(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2005-535989
(P2005-535989A)
(43)公表日　平成17年11月24日(2005.11.24)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 15/00	G06F 15/00	330F	5B035
G06F 1/00	G06F 1/00	370E	5B058
G06K 17/00	G06K 17/00	ZECV	5B285
G06K 19/10	H04L 9/00	673D	5J104
H04L 9/32	G06K 19/00	S	

審査請求 有　予備審査請求 有　（全 41 頁）

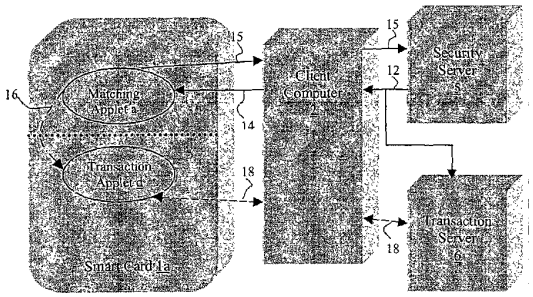
(21)出願番号　特願2004-530723 (P2004-530723)
(86)(22)出願日　平成14年8月8日 (2002.8.8)
(85)翻訳文提出日　平成17年4月8日 (2005.4.8)
(86)国際出願番号　PCT/SG2002/000199
(87)国際公開番号　WO2004/019190
(87)国際公開日　平成16年3月4日 (2004.3.4)
(81)指定国　AP (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), EA (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OA (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW

(71)出願人　502206429
ナンヤン　テクノロジカル　ユニヴァーシティ
NANYANG TECHNOLOGICAL UNIVERSITY
シンガポール共和国　シンガポール　６３７７２２　ナンヤン　ドライブ　１６　ユニット　２１３　ブロック　１　イノヴェーション　アンド　テクノロジー　トランスファー　オフィス
(74)代理人　100104765
弁理士　江上　達夫
(74)代理人　100107331
弁理士　中村　聡延

最終頁に続く

(54)【発明の名称】分散型認証処理

(57)【要約】
【解決手段】
処理能力の制限に伴う問題を解消する身元認証システム及び技術、並びにバイオメトリクス認証を処理するスマートカード技術が開示される。スマートカードとクライアント端末との間で身元認証処理のプロセスを分担させることにより、バイオメトリクス照合処理に伴う複雑な計算を実行して、スマートカード上に格納されたバイオメトリクス・パラメータを使用する認証を行うことができる。分散処理に係るシステム及び技術と共に、ユーザ認証に係るシステム及び技術が開示される。登録方法もまた開示される。



【特許請求の範囲】
【請求項１】
秘密部分と公開部分とに分けられるバイオメトリクス識別テンプレートを格納するユーザ提示装置上において、認証装置に提示されるユーザのバイオメトリクス・パラメータにより該ユーザを認証する方法であって、
前記認証装置に提示される前記ユーザのバイオメトリクス・パラメータから得られるデータをクライアント端末へ送る工程と、
ユーザ提示装置から前記クライアント端末へ、前記ユーザ提示装置に保持される前記バイオメトリクス識別テンプレートの公開部分のみを送る工程と、
前記クライアント端末において、前記データと前記公開部分との間で身元認証処理の第一ステージを実行し、前記身元認証処理の結果を前記ユーザ提示装置へ送る工程と、
前記ユーザ提示装置において、前記結果を使用して前記身元認証処理を完了する第二ステージを実行し、該実行に基づく認証結果を発行する工程と
を備えることを特徴とする方法。
【請求項２】
認証装置に提示されるユーザのバイオメトリクス・パラメータにより該ユーザを登録する方法であって、
前記認証装置において前記ユーザのバイオメトリクス・パラメータから得られるデータを、認証されたクライアント端末へ送る工程と、
前記認証されたクライアント端末において、算出されるバイオメトリクス識別テンプレートを、秘密部分と公開部分とに分割する工程と、
前記認証されたクライアント端末からユーザ提示装置へ、バイオメトリクス識別テンプレートの前記公開部分と前記秘密部分との両方を送る工程と、
前記公開部分と前記秘密部分とからなる前記バイオメトリクス識別テンプレートを、前記秘密部分が前記ユーザ提示装置内でのみアクセス可能であり且つ外部からはアクセス不能な状態で、前記ユーザ提示装置に格納する工程とを備えることを特徴とする方法。
【請求項３】
前記バイオメトリクス識別テンプレートの前記秘密部分は、詐称者の不正な改ざんによって該詐称者を本物のユーザとして誤認証させうるデータを含む部分であることを特徴とする請求項１又は２に記載の方法。
【請求項４】
前記バイオメトリクス識別テンプレートの前記公開部分は、詐称者の不正な改ざんによって該詐称者を本物のユーザとして誤認証させないデータを含む部分であることを特徴とする請求項１又は２に記載の方法。
【請求項５】
前記バイオメトリクス・パラメータは指紋であることを特徴とする請求項１、２又は３に記載の方法。
【請求項６】
前記バイオメトリクス識別テンプレートの前記公開部分は、当該バイオメトリクス識別テンプレートに固有な所定数の特徴からなるパラメータを含むことを特徴とする請求項１から５のいずれか一項に記載の方法。
【請求項７】
前記クライアント端末において実行される前記身元認証処理の前記第一ステージは、前記ユーザのバイオメトリクス・パラメータから得られるデータを使用して複数の固有の特徴を検出する工程と、該複数の固有の特徴を、前記ユーザ提示装置に保持される前記バイオメトリクス識別テンプレートからの前記所定数の固有の特徴と整合させる工程とを備えることを特徴とする請求項６に記載の方法。
【請求項８】
前記ユーザ提示装置において実行される前記身元認証処理の前記第二ステージは、当該ユーザ提示装置に格納されるローカル実行可能な照合プログラムを使用して実行されるこ

とを特徴とする請求項１から７のいずれか一項に記載の方法。
【請求項９】
前記クライアント端末において実行される前記身元認証処理の前記第一ステージは、クライアント実行可能な照合プログラムを使用して実行されることを特徴とする請求項１から８のいずれか一項に記載の方法。
【請求項１０】
前記クライアント実行可能な照合プログラムは、前記ユーザ提示装置又は前記認証装置に格納され、認証時に前記クライアント端末へ送られることを特徴とする請求項９に記載の方法。
【請求項１１】
前記クライアント実行可能な照合プログラムは、認証時に前記クライアント端末によってリモートメモリからダウンロードされることを特徴とする請求項９に記載の方法。
【請求項１２】
前記認証結果は、セキュアなトランザクションを許可するためのユーザの認証のために使用されることを特徴とする請求項１から１１のいずれか一項に記載の方法。
【請求項１３】
前記セキュアなトランザクションは、前記ユーザ提示装置に格納される、実行可能なトランザクションプログラムによって制御されることを特徴とする請求項１２に記載の方法。
【請求項１４】
前記認証結果が十分な適合性を示すとき、第一セキュリティアクセスチェックキーが前記認証結果を含んで作成されることを特徴とする請求項１から１３のいずれか一項に記載の方法。
【請求項１５】
第二セキュリティアクセスチェックキーが要求され、前記第一セキュリティアクセスチェックキーと比較され、該比較の結果が肯定的な認証結果であれば、該比較結果を使用して前記実行可能なトランザクションプログラムが有効にされることを特徴とする請求項１３及び１４に記載の方法。
【請求項１６】
前記第二セキュリティアクセスチェックキーはセキュリティサーバから発行されることを特徴とする請求項１５に記載の方法。
【請求項１７】
前記第一及び第二セキュリティアクセスチェックキーの夫々は、固有識別番号を含んでいることを特徴とする請求項１４から１６のいずれか一項に記載の方法。
【請求項１８】
前記固有識別番号は、無作為に作成される番号と前記認証結果とに基づく数学的演算により得られる番号を含んでいることを特徴とする請求項１５及び１７に記載の方法。
【請求項１９】
前記無作為に作成される番号は、該番号が使用される度に変化することを特徴とする請求項１８に記載の方法。
【請求項２０】
前記変化し且つ無作為に作成される番号は、現行の無作為に作成される番号として使用される第一部分と、次の無作為に作成される番号として使用される第二部分とからなる二つの部分に分割されることを特徴とする請求項１９に記載の方法。
【請求項２１】
前記固有識別番号は、ユーザにより記憶される番号を含むことを特徴とする請求項１７から２０に記載の方法。
【請求項２２】
前記固有識別番号に夫々組み込まれる２以上の認証方法を使用して前記認証結果を得ることができることを特徴とする請求項１８から２１に記載の方法。

【請求項２３】
前記アクセスが幾つかのレベルに分割され、ユーザに許可されるアクセスのレベルが、前記固有識別番号から得られる肯定的な識別の信頼度に依存することを特徴とする請求項１７から２２に記載の方法。
【請求項２４】
ユーザのバイオメトリクス・パラメータによって該ユーザを認証するシステムであって、当該システムは、
秘密部分と公開部分とに分割されるバイオメトリクス識別テンプレートが格納されるユーザ提示装置であって、前記公開部分のみが該ユーザ提示手段の外部へ送信可能なユーザ提示装置と、
ユーザから得られるバイオメトリクスデータを読取可能であって、前記ユーザ提示装置及びクライアント端末と通信する手段を備える認証装置と、
前記ユーザ提示装置に保持される前記バイオメトリクス識別テンプレートの前記公開部分と前記ユーザから得られるバイオメトリクスデータとを受け取るよう構成されたクライアント端末であって、前記データと前記公開部分との間で身元認証処理の第一ステージを実行し、該身元認証処理の結果を前記ユーザ提示装置へ送信可能なクライアントプロセッサを備えるクライアント端末とを備え、
前記ユーザ提示装置は、前記結果を使用して前記身元認証処理を完了し、該身元認証処理の完了に基づく認証結果を発行する第二ステージを実行可能なデバイスプロセッサを備えることを特徴とするシステム。
【請求項２５】
前記バイオメトリクス識別テンプレートの前記秘密部分は、不正な改ざんによって、詐称者を本物のユーザとして当該システムに誤認証させうるデータを含む部分であることを特徴とする請求項２４に記載のシステム。
【請求項２６】
前記バイオメトリクス識別テンプレートの前記公開部分が、不正な改ざんによって、詐称者を本物のユーザとして当該システムに誤認証させないデータを含む部分であることを特徴とする請求項２４に記載のシステム。
【請求項２７】
前記バイオメトリクス・パラメータは指紋であり、前記認証装置は指紋センサであることを特徴とする請求項２４に記載のシステム。
【請求項２８】
前記バイオメトリクス識別テンプレートの前記公開部分は、当該バイオメトリクス識別テンプレートに固有な所定数の特徴からなるパラメータを含むことを特徴とする請求項２４又は２７に記載のシステム。
【請求項２９】
前記ユーザ提示装置は、前記照合処理の前記第二ステージを実行するためのローカル実行可能な照合プログラムが格納されるメモリを備えていることを特徴とする請求項２４、２７又は２８に記載のシステム。
【請求項３０】
前記ユーザ提示装置の前記メモリは、前記照合処理の前記第一ステージを実行する前記クライアントプロセッサへ送られるクライアント実行可能な照合プログラムを格納することを特徴とする請求項２９に記載のシステム。
【請求項３１】
前記クライアント端末に接続されたセキュリティサーバを備えることを特徴とする請求項２４から３０のいずれか一項に記載のシステム。
【請求項３２】
前記セキュリティサーバは、前記照合処理の前記第一ステージを実行するためのクライアント実行可能な照合プログラムを保持することを特徴とする請求項３１に記載のシステム。

【請求項３３】
前記セキュリティサーバは、トランザクションを有効にするために前記クライアント端末から要求可能なセキュリティアクセスチェックキーを保持することを特徴とする請求項３１又は３２に記載のシステム。
【請求項３４】
セキュアなトランザクションを実行するよう構成されたトランザクションサーバであって、前記認証結果がセキュアなトランザクションを許可するためのユーザの認証に使用されるようにクライアント端末と通信するトランザクションサーバを備えることを特徴とする請求項２４から３３のいずれか一項に記載のシステム。
【請求項３５】
前記ユーザ提示装置は、前記セキュアなトランザクションを制御するための、実行可能なトランザクションプログラムを格納することを特徴とする請求項３４に記載のシステム。
【請求項３６】
２以上の認証方法を使用して前記認証結果を得ることができることを特徴とする請求項３４及び３５に記載のシステム。
【請求項３７】
前記トランザクションサーバへの前記アクセスが幾つかのレベルに分割され、ユーザに許可されるアクセスレベルが、前記各種認証方法を使用した結果に基づいて得られる肯定的な識別の信頼度に依存することを特徴とする請求項３４から３６に記載の方法。
【請求項３８】
第一及び第二プロセッサを使用して演算を実行する方法であって、
複数のプロセス名称を関連するプロセス識別子と共に含み、前記プロセス識別子夫々がプロセスロケータに関連付けられた第一のタスク表を第一プロセッサに格納する工程と、
前記複数のプロセス名称及び前記プロセス識別子を含む第二のタスク表を第二プロセッサに格納する工程と、
実行すべきプロセスを前記第二プロセッサにおいて識別し、前記プロセスを実行する前記第一プロセッサへ要求を発行する工程と、
前記プロセスロケータを使用して前記プロセスを特定し、前記第一プロセッサで前記プロセスを実行して結果を生成する工程と、
前記結果を前記第二プロセッサへ返す工程と
を備えることを特徴とする方法。
【請求項３９】
前記プロセス名称は、各オブジェクト識別子に関連付けられたオブジェクト名称を含むことを特徴とする請求項３８に記載の方法。
【請求項４０】
各オブジェクトは夫々に関連付けられた複数の関数を有し、各関数は、関数名称によって識別されると共に、前記第一及び第二のタスク表内において関数識別子に関連付けられていることを特徴とする請求項３９に記載の方法。
【請求項４１】
前記プロセスロケータは、プログラムメモリ内のプロセスの開始アドレスを識別することを特徴とする請求項３８、３９又は４０に記載の方法。
【請求項４２】
前記第二プロセッサは、前記第一プロセッサよりもかなり小さな処理能力を有する請求項３８から４１のいずれか一項に記載の方法。
【請求項４３】
前記第二プロセッサは、前記第一プロセッサにより実行される処理よりも小さな処理能力を要する処理をローカルで実行するよう構成されることを特徴とする請求項３８から４２のいずれか一項に記載の方法。
【請求項４４】

前記実行される演算は、前記第一プロセッサによって実行される基本特徴点検出処理と、前記第二プロセッサによって実行される特徴点照合処理とからなる指紋照合アルゴリズムであることを特徴とする請求項３８から４３のいずれか一項に記載の方法。

【請求項４５】

単一の第一プロセッサと通信する複数の第二プロセッサがあり、各第二プロセッサは夫々のタスク表を保持し、前記第一プロセッサは、前記第二プロセッサの前記タスク表によって識別される全てのプロセスを含む第一タスク表を保持することを特徴とする請求項３８から４４のいずれか一項に記載の方法。

【請求項４６】

前記第一プロセッサと前記第二プロセッサとがクライアントブリッジにより接続され、前記クライアントブリッジは、前記要求を前記第二プロセッサから前記第一プロセッサへ送信し、前記結果を前記第一プロセッサから前記第二プロセッサへ返信することを特徴とする請求項３８から４４のいずれか一項に記載の方法。

【請求項４７】

前記第一プロセッサはクライアント端末であり、前記第二プロセッサは携帯可能な機密処理用及びデータ格納用プラットフォームに埋め込まれたことを特徴とする請求項３８から４６のいずれか一項に記載の方法。

【請求項４８】

一又は複数の第二プロセッサにクライアントブリッジを介して接続する多数の第一プロセッサがあり、前記多数のプロセッサは、前記第二プロセッサの前記タスク表内における異なるプロセス小群を実行するよう構成されることを特徴とする請求項３８から４４のいずれか一項に記載の方法。

【請求項４９】

夫々が一のプロセスロケータに関連付けられた複数のプロセス名称及び複数のプロセス識別子を含む第一のタスク表を格納する第一プロセッサと、

前記プロセス名称を関連するプロセス識別子と共に含む第二のタスク表を格納する第二プロセッサとを備えてなり、

前記第二プロセッサは、実行すべきプロセスを特定し且つ前記プロセスを実行する前記第一プロセッサへ要求を発行する分散オブジェクト実行マネージャを含み、

前記第一プロセッサは、前記第一プロセッサにおける前記プロセスの実行を制御するクライアント分散オブジェクト実行マネージャを含み、前記第一プロセッサにおいて実行される前記プロセスの実行結果が前記第二プロセッサへ返信されることを特徴とする処理システム。

