａｉら［Ａ２５］は、係る方式の９つのセキュリティ要件及び１０の目標のリストを提供する。ただし、彼らが執筆した時代に得ることができた方式に関する彼らの検討は、すべてが期待外れであることを明らかにしている。Ｌｉａｏら［Ａ１４］は、１０の特性のリストを考え出し、Ｙａｎｇら［３１］がそれを５に削減した。最近、Ｗａｎｇ［Ａ２９］が、係る方式が抵抗すべき８つの攻撃のリストを考え出し、潜在的なアタッカーの３つのカテゴリを特定した。

【０１７４】

この先行技術によって動かされ、ここでは方式が満たすことができるだろう条件の独自のリストを示す。

１．プロトコルによって、クライアント及びサーバが相互に認証され、相互の暗号化キーを導出することになるべきである。

２．サーバにはＰＩＮに関係するデータは記憶されない。それにも関わらず、サーバはクライアントが自らの忘れられたＰＩＮを取り戻すのを支援できるべきである。

３．根本的な認証鍵交換は「鍵漏洩なりすまし」の影響を受けない。すなわち、サーバはクライアントになりすますことはできず、クライアントはサーバになりすますことはできない。

４．クライアントは、サーバを巻き込まずにローカルでクライアントのＰＩＮを変更することができるべきである。

５．認証サーバのシークレットを手に入れたアタッカーは（ａ）偽のサーバをセットアッ

プする、及び（ｂ）クライアントトークンを考慮して、彼らのＰＩＮを決定できるだけであるべきである。基本的にはこれが、係る任意の方式に対して期待できる最善であることに留意されたい。
６．サーバはクライアントのシークレットでいかなる小さなエラーＥの範囲も特定できるべきである。これが不正確なバイオメトリック測定の因子としての包含を容易にする。
７．方式は「前方秘匿性」の特性をサポートするべきである。
８．方式は真に「多因子」であるべきである。ｎ個の因子が関与している場合、ｎ－１個の因子の損失は最終的な因子の発見を可能にするほど十分であってはならない。これが内部関係者による攻撃に対しても当てはまらなければならないのは言うまでもない。例えば、有効なトークン及びＰＩＮを備え、別のクライアントのトークンを取り込むクライアントは、そのＰＩＮ番号を決定できてはならない。
９．ＰＩＮ予想値攻撃は、オンラインだけでしか実行できない（したがってサーバによって監視し、防ぐことができる）。
１０．全体的なシステムは単一障害点を有してはならない。

【０１７５】

多くの既存の提案及びいくつかの適応された提案がどのようにしてこれらの特性を満たすことができないのかをここで説明する。次に、これらの特性を満たす方式、並びにこの方式を改良する、またはこの方式に対する代替策もしくは修正策を提供する多くのプロトコル及びシステムを説明する。文献の検討によれば、本明細書に説明される方式以前に係る方式は存在しなかったことが示唆されている。本明細書に説明されるすべてのプロトコルがこれらの特性を満たすわけではない。本発明の実施形態は１０の特性を満たすプロトコルに限られず、本発明の実施形態のいくつかは、１０の特性を満たさないプロトコル、並びにプロトコルの修正策及び変形策を提供する。本明細書に説明されるプロトコルの例のいくつかは、１０の特性を満たす他のプロトコルまたは方式とともに使用されてよい。

【０１７６】

条件３は、サーバがクライアントの登録を管理できるいずれの方式にとっても明らかに満たすことができないことに留意されたい。セキュリティを考慮するとき、サーバとそのクライアントとの間の通信を完全に制御することができ、攻撃を受けている個人以外のユーザーの任意の数のトークン及び関連ＰＩＮを所有することがある敵対者を想定する［Ａ３１］の敵対者モデルをその全体として採用できる。

【０１７７】

追加の既存提案の検討

次に、今日までの提案を少し詳しく見る。大部分は、ちょうど２つの当事者、つまりクライアント及びサーバが関与する状況を本拠地としている。クライアントはサーバとなんらかの初期の対話に入り（登録段階）、なんらかのクレデンシャルを発行されている。これにはつねにサーバが、認証プロセスに関与するなんらかのマスターシークレットを所有していることが必要になる。我々が提案する係る方式は、サーバがハッキングされる場合（とき）このシークレットが発見される可能性があり、方式全体のセキュリティがほころびるので、現状を改良する可能性は低い。

【０１７８】

より簡略な提案の多くを厳重に検査すると、その中心で、クライアント認証及び認証された秘密鍵の確立のための基礎として、サーバのマスターシークレットに結び付けられるクライアントアイデンティティの一方向ハッシュを使用する簡略な考え方が明らかになる。したがって、クライアントアイデンティティＩＤ及びサーバマスターシークレットｓの場合、プロトコルは、例えばハッシュ関数として標準セキュアハッシュアルゴリズム（ＳＨＡ２５６）を使用して高エントロピー相互シークレットＨ（ＩＤ｜ｓ）に構築されるだろう。このシークレットは、（パスワードで隠された）そのトークンでクライアントによって記憶され、必要に応じてサーバによってオンザフライで生成されるだろう。係る方式は多くの場合、前方秘匿性を提供するためにディフィーヘルマン構成要素も含む。これらの簡略な考えの複数の組合せが提案されている［Ａ３２］、［Ａ１４］、［Ａ３１］、［

Ａ２６］が、大部分は長引く詳細な調査を乗り切れなかった。Ｙａｎｇら［Ａ３１］の独創的な考え方はパスワード認証鍵交換（ＰＡＫＥ）でパスワードとして実際にＨ（ＩＤ｜ｓ）を使用することである。しかし、すでに指摘されたように、これはトークンのためのスマートカード機能性及び長期のサーバシークレットｓを必要とし、彼らが記載するように「システム全体のセキュリティは本来ｓのセキュリティに依存するため、ｓの秘密性はきわめて重要である」。しかし、サーバのハッキングは、パスワードファイルを盗むことよりもさらに致命的となるだろうｓを明らかにすることがある。

【０１７９】

大部分の提案の魅力的な特長は、多くの場合セキュリティの証明にほとんど努力がなされないという点である。このことは、なぜそれほど多くの方式がそれほど迅速に破られるのかを説明するのに確実に役立つ。しかしながら、なんらかの種類のセキュリティの証明を提供するそれらの提案は、我々の１０の望ましい特性と対照して測定されるときに思わしくない傾向がある。そして、言うまでもなく、セキュリティの証明はその想定程度に良好であるにすぎない。例えば、２０１０年にＳｔｅｂｉｌａら［Ａ２４］が多因子パスワード認証鍵交換を提案した。しかし、それは我々の条件２を満たしておらず、したがってサーバのハッキングの成功が、すべての静的な非高エントロピーパスワードを明らかにする。Ｙａｎｇら［Ａ３１］による提案は条件３を満たしていない。Ｗａｎｇ［Ａ２９］による最近の提案は内部関係者による攻撃（我々の条件８）の可能性を考慮しておらず、したがって驚くべきことではないがそれには陥落する。Ｐｏｉｎｔｃｈｅｖａｌ及びＺｉｍｍｅｒ［Ａ１６］の方式は我々の条件２を満たしておらず、最近、Ｈａｏ及びＣｌａｒｋｅ［Ａ１２］は彼らの正式のモデルの欠陥に基づいたそれでの問題を発見した。

【０１８０】

今日まで満足の行く解決策がないことは、おそらくアイデンティティベース暗号化（ＩＢＥ）に関する状況を写し出している。この概念は１９８４年にＳｈａｍｉｒ［Ａ２２］によって最初に提案されて以来周知であったが、標準的な公開鍵基盤（ＰＫＩ）構成物を使用しても、ある重要な条件または別の重要な条件に違反しなかった、ＩＢＥにとって満足の行く方式は一度も見つからなかった。実際的なプロトコルが可能になった［Ａ４］のは、暗号ペアリングの当時新規の構成物を使用することによって２００１年にすぎなかった。したがって、おそらく我々が提案した解決策が、いくつかの実装では暗号ペアリングの特性を利用していることは驚くにあたらない。

【０１８１】

ペアリングに基づくより優れた解決策の可能性が、「明示的な相互認証を提供し、［Ａ１４］で与えられるすべての特性を満たすために拡張できるプロトコル」を説明するとして論文［Ａ１９］を特定するＹａｎｇら［Ａ３１］によって暗示される。

【０１８２】

我々の解決策は、いくつかの実装では、二因子認証のための最初のソフトウェアのみによる方法に相当する。いくつかの実装は、上記に特定した望ましい特性の１０個すべてを満たし、その最も簡略な形で、ソフトウェアトークン及び容易に覚えられるＰＩＮ番号だけを必要とする。しかしながら、上述されたように、以下に説明される他の実装及びプロトコルは１０すべての特性を満たしていない可能性がある。

【０１８３】

ＰＫＩの標準方法の利用

いくつかのアプリケーションに対しては、理想的には、クライアントとサーバの両方ともがおもにオフラインの十分に保護されているサードパーティの信頼機関によって方式に登録されるＰＫＩと大差ない方法が好まれる。登録または記載の役割をサーバの役割から取り出すことによって、サーバシークレットの損失は、明確に損害を与えるが、願わくば壊滅的にはなるべきではない。

【０１８４】

いくつかの証明されている方法が、ＰＫＩの方法を使用して１因子認証鍵交換（ＡＫＥ）について報告されている。基本的な考え方は、ユーザーに、ユーザーが次いで鍵交換で

使用できるユーザーアイデンティティのデジタル署名を発行することである。

【０１８５】

Ｆｉｏｒｅ及びＧｅｎｎａｒｏ［Ａ６］の実証可能に安全なアイデンティティベースのディフィーヘルマン方式を考える。ここで、信頼機関はランダム群生成元ｇ及びランダムマスターシークレットｘを選び、公開鍵ｙ＝ｇ^{x}．を発行する。次に、信頼機関はランダムｋを選び、$r_{ID}=g^{k}$及び$s_{ID}=k+xH(ID, r_{ID})$を計算することによって（r_{ID}，s_{ID}）として提供されるアイデンティティでＳｃｈｎｏｒｒシグナチャ［Ａ１８］を発行する。鍵交換は表１でのように進む。

アリス - アイデンティティID_a	ボブ - アイデンティティID_b
ランダム a く q を生成する	ランダムｂくｑを生成する
$u_A = g^a$	$u_B = g^b$
$ID_a, r_A, u_A \rightarrow$	$\leftarrow ID_b, r_B, u_B$
$k = (u_B r_B y^{H(ID_b, r_B)})^{a+s_A}$	$k = (u_A r_A y^{H(ID_a, r_A)})^{b+s_B}$

表１．アイデンティティベースのディフィーヘルマン - Ｆｉｏｒｅ及びＧｅｎｎａｒｏ［Ａ６］

【０１８６】

両方の参加者とも同じ鍵$k=g^{ab}$で終了する。ここで、この方式をクライアント‐サーバ設定に適応させ、第２の因子としてＰＩＮ番号を含もうとする。即座に２つの問題に直面する。第１に、このプロトコルは、クライアント‐サーバよりむしろ本質的にピアツーピアである。したがって、どのようにＰＩＮ番号を埋め込もうとしても、係る方式はオフラインの内部関係者辞書攻撃に敗れるはずである。基本的にはトークン及びＰＩＮを有している誰かが「サーバ」として自身をセットアップできる。ここで、彼らが別のトークンを盗む場合、彼らは機能するＰＩＮを見つけるまで、その「サーバ」と鍵交換をオフラインで実行できる。さらに、非内部関係者がシステムを壊すこともある。基本的には、トークン値が有効なシグナチャとして明らかにされるまで、トークンを盗み、考えられるあらゆるＰＩＮ番号を試す。シグナチャの有効性は、認証機関の公開鍵がすべてに利用可能であるので容易に検証できる。

【０１８７】

著者自体が「ユーザーは公開鍵ｙを使用し、方程式

$$g^{s_{ID}} = r_{ID} \cdot y^{H(ID, r_{ID})}$$

をチェックすることによってその秘密鍵の正しさを検証できる」と言う。方式の特徴として暗示されているが、我々にとってはそれは問題を表す。係る方程式の存在は、この方程式がオフライン辞書攻撃がそれを取り戻すことができるようにするため、シークレットシグナチャと併せてＰＩＮを安全に使用できないことを意味する。提案する方式のいくつかでは、係る方程式は存在していない。いくつかの実装では、秘密鍵の正しさを検証する唯一の方法は、秘密鍵を使用して本物のサーバとの鍵共有を完了することであるべきである。

【０１８８】

二因子認証のために標準的なＰＫＩ方法を利用することでのこれらの問題は、Ｗａｎｇ［Ａ２９］によっても指摘されている。

【０１８９】

ペアリング及びＰＩＮ

ペアリングの出現で新しい解決策が可能になった。ペアリングは、通常は$G_1 \times G_2 \rightarrow G_T$と示される、同じ素数位数ｑの３つの群のある特別なペアリングにフレンドリな楕円曲線で機能する［Ａ２］、［Ａ７］。ここでは、G_1及びG_2が別個であり、Ｃ＝ｅ（Ａ，Ｂ），$A \in G_1$，$B \in G_2$及び$C \in G_T$であるタイプ３－ペアリング［Ａ９］を想定す

る。これらの群の要素は混合できないことに留意されたい。G_1にクライアントを、G_2にサーバを入れることを意図しているので、これは我々にとって非常に重要である。

【０１９０】

ペアリングの主要な重要な特性は、双線型性の特性である。

$$e(aA, bB) = e(bA, aB) = e(A, B)^{ab}$$

【０１９１】

オンラインで必要とされていない独自のマスターシークレットを有する独立した信頼機関（ＴＡ）の存在を仮定する‐ＴＡはオフラインの登録／記載、並びにクライアントＩＤベースのシークレット及びサーバＩＤベースのシークレットの発行だけに責任がある。これは、セキュリティの余分な層を提供し、クライアントまたはサーバの長期シークレットの損失によって生じる損害を制限する。

【０１９２】

ID_αを仮定し、ID_sがそれぞれアリスのアイデンティティ、及びサーバのアイデンティティであると仮定する。H_1（．）は群G_1の位数ｑの点にハッシュするハッシュ関数であり、H_2（．）は群G_2の位数ｑの点にハッシュするハッシュ関数である。次いで、クライアントアリス及びサーバはそれぞれシークレットｓＡ及びｓＳを発行され、$A = H_1(ID_\alpha)$、$S = H_2(ID_S)$であり、ｓは特定のサーバとともに使用するためのＴＡのマスターシークレットである。ｓは難解な別個の対数問題によって完全に保護されているので、Ａ及びｓＡを知っていることがｓを明らかにしないことは言うまでもない。

【０１９３】

ここで、アリス及びサーバが同時に相互に認証し、共通鍵を導出できるようにする簡略なＳＯＫの非対話鍵交換アルゴリズム［Ａ１７］を検討する。アリスはそれをｋ＝ｅ（ｓＡ，Ｓ）として計算し、サーバはそれをｋ＝ｅ（Ａ，ｓＳ）として計算する。双線型性によって、両方とも明らかに同じである。しかし、ここでアリスは自分のシークレットからの自分の選択のＰＩＮ　αを抽出して、それをトークン部（ｓ－α）Ａ、及びＰＩＮ部αＡに分割する。明らかに、これらが互いに加算されるとき、完全なシークレットを再構築できる。（元は［Ａ１９］で示唆された）この考えは、我々が提案する二因子認証方式のいくつかの例の開始点を形成する。

【０１９４】

外部ディフィーヘルマン（ＸＤＨ）仮定は最初に非公式に［Ａ１９］で暗示され、現在では広く使用されている。ＸＤＨ仮定は公式に以下を述べている。

１．離散対数問題（ＤＬＰ）、計算ディフィーヘルマン問題（ＣＤＨ）、及び計算共ディフィーヘルマン（ｃｏ－Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ）問題はすべてＧ＿１及びＧ＿２で解決困難である。

２．効率的に計算可能なバイリニアマップ（ペアリング）$G_1 \times G_2 \rightarrow G_T$が存在する。

３．決定ディフィーヘルマン（ｄｅｃｉｓｉｏｎａｌ　Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ）問題（ＤＤＨ）はG_1で解決困難である。

【０１９５】

直感は、効率的に計算可能なタイプ‐３ペアリングにも関わらず、G_1が「通常の」ディフィーヘルマン群として働くという点である。したがって、ＤＤＨ問題は以下のように記述できる。生成元Ｇ、ａＧ、ｂＧ及びｃＧを所与として、ｃ＝ａｂであるかどうかを判断する。ここで、ｓＢを発行され、$B = H_1(ID_b)$であり、（ｓ－α）Ａを手に入れるアイデンティティがID_bの内部関係者のアタッカーボブを考える。ハッシュ関数がそのジョブを正しく実行している場合、Ａ及びＢは巡回群G_1の位数ｑの独立した生成元であり、なんらかの未知のｗの場合Ｂ＝ｗＡである。ボブの内部情報を利用する唯一の方法は、ボブがｓＢに関する内部の知識を使用していつ（ｓ－α）Ａ＋ｇＡ＝ｓＡなのかを特定できるまで、予想値ｇを試し続け、ｇＡを（ｓ－α）Ａに加算し続けることである。$G_1 = G_2$であるタイプ‐１ペアリングでは、ボブは、ｇ＝αのときにｅ（Ａ，ｓＢ）＝ｅ（（ｓ－α）Ａ＋ｇＡ，Ｂ）を使用できるので容易である。ただし、タイプ‐３ペアリングでは、ＸＤＨ仮定がG_1で決定ディフィーヘルマン（ＤＤＨ）問題を解決する能力を暗

示するので、ＸＤＨ仮定に従ってこれは可能ではない。

【０１９６】

証拠：Ａ＝ωＢのように代入し、Ａ、ωＡ、χＡ及びｓωＡを与えられた場合、ｘ＝ｓであるかどうかを判断できるＯｒａｃｌｅを仮定する。次いで、係るＯｒａｃｌｅは、ＤＨ問題を解決するために使用できる。単にＧ、ａＧ、ｂＧ、及びｃＧを入力すると、Ｏｒａｃｌｅはｂ＝ｃ／ａであるのか、それともｃ＝ａｂであるかどうかを判断する。

【０１９７】

したがって、実際には、アリスは自分のシークレットからの自分の選択のＰＩＮ　αを抽出し、それをトークン部分（ｓ－α）Ａ及びＰＩＮ部分αＡに分割できる。これらが互いに加算されるとき、完全なシークレットを再構築できることは明らかである。取り込まれたトークンはどのような考えられるＰＩＮとも互換性があるため、それがない場合は無駄である。

【０１９８】

ただし、この特性を保持する必要がある場合には非常に注意深くなくてはならない。この特性を保持することが所望される場合、ＰＩＮ抽出は、例えば、Ｂｏｎｅｈ及びＦｒａｎｋｌｉｎのＩＢＥ方式［Ａ４］、またはＣｈｅｎ及びＫｕｄｌａの認証鍵交換［Ａ５］（プロトコル１）、またはＧ２でその公開パラメータの一部として発生元の点Ｐ及びP_{pub}＝ｓＰを必要とする任意の方式（または例えば［Ａ２９］のＰＳＣＡｂ方式等、タイプ－１ペアリングで実装される任意の方式）等、多くのペアリングをベースとしたプロトコルと使用できない。係る場合、チャーリーが（ｓ－α）Ａを含んだアリスのトークンを取り込む場合、チャーリーは、

ｅ（（ｓ－α）Ａ＋ｇＡ，Ｐ）＝ｅ（Ａ，ｓＰ）

まですべてのｇを試験することによって迅速にアリスのＰＩＮを見つけることができるだろう。

【０１９９】

これは、オフライン辞書攻撃の別の例であり、それはきわめて致命的である。

【０２００】

サーバシークレットｓＳがかつて漏れている（すなわち、Ｓ及びｓＳを明らかにしている）、または実際にはＧ２でｓによって既知の点の任意の倍数である場合、次いでそれらの値は盗まれたトークンと関連付けられたＰＩＮを決定するために使用できる。言うまでもなく、これが可能であるべきであることは常識にすぎない。サーバシークレットが発見されると、発見者は自身の偽のサーバをセットアップし、盗まれたトークンに対して考えられるあらゆるＰＩＮを試し、このようにしてＰＩＮを発見できる。

【０２０１】

同じｓを使用する同じＴＡによって確証された任意のサーバにアクセスするすべてのクライアントが、それらのサーバの内のたった１台が不正アクセスされると危険にさらされることに留意されたい。このため、理想的には各個別サーバは異なるマスターシークレットｓと関連付けられるべきである。

【０２０２】

興味深いことに、サーバがＰＩＮに対するオフライン辞書攻撃に着手し、有用な特徴としてそれを利用できる状況を故意に作り出すことができる。サーバはクライアントに単純なＰＩＮ回復サービスを提供する（したがって我々の条件２を完全に満たす）。クライアントＡは安全なチャネルを介してサーバＳに対しＸ＝Ｈ（ｅ（（ｓ－α）Ａ＋ｇＡ，Ｓ））を送信し、これにはトークン及びＰＩＮｇの彼らの正しくない予想値だけが必要となる。次いで、クライアントは自らのアイデンティティ（母親の結婚前の名前等）を証明するためにどんなことでもすると推定される。いったんサーバの満足の行くようにそれが行われると、サーバはオフラインになり、サーバが一致Ｘ　＝　Ｙを得るまで考えられるすべてｉのためにＹ＝Ｈ（ｅ（Ａ，ｓＳ－ｉＳ））を計算する。それゆえｉ＝α－ｇとなる。予想値ｇと正しいＰＩＮαとの間のこの差異は、次いでｅメールでまたは他のなんらかの方法でクライアントに送り返すことができる。これがサーバにＰＩＮαを明らかにしない

ことに留意されたい。

【０２０３】

信頼機関のマスターシークレットが全体的なシステムでの故障の単一障害点を表すことが観測されてよい。ただし、上述されたように、このマスターシークレットは多くの独立した公的機関全体で容易にシークレット共有できる［Ａ４］。最も簡単な場合、ＴＡの対はそれぞれ独立してシークレットs_1及びs_2を生成し、クライアントに対し、$sA=s_1A+s_2A$を作成するためにクライアントによって加算できるs_1A及びs_2Aを発行する可能性がある。ハードウェアセキュリティモジュール（ＨＳＭ）－一般にＰＫＩ秘密鍵を保護するために使用される－が、追加のセキュリティのレベルを提供するためにこれらの計算の一方または両方に使用できることに留意されたい。ここで、マスターシークレットｓはいずれの単一のエンティティにも既知ではない。このようにして、正式に条件１０を満たすことができる。

【０２０４】

ＳＯＫベースのプロトコル

信頼機関公開鍵がこのプロトコルで必要とされないことを観察することが重要である。独占的にG_1にクライアントを、及び独占的にG_2にサーバを入れる考え方と組み合わされて、表１の試行されたＰＫＩベースの解決策と関連付けられた問題の両方を乗り切る。

【０２０５】

トークンを盗み、ＰＩＮを知らないでサーバにログオンを試みるアタッカーを考える。サーバはｅ（Ａ，ｓＳ）から正しく鍵を導出するのに対し、アタッカーはｅ（（ｓ－α）Ａ，Ｓ）から鍵を導出することができる。サーバが次いで正しい鍵で暗号化された何かをアタッカーに送信しようとした場合、アタッカーは、アタッカーが有効な暗号解読を提供する値ｇを見つけるまでｅ（（ｓ－α）Ａ＋ｇＡ，Ｓ）から導出される考えられるあらゆる鍵で解読しようと試みることによって、オフライン辞書攻撃を介してＰＩＮを発見できる。これを防ぐために、クライアントに最初に強制的にクライアントの計算された鍵から導出される認証符号を提出させる。サーバが、クライアントが（正しいＰＩＮを使用して）正しい鍵を導出したと納得するときだけ、プロトコルが続行される。

