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

特許第5735539号
(P5735539)

(45)発行日　平成27年6月17日(2015.6.17)　(24)登録日　平成27年4月24日(2015.4.24)

(51)Int.Cl.			F I		
H04L	9/36	(2006.01)	H04L	9/00	685
G09C	1/00	(2006.01)	G09C	1/00	660E
G06F	13/00	(2006.01)	G06F	13/00	520C

請求項の数 16　(全 33 頁)

(21)出願番号	特願2012-546558 (P2012-546558)
(86)(22)出願日	平成22年12月30日 (2010.12.30)
(65)公表番号	特表2013-516642 (P2013-516642A)
(43)公表日	平成25年5月13日 (2013.5.13)
(86)国際出願番号	PCT/IL2010/001097
(87)国際公開番号	WO2011/080745
(87)国際公開日	平成23年7月7日 (2011.7.7)
審査請求日	平成25年12月27日 (2013.12.27)
(31)優先権主張番号	61/306,207
(32)優先日	平成22年2月19日 (2010.2.19)
(33)優先権主張国	米国(US)
(31)優先権主張番号	61/291,398
(32)優先日	平成21年12月31日 (2009.12.31)
(33)優先権主張国	米国(US)

(73)特許権者　512172017
バウルティブ　リミテッド
ＶＡＵＬＴＩＶＥ　ＬＴＤ.
イスラエル国　６５１６２　テル　アビブ
ナチャラト　ビンヤミン　ストリート
４１　ファースト　フロア
(74)代理人　100068755
弁理士　恩田　博宣
(74)代理人　100105957
弁理士　恩田　誠
(74)代理人　100142907
弁理士　本田　淳
(72)発明者　マツケル、ベン
イスラエル国　５３２３４　ギバタイム
ハメオレア　ストリート　２１
最終頁に続く

(54)【発明の名称】ネットワークを介して送信されるデータの暗号化および復号化システム、装置、および方法

(57)【特許請求の範囲】
【請求項１】
方法であって、
クライアント装置からの入力テキストを中間モジュールで受信すること、
前記入力テキストを前記中間モジュールで処理して処理済みテキストを取得することであって、前記処理済みテキストにベイトを含めることを含む前記取得すること、
前記中間モジュールによって前記処理済みテキストをサーバに送信することであって、前記サーバが、複数の変換のうちの少なくとも１つを適用することによって、前記クライアント装置から受信したテキストを変換するように構成される、前記処理済みテキストを前記サーバに送信すること、
前記クライアント装置から要求があった場合に、前記サーバが前記複数の変換のうちの少なくとも１つを前記処理済みテキストに適用して取得した変換・処理済みテキストを前記サーバから前記中間モジュールで受信すること、
前記処理済みテキストと前記変換・処理済みテキストとの比較に基づいて、前記サーバが適用する前記変換のうちの少なくとも１つを前記中間モジュールにより決定することを備え、
前記比較は、前記処理済みテキストのベイトと前記変換・処理済みテキストのベイトとの間の比較を含む、方法。
【請求項２】
前記中間モジュールによって、前記処理済みテキストに逆処理を適用して未処理入力テ

キストを取得すること、
前記中間モジュールによって、前記決定した少なくとも１つの変換に基づいて前記未処理入力テキストを修正することをさらに備える、請求項１に記載の方法。
【請求項３】
前記中間モジュールによって、前記修正した未処理入力テキストを前記クライアント装置に送信することをさらに備える、請求項２に記載の方法。
【請求項４】
前記複数の変換のうちの少なくとも１つの変換が、前記処理済みテキストの少なくとも１つの変換可能な文字記号の対応する置換文字記号または置換文字記号列への置換を含み、
前記処理済みテキストにベイトを含めることは、前記処理済みテキストに前記少なくとも１つの変換可能な文字記号を含めることを含む、請求項１に記載の方法。
【請求項５】
前記中間モジュールによって、前記処理済みテキストに逆処理を適用して未処理入力テキストを取得すること、
前記中間モジュールによって、前記未処理入力テキストの少なくとも１つの変換可能な文字記号を前記対応する置換文字記号または置換文字記号列に置換することによって前記未処理入力テキストを修正することをさらに備える、請求項４に記載の方法。
【請求項６】
前記中間モジュールによって、前記修正した未処理入力テキストを前記クライアント装置に送信することをさらに備える、請求項５に記載の方法。
【請求項７】
前記複数の変換のうちの少なくとも１つの変換が、前記処理済みテキストにおけるＨＴＭＬタグの省略を含み、
前記処理済みテキストにベイトを含めることが、前記処理済みテキストにＨＴＭＬタグを含めることを含む、請求項１に記載の方法。
【請求項８】
前記中間モジュールによって、前記処理済みテキストに逆処理を適用して未処理入力テキストを取得すること、
前記中間モジュールによって、前記未処理入力テキストに含まれるＨＴＭＬタグを省略することによって前記未処理入力テキストを修正すること、
前記中間モジュールによって、前記修正した未処理入力テキストを前記クライアント装置に送信することをさらに備える、請求項７に記載の方法。
【請求項９】
クライアント装置とサーバとの間で送信されるデータを保護するシステムであって、前記サーバが、複数の変換のうちの少なくとも１つを適用することによって、前記クライアント装置から受信したテキストを変換するように構成されたシステムにおいて、
メモリと、
制御部であって、
入力テキストを受信し、
前記入力テキストを処理し、ベイトを含めることによって処理済みテキストを取得し、
前記処理済みテキストを前記サーバに送信し、
前記クライアント装置から要求があった場合に、前記サーバが前記複数の変換のうちの少なくとも１つを前記処理済みテキストに適用して取得した変換・処理済みテキストを前記サーバから受信し、
前記処理済みテキストと前記変換・処理済みテキストとの比較に基づいて、前記サーバが適用する前記変換のうちの少なくとも１つを決定するように構成された前記制御部を備え、
前記比較は、前記処理済みテキストのベイトと前記変換・処理済みテキストのベイトとの間の比較を含む、システム。

【請求項１０】
前記制御部がさらに、
前記処理済みテキストに逆処理を適用して未処理入力テキストを取得し、
前記決定した少なくとも１つの変換に基づいて前記未処理入力テキストを修正するように構成された、請求項９に記載のシステム。
【請求項１１】
前記制御部がさらに、前記修正した未処理入力テキストを前記クライアント装置に送信するように構成された、請求項１０に記載のシステム。
【請求項１２】
前記複数の変換のうちの少なくとも１つの変換が、前記処理済みテキストの少なくとも１つの変換可能な文字記号の対応する置換文字記号または置換文字記号列への置換を含み、
前記制御部が、前記処理済みテキストに前記少なくとも１つの変換可能な文字記号を含めることにより、前記入力テキストを処理して前記処理済みテキストを取得するように構成された、請求項９に記載のシステム。
【請求項１３】
前記制御部がさらに、
前記処理済みテキストに逆処理を適用して未処理入力テキストを取得し、
前記未処理入力テキストの少なくとも１つの変換可能な文字記号を前記対応する置換文字記号または置換文字記号列に置換することによって前記未処理入力テキストを修正するように構成された、請求項１２に記載のシステム。
【請求項１４】
前記制御部がさらに、前記修正した未処理入力テキストを前記クライアント装置に送信するように構成された、請求項１３に記載のシステム。
【請求項１５】
前記複数の変換のうちの少なくとも１つの変換が、前記処理済みテキストにおけるＨＴＭＬタグの省略を含み、
前記制御部が、前記処理済みテキストにＨＴＭＬタグを含めることにより、前記入力テキストを処理して前記処理済みテキストを取得するように構成された、請求項９に記載のシステム。
【請求項１６】
前記制御部がさらに、
前記処理済みテキストに逆処理を適用して未処理入力テキストを取得し、
前記未処理入力テキストに含まれるＨＴＭＬタグを省略することによって前記未処理入力テキストを修正し、
前記修正した未処理入力テキストを前記クライアント装置に送信するように構成された、請求項１５に記載のシステム。
【発明の詳細な説明】
【背景技術】
【０００１】
企業や組織は、インターネットおよびワールドワイドウェブにより、デジタル形式のウェブアプリケーション等のドキュメントで企業や個人にサービスを提供することができ、企業や個人は、パソコンおよびウェブブラウザを用いてこれらサービスへのアクセスおよび利用が可能である。ネットワークを介して入手できるこのようなドキュメント、特にアプリケーションの作成は通常、サービス型ソフトウェア（ＳａａＳ：Ｓｏｆｔｗａｒｅ　ａｓ　ａ　Ｓｅｒｖｉｃｅ）と称する。ＳａａＳ形式で提供されるアプリケーションの例としては、電子メール、インスタントメッセージ、生産性ツール、顧客関係管理（ＣＲＭ）、企業資源計画（ＥＲＰ）、人的資源アプリケーション、ブログ、ソーシャルネットワーキングサイト等が挙げられる。
【０００２】

ただし、このモデルには本質的に、セキュリティ上のリスクが存在する。メッセージ、顧客記録、および企業財務等のユーザデータは、リモートサーバに格納されるため、ユーザデータのデータ提供者が管理できなくなる。個人情報や企業情報がリモートサーバに格納されると、データ所有者は多くのリスクに曝される。このことは、情報をホスティングするコンピュータシステムおよび情報所有者とホスティングシステムとを接続するネットワークを所有する事業体を情報所有者が信頼する必要があることを暗に示している。

【０００３】

たとえば、周知の会計ソフトウェアソリューションでは、ソリューションプロバイダのサーバに格納する会計情報を顧客がポスティングする必要がある。このようなシステムでは、顧客がソリューションプロバイダに会計情報を委ねるため、そのプライバシーおよびインテグリティに対する一定の管理を放棄することが必要となる。

【０００４】

特定のソフトウェアアプリケーションにおいては、然るべき復号化法または復号化鍵を持たない誰に対してもデータが解読不能となるように、様々な暗号化法が用いられている。たとえば、情報所有者には、アプリケーションプロバイダによって、セキュアソケットレイヤー（secure service socket:ＳＳＬ）暗号化または別の方法を用いたクライアントとホスト間の送受信データの暗号化が可能となる場合および要求される場合のうちの少なくとも一方がある。これにより、インターネットサービスプロバイダ（internet service provider:ＩＳＰ）およびその他の潜在的な盗聴者による送受信中のデータの閲覧を阻止することができる。したがって、データは、ホスティングされたアプリケーションへの到着時に復号化され、そのホスティングされたアプリケーションのベンダーは、所有者の非暗号化データの閲覧および操作が可能となる。ただし、この方法では、ホスティングされたアプリケーションのベンダーに機密データが曝される。

【０００５】

特許文献１は、クライアントとサーバ間でネットワークを介して送信されたデータの一部を選択的に暗号化する装置および方法を記載している。この装置は、データの第１の部分をデータの第２の部分から分離する解析手段と、データの第１の部分のみを暗号化する暗号化手段と、暗号化されたデータの第１の部分をデータの第２の部分と結合する結合手段とを備えている。また、この装置は、クライアントに組み込まれ、データの暗号化部分を復号化する復号化手段をさらに備えている。

【０００６】

特許文献２は、ダウンロードされたソフトウェアオブジェクトによるコンピュータネットワーク暗号化の改良を開示している。この出願は、ウェブサーバコンピュータとリモートクライアントコンピュータとを接続するワールドワイドウェブ等の公衆ネットワーク上の送受信信号に含まれる財務等の極秘データを保護する方法およびシステムを記載している。ウェブサーバとクライアント間のすべての機密通信に用いる所望の（通常は強固な）個別暗号化規格を決定し、ウェブサーバからクライアントに自動でダウンロードすることにより当該規格に応じて暗号化する機能をクライアントに「プッシュ」し、クライアントのウェブブラウザでソフトウェアオブジェクトを実行することにより選択した規格に準じて暗号化／復号化タスクを実行することによって、クライアントが元々はそのような強固な暗号化機能を有していなくても強固な暗号化が容易に保証される。

【０００７】

これらの手法をホスティングされたＳａａＳアプリケーションに適用する際の問題点として、このようなアプリケーションにおいては、ネットワークを介して操作可能とされたデータ等の運用情報を非暗号化することによってアプリケーションプロバイダによる情報操作を可能とする必要があるため、アプリケーションプロバイダにデータが曝されることになる。あるいは、セキュリティ上の利害関係者に対して、操作中にデータが脆弱となってしまう。

【先行技術文献】

【特許文献】

【０００８】
【特許文献１】米国特許第７，１６５，１７５号明細書
【特許文献２】国際公開第０１／０４７２０５号
【図面の簡単な説明】
【０００９】
【図１】本発明の一実施形態に係る、中間モジュールおよびその周囲を含むシステムを示した図である。
【図２】本発明の一実施形態に係る、クライアント端末からネットワークノードへのデータフローを示した図である。
【図３】本発明の一実施形態に係る、ネットワークノードからクライアント端末へのデータフローを示した図である。
【図４】本発明の一実施形態に係る、サーバ側での検索を可能にするデータの暗号化方法および暗号化データのインデキシング方法を示した図である。
【図５】正規化プロセスおよびセンテンスを含む入力テキストの一例を示した図である。
【図６】本発明の一実施形態に係る、単語の処理例を示した図である。
【図７】本発明の一実施形態に係る、暗号化データのサーバ側でのソートを可能にするデータの暗号化方法を示した図である。
【図８】本発明の一実施形態に係る、順序維持関数の生成方法を示した図である。
【図９】本発明の一実施形態に係る、３つの異なる鍵を用いて生成された３つの順序維持暗号化関数の一例を示した図である。
【図１０】本発明の一実施形態に係る、暗号化ユーザデータの検索を可能にするデータフローを模式的に示した図である。
【発明を実施するための形態】
【００１０】
本発明の上記およびその他の目的、特徴、および利点については、添付の図面とともに以下の詳細な説明を考慮することによって、より明らかとなるであろう。図面中、異なる図であっても、同様の要素には同一の符号を付与している。
【００１１】
以下の詳細な説明においては、本発明が十分に理解されるように、多くの具体的詳細を記載している。ただし、当業者には当然のことながら、本発明は、これらの具体的詳細なしに実行してもよい。他の例では、本発明が不明瞭になることのないように、周知の方法、手順、および構成要素については詳細に説明していない。
【００１２】
一般的なデータフロー
図１を参照して、この図は、本発明の一実施形態に係る、中間モジュール２００およびその周囲を含むシステムを示すとともに、ワークステーション２３０のクライアントモジュールからネットワークノード２６０のアプリケーションサービスプロバイダへのデータフローを示している。
【００１３】
中間モジュール２００は、阻止モジュール２１０およびデータ保護モジュール２２０を備えていてもよい。また、中間モジュール２００は、公衆ネットワーク２５０等のネットワークを介して、クライアント端末２３０（たとえば、トラステッド（信頼できる）ワークステーション等）およびネットワークノード２６０（たとえば、アプリケーションサービスプロバイダ等）と動作可能に接続されていてもよい。当然のことながら、図１は本発明の一実施形態を例示したものに過ぎず、その他のネットワーク構成も可能である。たとえば、トラステッドワークステーション２３０と中間モジュール２００とが互いにリモートな関係となって、トラステッドネットワークリンクを介して動作可能に接続されていてもよい。
【００１４】
たとえば、トラステッドワークステーション２３０は、複数の組織に対応する複数の中