【請求項５０】

前記第一プロセッサは、前記第一プロセッサと前記第二プロセッサとの間の通信を行うクライアントマネージャを含むことを特徴とする請求項４９に記載の処理システム。

【請求項５１】

前記第一プロセッサは、プロセスの実行を行う実行マネージャを含むことを特徴とする請求項４８又は４９に記載のシステム。

【請求項５２】

前記第一プロセッサは、前記プロセスを保持するプログラムストアを備えており、前記プログラムストア内の前記プロセスの位置を識別するために前記プロセスロケータが使用されることを特徴とする請求項４９から５１のいずれか一項に記載のシステム。

【請求項５３】

前記第二プロセッサは、前記要求を前記第一プロセッサへ送信するリモート装置マネージャを含むことを特徴とする請求項４９から５２のいずれか一項に記載のシステム。

【請求項５４】

前記第二プロセッサが、前記第一プロセッサから返信される結果を保持するスタックを備えることを特徴とする請求項４９から５３のいずれか一項に記載のシステム。

【請求項５５】

前記第二プロセッサは、前記プロセスを保持するプログラムストアを含むことを特徴とする請求項４９から５４のいずれか一項に記載のシステム。
【請求項５６】
前記第一プロセッサはクライアント端末を備えることを特徴とする請求項４９から５５のいずれか一項に記載のシステム。
【請求項５７】
複数の第一プロセッサを備え、前記複数の第一プロセッサと前記第二プロセッサとの間の通信を行うクライアントブリッジを更に備えることを特徴とする請求項３８から５６のいずれか一項に記載のシステム。
【請求項５８】
各第一プロセッサはサーバを備えることを特徴とする請求項５７に記載のシステム。
【請求項５９】
前記クライアントブリッジは、前記要求に含まれるプロセッサ識別子に基づいて、前記第二プロセッサから前記複数の第一プロセッサのうち適切な一の第一プロセッサに前記要求を送信するネットワーク実行マネージャを含むことを特徴とする請求項５７又は５８に記載のシステム。
【請求項６０】
複数の第二プロセッサと、
前記複数の第二プロセッサと前記第一プロセッサとを接続するクライアントブリッジとを備えることを特徴とする請求項３８から５６のいずれか一項に記載のシステム。
【請求項６１】
第二プロセッサ各々が、スマートカードなどの携帯可能な機密処理用及びデータ格納用プラットフォームへ夫々埋め込まれたことを特徴とする請求項３８から６０のいずれか一項に記載のシステム。
【発明の詳細な説明】
【背景技術】
【０００１】
電子取引若しくは他の遠隔取引において、本物のユーザを識別することは、実際の接触がないため困難である。例えば、誰かがインターネット上で何かを購入しようとすると、大抵の場合クレジットカード番号が要求されるが、認証用カードを物理的に提示することまでは要しない。しかしながら、そのような番号はハッカーによって相当容易に入手可能である。また多数の偽造クレジットカードが出回っていることはいうまでもない。銀行カードは、ユーザの身元を認証するためにＰＩＮ（Personal Identification Number）番号を必要とする。しかしながら、多くの状況において、ＰＩＮ番号は容易に入手可能である。それは、ユーザがＰＩＮ番号をカード自体やカードにまつわる場所に書き込んだり、あるいは不正手段によってＰＩＮ番号が入手されたりするからである。また複数の銀行口座を持っている顧客もおり、そのような場合、日常使用する各種ＰＩＮやパスワードに加えて種々のＰＩＮ番号全てを覚えていることは困難である。ｍコマース（モバイルコマース）又はeコマース（電子商取引）において最も重大な問題は、複雑で煩わしい認証機構に頼ることなく、如何にして「本物の」顧客を識別するかということにある。バイオメトリクスは、この問題を解決する最適な方法の一つである。
【０００２】
スマートカードは、小型、携帯性、機密処理プラットフォームを含む点などから、今日のｅコマーストランザクション（電子商取引）にとって一般的になってきている。スマートカードを使用する従来方法は、カード上でユーザのＰＩＮ番号を格納及び認証しているだけであった。例えば、ＧＳＭ携帯電話で使用されるスマートカードの一種であるＳＩＭ（Subscriber Identity Module）カードは、ユーザのＰＩＮ番号や移動局へのアクセスセキュリティコードを格納するために使用される。この場合もやはりＰＩＮ番号の認証が行われる。
【０００３】

ＰＩＮ番号の代わりに、あるいはＰＩＮ番号に加えて、ユーザのバイオメトリクスデータを使用して認証を行うバイオメトリクス認証を処理するには、スマートカード技術を使用することが望ましい。スマートカードを使用する際の問題は、その処理能力とカード上の使用可能メモリが制限されている点にある。そのことにより、バイオメトリクスデータを使用する認証技術は往々にして拡張処理能力を必要とする。ＵＳＢ（Universal Serial Bus）トークン、非接触カード、マルチメディアカード、メモリスティック、セキュアプロセッサチップ及びスマートウォッチを含む携帯可能な機密処理及びデータ格納用プラットフォーム全般において、同様の状況に直面する。

【０００４】

特許文献１には、多数の認証確認ゾーンを有するスマートカードが記載されている。これらゾーンはカウンタ及び証明書によって形成され、バイオメトリクステンプレートには関与していない。

【０００５】

特許文献２には、金融取引システム用のスマートカードが記載されている。このスマートカードに格納されている記録情報へのアクセスは、３レベル以上の階層に分割される。このシステムは、認証用にバイオメトリクス及びＰＩＮを組み込むことが可能であるが、これらはスマートカードリーダレベルで実行されている。

【０００６】

特許文献３には、スマートカード用に複数のインタラクションコンテクストを有するコヒーレントデータ構造の発明が記載されている。ここで紹介された技術は、スマートカード内での独立された実行環境において好適であり、リソースを共有すると共に、多数のアプリケーションに共有されたデータに複数のアクセス条件を定義する方法が紹介されている。しかしながら、スマートカードの処理能力によりこの操作は制限され、外部の追加リソースを得ることもしていない。

【０００７】

特許文献４には、端末に接続されたスマートカードが記載され、この端末はホストコンピュータ及び／又はネットワークに接続されている。スマートカードは端末と通信を開始するよう構成され、そのことによりスマートカードは端末、ホストコンピュータ又はネットワークを制御できるようになり、端末、ホストコンピュータ又はネットワークに接続されたリソースにアクセスできるようになる。通信プロトコルは、スマートカードが送ることのできるコマンドを定義し、非同期又は論理的非同期通信による通信をスマートカードに許可する。通信速度が遅いので、多数のコマンドを含む計算集約的タスクを実行するにはこの方法は不適当である。

【０００８】

特許文献５には、クレジットカードの形状を持つスマートカードを使用して指紋インプリントをライブスキャン装置へ送る指紋認証方法が記載されている。スマートカードは、主に指紋を格納するために使用されており、その処理側面については殆ど記載されていない。

【０００９】

特許文献６には、スマートカードを使用してインターネット端末においてユーザセッションを開始する方法及びシステムが記載されている。インターネット端末は、サーバシステムに連結している。インターネット端末は、固有のスマートカード識別子を格納しているスマートカードの存在を検知し、この識別子を使用してサーバの特定ユーザに関連する構成情報を検出或いは検索する。構成情報は、例えば、インターネット端末によって提示されるオンスクリーンキーボードの種類、テキストを表示するためインターネット端末によって使用されるフォント、ＢＧＭオプション、電子メールオプションなどインターネット端末の動作をカスタマイズする、定義された顧客環境又は顧客嗜好を含んでいる。スマートカードセキュリティや身元認証には関与していない。

【００１０】

特許文献７には、ネットワーク上でユーザを認証する方法及び装置が記載されている。

このネットワークは、クライアントコンピュータ及びサーバコンピュータを備えており、このクライアントコンピュータはスマートカード及びスマートカードリーダを備えている。クライアントは、サーバに格納された制限情報にアクセスする要求をサーバへ送る。サーバは、スマートカードインタフェースモジュールをクライアントへ送る。サーバは、スマートカードにアクセスするユーザからのアクセスコードを要求する。サーバがアクセスコードを受け取ると、サーバはプログラム及びアクセスコードを使用して、スマートカードに格納されたユーザ情報にアクセスする。サーバは、ユーザ情報を、サーバだけが使用可能でクライアントが使用不能な認証情報と比較する。ユーザ情報が認証情報と合致すると、サーバは、クライアントが制限情報にアクセスすることを許可する。アクセスコードは、バイオメトリクスの一部でもなく、バイオメトリクス情報を処理することも含んでいない。

【００１１】

【特許文献１】米国特許第５，７６７，５０４号明細書
【特許文献２】米国特許第６，０１２，０４９号明細書
【特許文献３】米国特許第６，０５２，６９０号明細書
【特許文献４】米国特許第６，１５７，９６６号明細書
【特許文献５】米国特許第６，１８２，８９２号明細書
【特許文献６】米国特許第６，１９９，１１４号明細書
【特許文献７】米国特許第６，２２６，７４４号明細書

【発明の開示】

【発明が解決しようとする課題】

【００１２】

本発明の一態様における目的は、バイオメトリクスを使用し、更に、スマートカードのようなユーザ提示装置における処理によって身元認証の安全度を高めることにある。

【００１３】

本発明の他の一態様における目的は、バイオメトリクス認証を行うスマートカードの小さな処理能力に伴う問題を解消することにある。

【００１４】

本発明の更に他の一態様における目的は、使用する個々のプロセッサの処理能力を有効に活用するよう、複数のプロセッサにおけるタスク分散処理技術を提示することにある。

【課題を解決するための手段】

【００１５】

本発明の一態様によれば、秘密部分と公開部分とに分けられるバイオメトリクス識別テンプレートを格納するユーザ提示装置上の認証装置に提示されるユーザのバイオメトリクス・パラメータにより該ユーザを認証する方法であって、前記認証装置に提示される前記ユーザのバイオメトリクス・パラメータから得られるデータをクライアント端末へ送る工程と、ユーザ提示装置から前記クライアント端末へ、前記ユーザ提示装置に保持される前記バイオメトリクス識別テンプレートの公開部分のみを送る工程と、

前記クライアント端末において、前記データと前記公開部分との間で身元認証処理の第一ステージを実行し、前記身元認証処理の結果を前記ユーザ提示装置へ送る工程と、前記ユーザ提示装置において、前記結果を使用して前記身元認証処理を完了する第二ステージを実行し、該実行に基づく認証結果を発行する工程とを備える方法が提示される。

【００１６】

第一ステージの結果は、直接又は認証装置を介して、ユーザ提示装置へ送られる。

【００１７】

本発明の他の一態様は、ユーザのバイオメトリクス・パラメータによって該ユーザを認証するシステムであって、当該システムは、秘密部分と公開部分とに分割されるバイオメトリクス識別テンプレートが格納されるユーザ提示装置であって、前記公開部分のみが該ユーザ提示手段外部へ送信可能なユーザ提示装置と、ユーザから得られるバイオメトリクスデータを読取可能であって、前記ユーザ提示装置及びクライアント端末と通信する手段

を備える認証装置と、前記ユーザ提示装置に保持された前記バイオメトリクス識別テンプレートの前記公開部分と前記ユーザから得られるバイオメトリクスデータとを受け取るよう構成されたクライアント端末であって、前記データと前記部分との間で身元認証処理の第一ステージを実行し、該身元認証処理の結果を前記ユーザ提示装置へ送信可能なクライアントプロセッサを備えるクライアント端末とを備え、前記ユーザ提示装置は、前記結果を使用して前記身元認証処理を完了し、該身元認証処理の完了に基づく認証結果を発行する第二ステージを実行可能なデバイスプロセッサを備えるシステムを提供する。

【００１８】

認証装置及びユーザ提示装置は、物理的に分離した装置としてもよいし、物理的に単一の装置としてもよい。物理的な単一の装置である場合には、ユーザ提示装置のみが高セキュリティ処理可能な単一装置における処理部分又は内部特別部分であり、認証装置を形成する一般的処理装置とは分離されている。例えば、ASICチップとすることができるが、その場合、ユーザ提示装置がセキュアプロセッサモジュールであり、チップの残りの部分が認証装置である。

【００１９】

認証装置が十分な処理リソース及びメモリリソースを有していれば、クライアント端末及び認証装置もまた別々の装置であってもよいし、物理的に単一の装置であってもよい。後者の場合、クライアント端末と認証装置とは区別されない。

【００２０】

本発明の他の一態様は、認証装置に提示されるユーザのバイオメトリクス・パラメータにより該ユーザを登録する方法であって、前記認証装置において前記ユーザのバイオメトリクス・パラメータから得られるデータを、認証されたクライアント端末へ送る工程と、前記認証されたクライアント端末において、算出されるバイオメトリクス識別テンプレートを、秘密部分と公開部分とに分割する工程と、前記認証されたクライアント端末からユーザ提示装置へ、バイオメトリクス識別テンプレートの前記公開部分と前記秘密部分との両方を送る工程と、前記公開部分と前記秘密部分とからなる前記バイオメトリクス識別テンプレートを、前記秘密部分が前記ユーザ提示装置内でのみアクセス可能であり且つ外部からはアクセス不能な状態で、前記ユーザ提示装置に格納する工程とを備える方法を提供する。

【００２１】

ここで説明する具体的なバイオメトリクス・パラメータは指紋であるが、本発明は任意の好適なバイオメトリクス・パラメータにも適用可能であることは当然である。加えて、ここで説明する具体的ユーザ提示装置はスマートカードであるが、本発明が任意の一般的な携帯型機密処理データ格納プラットフォームに適用可能であることは当然である。

【００２２】

本発明の他の一態様は、第一及び第二プロセッサを使用して演算を実行する方法であって、複数のプロセス名称を関連するプロセス識別子と共に含み、前記プロセス識別子夫々がプロセスロケータに関連付けられた第一のタスク表を第一プロセッサに格納する工程と、前記複数のプロセス名称及び前記プロセス識別子を含む第二のタスク表を第二プロセッサに格納する工程と、実行すべきプロセスを前記第二プロセッサにおいて特定し、前記プロセスを実行する前記第一プロセッサへ要求を発行する工程と、前記プロセスロケータを使用して前記プロセスを検出し、前記第一プロセッサで前記プロセスを実行して結果を生成する工程と、前記結果を前記第二プロセッサへ返す工程とを備える方法を提供する。

【００２３】

本発明の他の一態様は、複数のプロセス名称及び夫々が一のプロセスロケータに関連付けられた複数のプロセス識別子を含む第一のタスク表を格納する第一プロセッサと、前記プロセス名称を関連するプロセス識別子と共に含む第二のタスク表を格納する第二プロセッサとを備えてなり、前記第二プロセッサは、実行すべきプロセスを特定し且つ前記プロセスを実行する前記第一プロセッサへ要求を発行する分散オブジェクト実行マネージャを含み、前記第一プロセッサは、前記第一プロセッサにおける前記プロセスの実行を制御す

るクライアント分散オブジェクト実行マネージャを含み、前記第一プロセッサにおいて実行される前記プロセスの実行結果が前記第二プロセッサへ返信される処理システムを提供する。

【００２４】

本発明は、第二プロセッサが第一プロセッサに比べ処理能力がかなり小さい場合や、第二プロセッサが他のアプリケーションの用途で第二プロセッサ自身の処理能力を使用しており且つ第一プロセッサが安全で信頼できる場合に有用である。

【００２５】

本発明の上記態様を具体的に適用すると、指紋認証処理をクライアント端末とスマートカードとに分散させることができる。

【００２６】

このように、以下に説明する本発明の実施形態は、バイオメトリクス認証を行うスマートカード技術における処理能力の制限に伴う問題を解消する。現在、スマートカードは、制限を有するＲＡＭ（2ｋバイト未満）、EEPROM（64ｋバイト未満）及び処理能力を有している。指紋照合処理の一部に関連する複雑な計算を行うためのコプロセッサとしてＰＣを使用することによって、スマートカードに格納されたバイオメトリクス・パラメータを使用する認証が可能になる。以下、電子商取引（ｅコマーストランザクション）又はモバイル商取引（ｍコマーストランザクション）を保護するセキュアな認証プロトコルについて説明する。全スキームは、セキュアな認証トランザクションを実行する任意の携帯可能な計算装置に適用可能である。