【０２０６】

ここで、簡略なＳＯＫをベースにした解決策を提示する準備ができている。プロトコルが成功すると仮定し、両側とも鍵Ｋを使用するために進む。表２を参照すること。

アリス - アイデンティティID_a	サーバ - アイデンティティ ID_s
$ID_a \rightarrow$	$\leftarrow ID_s$
$S = H_2(ID_s), A = H_1(ID_a)$	$A = H_1(ID_a), S = H_2(ID_s)$
$k = e((s-\alpha)A + \alpha A, S)$	$k = e(A, sS)$
$K = H(k)$	$K = H(k)$
$M = H(ID_a \mid ID_s \mid K)$	$N = H(ID_a \mid ID_s \mid K)$
$M \rightarrow$	$M \neq N$の場合、接続を中断する

表２．簡略なＳＯＫベースのプロトコル

【０２０７】

次に、１０の特性に対照してこの簡略なプロトコルを試験する。すでに示されているように、特性２及び１０を満たすことは成功した。また、特性１及び９も満たす。クライアントのトークンに単にＡの倍数を加算する／クライアントのトークンから単にＡの倍数を減算することによって、クライアントがクライアントのＰＩＮをローカルで変更できることは明らかであり、したがって４も満たされる。

【０２０８】

条件８は十分に満足できる‐Ｂｏｎｅｈ及びＦｒａｎｋｌｉｎ［Ａ４］によって指摘されるように、クライアントシークレットの（クライアントアイデンティティの倍数として

の）簡略な形は、それがシークレット共有方式を使用してさまざまな方法で安全に分割できることを意味する。クライアントの再構築されたシークレットが少量分オフである場合も、相互鍵はまだ、エラーの範囲を決定し、エラーを相殺できるサーバによって計算できる。例えば、クライアントがそのシークレットとしてｓＡ＋εＡを使用する場合、サーバはサーバのシークレットとしてｓＳ＋εＳを使用することによって相殺できる。さらに、サーバは権利εを検索するなんらかの時間を費やしても差し支えない。したがって、この方式は、特性６によって考えられるバイオメトリック因子をサポートするために必要とされる鍵補正機能をサポートする。発明者は、同じ挙動を他に利用できることがわかった。クライアントが間違ったＰＩＮを入力した場合、サーバはこの鍵補正機能を使用して容易にエラーδの範囲を決定できる。したがって、例えばクライアントが１２３４の代わりに１２２４と入力した場合には、サーバはＰＩＮが１０、「外れ」ていたと見当をつけることができるだろう。また、この挙動は入力されている間違ったＰＩＮにインテリジェントに対応するためにも利用できる‐それがかなり外れている場合、別の試行を許さず、それが１桁しか外れていない場合には、追加の試行を許す等である。例えば、ＰＩＮが最後の数字で１だけ外れていた場合、サーバはこれを、クライアントが物理的な脅威を受ける間にＰＩＮを入力としたと解釈し、適切に応答する可能性がある等、「強制」規約に同意できるだろう。

【０２０９】

それによって、特性３、５、及び７はまだ満たされないままである。サーバは任意のクライアントＣの鍵をｅ（Ｃ，ｓＳ）として導出できるので、サーバがログインするために任意のクライアントになりすますことができることは明らかであり、このことは３及び５に違反する。複数の他の問題もある。クライアント及びサーバの同じ対はつねに同じ鍵ｋを導出するため、リプレー攻撃を受けやすい。アリスのトークンを取り込み、暗号化された会話を盗聴するアタッカーは、関連付けられたＰＩＮを容易に導出できる。

【０２１０】

チャレンジは、すでに達成した特性を失うことなくこれらの攻撃を遮断し、残りの特性を満たすことである。主要な考え方は、ＳＯＫ構成を別のいくぶんより詳細な構成で置き換えることである。

【０２１１】

Ｗａｎｇの提案

任意の認証鍵交換プロトコルの強力な特性は鍵漏洩なりすまし（ＫＣＩ）攻撃に対する抵抗である［Ａ３］。鍵漏洩なりすまし攻撃では、アリスのクレデンシャル（トークンとＰＩＮ）を取り込むアタッカーのチャーリーがアリスであるふりをしてサーバにログインする立場にいるだけではなく（このことは予想されるにすぎない）、本物のアリスに対してもサーバであるふりができる。サーバがｅ（Ａ，ｓＳ）として鍵を計算できる一方、アリスのクレデンシャルを盗んだチャーリーがｓＳを知らなくてもｅ（ｓＡ，Ｓ）と同じ値を計算できるので、我々のＳＯＫベースのプロトコルはこの種の攻撃を広く受けやすい。問題は、マスターシークレットｓを含んだ構成要素をペアリングの一方の側から他方の側に移すために双線型性を利用できる能力である。解決策は、ペアリングの各側を他の必要な計算で「オーバーロードし」、したがってこの可能性をブロックすることである。

【０２１２】

（鍵交換が通常説明されるピアツーピア設定よりむしろ）クライアントサーバ設定では、ＫＣＩは、サーバがクライアントを装うことができるサーバ対クライアントＫＣＩ、またはクライアントがサーバを装うことができるクライアント対サーバＫＣＩ、または両方のケースが可能である相互ＫＣＩに分類できる。一方または他方が可能であるが、両方は可能ではないシナリオがあるだろうという要点。ここでは両方の可能性を阻止することを好む。

【０２１３】

［Ａ２８］で、Ｗａｎｇは効率的なアイデンティティベースの認証鍵共有プロトコルを提案している。プロトコルはピアツーピア設定で説明されているが、該プロトコルをタイ

プ‐３ペアリングで実装することによって該プロトコルをクライアント‐サーバプロトコルに適応させる。元のプロトコルは決定バイリニアディフィーヘルマン（ＤＢＤＨ）仮定に基づくセキュリティの証拠を有する。該仮定はここでは特性３及び５に対するサポートを提供する一方で、ＳＯＫプロトコルの当座の代替を提供する。Ｗａｎｇのプロトコルは、Ｐ１３６３．３によって提案されるＩＥＥＥ規格の部分を形成する［Ａ１］。

【０２１４】

Ｗａｎｇのプロトコルのはるかに簡略化されたバージョン‐表３で開始する。両方の関係者にとって、一致した鍵ｋ＝ｅ（Ａ，Ｓ）^{sxy}である。ＷａｎｇのプロトコルはＫＣＩ攻撃の影響を受けやすくない。しかし、P_a及びP_sを盗聴した信頼機関は、鍵をｅ（P_a，P_s）^{s}として容易に計算できる。これは、完全前方秘匿性（ＰＦＳ）の特性（我々の７番）を有さないプロトコルの直接的な結果としてである。

アリス‐アイデンティティID_a	サーバ‐アイデンティティID_s
ランダム$x < q$を生成	ランダム$y < q$を生成
$ID_a \rightarrow$	$\leftarrow ID_s$
$S = H_2(ID_s), A = H_1(ID_a)$	$A = H_1(ID_a), S = H_2(ID_s)$
$P_a = xA$	$P_s = yS$
$P_a \rightarrow$	$\leftarrow P_s$
$k = e(x.((s-\alpha)A + \alpha A), P_s)$	$k = e(P_a, y.sS)$
$K = H(k)$	$K = H(k)$
$M = H(ID_a \mid ID_s \mid P_a \mid P_s \mid K)$	$N = H(ID_a \mid ID_s \mid P_a \mid P_s \mid K)$
$M \rightarrow$	$M \neq N$の場合、接続を中断する

表３．［Ａ２８］に基づく簡略化されたプロトコル

【０２１５】

特性ＰＦＳを含めるために、Ｗａｎｇは鍵交換の中に縦一列のディフィーヘルマンを含むこの修正を提案している‐表４。［Ａ５］も参照すること。

アリス‐アイデンティティID_a	サーバ‐アイデンティティID_s
ランダム$x < q$を生成する	ランダムy，$w < q$を生成する
$ID_a \rightarrow$	$\leftarrow ID_s$
$S = H_2(ID_s), A = H_1(ID_a)$	$A = H_1(ID_a), S = H_2(ID_s)$
$P_a = xA$	$P_s = yS, P_g = wA$
$P_a \rightarrow$	$\leftarrow P_s, P_g$
$k = e(x.((s-\alpha)A + \alpha A), P_s)$	$k = e(P_a, y.sS)$
$K = H(k \mid xP_g)$	$K = H(k \mid wP_a)$
$M = H(ID_a \mid ID_s \mid P_a \mid P_s \mid P_g \mid K)$	$N = H(ID_a \mid ID_s \mid P_a \mid P_s \mid P_g \mid K)$
$M \rightarrow$	$M \neq N$の場合、接続を中断する

表４．完全前方秘匿性のある簡略化されたプロトコル

【０２１６】

両方の鍵とも同じＫ＝Ｈ（ｅ（Ａ，Ｓ）^{sxy}｜ｘｗＡ）である。Ｗａｎｇの最初の論文［Ａ２８］では、ｗ＝ｙを使用することが提案されている。しかしながら、これはｅ（ｘｓＺ，ｙＳ）＝ｅ（ｘｙＺ，ｓＳ）であるという観察に基づいて鍵漏洩なりすまし攻撃を再有効化し、したがってｓＳが取り込まれると、チャーリーは任意のアイデンティティＺでＳにログインできる。

【０２１７】

二因子クライアント‐サーバプロトコル

それについて上述したように、Ｗａｎｇのプロトコルは問題を有している。Ａ（及びＳ）のアイデンティティは鍵の計算に直接的に含まれていない。したがって、ボブはアイデンティティアリスを主張することができるが、自分自身のクレデンシャルを使用してログオンすることもできる。これはいくつかのアプリケーションでは十分ではない。分かるように、Ｗａｎｇはこれを修正するためにやや複雑な方法を使用している。Ｈｑ（．）は１からｑの範囲の数にハッシュするハッシュ関数である（しかし、Ｗａｎｇによると、この範囲をｑのビット数の半分のサイズまで削減することは大丈夫であり、いくぶん性能向上がある）。新しい方式は表５に示される。

アリス - アイデンティティ ID_a	Server - identity ID_s
ランダム $x < q$ を生成する	ランダム $y,\ w < q$ を生成する
$ID_a \rightarrow$	$\leftarrow ID_s$
$S = H_2(ID_s), A = H_1(ID_a)$	$A = H_1(ID_a), S = H_2(ID_s)$
$P_a = xA \rightarrow$	$\leftarrow P_s = yS,\ P_g = wA$
$\pi_a = H_q(P_a \mid P_s \mid P_g),\ \pi_s = H_q(P_s \mid P_a \mid P_g)$	$\pi_s = H_q(P_s \mid P_a \mid P_g),\ \pi_a = H_q(P_a \mid P_s \mid P_g)$
$k = e((x+\pi_a).((s-\alpha)A + \alpha A), \pi_s S + P_s)$	$k = e(\pi_a A + P_a, (y+\pi_s)sS)$
$K = H(k \mid xP_g)$	$K = H(k \mid wP_a)$
$M = H(ID_a \mid ID_s \mid K)$	$N = H(ID_a \mid ID_s \mid K)$
$M \rightarrow$	$M \neq N$の場合、切断を中断する

表５．二因子認証プロトコル

【０２１８】

両方の当事者に対して、

$$k = e(A,S)^{s(x+\pi_a)(y+\pi_s)}$$

であることを観察する。アリスのトークン及びＰＩＮが、それらから計算される任意の値が送信される前にローカルで再結合されることを観察する。間違ったＰＩＮが入力されると、サーバは接続を中断（し、自称「アリス」に関してより思い切った対応を取る前にさらに数回の試行だけを可能に）する。

【０２１９】

ペアリングベースのプロトコルは現実世界のアプリケーションには本質的に遅すぎる可能性があるという不安が広がっている。ただし、効率的な実装の最近の進歩は疑惑を抱く人に明確に回答している（最近の進歩のレビューについては、［Ａ２１］を参照すること）。Ｃ及びなんらかの自動的に生成されるアセンブリ言語の混合物を使用して、クロックが２．４ＧＨｚである６４ビットのＩｎｔｅｌ　ｉ５　５２０Ｍプロセッサで標準高度暗号化標準‐１２８（ＡＥＳ－１２８）ビットレベルのセキュリティで、Ｂａｒｒｅｔｏ－Ｎａｅｈｒｉｇ（ＢＮ）曲線［Ａ２］を使用して表５のプロトコルを実装した。

【０２２０】

ペアリング自体のために最適技法を使用するだけではなく［Ａ２１］、G_2［Ａ８］への高速ハッシュのためのＦｕｅｎｔｅｓ－Ｃａｓｔａｎｅｄａら方法［Ａ８］、G_1での点乗算のための周知のＧａｌｌａｎｔ、Ｌａｍｂｅｒｔ、及びＶａｎｓｔｏｎｅ（ＧＬＶ）方法［Ａ１１］、及びG_2での高速点乗算のためのＧａｌｂｒａｉｔｈ－Ｓｃｏｔｔ技法［Ａ１０］も利用できる。

【０２２１】

事前計算［Ａ２１］なしでも、上記プロトコルは十分に実用的である。試験ハードウェアで、計算のサーバ側は４．４８ミリ秒を要し、クライアント側は４．１０ミリ秒を要し

た。
【０２２２】
Ｗａｎｇのプロトコルはここでは唯一の選択肢ではないことに留意されたい。例えば、やはりＰＩＮ及び鍵補正をサポートするＳ．Ｗａｎｇらによる代替プロトコル［Ａ２７］がある。それはＷａｎｇのプロトコルよりさらに簡潔であり、より高速であると主張している。また、それは素晴らしく且つ簡略なセキュリティの証拠も有する。別の代替策は、Ｃｈｅｎ及びＫｕｄｌａ［Ａ５］からのプロトコル２である。
【０２２３】
サーバのシークレットｓＳを明らかにするサーバに対するハッキングの成功の意味を考えてみる。サーバに記憶されているクライアントに関係するシークレット（トークン、ＰＩＮ、またはバイオメトリック）がないことに留意されたい。ただし、この攻撃は明白に偽のサーバをセットアップできるようにし、クライアントは疑わずに該サーバ上にログオンする可能性がある。次いで成功したハッカーがトークンを取り込むと、ハッカーは関連付けられたＰＩＮを容易に見つけることができる。ただし、ハッカーは取り込んだサーバシークレットを使用して本物のサーバにログオンし、サーバのデータにアクセスすることはできない。ハッカーは任意の他のクライアントを登録できない。したがって、我々は特性５を完全に達成し、サーバシークレットの損害は重大であるが、影響は最大限に緩和される。
【０２２４】
クライアント記載の役割をサーバから削除し、代わりに役割を独立したサードパーティにゆだねることによって、二因子認証の問題に対するＰＫＩ／ＳＳＬのような解決策を可能にしてきた。我々は、表５に示される方式が、同時によりユーザーフレンドリ且つより安全であるので、現在ユーザー名／パスワードを使用している任意のウェブベースのアプリケーションにとって非常に適した代替策であると主張する。ただし、該方式は、サーバだけではなくクライアントもペアリングを実行することを必要とする。
【０２２５】
上述されたプロトコルを改良する、及び／または上述されたプロトコルの代替策を提供するプロトコルのいくつかの例がここで説明される。プロトコル及び実装のいくつかは上記に一覧された１０のセキュリティ特徴のすべてではないが、いくつかを満たす。
【０２２６】
Ｍ－Ｐｉｎプロトコル
効率的な二因子クライアント‐サーバ認証機構を特徴とするいくつかのＭ－Ｐｉｎプロトコルがここで説明される。特に、Ｍ－Ｐｉｎクライアントは多くの場合、計算機能が限られた環境で実装されるので、プロトコルのクライアント側にかかる負荷を軽減する方法を示す。ここでの説明は、ＡＥＳ－１２８ビットレベルのセキュリティでのＢａｒｒｅｔｏ－Ｎａｅｈｒｉｇ（ＢＮ）曲線［Ｂ４］との関連であるが、ここに含まれている考えは容易に他の状況に拡張する。
【０２２７】
上述されたように、Ｍ－Ｐｉｎはサーバに対してクライアントを認証するゼロ知識認証プロトコルである。Ｍ－Ｐｉｎの固有の特徴は、Ｍ－Ｐｉｎがクライアントシークレットから短いＰＩＮ番号１１を抽出して、トークン＋ＰＩＮ組合せを作成し、二因子認証を容易にできる点である。該考え方は、多因子認証をサポートするために容易に拡張できる。さらに、クライアントシークレット７から抽出される第２の因子１１はＰＩＮである必要はない。第２の因子１１は任意のデータ列であることがある。
【０２２８】
強力なクライアント‐サーバプロトコルは、（ａ）サーバに対してクライアントを認証する、（ｂ）クライアントに対してサーバを認証する必要があり、（ｃ）以後の通信を暗号化できる交渉済みの暗号鍵を生じさせる必要がある。上述されたように、今日までこれを実装する標準的な方法は、サーバに対してクライアントを認証し、周知のＴＬＳ／ＳＳＬプロトコルを使用してクライアントに対してサーバを認証し、暗号鍵を確立するために

ユーザー名／パスワードの仕組みを使用することによっていた。ここで最弱のリンクは、壊れていると広く見なされているユーザー名／パスワードの仕組みである。ＳＳＬ自体、より少ない程度に、なんらかの利用可能な脆弱性を明らかにした集約的な詳細な調査によって弱められている。

【０２２９】

ユーザー名／パスワードを置き換えるために、多因子認証は最も頻繁に勧められている解決策である。すべての考えられるフォームファクタの内、トークン及びＰＩＮ番号の簡略なＡＴＭ状の組合せが最もユーザーになじみがあり、ユーザーフレンドリである。しかし、ごく最近まで、それをサポートするために、旧い対称暗号法に由来する単純化した方法を使用したプロトコル以外、適切な純粋に暗号法のプロトコルは利用できなかった。実装はつねになんらかの種類の（潜在的に高価な）ハードウェアトークンを含んでいた。現代的な非対称暗号法に基づく暗号解決策は、なんらかの容易にアクセス可能な側の情報を所有し、ソフトウェアトークンを取り込んだアタッカーが、関連付けられたＰＩＮ番号を数学的に即座に決定できるようにした弱点を抱えていた。したがって、該暗号解決策は真に二因子ではなかった。

【０２３０】

ここで、Ｍ－Ｐｉｎ技術解決策を説明する。これは２つの特徴、つまり（ａ）ＳＳＬを利用し続けながらユーザー名／パスワードを単に置換する基本的なＭ－Ｐｉｎプロトコル、及び（ｂ）ＳＳＬも置換するＭ－Ｐｉｎ　Ｆｕｌｌ変形に分けられる。

【０２３１】

図１に関して説明されるように、Ｍ－Ｐｉｎの重要な態様は、図１の説明で信頼機関４と呼ばれるサードパーティの関与である。旧いユーザー名／パスワード方式では、クライアントは、クライアントパスワードの「暗号化された」（事実上ハッシュされた）ファイルを維持するサーバと直接的に登録する。したがって、サーバは単に日々の業務に責任があるだけではなく、サーバはクライアント登録にも責任がある。これは、Ｖｅｒｉｓｉｇｎ等のすでに証明書発行機関（ＣＡ）の形をとるサードパーティを含む（はるかに成功している）ＳＳＬプロトコルと対照的であることに留意されたい。上述されたように、この証明書をベースにした技術は公開鍵基盤（ＰＫＩ）として知られている。ＳＳＬの同類を用いて、サーバ登録は日々の業務から完全に分かれている。Ｍ－Ｐｉｎを用いると、クライアントに対して類似したことを行う。つまり、日々のサーバの機能性を、現在では信頼機関（ＴＡ）によって処理されているクライアント登録から取り出す。したがって、ＣＡがＳＳＬ／ＰＫＩに対するの同様にＴＡはＭ－Ｐｉｎに対する。この手法の膨大な利点の内の１つは、ＴＡが破られていないままである限り、サーバに押し入ることはほぼ損害を生じさせないという点である。ハッキングと全く同じように、単一のＳＳＬサーバはＶｅｒｉｓｉｇｎをハッキングするほど悪くない。

【０２３２】

ＴＡはクライアントシークレット及びサーバシークレットを発行する。Ｍ－Ｐｉｎを使用すると、クライアントシークレット、つまりクライアントシークレットから導出される値はサーバに記憶されない。Ｍ－Ｐｉｎを用いて、アリスは、有効なシークレットについて何も明らかにしないで、アリスが有効なシークレットを所有していることをサーバに対して証明する。

【０２３３】

ペアリングベースの暗号法

さらに上述されたように、解決策を実現するために、ペアリングベースの暗号法（ＰＢＣ）の相対的に新しい科学を利用することになるだろう。ペアリングベースクリプト（Ｐａｒｉｎｇ－Ｂａｓｅｄ　Ｃｒｙｐｔｏ）は、多くの場合、公開鍵暗号法の標準的な数学にとって解決困難であることが判明した複雑な問題に対する解決策を可能にする追加の構造を提供する。ＰＢＣのポスターチャイルドは、クライアントのアイデンティティがその公開鍵となるアイデンティティベースの暗号化であった。この考え方は長い間あったが、従来の暗号プリミティブは再三にわたり解決策を生み出すことができなかった。しかしな

がら、ＰＢＣが導入されて、解決策が見つかった［Ｂ６］。

【０２３４】

やはり上述されたように、タイプ‐３ペアリング［Ｂ１２］は、マッピング$G_1 \times G_2 \rightarrow G_T$である。群$G_1$、$G_2$、及び$G_T$は、すべて同じ素数位数ｑである。ペアリングは特別なペアリングにフレンドリな楕円曲線［Ｂ４］、［Ｂ１１］に作用する。ＢＮ曲線の場合、G_1は基本フィールドF_p上の曲線上の点であり、G_2は二次拡張フィールド

$$F_{p^2}$$

上の曲線の６次ツイスト上の点であり、G_Tは有限拡大フィールド

$$F_{p^{12}}$$

に埋め込まれた円分の下位群内の要素である。パラメータｐはペアリングの素数係数を示す。ペアリング自体は、２つの入力Ｃ＝ｅ（Ｐ，Ｑ）を有する関数として作成され、Ｐ∈G_1、Ｑ∈G_2、及びＣ∈G_Tである。ＢＮペアリングフレンドリ曲線がＡＥＳ－１２８レベルでのセキュリティにぴったり合い、したがってここでのその使用を仮定することが理解される。上述されたように、ペアリングの最も重要な特性は、ペアリングの双線型性である。

$$e(aA, bB) = e(bA, aB) = e(A, B)^{ab}$$

【０２３５】

クライアント‐サーバプロトコルを作成するためには、クライアントシークレット及びサーバシークレットを別個に保たなければならないことが重要である。タイプ‐３ペアリングの構造を利用する簡単な方法は、G_1にクライアントシークレットを、及びG_2にサーバシークレットを入れることである。タイプ‐３ペアリングの場合、両方とも同じ位数であったとしてもこれらの群間に計算可能な同型マッピングはない。

【０２３６】

図１から図５に関して言及されたように、信頼機関はマスターシークレットｓを所有する。マスターシークレットｓはＦｑの確率要素であってよい。一意のシークレット値ｓは、信頼機関の特定のサーバに対するサポートと関連付けられてよい。クライアントシークレットは、ｓ．Ｈ（ＩＤ）の形式となり、ＩＤはクライアントアイデンティティであり、Ｈ（．）はG_1上の点Ａにマッピングするハッシュ関数である。ここで、［Ｂ２０］に従い、Ｈがランダムオラクルとしてモデル化され、$H(ID) = r_{ID} \cdot P$であり、r_{ID}はランダムであり、ＰはG_1の固定生成元であると仮定する。サーバはシークレットｓＱを発行され、ＱはG_2の固定生成元である。言い換えると、シークレットｓＱは、ＴＡシークレットｓで乗算される特別な楕円曲線上の固定点Ｑを表す。固定点Ｑは、ＴＡによって任意に選ばれてよい。また、サーバは、固定点Ｑの座標も発行される。たとえハッカーがｓＱ及びＱを手に入れても、マスターシークレットｓは離散対数問題により保護されているので、ハッカーはマスターシークレットｓを決定できないだろう。いくつかの実装では、これはＴＡによってかつて提供されたG_2のｓの唯一の倍数となる。いくつかの実装では、サーバはつねに独自の一意のマスターシークレットと関連付けられる。言い換えると、ｓの値はあらゆるサーバにとって一意である。曲線はその上に点Ａが位置する同じ曲線であるが、点Ａ及びＱは同じ位数ｑの異なる群G_1及びG_2に属する。