間モジュールに接続され、公衆ネットワークを介して１または複数のアプリケーションサービスプロバイダとのデータトラフィックを仲介するようにしてもよい。

【００１５】

ただし、中間モジュールは、本出願の全体にわたって参照するに、クライアント装置上に存在していてもよく、たとえば、クライアント装置に関連する施設のゲートウェイサーバまたはトラステッドクライアント装置および非トラステッド（信頼できない）サーバと接続した１または複数の独立サーバに設けられていてもよい。

【００１６】

したがって、阻止モジュールおよびデータ保護モジュールのうちの少なくとも一方は、たとえばブラウザプラグイン、オペレーティングシステムドライバまたはモジュール、ソフトウェアライブラリ、または別のソフトウェアコンポーネントとしてトラステッドワークステーションに組み込まれていてもよい。

【００１７】

別の例として、中間モジュールは、非トラステッドアプリケーションの直前に配置して、当該非トラステッドアプリケーションへのすべてのアクセスが中間モジュールを通過するようにしてもよい。

【００１８】

さらに別の例として、中間モジュールは、クライアントモジュールから入力データが送信され、処理済みデータを非トラステッドサーバに送信する独立サーバであってもよい。

トラステッドワークステーション２３０は、中間モジュールと相互作用可能なクライアントコンポーネント２４０が組み込まれたクライアントコンピュータであってもよい。また、クライアントコンポーネント２４０は、ウェブブラウザで動作するウェブアプリケーションのＨＴＭＬ形式であってもよい。一方、ネットワークノード２６０は、ＳａａＳベンダーのＨＴＴＰウェブサーバであってもよい。クライアントコンポーネント２４０は、ＡＰＩクライアントソフトウェアを備えていてもよい。また、その追加または代替として、ネットワークノード２６０にリモートアクセスするその他任意の方法を備えていてもよい。

【００１９】

エンドユーザは、クライアントコンポーネント２４０を用いて、ネットワークノード２６０との間での受け渡しまたは読み出しを目的としたデータの入力、読み出し、および操作を行うことができる。エンドユーザとしては、ソフトウェアエージェント（たとえば、ウェブブラウザ等）を利用する人間およびクライアントＡＰＩを使用する自動化エージェント等が挙げられる。

【００２０】

中間モジュール２００の阻止モジュール２１０は、トラステッドワークステーション２３０からの（未処理）入力テキストを阻止するか、あるいは受信し、当該入力テキストをデータ保護モジュール２２０に供給して処理してもよい。阻止モジュール２１０は、クライアントコンポーネント２４０とネットワークノード２６０間を流れるデータを阻止してもよいし、その修正や通常のデータフローに対する割り込みも可能である。たとえば、阻止モジュールは、認証セッションを開始することにより、ネットワークノード２６０に格納されたデータにエンドユーザがアクセス可能であることを判定するようにしてもよい。阻止モジュール２１０は、ウェブプロキシサーバであってもよい（または、ウェブプロキシサーバにより実行してもよい）。

【００２１】

データ保護モジュール２２０は、入力テキストを受信して選択的に処理してもよい。選択処理されない入力テキストは、実質的に処理されずに、または選択処理されたテキストよりも少ない処理で未処理テキストとしてネットワークノード２６０に送信され、記憶システム２７０での操作および記憶のうちの少なくとも一方を行うようにしてもよい。処理対象のテキストについては、データ保護モジュール２２０が入力テキストを処理して処理済みテキストを供給するようにしてもよい。また、公衆ネットワーク２５０を介して非ト

ラステッドアプリケーションサービスプロバイダ２６０に供給し、記憶や操作等を行うようにしてもよい。したがって、本発明の実施形態によれば、アプリケーションサービスプロバイダ２６０が未処理テキストを受信するのではなく、処理済みテキストを記憶して操作するようにしてもよい。この処理には、以下に説明するように、検索およびソートのうちの少なくとも一方が可能な暗号化法の適用による暗号化テキストデータの供給が含まれる。本発明の実施形態によれば、この処理において、アプリケーションサービスプロバイダ２６０にいずれの入力テキストを処理済み形式で送信し、いずれの入力テキストを未処理形式で送信するかを選択することにより、テキストを選択的に暗号化するようにしてもよい。

【００２２】

当然のことながら、中間モジュール２００は、１または複数のサーバ、１または複数のワークステーション、１または複数のパソコン、１または複数のラップトップコンピュータ、１または複数のメディアプレーヤ、１または複数の携帯データ端末、１または複数の集積回路、および１または複数のプリント配線板、専用ハードウェアのうちの少なくともいずれか、またはそれらの組み合わせを備えていてもよい。

【００２３】

データフロー割り込み

中間モジュール２００は、暗号化および復号化のうちの少なくとも一方に対して追加的または無関係な機能を具備または提供してもよい。また、クライアントであるトラステッドワークステーション２３０とサーバである非トラステッドアプリケーション２６０間の通常のメッセージフローを変更してもよい。このような追加機能には、暗号化により失われるサーバ側の機能を補償する効果があってもよい。

【００２４】

本発明の実施形態によれば、中間モジュールは、クライアント装置から入力データを受信し、当該入力データのサーバへの送信を阻むか、または許可しない等により送信を阻止してもよい。また、中間モジュールは、サーバの代わりに、関連する機能を入力データに適用するようにしてもよい。たとえば、中間モジュールは、この機能の結果に基づいて、クライアント装置への少なくとも１つのメッセージを生成するようにしてもよい。

【００２５】

本発明の一部の実施形態によれば、中間モジュールは、上記少なくとも１つのメッセージに対する応答をクライアント装置から取得し、その応答に基づき、入力テキストを処理して処理済み入力テキストを取得し、当該処理済み入力テキストをサーバに送信するようにしてもよい。

【００２６】

たとえば、サーバは一般的に、入力テキストのスペルをチェックし、たとえばスペルの間違った単語や訂正箇所を提示するフィードバックメッセージをユーザに供給してもよい。ただし、サーバが受信したテキストが暗号化されている場合、本発明の実施形態によれば、サーバは、処理済みテキストの復号化なしでのスペルチェックは行えなくてもよい。したがって、本発明の実施形態によれば、中間モジュールは、スペルチェック等の追加機能を入力テキストに適用するとともに、当該入力データに対するスペルチェック機能の結果としてのエラーメッセージ、スペル訂正の提示、エラー非検出メッセージ等のフィードバックメッセージをユーザに供給するようにしてもよい。

【００２７】

本発明の一実施形態において、上記のような追加機能としては、たとえばユーザデータ（またはその一部）のコピーを格納し、クライアントからの検索要求に応じて中間モジュールを検索することによりサーバ側の検索機能を置き換えること等が挙げられる。

【００２８】

また、本発明の一実施形態において、上記のような追加機能としては、ユーザデータの暗号化および復号化が可能となる前にクライアントと中間モジュールとの間で認証セッションを開始すること等が挙げられる。

【００２９】
また、本発明の一実施形態において、上記のような追加機能としては、入力データの書式チェック等が挙げられる。また、必要に応じて、入力データが第１の書式である場合は、当該第１の書式と異なる第２の書式で情報を送るようにクライアントに要求すること等が挙げられる。そのような受信および要求書式のうちの少なくとも一方としては、たとえば（ａ）入力テキストの既知の型からの差異のみを送信するデルタ符号化書式、（ｂ）完全入力テキスト型、（ｃ）特定のドキュメント書式に含まれる入力テキスト、またはそれらの組み合わせ等が挙げられる。たとえば、入力データがデルタ符号化書式で受信され、中間モジュールが完全入力テキスト書式の入力データを要求するようにしてもよい。特定のドキュメント書式のその他の例としては、ＰＤＦ、ＤＯＣ、ＨＴＭＬ等が挙げられるが、これらに限定されるものではない。

【００３０】
本発明の実施形態によれば、処理済みテキストは、ネットワークノード２６０のたとえば記憶システム２７０に格納し、公衆ネットワーク２５０を介してリモート操作するようにしてもよい。この処理は、以下に説明するように、アプリケーションサービスプロバイダで処理済みデータを復号化することなく、トラステッドユーザおよび非トラステッドサーバアプリケーションのうちの少なくとも一方に対してトランスペアレントまたは不可視となるように、処理済みテキストに検索およびソートのうちの少なくとも一方が適用可能となされていてもよい。記憶システム２７０は、以下の説明においてデータベースを示す場合もあるが、任意の適当なデジタル記憶アーキテクチャであって、ＲＡＩＤ（レイド：Ｒｅｄｕｎｄａｎｔ Ａｒｒａｙ ｏｆ Ｉｎｄｅｐｅｎｄｅｎｔ Ｄｉｓｋｓ）等の任意の適当なハードウェア上に格納されていてもよい。

【００３１】
以上から、図１のデータフローに例示するように、トラステッドワークステーション２３０は、アプリケーションサービスプロバイダ２６０が使用する「Ａｃｍｅ Ｃｏｒｐ．」等の未処理入力データを供給するようにしてもよい。この入力テキストは、中間モジュール２００のたとえば阻止モジュール２１０により阻止され、データ保護モジュール２２０により処理されるようにしてもよい。データ保護モジュール２２０は、入力テキストを処理することにより、処理済みデータ「ＤＨＦＯＥＦＲＧＥＪＩＣ」として模式的に示すように、トークンと称する１または複数の個々のテキスト単位および暗号化可能な制御データを生成し、ネットワーク２５０を介して当該処理済みデータを非トラステッドアプリケーションサービスプロバイダ２６０に送信するようにしてもよい。そして、ユーザによる操作およびデータベース２７０への格納のうちの少なくとも一方を行ってもよい。当然のことながら、「ＤＨＦＯＥＦＲＧＥＪＩＣ」は模式的に示したに過ぎず、任意の適当な暗号化アルゴリズム等を用いて、たとえば任意の記号セットを得るようにしてもよい。以下に説明するように、本発明の一実施形態によれば、韓国語または中国語の記号等、ラテン語ではない文字記号または記号を用いてもよい。

【００３２】
図２を参照して、この図は、本発明の一実施形態に係る、クライアント端末２３０からアプリケーションサービスプロバイダ２６０への一般化されたデータフローを示した図である。エンドユーザは、暗号化されていない（プレーンテキストの）入力テキストを供給してもよい。この入力データは、クライアント端末２３０からネットワークノード２６０側へ送信され、阻止モジュール２１０で阻止されてもよい。阻止モジュール２１０は、入力データを処理して処理済みデータを供給するデータ保護モジュール２２０に入力テキストを供給してもよい。この処理には、入力テキストの少なくとも一部の暗号化が含まれる。そして、処理済みデータは、阻止モジュール２１０への送信後、公衆ネットワーク２５０を介して送信し、ネットワークノード２６０で受信して、ＳａａＳアプリケーション等のアプリケーションで操作してデータベース２７０に格納してもよい。当然のことながら、入力データは、記憶システム２７０に格納される新規または更新データであってもよいし、検索コマンド等の１または複数のパラメータ等、ＳａａＳアプリケーションに渡され

てリアルタイム操作される任意のデータであってもよい。

【００３３】

図３を参照して、この図は、本発明の一実施形態に係る、ネットワークノード２６０からクライアント端末２３０へのデータフローを示した図である。このようなプロセスは、ユーザが読み出すかまたは検索要求を行うことによってワークステーション２３０で開始してもよい。検索する用語等の要求パラメータは、図２に関連して上述したように処理してもよいし、ネットワークノード２６０のアプリケーションは、供給された処理済みパラメータに基づいて処理済みデータの検索またはソートを行ってもよい。また、ネットワークノード２６０は、たとえば検索または読み出し要求に応じて処理済みデータを読み出してもよい。この場合、処理済みデータには、暗号化された部分が含まれていてもよい。この処理済みデータは、公衆ネットワーク２５０を介して、クライアント端末２３０側に送信されてもよい。阻止モジュール２１０は、処理済みデータを阻止してデータ保護モジュール２２０に供給することにより、処理済みデータ内の任意の暗号化データを識別するようにしてもよい。識別した任意の暗号化データは、復号化して阻止モジュール２１０に供給することにより、データ通信を再開するようにしてもよい。また、阻止モジュール２１０は、未処理データ（復号化プレーンテキストデータ）をクライアントコンポーネント２４０に転送してユーザに表示するようにしてもよい。

【００３４】

トークン化および正規化全般

ネットワークノード２６０で動作するアプリケーションは、格納データを検索して結果を返すよう要求される場合がある。図１０は、本発明の一実施形態に係る、暗号化ユーザデータの検索を可能にするデータフローを模式的に示した図である。

【００３５】

まず、クライアント２４０は、データを入力し、中間モジュール２００を介して非トラステッドアプリケーション２６０に複数の格納要求を出してもよい。中間モジュールは、検索可能なすべての単語が暗号化された検索可能な単語にマッピングされて検索可能なすべての入力単語が厳密に１つの対応する暗号化された検索可能な単語を有するようにユーザ入力を暗号化する。暗号化された検索可能な単語は、暗号化前に正規化されていてもよい。

【００３６】

たとえば、図１０において、「ＢＡＤ」、「Ｂａｄ」、および「ｂａｄ」という単語はすべて、「ｃｃｃｃ」という単語に暗号化されるため、「ｂａｄ」を検索すると「ＢＡＤ」および「Ｂａｄ」を含む結果が供給される。

【００３７】

図１０において、「ｔｈｅ」および「ａ」という単語は、検索不可能と考えられるため、個々の暗号化された検索可能なトークンにはならない。これに対して、「ｄｏｇ」および「ｃａｔ」という単語はそれぞれ、「ｅｅｅｅ」および「ｂｂｂｂ」という暗号化された検索可能な単語にマッピングされる。検索可能な単語および検索不可能な単語の格標識を有する情報は、「ＺＺＺｙｔｕｖ」および「ＺＺＺａｂｃｄ」という暗号化されたトークンに含まれる。

【００３８】

図４を参照して、この図は、本発明の一実施形態に係る、ユーザテキストデータのサーバ側での検索およびインデキシングのうちの少なくとも一方を可能にするよう設計されたデータ処理方法１００を示した模式図である。上述の通り、この方法１００は、中間モジュールのたとえばデータ保護モジュールにより適用してもよい。当然のことながら、処理済みデータを受信して未処理データに変換する方法は、上記方法と実質的に逆であってもよい。