【００２７】

上記実施形態においては、スマートカードに格納された完全な指紋テンプレート（又は他のバイオメトリクステンプレート）がスマートカードから送信されることがないので、システムのセキュリティは向上する。好適な実施形態において、ユーザの身元は、バイオメトリクス照合の結果を取り入れた固有ＰＩＮ番号の形態で外部世界に伝達される。

【００２８】

特に、例として掲げる好適な実施形態の具体的説明において、以下の要素を更に詳細に説明する。

【００２９】

＜プロトコル＞

指紋テンプレートの一部をスマートカードからクライアント端末へ送る機密保護指紋照合プロトコルが開発された。このプロトコルによれば、個人の指紋テンプレートが漏洩するセキュリティ上のリスクともなり得るスマートカードからＰＣへのテンプレート全体の送信が不要となる。スマートカード内の指紋テンプレートは、秘密部分と公開部分とに分けられる。秘密部分はどのような状況でも常にスマートカード内に留められる。指紋テンプレートの公開部分のみがスマートカードからクライアント端末へ送信される。この送信されたテンプレートの公開部分だけからはスマートカードへのアクセス権を得る偽造テンプレートを作成することはできない。

【００３０】

また、トランザクションサーバへ進入し不正アクセスを試みようとする侵入者を排除する機密保護指紋スマートカード有効トランザクションプロトコルを説明する。全てのトランザクションがスマートカード上に格納された指紋有効トランザクションアプレットの実行を必要とするので、スマートカードは、トランザクション用の物理的キーとなる。

【００３１】

＜負荷分担＞

スマートカードのプロセッサとクライアント端末のプロセッサとで指紋認証処理を分担することによって、スマートカードへの負荷が軽減される。従って、高い認証精度でカード上の照合を行うために、処理能力の大きな高価なリソースを有するスマートカードを使用する必要がない。また、クライアント端末が信頼でき安全であることを保証することができ、指紋認証処理を実行するコードが安全であることを保証することができる。

【００３２】
＜ユーザ識別番号＞
使用の度に変更することができる固有ユーザ識別番号が提供される。ユーザはその識別番号を覚えておく必要がない。代わりに、発行会社のシステムとユーザのスマートカードのみが、ユーザ識別番号の写しを保存している。ユーザは、トランザクションへのアクセス権付与を決定するユーザ識別番号を合成するために、もう一つの個人識別番号やバイオメトリクスデータに依存している。
【００３３】
＜分散型リモート実行マネージャプロトコル＞
このプロトコルは、演算の並行処理を、第一及び第二プロセッサ、とりわけスマートカード及びクライアント端末に実行させることを可能にする。これにより、処理能力が小さいがゆえに、指紋照合のような計算集約技術をスマートカード上で実行するには時間がかかりすぎるというスマートカード又は他のモバイル装置に伴う問題が解消される。処理能力の大きなクライアント端末に負荷を分担させることによって処理速度が向上する。このプロトコルはスマートカード上のみならず、ＰＤＡ（Personal Digital Assistants）を含む処理能力の小さな任意の装置において実行可能である。
【００３４】
本発明を更に理解し、本発明がどのように実施されるかを示すために、以下、添付の図面を例として参照しつつ説明を続ける。
【発明を実施するための最良の形態】
【００３５】
以下に、モバイル型機密保護電子商取引の方式を説明する。図１は、一態様における各構成要素の繋がりを示している。この方式は、スマートカードリーダ１、ローカルクライアントコンピュータ２、ネットワーク接続４、セキュリティサーバ５及びトランザクションサーバコンピュータ６からなる。ここで処理方法の説明の前に、関連する要素及び使用する用語を全て定義しておく。スマートカードリーダ１は、スマートカード１ａとローカルクライアントコンピュータ２との間の通信のためのハードウェアデバイスである。ローカルクライアントコンピュータ２は、処理を行う（プロセスを実行する）サーバ６にユーザが接続するために用いられる端末である。端末２としては、通信装置を組み入れたPC、PDA又は携帯型コンピュータを挙げることができる。ネットワーク接続４は、クライアント２とサーバ６との間の通信方法であり、例えばＬＡＮ（ローカルエリアネットワーク）又はＷＡＮ（広域ネットワーク）４を基調とした通信を提供する有線又は無線接続である接続リンク１を含んでいる。無線接続としては、ＧＳＭ、IEEE802.11ｂに準拠する無線ＬＡＮ、ブルートゥース又はＩｒＤＡ赤外線データ通信規格いずれかを使用するものを挙げることができる。有線接続としては、イーサネット、RS-232C、ＩＢＭトークンリング等を挙げることができる。セキュリティサーバ５は、トランザクションサーバ６を侵入者から保護すると共に、ユーザのポートフォリオを管理する。トランザクションサーバ６は、実際の商取引全体を行うサーバである。トランザクションサーバ６の位置は、セキュリティサーバファイアウォール７によって保護されたセキュリティサーバ５の後方とするか、ローカルファイアウォール７及び８によって保護された独立のサーバ６とすることができる。どのようにしてファイアウォールを有効にするかについての詳細は後述する。スマートカードリーダ１は、例えば標準的なRS-232C又はＵＳＢ接続によりローカルクライアントコンピュータ２に接続される。この接続自体は、データ暗号化機能を持たない。スマートカード１ａ及びクライアントコンピュータ２が情報を秘密にするための暗号化を行う。本実施形態におけるスマートカード１ａは、Javaバイトコードを実行しうるスマートカードの一種であるJavaカードからなる。図１Ａでは、スマートカード１ａは、以下の情報、即ち（ａ）ローカル照合アプレット、（ｂ）クライアント照合アプレット、（ｃ）セキュリティサーバのＵＲＬ（Uniform Resource Locator）、（ｄ）トランザクションアプレット及び（ｅ）ＳＡＣＫを比較するセキュリティアプレットを格納するメモリ１０を有している。

【００３６】
またメモリ１０は、スマートカードの正規ユーザの指紋認証テンプレート１１を保持している。またスマートカード１ａは、スマートカード１ａ上でコードを実行するプロセッサ１３を有している。
【００３７】
ローカル照合アプレット(ａ)は、スマートカード１ａ上で実行可能なJavaバイトコードのような、実行可能な小プログラムである。クライアント照合アプレット（ｂ）は、クライアントコンピュータ２において実行可能なJavaバイトコードのような、別の実行可能な相補プログラムである。クライアント照合アプレット（ｂ）は、スマートカードのメモリ１０に格納することができる。しかしながら、メモリ制限のあるスマートカードにおいては、クライアント照合コードのサイズが大きすぎる場合がある。そこで、クライアント照合アプレット用のコードを得ることができるセキュリティサーバのＵＲＬ（ｃ）が採用される。ＵＲＬ（ｃ）は、クライアントがインターネットにアクセスして、サーバと接続しコードをダウンロードする際に必要である。クライアントがどこでクライアント照合コードをダウンロードしようとも、クライアントはそのJavaバイトコードの完全性を常に確認すべきである。トランザクションアプレット（ｄ）は、実際の商取引を実行する。このトランザクションアプレットを有効にする唯一の方法は、ローカル照合アプレットによるものである。後で更に詳細に説明するように、ユーザの指紋が有効であることをローカル照合アプレットが確認すると、該ローカル照合アプレットはセキュリティ有効コードをスマートカードの内部ファイアウォールへ送信し、トランザクションサーバ６に商取引を行わせる。トランザクションサーバ６が如何にして商取引を実行するかについての詳細は後述する。
【００３８】
指紋センサ３は、指紋画像を取り込み、それをクライアントコンピュータ２に送信する。クライアントコンピュータ２は、その画像を処理し、指紋テンプレートを作成する。指紋テンプレートは、指紋を個別に識別し得る指紋特徴の主要な情報を含んでいる。クライアント照合アプレット（ｂ）及びローカル照合アプレット（ａ）は、センサ３からのテンプレートと、スマートカード１ａ上に格納されたテンプレート１１とを比較する。この比較の結果、指紋センサ３から得られた指紋テンプレートとスマートカード１ａに格納された指紋テンプレートとの一致の度合いを示す類似度が得られる。
【００３９】
リンク１は、ＬＡＮやＷＡＮ４への有線又は無線接続である。ＬＡＮ及びＷＡＮ４は、ファイアウォール７、８を介してサーバ５、６に接続している。セキュリティサーバ５及びトランザクションサーバ６は、同一サーバ装置として実現してもよく、あるいは異なる装置として実現してもよい。両サーバが同一装置として設置される場合、ファイアウォールは一つあれば足りる。両サーバが別個の装置として設置される場合、各サーバは各自のファイアウォールを備えるべきである。リンク４は、セキュリティサーバ５とトランザクションサーバ６との間の付随的ネットワーク接続である。両サーバがＬＡＮ４内の同一サブネット内にあり、信頼関係（trust relationship）が確立している場合、通信はＬＡＮ４を介した直接接続とすることができる。しかしながら、ＷＡＮ４を介して情報が流れる場合は、情報漏洩や通信エラーを防止するために暗号化並びにエラー検出方式が必要となる。
【００４０】
クライアントコンピュータ２は、セキュリティサーバ５にアクセスし、身元認証に必要な全ての要素を入手する。クライアント照合アプレット（ｂ）は、セキュリティサーバ又はスマートカード１ａからクライアントコンピュータ２にダウンロードできる。クライアントコンピュータ２が、指紋の検出及び照合用の全要素を入手すると、ユーザは、以下の認証の組合せ、即ち（１）指紋のみ、（２）ＰＩＮ（識別番号又はパスワード）のみ、（３）指紋＋ＰＩＮ、（４）他のバイオメトリクスのみ、（５）他のバイオメトリクス＋指紋、又は（６）他のバイオメトリクス＋指紋＋ＰＩＮを選択することができるようになる

。
【００４１】
他のバイオメトリクスとしては、筆跡認証、顔認証、網膜スキャン、その他好適なバイオメトリクス識別要素を挙げることができる。トランザクションサーバ６は、トランザクションを実行するユーザのアクセス制限を調整するように、選択された組合せに基づいてコンフィデント・インデックス（Confident Index）を調整することができる。発行者は、バイオメトリクスの種類毎に正確度を割り当てることができる。次式は、指紋とＰＩＮとが使用される場合にコンフィデント・インデックスを使用して照合スコアを計算する例である。
$CI = FM^{*} K_1 + PIN$　・・・・・・（１）
ＣＩ：コンフィデント・インデックス
ＦＭ：指紋照合スコア（0-100）
K_1：係数
ＰＩＮ：ＰＩＮスコア
ＰＩＮスコアは、50（正当ＰＩＮ）か０（不正ＰＩＮ）である。K_1が0.5であるとき、ＣＩの範囲は、０から100となる。低セキュリティ用途の場合、スコアの境界は、50未満とすることができる。よって、システムにアクセスしうるユーザは、正当なＰＩＮか正当な指紋を有していることになる。高セキュリティ用途の場合、スコアの境界は、50を超える値にすることができる。この場合、ユーザは、トランザクションサーバ６にアクセスするためには、正当な指紋とＰＩＮ番号との両方を有していることが必要となる。
【００４２】
＜登録＞
ユーザがスマートカードを使用して認証を行えるようにする前に、ユーザはその指紋をスマートカードに登録しなければならない。図１は、トランザクション（取引）及び指紋認証を行うシステムを示している。登録はトランザクションを一切含まない。よって、トランザクションサーバへのリンク３は取り除いてもよい。スマートカードリーダ１、スマートカード１ａ、クライアントコンピュータ２及び指紋センサ３が登録に必要である。セキュリティサーバは新規ユーザの情報を記録しておく必要があるので、セキュリティサーバへの接続も同様に必要である。セキュアな登録のためには、認証されたクライアント端末２を使用する必要がある。ユーザは、指紋センサに指紋を提示する。センサは、バイオメトリクス・パラメータを取り込み、このパラメータをクライアント端末２に送信する。クライアント端末２は、秘密部分と公開部分との二つの部分からなる指紋テンプレートを作成する。クライアント端末２は、テンプレートの両部分をスマートカード１ａにアップロードする。重要情報を含む秘密部分は、ユーザ提示装置に格納され、この部分はこの装置の外部へ送られることはない。重要度の低い情報を含む公開部分は、圧縮された形でスマートカード１ａに格納される。クライアント端末２が公開部分と秘密部分とからなるテンプレートに関する処理を完了すると、該クライアント端末は、ユーザ情報をスマートカード１ａ及びセキュリティサーバ５両方にアップロードして記録する。
【００４３】
＜スマートカード上における指紋照合プロトコル＞
文献XD Jiang, WY Yau, “Fingerprint Minutiae Matching Based on the Local and Global Structures”, 15th International Conference on Pattern Recognition, Proc. ICPR 2000, Barcelona, Spain, Sept. 2000に開示された指紋照合アルゴリズムが、この方式において好適に使用される。このアルゴリズムは、二つのステージ、即ちローカルステージとグローバルステージとに分けられる。ローカルステージは、指紋特徴点のサブセット（小群）（前述の公開部分）を使用して、ユーザが提示した指紋の特徴点と、格納されたテンプレート１１における特徴点との対応を確立する。十分な対応が確認されると、格納されたテンプレート１１における特徴点のサブセットに対してユーザが提示した指紋の特徴点をマッピングする変換関数を計算する。この変換関数は、特徴点を使用して二つのテンプレートを整合させる。続いて、グローバル照合が行われ、問い合わせの指紋が登

録されている指紋に類似している度合いを示す信頼度が算出される。このようなアルゴリズムは、当然、分散処理又はクライアントーサーバ間処理に好適である。前項で記載したように、指紋照合処理は、二つの側面に分類される。スマートカード１ａ側とクライアントコンピュータ２側である。この両側面が連携して照合処理を実行する。クライアントコンピュータ２は、全ての前処理パラメータを計算し、クライアントコンピュータ２側及びスマートカード１ａ側両方からのテンプレートを再整合させる。その後、クライアントコンピュータ２は、それらを、最終的な照合を行うスマートカード１ａに送る。このような方法で指紋テンプレートを照合する理由は、以下の通りである。

【００４４】

（１）初めに登録されたテンプレート１１は決して公開されず、特徴点のサブセットのみが公開されるため認証安全度が向上する。

【００４５】

（２）非常に制限されたスマートカード１ａプロセッサの処理能力を増補するよう、クライアントコンピュータ２の処理能力を使用することにより照合速度が向上する。

【００４６】

（３）スマートカード１ａに固有の高セキュリティ特性のため、ハッカーがソフトウェア的手法を用いて、スマートカード１ａ内の照合プログラムを見ることができない。

【００４７】

図２は、スマートカード１ａ上で照合を行うプロシージャを示している。矢印は全て、データが流れる方向を示している。以下の記載は、各矢印の説明である。

【００４８】

（Ａ）クライアントコンピュータ２（ＰＣ）が、指紋センサ３から得られた指紋のテンプレートを算出する（ステップＳ１）。

【００４９】

（Ｂ）初期化後（ステップＳ２）、スマートカード１ａは、特徴点の基本部分情報を送り（ステップＳ３）、ＰＣが照合係数（暗号化情報）を計算する（ステップＳ４）。

【００５０】

（Ｃ）ＰＣが、最大で９部位の特徴点の座標をスマートカード１ａに要求する。これら特徴点を使用して、二つのテンプレートが同一座標空間に存在するようにそれらを並べる。

【００５１】

（Ｄ）スマートカード１ａが、特徴点の暗号化座標（最大９個の特徴点）をＰＣに送る（ステップＳ５）。

【００５２】

（Ｅ）ＰＣがテンプレートを並べ（ステップＳ６）、並べられた入力指紋テンプレート及び照合係数をスマートカード１ａに送る。

【００５３】

（Ｆ）スマートカード１ａは、テンプレート及び照合係数を受け取り、照合結果を計算し（ステップＳ７）、テンプレート照合の終了を知らせる認知信号をＰＣへ送る（ステップＳ８）。

【００５４】

リモート処理（即ちクライアント端末２における処理）は、二つの特徴点セットを整合させるためにのみ使用され、最大で９個の特徴点からなる完全な整合用特徴点情報すら持たず、実際のグローバル照合処理自体に影響を与えない。従って、この提唱された分散処理は、指紋照合の安全性を何ら犠牲にすることがない。

【００５５】

＜指紋スマートカード有効型トランザクションプロトコル＞

スマートカード１ａは、照合結果の計算を終了すると、その事実を、クライアントコンピュータ２を介して肯定信号（ACK Signal）によってセキュリティサーバ５に通知し、該サーバ５に対しＳＡＣＫ（Security Access Check Key）を要求する。ＳＡＣＫを使用す