【０２３７】

ＴＡの機能性は、全体的なシステムから単一障害点を削除するために、シークレット共有方式を使用して分散できることに留意されたい。考えられる最も簡略なケースでは、それぞれが独立してマスタ鍵の独自の分担を維持する２つの分散信頼機関（ＤＴＡ）があることがある。したがって、$s = s_1 + s_2$であり、各ＤＴＡは、クライアントがその完全なシークレットを形成するためにともに加算する部分‐クライアントシークレット$s_1 H(ID)$及び$s_2 H(ID)$を発行する。ここで、１つのＤＴＡが不正アクセスされても、クライアントシークレットは依然として安全である。

【０２３８】

重要な考え方は、ＰＩＮ番号αが、トークンを作成するためにクライアントシークレットからクライアントによって抽出される点である［Ｂ１６］。したがって、認証の２つの因子は、この記憶されているＰＩＮ及び残りのトークンである。明白にするために、アイデンティティが点$A \in G_1$にハッシュする個人のアリスの場合、ＴＡによって発行されるアリスの完全なシークレットは点ｓＡである。これは、トークン（ｓ－α）Ａ、及び選ばれたＰＩＮ番号αに分割される。トークンは、αＡを計算し、αＡをｓＡから減算することによって作成される。完全なシークレットは、ｓＡ＝（ｓ－α）Ａ＋αＡとしてアリスによってその２つの構成要素から再構築できる。結果的に、トークン（ｓ－α）Ａは、いくつかの例では楕円曲線上の点であってよく、その点の座標を表す任意のデータ列となるだろう。図１に関して説明されるように、トークン１０はクライアントに記憶される。ＰＩＮ１１はユーザーによって覚えられており、ユーザー６がクライアント２に認証サーバ３とのセッションを確立してほしいときに入力される。ＰＩＮは、人間によって選ばれることもれ場、コンピュータによって自動的に生成されることもある。ただし、ＰＩＮは、いつでも他のどの当事者も巻き込むことなくクライアントによって変更できる。

【０２３９】

Ｍ－Ｐｉｎ

この提案されているプロトコルは、Ｆｉａｔ及びＳｈａｍｉｒ［Ｂ１０］の重要な著作物に端を発するアイデンティティベースの識別（ＩＢＩ）プロトコルの長い歴史に由来する。基本的な考え方は、証明者が、検証者にそのシークレットの何も明らかにしないが、信頼機関によって発行されるアイデンティティに関係するシークレットを使用して自身を識別することである。シークレットを所有することの証拠は、シークレット自体の何も明らかにしない間に確立されるので、これは多くの場合ゼロ知識証明と呼ばれる。識別プロトコルにより、検証者（この場合はサーバ）が証明者（この場合はクライアント）を受け入れる、または拒否するのどちらかとしかならない点を強調することが重要である。

【０２４０】

ペアリングベースのＩＢＩプロトコルは、多くの標準的な識別（ＳＩ）方法がＩＢＩ方式、つまり実際には完全アイデンティティベースシグナチャ（ＩＢＳ）方式として「浮上」できるより大きなフレームワークの部分としてＢｅｌｌａｒｅら［Ｂ５］によって徹底的に研究された。いくつかの提案された（が、証明されていない）ＩＢＳ方式をその根本的なＳＩ説明まで「押し」下げることによって、彼らは今回は、完全なセキュリティの証拠をもってであるが、それらを再浮上させることができた。彼らはなんらかのペアリングベースの方式を考え、その内の最善の方式は、最初に［Ｂ８］及び［Ｂ２４］によって無関係に提案されたＩＢＳ方式として出現した方式のようである。（［Ｂ５］とは）無関係に、Ｋｕｒｏｓａｗａ及びＨｅｎｇ［Ｂ１３］は、多かれ少なかれ同じ考え方を考え付いた。この研究のすべては、Ｂｏｎｅｈ、Ｌｙｎｎ、及びＳｈａｃｈａｍ［Ｂ７］の元の（アイデンティティに基づいていない）ショートシグナチャ方式のおかげである。

【０２４１】

ＩＢＩ方法は、ここでは開始点として利用するものでる。標準的なＩＢＩプロトコルでは、検証者はシークレットを有さないことに留意されたい。証明者は単に、証明者が証明者の主張しているアイデンティティに対して資格があることを、関心がある誰かに証明しようとしているだけである。ただし、以下に説明されるいくつかの例では、シークレットｓＱを所有している一意の検証者だけが検証を実行する立場にある。これを行うことによって、堕落した検証者は、関連付けられたトークンを盗んだ誰かによるＰＩＮ予想値を試験するためのＯｒａｃｌｅとして使用することはできない。したがって、ＳＳＬまたはその同等物を実行中のサーバは、ｓＱを保護することに対しても責任を負っていると仮定する。サーバはデータストレージ３３の安全なストレージにｓＱを記憶してよい。我々が加える別の変更は、上述されたように、クライアントシークレットをトークン及びＰＩＮに分割することである。

【０２４２】

Ｍ－Ｐｉｎプロトコルは表６に示されている。クライアント側では、計算のすべてがよ

り簡略な群G_1にあることを観察する。サーバ側では、２つのペアリングの積を、標準的なマルチペアリング方法［Ｂ１７］を使用して計算することができ、２つの別々のペアリング計算よりもはるかに費用が少なくなる。プロトコルの正確さは、双線型性特性を使用して迅速に立証できる。

アリス - アイデンティティID_α	サーバ
ランダム$x < q$を生成する	ランダム$y < q$を生成する
$A = H_1(ID_\alpha)$	$A = H_1(ID_\alpha)$
$U = xA$	
$ID_\alpha, U \rightarrow$	$\leftarrow y$
$V = -(x + y)((s - \alpha)A + \alpha A) \rightarrow$	
	$g = e(V,Q).e(U + yA, sQ)$
	$g \neq 1$の場合、接続を拒否する

表６．Ｍ－Ｐｉｎ

【０２４３】

双線型性を使用すると、以下を示すことができる。

ｇ＝ｅ（Ｖ，Ｑ）．ｅ（Ｕ＋ｙＡ，ｓＱ）

＝ｅ（－（ｘ＋ｙ）ｓＡ，Ｑ）．ｅ（ｘＡ＋ｙＡ，ｓＱ）

＝ｅ（Ａ，Ｑ）$^{-(x+y)s}$．ｅ（Ａ，Ｑ）$^{(x+y)s}$

結果的に、クライアントが完全なクライアントシークレットを再構築できるようにするために、正しいＰＩＮがクライアントで受信されると、ｇは１に等しくなる。

【０２４４】

表６のプロトコルは、ここで図６及び図７に関してより詳細に説明される。図６に関して、ステップ６．１で、クライアント２はＰＩＮ予想値１１’をユーザーから受信する。表６に示されるように、ステップ６．２で、クライアントは次いでそのアイデンティティをＧ$_1$の楕円曲線上の点Ａにハッシュする。アイデンティティはストレージ２３から取り出されてよい。ステップ６．３で、クライアントはランダムｘも生成し、別の点Ｕに対し、楕円曲線上で点Ａの乗算を実行する。楕円曲線上の乗算及び加算は当業者によって知られており、ここでは詳細に説明されない。ｘの値は群Ｇ$_1$、Ｇ$_2$、及びＧ$_T$の素数位数ｑよりも小さい。ステップ６．４で、クライアントは、クライアントが認証を希望するサーバ３にコミットメントを発行し、そのアイデンティティ及び曲線上の点Ｕの座標を送信する。クライアントは次いでステップ６．５でサーバからチャレンジｙを受信し、ｙもｑより小さい乱数である。ステップ６．６で、クライアントは次いでＰＩＮ及びトークンからクライアントシークレットを取り戻し、楕円曲線上で、クライアントシークレットに相当する点を、－ｘ及び－ｙの合計で乗算して、別の点Ｖの座標を得る。クライアントは次いで、サーバがクライアントを認証できるようにするためにこれらの座標をサーバに送信する。ステップ６．７で、クライアントは認証プロセスの結果の通知をサーバから受信してよい。これは、以下により詳細に説明される。

【０２４５】

図６のステップは、図２に関して説明されるように、認証プログラム２５の命令を実行するクライアント２のプロセッサ２１によって実施されてよい。

【０２４６】

図７に関して、サーバ３は、クライアントの主張されているアイデンティティ及び楕円曲線上の点Ｕの座標を含んだ、ステップ７．１で認証するというコミットメントを受信する。また、サーバは、ステップ７．１の前にクライアントから、クライアントが接続を確立することを望んでいることを示す別のメッセージも受信した可能性がある。サーバは、Ｕのアイデンティティ及び座標を送信するようにクライアントに要請した可能性がある。ステップ７．２で、サーバは次いでやはりｑより小さいランダムｙを生成し、ｙの値をク

ライアントに対するチャレンジとして送信する。ステップ７．３で、サーバはクライアントアイデンティティを楕円曲線上の点Ａにハッシュする。ステップ７．４で、サーバは楕円曲線上の点Ｖの座標を受信する。

【０２４７】

サーバは次いでステップ７．５でペアリングの積を計算する。サーバは２つのペアリングを先に計算し、次に２つのペアリングを互いに乗算してよい。代わりに、サーバは、ペアリングの積をはるかに速く計算できるようにするマルチペアリングを計算してよい。第１のペアリングは、入力として点Ｖ及び点Ｑを採り、点Ｖ及び点ＱをG_Tの要素にマッピングする。図６の説明に示されるように、Ｖは、クライアントシークレットに相当する点から開始することによって得られるG_1での曲線上の点である。すでに上述されたように、ＱはG_2の固定生成元である。他のペアリングは、入力として、クライアントのアイデンティティに対応する点及びサーバシークレットｓＱに対応する点から開始することから得られるG_1の別の点の座標を入力として採り、該点をG_Tの要素にマッピングする。点ｓＱは言うまでもなくG_2の別の点である。G_1の点は、楕円曲線上でG_1の点Ｕを加算することによって得られ、その座標はステップ７．１のコミットメントでクライアントから受信され、言うまでもなくやはりG_1での点は乱数値ｙでクライアントアイデンティティに対応する点を乗算することによって得られる。ペアリングの積の結果は値ｇを出す。双線型性のため、正しいＰＩＮが入力され、したがって正しいクライアントシークレットが取り戻された場合は、ｇの値は１と等しくなるはずである。ステップ７．６で、サーバは、ｇの値が１に等しいかどうかをチェックし、１に等しい場合、サーバはステップ７．７でクライアントを認証し、ステップ７．８で接続を受け入れる。接続を受け入れることは、クライアントが認証されたことをクライアントに知らせるためにメッセージをクライアントに送信することを含んでよい。

【０２４８】

逆に、ｇの値が１に等しくない場合、サーバはステップ７．９でクライアントが認証されていないと判断する。サーバは、その場合先に進むためのいくつかのオプションを有する。ステップ７．１０で、サーバはただちに接続を拒否するか、またはサーバは、図８に関して以下に説明されるように、ＰＩＮが正しくなかったに違いない範囲を決定するためにエラー処理プロセスを実行できる。

【０２４９】

図７のステップは、図３に関して説明されるサーバ認証プログラム３４の命令を実行するサーバ３のプロセッサ３１によって実施されてよい。

【０２５０】

図６及び図７のステップのいくつかの順序を変更できることが理解される。例えば、ＰＩＮは、ＰＩＮがステップ６．６で使用されるまで受信されないことがある。さらに、サーバは、サーバがコミットメントを受信する前にｙを生成してよい、及び／またはサーバは、サーバが点Ｖの座標を受信した後にクライアントアイデンティティをハッシュしてよい。

【０２５１】

サーバは、当業者が理解するように、ペアリングを実行するために任意の適切なペアリングを使用してよい。例えば、クライアントは効率的なＲ－エイトペアリングを使用してよい。さらに、楕円曲線が使用されることが説明されてきたが、任意の適切な代数曲線が使用されてよい。例えば、楕円曲線の代わりに超楕円曲線が使用されてよい。

【０２５２】

プロトコル出力

上述されたように、プロトコル実行の最後に、クライアントは、クライアントが成功したのかどうかをまだ知らない。ここで不成功に終わった接続を処置を適切に講じることができるので、これは重要である。この責任は、使用可能なプロトコル出力によって知らされる別の非暗号プロセスに渡されてよい。完了後Ｍ－Ｐｉｎのようなプロトコルはどのような出力を返す必要があるのだろうか。ユーザー名／パスワードの場合、応答は通常ブー

リアン、つまり真か、それとも偽かである。結果は基本的にアクセスを許すか、それともアクセスを拒否するかのどちらかであるが、オンラインパスワード推測攻撃を阻止するためになんらかの簡略な機構も実施されている。

【０２５３】

結果が成功だった場合、なされるべきことはほとんどない。クライアントは信頼機関に正しく登録されたに違いないため、有効なトークンを発行され、クライアントの主張するアイデンティティと関連付けられた有効なＰＩＮを入力した。応答は、単に、接続が確立されるということであってよい。プロトコルが失敗した場合、サーバが適切な応答を策定するために協力することは相対的にほとんどないと思われる。ただし、いくつかの例では、サーバは、トークンを取り込んだアタッカーが、アタッカーが正しいＰＩＮにあたるまで考えられるすべてのＰＩＮを試さないようにするための三振即アウト戦略を実施してよい。サーバが必ずしも記載されたユーザーのリストを有していなくてもよいことに留意されたい。つまり、係るリストが信頼機関によって保守及び管理されるのは当然であろう。

【０２５４】

（ａ）サーバ３が、失敗したクライアントが有効なトークンを有していたかどうかを知っていた場合、及び（ｂ）サーバ３が入力されたＰＩＮのエラーの範囲を知っていた場合、適切な応答を決定することはサーバ３に役立つことがある。いくつかの実装では、ＰＩＮエラーの範囲を失敗したプロトコル実行から導出する方法がある。ＰＩＮエラーの範囲が有効なＰＩＮの範囲外にある場合には、これは「クライアント」が有効なトークンを有していないことを暗示する。

【０２５５】

Ｍ－Ｐｉｎプロトコルでは、クライアント及びサーバの両方ともそれらの計算に含まれるマスターシークレットｓの同じ値を有していることを観察する。クライアントが正しくないＰＩＮを入力する場合、それは、あたかもクライアントがその計算で代わりにｓ＋δを使用したかのようであり、δはそのエラーの範囲である。サーバは次いで、プロトコルの結果が正しくなるまでサーバの側でｓＱ＋δ．Ｑのすべての可能性を反復することによってδを検索できる。ペアリングの双線型性を利用することによって、これは候補δごとにG_Tで１回だけ乗算を必要とするように行うことができる。いくつかの例では、さらにうまくできる。プロトコルの最終ステップの結果、ｇ≠１の値が生じる場合には、実際にはｇ＝ｅ（Ｕ＋ｙＡ，Ｑ）$^{\delta}$

となる。δを見つけるには、ペアリングの計算及び離散対数問題の解が必要である。

【０２５６】

これを達成するための１つの適切なアルゴリズムがＰｏｌｌａｒｄのＫａｎｇａｒｏｏｓの方法［Ｂ１５］である。これは「平方根」アルゴリズムであり、つまり４桁のＰＩＮの場合、δを見つけ出すためにG_Tで数百の乗算だけが必要となり、このことは完全に実際的である。

【０２５７】

ＰＩＮエラーの範囲がサーバによって決定でき（これが潜在的にいずれの二因子認証方式にも当てはまるが、特にここで提案されているプロトコルに当てはま）る場合、旧く、単純化した「三振即アウト」機構よりむしろ、よりインテリジェントな応答が可能である。サーバは、トークンを取り込んで、考えられるすべての組合せを試すことによってＰＩＮを見つけ出そうとする不良エンティティと、ＰＩＮを間違って入力した、または間違ったＰＩＮを不注意に入力したかのどちらかの良いエンティティをインテリジェントに区別しようと試みる可能性がある。ＰＩＮエラーの知識を利用するための１つの簡略な方法は、同じ間違ったＰＩＮを複数回入力するユーザーを再度罰しないことである。同じＰＩＮを２回予想値とすることは不良エンティティにとって価値がなく、したがってこれを行うことによって何も失われないことに留意されたい。

【０２５８】

間違って入力されたＰＩＮの場合、エラーは通常１桁においてだけとなる。再び、これはＰＩＮエラーからサーバによって検出できる。したがって、より入念なタイプのスコア

リング機構の例を提案する。例えば、ユーザーは１０を超えるエラースコアに達するとユーザーがロックアウトされるだけの場合がある。完全に間違ったＰＩＮは４点となり、同じ間違ったＰＩＮが再び入力されると０点となり、１桁だけ外れたＰＩＮは２点となり、２桁だけ外れたＰＩＮは３点となる。このようにして、不良エンティティは通常３回だけ試した後にそこに到達するのに対し、本物のよいエンティティは１０点得点するために奮闘する。言うまでもなく、１０の最大エラースコアは例にすぎず、別の最大エラースコアが選択されてよい。

【０２５９】

特定のアイデンティティをロックアウトするかどうかに関するインテリジェントな決定に混ぜることができる他の側の情報をサーバが利用できることに留意されたい。また、サーバは、クライアントのＩＰアドレス、試行されたログイン時刻も知るだろう。アイデンティティ及びＰＩＮエラーと組み合されると、これは失敗した接続ごとに誰が、何が、どこで、及びいつに相当する。

【０２６０】

ここで図８に関して、第２の因子でエラーを処理する１つの例示的なプロセスが説明される。図８に関して説明されるプロセスでは、ＰＩＮは４桁の数であってよい。しかしながら、エラー処理プロセスが任意の形をとるＰＩＮに適用可能であってよいことが理解される。エラー処理プロセスはＰＩＮに関して説明されるが、エラー処理プロセスは、クライアントシークレットの因子として使用される任意のデータ列でのエラーまたは差異を見つけ出すために使用できる。エラー処理プロセスは、数字のストリングまたは番号に約分できる任意のタイプの情報に相当する、シークレットの因子とともに使用できる。ステップ８．１で、サーバ３はクライアント２で入力されたＰＩＮが間違っていたと判断する。これが、ＰＩＮがカレントセッション中に入力された初めてである場合、サーバはエラーカウンタｎの値を０に設定する。ステップ８．２で、サーバは、上述されたようにＰＩＮエラーの範囲を決定する。サーバはプロトコルの最後に得られたｇの値から、及びｇ＝ｅ（Ｕ＋ｙＡ，Ｑ）$^{\delta}$を使用することによってδを見つけ出す。サーバは次いで、ステップ８．３でＰＩＮが１桁外れているかどうかをチェックし、外れている場合、ステップ８．４でエラーカウンタｎにエラーカウント「ａ」を加算する。サーバは次いでステップ８．１０に進む。エラーカウント「ａ」の値は、例えば２であってよい。ただし、他の実装では、エラーカウント「ａ」は２より小さいこともあれば、２より大きいこともある。ステップ８．３でＰＩＮが１桁分外れていないと判明すると、プロセスは代わりにステップ８．５に続行し、ピンが２桁分外れていたかどうかをチェックする。ＰＩＮが２桁分外れていた場合、ステップ８．６で適切なエラーカウント「ｂ」がエラーカウンタｎに加算される。サーバは次いでステップ８．１０に進む。２分外れているＰＩＮのエラーカウント「ｂ」は、例えば３であってよい。ただし、他の実装では、エラーカウント「ｂ」は３より小さいこともあれば、大きいこともある。ＰＩＮが１桁分も、２桁分も外れていない場合、プロセスは、ステップ８．７でＰＩＮが以前に入力されたことがあるかどうかをチェックする。上述されたように、不良エンティティにとって複数回同じ間違ったＰＩＮを入力することは価値がないので、ＰＩＮが以前に入力されたことがある場合、サーバ３はエラーカウントをそれ以来増やさないことがある。ステップ８．８は、カレントセッションで前に間違ったＰＩＮが入力されたことがあると判断されると、エラーカウントが値「ｃ」分増加することを示す。説明されるように、「ｃ」の値は０に等しくてよい。ただし、他の実装では、適切な場合、「ｃ」の値は０より大きくてよい。ＰＩＮが１桁または２桁のどちらか分外れておらず、ＰＩＮが以前に入力されたことがない場合、サーバ３は、該ＰＩＮは初めて入力された完全に間違ったＰＩＮであると判断し、ステップ８．９でエラーカウンタに適切なエラーカウント「ｄ」を加算する。エラーカウント「ｄ」の値は４であってよいが、該値は他の実装ではより小さいこともあれば、より大きいこともあるだろう。

【０２６１】

サーバは次いで、ステップ８．１０でエラーカウンタの値が最大エラーカウントN_{ma}

$_x$より小さいかどうかをチェックするために進む。上述されるように、この最大エラーカウントは１０であることがあるが、この最大エラーカウントはより小さいこともあれば、より大きいこともある。最大エラーカウントは、例えば２または３等の非常に小さい値から、１０よりはるかに大きい値であってよい。言うまでもなく、最大エラーカウントは、「ａ」、「ｂ」、「ｃ」及び「ｄ」の値にも依存し、「ａ」、「ｂ」、「ｃ」及び「ｄ」の値を考慮に入れることによって選択された任意の適切な値であるだろう。エラーカウンタの値が最大エラーカウント値よりも小さい場合、サーバはステップ８．１２で再び認証を試みるようにクライアントに要請してよい。クライアントは次いでユーザー６に新しいＰＩＮを試すように依頼する。ただし、サーバがクライアントに新しＰＩＮを試すように要請する前に、サーバは、ステップ８．１１で、クライアントの場所、時刻、クライアントのアイデンティティ、またはクライアントのＩＰアドレス等であるが、これに限定されるものではない、検討される必要がある任意の他の情報があるかどうかをチェックしてよい。サーバはこの追加情報に基づいて進める方法を決定するための多様な規則を記憶してよい。サーバが依然としてクライアント２に新しいＰＩＮを試すように要請すると決定する場合、サーバはそうするようにクライアントにメッセージを送信する。クライアントがユーザー６から新しい間違ったＰＩＮを受信すると、プロセスはステップ８．１から再開する。ＰＩＮが再び間違っている場合、エラーカウンタｎの値は、ステップ８．３から８．９に関して説明されるように増加する。

【０２６２】

ステップ８．１０でエラーカウンタｎの値が最大エラーカウント以上である場合、サーバは再び、接続を拒否するデフォルトの決定を変更する可能性がある追加の情報があるか考えてよい。上述のように、追加情報は、クライアントの場所、時刻、クライアントアイデンティティ、及びＩＰアドレスを含んでよい。サーバはステップ８．１３でこの情報を検討し、クライアントに新しいＰＩＮを試すように要請する決定が下されると、プロセスはステップ８．１２に進む。代わりに、接続を拒否する決定が下されると、サーバはステップ８．１４で接続を拒否する。また、サーバは、ステップ８．１１で検討された追加情報が、たとえエラーカウントが最大エラーカウントよりも小さくても、接続は拒否されるべきであることを示す場合、ステップ８．１４で接続を拒否するように進む。また、サーバは、アカウントがブロックされていること、及びそのアカウントをリセットする、または新しいアカウントを開く前にＴＡ４に接触する必要があることをクライアント２にも知らせてよい。これは、クライアントが新しいクライアントシークレット７を発行されること必要とすることがあるだろう。