【００３９】

方法１００は、まずステップ１１０において、たとえばクライアント端末とネットワークノード間に動作可能に接続された中間モジュールにより、入力メッセージを受信する。

ステップ１１１においては、処理対象の入力メッセージ内の個々のデータ単位を識別する。たとえば、入力メッセージとしては、名前、名字、およびドキュメント本文等のフィールドが挙げられる。

【００４０】

ステップ１１２においては、識別したすべてのデータ単位に対して反復的に、まず未処理データ単位を取得し（ステップ１１３）、取得したデータ単位を処理するか否かを選択する。処理済みのデータ単位は、個々に処理してもよいし、まとめて処理してもよい。

【００４１】

ステップ１１４においては、入力データを処理するか否かを判定し、修正されない入力データについては保持する（ステップ１３０）。ステップ１１５においては、入力データの単位テキストを処理すべきか否か、およびいずれの部分を処理すべきか、のうちの少なくとも一方を判定する。たとえば、暗号化に適さない入力テキストの部分としては、「ＯＲ」や「ＡＮＤ」等の検索接続語またはデータに施す特殊なサーバ処理を示す「｛ｉｍｐｏｒｔａｎｔ｝」や「@ｌｏｃａｔｉｏｎ」等のアプリケーション固有の有意なテキストマークアップ等が挙げられる。

【００４２】

処理対象の入力テキストについては、ステップ１１６に進んで、入力テキストをトークンと称する個々のテキスト単位に分割する（入力テキストからトークンを決定するプロセスは、本明細書ではトークン化と称する）。トークン化は任意であって、方法１００には、（ａ）すべての入力データの単一トークンとしての暗号化、（ｂ）暗号化に適すると判定された入力データの個別暗号化による複数の処理済みトークンの供給（各処理済みトークンは１つの入力テキストを表す）、または（ｃ）それらの組み合わせが含まれていてもよい。

【００４３】

次にステップ１１７に進んで、特定の入力トークンを検索に不適と評価してもよい。たとえば、個々の単語を判定する基準は、所定の単語のリスト、英語辞書頻度リスト等の単語頻度リストにおける単語頻度閾値、単語の長さ、またはそれらの組み合わせであってもよい。

【００４４】

ステップ１１８においては、たとえば文字種、発音区別符号、合字分割、ユニコード文字記号の合成または分解（ユニコード規格により規定）等、検索に重要ではない情報を検索可能な入力トークンから抽出する。抽出した情報は、別個の場所に格納して後々利用するようにしてもよいし、制御トークンと称する出力トークンに設定してもよい。また、テキストトークンは、抽出情報を含まない正規化形式に変換してもよい。本明細書においては、このプロセスを正規化と称する。正規化は任意であって、任意の適当な方法で行ってもよい。

【００４５】

ステップ１１９においては、検索可能なトークン、検索可能なトークンから抽出された情報、および入力のその他の部分等、暗号化するすべての情報単位のビット表示を取得することにより、暗号化法を用いて暗号化する。情報単位は、検索可能または検索不可能として分類してもよい。検索不可能な情報単位は、結合してもよいし分解してもよい。また、入力テキストにおける検索可能なトークンの順序は変更してもよいし、元の順序を示す指標を検索不可能な情報単位に付加してもよい。

【００４６】

ステップ１２０においては、ＡＥＳ（Ａｄｖａｎｃｅｄ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）またはＤＥＳ（Ｄａｔａ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）等の暗号化法を用いて情報単位を暗号化する。

【００４７】

ステップ１２１においては、以下に詳細に説明するように、たとえばユニコードの１または複数の所定の隣接部分等、文字記号セットから得られた文字記号シーケンスから成る

出力テキスト単位に暗号化ビット表示を変換する。この文字記号セットは、前もって定義しておくことにより復号化を補助するようにしてもよい。

【００４８】

ステップ１２２においては、ステップ１２１で得られた出力テキストにより入力メッセージ中の入力データ単位を置き換える。

この方法では、識別したすべての入力単位に対してステップ１１２～１２２を適用し続けた後、サーバアプリケーションをホスティングするネットワークノードに処理済みメッセージを送信する（ステップ１３１）。

【００４９】

トークン化

上述の通りデータ処理方法がトークン化を含み、トークン化が多数の工程を含んでいてもよい。当然のことながら、以下のトークン化との関連で説明する工程の一部は任意である。さらに、当然のことながら、非トークン化すなわちトークン化処理済みデータの未処理データへの変換は、上記方法と実質的に逆であってもよい。

【００５０】

暗号化ユーザデータ上の検索を可能にするため、入力テキストは、トークン化と称するプロセスにおいて多数のセグメントに分割してもよい。個々に検索可能な用語を有するセグメントは（未処理）入力トークンと称するが、通常、入力トークンは全単語である。トークンでない入力セグメントは、「検索不可能な情報セット」と称する情報セットに付加される。このようなセグメントとしては、句読点や空白等の文字記号が挙げられる。

【００５１】

トークン化に関連して、複数の単語を１つのトークンとして結合してもよいし、１つの単語を２つ以上の構成トークンに分割してもよい。たとえば、「ｗｈｉｔｅｂｏａｒｄ」という複合語は、個々に検索可能な「ｗｈｉｔｅ」および「ｂｏａｒｄ」というトークンに分解してもよい。たとえば、中国語または日本語等の言語では通常、空白または文語テキストにおいて単語を分離する別の明確な文字記号は使用しない。このため、１つの中国語入力テキストは、複数の入力トークンに分割してもよい。また、このような結合または分割の指標は、検索不可能な情報セットに付加してもよい。

【００５２】

トークン化には、単語の形態的異形の検出、入力トークンの正規化形式への変更、および元の入力トークンの指標の検索不可能な情報セットへの付加が含まれていてもよい。たとえば、単語の形態的異形としては、名詞の複数形と単数形（「ｗｏｒｄ」、「ｗｏｒｄｓ」）、動詞の活用（「ｃｒｙ」、「ｃｒｉｅｄ」、「ｃｒｙｉｎｇ」）等が挙げられる。

【００５３】

トークン化には、検索される可能性が低い単語の検出、検索可能な入力トークンセットからのそれら単語の削除、および検索不可能な情報セットへの付加が含まれていてもよい。たとえば、このような検出には、（ａ）所定の単語セット、（ｂ）（この頻度閾値を超える頻度の単語は検索不可能と考えられる）単語頻度リストおよび頻度閾値を有する辞書、（ｃ）検索可能な単語の最小長さおよび最大長さのうちの少なくとも一方、または（ｂ）それらの任意の組み合わせを使用してもよい。

【００５４】

トークン化は、文字種、発音区別符号、合字、またはユニコードの合成／分解等の特定の文字記号特性を無視するサーバ側の検索およびインデキシングのうちの少なくとも一方に対応していてもよい。たとえば、「ＴｏＫｅＮ」および「ｔＯｋＥｎ」の検索では、テキスト検索時に同じ結果が得られてもよく、「ｔｏｋｅｎ」という単語の異形を含むすべての文字列が検索結果に現れる。

【００５５】

特性を区別しない上記のような検索は、（１）すべての入力文字記号を１つの正準形式に変換し、（２）元の文字記号を示す指標を生成し、（３）この指標を検索不可能な情報

セットに付加することによって対応するようにしてもよい。たとえば、トークン化は、入力トークン文字記号を１つの文字種（たとえば、小文字）に変換し、元の文字種を示す指標を検索不可能な情報セットに付加することによって、サーバ側で文字種を区別しない検索に対応するようにしてもよい。

【００５６】

【数１】

たとえば、発音区別符号（「Ë」、「Ë」、または「E」等の追加、削除、または修正した発音区別符号）は、検索中は無視してもよい。たとえば、「ｃａｆｅ」を検索すると、「Ｃａｆé」、「ＣＡＦＥ」、「cÄｆｅ」、または「çａｆｅ」等のユーザデータにマッチする。このシステムでは、これらすべての単語インスタンスを正規化形式「ｃａｆｅ」に変換して、元の発音区別符号を示す指標を検索不可能な情報セットに付加するようにしてもよい。

たとえば、このシステムでは、合字を区別しない検索（たとえば、ｄæｍｏｎとｄａｅｍｏｎ等）に対応するようにしてもよい。このシステムでは、「æ」の「ａｅ」への変換等により合字を正規化形式に変換し、元の合字を示す指標を生成して、検索不可能な情報セットに付加するようにしてもよい。

図６を参照して、この図は、「Ｃａｆé」という単語の処理例を示した図である。入力テキストは、大文字と発音区別符号とが取り除かれ、「ｃａｆｅ」というトークンに変換される。これに付随する制御トークンは、最初の文字が大文字で、４番目の文字が揚音アクセントを有することを示している。本発明の一部の実施形態によれば、文字は発音区別符号のない小文字と仮定してもよく、その場合、制御トークンは小文字であることや発音区別符号がないことを示す必要はない。

テキストマークアップおよび拡張情報

本発明の一実施形態によれば、入力テキストの処理には、アプリケーション固有のテキスト（少なくとも１つの処理命令）の検出が含まれていてもよい。また、これらの処理命令を非確定的変換済みテキストに付加してもよいし、この情報を処理済みテキストのプレーンテキストに残しておいてもよい。これにより、非トラステッドサーバは、このテキスト拡張情報に関連する任意の処理を適用してもよい。たとえば、ＨＴＭＬは、ＨＴＭＬタグをテキストに埋め込むことによって書式情報をユーザテキストに付加可能なテキスト拡張である。このシステムでは、（１）ＨＴＭＬタグの検索不可能情報への付加、（２）暗号化を伴わない入力ＨＴＭＬタグの出力処理済みテキストへの包含によるサーバ側処理の許可、および（３）ＨＴＭＬタグの通常テキストとしての取り扱い（たとえば、非ＨＴＭＬタグの入力テキストに対する任意の処理のＨＴＭＬタグへの適用）のいずれか少なくとも１つにより入力ＨＴＭＬタグを処理するようにしてもよい。

【００５７】

本発明の一部の実施形態によれば、入力テキストに少なくとも１つの処理命令を検出した場合に、中間モジュールは、当該少なくとも１つの処理命令を変換しないと決定してもよい。

【００５８】

本発明の一部の実施形態によれば、入力テキストに少なくとも１つの処理命令を検出した場合に、中間モジュールは、当該少なくとも１つの処理命令を非確定的に変換することを決定してもよい。

【００５９】

このシステムでは、時間、ユーザ、または処理済みテキスト生成時にシステムが把握しているその他の情報等の文脈情報を、検索不可能な情報セットに付加してもよい。

たとえば、本発明の実施形態によれば、このシステムでは、「重要」または「機密」等の特別指標を暗号化トークンに付加してもよい。これにより、復号化に際してこれらの指標が通知され、入力情報の復号化を示すイベントが生成され、たとえばレコードをログフ

ァイルに追加することによってこのイベントが処理されるようにしてもよい。

【００６０】

トークンの順序付け

入力テキストの処理には、処理済みテキストにおける入力トークンの順序の変更が含まれていてもよい。順序の変更に際しては、元の入力テキストにおける入力トークンの順序を示すトークン順序指標を生成し、検索不可能な情報セットに付加するようにしてもよい。

【００６１】

余剰トークン

入力テキストの処理には、出力テキストに含める少なくとも１つの偽造または擬似余剰トークンの生成が含まれていてもよい。このような擬似トークンによれば、暗号化テキストの統計解析に対する堅牢性を向上させることができる。余剰擬似トークンには、設定目標の統計分布を付加することによって、擬似トークンを隠蔽するとともに統計解析による復号化をさらに困難化するようにしてもよい。この少なくとも１つの余剰トークンは、秘密鍵へのアクセス後にのみ処理済みテキストに含まれるその他のトークンと識別可能である。たとえば、擬似トークンの目標分布のモデルとしては、英語の単語頻度を使用してもよい。

【００６２】

トークン化プロセス

検索不可能な情報セットは、１または複数の検索不可能なトークン（本明細書では制御トークンとも称する）に配置してもよく、これらのトークンは処理済み出力テキストに含めてもよい。制御トークンは、正規化入力トークンセットの前、後ろ、または内部に設けてもよい。検索不可能な情報セットは、全部または一部を暗号化した後、処理済み出力テキストに含めるようにしてもよい。

【００６３】

暗号化の前に、検索不可能な情報セットおよび検索可能なトークンのビット表示を取得してもよい。そのようなビット表示の取得には、特定の符号化および圧縮方法での入力データの圧縮および符号化が含まれていてもよい。

【００６４】

エラー検出指標を生成して検索不可能な情報セットに付加してもよい。たとえば、入力テキストのチェックサムを計算して検索不可能な情報セットに付加してもよい。

取得した入力トークンのビット表示は、場合によっては検索不可能な情報セットとともに、全部または一部を暗号化してもよい。検索可能な入力トークンの暗号化では、いずれの入力トークンインスタンスに対しても１つの暗号化形式が提供される。一方、検索不可能な情報の暗号化では、いずれの同じ情報セットインスタンスに対しても１または複数の暗号化形式が提供される。複数の暗号化形式によりセキュリティが向上する場合もあるが、ユーザデータを復号化せずに特定のサーバ側演算を行うのが困難または不可能となる可能性がある。複数の暗号化形式には、暗号化形式に埋め込まれた少なくとも１ビットの暗号化ソルトを使用してもよい。

【００６５】

その後、暗号化形式は、適当な符号化方法によりテキスト形式に変換してもよい。このような符号化方法は、次の特性のうちの少なくとも１つを有するものであってもよい。すなわち、（ａ）暗号化トークンを分離することにより、非トラステッドサーバアプリケーションが処理済みテキスト内の検索可能な単位を決定できるようにしてもよいし、（ｂ）非トラステッドサーバアプリケーションが検索可能な単位を決定しない文字記号セットを使用してもよいし（たとえば、文字記号「＋」を用いて非トラステッドサーバアプリケーションにより単語を分離してもよく、このため、暗号化トークンの符号化には不適であってもよい。また、たとえば、英語とヘブライ語の両者の文字記号を用いることにより、アプリケーションが両セットのシーケンスを分離するようにしてもよい）、（ｃ）サーバ側の長さ制限が満たされにくくなるようにコンパクトな表示を行ってもよいし、（ｄ）中間