ることによって、照合アプレット（ａ）は、スマートカード１ａ上でトランザクションアプレット（ｄ）を始動させる。ＳＡＣＫコードは、タイムスタンプ付きのＵＩＮ（Unique Personal Identification Number）である。ＳＡＣＫキーの詳細は後述する。図３は、システムが如何にしてトランザクションを行うかについてのメッセージパスを示している。以下のシーケンスは、ＳＡＣＫを使用してトランザクションアプレット（ｄ）を有効にする方法を示している。

【００５６】

（１．１）照合処理が完了したことを、照合アプレット（ａ）が、セキュリティサーバ５に通知する（パス１５）。

【００５７】

（１．２）セキュリティサーバ５は、スマートカード１ａ上の照合アプレット（ａ）並びにトランザクションサーバ６にＳＡＣＫキーを送る（パス１２）。

【００５８】

（１．３）照合アプレット（ａ）がＳＡＣＫを受け取ると（パス１４）、照合スコアをＳＡＣＫに付加する（以下の式２）。

【００５９】

（１．４）照合アプレット（ａ）が、トランザクションアプレット（ｄ）にＳＡＣＫを内部送信する（パス１６）。

【００６０】

（１．５）トランザクションアプレット（ｄ）が、ＳＡＣＫをデコードし、照合スコアを確認する。認証が成功すると（即ち、スコアがセキュリティ閾値より大きい場合）、アプレット（ｄ）自体がトランザクション有効状態に切り替わり、そうでない場合（即ち、スコアがセキュリティ閾値より小さい場合）、トランザクションアプレット（ｄ）自体が即座に無効となり、セキュリティ例外を扱うカードマネージャに通知する。

$SACK_{Matching\ applet}$::UIN＝$SACK_{Security\ Server}$::UIN+照合スコア

・・・・・（２）

（記号::は、その要素であることを意味する。従って、$SACK_{Matching\ applet}$::ＵＩＮは、$SACK_{Matching\ applet}$のＵＩＮフィールドであることを意味する。）

この時点までは、セキュリティサーバ５によって発行されるＳＡＣＫキーを除いて、トランザクションアプレット（ｄ）用のセキュリティ有効化のための手順全てが、スマートカード１ａの中核の内部で行われる。外部のクライアントコンピュータ２は、照合処理の補助を行うのみである。最終的な照合結果、セキュリティ有効コード及びトランザクションアプレット（ｄ）の始動は、スマートカード１ａ上で行われ、この情報はどれもクライアントコンピュータ２に送信されない。クライアントコンピュータ２の役割は、サーバとスマートカード１ａとの間のまさに通信ブリッジであり、スマートカード１ａ用のコプロセッサである。最終的判断もやはり、スマートカード１ａ上の照合結果に依存している。手ｒ順（１．５）におけるセキュリティアプレット（ｅ）自体が無効となり、セキュリティ例外が生じると、アプレット（ｅ）はセキュリティ例外を取り扱うカードマネージャに通知する。カードマネージャはセキュリティアプレット（ｅ）に対する全てのトランザクションを無効にし、セキュリティ例外通知を出力することによってクライアントコンピュータ２に報告する。当然、クライアントコンピュータ２は、例外通知を処理する機能を有している。クライアントコンピュータ２はスマートカード１ａから例外通知を受け取ると、そのようなエラーをユーザ並びにトランザクションサーバ６に報告するセキュリティ例外対処ルーチンを実行する。トランザクションサーバ６は、トランザクションを終了させ、管理者に報告し次の指示を待つ。

【００６１】

トランザクションアプレット（ｄ）が有効になると、トランザクションサーバ６は商取引を実行する。例えば、ユーザがクレジットカード会社に１００ドルを支払いたいとする。以下のプロシージャを使用して、商取引を実行することができる。トランザクション用に双方向通信チャネル（パス１８）が確立される。

【００６２】
（２．１）クライアントコンピュータ２が、購入要求と取引額（１００ドル）とをパス１８を介してトランザクションサーバ６に送信する。またクライアントコンピュータ２は、取引額をスマートカード１ａにも同様に送信する。
【００６３】
（２．２）トランザクションサーバ６は、新しいタイムスタンプ付きのＳＡＣＫキーをスマートカード１ａに送信する。
【００６４】
（２．３）スマートカード１ａのトランザクションアプレット（ｄ）が、ＳＡＣＫキーを比較し、タイムスタンプを確認する（一方は照合アプレットから、他方はトランザクションサーバ６からのものである）。
【００６５】
（２．４）両方のキーが同一であり、タイムスタンプがタイムリミット内である場合、アプレット（ｄ）は、その値（１００ドル）をスマートカード１ａから差し引く。続いて、アプレット（ｄ）はトランザクションサーバ６に、トランザクションが正常に完了したことを通知する。
【００６６】
（２．５）両方のキーが不正であるか、タイムスタンプがタイムリミットを外れている場合、トランザクションサーバアプレット（ｄ）は、トランザクション失敗メッセージを送信する。トランザクションサーバ６は、トランザクションを中止し、即座に管理者に通知する。
【００６７】
（２．６）トランザクション（サーバ）が、トランザクション成功メッセージをスマートカード１ａから受け取った場合、トランザクションサーバは、クレジットカード会社の口座に１００ドルを預けるべく、クレジットカード会社と実際のトランザクション（取引）を行う。このとき、ユーザは、電子取引によってクレジット会社に１００ドルを支払ったことになる。
【００６８】
プロシージャ（２．２）において、ＳＡＣＫキーは、実際、前述の如くセキュリティサーバ５からくるものである。唯一の違いは、そのＳＡＣＫキーが、照合スコアを伴わない新しいタイムスタンプを有している点にある。プロシージャ（２．３）において、ＳＡＣＫキーを認証する方法は、ＳＡＣＫ由来のＵＩＮフィールドを検証することによる。両方のキーがセキュリティサーバ５からきているので、それらは照合スコアを除いて同一のＵＩＮキーを有していることになる。よって、ＵＩＮキーを認証する方法は、次式（３）が成り立つなら、次式（４）となる。

$SACK_{Matching\ applet}::UIN - SACK_{Transaction\ Server}::UIN = 照合スコア$

・・・（３）

$SACK_{Transaction\ Server}::UIN = SACK_{Security\ Server}::UIN$

・・・（４）

スマートカード１ａ上のトランザクションアプレット（ｄ）は、両方のキーの減算を行う。その結果は照合スコアとなる。そうならない場合は、いずれか一方又は両方のキーが不正ということになる。タイムスタンプフィールドは、ログイン時間又はトランザクション実行時間を示している。式５は、ログインとトランザクションとの間の時間差を算出するものである。

$時間差 = SACK_{Transaction\ Server}::タイムスタンプ - SACK_{Matching\ Applet}::タイムスタンプ$

・・・（５）

ログインタイム（$SACK_{Matching\ Applet}$::タイムスタンプ）とトランザクション時間（$SACK_{Transaction\ Server}$::タイムスタンプ）との間の時間差が、タイムリミット（例えば５分）よりも長い場合、トランザクションアプレット（ｄ）はトランザクションを中止し

、エラーメッセージをトランザクションサーバ６に送り返す。この場合、トランザクションを続行するためにはユーザが再度ログインする必要がある。

【００６９】

＜ＵＩＮ及びＳＡＣＫ＞

ＳＡＣＫはセキュリティアプレット（ｅ）がトランザクションアプレット（ｄ）を有効にするためのキーであるので、トランザクションアプレット（ｄ）にトランザクションを実行させる唯一のキーである。このため、ＵＩＮが同様に提唱される。

【００７０】

各人に対し、ＵＩＮが割り当てられるが、そのＵＩＮを知ることも覚えておく必要もない。ＵＩＮは無作為に作成される。またＵＩＮは、発行会社や使用するバイオメトリクスシステムを識別しうる番号を含んでいてもよい。加えて、例えば指紋のような各人のバイオメトリクスを獲得する。既に説明したように、各人は、既にそのようなサービスを登録した指紋認証システムに自分の指紋を提示することによって、指紋による身元認証を行うことができる。指紋認証システムは、その人物がシステムに登録された人物と同一であるか否かについての信頼度に対応した番号を生成する。換言すれば、指紋認証システムは、その人物が自分自身であると主張する人物であることの確実性の度合い、所謂照合スコアを作成する。この番号がＵＩＮに加えられ、その新たな番号はバイオメトリクス識別番号（ＢＩＮ：biometric identification number）と称される。その人物が本人である可能性が１００％であることに対応する最大可能照合スコア（ＡＭＳ：achievable matching score）は、どのような番号でもよく、必ずしも１００である必要はなく、１００００であってもよい。同様に、最小照合スコア（ＩＭＳ：minimum matching score）をシステムに割り当てることも可能である。セキュリティを向上させるため、タイムスタンプ及びランダムキーがＳＡＣＫキーに加えられる。図４はＳＡＣＫキーのフォーマットを図示しており、各フィールドは以下の通りである。

【００７１】

Ａ１：スクランブル関数キー

Ａ２：ランダムキー

Ａ３：タイムスタンプ

Ａ４：ＵＩＮ

ＣＳ：チェック・サム

Ａ１はスクランブル関数キーであるので、Ａ１は暗号化されない。Ａ１は、スクランブル関数（シフト、回転、定数加算等）を選択し、セキュリティ上の観点から後続のフィールド（Ａ２－Ａ４）をスクランブル化ために、どのビット又はバイトがどのデータタイプに関連しているかを選択するために使用される。Ａ２及びＡ３は夫々、前項で説明したランダムキー及びタイムスタンプである。ＣＳは、送信エラーを防止するためのキー全体のチェック・サムである。Ａ４はＵＩＮである。

【００７２】

ＵＩＮの一例を以下に挙げる。

【００７３】

ＵＩＮ：2 345 678 988 011 009

最大照合スコア：10 000

最小照合スコア：2500

最大ＢＩＮ：2 345 678 988 011 009 + 10 000 = 2 345 678 988 021 009

最小ＢＩＮ：2 345 678 988 011 009 + 2500 = 2 345 678 988 013 509

ユーザはＵＩＮを知らないので、ＢＩＮもまたユーザには知られることはない。同様に、ユーザはこの番号を覚えておく必要もない。ＵＩＮは、それを発行するセキュリティサーバ５など、発行会社やユーザのコンピュータ記録媒体によって保管される。例えば、そのシステムを使用したいユーザは、そのサービスに登録することによって、指紋などのバイオメトリクスを獲得する。発行会社は、ＵＩＮ及び指紋テンプレートをスマートカード１ａに格納し、それをユーザに付与する。セキュリティ向上のため、システムを使用する

毎にＵＩＮを変更することも可能である。毎回ＵＩＮを変更するには、セキュリティサーバ５及びリモートクライアント装置（スマートカード１ａ又はクライアントコンピュータ２）は、ＵＩＮの変更を管理するセキュリティアプレット（ｅ）又はセキュリティプロセスを有していることが必要となる。サーバは新しく暗号化されたＵＩＮをリモートクライアント装置に送信する。クライアント装置は、ＵＩＮを解読し、それを古いＵＩＮに追加する。メッセージの前半部分は、古いＵＩＮであり、後半部分は新しいＵＩＮである。トランザクションが完了すると、ＵＩＮの前半部分が消去され、ＵＩＮの後半部分が前半部分に移動して次のトランザクションに備えられる。以下の例はＵＩＮの作成方法及び使用方法を示している。

【００７４】

受信番号：2 345 678 988 011 009
現行ＵＩＮ：23 456 789
次回ＵＩＮ：88 011 009
最大照合スコア：10 000
最小照合スコア：2500
最大現行ＢＩＮ：23 456 789 + 10 000 = 23 466 789
最小ＢＩＮ：23 456 789 + 2500 = 23 459 289
次の工程における受信番号を8 801 100 977 123 456と仮定する。

【００７５】

現行ＵＩＮ：88 011 009
次回ＵＩＮ：77 123 456
新しい装置が設置された場合や新しいユーザのために、ヌル（null）ＵＩＮのようなリセットコードを使用して、ＵＩＮをリセットすることもできる。

【００７６】

リセットの際、ＵＩＮはクライアント装置にもセキュリティサーバ５にも無視され、クライアント装置はセキュリティサーバ５により生成された新しいＳＡＣＫに基づく新しいＵＩＮを使用する。

【００７７】

＜バイオメトリクスを使用した識別プロシージャ＞
ユーザがシステムにアクセスしようとするとき、以下のシーケンスが発生する。

【００７８】

（３．１）ユーザがスマートカードリーダ１にスマートカード１ａを挿入する。

【００７９】

（３．２）クライアントコンピュータ２が、パスワード又はＰＩＮの入力を要求する。

【００８０】

（３．３．）ユーザがクライアントコンピュータ２にパスワード又はＰＩＮを入力する。

【００８１】

（３．４）クライアントコンピュータ２が指紋を要求する。

【００８２】

（３．５）ユーザがセンサ３上に指を置く。

【００８３】

（３．６）クライアントコンピュータ２が指紋を獲得し、指紋テンプレートを抽出する。

【００８４】

（３．７）クライアントコンピュータ２が、スマートカード１ａにＰＩＮ及び指紋テンプレートを提供する。

【００８５】

（３．８）スマートカード１ａは、その指紋を保存されているテンプレート１１と照合する。

【００８６】
（３．９）照合アプレット（ａ）が、セキュリティサーバ５又はスマートカード１ａからのＵＩＮを要求する。
【００８７】
（３．１０）得られた照合スコアが正しい場合（即ち、得られた照合スコアがＩＭＳ（最小照合スコア）とＡＭＳ（最大照合スコア）との間にある場合）、得られた照合スコアをＵＩＮに加算して、バイオメトリクス識別番号（ＢＩＮ）を得る。
【００８８】
例：得られた照合スコア＝4500
ＢＩＮ＝2 345 678 988 011 009 + 4 500 = 2 345 678 988 015 509
（３．１１）指紋が、ＩＭＳを下回るか、ＡＭＳを超える場合は、ＡＭＳより大きな無作為番号がＵＩＮに加算されてＢＩＮを得る。
【００８９】
例：得られた照合スコア＝100
作成された無作為照合スコア＝18 000
ＢＩＮ＝2 345 678 988 011 009 + 18 000 = 2 345 678 988 029 009
（３．１２）スマートカード１ａがＰＩＮとＢＩＮとを組合せ、それらを暗号化してクライアントコンピュータ２に送信し、それからクライアントコンピュータ２がトランザクションサーバ６にそれらを送信する。ユーザ名等の他の情報を、この送信メッセージに含ませてもよい。またトランザクションサーバ６は、ＳＡＣＫキーも算出する。
【００９０】
（３．１３）トランザクションサーバ６は、ＰＩＮを照合し、ＢＩＮからＵＩＮを差し引いて照合スコア（ＭＳ）を得る。以下のような６通りの場合が想定される。
【００９１】
（ａ）ＰＩＮが合致し、ＭＳが可能スコア内である（ＩＭＳ≦ＭＳ≦ＡＭＳ）。
【００９２】
（ｂ）ＰＩＮは合致するが、ＭＳが可能スコア外である。
【００９３】
（ｃ）ＰＩＮは合致しないが、ＭＳは可能スコア内である。
【００９４】
（ｄ）ＰＩＮが合致せず、ＭＳも可能スコア外である。
【００９５】
（ｅ）ＰＩＮが合致し、ＭＳ＝０である。
【００９６】
（ｆ）ＰＩＮが合致せず、ＭＳ＝０である。
【００９７】
（注：トランザクションアプレット（ｄ）が有効にされない場合、それはスマートカード１ａ側からのログイン失敗を意味しており、トランザクションはここで停止する。）
（３．１４）上記の結果に対応する判断がなされる。
【００９８】
例：上記（ａ）の場合、完全アクセスが許可される。
【００９９】
上記（ｂ）の場合、制限付きアクセスが許可される（又はユーザが再度試みる）。
【０１００】
上記（ｃ）の場合、制限付きアクセスが許可される（又はユーザが再度試みる）。
【０１０１】
上記（ｄ）の場合、スマートカード１ａが保留される。
【０１０２】
上記（ｅ）の場合、スマートカードはパスワードにより保護されたクレジットカードとして取り扱われる。