【０２６３】

図８に関して説明されたプロセスがエラー訂正のための１つのプロセスにすぎず、多様な修正形態及び変形形態が考えられることが理解されるだろう。例えば、追加情報の検討は、エラーカウンタの値が調整され、チェックされる前に、または後に行われてよい。さらに、代替実施形態では、サーバは追加情報を検討しないことがある。さらに、図８では１桁または２桁の特定のＰＩＮエラーが検討されているが、エラーカウンタは他のＰＩＮエラーについて調整されてもよい。例えば、プロセスは、ＰＩＮが３桁もしくは４桁外れている、または別の特定のＰＩＮエラーが決定された場合、特定の値でエラーカウンタを増加することを含んでよい。また、エラーカウント値は、エラーの正しくない及びその桁のエラーの範囲がＰＩＮのどの桁にあるのかに応じて変化してもよい。さらに、サーバはエラーカウンタパラメータを使用しないことがある。代わりに、サーバは、各認証試行のエラー値を記憶してよい。サーバは、サーバが再び認証するようにクライアントに要請するかどうかを決定する必要があるたびにエラー値を加算してよい、またはサーバは別個のエラー値に基づいて決定を下してよい。

【０２６４】

エラー処理プロセスは、表６の特定のプロトコルに関して説明されてきたが、エラー処理プロセスは、クライアントシークレットの因子のエラーを決定できるようにするだろう任意のプロトコルと使用できる。エラー処理プロセスは、因子の内の１つでのエラーの範

囲が認証側エンティティによって決定できる任意の多因子プロトコルに適用してよい。
【０２６５】
さらに、エラー処理は４桁のＰＩＮに関して説明されてきたが、エラー処理は４桁よりも短い長さのＰＩＮまたは５桁もしくは６桁もしくはなおさらに長い長さのＰＩＮにも適用されてよい。エラー処理プロセスは、多因子シークレットの因子の内の１つを形成する任意の桁数のＰＩＮに適用されてよい。ただし、ＰＩＮが長くなるほどエラーの計算に要する時間も長くなることが理解される。
【０２６６】
ここで、提案されている方式のセキュリティを考える。使用している根本的なＩＢＩ方式は、方式を壊すことがもう１つの計算ディフィーヘルマン問題と同程度に困難であると示されている点で、［Ｂ５］によって安全であると証明された。彼らは、（$G_1 = G_2$である）タイプ‐１ペアリングとの関連でプロトコルを検討した。これは、G_1とG_2の間に計算可能な同型マッピングが存在するタイプ‐３ペアリングと見なすこともできる。セキュリティ証拠をタイプ‐１からタイプ‐３ペアリングに移す問題は、Ｓｍａｒｔ及びＶｅｒｃａｕｔｅｒｅｎ［Ｂ２０］によって検討された。彼らの最も簡略な解決策は、アタッカーが利用できるようにされ、同型マッピングを実装したＯｒａｃｌｅとの関連でセキュリティ証拠を「相対化する」ことであった。係る同型マッピングが知られていないという事実は明らかに証拠を弱めない。
【０２６７】
次の懸念は、トークンを取り込むアタッカーは、サーバの自発的な参加なしに、関連付けられたＰＩＮ番号を決定することができてはならないという点である。特に強力なアタッカーは、自らサーバのクライアントである、またはその完全なクライアントシークレットを進んで提供する内部関係者の連合体を採用できた者だろう。
【０２６８】
したがって、ｓＡ、ｓＢ、ｓＣ．．．及び被害者者トークン（ｓ－α）Ｚを所有していたアタッカーを考える。被害者トークンに絶えずＺを加算し、それがやはりｓＺに等しくなったときのケースを区別することが可能だろうか。これはまさに、Ｂａｏ、Ｄｅｎｇ、及びＺｈｕ［Ｂ３］によって考えられるような一般化された決定ディフィーヘルマン問題である。彼らは、これが群G_1での標準的なＤＤＨ（決定ディフィーヘルマン）仮定まで縮小したことを証明した。
【０２６９】
ＤＤＨ問題がタイプ‐３ペアリングでG_1では困難であることは、最初は非公式に［Ｂ１６］で暗示され、［Ｂ２］で明示されたＸＤＨ仮定として知られている。
【０２７０】
最後に、被害者トークンにアクセスし、プロトコルの実行をなんとか盗聴する、またはクライアントに、それとともに直接的に実行されるプロトコルに関与させるために「フィッシング」攻撃を介してそらく成功するアタッカーを考える。係るアタッカーは、値ｚＡ、（ｓ－α）Ａ、及びｚｓＡ、つまり実行されるプロトコルから得られる最初の値及び最後の値を収集できる。ただし、ＰＩＮを見つけ出すことは、アタッカーが、この関係性なしに３つの値からｚＡ、ｓＡ、及びｚｓＡを区別できることが必要とし、このことは再びＸＤＨ仮定によって扱われる。
【０２７１】
完全なクライアントシークレットが、それがＩＢＩプロトコルで使用される前にトークン及びＰＩＮから再構築されることに留意されたい。また、サーバはクライアントにＰＩＮを決定する際に役に立つ可能性がある何かを送信しない。つまり事実上、サーバはランダムチャレンジｙしか送信しない。
【０２７２】
ここで、ハッカーによるサーバへの侵入の成功を考える。この意味するところを考える前に、サーバシークレットが（それがサポートするクライアント数とは関わりなく）単一点$sQ \in G_2$にすぎないことを指摘する価値がある。したがって、ハードウェアセキュリ

ティモジュール（ＨＳＭ）の内部でこのシークレットを保護することは容易だろう。ただし、このシークレットが取り込まれると仮定すると、これによりハッカーはクライアントがひきつけられる偽のサーバをセットアップできるだろう。これにより、取り込まれたトークンに関係するＰＩＮをハッカーが見つけ出せるようにもなる。実際のところ、ハッカーは、サーバシークレットを知っているため、偽のサーバを作り出し、それに対してＰＩＮ予想値を試すことができるので、これは　任意の二因子認証方式についても当てはまらなければならない。しかしながら、成功したハッカーは完全なクライアントシークレットを再構築することはできないため、それ自体、本物のサーバに、及びクライアントアカウントの中にログオンすることはできない。したがって、損害の可能性は大幅に削減される。

【０２７３】

いくつかの実装例では、図５に関してすでに説明されているように、セキュリティはさらに分散サーバ配置を使用することによって高められてよい。図５のシステムでは、サーバシークレットは、完全なサーバシークレットｓＱを作成するために追加される２つの部分s_1Ｑ及びs_2ＱでＴＡによって発行される、ｓ＝s_1＋s_2である。プロトコルは、２つのシークレットs_1Ｑ及びs_2Ｑを別々に保ちながら完了される。サーバ側では、サーバシークレットを含んだ計算の部分は、なんらかのランダムＸのためのペアリングｅ（Ｘ，ｓＱ）である。しかし、双線型性の魔法によって、ｅ（Ｘ，ｓＱ）＝ｅ（Ｘ，s_1Ｑ）．ｅ（Ｘ，s_2Ｑ）である。

【０２７４】

したがって、例えば、サーバは（それぞれ異なる製造メーカから）２つのＨＳＭを有するだろう。一方はs_1Ｑを記憶し、他方はs_2Ｑを記憶する。一方はｅ（Ｘ，s_1Ｑ）を計算し、他方はｅ（Ｘ，s_2Ｑ）を記憶する。サーバプロセスは、次いで単にこれらの値を乗算し、通常通りプロトコルを続行するだろう。ただし、例えばｅ（Ｘ，s_1Ｑ）を知ること、及びＸを知ることはs_1Ｑについて何も明らかにしない。それは、困難であると考えられている逆ペアリング問題である。したがって、単一のエンティティはサーバシークレットｓＱを知り、考えられる単一障害点が排除される。

【０２７５】

図５に示されるように、これは、２つの異なるサーバ３ａ、３ｂ、及び該２台のサーバからペアリングの結果を受信し、結果を乗算して通常通りプロトコルを続行するプロキシ３ｃとして実装できるだろう。代わりに、サーバの内の１つがメインサーバとなり、それが他のサーバでのペアリングの結果をその別のサーバから受信してもよい。さらに別の代替策として、上述されたように、２台のサーバは同じサーバの２つの安全なモジュールであってよい。

【０２７６】

図５及び図７に関して、サーバシークレットが２つのサーバシークレットとして記憶される場合、ステップ７．５は入力としてそのそれぞれのサーバシークレット及び点Ｕ＋ｙＡ，を採るペアリングを実行し、次いで２つのペアリングの積を実行する各ＨＳＭ３ａ、３ｂを含むように修正される。言い換えると、サーバ３ａの内の一方がｅ（Ｕ＋ｙＡ，s_1Ｑ）を計算し他方のサーバ３ｂがｅ（Ｕ＋ｙＡ，s_2Ｑ）を計算する。２つのペアリングは、プロキシが、ＨＳＭの積を実行するためにプロキシ３ｃに送信されてよい、または完全なサーバシークレットの一部を記憶するＨＳＭの一方は２つのペアリングの乗算を実行してよい。

【０２７７】

図２に関して言及されたように、配備が広がったために、このプロトコルのクライアント側がブラウザの内部で、理想的には低出力のモバイル機器上で実装されるブラウザの内部でもＪａｖａＳｃｒｉｐｔ（登録商標）に実装されることが必要とされる可能性があることが認識される。表面的には、これは非常に困難な見通しである。ただし、ＪａｖａＳｃｒｉｐｔ（登録商標）エンジンは、過去数年にわたってモバイル機器の処理能力が改善してきたように、根本的に改善してきた。

【０２７８】
Ｍ－Ｐｉｎプロトコルのクライアント側を調べると、それが、G_1で２点乗算を必要とすることがわかる。G_1でのこれらの点乗算は、Ｇａｌｌａｎｔ、Ｌａｍｂｅｒｔ、及びＶａｎｓｔｏｎｅ［Ｃ１］によって説明されるようにＢＮ曲線上に存在する効率的な自己準同形の恩恵を受けるだろう。これによって、点乗算はほぼ２倍速くなる。代わりに、他のタイプの曲線及び乗算を計算する方法を代わりに使用できる。

【０２７９】
ラップトップ及びデスクトップの場合、クライアント側のタイミングは感知できない。モバイル機器のためのいくつかのタイミングは表７に示される。サーバ側では、現代のＩｎｔｅｌプロセッサでは、処理時間は接続あたりほぼ数ミリ秒程度である。

製造メーカ	モデル	OS	ブラウザ	Ｍ－Ｐｉｎタイミング（秒単位）
Apple	iPhone 4	iOS 6	Safari	2.5
Apple	iPhone 5	iOS 6	Safari	1
Apple	iPad 2	iOS 6	Safari	1
Samsung	Galaxy S3	Android	Chrome	1
Motorola	MB256	Android	Firefox	3

表７．Ｍ－Ｐｉｎクライアント側タイミング

【０２８０】
図６及び図７に関して説明された方式は、サーバ側でＳＳＬの保護の下実行できるだろう。ただし、他の例では、ＳＳＬは使用されないことがある。Ｍ－Ｐｉｎはそれ自体ペアリングベース及びアイデンティティベースのプロトコルであるので、ＰＫＩベースのＳＳＬに代わるペアリングベースアイデンティティベースの暗号化（ＩＢＥ）が検討されてよい。例えばＳａｋａｉ及びＫａｓａｈａｒａ［Ｃ２］のＩＢＥプロトコルが使用されてよい。このプロトコルはクライアント側でのペアリング計算を必要とせず、サーバ側の単一のペアリングを必要とする。したがって、Ｍ－Ｐｉｎの実装は、ＢＮ曲線を使用してＩＢＥを実装するための重要な構成要素のすべてをすでに含んでいる。いくつかのアプリケーションでは、認証だけが必要とされ、クライアント及びサーバは暗号鍵を確立する必要はない。したがって、いくつかの実装では、暗号鍵は確立されない。

【０２８１】
さらに、表６のプロトコルでは、クライアントはサーバを認証していない。サーバの認証が必要とされる場合、これはＳＳＬを使用してまたは別の適切な方式を使用して実施できる。

【０２８２】
Ｍ－Ｐｉｎ　Ｆｕｌｌ
いくつかの例では、ユーザー名／パスワードだけではなくＳＳＬの機能性も置き換えるプロトコルも提供される。ここでは、これを達成するいくつかの方法も説明する。

【０２８３】
表６のプロトコルは、クライアント及びサーバで鍵を確立するためにも拡張できる。将来のメッセージを暗号化し、解読するためのセッション鍵も確立する拡張暗号プロトコルの例が表８に示される。

【０２８４】
文献に説明されるペアリングベースの鍵交換プロトコルでの１つの問題は、プロトコル

が計算負荷の観点で「平衡状態」にあり、クライアントもペアリングを計算するように要求される点である。これは、近年行われてきた改善［Ｂ１］のすべてをもってしても概してかなりの計算として認識される。上記に検討された既存の方式のすべては、両方の当事者がペアリングを計算することを必要とする。これは、低出力のクライアントが高出力のサーバとの鍵交換を試行している場合がある事情では、特に不適切である。

【０２８５】

ペアリングの計算を、我々のため重労働を実行できる計算上より恵まれたペアリングサーバにアンロードすることである。本明細書に説明される例のいくつかでは、方法はプロトコルの「平衡を失わせる」ように提供され、これによってクライアントはサーバによるペアリング計算の重荷から解放される。ペアリング計算がＣ＝ｅ（Ｐ，Ｑ）である場合、考え方は、Ｐ及びＱはペアリングサーバに渡され、ペアリングサーバはペアリングＣ＝ｅ（Ｐ，Ｑ）を計算し、Ｃの値を戻すということである。

【０２８６】

ただし、これは明らかなセキュリティの意味合いをもっている。パラメータＰまたはＱの内の少なくとも１つは慎重に扱うべきシークレットとなり、この値を必ずしも信頼されていない当事者に渡すことは望まれていない。Ｐ及びＱの値は公然と通信されるべきではない。Ｐがシークレットパラメータであると仮定して、ペアリング値が我々に戻されるときにペアリング値を手軽に「暴露する」ことができるように、Ｐを「隠す」ためにペアリングの特性を利用できる。したがってランダムマスクｍを生成し、ｍＰ及びＱをペアリングサーバに渡し、サーバはＤ＝ｅ（ｍＰ，Ｑ）を返す。次いでＣ＝Ｄ$^{1/m}$を取り戻す。これで、すべてｅ（Ｐ，Ｑ）＝ｅ（ｍＰ，Ｑ）$^{1/m}$として双線型性のためにうまくいく。

【０２８７】

ペアリング計算を委託する考え方は、Ｃｈｅｖａｌｉｅｒ－Ｍａｍｅｓら［Ｂ９］によって検討される。彼らは、悪意のあるサーバが間違った結果で応答し、このことが結果的にプロトコルにとって不幸な結果を引き起こす可能性があるという可能性を考えて、念入りな解決策を考える。しかし、我々のケースでは、最終的な鍵は、ペアリング値から直接的に導出され、したがって偽のペアリング値は例えば通信故障と識別できないように、単にプロトコルを失敗させる。したがって、上記解決策は我々の目的には十分である。

【０２８８】

ただし、いくつかの懸念がある。一見したところ、クライアントが、Ｄの受信された値が正しい位数であることをチェックすることが重要である。これは小さいサブグループ制限攻撃［Ｂ１４］を妨げるためである。ペアリング値が

$$F_{p^{12}}$$

の要素であり、ＢＮ曲線のためのサイズで２５６ビットである素数位数ｑを有する必要があることを思い出す。ただし、

$$F_{p^{12}}$$

の乱数値はｐ^{12}－１の任意の正確な約数である位数を有することがあり、ｐは素数係数である。位数３の値が存在すると想像する。次いで、クライアントは、クライアントのマスキング値ｍとは関係なく、ｋに３つの考えられる値の内の１つだけを計算できるため、Ｍ－Ｐｉｎサーバを装うエンティティと共謀する悪意のあるペアリングサーバは係る値ｔを渡すことによって表８に示されるプロトコルを完了できる。偽の「サーバ」は、彼らが正しいものを見つけ出すまで３つすべてを単に試すことができる。

【０２８９】

この攻撃に対する防御の第１のラインは、ペアリング値が、位数$\Phi_{12}(p)=p^4-p^2+1$の円分の下位群の要素でなければならないように、位数$\Phi_{12}(p)=p^4-p^2+1$の円分の下位群の要素であることの迅速且つ手軽なチェックを実行することである。これは、フロベニウス作用を利用し、

$$D.D^{p^4} = D^{p^2}$$

であることをチェックすることによってほぼ無料で行うことができる。Ｄ≠１であることのチェックも実行される必要がある。ただし、ペアリング円分の最終的な累乗法の「ハード部」と呼ばれることがある、下位群内部の余因子、（ｐ^4－ｐ^2＋１）がそれ自体小さい約数を有する可能性があることも依然として考えられる。

【０２９０】

この余因子が素数である（ｑよりもはるかに大きい）ＢＮ曲線を使用することによって攻撃経路をブロックできる。これは、例えば、（ｐ^4－ｐ^2＋１）／ｑが素数となり、明確にｑよりもはるかに大きくなるように、ＢＮ曲線パラメータを選ぶことによって達成できる。係るＢＮ曲線を「ＧＴ－Ｓｔｒｏｎｇ」と呼ぶ。

【０２９１】

ＢＮ曲線は有り余るほどなので、係る曲線を見つけることは難しくなかった。「ＧＴ－Ｓｔｒｏｎｇ」曲線を見つけ出すことは、ｘの極端に低いハミング重みを犠牲にして行うのは難しくない。例えば、曲線パラメータｘ＝－４０００８０６０００００４０８１１６$_{16}$（［Ｂ４］参照）の結果、効率的な実装のために望ましいように「Ｂ１」、ｘに相対的に低いハミング重みを維持しつつ適切な曲線が生じる。

【０２９２】

ここで、小さい下位群は存在していないので、群メンバーシップテストを省略し、ある程度節約することができる。

【０２９３】

Ｍ－Ｐｉｎプロトコルとともに使用される「ＧＴ－Ｓｔｒｏｎｇ」曲線の使用が上述されてきたが、「ＧＴ－Ｓｔｒｏｎｇ」曲線は「ＧＴ－Ｓｔｒｏｎｇ」曲線の特徴が適している任意のペアリングベースの暗号化プロトコルで使用されてよい。さらに、Ｍ－Ｐｉｎ及びＭ－Ｐｉｎ　Ｆｕｌｌプロトコル、及び本明細書に説明される他のプロトコルは、「ＧＴ－Ｓｔｒｏｎｇ」曲線を使用しなくても実現できることが理解される。

【０２９４】

次の質問は、Ｄ$^{1/m}$をどのようにして最もよく計算するのかである。これはフルペアリングを計算するよりもはるかに安価であるが、依然としてなんらかのコストは有している。いくつかの試験は、最も高速の解決策がペアリング値を

$$\mathbb{F}_{p^4}$$

の要素に圧縮し［Ｂ１９］、Ｓｔａｍ及びＬｅｎｓｔｒａ［Ｂ２１］によって説明される高速ＸＴＲ方法を使用することである。次の考え方は、別個のペアリングサーバを配備する代わりに、なぜＭ－Ｐｉｎサーバ自体にこの役割を果たさせないのかという点である。結局、Ｍ－Ｐｉｎはオンラインプロトコルであり、Ｍ－Ｐｉｎサーバはそれを実行するために利用可能でなければならない。また、Ｍ－Ｐｉｎサーバがすぐれた計算能力を有することは妥当な仮定である。事実上、これらの役割を組み合わせることによって、分かるようにいくつかの追加の簡略化を達成する。

【０２９５】

いったんクライアントアイデンティティが表６のプロトコルを使用して認証されると、プロトコルは鍵交換を完了するために続行する。プロトコルの２つの部分は最大効率のために注意深く結合されてよい。このプロトコルは、その入力をシリアルに変換し、２５６ビット値Ｋにハッシュする、別の一般的なハッシュ関数Ｈ$_g$（．）も必要とする。両方の側とも次いで高度暗号化標準（ＡＥＳ）鍵をこの値Ｋから抽出できる。完全なプロトコルは表８に提供される。

アリス - アイデンティティID_α	サーバ
ランダム$x,w < q$を生成する $A = H(ID_\alpha)$	ランダム$n,y < q$を生成する
$U = xA$ $ID_\alpha, U \rightarrow$	
	$\leftarrow y$
$m = -(x + y)$ $V = m((s-\alpha)A + \alpha A) \rightarrow$	$A = H(ID_\alpha))$ $t = e(V,Q)$
	$g = t.e(U + yA, sQ)$ $g \neq 1$の場合、接続を拒否する
$W = wA \rightarrow$	$\leftarrow r = t^n$
$k = r^{w/m}$ $K = H_g(k)$	$k = e(nW, sQ)$ $K = H_g(k)$

表８．Ｍ－Ｐｉｎ Ｆｕｌｌ

【０２９６】
プロトコルの平衡を失わせることは、プロトコルに「完全前方秘匿性」の望ましい特性を与えるのに十分である。サーバは三を計算しなくてはならないが、クライアントはペアリングを計算する必要はない。また、クライアントは群G_2でなんらかの算術を行う要件も有していない。実際に、すべての点乗算はより簡略な群G_1にある。クライアントとサーバの双方が同じ鍵で終了することを確認するために双線型性を利用することは読者にとって簡単な練習として残される。プロトコルの第１の部分は基本的には元のＭ－Ｐｉｎプロトコルにすぎないので、その特徴のすべては依然として適用することに留意されたい。

【０２９７】
プロトコル出力は、ＰＩＮエラーを決定し、表６のＭ－Ｐｉｎプロトコルに関して上述されたようにＰＩＮエラーの範囲に基づいてクライアントに応答するために利用できる。例えば、図８のエラー処理プロセスは、プロトコルの鍵共有部分が実行される前にこのプロセスと使用できるだろう。

【０２９８】
表８のＭ－Ｐｉｎ Ｆｕｌｌプロトコルは、ここで図９及び図１０に関してより詳細に説明される。

【０２９９】
クライアント側でのステップは図９に関して最初に説明される。プロトコルの始まりは、事実上、ステップ６．１からステップ６．６に関してすでに説明されている。ただし、クライアントがステップ６．６で別個のパラメータｍ＝－（ｘ＋ｙ）を記憶してよいことが留意される。図６のステップ６．７は、図９のステップ９．１から９．４で置き換えられてよく、これらのステップがここで説明される。クライアント２は、ステップ９．１でサーバ３から、Ｖ及びＱに基づくペアリングの結果から導出された値ｒを受信する。この値を受信することによって、クライアントは、入力されたＰＩＮが正しかったこと、及びＰＩＮが認証されていることを知る。代わりに、クライアントはクライアントが認証されていることを示す追加の情報を受信してもよい。サーバは、クライアントが鍵を導出するために必要とされるペアリングを計算し、ｒはそのペアリングの結果ｔから導出される。結果的に、クライアント２はペアリングを計算する必要はない。ただし、ペアリングに対

する入力の１つは、ｍでマスクされたクライアントシークレットの１つであり、クライアントはここで、クライアントがｒからｋを導出するときに、サーバにとって未知であるそのマスキング値ｍを考慮する必要がある。ペアリングｔの結果はｒを得るためにサーバでｎ乗され、クライアントはｎの値を知らない。しかし、クライアントはペアリングの結果を取り戻す必要はなく、クライアントはサーバと同じｋを確立しさえすればよい。したがって、サーバが実際のマスキング値ｍを共有することなく同じ鍵を計算できるようにするために、クライアントはサーバと情報を共有する必要がある。したがって、ステップ９．２で、クライアントは、ｑよりも小さい乱数値ｗを生成し、楕円曲線上で点Ａの乗算を実施して新しい点Ｗの座標を見つけ出す。クライアントは、次いでこの点の座標をサーバに送信する。ステップ９．３で、クライアントは次いで受信した値ｒをｗ／ｍ乗する。ステップ９．４で、クライアントは結果ｋを鍵Ｋにハッシュする。クライアントは、それによって、追加の通信を暗号化するために使用できるセッション鍵を得た。