モジュールにおいて、符号化および復号化に効率的なアルゴリズムを使用してもよい。

【００６６】

本発明の一部の実施形態によれば、処理済みテキストは、たとえばユニコード文字記号セットの少なくとも１つの隣接サブセットを含む文字記号セットのような、所定の文字記号セットから選択された文字記号列を含んでいてもよい。一部の実施形態において、この少なくとも１つの隣接サブセットは、文字、数字、または両者のカテゴリの文字記号を含んでいてもよい。また、一部の実施形態において、処理済みテキストでの使用のため選択される文字記号は、ユニコード文字記号セットの複数の隣接サブセットから選択されたものであってもよく、たとえば、ユニコード文字記号セットの２つ、３つ、４つ、または５つの別個のサブセットが選択されてもよい。一部の実施形態において、ユニコード文字記号セットのサブセットの数は、１よりも大きく１０以下であってもよい。

【００６７】

本発明の一部の実施形態において、ユニコード文字記号セットのサブセットは、ハングル、中国・日本・韓国（ＣＪＫ）統合表意文字、およびそれらの組み合わせから選択された１または複数のサブセットであってもよい。したがって、ＵＴＦ－１６符号化によりユーザ入力を格納するサーバアプリケーションには、たとえば韓国語文字記号を用いてもよい。文字記号のみを含む韓国語文字記号は、ユニコード文字記号セットの１つの範囲を表すため、効率的な符号化および復号化が可能である。たとえば、同じ理由で中国語文字記号セットを用いてもよい。中国語文字記号セットは、韓国語よりも範囲が広いものの、その使用は、個々の中国語文字記号の検索およびインデキシングのうちの少なくとも一方を個別に行うサーバアプリケーションには不適な場合がある。

【００６８】

ＵＴＦ－８符号化によりユーザ入力を格納するサーバアプリケーションには、たとえばＢＡＳＥ６４符号化を場合により修正して用いてもよい。ＢＡＳＥ６４符号化自体には、文字記号「＋」および「／」を含み、これによりサーバアプリケーションは、１つの暗号化トークンが１または複数の暗号化単語を有すると結論付けてもよい。

【００６９】

暗号化トークンの分離には、たとえば空白文字記号を用いてもよい。たとえば電子メールアドレスのフィールド等、空白文字記号が期待できない場合は、暗号化トークンの分離にピリオド「．」等の別の文字記号を用いてもよい。

【００７０】

処理済み出力テキストは、非トラステッドサーバから送信された場合、中間モジュールでの受信時に非暗号化テキストに含めてもよい。このシステムでは、復号化を開始するため、統計的に有意な特徴を処理済みテキストに生成してもよい。たとえば、このシステムでは、非暗号化テキスト内で暗号化テキストを検出する際に、希少な文字記号またはその組み合わせを検索対象の処理済みテキストに含めるようにしてもよい。

【００７１】

本発明の一部の実施形態によれば、出力トークンが特定の長さ制限を越えないように、処理済み出力テキストを２つ以上の出力トークンに配置してもよい。たとえば、第１の出力トークンには５０文字の長さ制限を適用し、後続の出力トークンには１０００文字の長さ制限を適用してもよい。

【００７２】

確定的暗号化と非確定的暗号化との組み合わせ

本発明の一部の実施形態では、入力テキストの確定的変換、非確定的変換、またはそれらの組み合わせを用いてもよい。本発明の実施形態では、入力データ（またはその一部）を確定的、非確定的、またはそれらの組み合わせのいずれで変換するかを決定した後、その決定に基づき、少なくとも１つの秘密鍵を用いて、入力テキストを確定的、非確定的、またはそれらの組み合わせで変換して処理済みテキストを取得し、当該処理済みテキストをサーバに送信するようにしてもよい。

【００７３】

本明細書において、入力テキストの非確定的変換は、その結果が複数の出力候補の１つとなる変換である。入力テキストの確定的変換は、出力候補を１つだけ含む変換である。通常はいずれの変換においても、１または複数の出力候補の決定に秘密鍵を用いるか、または秘密鍵に依存してもよい。

【００７４】

本発明の実施形態によれば、たとえば秘密鍵に応じて可逆暗号化を適用するか、または秘密鍵を用いて不可逆暗号化を行うことにより、確定的トークン表示を取得するようにしてもよい。また、たとえば秘密鍵を用いて対称暗号化アルゴリズムを適用するか、公開／私有鍵対の私有鍵を秘密鍵として用いて非対称暗号化アルゴリズムを適用するか、または秘密鍵に応じて他の可逆変換を行うことにより、非確定的トークン表示を取得するようにしてもよい。

【００７５】

本発明の一部の実施形態において、サーバは、過去に入力された入力テキストに対する検索機能を提供するものであってもよい。このような場合、中間モジュールは、入力テキストにおいて個々の検索可能なトークンを確定的に変換することを選択してもよい。このような確定的な変換により、処理済みの検索可能な用語を含む将来の検索クエリーがサーバで正しく処理可能となってもよい。また、たとえばセキュリティの向上のため、入力テキストの一部を非確定的に変換してもよい。本発明の実施形態によれば、入力テキストの一部を確定的に変換することにより、入力テキストの一部の反復インスタンス間の正確なマッチを要するサーバ側機能が得られるようにしてもよい。たとえば、前後の変更がわずかに異なる入力テキストの複数の変更をサーバが比較する場合、当該サーバは、単語ごとまたは行ごとの差異解析を行ってもよい。したがって、このような例では、入力テキストの単語または行を確定的に変換することにより、上記のようにサーバ上で正確なマッチ動作が得られる。

【００７６】

たとえば、本発明の一実施形態における入力テキストの処理工程には、（１）入力テキストの一部または全部を１または複数の処理済みトークンへ非確定的に暗号化する工程と、（２）入力テキストの一部または全部の適当な入力トークンに対応する処理済みトークンを生成する工程と（たとえば、入力テキストのトークン化後、正規化後等）、（３）非確定的および確定的に変換された処理済みデータを処理済み出力テキストに含めてネットワークノードに送信および格納する工程とを含んでいてもよい。本発明の一部の実施形態によれば、入力テキストを確定的、非確定的、またはそれらの組み合わせのいずれで変換するかの決定は、当該単語が単語セットの要素であるか否かに基づいて行ってもよい。このように、たとえば検索に利用可能となる入力トークンを確定的に変換して、上記単語を検索可能とするようにしてもよい。検索に基づくレコードの格納に際しては、確定的または非確定的に変換された処理済みデータを含む処理済み入力テキストを検索結果として返すようにしてもよい。これに対して、検索に利用可能とならない入力トークンについては、確定的に変換する必要はない。

【００７７】

本発明の一部の実施形態において、入力テキストを確定的、非確定的、またはそれらの組み合わせのいずれで変換するかの決定は、当該単語の長さに基づいて行ってもよい。これにより、たとえば、入力テキストの単語をその長さに基づいて非確定的に変換することを決定してもよい。また、本発明の一実施形態の一例として、たとえば２文字以下の短い単語については非確定的に変換し、３文字以上の長い単語については確定的に変換するようにしてもよい。以上、このような方法では、最小文字数を下回る短い単語が検索不可能であってもよい。

【００７８】

本発明の一実施形態において、非確定的な変換は第１の鍵を用いて行い、確定的な変換は第２の鍵を用いて行うようにしてもよい。

本発明の一部の実施形態において、第１の鍵と第２の鍵とは同一であってもよい。また

、本発明の別の実施形態において、第１の鍵と第２の鍵とは異なっていてもよい。
【００７９】
本発明の一部の実施形態において、出力テキストの全長が長さ制限を超える場合は、１または複数の確定的に生成されたトークンを省略または削除するようにしてもよい。また、本発明の一部の実施形態においては、入力テキストの少なくとも一部を変換しないよう決定してもよい。
【００８０】
本発明の実施形態に係る処理済みテキストの読み出しプロセスは、実質的に逆に作用してもよい。すなわち、処理済みテキストを中間モジュールで受信するとともに、処理済みテキストに適当な逆処理を適用して元の入力テキストを取得するようにしてもよい。本発明の一部の実施形態においては、元の入力テキストをクライアント装置に送信あるいは他の方法によって供給することにより、たとえばクライアント装置を操作するユーザまたはアプリケーションに対して表示または提供するようにしてもよい。
【００８１】
検索クエリーの処理
中間モジュールで受信される入力テキストは、検索対象の少なくとも１つの検索語を含む検索クエリーであってもよい。検索クエリーの入力テキストは、（ａ）ネットワークノードにおいて正しい検索機能を促進するとともに、（ｂ）ネットワークノードが当該テキストをクライアントに送り返した場合に中間モジュールで検索クエリーの復号化が行えるように、中間モジュールで処理してもよい。検索クエリーは一般に、他の入力テキストと同様にネットワークノードで処理されるが、別の処理ステップを適用してもよい。
【００８２】
本発明の一部の実施形態において、入力テキストを変換する工程は、第１の鍵を用いて検索クエリーの少なくとも１つの検索語を確定的に変換することにより、少なくとも１つの確定的変換済み検索語を生成する工程を含んでいてもよい。これにより、処理済み入力テキストをサーバに送信する工程は、複数の確定的変換済み検索語をサーバに送信する工程を含んでいてもよい。また、本発明の一部の実施形態において、検索クエリーの複数の検索語は、取り扱いおよび変換を別個に行うようにしてもよい。
【００８３】
本発明の一部の実施形態において、処理済み検索クエリーは実質的に、確定的変換済み検索語のみを含んでいてもよく、その確定的な変換が可逆変換であってもよい。ネットワークノードは、処理済みの用語を検索して、その結果セットをクライアントに返すようにしてもよい。また、中間モジュールは、処理済みの検索語を用いて元の入力テキストを取得してもよい。
【００８４】
本発明の一部の実施形態において、検索クエリーを変換する工程は、第２の鍵を用いて検索クエリーの実質的に全体を非確定的に変換することにより、非確定的変換済みテキストを生成する工程と、論理和演算子（たとえば、「ＯＲ」演算子等）を用いて上記少なくとも１つの確定的変換済み検索語と非確定的変換済みテキストとを結合して結合処理済みテキストを取得する工程とを含んでいてもよく、処理済み入力テキストをサーバに送信する工程は、当該結合処理済みテキストをサーバに送信する工程を含む。ネットワークノードは、処理済み検索語および非確定的処理済みテキストを分離検索し、確定的変換済み検索語に基づいて結果を取得する（または非検出となる）が、非確定的変換済みテキストについては結果を取得しない。したがって、検索の結果は、処理済み検索語の検索の結果を返すことになってもよい。本発明の一実施形態に係る上記方法を用いることにより、中間モジュールは、ネットワークノードから非確定的変換済みテキストを受信し、そこから検索クエリーの元の入力テキストを取得するようにしてもよい。
【００８５】
処理済みテキストの保存場所
ネットワークノードサーバの中には、クエリー等の要求に応じて不完全な検索結果を返

すものがあってもよい。たとえば、検索クエリーの結果が１００文字のフィールドである場合、サーバは、フィールドの先頭から２０文字のみを返すようにしてもよい。そして、ユーザが検出レコードを選択した場合、サーバは、フィールド全体を提供する。本発明の実施形態によれば、中間モジュールは、このような制約の中でも動作できるものとする。また、本発明の実施形態によれば、サーバが処理済みテキストの複数単位を省略する場合、これらの単位は、処理済みテキスト内の個々のトークン、処理済みテキスト全体、またはその両者であってもよい。

【００８６】

本発明の実施形態によれば、上記の問題は、たとえば中間モジュールまたは中間モジュールが管理、制御、あるいはアクセス可能な記憶装置に処理済みテキストの保存場所を設けることによって解決してもよい。このシステムでは、復号化ステップにおいて元の入力テキストを取得する前に、不完全な状態から以下のように再生を試みてもよい。すなわち、（１）中間モジュールは、たとえば非トラステッドサーバやそれに付随する記憶装置を介さずに、復号化ステップにおいてトラステッド記憶装置に完全な処理済みテキスト単位を記憶させるようにしてもよい。（２）不完全な処理済みテキストがサーバから送信されて中間モジュールで受信された場合は、トラステッド記憶ユニットを参照して、当該不完全な処理済みテキスト単位にマッチするかまたは対応する１または複数の完全な処理済みテキスト単位が存在するか否かを判定する。（３）存在する場合は、中間モジュールが不完全な処理済みテキスト単位を対応する完全な処理済みテキスト単位で置き換えることにより、再生処理済みテキストを取得する。（４）再生処理済みテキストは、逆処理法（たとえば、秘密鍵を用いた復号化等）により処理して、元の入力テキストを取得する。そして、必要に応じて、元の入力テキストすなわち未処理テキストをクライアント装置に供給してもよい。

【００８７】

本発明の一部の実施形態において、保存場所に格納されるものは、処理済みテキストに付随する少なくとも１つの完全な処理済み要素であってもよい。たとえば、処理済み要素としては、上記の処理済みテキスト全体であってもよいし、処理済みテキストに含まれる単語またはその他の部分であってもよい。

【００８８】

このような保存場所を用いるシステムおよび方法は、たとえば検索要求、記録要求、または報告要求等、クライアント装置からの任意の適当な要求に適用してもよい。

ベイトを用いた非トラステッドサーバ変換の検出

非トラステッドサーバは、多数の変換のうちの１または複数を処理済みユーザデータのインスタンスに頻繁に適用してもよい。このような変換は、トラステッドワークステーション上のクライアントコンポーネントによって期待されてもよいが、本明細書に記載の中間モジュールは把握していなくてもよい。したがって、本発明の実施形態によれば、中間モジュールは、処理済みユーザデータに適用される変換の種類を推測する方法を利用してもよい。

【００８９】

本発明の一実施形態によれば、中間モジュールは、既知の場所の暗号化ユーザデータに超過情報を付加してもよい（本明細書ではベイトと称する）。ベイトは、処理済みユーザデータが中間モジュールで受信された場合、処理済みユーザデータに適用される変換の種類を推測するために使用してもよい。ベイトを使用可能な変換の適用例としては、特定の文字記号符号化方法およびＨＴＭＬタグの除去等が挙げられるが、これらに限定されるものではない。

【００９０】

たとえば、非トラステッドサーバは、受信した暗号化ユーザデータに様々な、かつ場合に応じた符号化方法やそれらの組み合わせをその時に適用してもよい。暗号化テキストが非トラステッドサーバから中間モジュールで受信された場合は、非トラステッドサーバアプリケーションが使用する多数の符号化方法のうちの１つで暗号化テキストを符号化する

ことにより、トラステッドワークステーション上のクライアントコンポーネントと接続するようにしてもよい。この符号化方法は、サーバが生成するメッセージで示唆してもよいし、示唆しなくてもよい。クライアントコンポーネントは通常、サーバコンポーネントを認識して、使用される符号化方法を確実に把握していてもよい。ただし、中間モジュールは、暗号化テキストのインスタンスごとに使用される個別の符号化を把握していなくてもよい。それでも、クライアントコンポーネントへの供給前にユーザデータを復号化する場合、本発明の実施形態に係る中間モジュールは、サーバが適用しクライアントが期待する符号化方法と同じものを使用できるものとする。すなわち、非トラステッドサーバおよびトラステッドワークステーションが使用する符号化方法を中間モジュールが把握していない場合は、中間モジュールによる処理および再処理で情報が喪失または歪曲される可能性がある。