【０１０３】
上記（ｆ）の場合、スマートカードは通常のクレジットカードとして取り扱われる。
【０１０４】
（３．１５）トランザクションサーバ６が、その判断をスマートカードに送信し、スマートカードはその判断に従って機能する。
【０１０５】
ＵＩＮに、例えば銀行コードなど、組織を識別する識別要素を含ませることも可能である。同様に、ＵＩＮに、複数のバイオメトリクス・パラメータから得られる認証結果を取り込むことも可能であり、また使用するバイオメトリクスの種類やバイオメトリクスシステムのプロバイダなども併せて含ませることも可能である。代替手段として、このような番号を、暗号化されたメッセージにヘッダ又はフッタとして添付することも可能である。このような番号は暗号化する必要がない。
【０１０６】
バイオメトリクスシステムに識別コードを組み合わせることは、以下の利点を有する。
【０１０７】
１．スマートカードを発行する会社が、一つのバイオメトリクスシステムプロバイダに拘束される必要がない。発行会社によって特定されたプロトコルを満たすスマートカードであれば任意のものが使用可能である。
【０１０８】
２．ユーザが自分に都合のよいバイオメトリクスを選択することができる。
【０１０９】
３．発行会社は使用したバイオメトリクスの全履歴を保存しておく必要がなく、サーバ内でバイオメトリクス照合を実行する必要がない。上記の如く、ＢＩＮによってバイオメトリクスシステムの組合せや照合が可能になる。
【０１１０】
４．ＢＩＮは今日使用されているＰＩＮシステムに類似しているので、発行会社は、現行のＰＩＮシステムを使用している現行のトランザクションサーバ６を容易にアップグレードして、バイオメトリクス特性を取り入れることができる。
【０１１１】
またユーザ身元認証の最もセキュアな又は好適な方法を自動的に選択することも可能である。認証方式としては、使用するバイオメトリクスの種類又はパスワード等がある。各認証方式には、固有の番号、即ち認証装置識別番号が付与される。クライアントコンピュータ２は、全ての可能な認証方式を識別し、これら装置の識別番号を保存する。セキュリティサーバ５側では、これら認証方式各々の安全性や信頼度をサーバシステムが等級付けする。等級付けは、通常、ユーザの嗜好や選択に依存する。このような等級付けは、装置記識別番号が格納された項目表（エントリテーブル）を作成することによってなされる。項目表における最上位の項目は、最も信頼度の高いユーザ認証手段に対応しており、最下位の項目は、最も信頼度の低い手段である。身元認証が必要なとき、クライアントコンピュータ２は、使用可能な認証装置識別番号を提示することによって、使用可能な認証方式を提示する。続いてセキュリティサーバ５が、ユーザの身元認証に最も適切な手段をこれらの項目から選択する。
【０１１２】
プロトコルは以下の通りである。
【０１１３】
（４．１）セキュリティサーバ５は、会社に認容された認証形式の信頼度を等級付けする項目表を準備する。
【０１１４】
（４．２）クライアントコンピュータ２が使用される際、使用可能な認証形式のインデックスがセキュリティサーバ５に送信される。

【０１１５】
（４．３）セキュリティサーバ５は、使用する認証形式の種類を決定し、クライアントコンピュータ２に返信する。
【０１１６】
（４．４）クライアントコンピュータ２は、コマンドを獲得し、ユーザを識別するために使用するよう要求された認証形式を実行する。
【０１１７】
（４．５）続いて、セキュリティサーバ５は、クライアントコンピュータ２及びスマートカード１ａによって集積され処理されたデータからの応答を照合し、ユーザの身元認証を行う。
【０１１８】
＜分散リモート実行マネージャプロトコル＞
以上、クライアントコンピュータ２による補助を受けたスマートカード１ａ上における指紋照合プロトコルを説明してきた。ここで、スマートカード１ａがクライアントコンピュータ２と並行してどのように関数を実行するかについての詳細を説明する。
【０１１９】
従来のスマートカードは、限られたメモリ（ＲＡＭ／ＥＥＰＲＯＭ）、処理能力（８ビット／１６ビット）及び速度（１５MHz未満）などに例示されるように、リソースが制限されていた。またクライアントコンピュータ２からスマートカード１ａへのデータ転送速度も遅い。以下に説明する分散リモート実行マネージャプロトコルは、分散処理によりデータ転送速度の遅さを解消するものである。このプロトコルは、上記リモート認証処理について説明されるが、処理能力が低く、通信リンクが低速な他の演算装置においても適用可能である。プロトコルを支持するアーキテクチャは、リモート実行を要求する複数のリモート装置あるいは要求されたプロセスを管理する複数のサーバに拡大することができる。
【０１２０】
＜分散処理アーキテクチャ＞
図５は、マルチ処理ユニットを単純化したアーキテクチャを示している。
【０１２１】
処理ユニット１（ＰＵ１）は処理ユニット２（ＰＵ２）よりも大きな処理能力を有しているものとする。ＰＵ１及びＰＵ２は、夫々、クライアントコンピュータ及びリモート装置（スマートカード又はＰＤＡのような処理能力の小さな装置）である。クライアントコンピュータＰＵ１は、タスク処理を行う。リモート装置ＰＵ２は、クライアントコンピュータＰＵ１に、タスク実行を要求する。各処理ユニットは、タスク詳細を記録したタスク表の写しを有している。図６は、タスク表の一例である。
【０１２２】
タスク表において、第一欄は、オブジェクト名及びそれらの関数を含んでいる。第二欄は、オブジェクトＩＤ及び関数ＩＤを含んでいる。これら名称はいずれもASCIIフォーマットである。オブジェクトＩＤは例えば32ビットの符号付き整数であり、関数ＩＤは、例えば16ビットの符号付き整数である。ＥＯＴは、表の終了（End Of Table）を意味しており、例えば16ビットの符号付き整数とすることができる。ＥＯＴは、「－１」である。タスク表の第三欄及び第四欄は、クライアント側（ＰＵ１）専用である。リモート装置（ＰＵ２）には、第三欄及び第四欄はない。第三欄は、パラメータ表であり、パラメータのタイプ・キャスティングを記載している。第四欄は、関数又はオブジェクトの開始アドレスである関数／オブジェクト・エントリーポイントを記載している。タスク表内のオブジェクト数に制限はないが、プログラマーは、メモリのオーバーフロー例外を防止するため、リモート装置（ＰＵ２）のメモリサイズを意識するべきである。タスク表は、ＩＤ番号に対する関数名／オブジェクト名のマッピングである。リモート装置（ＰＵ２）において実行される分散型オブジェクト実行マネージャ（Ｄ－ＯＥＭ：Distributed-Object Execution Manager）は、このタスク表を使用して、タスク（オブジェクト及び関数）を実行する

クライアント側ＰＵ１のマネージャＤ－ＯＥＭ（Ｄ－ＯＥＭ＿client）に通知する。クライアント側ＰＵ１のＤ－ＯＥＭ＿clientは、タスク表を用いて関数の開始アドレスにＩＤをマッピングすることによって関数を実行する。

【０１２３】

図７は、分散処理のアーキテクチャを示す模式図である。ＰＵ１及びＰＵ２内のタスク表の全項目を初期化するには、動的束縛（dynamic binding）が使用され、ランタイム中に関数エントリポイントが決定される。クライアント処理ユニットＰＵ１は、４つのレベルに分けられている。第一のレベルは、図７における黒い実線で示される通信チャネル（ＣＬ）を使用する通信を処理するクライアントマネージャ（ＣＭ）である。第二のレベルは、Ｄ－ＯＥＭ＿clientである。タスク表ＴＴ１は、Ｄ－ＯＥＭ＿client内に格納される。第三のレベルは、関数の実行を制御する実行マネージャＥＭ１である。第四のレベルは、プログラム領域ＰＡ１である。全ての関数及びオブジェクトは、このレベルに格納される。

【０１２４】

リモート装置ＰＵ２も同様の構造を有しているが、クライアントマネージャＣＭの代わりに、リモート装置マネージャＲＤＭを有しており、また実行マネージャを持たない。その代わり、ＰＵ２におけるＤ－ＯＥＭは、関数の戻り値を処理するスタックＳを有している。このスタックを用いる詳細については後述する。

【０１２５】

タスク表の初期化前は、オブジェクト名欄を除くタスク表内の全項目が空白である。リモート装置ＰＵ２がローカルクライアントＰＵ１内の関数又はオブジェクト呼び出す前に、Ｄ－ＯＥＭはクライアントＰＵ１及びリモート装置ＰＵ２両方のタスク表を初期化する必要がある。図７の各工程は、タスク表を初期化する方法を示している。

【０１２６】

ステップＳ１０：リモート装置ＰＵ２が、要求コマンド（Initialization_0f_0bject）ＴＴ１をクライアントＰＵ１に送信する。要求コマンドInitialization_0f_0bjectのフォーマットを図８に示す。

【０１２７】

ステップＳ１２：Ｄ－ＯＥＭ＿clientが、タスク表ＴＴ１内のオブジェクト名を検索する。

【０１２８】

ステップＳ１４：Ｄ－ＯＥＭ＿clientがタスク表ＴＴ１内のオブジェクトを発見すると、Ｄ－ＯＥＭ＿clientは実行マネージャＥＭ１を呼び出して、オブジェクトのコンストラクタを呼び出し、全てのオブジェクト変数及び関数を初期化する。実行マネージャＥＭ１は、全パラメータ及びエントリポイントを計算し、タスク表ＴＴ１内のこれらの項目（第三欄及び第四欄）を埋める。

【０１２９】

ステップＳ１６：Ｄ－ＯＥＭ＿clientが所定名のオブジェクトを発見できない場合、Ｄ－ＯＥＭ＿clientは、当該システムに例外をスローして工程を終了させる。

【０１３０】

ステップＳ１８：実行マネージャＥＭ１がオブジェクト作成を終了すると、Ｄ－ＯＥＭ＿clientは、パブリック（public）な関数、保護された（protected）関数及びオブジェクト自体のＩＤを全て作成する。Ｄ－ＯＥＭ＿clientはタスク表ＴＴ１にＩＤを記載する（第二欄）。

【０１３１】

ステップＳ２０：Ｄ－ＯＥＭ＿clientが第二欄のＩＤをリモート装置ＰＵ２に送信する。

【０１３２】

ステップＳ２２：リモート装置ＰＵ２がＩＤを受信し、それら項目をローカルタスク表ＴＴ２に保存する。

【０１３３】
システムがオブジェクトを使用する必要があるとき、オブジェクトはいつでも初期化可能である。システムが異なるオブジェクトを呼び出すためにＤ－ＯＥＭがタスク表ＴＴ２のマルチコピーを作成するとき、オブジェクトアレイを使用することができる。ＩＤ作成についての詳細は後述する。リモート装置ＰＵ２上のコードが終了すると、タスク表内の全項目が一気に消去される。
【０１３４】
図８には、Initialization_0f_0bjectコマンドのフォーマットが示されている。第一のフィールドは、リモート装置ＩＤ８２である。スマートカードの場合、このＩＤをスマートカードＩＤとすることができる。ネットワークを使用するモバイル装置の場合、このＩＤはＩＰアドレス又はＭＡＣアドレスとすることができる。Ｄ－ＯＥＭ開始コマンド８０は、Ｄ－ＯＥＭ＿clientにリモートオブジェクト実行初期化を行わせる固有バイナリコマンドである。オブジェクト名８４は、ASCIIフォーマットのオブジェクト名である。パラメータ８６は、オブジェクト作成の初期値である。エンドコード８８は、送信終了を示している。
【０１３５】
＜リモート関数の実行と同期化＞
リモート関数を実行するには、リモート装置ＰＵ２が、そのＤ－ＯＥＭを使用して関数を呼び出し、クライアントコンピュータＰＵ１側でその関数を実行することが必要である。オブジェクト１のメンバ関数である関数１をリモート装置ＰＵ２が呼び出す必要がある場合の一例をここに挙げる。戻り値は、D_OEM・execute（“object1”，“function 1”，wait_status，para1，para2,…)である。
【０１３６】
Ｄ－ＯＥＭは静的ベースオブジェクトである。このオブジェクトは、ローカル関数又はオブジェクトがＤ－ＯＥＭオブジェクトのコピーを作成又は継承（inherit）する場合、動的オブジェクトとなり得る。関数execute（…）は、Ｄ－ＯＥＭのパブリックメンバ関数である。この関数は、クライアントコンピュータＰＵ１において関数を遠隔的に開始する。第一のパラメータは、オブジェクト名である。第二のパラメータは、関数名である。第三のパラメータは、同期待機状態の種類である。次の関数の実行に戻り値が不要であり、この関数と同時に実行可能であるなら、「no_wait」を使用して同時処理を実行することができる。それ以外の場合、システムは、リモート処理からの応答を待つ。リモートプロシージャ応答の待機時間は、1から65535（16ビット符号なし整数の場合）である。「０」は「no_wait」を意味している。単位はミリ秒である。タイムアウトした後でプロセスが何の応答も受信できない場合、タイムアウト例外がＤ－ＯＥＭにスローされる。他のパラメータ（para1，para2,…）は、関数１のパラメータである。Ｄ－ＯＥＭは、戻り値、オブジェクトＩＤ及び関数ＩＤなどの情報をスタックに格納する。スタックは、戻りパラメータを処理するために使用される。Ｄ－ＯＥＭは、パラメータを暗号化してコマンドストリング（図９）を作成し、このストリングをリモート装置マネージャＲＤＭに送信する。Ｄ－ＯＥＭは、オブジェクトＩＤ９１及び関数ＩＤ９２をタスク表ＴＴ２から探す。図９では、コマンドストリングは二つのパラメータ（パラメータ１（９４）とパラメータ２（９５））のみを有している。パラメータの数は、関数自体に依存する。よって、ストリングは任意の数のパラメータ（必ずしも図９に例示するように二つに限定されない）を含むことができる。
【０１３７】
このコマンドストリングは、リモート装置マネージャＲＤＭへ送信される。その後、リモート装置マネージャＲＤＭはこれをクライアントコンピュータＰＵ１に通信リンクＣＬを介して送信する。任意の装置搭載型暗号エンジンを使用して、セキュアなデータ転送のためにストリングを暗号化することができる。クライアントコンピュータＰＵ１がこのコマンドストリングを受信すると、クライアントコンピュータはそれをクライアントマネージャＣＭに送信する。Ｄ－ＯＥＭ_func_call９０は、リモート装置ＰＵ２がリモート関数

呼出を要求することを示すバイナリコマンドである。Ｄ－ＯＥＭ＿clientは、このストリングを解読し、タスク表ＴＴ１を使用して、オブジェクト及び関数のソフトウェア・エントリポイントを調べる。Ｄ－ＯＥＭ＿clientは、エントリポイントと全ての関数パラメータを実行マネージャＥＭに送り、関数を開始する。実行マネージャＥＭを使用する理由は、関数を正規関数又はスレッドとなり得るからである。関数がスレッドの場合、実行マネージャＥＭはリモート装置ＰＵ２の同期化を行う。リモート装置ＰＵ２は、リモートスレッドを直接、仮想的に制御することができる。関数が終了すると、その関数は戻り値をＤ－ＯＥＭ＿clientに返信する（図７における破線）。Ｄ－ＯＥＭ＿clientは、クライアントマネージャＣＭ、通信リンクＣＬ及びリモート装置マネージャＲＤＭを介して戻り値をＤ－ＯＥＭに送る。図１０は、クライアントからの戻りパラメータのフォーマットを示している。

【０１３８】

リモート装置ＰＵ２上のＤ－ＯＥＭは、戻りパラメータのヘッダを調べる。D-OEM_return_param１００は、バイナリストリングが関数の戻りパラメータを含んでいることを示している。Ｄ－ＯＥＭは、正しいヘッダを持つバイナリストリングのみを受け入れる。Ｄ－ＯＥＭは、スタックＳ内のオブジェクトＩＤ１０２及び関数ＩＤ１０４によって、戻りオブジェクトの情報を検索する。その後、Ｄ－ＯＥＭは、戻り値を変数として保存し、スタックＳにこの変数を格納する。タイプ（種類）１０６は、戻り値のタイプ・キャスティングを示している。第五のフィールドは、関数の実際の戻り値１０８を含んでいる。エンドコード１１０はストリングの終了を意味している。

【０１３９】

並列関数の同期化は必要である。「wait_status」が「no_wait」に設定される場合、リモート装置ＰＵ２上のＤ－ＯＥＭは、同期化のために特別なプロシージャを行うことが必要となる。プログラムが明示的な同期化を必要とする場合、関数synchronized（…）を使用することができる。以下の表１に示すJavaコードフラグメントを考える。