【０３００】

図９のステップは、図２に関して説明される認証プログラム２５の命令を実行するクライアント２のプロセッサ２１によって実施されてよい。

【０３０１】

ここで、サーバ側でのステップが図１０に関して説明される。再び、第１のステップは図７のステップ７．１からステップ７．６に関してすでに説明されている。ただし、サーバが、サーバが後に値ｒを導出できるようにステップ７．５で第１のペアリングの結果ｔを記憶してよいことが留意される。サーバは、サーバが第２のペアリングの結果でそれを乗算する前にｔを得るために第１のペアリングを実行してよい。ｇの値が１に等しい場合、または例えば図８に説明されるようにエラー処理プロセスを使用して正しいＰＩＮが後に入力される場合、サーバ３は次いで、ここでステップ１０．１からステップ１０．５に関して説明される鍵導出プロセスを進める。ステップ１０．１で、サーバはｑより小さいランダムｎを生成する。ステップ７．５で、サーバはペアリング計算を実行した。ペアリングの内の１つは、入力として、クライアントから得られたＶの座標、及び固定点Ｑの座標を採った。サーバはここでそのペアリングの結果をｎ乗して、値ｒを得る。サーバは次いでステップ１０．２で値をクライアントに送信する。したがって、サーバはクライアントのための鍵を導出するために必要とされたペアリングを計算し、すべての大量の処理はサーバにある。クライアント２は、次いで、図９に関して上述されたように、ｒの値を使用して鍵を導出できる。

【０３０２】

サーバ３は次いでステップ１０．３でクライアントから点Ｗの座標を受信する。Ｗは、クライアント側で鍵にｒを変形するために使用される値ｗを使用して得られた。したがて、サーバは、双線型性を使用して同じ鍵を導出するためにＷの座標を必要とする。ステップ１０．４で、サーバは、楕円曲線上で点Ｗをｎで乗算する。サーバは、次いで、入力としてＷ及びｎの乗算から生じる点の座標及びサーバシークレットｓＱの座標を採る、別のペアリングを計算する。サーバは、次いで鍵Ｋを得るためにペアリングの結果ｋをハッシュする。これは、ペアリングの双線型性特性を使用することによって容易に認識されるようにクライアント側で得られる鍵と同じ鍵である。

【０３０３】

図１０のステップは、図３に関して説明されるサーバ認証プログラム３４の命令を実行するサーバ３のプロセッサ３１によって実施されてよい。

【０３０４】

表８に提示され、図９及び図１０に関して説明されるプロトコルが、ＳＳＬと組み合わされた表６に提示されるプロトコルの組合せに完全に同等ではないことが指摘されるべきである。ＳＳＬを用いると、Ｍ－Ｐｉｎプロトコル全体がＳＳＬに隠れて実行し、したがって匿名性特徴を提供するのに対し、クライアントアイデンティティはＭ－Ｐｉｎ　Ｆｕｌｌで平文で送信される。他方、サポートされている前方シークレットモードがあるが、ＳＳＬは通常この機能なしで操作されるのに対し、Ｍ－Ｐｉｎ　Ｆｕｌｌは完全前方秘匿

性を有する。言うまでもなく、Ｍ－Ｐｉｎ　Ｆｕｌｌプロトコルをｓｓｌと併せて実行することはつねに可能である。

【０３０５】

有効なシークレットを所有していない偽のサーバが、$r^{w/m}$がそれも計算可能な値となるように特別に仕組まれたｒの値を返す場合どうなるのか。ｗ及びｍの値はアタッカーにとって未知であるので、ランダムｒを送るだけでは機能しないのは明らかである。しかしながら、もっともらしい攻撃は、$e(C,D)^{w/m}$が既知となるように値ｓ＝ｅ（Ｃ，Ｄ）を構築することである可能性がある。ただし、未知のｍの影響を取り消すには、ペアリングはパラメータとしてｍの倍数を有さなければならず、Ｖはそれが利用できる唯一の係る倍数である。また、ペアリングは、ペアリングがｗ乗されるときに、次いでＷ＝ｗＡはそれにとって既知のｗの唯一の倍数であるので、パラメータとしてＡも有さなければならない。しかし、Ｖ及びＡの両方ともG_1からの要素であり、G_1からの唯一のパラメータはペアリングに提出できる。したがって、係る値は構築できない。

【０３０６】

これの１つの結果は、フィッシングウェブサイトが相互鍵Ｋを確立できないので、いわゆる「フィッシング」攻撃がこのプロトコルに対して有効ではない点である。

【０３０７】

時間許可証

プロトコルは時間許可証を含んでもよい。時間許可証は簡略な交互取消し機能を提供する。考え方は、いくつかの例では、サーバがアリスのハッシュ済みのアイデンティティのその構築で期間を含むということである。アリスが同じ期間に対する対応する「時間許可証」を保持しない限り、アリスはプロトコルを完了できない。時間許可証は、例えば１日または２日間有効であってよい。ただし、期間が、例えば１週間または１月、または１週間もしくは１月よりも長く等、はるかに長い間、有効であってよいことが理解される。

【０３０８】

表６及び表８のプロトコルでは、代わりにプロトコルの両側で$A=H(ID_\alpha)+H_T(ID_\alpha|Ti)$を計算でき、上式で$T_i$はｉ番目の期間のテキスト記述であり、$H_T(.)$は$H(.)$と異なるハッシュ関数であり、｜は連結を示す。$T_i$は、１日または１月等の明示的に記述されるタイムスロットであってよい。結果的に、点Ａ及びクライアントシークレットｓＡは、ここでシークレットが有効である期間からも導出される。プロトコルが正しく機能するために、アリスは、信頼機関４によって、ｓＡを作成するためにアリスの結合されたＰＩＮプラストークンシークレットｓ．$H(ID_\alpha)$に追加される許可ｓ．$H_T(ID_\alpha|T_i)$を発行されなければならない。許可が他の当事者には役に立たず、したがって公に発行できる、または単にｅメールでアリスに送信できる、またはサーバを介して配信できることを観察する。図１に関して言及されたように、クライアントはストレージ２３に時間許可証２７を記憶してよい。Ｂｏｎｅｈ及びＦｒａｎｋｌｉｎのＩＢＥとの関連でのこの考えのためのセキュリティの証拠は、［Ｂ２２］に記載されている。Ｍ－Ｐｉｎプロトコルは上述されたＰＩＮエラーを決定するための機構をサポートし続け、例えば図８に関して説明されたＰＩＮエラー処理プロセスを実施するために小さな修正を必要とする。このエラーは、ＰＩＮプラストークン構成要素に反映され、時間許可証には反映されないので、クライアントはプロトコルの第１のパスでサーバ$R=x.H(ID_\alpha)$も送信しなければならない。

【０３０９】

表６のプロトコルに基づき、時間許可証を含んだプロトコルは、表９に示される。表９のプロトコルでは、表記は上記の時間許可証の説明で使用される表記に対して異なっている。表９のプロトコルでは、アイデンティティ及びクライアントシークレットは表６及び表８と同じままであるが、プロトコルの両側で新しい点$D=H(ID_\alpha)+H_T(ID_\alpha|T_i)$を導出し、上式で$T_i$はまだｉ番目の期間のテキスト記述であり、$H_T(.)$は$H(.)$とは異なるハッシュ関数であり、｜は連結を示す。また、次いで点Ｖの導出元である新しい点を得るために、クライアントシークレットｓＡは時間許可証ｓＴに追加され

る。これは、表記は異なっているが上述された方法と数学的には同等である。

アリス - アイデンティティID_α	サーバ
ランダム$x < q$を生成する	ランダム$y < q$を生成する
$A = H(ID_\alpha)$	
$T = H_T(T_i \mid ID_\alpha)$	
$D = A + T$	
$U = xD$	
$R = xA$	
$ID_\alpha, U, R \rightarrow$	$\leftarrow y$
$V = -(x + y)((s - \alpha)A + \alpha A + sT) \rightarrow$	
	$D = H(ID_\alpha) + H_T(T_i \mid ID_\alpha)$
	$g = e(V, Q).e(U + yD, sQ)$
	$g \neq 1$の場合、接続を拒否する

表９-時間許可証を有するＭ－Ｐｉｎ

【０３１０】

ＰＩＮエラーを決定するための機構に関して、ここで、プロトコルによって返されるｇの値はｇ＝ｅ（Ｒ＋ｙＡ，Ｑ）$^{\delta}$であり、上式でδはＰＩＮエラーである。

【０３１１】

表９のプロトコルは、ここで図１１及び図１２に関してより詳細に説明される。

【０３１２】

ステップ１１．１で、クライアント２は、ＴＡから時間許可証ｓＴを得る。クライアントは、ＴＡ４から時間許可証ｓＴに相当する点の座標を得てよい。また、クライアントは、ＴＡから時間許可証が有効である期間を示すＴ$_i$も得てよい、またはクライアントはそれを内部で生成してよい。例えば、期間は、特定のフォーマットで作成される、時間許可証が有効である日付であってよい。ステップ１１．２及びステップ１１．３は、図６のステップ６．１及びステップ６．２と同じである。クライアントはＰＩＮ予想値１１’をユーザーから受信し、クライアントはそのアイデンティティをＧ$_1$の楕円曲線上の点にハッシュする。ステップ１１．４で、クライアントは次いでアイデンティティ及び期間Ｔ$_i$を連結し、連結されたストリングをＧ$_1$の楕円曲線上の点Ｔにハッシュする。クライアントは次いで、ステップ１１．５で、新しい点Ｄの座標を得るために点Ａ、及び点ｔを楕円曲線上で追加する。ステップ１１．６で、クライアントは次いで２つの乗算を曲線上で実施する。クライアントは、ランダムｘを生成し、点Ｄ及び点Ａの両方ともをｘで乗算して、２つの新しい点Ｕ及びＲを得る。ｘの値は群Ｇ$_1$、Ｇ$_2$、及びＧ$_T$の素数位数ｑよりも小さい。クライアントは次いで、ステップ１１．７で、クライアントのアイデンティティ及び点Ｕ及びＲの座標を含んだコミットメントをサーバ３に送信する。いくつかの例では、クライアントはサーバに期間Ｔ$_i$を送信してもよい。

【０３１３】

ステップ１１．８で、クライアントは、クライアントが図６のステップ６．５でチャレンジｙを受信したのと同じようにチャレンジを受信する。ｙの値はやはりｑよりも小さい乱数である。クライアントは、次いで点Ｖの座標を計算するために進む。表６のプロトコルでは、この点の座標は、取り戻されたクライアントシークレットに相当する点を－（ｘ＋ｙ）で乗算することによって導出された。表９のプロトコルでは、Ｖは代わりにトークン及びＰＩＮから得ることができるクライアントシークレットｓＡ、並びにＴＡから得られる時間許可証ｓＴから導出される。より詳細には、ステップ１１．９で、クライアントは、図６のステップ６．６と同様にＰＩＮ及びトークンからクライアントシークレットを

取り戻す。また、クライアントは２つの乱数値ｘ及び６を互いに加算し、－１で乗算して、－（ｘ＋ｙ）を得る。クライアントは次いで、楕円曲線上で、クライアントシークレットｓＡに相当する点、及び時間許可証ｓＴに相当する点を加算して、新しい点を見つけ出し、次いで曲線上で新しい点の値－（ｘ＋ｙ）での乗算を実施して、点Ｖの座標を得る。クライアントは次いで、サーバがクライアントを認証できるようにするためにこの点の座標をサーバに送信する。ステップ１１．１０で、クライアントは認証の結果を受信してよい。ステップ１１．１０はより詳細に以下に説明される。

【０３１４】

図１２に関して、サーバ３はステップ１２．１で期間T_iを得る。サーバはＴＡ４から期間を得て、例えば期間が日付等の既知のフォーマットである場合、内部で期間を生成する、またはクライアント２から期間を受信することができる。サーバは次いで、ステップ１２．２で、クライアントの主張されているアイデンティティ及び楕円曲線上のG_1の点Ｕ及びＲの座標を含んだ、認証するというコミットメントを受信する。また、サーバは、クライアントが接続を確立することを希望することを示す別のメッセージをクライアントから受信した可能性もある。サーバは、アイデンティティ並びに点Ｕ及びＲの座標を送信するようにクライアントに要請した可能性がある。いくつかの例では、ステップ１２．１及び１２．２は結合でき、サーバはコミットメントで期間T_iを受信する。ステップ１２．３で、サーバは次いで、やはりｑよりも小さいランダムｙを生成し、チャレンジとしてｙの値をクライアントに送信する。サーバは次いで、ステップ１２．４でクライアントアイデンティティ及び期間を使用して、点Ｄの座標を計算する。ステップ１２．５で、サーバは楕円曲線上の点Ｖの座標を受信する。

【０３１５】

サーバは、次いでステップ１２．６でペアリングの積を計算する。第１のペアリングは、入力として点Ｖ及び点Ｑを採り、該点をG_Tの要素にマッピングする。図１１の説明に示されるように、Ｖは、クライアントシークレットｓＡに相当する点、及び時間許可証ｓＴに相当する点から得られるG_1の曲線上の点である。すでに上述されたように、ＱはG_2の固定生成元である。他のペアリングは、入力として、点Ｕ及びＤ、並びにサーバシークレットｓＱに相当する点から得られる、G_1の別の点の座標を採り、該点をG_Tの要素にマッピングする。点ｓＱは言うまでもなくG_2の別の点である。G_1の点は、楕円曲線上で、ステップ１２．２のコミットメントで座標がクライアントから受信されたG_1の点Ｕと、及びステップ１２．４で計算された点Ｄを乱数値ｙで乗算することによって得られる、言うまでもなくやはりG_1の点とを加算することによって得られる。サーバは、値ｇを与えるために２つのペアリングの積を得る。計算は、サーバが、別々のペアリングを、それらを互いに乗算する前に実行するよりむしろ、ペアリングの積を１つの計算として計算するマルチペアリングとして実行されてよい。サーバシークレットｓＱが、それぞれが別個のモジュールによって記憶される２つの部分に分割される場合、各モジュールは代わりにそのそれぞれのシークレット部分に基づくペアリングを計算し、２つのペアリングは次いで、ｅ（Ｕ＋ｙＤ，ｓＱ）を得るために互いに乗算される。サーバシークレットs_1を記憶するモジュールは、次いでｅ（Ｕ＋ｙＤ，s_1Ｑ）を計算し、サーバシークレットs_2を記憶するモジュールは次いでｅ（Ｕ＋ｙＤ，s_2Ｑ）を計算するだろう。

【０３１６】

図７に関して上述されたように、双線型性のため、正しいＰＩＮが入力され、したがって正しいクライアントシークレットが取り戻された場合、ｇの値は１に等しくなるはずである。ステップ１２．７で、サーバ３は、ｇの値が１に等しいかどうかをチェックし、ｇの値が１に等しい場合、サーバ３はステップ１２．８でクライアントを認証し、ステップ１２．９で接続を受け入れる。接続を受け入れることは、クライアントに、クライアントが認証されたことを知らせるためにメッセージを送信することを含んでよい。

【０３１７】

逆に、ｇの値が１に等しくならない場合、サーバはステップ１２．１０で、クライアントが認証されていないと判断し、次いでいくつかのやり方で進むことができる。ステップ

１２．１１で、サーバはただちに接続を拒否することができるか、またはサーバは、ＰＩＮが正しくなかったに違いない範囲を決定するためにエラー処理プロセスを実施できるかのどちらかである。サーバは、例えば図８に関して説明されるプロセスを実行してよい。

【０３１８】

図１１及び図１２のステップのいくつかの順序を変えることができることが理解されるだろう。例えば、ＰＩＮは、それがステップ１１．９で使用されるまで受信されないことがある。さらに、期間及び時間許可証は、それらがそれぞれステップ１１．４及びステップ１１．９で使用されるまでクライアント側で得られないことがある。さらに、サーバは、サーバが点Ｖの座標を受信した後にクライアントアイデンティティをハッシュしてよい。また、サーバは、サーバがステップ１２．４で期間を使用するまで期間を得ることを延期してもよい。さらに、サーバはシークレットの第２の因子でのエラーのためにより高度なエラー処理プロセスをサポートしない場合、点Ｒの座標はサーバに送信されなくてもよい。

【０３１９】

サーバは、当業者が理解するように、ペアリングを実行するために任意の適切なペアリングを使用してよい。例えば、クライアントは、効率的なＲ－エイトペアリングを使用してよい。さらに、楕円曲線が使用されることが説明されているが、任意の適切な代数曲線が使用されてよい。例えば、楕円曲線の代わりに、超楕円曲線が使用されてよい。

【０３２０】

やはり時間許可証を含めるための表８のＭ－Ｐｉｎ　Ｆｕｌｌプロトコルの修正形態が表１０に示される。

アリス - アイデンティティID_α	サーバ
ランダム x, $w < q$ を生成する	ランダム n, $y < q$ を生成する
$A = H(ID_\alpha)$	
$T = H_T(T_i \mid ID_\alpha)$	
$D = A + T$	
$U = xD$	
$R = xA$	
$ID_\alpha, U, R \rightarrow$	
$m = -(x + y)$	
$V = m((s - \alpha)A + \alpha A + sT) \rightarrow$	$\leftarrow y$
	$D = H(ID_\alpha) + H_T(T_i \mid ID_\alpha)$
	$t = e(V, Q)$
	$g = t.e(U + yD, sQ)$
	$g \neq 1$の場合、接続を拒否する
$W = wD \rightarrow$	$\leftarrow r = t^n$
$k = r^{w/m}$	$k = e(nW, sQ)$
$K = H_g(k)$	$K = H_g(k)$

表１０-時間許可証を有するＭ－Ｐｉｎ　Ｆｕｌｌ

【０３２１】

表９に関して上述されたように、ここではプロトコルによって返されたｇの値はｇ＝ｅ（Ｒ＋ｙＡ，Ｑ）$^{\delta}$であり、δはＰＩＮエラーである。

【０３２２】

表１０から明らかになるように、修正形態は、ステップ９．２で、点Ｗの座標を得るために、点Ａの乗算の代わりに、ｑよりも小さい乱数値ｗを用いて点Ｄの乗算が実施され、新しい点Ｗの座標がサーバに送信される以外に、時間許可証が使用されるときに、図９及び図１０に関して説明される鍵導出ステップに対して必要とされないことがある。しかしながら、プロトコルの始まりのステップは、図１１及び図１２に関して説明されるように

修正されてよい。

【０３２３】

いくつかの例では、Ｍ－Ｐｉｎプロトコルは、３つの当事者、つまり単一サーバ、多くのクライアント、及び信頼機関（ＴＡ）を含むプロトコルであることを覚えておくことが重要である。これの主要な利点の内の１つは、多くの責任がサーバから取り除かれ、ＴＡの手に移される点である。特に、サーバはクライアントシークレットに関係するデータを保持していない。これが、ハッキングされているサーバの問題を大幅に軽減する。

【０３２４】

慎重な設計により、サーバはアクティブクライアントステータスの動的なデータベースを維持するだけでよい。このデータベースは、例えば図３に示されるストレージ３３内等、サーバに記憶され、プロセッサがデータベースのデータに迅速にアクセスする必要があるときにプロセッサ３１の一時メモリにロードされてよい。例えば、新しいユーザーがログインに成功すると、サーバは、ユーザーがＴＡによって有効なクレデンシャルを発行されたことを知るため、追加チェックを行う必要がない。ここでセキュリティ専門知識はＴＡを制御する者の手の中だけにある。つまり、サーバはセキュリティの専門家外によって実装され、維持されるために信頼できる。

【０３２５】

時間許可証機構はＴＡが取消しを制御し、この潜在的な負担からサーバを解放する方法である。時間許可証はマスターシークレットを必要とするので、その発行はＴＡ４の裁量による。

【０３２６】

ＴＡは、例えばすべての時間許可証を放送してよい。クライアントは、例えば、ＴＡと関連付けられたウェブサイトにアクセスすることによって時間許可証２７を得ることができてよい。また、インポスターもクライアントの時間許可証を得ることができることがあるが、インポスターはクライアントシークレットｓＡを知らないので、この時間許可証の使い道がない。ＴＡが、引き続きクライアントがシステム及び関連するプロトコルを使用して、鍵を認証及び／または導出できるようにすることを望むかどうかについての決定は、ＴＡで新しい期間が始まるたびに下されてよく、ＴＡが望む場合、ＴＡは、関連するサーバと関連付けられたマスターシークレットｓ、クライアントアイデンティティ、及びクライアントが使用する期間から得られる時間許可証を発行する。

【０３２７】

いくつかの例では、時間許可証の代わりに、または時間許可証に加えて、クライアントシークレットｓＡの生成に追加の情報を含むことができる。時間許可証ｓＴは、例えば以下で置き換えることができ、

$$sT = s.H_T(ID_\alpha \mid T_i \mid Z)$$

上式で、ＺはＴＡ４が頼みにしている当事者に送信することを希望するデータである。クライアントは、時間許可証、及びデータも平文で受信する。クライアントは、いくつかの例では、プロトコルの部分としてこのデータを認証サーバ３に送信してよい。これは、表９及び表１０のプロトコルから明らかになるように、認証サーバが、

$$D = H(ID_\alpha) + H_T(ID_\alpha \mid T_i \mid Z)$$

を計算できなければならないためである。

【０３２８】

図１１及び図１２に関して、クライアント２は、ＴＡから追加データＺを得て、サーバがステップ１２．２で受信するために、例えばステップ１１．７のコミットメントで追加データを送信してよい。代わりに、サーバは、ステップ１２．１でクライアントまたはＴＡのどちらかから追加データを受信してよい。さらに、ステップ１１．４で、クライアントは点Ｔを得るために追加データＺも使用する。クライアントはクライアントアイデンティティ、期間、及び追加データを連結し、ハッシュ関数H_Tを使用して、G_1の楕円曲線上の点Ｔに結果をハッシュしてよい。

【０３２９】

図１３に関して、システムは、複数のクライアント２ａ、２ｂ、認証サーバ３、ＴＡ４、及び頼みにしている当事者１６を含んでいるとして示される。頼みにしている当事者１６は、サーバ、またはクライアントが接続を確立することを希望する他のコンピューティング装置であってよい。例えば、頼みにしている当事者は、クライアントがアクセスを希望するデータを記憶してよい、または頼みにしている当事者はインターネットショッピングサイトをホストしていてよい。

【０３３０】

ＴＡ４は、クライアント２ａにその時間許可証ｓＴ及び追加データＺを提供する。また、クライアントは、例えばＴＡから、または上述されたように期間T_iを内部で生成することによって期間T_iを得てもよい。また、認証サーバは期間T_iも得てよい。認証サーバ３は、それ自体を認証サーバに対して認証するクライアント２ａを通じて追加情報Ｚも提供されてよい。認証サーバ３は、いったんクライアントがサーバによって認証されるとクライアント２ａに認証トークン１７を発行してよく、クライアントはこの認証トークンを、頼みにしている当事者１６に提示することができ、頼みにしている当事者１６は次いで、トークンに基づいてリソースへのアクセスを許す、または妨げてよい。トークン１７は、アイデンティティ、成功／失敗、タイムスタンプ、及びＴＡから送信されたデータＺのためのフィールドを含んでよい。トークン１７は認証サーバによって署名され、ＡＥＳを使用して暗号化されてよく、ＡＥＳ鍵は、Ｓａｋａｉ－Ｋａｓａｈａｒａ鍵暗号化（ＳＡＫＫＥ）を使用してカプセル化されてよい。暗号化されたトークンは、例えばクライアントが頼みにしている当事者のコンテンツにアクセスすることを希望する場合、頼みにしている当事者１６にそれを提示しなければならないクライアントに送信される。頼みにしている当事者は、トークンを解読し、署名を検証し、データペイロードを抽出できる。したがって、データは、改ざんされることなくＴＡ４から頼みにしている当事者１６に送信されている。図１３で、第１のクライアント２ａだけがその時間許可証、追加データＺ、及び期間を所有しているとして示されているが、システムを使用するすべてのクライアントは、専用の時間許可証、追加データ、及び関連する期間を与えられてよい。