【００９１】

符号化方法の検出を容易にするため、中間モジュールは、符号化ベイトとして知られる所定の文字記号を暗号化テキストに付加してもよい。符号化ベイトは、クライアントコンポーネントへの供給前に暗号化ユーザデータとともにサーバで暗号化してもよい。中間モジュールが暗号化トークンを検出した場合は、暗号化テキストのインスタンスの符号化に用いられる符号化方法の種類を推測するために符号化ベイトを調べてもよい。したがって、中間モジュールは、推測した符号化方法を用いて処理済みメッセージの復号化テキストを符号化してもよい。符号化方法の例としては、（ａ）ＵＴＦ－８符号化、（ｂ）ＵＴＦ－８が後続するＨＴＭＬエスケープ文字列を用いた符号化、（ｃ）ＪＡＶＡＳｃｒｉｐｔ（登録商標）エスケープ文字列を用いた符号化に続くＪａｖａＳｃｒｉｐｔエスケープ文字列の再使用およびその後のＬａｔｉｎ－１符号化（別称、ＩＳＯ－８８５９－１）の実行等が挙げられるが、これらに限定されるものではない。たとえば、ＪａｖａＳｃｒｉｐｔエスケープは通常、文字記号をバックスラッシュおよび別の文字記号で置き換えることにより作用する。一例として、改行文字記号は、バックスラッシュおよび文字記号「ｎ」、すなわち文字列「＼ｎ」で置き換えられる。

【００９２】

本発明の一部の実施形態において、ベイトは、処理済みテキストの少なくとも１つの変換可能な文字記号にマッチする置換文字記号または置換文字記号列（たとえば、１または複数のエスケープ文字記号）への置換を含む少なくとも１つの変換を検出するのに使用してもよい。

【００９３】

本明細書では、山括弧「＜」およびバックスラッシュ「＼」から成る符号化ベイトを用いる例を提供する。ユーザは、文字列「Ｔｈｉｓ‘ ｉｓ ａ ｑｕｏｔｅ」を入力してもよい。この文字列は、たとえば「ＱＩＦＪＤＪＮＺＯＰ」に暗号化される。暗号化においては、「ＱＩＦＪＤＪＮＺＯＰ」が「＜＼ＱＩＦＪＤＪＮＺＯＰ」になるように、暗号化トークンにベイトが付加される（＜＼がベイトである）。サーバは、暗号化文字列を受信し、ＪａｖａＳｃｒｉｐｔファイルでクライアントに送信してもよい。ＪａｖａＳｃｒｉｐｔファイルにおいては、サーバがバックスラッシュのみをエスケープする必要があり、山括弧はその必要がない。したがって、クライアントに送信されるメッセージには「＜＼＼ＱＩＦＪＤＪＮＺＯＰ」が含まれる。ここで、ベイトの元のバックスラッシュは、別のバックスラッシュによりエスケープされている。中間モジュールは、元の山括弧およびエスケープされたバックスラッシュで始まるメッセージの暗号化トークンを検出した場合、当該トークンがＪａｖａＳｃｒｉｐｔエスケープであると推測してもよい。その結果として、中間モジュールは、入力されたＱＩＦＪＤＪＮＺＯＰを「Ｔｈｉｓ‘ ｉｓ ａ ｑｕｏｔｅ」に復号化してもよい。ただし、クライアントがＪａｖａＳｃｒｉｐｔエスケープテキストを期待しているものと推測した場合、モジュールは、ＪａｖａＳｃｒｉｐｔエスケープを用い、たとえば引用符をエスケープして「Ｔｈｉｓ ＼‘ ｉｓ ａ ｑｕｏｔｅ」を生成することにより、復号化文字列を符号化するようにしてもよい。このように、復号化された引用符は、符号化ベイトにより推測された符号化規則を使用してい

る。その後、復号化および符号化された文字列はクライアントに転送される。

【００９４】

ベイトを使用可能な別の例としてはＨＴＭＬ変換が挙げられるが、そのうちのＨＴＭＬタグの除去は特殊なケースである。非トラステッドサーバは、ＨＴＭＬマークアップで補強されたテキストを受信し、ＨＴＭＬタグの全部または一部が除去された受信テキストのインスタンスを生成し、これらインスタンスをクライアントコンポーネントに返してもよい。この場合、中間モジュールは、処理済みユーザデータにＨＴＭＬタグベイトを含めてもよい。ＨＴＭＬタグベイトは、処理済みユーザデータの受信時に中間モジュールによって除去するとともに、その存否に基づいて、ＨＴＭＬタグが復号化ユーザデータから除去可能であるか否かを推測し、これにより、クライアントコンポーネントに返されるメッセージ中の復号化ＨＴＭＬタグを保持または除去するようにしてもよい。

【００９５】

一部の実施形態においては、複数のベイトを処理済みテキストに付加することにより、非トラステッドサーバが適用する複数の変換または符号化方法を検出するようにしてもよい。

【００９６】

長さ制限

本発明の一部の実施形態においては、入力テキストの複数の個別部分を変換してもよく、当該入力テキストの複数部分の少なくとも１つは、たとえば各部の省略により最大の文字記号数以上を含まない。また、本発明の一部の実施形態においては、入力テキストの複数の個別部分を変換してもよく、当該入力テキストの複数の各部分は、たとえば各部の省略により最大の文字記号数以上を含まない。

【００９７】

トークン化の例

図５を参照して、この図は、センテンス「Ｔｈｉｓ　ｓｅｎｔｅｎｃｅ　ｈａｓ　ＦＩＶＥ　ｗｏｒｄｓ！」を含む入力テキストの正規化およびトークン化を示した図である。入力テキスト５１０は、センテンス「Ｔｈｉｓ　ｓｅｎｔｅｎｃｅ　ｈａｓ　ＦＩＶＥ　ｗｏｒｄｓ！」を含む。このセンテンスは、「Ｔｈｉｓ」、「ｓｅｎｔｅｎｃｅ」、「ｈａｓ」、「ＦＩＶＥ」、「ｗｏｒｄｓ」、および「！」という入力トークンにトークン化してもよい。また、これらの入力トークンは、正規化により正規化入力トークンおよびメタデータを供給してもよい。正規化入力トークンは、「Ｔｈｉｓ」、「ｓｅｎｔｅｎｃｅ」、「ｈａｓ」、「ｆｉｖｅ」、「ｗｏｒｄｓ」、および「！」という書式となる。また、「ｓｅｎｔｅｎｃｅ」に付随するメタデータは「小文字」である。「ＦＩＶＥ」に付随するメタデータは「大文字」、「ｗｏｒｄｓ」に付随するメタデータは「小文字」および「複数」である。

【００９８】

次に、共通の入力トークンである単語「ｔｈｉｓ」、「ｈａｓ」、および非単語「！」を検出する。これらの入力トークンは、非確定的に暗号化してもよく、たとえばソルト（「＊」と表示）で暗号化してもよい。

【００９９】

そして、共通ではない入力トークン「ｗｏｒｄ」、「ｓｅｎｔｅｎｃｅ」、および「ｆｉｖｅ」を検出する。これらの単語は、確定的に暗号化してもよい。

入力トークンの順序は変更してもよく、これに応じて、順序メタデータを生成してもよい。この順序メタデータ、文字種メタデータ、および複数メタデータは、制御トークン５３０に含まれていてもよい。

【０１００】

ソート支援

多くのＳａａＳアプリケーションに共通のテキスト処理機能は、特定のフィールドまたはその他属性の辞書式順序でレコードをソートすることである。したがって、この機能は、順序維持暗号化プロセスにより処理済みテキストを供給する際に有用となる場合がある

。
【０１０１】
順序維持手法は多数存在するが、いずれを実行してもよい。たとえば、順序維持は、以下のいずれの方法でも得られる。（ａ）全レコードのリストを阻止モジュール上に保持し、必要に応じて部位固有の順序付けを行う。この方法ではほとんどの場合、表示およびデータ管理の両方に各サーバ機能の複製が必要となる。（ｂ）サーバにＡＰＩを提供して、特定文字列のソート順序のクエリーを行う。または、（ｃ）ネットワークノードでの修正なしに実際のソート順序を維持する、辞書式にソート可能な表示を生成する。
【０１０２】
本発明に係る暗号化方法では、以下の各ステップまたはそれらの組み合わせを適用することによって、入力テキストのレコードの順序を維持するようにしてもよい。すなわち、（１）入力データ（数値化されていない場合）を数値に変換し、（２）数値に順序維持変換を適用して出力数値を取得し、（３）出力数値から辞書式にソート可能な表示を取得し、（４）処理済み出力テキストにおいて、辞書式にソート可能な表示を（テキストデータ中の）接頭辞文字列または出力データ全体として使用する。この順序維持変換は、単調増加関数であってもよい。また、この順序維持関数は、乱数源から生成可能な私有鍵を用いることにより、その機能をパラメータ化してもよい。私有鍵は、まとめてソートされた入力セットごとにセットで生成してもよい。本発明の実施形態によれば、以下に説明するように、順序情報を生成する工程は、秘密鍵依存の順序維持関数を入力テキストに適用する工程を含んでいてもよい。
【０１０３】
本発明の一部の実施形態によれば、入力テキストの不完全型に基づいて順序情報を生成してもよい。本発明のさらに別の実施形態によれば、入力テキストの複数の不完全単語に基づいて、当該単語の登場順に順序情報を生成してもよい。
【０１０４】
本発明の一部の実施形態によれば、中間モジュールは、順序維持変換を適用することによって入力テキストを処理してもよく、当該順序維持変換は、入力テキストに基づいて、照合規則に応じた入力テキスト候補セットにおける当該入力テキストの相対順序を示す順序情報を生成すること、入力テキストを変換して処理済みテキストを取得すること、処理済みテキストをサーバに送信することを含む。また、本発明の一部の実施形態によれば、順序情報を接頭辞として処理済み入力データに付加するとともに当該結合した順序情報および処理済み入力データをサーバに送信することにより、当該処理済み入力テキストに関連して当該順序情報をサーバに送信してもよい。
【０１０５】
順序維持暗号化方法に関連するセキュリティ上のリスクを低減するため、中間装置は、順序維持出力の生成時に、入力データの縮小部分のみを考慮してもよい。入力を縮小して入力データの取得部分を抑えるには、（ａ）「ｔｈｅ」や「ａ」等の特定の単語を無視したり、（ｂ）あらゆる単語の特定箇所またはそれ以降のすべての文字記号を無視したり（たとえば、「ｚｅｂｒａ」の文字記号における「ｒａ」を無視したり）、（ｃ）レコード内の最後の単語を無視したり、（ｄ）順序維持関数の入力定義域を縮小したり、（ｅ）文字種等の特定の文字記号特性を無視したり、または（ｅ）それらを組み合わせたりしてもよい。
【０１０６】
図７は、本発明の一実施形態に係る、処理済みテキストに含まれるテキストデータの順序維持表示の取得に利用可能な方法１７０の様々なステップを示した図である。ステップ１７１においては、暗号化する入力テキストを受信する。ステップ１７２においては、入力テキストから特定の単語を除去する。ステップ１７３においては、文字種、発音区別符号、合字等の特定の文字記号特性を除去する。ステップ１７４においては、入力テキストから最後の文字記号が除去されるように、暗号化方法の所定のパラメータに応じて入力単語を省略する。

【０１０７】
ステップ１７５においては、入力テキストにおける特定の最後の単語を除去する。以上から、任意のステップ１７２、１７３、１７４、および１７５の１または複数を実行することにより、生成される入力テキストの長さを抑えるようにしてもよい。ステップ１７６においては、（任意で長さを抑えた）入力テキストを数値に変換して入力数値を取得する。ステップ１７７においては、入力数値に順序維持関数を適用して出力数値を取得する。ステップ１７８においては、出力数値から順序維持表示を取得する。最後に、ステップ１７９においては、処理済みテキストの接頭辞または暗号化データ全体として順序維持表示を設定する。

【０１０８】
ステップ１７２～１７６の適用を示す以下の例では、入力テキスト「Ｔｈｅ　Ｇｒｅｅｎ　Ｚｅｂｒａ」の入力数値を以下のように計算する。すなわち、（ａ）入力トークンセット「Ｔｈｅ　Ｇｒｅｅｎ　Ｚｅｂｒａ」を受信し、（ｂ）重要ではない入力トークン「ｔｈｅ」を無視して有意な入力トークン「Ｇｒｅｅｎ　Ｚｅｂｒａ」を提供し、（ｃ）有意な入力トークンを正規化して「ｇｒｅｅｎ　ｚｅｂｒａ」を提供し、（ｃ）たとえばユーザ定義に基づき、各入力トークンの先頭から３文字のみを選択して６つの有意な文字記号「ｇｒｅ　ｚｅｂ」を提供し、（ｄ）各文字の数値をそれぞれの入力トークンにおける位置の重みに基づいて表１のように計算し、（ｅ）文字の数値を合算して入力トークンセットの数値０．２９６１９９７９００６８３４５を提供する。

【０１０９】
重みＷは、アルファベットサイズＡに対する文字記号位置の負の指数Ｐとして、$W=A^{-P}$と表してもよい。英語テキストの場合、アルファベットサイズは２６である。

【０１１０】
【表１】

文字	アルファベット値	位置 (P)	重み (W)	重み付けされた値
G	7	1	0.03846153846153850000	0.26923076923076900000000000
R	18	2	0.00147928994082840000	0.02662721893491120000000000
E	5	3	0.00005689576695493860	0.00028447883477469300000000
Z	26	4	0.00000218829872903610	0.00005689576695493860000000
E	5	5	0.00000008416533573216	0.00000042082667866078800000
B	2	6	0.00000000323712829739	0.00000000647425659478136000