【０１４０】

【表１】

```
1 : r_value = D_OEM・execute (“object1”, “function1”, no_wait, para1, p
ara2);
2 : coef = calc (input);
3 : result = coef*return_value;
……
```

【０１４１】

第一行はr_valueを計算する。第二行はcoefを計算する。これらの行は互いに依存していないので、同時に実行することが可能である。第一行はクライアントコンピュータＰＵ１上で実行され、第二行はローカルで実行、即ちリモート装置ＰＵ２で実行される。しかしながら、第三行の結果を得るには全ての戻り値が必要である。第一行の関数が終了しておらず、第二行の関数が終了している場合（即ち、coefは使用可能であるが、r_valueがまだ準備できていない場合）、r_valueの計算結果を待たずに第三行を実行すると第三行の関数はエラーを発生する。よって、このような場合、同期化を行うことが必要である。以下の表２に示すコードフラグメントを考える。

【０１４２】

【表２】

```
1 : r_value = D_OEM・execute (“object1”, “function1”, no_wait, para1, p
ara2);
2 : coef = calc (input);
3 : D_OEM・synchronize (“object1”, “function1”, timeout);
4 : result = coef*return_value;
.....
```

【０１４３】

このコードフラグメントには更に別の行が追加されている。第三行は同期化処理を行う。関数synchronize（…）は、Ｄ－ＯＥＭのパブリックメンバ関数である。この関数によって、システムは、変数（r_value）がクライアントコンピュータＰＵ１からの戻り値を受け取るまで待機する。最初の二つのパラメータは、どの関数が同期化の実行を必要として呼び出されたかを識別するために使用される。第三のパラメータは、最大タイムアウト時間をミリ秒単位で示している。Ｄ－ＯＥＭオブジェクトがタイムアウトになると、システムに例外がスローされる。リモート装置マネージャＲＤＭは例外を捕捉して、エラーを処理する。戻り値（return_value）を受け取ると、システムは次の行を実行して計算結果を出す。以下のプロシージャは、Ｄ－ＯＥＭがどのように同期化を行うかを示している。

【０１４４】

（５．１）同期化方法を開始し、タイマをリセットする。

【０１４５】

（５．２）タイマ値を調べ、もしその値がタイムアウトしていたら、例外をスローし、同期化方法を中止する。

【０１４６】

（５．３）Ｄ－ＯＥＭコミットバッファが戻りパラメータを受け取っているかを調べる。正しいパラメータがその関数に属していれば、同期化方法を終了し、そのデータを変数（r_value）に差し戻す。

【０１４７】

（５．４）ステップ２にループバックして関数を待ち続ける。

【０１４８】

＜オブジェクトＩＤ及び関数ＩＤの作成＞

これは分散処理アーキテクチャであるので、通信はどのような種類のもの（有線又は無線）でもよい。複数のリモート装置を一つのクライアントコンピュータＰＵに接続することも可能である。この場合、クライアントコンピュータＰＵは、複数のリモート装置に対処するために、より大きな処理能力を必要とする。そこで、処理ユニットを高パフォーマンスサーバとすることができる。システムが単独のクライアントを有しているか複数のクライアントを有しているかに拘わらず、ＩＤ作成方法、並びに複数の要求を処理するプロトコルは同一である。

【０１４９】

図１２は、一つのクライアントコンピュータＰＵに接続された二つのリモート装置ＲＤ１、ＲＤ２を示している。各リモート装置は、オブジェクトを実行するようにクライアントコンピュータＰＵに要求する。これらのオブジェクトは同一プロセスであっても異なるプロセスであってもよい。どの処理（プロセス）又はオブジェクトがリモート装置から要求されても、実行マネージャは固有ＩＤをオブジェクト及びそのメンバ関数に割り当てることが必要である。固有ＩＤを作成するには、リモート装置毎に別々のタスク表を作成する。全てのオブジェクトＩＤの作成に24ビットカウンタを使用し、全ての関数ＩＤの作成に16ビットカウンタを使用する場合、メンバ関数ＩＤは48ビット（16ビット+32ビット）となる。並行タスクを同時に要求する多数のリモート装置がある場合であっても、カウンタがオーバーフローすることで重複するＩＤが生成されることは考えにくい。オブジェク

トＩＤが32ビットの整数である場合、オブジェクトカウンタは、24ビットのＩＤを発生するのみである。最後の8ビットは、マルチサーバ環境におけるサーバＩＤ用にリザーブされている。マルチサーバ環境については後述する。この場合、最後の8ビットは全て「０」に設定される。実行マネージャは図１２Ａに図示する工程を使用して、ソフトウェアオブジェクト及びそのメンバ関数を初期化する。

【０１５０】

ステップＳ３０：Ｄ－ＯＥＭ_clientが、リモート装置の一つから要求を受け取る（Initialization_0f-0bject）。

【０１５１】

ステップＳ３２：Ｄ－ＯＥＭ_clientが、タスク表内のオブジェクト名を使用してオブジェクトを検索する。Ｄ－ＯＥＭ_clientが所定名のオブジェクトを発見できなければ、例外がスローされる（ステップＳ３４）。そうでなければ、オブジェクト作成（ステップＳ３６）が実行される。実行マネージャがオブジェクトを作成し、オブジェクトＩＤカウンタからの新しいオブジェクト名を要求する。続いて、マネージャがカウンタをインクリメントし、次のオブジェクトに備える。

【０１５２】

ステップＳ３８：実行マネージャが、オブジェクトのパブリックメンバ関数及び保護されたメンバ関数について、関数ＩＤカウンタを使用して全ての関数ＩＤを作成する。

【０１５３】

ステップＳ４０：実行マネージャがＤ－ＯＥＭ_clientに全ＩＤを送り返す。

【０１５４】

ステップＳ４２：Ｄ－ＯＥＭ_clientが、特定のリモート装置に属しているタスク表の存在を調べる。そのようなタスク表がない場合、そのリモート装置ＩＤを持つ新しいタスク表のコピーを作成する。そうでない場合、Ｄ－ＯＥＭ_clientが、既存の表に項目を追加する。クライアントコンピュータＰＵが複数のリモート装置に接続している場合、Initialization_0f-0bjectを送信する特定装置について独立のタスク表が作成される。

【０１５５】

ステップＳ４４：Ｄ－ＯＥＭ_clientが全ＩＤを要求元（リクエスタ）に送り返す。

【０１５６】

ステップＳ４２において、一つのリモート装置ＲＤ１のみがクライアントに接続されている場合、タスク表のコピーが只一つ作成される。複数装置の場合、各装置がタスク表のコピーを夫々有する。図１３は、オブジェクト名及び関数ＩＤを暗号化するバイナリストリングを示している。

【０１５７】

Ｄ－ＯＥＭ_clientは、このストリングをクライアントマネージャに送信する。クライアントマネージャは、このバイナリストリングを正しいリモート装置に返信するパスを認識している。リモート装置がこのストリングを受け取ると、そのリモート装置のリモート装置ＩＤが調べられる。そのＩＤが正しければ、このストリングがＤ－ＯＥＭに送られる。Ｄ－ＯＥＭは全ＩＤを解読し、解読したＩＤをタスク表に格納する。ＩＤ数は、ストリング中のＩＤ（オブジェクトＩＤも含む）総数を示している。

【０１５８】

＜スマートカードにおける指紋照合の負荷を分散する分散処理の使用＞

処理能力の低いリモート装置コンピュータ又はスマートカード１ａにおいて完全照合を行うには非常に長い時間を要する。この問題を解消するため、処理能力の大きなローカルクライアントコンピュータ２を使用する負荷分担機構によって、指紋照合の処理時間を短縮することができる。以下、分散リモート実行プロトコルを使用する前記指紋照合の負荷分担について説明する。

【０１５９】

特徴点検出を除いて、上述の特徴点を使用する照合アルゴリズムは二つの部分、即ち基本検出（ローカルステージ）及び特徴点照合（グローバルステージ）に分けることができ

る。基本検出プロシージャは、最適な基本特徴点を検出し、異なる二つの指紋テンプレートを並べる（整合する）ものである。基本検出プロシージャは、二つのテンプレートの整合に必要な幾つかのパラメータを計算し、これらパラメータを照合ステージに送り、そこで入力された指紋が合致するのか否かが決定される。しかしながら、基本検出部分は計算集約的（computationally intensive）である。この部分がスマートカード１ａで計算されるとすると、照合結果を得るには１分以上要することになる。速度増加のため、クライアントコンピュータ２がその計算の一部を担う。クライアントコンピュータ２は、認証を行うスマートカード１ａのコプロセッサとして機能する。前述の通り、この方法によれば、基本検出の際には極一部の指紋特徴点群（サブセット）のみを処理するのでセキュリティが犠牲とならない。図１３Ａの工程は、スマートカード１ａにおける指紋照合を負荷分担させる分散リモート実行プロトコルの使用方法の一例を示している。

【０１６０】

ステップＳ５０：指紋照合を開始する。

【０１６１】

ステップＳ５２：ベース・エスティメーション・オブジェクト（base estimation object）の初期化を行うため、スマートカード１ａ上のＤ－ＯＥＭが、Initialization_0f-0bjectをクライアントコンピュータ２に送信する。

【０１６２】

ステップＳ５４：クライアントコンピュータ２におけるＤ－ＯＥＭ_clientが実行マネージャを介してオブジェクトを初期化し、オブジェクトＩＤ及び関数ＩＤをスマートカード１ａに返信する。

【０１６３】

ステップＳ５６：スマートカード１ａが基本情報をクライアントコンピュータ２に送信する。

【０１６４】

ステップＳ５８：スマートカード１ａのＤ－ＯＥＭが、リモート処理要求コマンドを送り、ベース・エスティメーションをリモートにクライアントコンピュータ２上で開始する。

【０１６５】

ステップＳ６０：Ｄ－ＯＥＭ_clientが、実行マネージャにベース・エスティメーション・ルーティンを実行するよう促す。

【０１６６】

ステップＳ６２：スマートカード１ａは、クライアントコンピュータ２がベース・エスティメーションを終了するまで待機する。

【０１６７】

ステップＳ６４：ベース・エスティメーションの結果、特徴点指数の数が「０」であれば、スマートカード１ａは指紋照合を停止し、「照合失敗」をクライアントコンピュータ２に返信する。

【０１６８】

ステップＳ６６：スマートカード１ａは、プレ照合係数及び並べられた入力指紋テンプレートをダウンロードする。

【０１６９】

ステップＳ６８：スマートカード１ａ上で最終的照合を行う。

【０１７０】

ステップＳ５４及びＳ６６は、前記プロトコルを使用して、スマートカード１ａとクライアントコンピュータ２との間で照合情報を送受信する。

【０１７１】

＜複数の処理ユニットによるプロセスの実行＞

前項で説明したシナリオは、単純なＰ２Ｐ（ピアツーピア）通信及び処理である。即ち、処理能力の小さなリモート装置２００がクライアントコンピュータ２０２に処理負担を

要求するものである。しかしながら、クライアントコンピュータ２０２で実行することがセキュアではない処理がある場合や、クライアントコンピュータ２０２が非常に計算集約的なコード処理をすることができない場合など、マルチプロセッササーバ２０４、２０６のような処理能力の大きな処理ユニットを使用して、実行を補助する。図１４の場合を考える。

【０１７２】

図１４では、リモート装置２００がクライアントコンピュータ２０２に接続している。クライアントコンピュータ２０２は、処理能力に制限のある端末であり、負荷分担によりリモート装置２００を補助することができない。更に、クライアントコンピュータ２０２がパブリックコンピュータであって、ネットワークサーバがこのクライアントコンピュータ２０２と何の信頼関係も構築していない場合、リモート装置２００との負荷分担を行う目的の認証プロシージャを実行することは全くセキュアではない。従って、低性能クライアントコンピュータ２０２や信頼性のないクライアントコンピュータ２０２については、サーバ側で負荷分担を行うことができる。クライアントコンピュータ２０２は、リモート装置２００とサーバ２０４、２０６との間の通信ブリッジとして機能する。

【０１７３】

図１４では、二つのサーバ２０４、２０６がクライアントコンピュータ２０２に接続している。リモート装置２００が指紋認証処理を実行する必要がある場合、リモート装置２００は、認証サーバ２０４においてリモート処理要求を開始する。前項に記載したものと同様のアーキテクチャを使用することができる。リモート装置２００におけるＤ－ＯＥＭを使用して、リモート処理を要求することができ、認証サーバ２０４におけるＤ－ＯＥＭ_clientを使用して、関数／オブジェクトのプロセスを管理することができる。クライアントコンピュータ２０２の場合、Ｄ－ＯＥＭ_bridgeを使用してブリッジ・プロシージャを管理する。Ｄ－ＯＥＭ及びＤ－ＯＥＭ_clientの構造及びコマンドは、前記オリジナルフォーマットと同一であり、この新しいシステム環境に対応するための修正は不要である。Ｄ－ＯＥＭ_bridgeは、Ｄ－ＯＥＭ_clientと類似のアーキテクチャを持つ。しかしながら、前者は実行マネージャ及びプログラム領域を持たない。その代わり前者はネットワーク実行マネージャを有する。図１５は、Ｄ－ＯＥＭ_bridgeの基本アーキテクチャを示している。

【０１７４】

Ｄ－ＯＥＭ_bridgeは、クライアントコンピュータ２１２上で実行され、リモート装置２１０とクライアントコンピュータ２１２との間の通信を処理するクライアントマネージャを有している。またＤ－ＯＥＭ_bridgeは、タスク表ＴＴも有している。しかしながら、最初の二つの欄のみ、即ちオブジェクト名及びオブジェクトＩＤのみが、図６の表と同一である。第三欄は、サーバのネットワークアドレスである。このアドレスは、ＵＲＬ又は選択可能なポート番号を持つＩＰアドレスとすることができる。図１６は、Ｄ－ＯＥＭ_bridge部分のタスク表の例である。ネットワーク実行マネージャＮＥＭを使用して、リモート装置２１０とサーバ２１４、２１６（これより多くのサーバを置くこともできる）との間のプロセス又はスレッドの同期化をモニタリングする。ＮＥＭは、有線又は無線ネットワークを介してサーバ２１４、２１６に接続する。ＮＥＭの役割は、リモート装置２１０がリモート処理をサーバ２１４、２１６上で直接開始することができるように、リモート装置２１０を補助することである。ＮＥＭは、リモート装置２１０及びサーバ２１４、２１６用のデータ及びネットワークコマンドを制御、変換及び同期化する。実際のところ、ＮＥＭは、複数のサーバを使用する分散型演算処理を管理するソフトウェアエージェントである。

【０１７５】

＜Ｄ－ＯＥＭオブジェクトの初期化及びオブジェクトＩＤ＞

Ｄ－ＯＥＭオブジェクトの初期化は、図７Ａで示した方法と類似している。しかしながら、この場合、クライアントコンピュータ２１２は処理を全く行わない。クライアントコンピュータ２１２は、サーバ２１４、２１６にアクセスする方法を知っておく必要がある

。そこで、リモート装置２１０上のＤ－ＯＥＭが、実際のアドレス（ＵＲＬかＩＰアドレス）をクライアントコンピュータ２１２に通知することができる。以下のプロシージャは、Ｄ－ＯＥＭ_bridge及びＤ－ＯＥＭ_clientオブジェクトの開始方法を示している。

【０１７６】

（６．１）リモート装置２１０上のＤ－ＯＥＭが、ＵＲＬ／ＩＰアドレスと共に要求（Initialization_0f-0bject）をクライアントコンピュータ２１２に送信する。

【０１７７】

（６．２）Ｄ－ＯＥＭ_bridgeがこの要求を受信する。Ｄ－ＯＥＭ_bridgeは、この要求を、リモート装置２１０からのＵＲＬ／ＩＰアドレスによって、サーバ２１４、２１６に送信する。