【０３３１】

ＴＡは、認証サーバ３を実行するカスタマにとって役立つことがある豊富なリアルタイム解析データのデータベースを維持する。時間許可証の生成で追加情報を含むための上述された方法は、クライアント自体を認証サーバに認証するクライアントを介して頼みにしている当事者にデータを送信する方法として使用できる。さらに、認証サーバがそのＰＩＮエラー処理プロセスを実施するために役立つことがあるクライアントについての追加データは、このようにして送信されてよい。例えば、図８のステップ８．１１及びステップ８．１３で検討される追加データのいくらかは、このようにして認証サーバに送信されてよい。

【０３３２】

別の鍵共有プロトコル

表８のＭ－Ｐｉｎ　Ｆｕｌｌプロトコルの代替策は、鍵を確立し、クライアントを認証するために使用されてよい。開始点は、表５の最終プロトコルである。これは、Ｗａｎｇ［２３］による、ペアリングを使用する完全認証鍵交換プロトコルに基づいているが、修正を有する。このプロトコルでは、ｇを計算し、鍵Ｋが計算される前にｇ＝１かどうかをチェックする代わりに、Ｋが計算され、鍵Ｋから導出される値Ｍが次いでクライアントからサーバに送信されて、サーバがクライアントの認証を実行できるようにする。このプロトコルは表１１に示される。

アリス - アイデンティティID_α	サーバ
ランダム$x,m < q$を生成する	ランダム$y,w < q$を生成する
$A = H_1(ID_\alpha)$	
$U = xA$	
$V = m((s-\alpha)A + \alpha A)$	
$ID_\alpha, U, V \rightarrow$	
	$A = H_1(ID_\alpha)$
	$W = wA$
	$t = e(yV, Q)$
	$\leftarrow t, W$
$k = t^{(x+1)/m}$	
$K = H(k\|xW)$	
$M = H(ID_\alpha\|U\|V\|W\|K)$	
$M \rightarrow$	
	$k = e(y(A+U), sQ)$
	$K = H(k\|wU)$
	$N = H(ID_\alpha\|U\|V\|W\|K)$
	$M \neq N$の場合、接続を拒否する

表１１．鍵共有及び認証のためのプロトコル

【０３３３】

表８のプロトコルと同様に、クライアントはもはやペアリングを計算する必要はない。サーバは二を計算する必要がある。また、クライアントは群Ｇ２でいかなる算術も行う必要はない。実際に、すべての点乗算はより簡略な群Ｇ１にある。再び、クライアント及びサーバの双方とも同じ鍵で終了することを確認することは読者にとっての簡単な練習として残される。鍵のハッシュは実際にはサーバに送信されるので、表８のプロトコルのＭ－Ｐｉｎ　Ｆｕｌｌとは異なり、表１１の代替プロトコルがゼロ知識証明プロトコルではないことが観察される。

【０３３４】

表１１のプロトコルは、ＰＩＮエラーを決定し、ＰＩＮの範囲に基づいてクライアントに応答するためにも使用できる。例えば、図８のエラー処理プロセスは、プロトコルの鍵共有部分が実施される前にプロトコルとともに使用できるだろう。

【０３３５】

セットアップ

ここで、図１４から図１６に関してセットアッププロセスの例がより詳細に説明される。

【０３３６】

図１４に関して、ステップ１４．１で、ＴＡ４はサーバ３からサーバシークレットを得る要求を受信する。ステップ１４．２で、ＴＡは次いで認証プロセスで使用される楕円曲線のパラメータ及びハッシュ関数の詳細をサーバに提供する。ＴＡは、認証プロセスで使用される楕円曲線およびハッシュ関数の詳細をすでに選択した可能性がある。楕円曲線パラメータ及びハッシュ関数詳細は、事前選択された可能性があり、情報４６はＴＡのストレージ４３に記憶された可能性がある。また、ＴＡは曲線のパラメータ、ハッシュ関数の詳細、及びサーバが認証プロセスを実行できるようにするための他の公開情報を公開した可能性もある。結果的に、ステップ１４．２は、サーバを情報が公開されたところへ導くことを含んでよい。代わりに、ＴＡはサーバに情報を送信してよい。サーバがＴＡからサーバ認証プログラム３４を受信する場合、プログラムは、サーバが必要とするパラメータ及び他の情報を含んでよい。

【０３３７】

ＴＡは、次いでステップ１４．３でサーバにとって一意であるマスターシークレットｓを生成する。ステップ１４．４で、ＴＡは次いで楕円曲線で点Ｑを生成し、選択された曲線上で点Ｑをマスターシークレットｓで乗算することによって点ｓＱを得る。ｓはいくつかの実装ではサーバごとに一意であるので、サーバのアイデンティティは、それらの実装でのＱ及びｓＱの生成では必要とされない。点Ｑは、任意に選択されてよい。ＴＡは安全な通信チャネルで点Ｑ及び点ｓＱの座標をサーバ３に送信する。いくつかの例では、ステップ１４．２は、ステップ１４．４と組み合されてよく、ＴＡ４はステップ１４．４でサーバが必要とするすべての情報を送信してよい。また、ＴＡは、ステップ１４．５で、サーバのための識別情報を受信し、システムを使用するサーバ及びクライアントを示すストレージ４３のレコード４５に追加してもよい。レコードは、例えばデータベースまたは他のデータ構造であってよい。

【０３３８】

少し経ってから、ＴＡ４は、ステップ１４．６でサーバ４と認証することを希望するクライアント２のアイデンティティを受信する。ＴＡ４は、例えばサーバから、またはＴＡがどのサーバを認証することを希望するのかの表示とともに、クライアントから直接的にアイデンティティを受信する。ＴＡは、クライアントがステップ１４．７で認証することを希望するサーバにとって一意であるマスターシークレットｓを取り出し、サーバがクライアントに送信するクライアントシークレットを生成する。クライアントシークレットは、上述されたように曲線上の点Ａにクライアントアイデンティティをハッシュし、この点をマスターシークレットｓで乗算して、曲線上の新しい点ｓＡを得ることによって生成される。また、ＴＡは、ＴＡがステップ１４．８で公開するまたはクライアントに送信するクライアント用の時間許可証を生成してもよい。また、ステップ１４．９で、ＴＡはクライアントのアイデンティティを、システムを使用するクライアント及びサーバのレコード４５に追加してもよい。曲線のパラメータ及びハッシュ関数の詳細が公開されておらず、他の手段によってクライアントに提供されない場合、ＴＡはその情報を、クライアントが必要とすることがある他の情報とともにクライアントに送信してもよい。

【０３３９】

少し経ってから、ステップ１４．１０で、時間許可証が発行された期間が満了すると、ＴＡ４はそのレコード４５をチェックし、まだシステムを使用する資格のあるクライアントを決定する。ステップ１４．１１で、ＴＡは、まだシステムを使用する資格のあるクライアントに新しい時間許可証を発行する。例えば、クライアントは、認証サービスを使用する料金を支払う必要がある場合があり、ＴＡは料金を支払ったクライアントに対して時間許可証を生成し、発行するにすぎない。

【０３４０】

ここで、図１５に関してクライアント２でのセットアッププロセスの例が説明される。プロセスは、ＴＡによって制御されている認証システムを使用している認証サーバ３が認証側サーバの機能を果たすサーバまたはリソースにアクセスしようと試行するクライアントによってトリガされてよい。クライアントは、ステップ１５．１でＴＡからクライアントシークレットを受信する。また、クライアントは、ステップ１５．２でＴＡから、使用するための楕円曲線のパラメータ及びハッシュ関数の詳細も得る。例えば、ＴＡがウェブサイトでこの情報を公開している場合、クライアントはウェブサイトから、クライアントがサーバに対して認証するために必要とすることがある他の公開情報とともに該情報を得てよい。いくつかの例では、クライアントはＴＡからの初期化プログラム２４を提供されてよく、プログラムはすでに該情報を含んでいてよい。初期化プログラムは、クライアントがそのシークレットを受信する前にクライアントに提供されてよい。ステップ１５．３で、ユーザーインタフェースを使用して、クライアントは、ユーザーにＰＩＮを入力するように要請する。ＴＡからの、クライアントのブラウザで実行される初期化プログラムは、ユーザーがＰＩＮを入力するためのグラフィックユーザーインタフェースを提供してよい。ステップ１５．４で、クライアントはトークンを生成する。クライアントは、曲線上の点Ａにクライアントのアイデンティティをハッシュするために指定されたハッシュ関数

を使用してよい。クライアントは、次いで、新しい点αＡを得るために、曲線上で点Ａを受信されたＰＩＮ値で乗算する。クライアントは、トークン（ｓ－α）を得るために、クライアントシークレットに相当する点からこの点を除算する。クライアントは、次いでそのアイデンティティとともにストレージにこのトークン１０を記憶する。

【０３４１】

サーバ３のセットアッププロセスの例がここで説明される。サーバは、ステップ１６．１でＴＡ４に記録する要求を送信する。ステップ１６．２で、サーバは点Ｑ及びサーバシークレットｓＱに相当する点の座標を受信し、これらを安全に記憶する。少し経ってから、クライアントは認証しようと試みるとき、クライアントはＴＡ４にクライアントのアイデンティティを送信する。また、クライアントは、それがクライアントを認証するためのプロトコルを実施できるようにするために、クライアントが必要とすることがある他の情報とともに曲線パラメータ及びハッシュ関数詳細を調べてもよい。

【０３４２】

図１４、図１５、及び図１６に関して説明されるプロセスは例にすぎず、ＴＡ、クライアント、及びサーバでセットアップを実行するための他のプロセスが期待される。例えば、時間許可証が使用されない場合、時間許可証に関係するステップは実施されない。さらに、図１４、図１５、及び図１６のステップは特定の順序で説明されているが、いくつかのステップの順序は、当業者によって理解されるように変えられてよい。

【０３４３】

シークレットのチャネル

図８の方法はエラー処理プロセスとして説明されたが、類似する方法は、エラーが故意に入力された場合にも使用できるだろう。例えば、ユーザーの銀行口座またはユーザーが保護を希望する別のリソースにアクセスするために、他の人の前でＰＩＮを入力するように別の人物によって強制されているユーザーは、事前に合意されたエラーをＰＩＮに入力することがある。認証サーバは特定のエラーを、ユーザーがそのＰＩＮを入力するように強制されているしるしとして認識し、接続を拒否する。認証サーバはユーザーを支援するために自動的に追加のステップを講じてよい。したがって、システムはシークレットのチャネルを提供する。

【０３４４】

シークレットのチャネルは、ユーザーがＰＩＮを入力するように強制されていることを示す情報以外の他の情報をサーバに送信するためにも使用できる。入力されたＰＩＮと実際のＰＩＮとの差異はまったくエラーではなく、クライアントサーバとの間で実際に送信されることなくサーバ側に表示されるクライアントからのメッセージまたはメッセージの一部にすぎないことがある。

【０３４５】

上述されたシステムのシークレットのチャネルは、狭帯域サブリミナルチャネルと見なされることがある。該チャネルは、プロトコルの通常の（失敗した）実行のように外部観察者には見える（事実上、プロトコルの通常の（失敗した）実行と区別できない）一方、クライアントと（互換性のあるシークレットを発行されている）サーバとの間で少量の情報を無意識に渡すことを可能にする。

【０３４６】

サーバにメッセージを伝達するためにサブリミナルチャネルが使用されるいくつかのアプリケーションでは、本明細書に説明されるプロトコルのいくつかはクライアントを認証するために使用されないことがあるが、クライアントが実際に情報を送信することなく、情報がサーバで表示できるようにするためだけに使用されてよい。ＰＩＮは４個の数字しか含んでいないと再び仮定すると、ユーザーがメッセージまたは４を超える文字を必要とするデータを通信することを希望する場合、クライアントはＴＡからいくつかの異なるシークレットを得て、シークレットごとにＰＩＮをセットアップできる。

【０３４７】

情報を通信するこのプロセスの例は、ここで図１７、図１８、及び図１９に関して説明

される。プロセスは、クライアントがサーバにクレジットカード情報を送信することなく、クライアント２からのクレジットカード情報をサーバ３に表示させることの関連で説明される。ただし、方法が、メッセージを表す任意のデータ列を、実際に該ストリングをクライアントに送信することなくサーバに通信するために使用できることが理解される。

【０３４８】

ＴＡで実施されるステップがここで説明される。ステップ１７．１で、ＴＡ４は、クライアントがサーバに安全に情報を通信できるようにするために登録する要求をクライアントから受信する。メッセージは、クライアントがサーバに通信することを希望するクレジットカード情報を含んでよい。ステップ１７．２で、ＴＡ４は受信されたクレジットカード情報と関連付けられた複数のクライアントシークレットを発行するために進む。クライアントシークレットは、カードの裏側の、ＣＶＣ２番号、または他のセキュリティコードを加えたクレジットカード番号の４つの数字ごとに１つのクライアントシークレットを含んでよい。クライアントシークレットは、クレジットカード番号全体のための１つのクライアントシークレットも含んでよい。結果的に、クレジットカード番号は、カードの裏側の３つの数字を加えて１６の数字を有し、ＴＡは６つのクライアントシークレットを発行する必要がある。クライアントシークレットは、マスターシークレットｓを使用して確立される。クライアントシークレットは、クライアントがクレジットカード情報を伝達することを希望するサーバのサーバシークレットを導出するためにも使用されたマスターシークレットｓを使用して確立される。セキュリティコードを加えたクレジットカード番号の各部分の識別情報、及びクレジットカード番号全体の識別情報は、選択され、曲線上の点にハッシュされてよく、点は次いで、シークレットを得るために、上述されたようにマスターシークレットで乗算される。識別情報は、クレジットカード番号の情報を明らかにしない任意の情報であってよい。識別情報は、本明細書では、多様なシークレットと関連付けられるアイデンティティと呼ばれる。クレジットカード番号のためのクライアントシークレットがどのようにして生成されるのかの上記の説明が一例にすぎず、クライアントシークレットが任意の適切な方法で確立されてよいことが理解される。クライアントがクレジットカードの有効期限及び他の情報を送信する必要がある場合、追加のシークレットが必要とされることがある。

【０３４９】

ステップ１７．３で、ユーザーがＰＩＮ番号を選択する代わりに、ＴＡ４がクライアントシークレットのためにＰＩＮ番号も事前に選択する。ＰＩＮ番号のいくつかは、順番に、それぞれカード上に表示される４つの数字に相当する。結果的に、第１のクライアントシークレットは、ＰＩＮ番号として選択されたカード番号の最初の４つの数字を有してよく、第２のクライアントシークレットはＰＩＮ番号として選択されたカード番号の第２の４つの数字を有する等である。第５のクライアントシークレットはＣＶＣ２番号及びＰＩＮ番号として選択されたヌルのための０を有する。さらに、第６のクライアントシークレットは、ＰＩＮとして選択された裏側のＣＶＣ２番号の数字の数に加えて、カード番号である数字の数を有する。結果的に、それがカードの裏側の３桁のＣＶＣ２番号を加えた１６桁のクレジットカード番号である場合、ＰＩＮは「１９００」となるだろう。

【０３５０】

例えば、７６７のＣＶＣ２が加えられた６４５６　６５６５　７８７８　９８９８のカード番号の場合、第１のＰＩＮは６４５６として選択され、第２のＰＩＮは６５６５として選択され、第３のピンは７８７８として選択され、第４のピンは９８９８として選択される。第５のＰＩＮは７６７０として選択される。さらに、第６のＰＩＮは上述された「１９００」として選択されてよい。

【０３５１】

ＰＩＮは、次いで６個のトークンを生成するためにシークレットから抽出され、トークンはステップ１７．４での応答でクライアントに送信される。多様なシークレットと関連付けられるアイデンティティもクライアントに送信される。アイデンティティはステップ１７．４で応答に含められてよい。トークン及びアイデンティティはクライアントのソフ

トウェアウォレットに送信されてよい。例えば、ソフトウェアウォレットはブラウザのストレージに位置してよい。代わりに、シークレット、アイデンティティ、及びＰＩＮはクライアントに送信され、クライアントはそれ自体ＰＩＮを生成してよい。また、ＴＡは、クライアントが、それがすでにそうしていない場合、クレジットカード情報を送信することを希望するサーバにサーバシークレットを発行してもよい。

【０３５２】

クライアント側で実施されるステップは、ここで図１８に関して説明される。ステップ１８．１で、クライアント２はクレジットカード情報とともにメッセージをＴＡ４に送信する。ステップ１８．２で、クライアント２は６つのクライアントシークレットのためのトークン及びＴＡ４から戻される多様なクライアントシークレットと関連付けられるアイデンティティを含む応答を受信する。代わりに、クライアント２は６つのクライアントシークレット及び関連付けられたＰＩＮを受信してよく、トークン自体を生成するためにシークレットからＰＩＮを抽出してよい。少し経ってから、クライアントはシークレットのチャネルでシークレット情報をサーバ３に通信するためのプログラムを起動する。このプログラムは、認証プログラム２５の部分を形成することもあれば、メモリ２２に記憶される別のプログラムであることもある。該プログラムはサーバ３からウェブインタフェースを介してクライアントに提供されてよい。

【０３５３】

ステップ１８．３で、クライアントは、クライアントがシークレットのチャネルでサーバに情報を通信することを始めることを示すメッセージをサーバに送信する。ステップ１８．４で、クライアントは、全体のメッセージが送信されるために必要となるＰＩＮの番号と等しいようにパラメータＰを設定する。パラメータＰは、実際のＰＩＮとクライアントで使用されたＰＩＮの差異をサーバで何回決定する必要があるのかを制御する。上述されたクレジットカード例では、６つの別々のクライアントシークレットがあるので、Ｐは６に設定される。また、クライアントはカウンタｐをゼロとなるように設定する。ステップ１８．５で、クライアントはカウンタｐの値を１に増加して、メッセージの第１の部分を伝達するために準備する。それから、クライアントはサーバにコミットメントを送信し、ステップ１８．５でサーバからチャレンジを受信する。ステップ１８．６で、クライアントは次いで、メッセージの第１の部分と関連付けられた記憶されていたトークンを取り出す。ステップ１８．７で、クライアントは、「シークレット」を作成し、Ｖを生成するために「正しくない」ＰＩＮ予想値「０００００」を使用する。コミットメント及びチャレンジの交換、並びにＶの計算は、上記表６、図８、図９、図１０、または図１１のプロトコルに関して説明されるように実施できる。例えば、クライアントは、図６のステップ６．２からステップ６．６、または他のプロトコルについて説明された対応するステップを使用してよい。ユーザーからＰＩＮを受信する代わりに、クライアントはＰＩＮ「００００」を使用して、次いでＶを作成するために使用される「ダミーシークレット」を作成する。

【０３５４】

シークレットを作成するために使用されるＰＩＮはシークレットをＰＩＮ及びトークンに分割するためにＴＡによって最初に入力されたＰＩＮではないので、「シークレット」はメッセージの部分と関連付けられたアイデンティティのためにＴＡによって発行される本当のクライアントシークレットではない。

【０３５５】

クライアントは、次いで、ステップ１８．８で、ｐがパラメータＰより小さいかどうかをチェックする。ｐがＰよりも小さい場合、メッセージの第２の部分と関連付けられたトークンについて、ステップ１８．５から１８．８が繰り返される。ステップ１８．５から１８．８は、実際にメッセージのいずれの部分も送信することなくメッセージ全体がサーバに通信されるまで繰り返される。例として表６のプロトコルを使用すると、ステップ１８．５が繰り返されるたびに、ｐは１、増加し、メッセージの次の部分のアイデンティティが新しいＵを生成するために使用され、アイデンティティ及び新しいＵはサーバに送信

される。さらに、ステップ１８．６が繰り返されるたびに、メッセージの新しい部分と関連付けられたトークンは「正しくない」ＰＩＮ予想値「０００００」と結合されて、サーバに送信される新しいＶの座標を生成する。カウンタｐがパラメータＰに等しいとステップ８で判断されると、プログラムは終了する。

【０３５６】

ここでサーバ側のステップが、図１９に関して説明される。サーバはステップ１９．１で、クライアントがシークレットのチャネルで情報の通信を開始することを示すメッセージを受信する。サーバはコミットメントを受信し、ステップ１９．２でチャレンジを生成し、送信する。ステップ１９．３で、点Ｖの座標を受信し、サーバは、クライアントでＶを導出するために使用されるＰＩＮと最初にトークンを生成するために使用されたＰＩＮとの間の差異を決定する。サーバはＴＡから独自のシークレットｓＱを発行されているので、サーバはこれを行うことができる。サーバはコミットメント及びチャレンジの交換、ならびに上記の表６、表８、表９、表１０、または表１１のプロトコルに関して説明されたように、入力としてＶを採るペアリング計算を実行してよい。例えば、クライアントは、図７のステップ７．２からステップ７．５、または他のプロトコルについて説明された対応するステップを使用してよい。サーバはｇについて得られた値からδの値、及び時間許可証のないプロトコルが使用されるときの関係性$g = e(U + yA, Q)^{\delta}$、または時間許可証が使用されるときの関係性$g = e(R + yA, Q)^{\delta}$を決定する。

【０３５７】

クライアントは次いで、ステップ１９．４でＰＩＮ差異を記憶し、ステップ１９．５でさらに多くの情報を送信するかどうかをチェックする。クライアントがさらに多くの情報を送信する場合、サーバは、サーバが、クライアントが伝達を希望するすべての情報を取り戻すまでステップ１９．２からステップ１９．４を繰り返す。いくつかの実装では、サーバがすでにＰＩＮ差異決定プロセスを何回繰り返す必要があるのかを知っているように、クライアントはＰの値をサーバにも送信してよい。サーバが、完全なメッセージが受信されたと判断すると、サーバは次いでメッセージを処理するために進む。結果的に、蒸気の例のクレジットカードを使用して、サーバは、サーバが６４５６６５６５７８７８９８９８７６７０、及び最後のシークレットと関連付けられた情報も受信するまでプロセスを繰り返す。クレジットカード番号全体に対応する最後のシークレットと関連付けられる、ＰＩＮ差異決定プロセスの最後の反復で得られた情報は、クライアントシークレットでＰＩＮエラーの範囲を暴露することによってサーバが正しい数字をアセンブルしたことを確認するために使用される。

【０３５８】

サーバは、例えば、トークンで、及び安全なチャネル上で別のサーバにクレジットカード番号を転送して、クライアントが、実際にいずれのクレジットカード情報も送信することなく、クレジットカード情報を使用してそのサーバとのトランザクションを実施できるようにする他のサーバは、例えば銀行によって運用されるサーバであってよい。

【０３５９】

いくつかの例では、クレジットカード番号全体に基づくシークレットは、セキュリティコードを加えたクレジットカード番号のハッシュに基づいてよい。サーバは次いで、ＰＩＮ差異決定プロセスを使用してサーバがアセンブルしたものを、セキュリティコードが加えられたクレジットカードのハッシュ「ｈ（６４５６６５６５７８７８９８９８７６７）に対してチェックし、そのシークレットＰＩＮエラーの範囲を使用して、それがアセンブルされた情報から、この場合は１９である、何桁の数字が選び取られると仮定されるのかを知っていることを確かめることができる。