図８は、本発明の一実施形態に係る、順序維持関数の生成方法３００を示しており、たとえば方法１７０のステップ１７７で使用する。ステップ１８０においては、たとえばユーザまたはプログラムによる設定に応じて関数の定義域（D_1，D_2）および値域（R_1，R_2）を決定する。ステップ１８１においては、順序維持関数の出力値の計算に使用する私有鍵Ｋを取得する。ステップ１８２においては、（場合により、方法１７０のステップ１７６から）入力値V_{in}を受信する。ステップ１８３および１８４においては、元の値域に含まれる鍵依存の位置が始点および終点となるように関数の値域を変更する。ステップ１８５においては、関数の定義域に含まれる関数の鍵Ｋに依存した点D_{mid}が$D_{mid}=f_1(D_1, D_2, K)$を満たすように選択する。ステップ１８６においては、点$R_L=f_2(R_1, R_2, K, n)$および$R_H=f_3(R_1, R_2, K, n)$が$R_1<R_L<R_h<R_2$を満たすように選択する。ここで、R_LおよびR_hは、関数の鍵Ｋおよび反復回数ｎのうちの少なくとも一方（初期値はｎ＝１）によって決まる。ステップ１８７においては、入力数値V_{in}が現在の定義域（D_1，D_2）の下方部（D_1，D_{mid}）または上方部（D_{mid}，D_2）のいずれに含まれるかを確認する。V_{in}が下方部に含まれる場合は、ステップ１８８ａを実行し、そうでなければステップ１８８ｂを実行する。ステップ１８８ａおよび１８８ｂにおいては、関数の定義域（D_1，D_2）および値域（R_1，R_2）を修正する。すなわち、ステップ１８８ａにおいては、（D_1，D_2）を

（D_1，D_{mid}）に設定し、（R_1，R_2）を（R_1，R_L）に設定する。ステップ１８８ｂにおいては、（D_1，D_2）を（D_{mid}，D_2）に設定し、（R_1，R_2）を（R_H，R_2）に設定する。ステップ１８５～１８８は、ステップ１８９で所定の停止基準が満たされるまで繰り返す。停止基準の例としては、閾値サイズ$D_{threshold}$が現在の定義域サイズ｜Ｄ｜＝D_2-D_1を超えること、閾値サイズ$R_{threshold}$が現在の値域サイズ｜Ｒ｜＝R_2-R_1を超えること、またはそれらの組み合わせが挙げられる。

【０１１１】

以下に、方法１７０のステップ１７８で利用可能な符号化方法を例示する。まず、順序維持関数により生成された変換数値が０．３４４３２３９４７であり、辞書式にソート可能な表示が１０文字長で英語の小文字のみを含むものと仮定する。表２は、算術符号化方法の１０回反復による１０文字長の辞書式にソート可能な表示の生成を示したものである。

【０１１２】

【表２】

文字ナンバー	$Value_n$ (=26 × ($Value_{n-1}$ − $Rounded_{n-1}$ ÷ 26))	文字値 (×26)	四捨五入された値	出力文字
1	0.344323947	8.952422617	8	h
2	0.952422617	24.76298804	24	x
3	0.762988037	19.83768896	19	s
4	0.837688957	21.77991288	21	u
5	0.779912877	20.2777348	20	t
6	0.277734797	7.221104712	7	g
7	0.221104712	5.748722505	5	e
8	0.748722505	19.46678512	19	s
9	0.46678512	12.13641313	12	l
10	0.136413127	3.546741304	3	c

表２に示すように、辞書式にソート可能な表示は「ｈｘｓｕｔｇｅｓｌｃ」である。

【０１１３】

ここで、物理的なコンピュータ可読媒体を設けることができる。この媒体には、プロセッサによる実行に際して、プロセッサに方法１００またはその一部を実行させることができる命令を格納する。このような物理的なコンピュータ可読媒体としては、ディスク、ディスケット、テープ、カセット、メモリースティック、フラッシュメモリーユニット、揮発性メモリーユニット等が考えられる。

【０１１４】

本明細書では、本発明の特定の特徴を例示・説明したが、当業者であれば多くの改良、置換、変形、および均等物を想到し得るであろう。したがって、当然のことながら、添付の請求の範囲は、このような改良や変形がすべて本発明の精神に含まれるように網羅するものである。

【図９】

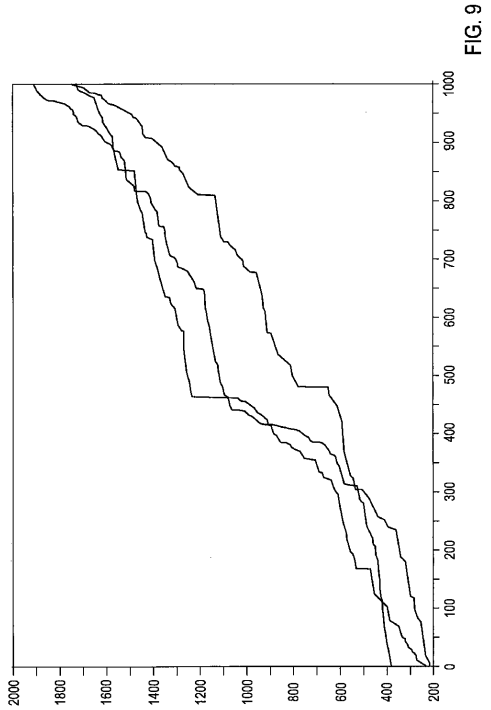
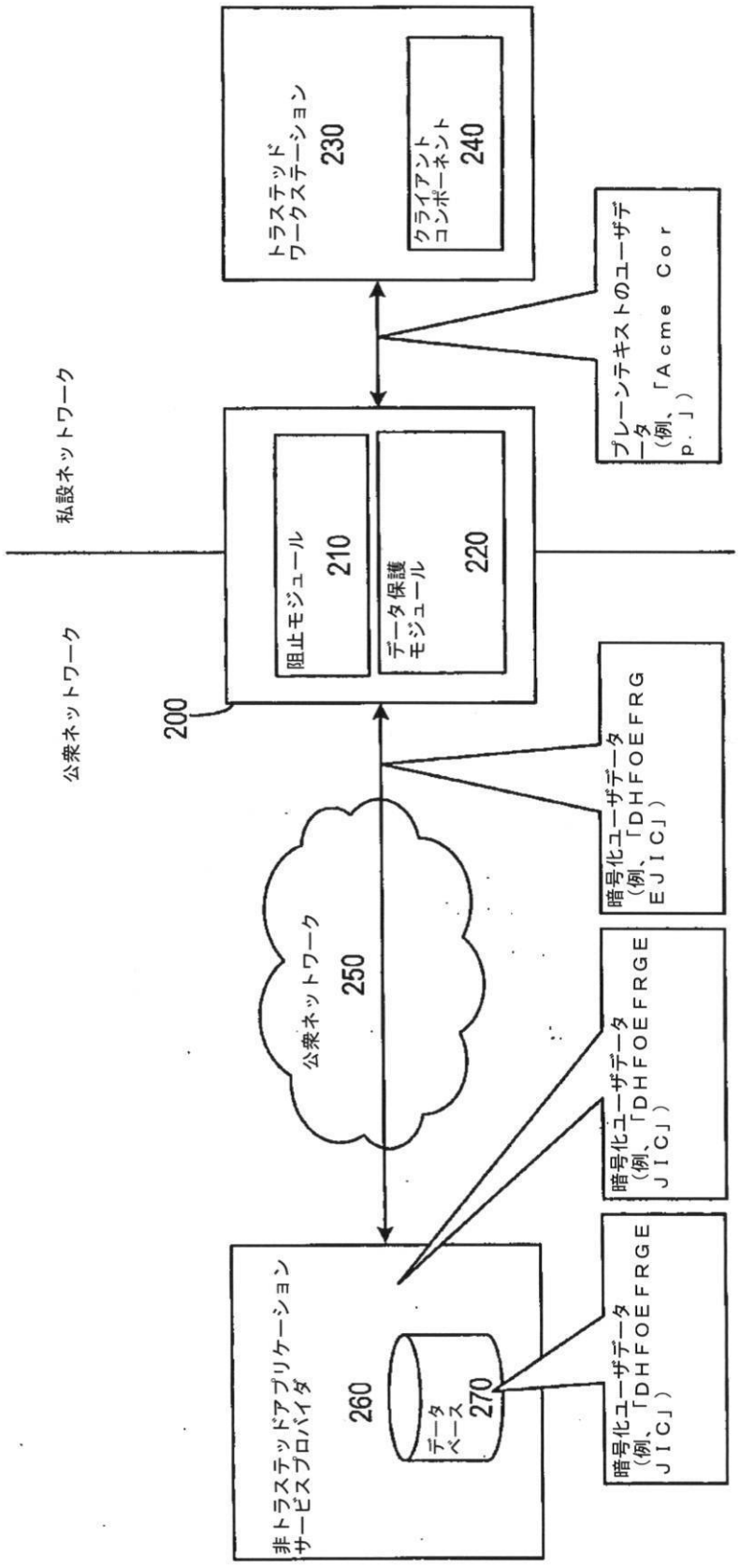
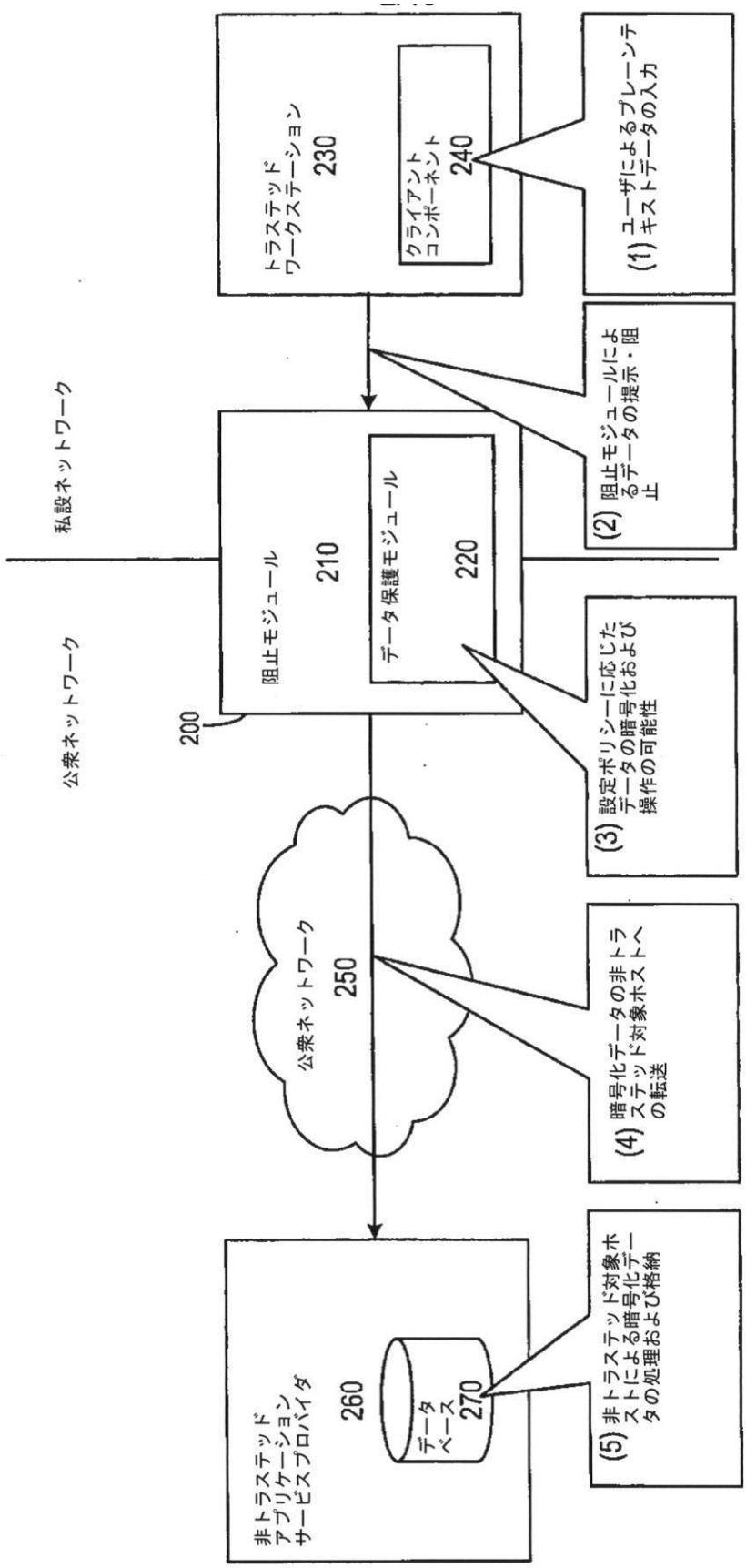

2000
1800
1600
1400
1200
1000
800
600
400
200
0
100
200
300
400
500
600
700
800
900
1000

FIG. 9

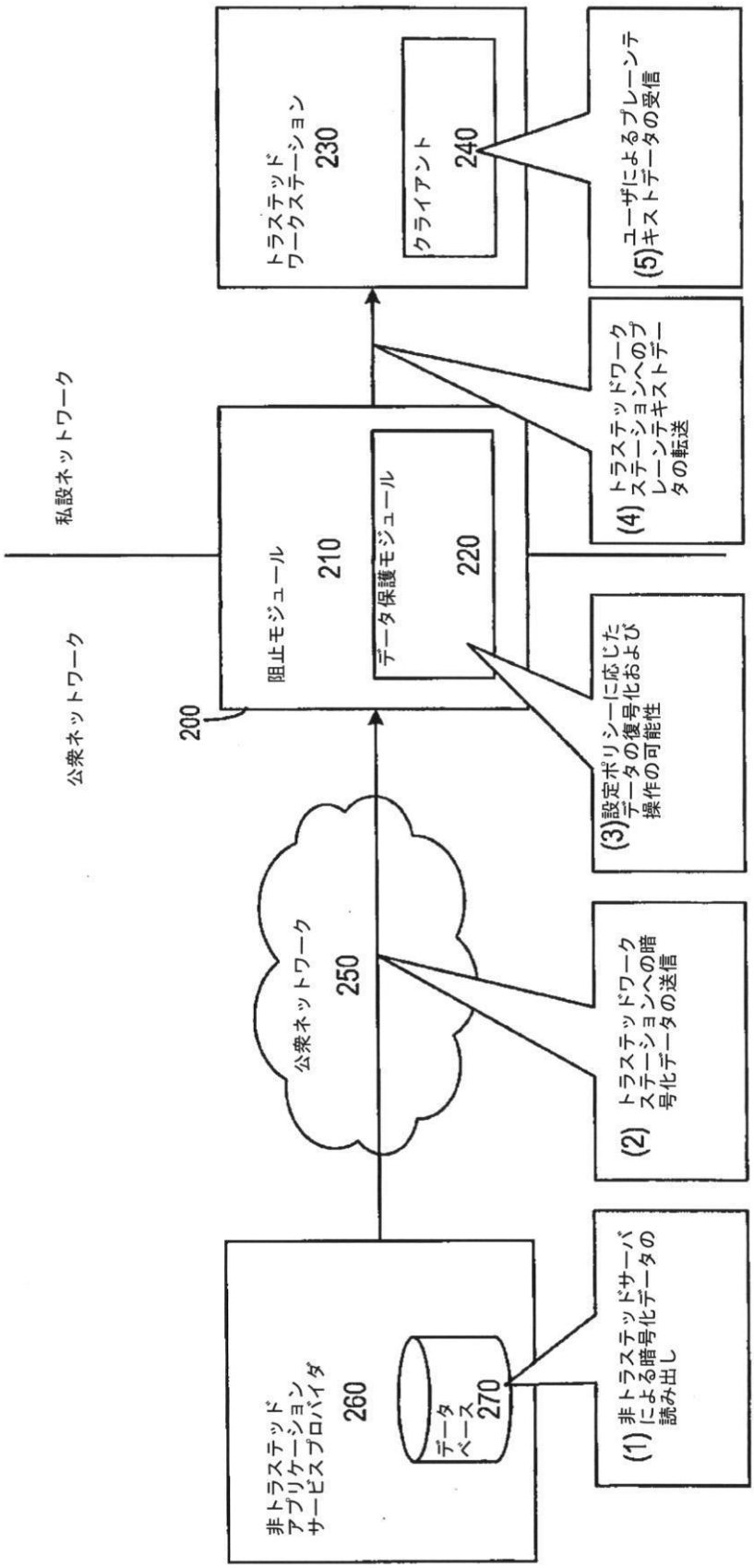
【図１】


トラステッド
ワークステーション
230
クライアント
コンポーネント
240
プレーンテキストのユーザデータ
（例、「Acme Corp.」）
私設ネットワーク
公衆ネットワーク
阻止モジュール
210
データ保護
モジュール
220
200
暗号化ユーザデータ
（例、「DHFOEFRGEJIC」）
公衆ネットワーク
250
暗号化ユーザデータ
（例、「DHFOEFRGEJIC」）
非トラステッドアプリケーション
サービスプロバイダ
260
データベース
270
暗号化ユーザデータ
（例、「DHFOEFRGEJIC」）