【０１７８】

（６．３）サーバ２１４、２１６上のＤ－ＯＥＭ_clientが、全オブジェクト及びタスク表を初期化する。

【０１７９】

（６．４）サーバ２１４、２１６が、タスク表をＤ－ＯＥＭ_bridgeに返信する。

【０１８０】

（６．５）ＮＥＭが、オブジェクトＩＤにサーバＩＤを加える。

【０１８１】

（６．６）Ｄ－ＯＥＭ_bridgeが、オブジェクトＩＤ及びＵＲＬ／ＩＰアドレスをローカルタスク表ＴＴ中に記録する。

【０１８２】

（６．７）Ｄ－ＯＥＭ_bridgeが、オブジェクトＩＤをリモート装置２１０に送信して前述したようなＤ－ＯＥＭの初期化を行う。

【０１８３】

プロシージャ（６．５）において、オブジェクトＩＤは異なるサーバから作成することができる。各サーバは、図１２Ａに示したプロシージャに従って、全ＩＤを作成する。異なる二つのオブジェクトが異なるサーバで初期化される場合、異なるオブジェクトが同一オブジェクトＩＤを有することも可能である。前出の例において、８ビットのサーバＩＤフィールドが、32ビットのオブジェクトＩＤ内にリザーブされている。図１５では、ネットワーク実行マネージャＮＥＭがサーバＩＤカウンタ（ＳＩＣ）を有している。このカウンタＳＩＣを使用して、サーバ毎に固有サーバＩＤを作成する。以下のプロシージャは、新しいロケーションに新しいサーバＩＤを割り当てる方法を示している。

【０１８４】

（７．１）ＮＥＭが、Ｄ－ＯＥＭ_bridgeからInitialization_0f-0bjectコマンド要求を受信する。

【０１８５】

（７．２）ＮＥＭは、予め割り当てられたサーバＩＤがあるロケーションを調べる。

【０１８６】

（７．３）ＮＥＭがサーバのアドレスに最近アクセスしていれば、ＮＥＭは、ＮＥＭキャッシュメモリにある古いサーバＩＤを使用する。

【０１８７】

（７．４）ＮＥＭが分散実行用に要求されたロケーションに最近アクセスしていなければ、ＮＥＭは、新しいＩＤをサーバＩＤカウンタから入手する。ＮＥＭはカウンタを「１」増分させ、次のサーバＩＤとする。

【０１８８】

（７．５）ＮＥＭは、サーバＩＤをＤ－ＯＥＭ_bridgeに返信し、新しいオブジェクトＩＤを計算する。

【０１８９】

新しいオブジェクトＩＤは、サーバからのオブジェクトＩＤとＮＥＭからのサーバＩＤとの組合せである。図１７は、組み合わされたオブジェクトＩＤのフォーマットを示して

いる。各オブジェクトＩＤは、関数を誤って呼び出すことを避けるため、タスク表内で固有の番号を有している。ＮＥＭは、終了したオブジェクトからリリースされたサーバＩＤを検索する。ＮＥＭは、任意の使用済みサーバＩＤを再使用しうるが、古いオブジェクトへのレファレンスは失われる。８ビットのサーバＩＤの場合、最大可能サーバ数は２５６である。

【０１９０】

＜実行、同期化及び戻りパラメータ＞

オブジェクト及び関数をリモート実行する方法は、上記方法に類似しているが、クライアントコンピュータ２１２をブリッジとして機能させている。リモート処理に係る全ての要求は、サーバ２１４、２１６へ直接送られる。Ｄ－ＯＥＭ_bridgeは、タスク表ＴＴからＵＲＬ／ＩＰアドレスを探し、コマンド及び全パラメータをサーバ２１４、２１６へ送信する。サーバ２１４、２１６は、「リモート関数の実行及び同期化」の見出しで説明した前記方法に従い、オブジェクト／関数の処理及び同期化を行う。サーバＩＤは、ブリッジが異なるサーバを区別する際に用いられるＩＤであるため、サーバ２１４、２１６はサーバＩＤを無視する。サーバがオブジェクトＩＤを受信すると、オブジェクトＩＤを32ビットから元の24ビットにする場合、式６に示すように論理積演算（ＡＮＤ演算）が実行される。このオブジェクトＩＤを使用して、サーバはタスク表ＴＴから関数のエントリポイントを探す。続いて、サーバ２１４、２１６は、上記関数を実行する。

サーバオブジェクトID = (D-OEM_Bridgeオブジェクト ID) AND*0x00FFFFFF

・・・（６）

（AND* : 論理積演算）

同期化の場合、リモート装置２１０は、処理サーバからの応答を待つ。サーバ２１４、２１６のいずれかが操作を終了すると、サーバは戻りパラメータをＤ－ＯＥＭ_bridgeへ送り返す。Ｄ－ＯＥＭ_bridgeが戻りパラメータを受け取ると、Ｄ－ＯＥＭ_bridgeは、各サーバからの戻りパラメータをリモート装置２１０に通知する。実際、Ｄ－ＯＥＭ_bridgeは、リモート装置２１０とサーバ２１４、２１６との間で全情報を送受信する通信ブリッジとして機能する。従って、リモート装置２１０は複数サーバとの直接通信を仮想的に指令できる。Ｄ－ＯＥＭ_bridgeは、複数サーバにアクセスするプロシージャを処理する。

【０１９１】

戻りパラメータフォーマットは、図１０に示すものと同一である。しかしながら、唯一の変更は、オブジェクトＩＤ１０２である。Ｄ－ＯＥＭ_bridgeが戻りパラメータストリングをサーバ２１４、２１６から受け取ると、Ｄ－ＯＥＭ_bridgeはオブジェクトＩＤ１０２を抽出する。オブジェクトＩＤ１０２内のサーバＩＤフィールドは空いているので、このサーバ接続に既に割り当てられたサーバＩＤをオブジェクトＩＤ１０２に加える。更新されたオブジェクトＩＤ１０２は、戻りパラメータストリングに戻されて、リモート装置２１０へ送り返される。リモート装置２１０は、既にスタックに保存されている関数戻りアドレスを使用して、目的のメモリロケーションに対し戻り値を保存する。

【０１９２】

＜ネットワーク実行マネージャ（ＮＥＭ）＞

事実上ソフトウェアエージェントであるＮＥＭのアーキテクチャを、以下に説明する。このエージェントの役割は、サーバ２１４、２１６において、リモート装置２１０のために、リモート実行を開始することである。ＮＥＭにより以下のタスクが実行される。

【０１９３】

（１）通信コントローラとしてリモート装置２１０とサーバ２１４、２１６とを接続する。

【０１９４】

（２）リモート装置２１０とサーバ２１４、２１６との間の通信プロトコルを変換する。

【０１９５】

（３）ネットワーク例外を処理する。

【０１９６】

（４）複数サーバのサーバＩＤを管理する。

【０１９７】

（５）ネットワーク渋滞を最小化するための要求をバッファリングする。

【０１９８】

（６）不正アクセスを防止するファイアウォールとして機能する。

【０１９９】

図１８は、ＮＥＭの基本構造を示している。この基本構造にはＤ－ＯＥＭ_bridge３０２からのコマンドを処理するコマンドコントローラ３００が含まれている。プロトコルトランスレータ３０４は、一のネットワークから他のネットワークへのデータ及びコマンドの変換を管理する。例えば、スマートカードはIS07816-4フォーマットを使用している。ネットワークへの要求を入力する前に、プロトコルはIS07816-4からインターネット接続用のTCP/IPへ変換される。サーバマネージャ３０６は、サーバの接続とサーバＩＤとを管理する。ネットワークコントローラ３０８は、ネットワークへの接続を管理する。またネットワーク渋滞を最小化するローカルネットワークバッファも基本構造に含まれている。ファイアウォール３１０は、不正接続を防止するために使用される。加えて、ネットワーク例外コントローラ３１２は、ネットワークエラーに対処するために使用され、ネットワーク例外が発生したことをコマンドコントローラ３００へ直接報告する。

【０２００】

＜分散型オブジェクト実行マネージャ（Ｄ－ＯＥＭ）＞

Ｄ－ＯＥＭは、リモート装置２１０のリモート実行要求を管理する。リモート装置２１０は、負荷分担要求をローカルクライアント／サーバへ送信する。クライアントコンピュータ２１２がオブジェクト／関数を処理できる場合、Ｄ－ＯＥＭ_clientは、実行マネージャによってローカルに関数を実行する。そうでない場合、Ｄ－ＯＥＭ_bridgeが、要求をサーバ２１４、２１６へ送り、サーバ２１４、２１６側で関数が実行される。図１９は、Ｄ－ＯＥＭの基本構造を示す。Ｄ－ＯＥＭは、単一エントリポイントと共にカーネル４１４を有している。それはＤ－ＯＥＭオブジェクト内の全てのリソースにアクセスする権限を有している。Ｄ－ＯＥＭは、オブジェクト内の全ての要素を初期化する、オブジェクトのコンストラクタ４１６を有している。コマンドエンコーダ・デコーダ４１８は、上記Ｄ－ＯＥＭコマンドを合成する。ネットワークインタフェース４２０は、外部パーティとの通信を確立するための、外部ネットワークオブジェクトと結合するソフトウェアインタフェースである。同期化マネージャ４２２は、全ての同期化要求を管理する。例外ハンドラ４２４は、ローカルＤ－ＯＥＭ例外又はリモート関数からの例外の全てを管理する。スタックコントローラ４２６は、スタック及び関数の戻りパラメータを管理する。オブジェクト・関数疑似エントリポイント領域４２８は、各関数／オブジェクトに対しローカル関数が仮想呼出を実行できるリモート処理（リモートプロセス）の記述子である。これにより、ローカル関数は、ローカル関数を呼び出すのと同じくらい容易に、任意のリモート関数を呼び出すことができる。カーネル４１４は、実際のリモート処理の呼出を行う。タスクマネージャ４３０は、リモートオブジェクト及び関数の初期化を制御する。またタスクマネージャ４３０は、タスク表ＴＴ２を管理する。

【０２０１】

Ｄ－ＯＥＭ_clientの場合は、図２０に示すように、Ｄ－ＯＥＭの場合とは多少異なる。

【０２０２】

図２０には、Ｄ－ＯＥＭ_clientの基本構造が示されている。それはＤ－ＯＥＭに類似しているが、全ての戻りパラメータは、リモート装置２１０へ直接送り返される。パラメータは、Ｄ－ＯＥＭ_client内に格納する必要はないので、スタックコントローラ４２６は不要である。実行マネージャＥＭは、関数及びスレッドの実行を制御する。ネットワークインタフェース５２０は、リモート装置２１０から要求コマンドを受け取る。全てのコ

マンドは、処理を行う（プロセスを実行する）ためにカーネル５１４へ送られる。オブジェクトの初期化は、このシステムによって行われる。他の要素は、Ｄ－ＯＥＭの場合と機能的に同一である。実行マネージャＥＭは、ローカルで実行する必要があるとして要求されたオブジェクト又は関数を管理する。Ｄ－ＯＥＭ_clientが実行コマンドを解読し、プロセスエントリポイントを取得すると、このエントリポイントは処理（プロセス）を実行するために実行マネージャＥＭに送られる。実行マネージャＥＭの詳細は後述する。

【０２０３】

Ｄ－ＯＥＭ_bridgeの場合、その役割は、サーバ２１４、２１６とリモート装置２１０との間でコマンド及びパラメータを送受信することである。その構造は、Ｄ－ＯＥＭやＤ－ＯＥＭ_clientよりも単純である。

【０２０４】

図２１は、Ｄ－ＯＥＭ_bridgeの構造図である。Ｄ－ＯＥＭ_bridgeは、リモート装置２１０とサーバ２１４、２１６との間の通信ブリッジとして機能する。カーネル６１４は、オブジェクト内の全ての要素を制御し、初期化する。タスクマネージャ６３０は、タスク表ＴＴ２を管理する。システムバッファ６３２は、カーネル６１４が後に処理する未終了タスクを格納する。ＮＥＭブリッジ６３４は、ＮＥＭに対するソフトウェアインタフェースである。ＮＥＭは、上記の如く高レベルのネットワーク管理を制御する。

【０２０５】

実行マネージャＥＭは、スレッド及び関数の実行を制御するために使用される。図２２は、実行マネージャの構造を示している。要求された処理（プロセス）がスレッドであれば、スレッドコントローラ７４０が、全ての同期化プロシージャを管理する。要求された処理（プロセス）が関数／オブジェクトであれば、関数エクセキューショナー７４２が処理の実行を開始し、モニタリングする。関数又はスレッドに由来する何らかの例外があれば、例外ハンドラ７４４が、処理の実行を担当する。このハンドラ７４４は、他の並行する処理に対するエラーの影響を最小化するために、問題のあるプロシージャを停止するよう動作する。

【図面の簡単な説明】

【０２０６】

【図１】認証システムの模式図である。図１Ａは、スマートカードの模式図である。

【図２】クライアント端末とスマートカードとで分担して照合を実行するデータフロー図である。

【図３】商取引を実行するためのメッセージの流れを模式に示すブロック図である。

【図４】セキュリティアクセスチェックキーのフォーマットを示す図である。

【図５】マルチ処理ユニットの簡略化アーキテクチャ図である。

【図６】タスク表を示す図である。

【図７】分散処理の簡略化アーキテクチャ図である。

【図７Ａ】タスク表を初期化するプロシージャを示すフロー図である。

【図８】Initialization_0f_0bjectコマンドのフォーマットを示す図である。

【図９】コマンドストリングのフォーマットを示す図である。

【図１０】戻りパラメータのフォーマットを示す図である。

【図１１】スタックのフォーマットを示す図である。

【図１２】単一クライアントプロセッサと複数リモート装置を示す模式図である。

【図１２Ａ】ソフトウェアオブジェクト及びそのメンバ関数を初期化するプロシージャを示すフロー図である。

【図１３】オブジェクト及び関数ＩＤ用の暗号化フォーマットを示す図である。

【図１３Ａ】指紋照合演算における負荷分担のプロシージャを示すフロー図である。

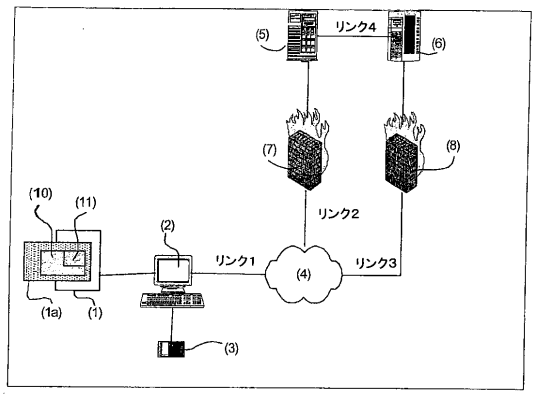
【図１４】複数の処理装置を備えたシステムを示す図である。

【図１５】分散型オブジェクト実行マネージャブリッジのアーキテクチャを示す図である。

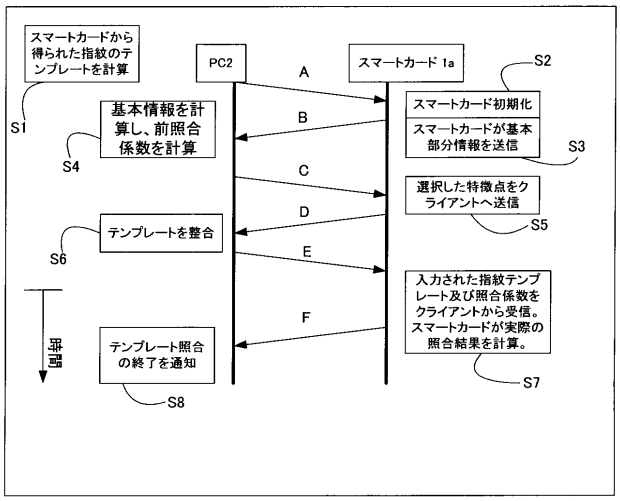
【図１６】分散型オブジェクト実行マネージャブリッジにおけるタスク表の一例を示す図

である。
【図１７】複数サーバ環境における一のオブジェクト識別子のフォーマットを示す図である。
【図１８】ネットワーク実行マネージャの模式的ブロック図である。
【図１９】リモート装置における分散型オブジェクト実行マネージャの模式的ブロック図である。
【図２０】クライアント端末における分散型オブジェクト実行マネージャの模式的ブロック図である。
【図２１】ブリッジにおける分散型オブジェクト実行マネージャの模式的ブロック図である。
【図２２】実行マネージャの模式的ブロック図である。