【０３６０】

クレジットカード番号がクライアントにいつも記憶される必要がないことが認識される。クレジットカード番号は、今までにサーバに送信されていない。サーバはクレジットカード番号を記憶する必要はなく、サーバは、サーバが支払いを行うように指示されるときに、クレジットカード番号及びＣＶＣ２番号をＰＩＮエラーの範囲からアセンブルするこ

とによってクレジットカード番号を一時的に作成するにすぎない。クライアントシークレットは、エンドユーザーのソフトウェアウォレットに、または、クライアントが支払いプロセッサに支払を行うように命令する必要があるときにそれらのシステムの業者によっても記憶できるトークン化システムとして機能する。さらに、ＣＶＣ２コードは上記に参照されてきたが、サーバ側でアセンブルされる情報は、さらにまたは代わりに、他のセキュリティコード情報、有効期限情報、及び／または他の情報を含むことがあることが認識される。

【０３６１】

図１７、図１８、及び図１９に関して説明されるプロセスは、サーバでシークレットクライアント情報を取り戻すためのＴＡ、クライアント及びサーバでのプロセスの例にすぎないことが理解される。他の実装が期待される。例えば、いくつかの例では、クライアント及びサーバはプロセスの始まりにコミットメント及びチャレンジを交換するだけでよい。クライアントは同じコミットメントのメッセージのすべての部分のＩＤを送信してよい、または同じＩＤがすべての部分に使用されてよい。同じＩＤがメッセージのすべての部分に使用される場合、複数のトークン及びＰＩＮが同じシークレットから抽出される必要があるか、またはＴＡが多くのクライアントシークレットを生成するために多くのマスターシークレットを使用するかのどちらかとなる。

【０３６２】

上述されたようなシークレットのチャネルの用途は多様である。例えば、Ｍ－Ｐｉｎエラー決定プロセスは、サーバが上述されたように購入を実施するために必要とされる有効期限及び他の情報とともにクレジットカード番号を取り戻すことができるようにするために使用されてよい。代わりに、それは、パスワード、銀行口座詳細またはクライアントが通信媒体全体で送信を希望しない他の情報を通信するために使用されてよい。

【０３６３】

本明細書で説明されるプロトコルは、１人または複数のクライアントをコンピューティングシステムまたはリソースに対して認証することを希望する任意の人物または組織によって使用できる。例えば、クライアントは、サーバにシークレット情報を送信することを希望してよく、サーバは、クライアントが、クライアントが自らが誰であると言っているものであること、及びクライアントが、クライアントが送信する情報の暗号化を希望することを立証することを希望することがある。また、プロトコルは、例えば会社または組織の内部でも使用されてよい。例えば、組織は独自のＴＡ及びいくつかの認証サーバをセットアップし、組織のメンバーにＩＤを発行してよい。そのようにして、組織内部のすべての通信は安全に送信され得る。

【０３６４】

本発明の特定の例及び実施形態が説明されてきたが、本発明の範囲は、添付の特許請求の範囲によって定められ、説明されている例及び実施形態に制限されていない。したがって、本発明は、他の方法で実現することができ、当業者によって認識されるように、多数の修正形態及び代替の構成が期待できる。

【０３６５】

既知のプロトコル、プロトコルの問題、及び安全なプロトコルの所望される特徴が事情を提供するためだけに説明されていることが理解される。重要な特徴、主要な特徴、または所望される特徴として説明されている特徴のいくつかは、いくつかのアプリケーションに対してだけ重要な特徴、主要な特徴、または所望される特徴であってよく、本発明のすべての実施形態に存在しないことがある。上述されたように、本発明の実施形態は、上述された１０のセキュリティ特徴の内のすべてまたは特定の数を満たすプロトコルに制限されない。

【０３６６】

特定のプロトコルが明細書中の表に示されてきたが、プロトコルに対する変形形態及び修正形態が期待される。特定の例として、認証サーバが入力としてＶを採るペアリングを実行する代わりに、クライアントは別個のサーバにこのペアリングをオフロードしてよい

。さらに、サーバはペアリング計算に対する入力としてＶを直接的に採るのではなく、Ｖに基づいてペアリング計算に対する入力を導出してよい。例えば、サーバはＶを入力として使用する前の値によってＶを乗算してよい。実装に応じて、双線型性を使用して、サーバはクライアントを依然として認証できるように、サーバは次いで同じ値のサーバシークレットに基づいてペアリングに対する入力を乗算する必要がある場合がある。

【０３６７】

例のいくつかでは、Ｒ－エイトペアリングが使用できることが説明されたが、任意のタイプの適切なペアリングが使用されてよいことが理解される。例えば、ウェイルペアリング、テートペアリング、及びエイトペアリングは、Ｒ－エイトペアリングの代わりに使用されてよい他のペアリングである。

【０３６８】

クライアントシークレットは、トークン及びＰＩＮに分割されるために説明されてきたが、本発明は二因子認証に制限されていない。上述されたように、クライアントシークレットは任意の数の因子に分割され、記憶されていない因子はＰＩＮに制限されない。ＰＩＮは、例えば、指紋、顔認識データ、または虹彩スキャン等のバイオメトリックで、またはソフトバイオメトリックで置換されてよい。ソフトバイオメトリックは、バイオメトリックを表すデータを削減するために、カテゴリに分類されたバイオメトリクスデータによって提供されてよい。例えば、すべての指紋は、特徴に基づいて、１００００のカテゴリに分類されてよく、ソフトバイオメトリクスデータは、ユーザーの指紋のカテゴリを示す。いくつかの例では、シークレットはトークン、ＰＩＮ、及びソフトバイオメトリックに分割されてよい。いくつかの例では、ＰＩＮ及び／またはトークンは、音声データもしくはユーザーのジオロケーションによって置換されてよい、またはシークレットの追加の因子は音声データもしくはユーザーのジオロケーションを含んでよい。

【０３６９】

クライアント及びサーバはインターネット等のデータネットワーク上で通信できると説明されてきたが、任意の通信媒体が期待される。例えば、本明細書に説明される例でのプロトコルは、無線通信技術であるが、無線ではない通信技術も使用して情報を送信するために使用されてよい。いくつかの特定の例として、プロトコルは、例えば３Ｇ、４Ｇ、ブルートゥース、赤外線通信、または光ファイバを使用して情報を送信するために使用されてよい。また、プロトコルは放送、デジタル記憶媒体、または印刷情報も含む通信方法で使用されてもよい。

【０３７０】

サーバがシークレットｓＱを発行され、ＱがG_2の固定生成元であることが説明されてきたが、Ｑは代わりに、例えば表５のプロトコルでの場合と同様に、サーバのアイデンティティから導出されてよい。

【０３７１】

さらに、クライアントがユーザーから受信されたＰＩＮをシークレットから抽出し、残っているトークンを記憶することが説明されてきたが、クライアントは、それがトークンから得られた値を抽出する前に、状況次第で受信されたＰＩＮに対してなんらかの処理を実行する。例えば、クライアントは特定の数を乗算する、またはＰＩＮに加算してよい。同様に、クライアントがＰＩＮ予想値を受信するとき、クライアントは、クライアントが元のＰＩＮで行ったのと同じＰＩＮ予想値の変形を実行し、次いで変形されたＰＩＮをシークレットを取り戻すために使用する。

【０３７２】

サーバが、鍵共有を実行する前に、サーバが認証を実行することが、Ｍ－Ｐｉｎ及びＭ－Ｐｉｎ　Ｆｕｌｌに関して説明されてきたが、クライアント及びサーバは鍵共有に直接的に進んでよい。例えば、係るプロトコルは、表１１に示されている。表１１では、認証は、鍵が確立された後に行われる。ただし、いくつかのアプリケーションでは認証は必要でないこともあれば、認証は代替プロセスを使用して実行された可能性がある。言い換えると、サーバはｇの値を計算せず、表８及び表１０のプロトコルでそれが１に等しいかど

うかをチェックしてよいが、代わりにＶに基づいてペアリングの結果からｒの値を計算し、クライアントがその鍵を導出するためにｒの値をクライアントに返す。結果的に、表８、表９、及び表１０で、ｇは計算されず、１に比較されるが、プロトコルの他のステップが同じままであってよい。

【０３７３】

さらに、信頼機関がシステムの重要な部分であることが説明されてきたが、セキュリティ要件がより低く、信頼機関が使用されないアプリケーションがあってよいことが理解されるだろう。

【０３７４】

参考文献による組込み

すべての特許、公開特許出願、及び本明細書に引用される他の参考文献の全体的なコンテンツは、参照することによりその全体でこれにより明示的に本明細書に組み込まれる。

【０３７５】

参考文献

参考文献リストＡ

1．ＩＥＥＥ Ｐ１３６３ホームページ。ｈｔｔｐ：／／ｇｒｏｕｐｅｒ．ｉｅｅｅ．ｏｒｇ／ｇｒｏｕｐｓ／１３６３／．

2．Ｐ．Ｓ．Ｌ．Ｍ．Ｂａｒｒｅｔｏ及びＭ．Ｎａｅｈｒｉｇ。Ｐａｉｒｉｎｇ－ｆｒｉｅｎｄｌｙ ｅｌｌｉｐｔｉｃ ｃｕｒｖｅｓ ｏｆｐｒｉｍｅ ｏｒｄｅｒ．Ｉｎ Ｓｅｌｅｃｔｅｄ Ａｒｅａｓ ｉｎ Ｃｒｙｐｔｏｌｏｇｙ － ＳＡＣ ２００５，ｖｏｌｕｍｅ ３８９７ ｏｆ Ｌｅｃｔｕｒｅ Ｎｏｔｅｓ ｉｎ Ｃｏｍｐｕｔｅｒ Ｓｃｉｅｎｃｅ，ｐａｇｅｓ ３１９－３３１．Ｓｐｒｉｎｇｅｒ－Ｖｅｒｌａｇ，２００６．

3．Ｓ．Ｂｌａｋｅ－Ｗｉｌｓｏｎ，Ｄ．Ｊｏｈｎｓｏｎ，ａｎｄ Ａ．Ｍｅｎｅｚｅｓ．Ｋｅｙ ａｇｒｅｅｍｅｎｔ ｐｒｏｔｏｃｏｌｓ ａｎｄ ｔｈｅｉｒ ｓｅｃｕｒｉｔｙ ａｎａｌｙｓｉｓ．Ｃｒｙｐｔｏｇｒａｐｈｙ ａｎｄ Ｃｏｄｉｎｇ，１３５５：３０－４５，１９９７．

4．Ｄ．Ｂｏｎｅｈ ａｎｄ Ｍ．Ｆｒａｎｋｌｉｎ．Ｉｄｅｎｔｉｔｙ－ｂａｓｅｄ ｅｎｃｒｙｐｔｉｏｎ ｆｒｏｍ ｔｈｅ Ｗｅｉｌ ｐａｉｒｉｎｇ．ＳＩＡＭ Ｊｏｕｒｎａｌ ｏｆ Ｃｏｍｐｕｔｉｎｇ，３２（３）：５８６－６１５，２００３．

5．Ｌ．Ｃｈｅｎ ａｎｄ Ｃ．Ｋｕｄｌａ．Ｉｄｅｎｔｉｔｙ ｂａｓｅｄ ｋｅｙ ａｇｒｅｅｍｅｎｔ ｐｒｏｔｏｃｏｌｓ ｆｒｏｍ ｐａｉｒｉｎｇｓ．Ｉｎ Ｐｒｏｃ．ｏｆ ｔｈｅ １６－ｔｈ ＩＥＥＥ Ｃｏｍｐｕｔｅｒ Ｓｅｃｕｒｉｔｙ Ｆｏｕｎｄａｔｉｏｎｓ Ｗｏｒｋｓｈｏｐ，ｐａｇｅｓ ２１９－２３３．ＩＥＥＥ Ｃｏｍｐｕｔｅｒ Ｓｏｃｉｅｔｙ，２００３．

6．Ｄ．Ｆｉｏｒｅ ａｎｄ Ｒ．Ｇｅｎｎａｒｏ．Ｍａｋｉｎｇ ｔｈｅ Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ ｐｒｏｔｏｃｏｌ ｉｄｅｎｔｉｔｙ－ｂａｓｅｄ．Ｉｎ Ｔｏｐｉｃｓ ｉｎ Ｃｒｙｐｔｏｌｏｇｙ － ＣＴ－ＲＳＡ ２０１０，ｖｏｌｕｍｅ ５９８５ ｏｆ Ｌｅｃｔｕｒｅ Ｎｏｔｅｓ ｉｎ Ｃｏｍｐｕｔｅｒ Ｓｃｉｅｎｃｅ，ｐａｇｅｓ １６５－１７８．Ｓｐｒｉｎｇｅｒ，２０１０．

7．Ｄ．Ｆｒｅｅｍａｎ，Ｍ．Ｓｃｏｔｔ，ａｎｄ Ｅ．Ｔｅｓｋｅ．Ａ ｔａｘｏｎｏｍｙ ｏｆ ｐａｉｒｉｎｇ ｆｒｉｅｎｄｌｙ ｅｌｌｉｐｔｉｃ ｃｕｒｖｅｓ．Ｊｏｕｒｎａｌ ｏｆ Ｃｒｙｐｔｏｇｒａｐｈｙ，２３：２２４－２８０，２０１０．

8．Ｌ．Ｆｕｅｎｔｅｓ－Ｃａｓｔａｎｅｄａ，Ｅ．Ｋｎａｐｐ，ａｎｄ Ｒ．Ｒｏｄｒｉｇｕｅｚ－Ｈｅｎｒｉｑｕｅｚ．Ｆａｓｔｅｒ ｈａｓｈｉｎｇ ｔｏ G_2．Ｉｎ Ｓｅｌｅｃｔｅｄ Ａｒｅａｓ ｉｎ Ｃｒｙｐｔｏｇｒａｐｈｙ － ＳＡＣ ２０１１，ｖｏｌｕｍｅ ７１１８ ｏｆ Ｌｅｃｔｕｒｅ Ｎｏｔｅｓ ｉｎ Ｃｏｍｐｕｔｅｒ Ｓｃｉｅｎｃｅ，ｐａｇｅｓ ４１２－４３０．Ｓｐｒｉｎ

ger−Verlag, 2011.
9. S. Galbraith, K. Paterson, and N. Smart. Pairings for cryptographers. Discrete Applied Mathematics, 156:3113−3121, 2008.
10. S. Galbraith and M. Scott. Exponentiation in pairing−friendly groups using homomorphisms. In Pairing 2008, volume 5209 of Lecture Notes in Computer Science, pages 211−224. Springer−Verlag, 2008.
11. R. P. Gallant, R. J. Lambert, and S. A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In Advances in Cryptology − Crypto 2001, volume 2139 of Lecture Notes in Computer Science, pages 190−200. Springer−Verlag, 2001.
12. F. Hao and D. Clarke. Security analysis of a multi−factor authenticated key exchange protocol. Cryptology ePrint Archive, Report 2012/039, 2012. http://eprint.iacr.org/2012/039.
13. H. S. Kim, S. W. Lee, and K. Y. Yoo. ID−based password authentication scheme using smart cards and fingerprints. ACM Operating Systems Review, 37(4):32−41, 2003.
14. I. Liao, C. Lee, and M. Hwang. A password authentication scheme over insecure networks. Journal of Computer and System Sciences, 72:727−740, 2006.
15. R. Martinez−Pelaez and F. Rico−Novella. Cryptanalysis of Sood at al.‘s authentication scheme using smart cards. Cryptology ePrint Archive, Report 2012/386, 2012. http://eprint.iacr.org/2012/386.
16. D. Pointcheval and S. Zimmer. Multi−factor authenticated key exchange. In ACNS‘08 Proceedings of the 6th international conference on Applied cryptography and network security, pages 277−295. Springer−Verlag, 2008.
17. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, 2000.
18. C. P. Schnorr. Efficient identification and signatures for smart cards. In Crypto‘89: Advances in Cryptology, volume 435 of Lecture Notes in Computer Science

, pages 239-252, 1989.
19. M. Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN number. Cryptology ePrint Archive, Report 2002/164, 2002. http://eprint.iacr.org/2002/164.
20. M. Scott. Cryptanalysis of an ID-based password authentication scheme using smart cards and fingerprints. Cryptology ePrint Archive, Report 2004/017, 2004. http://eprint.iacr.org/2004/017.
21. M. Scott. On the efficient implementation of pairing-based protocols. In Cryptography and Coding 2011, volume 7089 of Lecture Notes in Computer Science, pages 296-308. Springer-Verlag, 2011.
22. A. Shamir. Identity-based cryptosystems and signature schemes. In Advances in Cryptology: Proceedings of CRYPTO 84, volume 196 of Lecture Notes in Computer Science, pages 47-53, 1984.
23. S. Sood, A. Sarje, and K. Singh. An improvement of Liao at al's authentication scheme using smart cards. International Journal of Computer Applications, 1(8):16-23, 2010.
24. D. Stebila, P. Poornaprajna, and S. Chang. Multi-factor password-authenticated key exchange. In Australasian Information Security Conference, CPRIT volume 105, pages 56-66. Australian Computer Society, 2010.
25. C. Tsai, C. Lee, and M. Hwang. Password authentication schemes: Current status and key issues. International Journal of Network Security, 3(2):101-115, 2006.
26. D. Wang, C. Ma, and P. Wu. Secure password-based remote user authentication scheme with non-tamper resistant smart cards. Cryptology ePrint Archive, Report 2012/227, 2012.
http://eprint.iacr.org/2012/227.
27. Shengbao Wang, Zhenfu Cao, Zhaohui Cheng, and Kim-Kwang Raymond Choo. Perfect forward secure identity-based authenticated key agreement protocol in the escrow mode. Science in China Series F Information Sciences, 52(8):1358-1370, 2009.
28. Y. Wang. Efficient identity-based and authenticated key agreement protocol. Cr

yptology ePrint Archive, Report 2005／108, 2005. http：／／eprint.iacr.org／2005／108.
29. Y. Wang. Password protected smart card and memory stick authentication against off－line dictionary attacks. Cryptology ePrint Archive, Report 2012／120, 2012. http：／／eprint.iacr.org／2012／120.
30. T. Wu. The secure remote password protocol. In Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium, pages 97－111, 1998.
31. Guomin Yang, Duncan S. Wong, Huaxiong Wang, and Xiaotie Deng. Formal analysis and systematic construction of two－factor authentication scheme. In Proceedings of the 8th international conference on Information and Communications Security, ICICS‘06, pages 82－91. Springer－Verlag, 2006.
32. E. Yoon and K. Yoo. New authentication scheme based on a one－way hash function and Diffie－Hellman key exchange. In CANS‘05 Proceedings oN key exchange. In CANS’05 Proceedings of the 4th international conference on Cryptology and Network Security, volume 3810 of Lecture Notes in Computer Science, pages 147－160. Springer－Verlag, 2005.

【0376】

参考文献リストB

1. D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. Lopez. Faster explicit formulas for computing pairings over ordinary curves. Cryptology ePrint Archive, Report 2010／526, 2010. http：／／eprint.iacr.org／2010／526.
2. L. Ballard, M. Green, B. de Medeiros, and F. Montrose. Correlation－resistant storage via keyword－searchable encryption. Cryptology ePrint Archive, Report 2005／417, 2005. http：／／eprint.iacr.org／2005／417.
3. F. Bao, R. Deng, and H. Zhu. Variations of diffie－hellman problem. In ICICS 2003, volume 2836 of Lecture Notes in Computer Science, pages 301－312. Springer－Verlag, 2003.
4. P. S. L. M. Barreto and M. Naehrig. Pairing－friendly elliptic curves of prime ord

er. In Selected Areas in Cryptology – SAC 2005, volume 3897 of Lecture Notes in Computer Science, pages 319–331. Springer–Verlag, 2006.
5. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity–based identification and signature schemes. In Eurocrypt 2004, volume 3027 of Lecture Notes in Computer Science, pages 268–286. Springer–Verlag, 2004.
6. D. Boneh and M. Franklin. Identity–based encryption from the Weil pairing. SIAM Journal of Computing, 32(3):586–615, 2003.
7. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In Asiacrypt 2001, volume 2248 of Lecture Notes in Computer Science, pages 514–532. Springer–Verlag, 2001.
8. J. Cha and J. Cheon. An identity–based signature from gap diffie–hellman groups. In PKC 2003, volume 2567 of Lecture Notes in Computer Science, pages 18–30. Springer–Verlag, 2003.
9. B. Chevalier–Mames, J–S. Coron, N. McCullagh, D. Naccache, and M. Scott. Secure delegation of elliptic curve pairing. Cryptology ePrint Archive, Report 2005/150, 2005. http://eprint.iacr.org/2005/150.
10. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Crypto 1986, volume 263 of Lecture Notes in Computer Science, pages 186–194. Springer–Verlag, 1987.
11. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing friendly elliptic curves. Journal of Cryptography, 23:224–280, 2010.
12. S. Galbraith, K. Paterson, and N. Smart. Pairings for cryptographers. Discrete Applied Mathematics, 156:3113–3121, 2008.
13. K. Kurosawa and S–H. Heng. From digital signature to ID–based identification/signature. In PKC 2004, volume 2947 of Lecture Notes in Computer Science, pages 125–143. Springer–Verlag, 2004.
14. C. H. Lim and P. J. Lee. A key recove

ry attack on discrete log−based schemes using a prime order subgroup. In Crypto 1994, volume 1294 of Lecture Notes in Computer Science, pages 249−263. Springer−Verlag, 1994.
15. J. Pollard. Monte carlo methods for index computation mod p. Mathematics of Computation, 32, 1978.
16. M. Scott. Authenticated ID−based key exchange and remote log−in with simple token and PIN number. Cryptology ePrint Archive, Report 2002／164, 2002. http：／／eprint.iacr.org／2002／164.
17. M. Scott. Computing the tate pairing. In CT−RSA 2005, volume 3376 of Lecture Notes in Computer Science, pages 293−304. Springer−Verlag, 2005.
18. M. Scott. Replacing username／password with software−only two−factor authentication. Cryptology ePrint Archive, Report 2012／148, 2012. http：／／eprint.iacr.org／2012／148.
19. M. Scott and P. S. L. M. Barreto. Compressed pairings. Cryptology ePrint Archive, Report 2004／032, 2004. http：／／eprint.iacr.org／2004／032.
20. N. Smart and F. Vercauteren. On computable isomorphisms in efficient pairing−based systems. Discrete Applied Mathematics, 155：538−547, 2007.
21. M. Stam and A. K. Lenstra. Speeding up XTR. In Asiacrypt 2001, volume 2248 of Lecture Notes in Computer Science, pages 125−143. Springer−Verlag, 2001.
22. Y. Tseng and T. Tsai. Efficient revocable ID−based encryption with a public channel. The Computer Journal, 55（4）：475−486, 2012.
23. Y. Wang. Efficient identity−based and authenticated key agreement protocol. Cryptology ePrint Archive, Report 2005／108, 2005. http：／／eprint.iacr.org／2005／108.
24. X. Yi. An identity−based signature scheme from weil pairing. IEEE Communications Letters, 7：76−78, 2003.