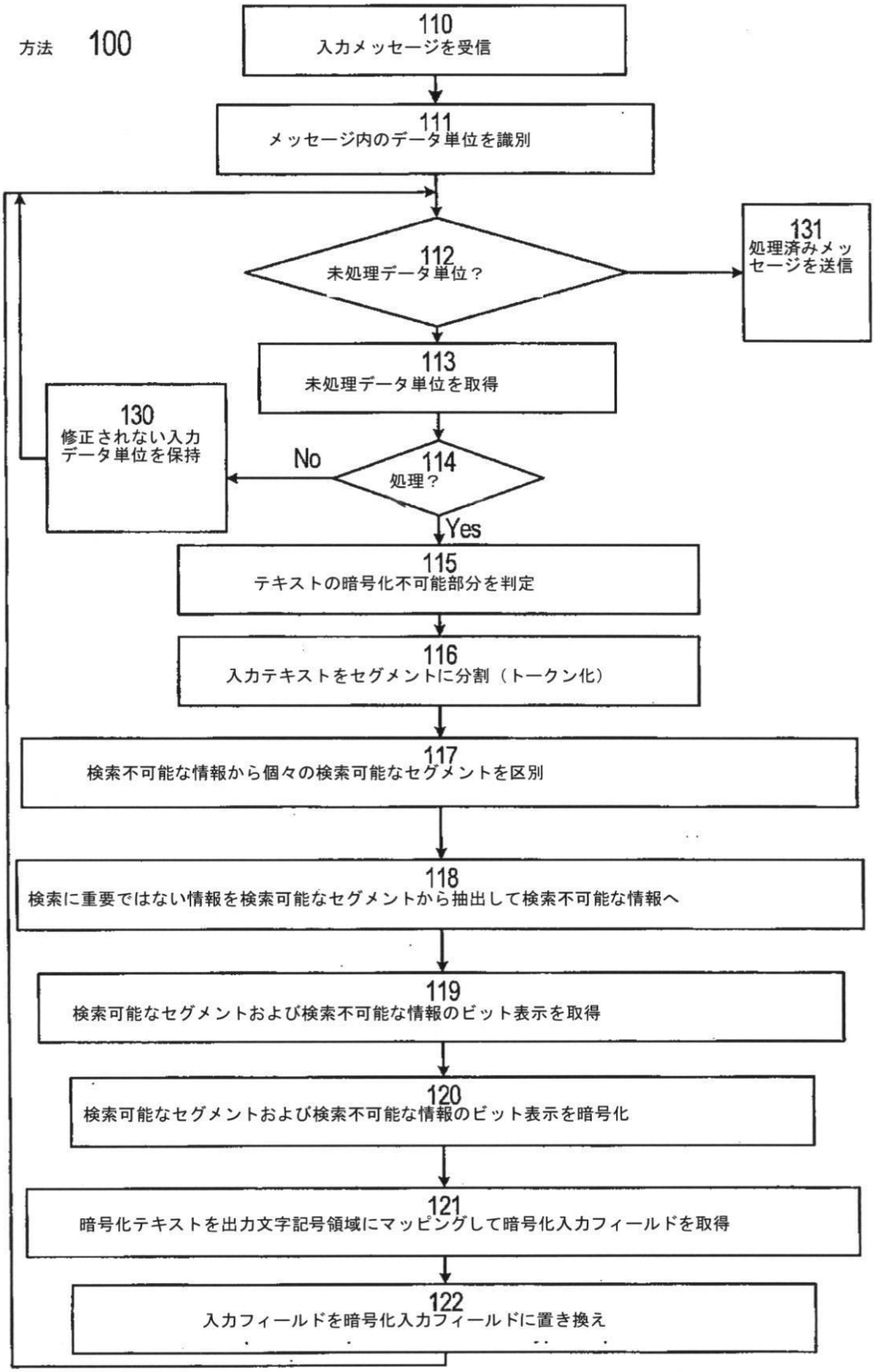
【図２】


私設ネットワーク
公衆ネットワーク
200
トラステッド
ワークステーション
230
クライアント
コンポーネント
240
(1) ユーザによるプレーンテキストデータの入力
阻止モジュール
210
データ保護モジュール
220
(2) 阻止モジュールによるデータの提示・阻止
(3) 設定ポリシーに応じたデータの暗号化および操作の可能性
公衆ネットワーク
250
(4) 暗号化データの非トラステッド対象ホストへの転送
非トラステッド
アプリケーション
サービスプロバイダ
260
データベース
270
(5) 非トラステッド対象ホストによる暗号化データの処理および格納

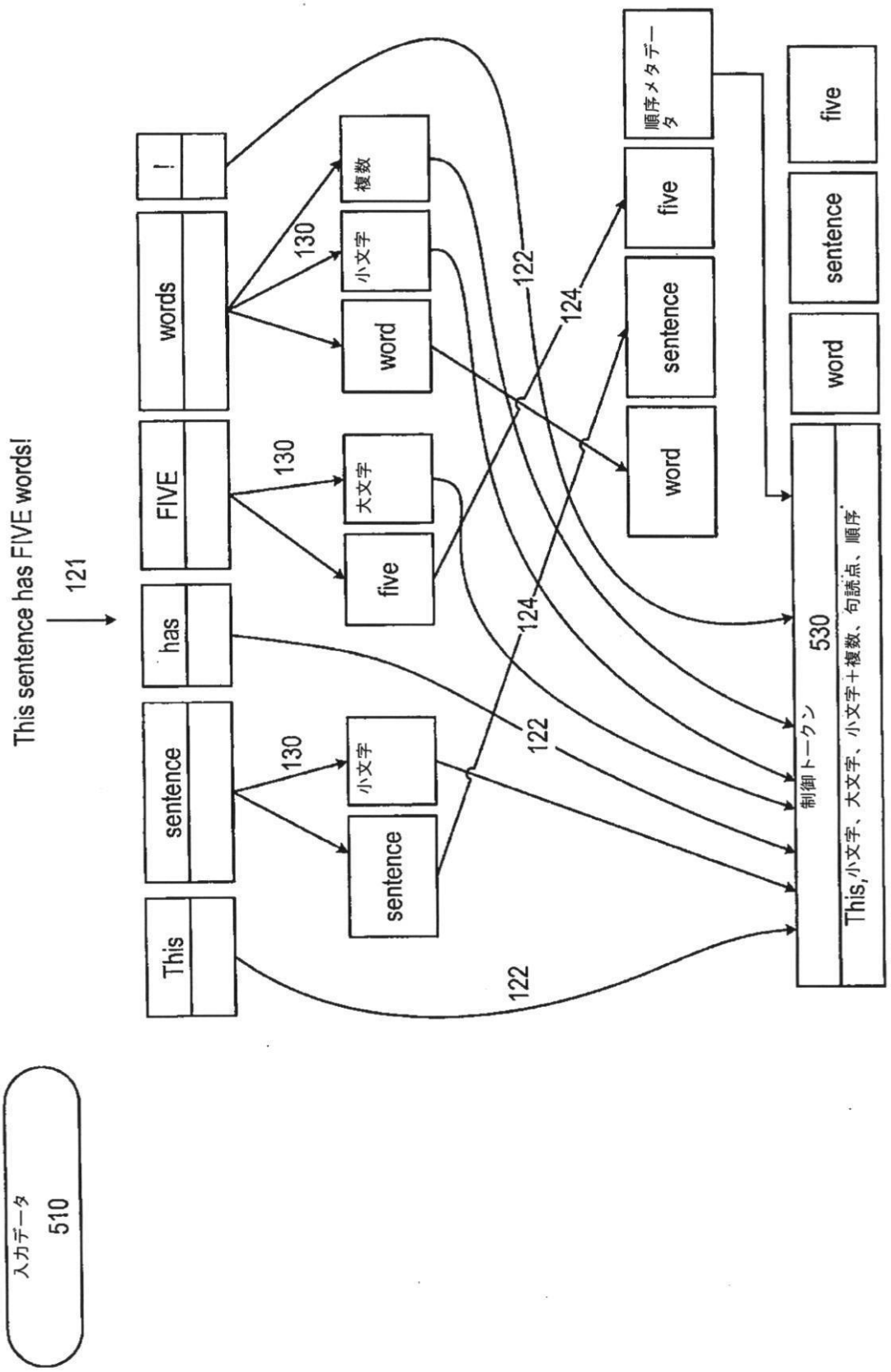
【図３】


非トラステッド
アプリケーション
サービスプロバイダ
260
データベース
270
(1) 非トラステッドサーバによる暗号化データの読み出し
公衆ネットワーク
250
(2) トラステッドワークステーションへの暗号化データの送信
200
阻止モジュール
210
データ保護モジュール
220
(3)設定ポリシーに応じたデータの復号化および操作の可能性
公衆ネットワーク
私設ネットワーク
(4) トラステッドワークステーションへのプレーンテキストデータの転送
トラステッド
ワークステーション
230
クライアント
240
(5)ユーザによるプレーンテキストデータの受信

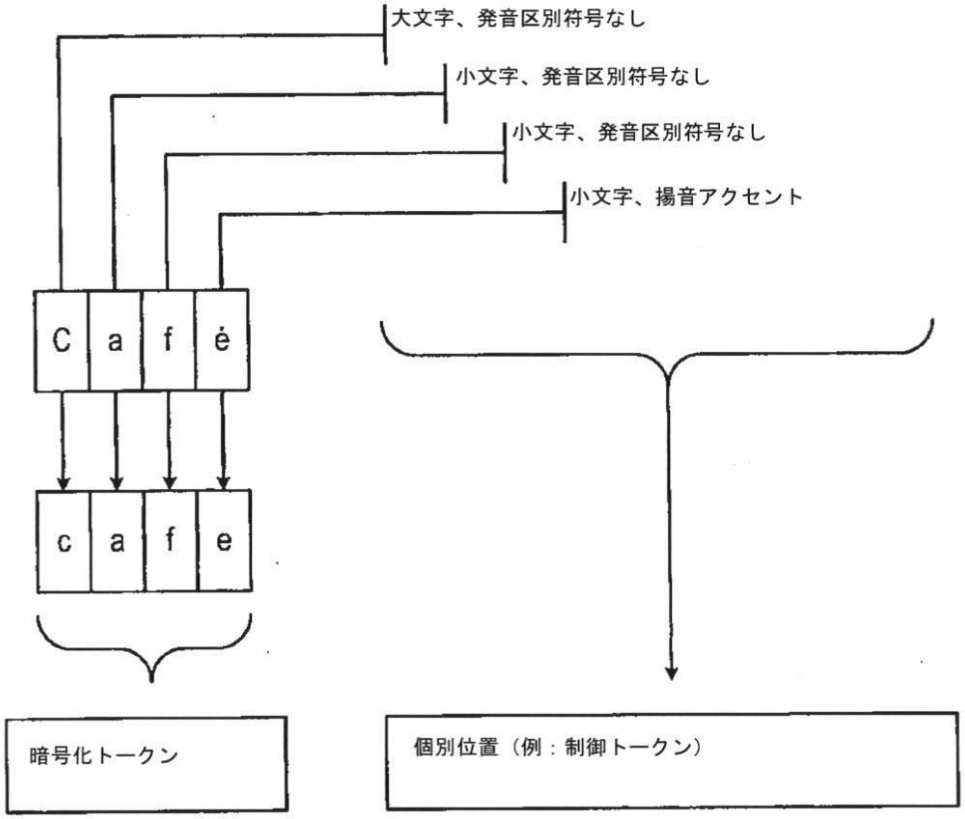
【図４】


方法 100
110
入力メッセージを受信
111
メッセージ内のデータ単位を識別
112
未処理データ単位？
131
処理済みメッセージを送信
113
未処理データ単位を取得
130
修正されない入力データ単位を保持
No
114
処理？
Yes
115
テキストの暗号化不可能部分を判定
116
入力テキストをセグメントに分割（トークン化）
117
検索不可能な情報から個々の検索可能なセグメントを区別
118
検索に重要ではない情報を検索可能なセグメントから抽出して検索不可能な情報へ
119
検索可能なセグメントおよび検索不可能な情報のビット表示を取得
120
検索可能なセグメントおよび検索不可能な情報のビット表示を暗号化
121
暗号化テキストを出力文字記号領域にマッピングして暗号化入力フィールドを取得
122
入力フィールドを暗号化入力フィールドに置き換え