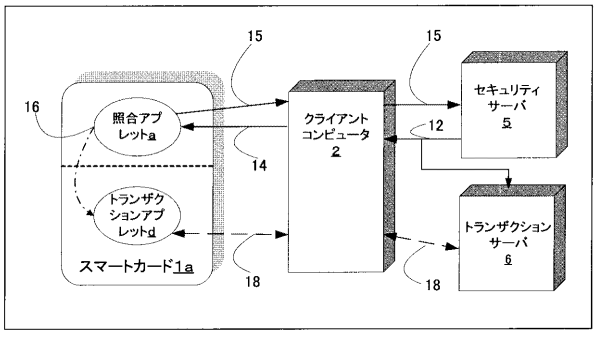
【図１】



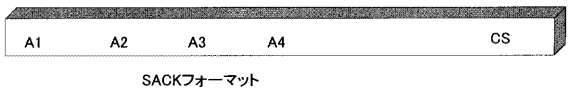
【図２】



【図３】

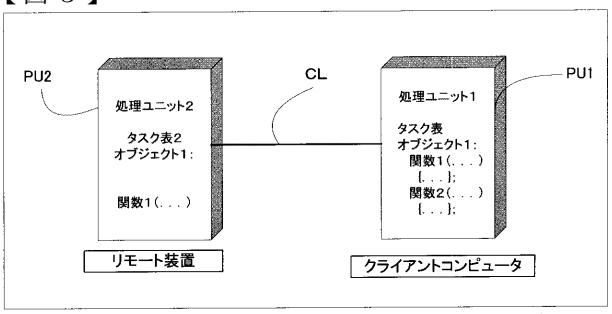


【図４】



SACKフォーマット

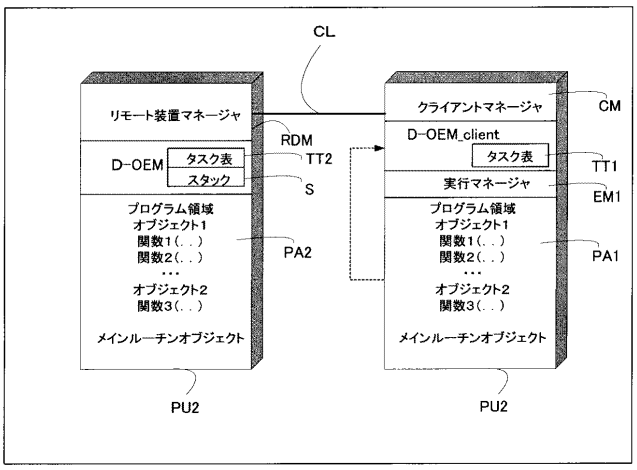
【図５】



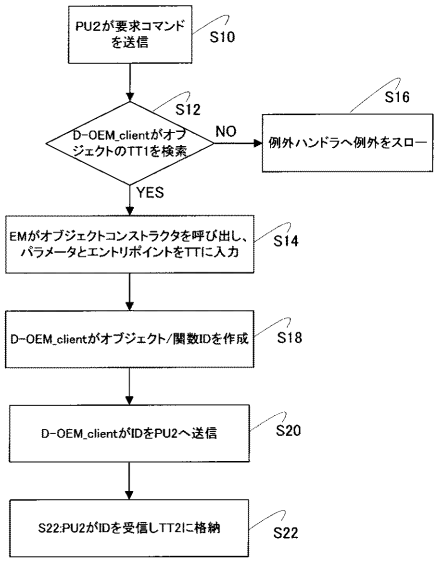
【図６】

オブジェクト名	オブジェクトID	パラメータ	エントリポイント
オブジェクト1	オブジェクトID	void	*オブジェクト1
関数1	関数1　ID	(int)P1, (int)P2	*関数1（..）
関数2	関数2　ID	(float)P1	*関数2（..）
関数3	関数3　ID	void	*関数3（..）
関数4	関数4　ID	(bool)P1, (bool)P2	*関数4（..）
・・・	・・・	・・・	・・・
関数N	関数N　ID	void	*関数N（..）
オブジェクト2	オブジェクト2　ID	(byte)P1	*オブジェクト2
関数5	関数5　ID	void	*関数5（..）
関数6	関数6　ID	void	*関数6（..）
関数7	関数7　ID	void	*関数7（..）
関数8	関数8　ID	void	*関数8（..）
・・・	・・・	・・・	・・・
関数N	関数N　ID	void	*関数N（..）
EOT	EOT	EOT	EOT

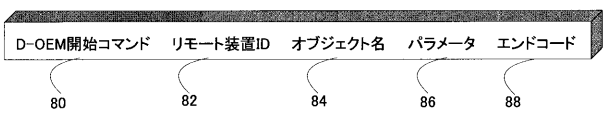
【図７】



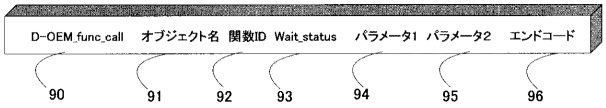
【図７Ａ】



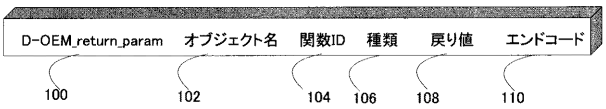
【図８】



【図９】



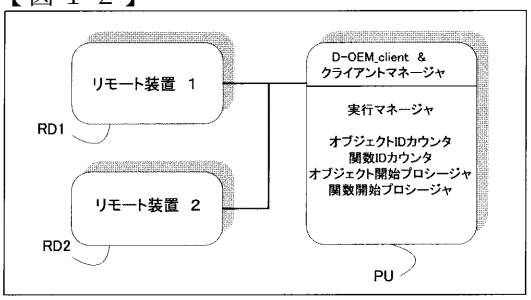
【図１０】



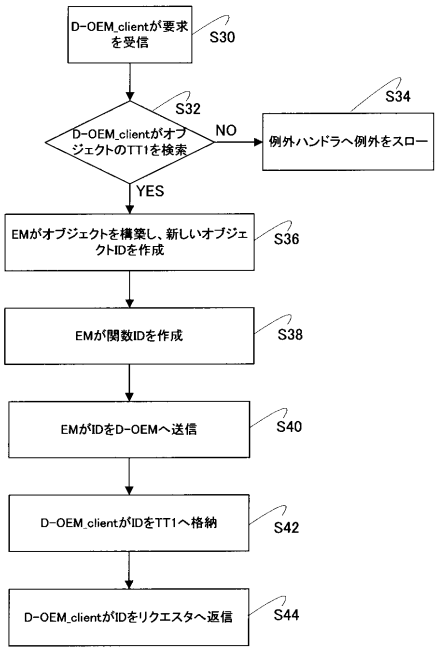
【図１１】

オブジェクトID	関数ID	リターンオブジェクトのレファレンス
OID 1	FID1	(int*)result
OID2	FID2	(float*)result2
OID3	FID3	(double*)coef
・・・	・・・	・・・
・・・	・・・	・・・
OIDN	FIDN	(int**)array

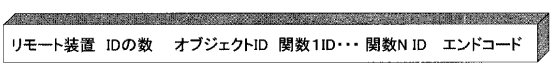
【図１２】



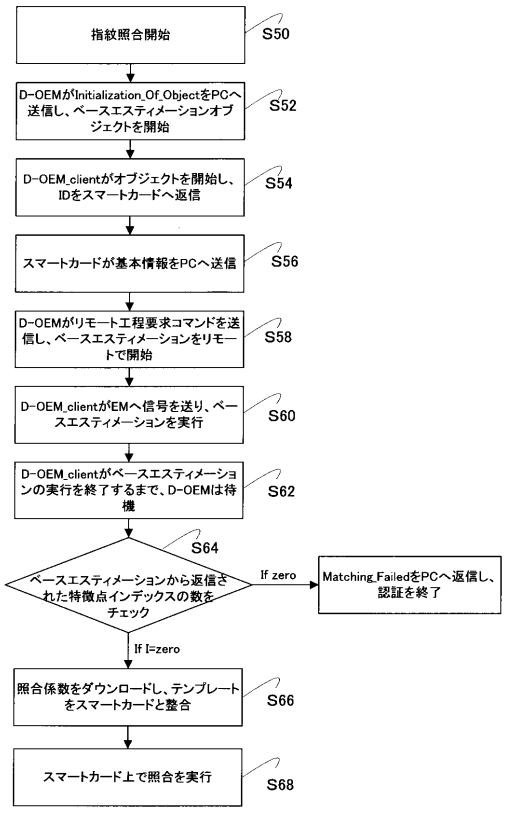
【図１２Ａ】



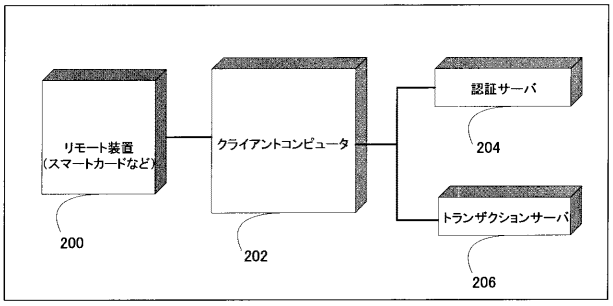
【図１３】



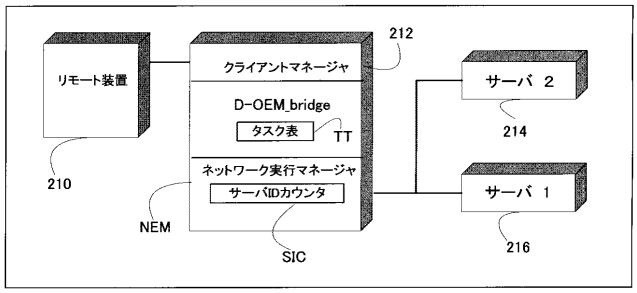
【図１３Ａ】



【図１４】



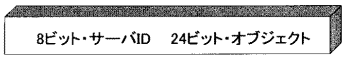
【図１５】



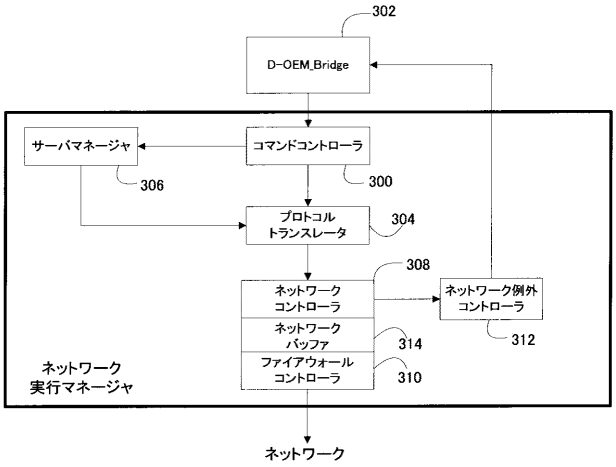
【図１６】

オブジェクト名	オブジェクトID	アドレス
オブジェクト1	オブジェクト1　ID	www.csp.ntu.edu.sg:1080
関数1	関数1　ID	www.csp.ntu.edu.sg:1080
関数2	関数2　ID	www.csp.ntu.edu.sg:1080
関数3	関数3　ID	www.csp.ntu.edu.sg:1080
関数4	関数4　ID	www.csp.ntu.edu.sg:1080
オブジェクト2	オブジェクト2　ID	131.120.12.1:70
関数1	関数1　ID	131.120.12.1:70
関数2	関数2　ID	131.120.12.1:70

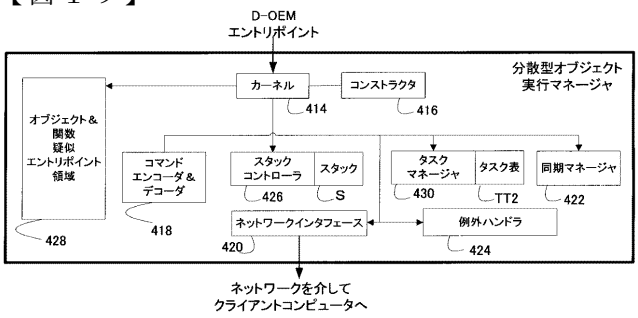
【図１７】



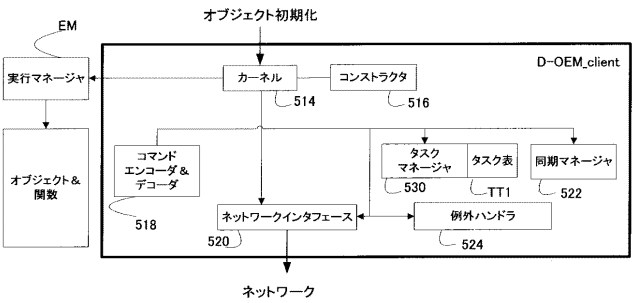
【図１８】



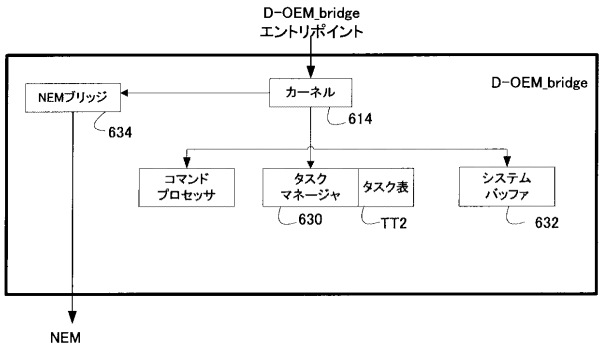
【図１９】



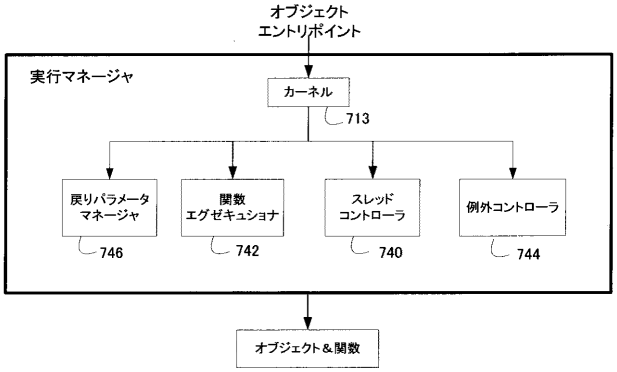
【図２０】



【図２１】



【図２２】


オブジェクト
エントリポイント
実行マネージャ
カーネル
713
戻りパラメータ
マネージャ
746
関数
エグゼキュショナ
742
スレッド
コントローラ
740
例外コントローラ
744
オブジェクト＆関数

【国際調査報告】

INTERNATIONAL SEARCH REPORT

PCT/SG 02/00199

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

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)
IPC 7 G06F

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 practical, 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.
A	WO 01 11577 A (PRECISE BIOMETRICS AB ;WIEBE LINUS (SE)) 15 February 2001 (2001-02-15) the whole document	1-37
A	US 6 219 439 B1 (BURGER PAUL M) 17 April 2001 (2001-04-17) the whole document	1-37
A	GB 2 331 825 A (NIPPON ELECTRIC CO) 2 June 1999 (1999-06-02) the whole document	1-37
A	US 6 003 135 A (BIALICK WILLIAM P ET AL) 14 December 1999 (1999-12-14) the whole document	1-37

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in 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 document 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
9 May 2003	16/05/2003

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, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Harms, C

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

PCT/SG 02/00199

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 0111577	A	15-02-2001	SE	518419 C2	08-10-2002
			AU	6194600 A	05-03-2001
			CA	2382042 A1	15-02-2001
			CN	1369085 T	11-09-2002
			EP	1210695 A1	05-06-2002
			WO	0111577 A1	15-02-2001
			SE	9902846 A	07-02-2001
			US	2002030359 A1	14-03-2002
US 6219439	B1	17-04-2001	NONE		
GB 2331825	A	02-06-1999	JP	2950307 B2	20-09-1999
			JP	11161793 A	18-06-1999
			AU	736113 B2	26-07-2001
			AU	9422298 A	17-06-1999
			CN	1221160 A	30-06-1999
US 6003135	A	14-12-1999	AU	7709498 A	21-12-1998
			WO	9855912 A1	10-12-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

（特許庁注：以下のものは登録商標）
イーサネット
ＪＡＶＡ

(72)発明者　チェン，タイ，パング
シンガポール国，６３７７２２　シンガポール，ナンヤン　ドライヴ　１６，ユニット　２１３，ビーエルケイ　１，シー／オウ　イノベーション　アンド　テクノロジー　トランスファー　オフィス
(72)発明者　ヤウ，ウェイ，ユン
シンガポール国，６３７７２２　シンガポール，ナンヤン　ドライヴ　１６，ユニット　２１３，ビーエルケイ　１，シー／オウ　イノベーション　アンド　テクノロジー　トランスファー　オフィス

Ｆターム(参考)　5B035 AA14 BB09 BC01 CA11
5B058 CA01 KA31 KA33 KA38 YA02 YA03
5B285 AA01 BA03 CA41 CA47 CB07 CB14 CB15 CB16 CB24 CB64
CB72 CB74 CB75 CB85
5J104 KA17 PA07 PA10