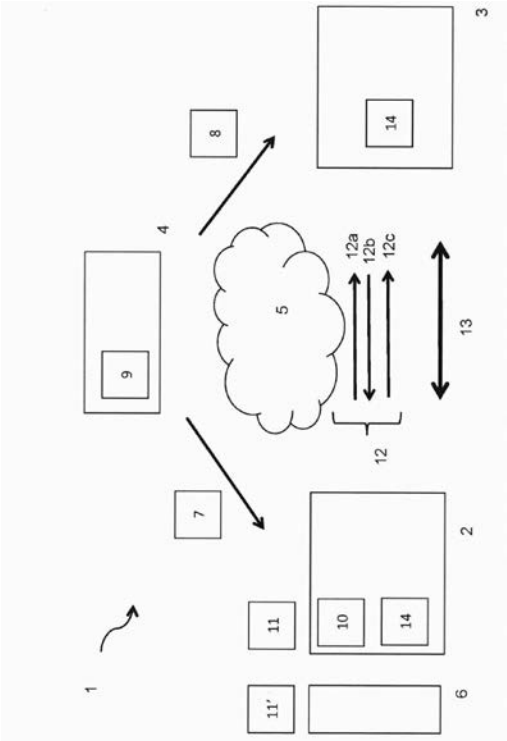
【0377】

参考文献リストC

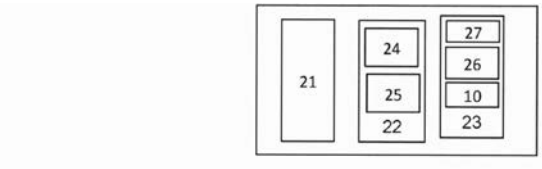
1. R. Gallant, R. Lambert, and S. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphism. I

n Crypto 2001, volume 2139 of Lecture Notes in Computer Science, pages 190－200. Springer－Verlag, 2001
2. R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003／054, 2003. http：／／eprint.iacr.org／2003／054

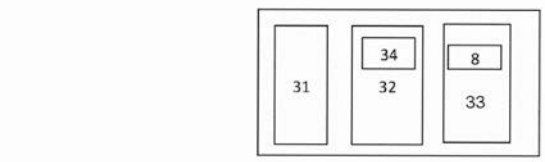
【図１】



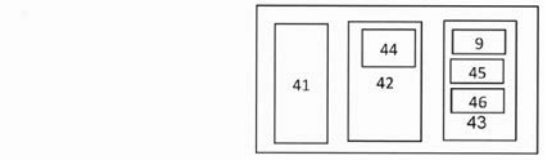
【図２】



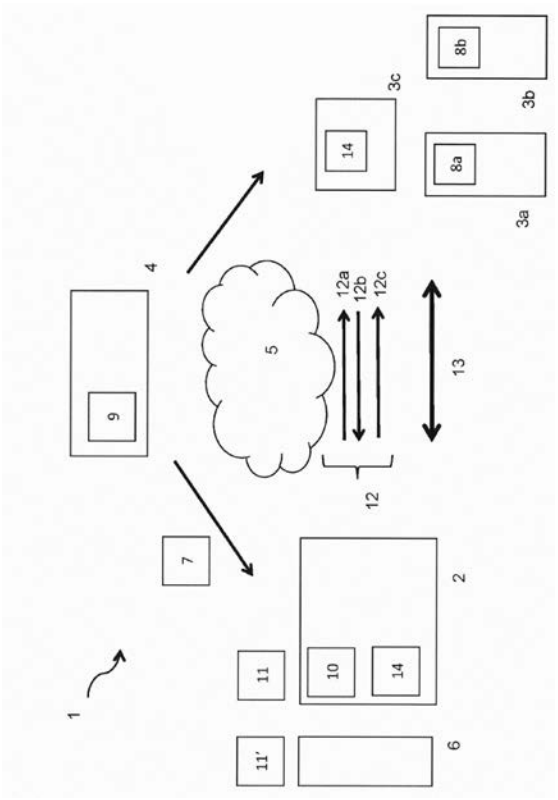
【図３】



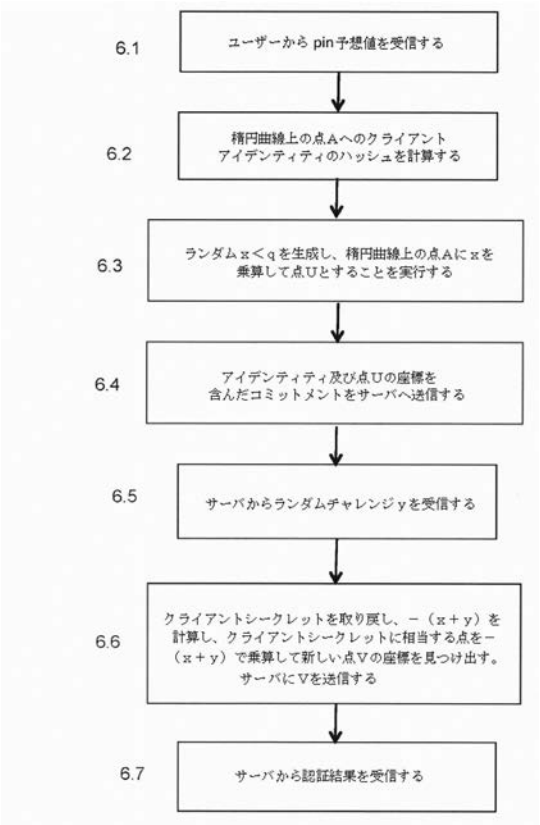
【図４】



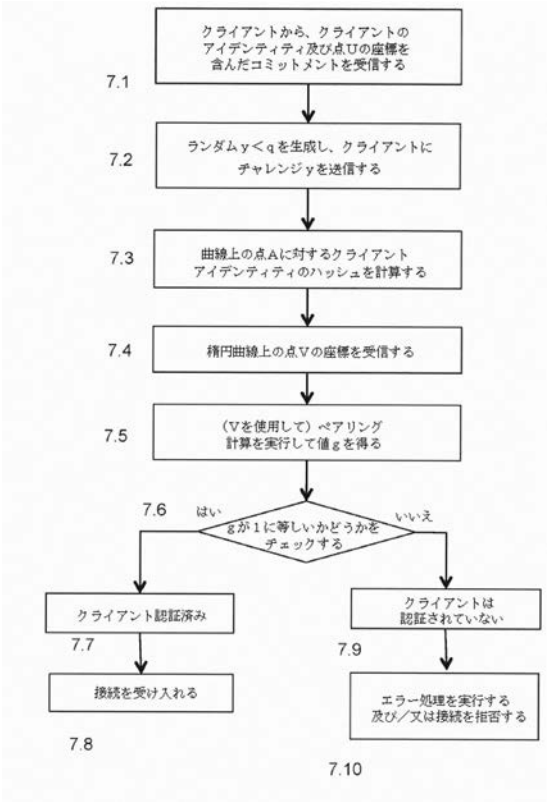
【図５】


8b
3c
3b
14
8a
3a
4
12a
12b
12c
5
13
9
12
7
2
11
10
14
1
11'
6

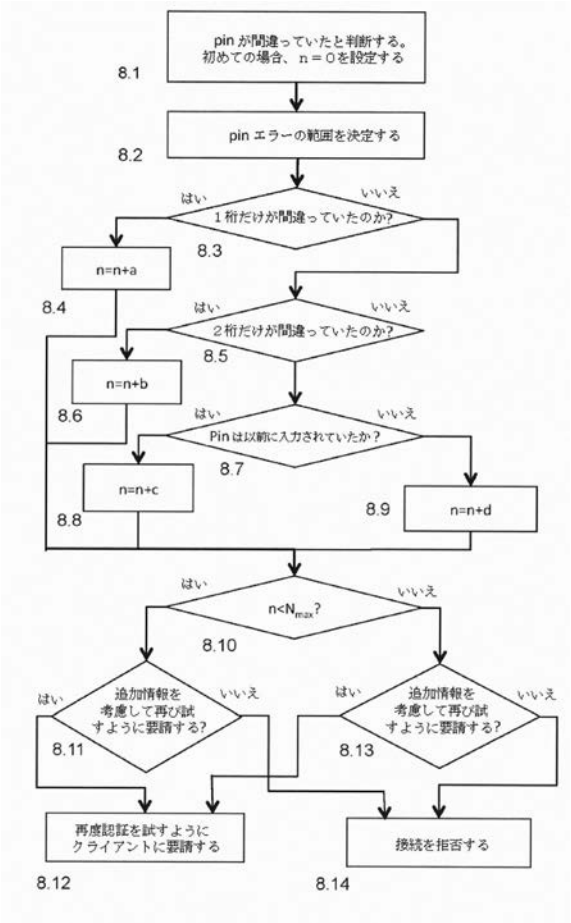
【図６】


6.1
ユーザーから pin 予想値を受信する
6.2
楕円曲線上の点Ａへのクライアント
アイデンティティのハッシュを計算する
6.3
ランダムｘ＜ｑを生成し、楕円曲線上の点Ａにｘを
乗算して点Ｕとすることを実行する
6.4
アイデンティティ及び点Ｕの座標を
含んだコミットメントをサーバへ送信する
6.5
サーバからランダムチャレンジｙを受信する
6.6
クライアントシークレットを取り戻し、－（ｘ＋ｙ）を
計算し、クライアントシークレットに相当する点を－
（ｘ＋ｙ）で乗算して新しい点Ｖの座標を見つけ出す。
サーバにＶを送信する
6.7
サーバから認証結果を受信する

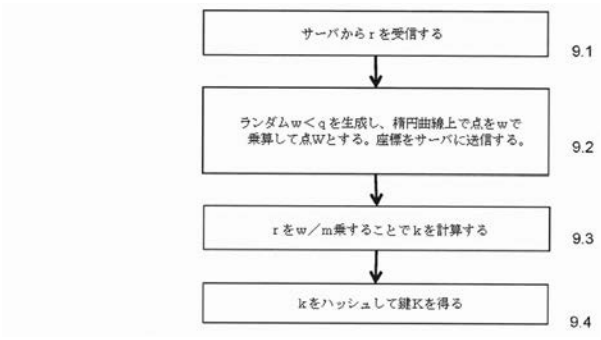
【図７】


7.1
クライアントから、クライアントの
アイデンティティ及び点Ｕの座標を
含んだコミットメントを受信する
7.2
ランダムｙ＜ｑを生成し、クライアントに
チャレンジｙを送信する
7.3
曲線上の点Ａに対するクライアント
アイデンティティのハッシュを計算する
7.4
楕円曲線上の点Ｖの座標を受信する
7.5
（Ｖを使用して）ペアリング
計算を実行して値ｇを得る
7.6
はい
ｇが１に等しいかどうかを
チェックする
いいえ
クライアント認証済み
7.7
接続を受け入れる
7.8
クライアントは
認証されていない
7.9
エラー処理を実行する
及び／又は接続を拒否する
7.10

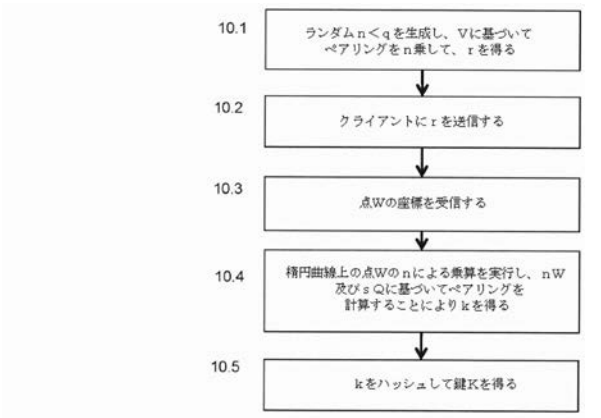
【図８】


8.1
pin が間違っていたと判断する。
初めての場合、ｎ＝０を設定する
8.2
pin エラーの範囲を決定する
はい
いいえ
１桁だけが間違っていたのか?
8.3
n=n+a
8.4
はい
いいえ
２桁だけが間違っていたのか?
8.5
n=n+b
8.6
はい
いいえ
Pin は以前に入力されていたか?
8.7
n=n+c
8.8
8.9
n=n+d
はい
いいえ
n<Nmax?
8.10
はい
追加情報を
考慮して再び試
すように要請する?
いいえ
8.11
はい
追加情報を
考慮して再び試
すように要請する?
いいえ
8.13
再度認証を試すように
クライアントに要請する
8.12
接続を拒否する
8.14

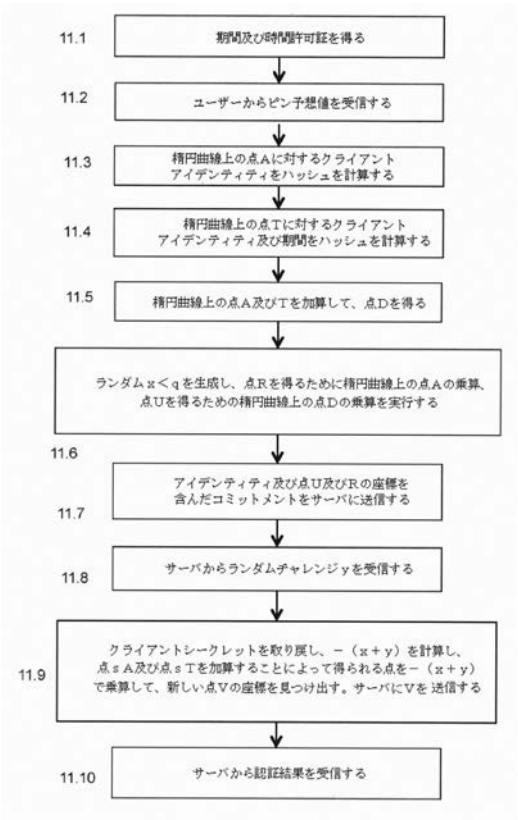
【図９】



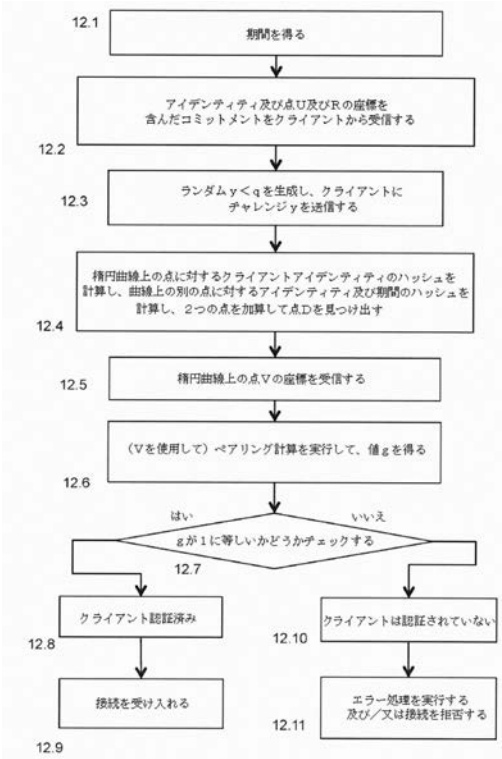
【図１０】



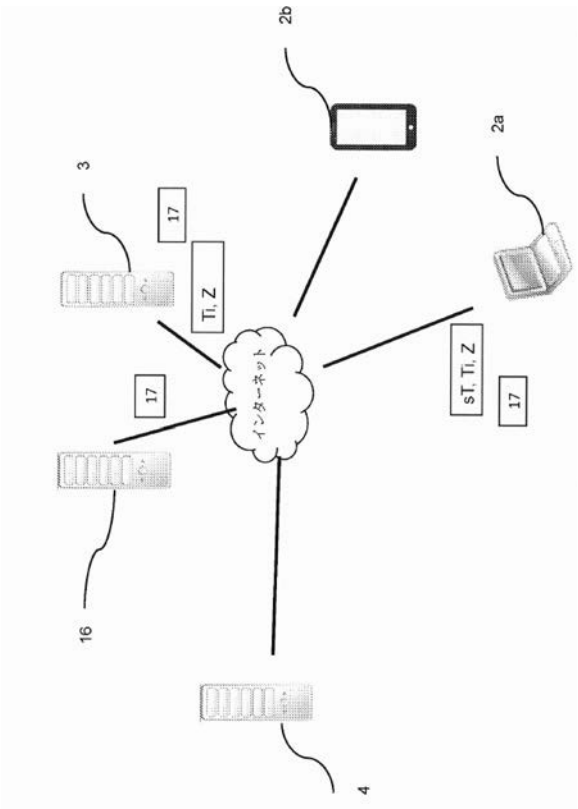
【図１１】



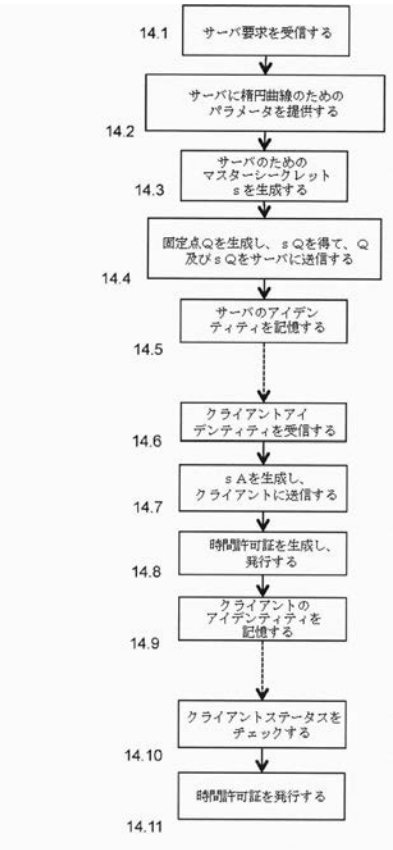
【図１２】



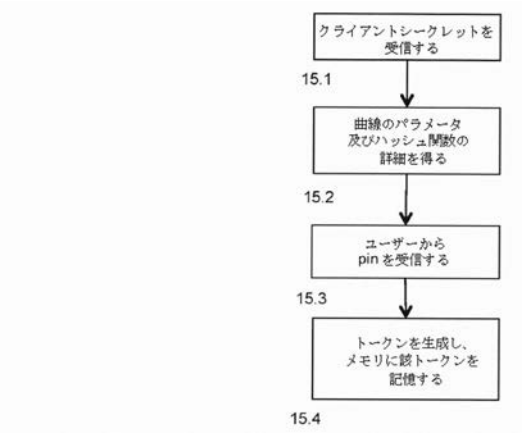
【図１３】



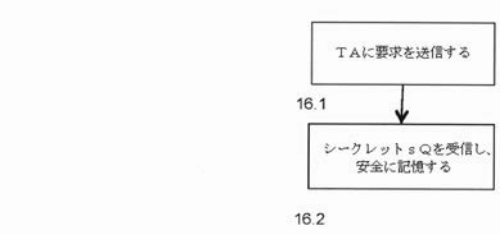
【図１４】



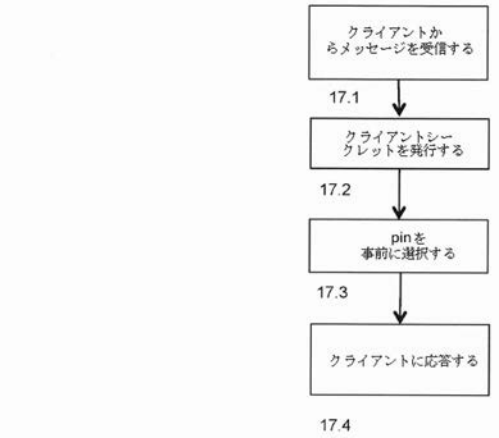
【図１５】



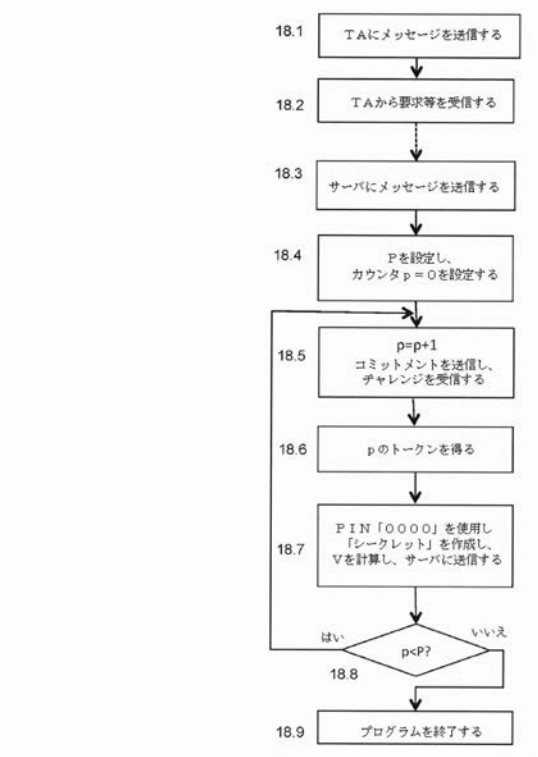
【図１６】



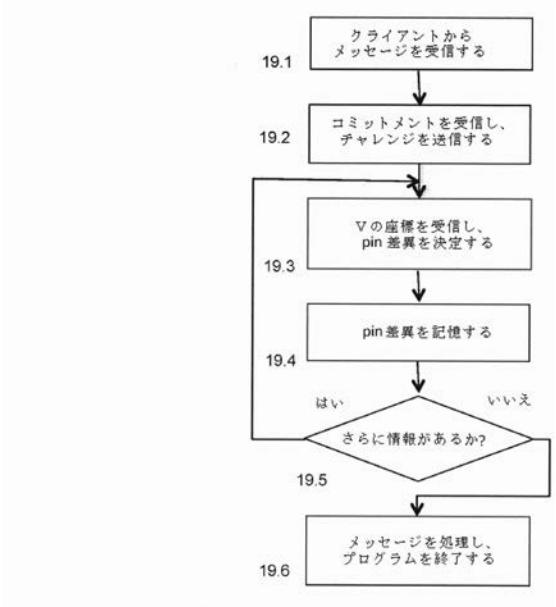
【図１７】



【図１８】



【図１９】


19.1
クライアントから
メッセージを受信する
19.2
コミットメントを受信し、
チャレンジを送信する
19.3
Vの座標を受信し、
pin差異を決定する
19.4
pin差異を記憶する
はい
いいえ
さらに情報があるか?
19.5
19.6
メッセージを処理し、
プログラムを終了する

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2014/051666

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/32
ADD.

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)
H04L

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)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Neyire Deniz Sarier: "Chapter 4: Practical Multi-factor Biometric Remote Authentication", Biometric Cryptosystems: Authentication, Encryption and Signature for Biometric Identities, Ph.D. Thesis, 8 April 2013 (2013-04-08), pages 110-134, XP055141965, Retrieved from the Internet: URL:http://hss.ulb.uni-bonn.de/2013/3181/3181.pdf [retrieved on 2014-09-23] the whole document -----	1-59, 67-77

☐ 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 but 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	Date of mailing of the international search report
23 September 2014	05/02/2015

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Di Felice, M

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB2014/051666

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-59, 67-77

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

International Application No. PCT/GB2014/051666

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-59, 67-77

Methods, apparatuses and computer programs for multi-factor zero-knowledge proof protocols.

---

2. claims: 60, 61

Computer-implemented method and corresponding medium for a pairing-based key agreement protocol using a Barreto-Naehrig elliptic curve.

---

3. claims: 62-64

Method, apparatus and computer program for multi-factor authentication wherein a secret is reconstructed from a plurality of factors, and a minimum combined error value over several attempts is evaluated to allow for an additional attempt even without success.

---

4. claims: 65, 66

Computer-implemented method and program of issuing a client and a server with secrets based on points on an algebraic curve.

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

（特許庁注：以下のものは登録商標）
１．ブルートゥース

(72)発明者 マッカーサー, キーラン
イギリス国 イーシー２エイ ３エイワイ ロンドン, リビングトン ストリート ８１ サーティボックス エルティーディー内
(72)発明者 スペクター, ブライアン
イギリス国 イーシー２エイ ３エイワイ ロンドン, リビングトン ストリート ８１ サーティボックス エルティーディー内
(72)発明者 スコット, ミハエル
イギリス国 イーシー２エイ ３エイワイ ロンドン, リビングトン ストリート ８１ サーティボックス エルティーディー内
Ｆターム(参考) 5J104 AA07 KA02 KA08 NA12 PA07

【要約の続き】
はサーバであってよい。信頼機関はクライアントにシークレットを発行した可能性があり、サーバが計算を実行して、クライアントがそのシークレットを所有しているかどうかを判断できるようにするために、サーバに別のシークレットを発行した可能性もある。
【選択図】図１