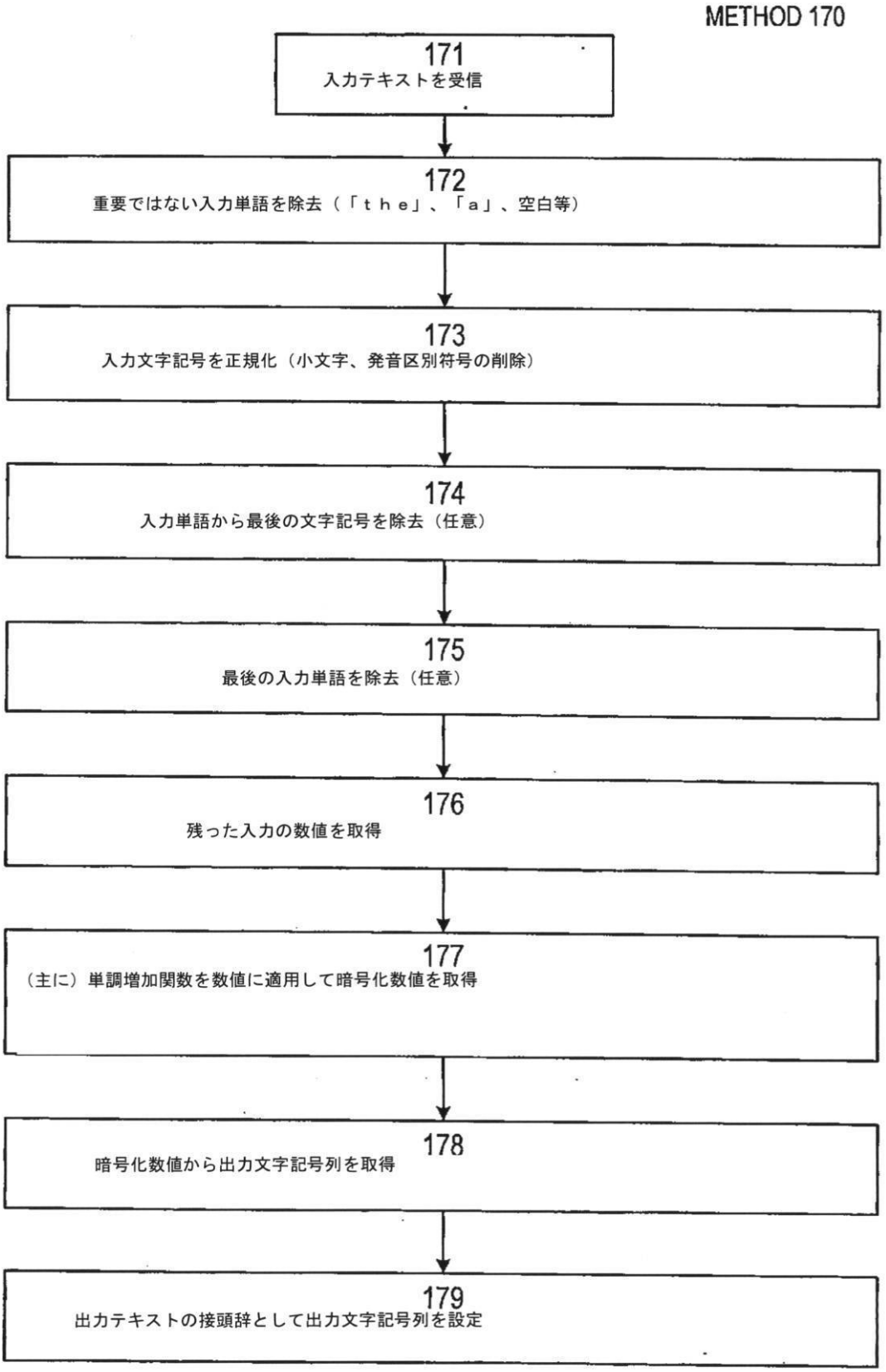
【図５】


入力データ
510
This sentence has FIVE words!
121
This
sentence
has
FIVE
words
!
小文字
sentence
小文字
five
複数
小文字
word
130
130
130
122
122
122
124
124
順序メタデータ
five
sentence
word
five
sentence
word
制御トークン
530
This,小文字、大文字、小文字＋複数、句読点、順序
word
sentence
five

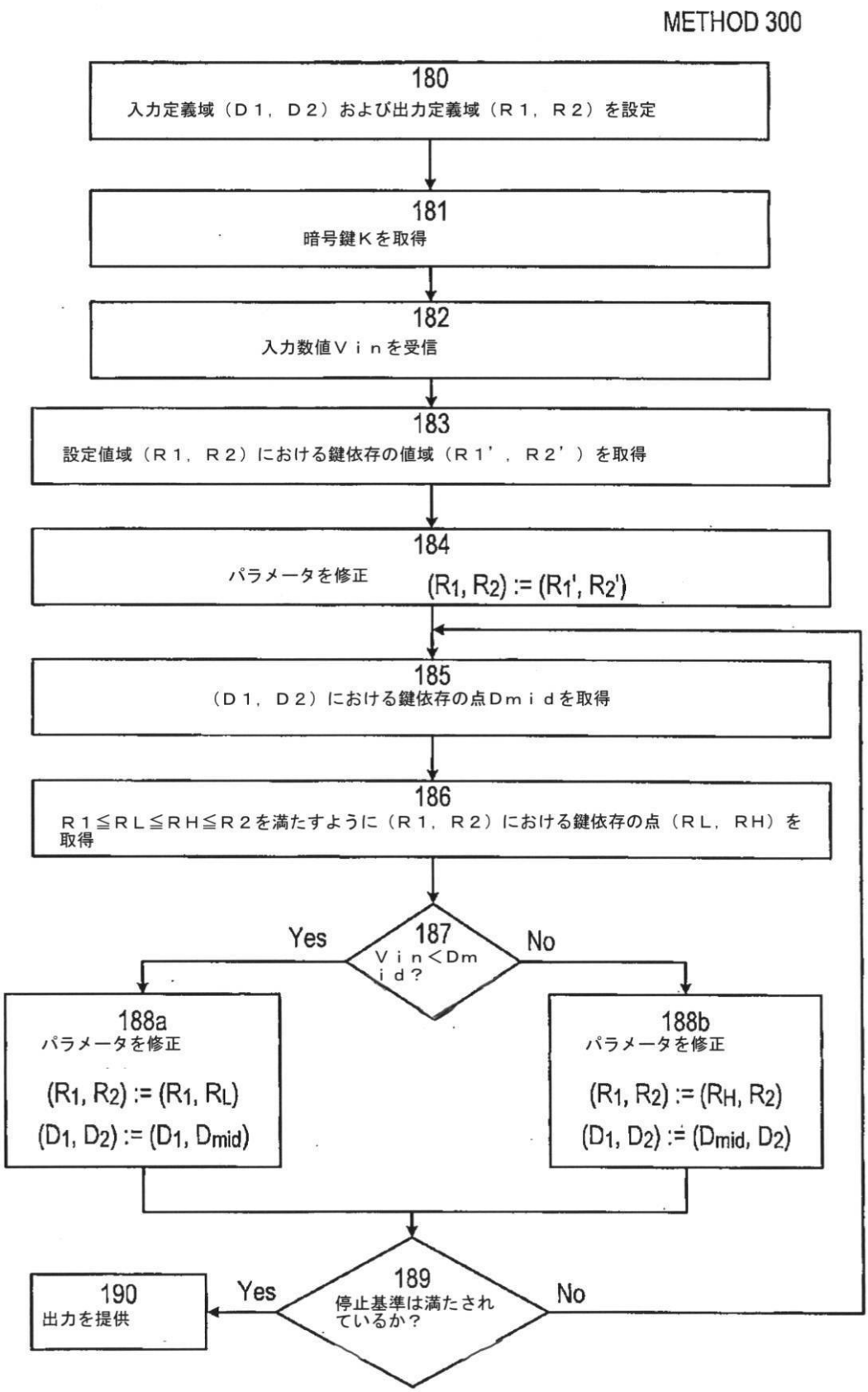
【図６】


大文字、発音区別符号なし
小文字、発音区別符号なし
小文字、発音区別符号なし
小文字、揚音アクセント
C a f é
c a f e
暗号化トークン
個別位置（例：制御トークン）

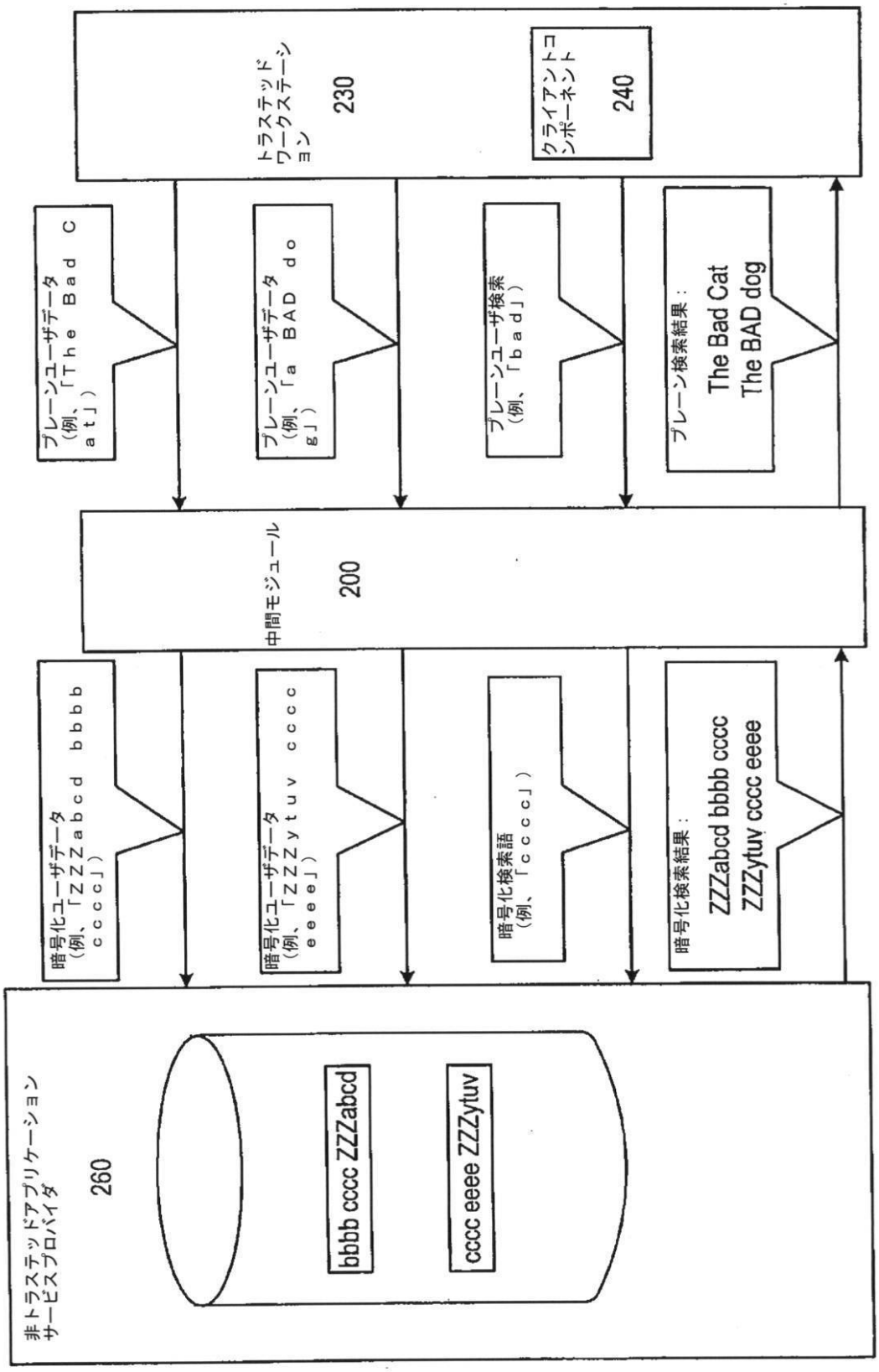
【図７】


METHOD 170
171
入力テキストを受信
172
重要ではない入力単語を除去（「ｔｈｅ」、「ａ」、空白等）
173
入力文字記号を正規化（小文字、発音区別符号の削除）
174
入力単語から最後の文字記号を除去（任意）
175
最後の入力単語を除去（任意）
176
残った入力の数値を取得
177
（主に）単調増加関数を数値に適用して暗号化数値を取得
178
暗号化数値から出力文字記号列を取得
179
出力テキストの接頭辞として出力文字記号列を設定

【図８】


METHOD 300
180
入力定義域（Ｄ１，Ｄ２）および出力定義域（Ｒ１，Ｒ２）を設定
181
暗号鍵Ｋを取得
182
入力数値Ｖｉｎを受信
183
設定値域（Ｒ１，Ｒ２）における鍵依存の値域（Ｒ１’，Ｒ２’）を取得
184
パラメータを修正
(R1, R2) := (R1', R2')
185
（Ｄ１，Ｄ２）における鍵依存の点Ｄｍｉｄを取得
186
Ｒ１≦ＲＬ≦ＲＨ≦Ｒ２を満たすように（Ｒ１，Ｒ２）における鍵依存の点（ＲＬ，ＲＨ）を取得
187
Ｖｉｎ＜Ｄｍｉｄ？
Yes
No
188a
パラメータを修正
(R1, R2) := (R1, RL)
(D1, D2) := (D1, Dmid)
188b
パラメータを修正
(R1, R2) := (RH, R2)
(D1, D2) := (Dmid, D2)
189
停止基準は満たされているか？
Yes
No
190
出力を提供

【図１０】


トラステッドワークステーション
230
クライアントコンポーネント
240
プレーンユーザデータ（例、「The Bad Cat」）
プレーンユーザデータ（例、「a BAD dog」）
プレーンユーザ検索（例、「bad」）
プレーン検索結果：
The Bad Cat
The BAD dog
中間モジュール
200
暗号化ユーザデータ（例、「ZZZabcd bbbb cccc」）
暗号化ユーザデータ（例、「ZZZytuv cccc eeee」）
暗号化検索語（例、「cccc」）
暗号化検索結果：
ZZZabcd bbbb cccc
ZZZytuv cccc eeee
非トラステッドアプリケーションサービスプロバイダ
260
bbbb cccc ZZZabcd
cccc eeee ZZZytuv

フロントページの続き

(72)発明者　タル、マーヤン
イスラエル国　８４５１５　ベエル　シェバ　ダビデ　ハレウベニ　ストリート　１５
(72)発明者　ラハブ、アビアド
イスラエル国　６５２０７　テル　アビブ　アハド　ハーム　ストリート　９８

審査官　金沢　史明

(56)参考文献　特開２００８－３０１３３５（ＪＰ，Ａ）
特開２００７－２５１５８５（ＪＰ，Ａ）
特開２００７－２４３６５０（ＪＰ，Ａ）
特開２００５－１３０３５２（ＪＰ，Ａ）
米国特許第０５９５８００６（ＵＳ，Ａ）
特開２００５－２４２７４０（ＪＰ，Ａ）
特開２００５－２８４９１５（ＪＰ，Ａ）
特開２００１－１４７９３４（ＪＰ，Ａ）
特開２００４－１０１９０５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／００
Ｈ０３Ｍ　　７／００
Ｈ０４Ｎ　　７／００