(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-4516
(P2010-4516A)
(43)公開日　平成22年1月7日(2010.1.7)

(51)Int.Cl.		F I			テーマコード（参考）
H04L 9/08	(2006.01)	H04L	9/00	601B	5J104

審査請求 有　請求項の数 25　ＯＬ　（全 21 頁）

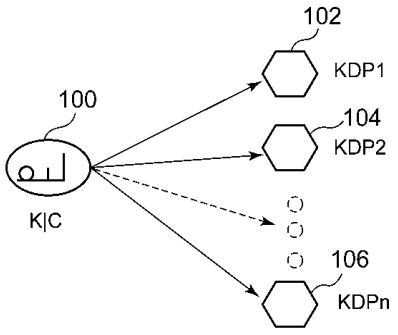
(21)出願番号　特願2008-267996 (P2008-267996)
(22)出願日　平成20年10月16日 (2008.10.16)
(31)優先権主張番号　12/142009
(32)優先日　平成20年6月19日 (2008.6.19)
(33)優先権主張国　米国 (US)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】暗号鍵および証明書の配備および配付の自動妥当性検査および実行のためのシステムおよび方法

(57)【要約】
【課題】　暗号鍵および証明書の配備および配付の自動妥当性検査および実行のためのシステムおよび方法を提供することにある。
【解決手段】　暗号鍵および証明書の配備および配付の自動妥当性検査および実行のための方法は、１つまたは複数の鍵を提供するステップと、１つまたは複数の鍵配備ポイントを提供するステップと、１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき１つまたは複数の鍵のそれぞれの行列またはパターン・マッピングに基づいて自動的に１つまたは複数の鍵を１つまたは複数の鍵配備ポイントに配布するステップとを含む。
【選択図】　図１



【特許請求の範囲】
【請求項１】
暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するための方法であって、
１つまたは複数の鍵を提供するか、あるいは配備中にオンデマンドで１つまたは複数の鍵を自動的に生成するステップと、
１つまたは複数の鍵配備ポイントを提供するステップと、
前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれのパターン・ベース・マッピングに基づいて自動的に前記１つまたは複数の鍵を前記１つまたは複数の鍵配備ポイントに配布するステップと、
を含む方法。
【請求項２】
前記１つまたは複数の鍵のうちの特定の鍵に証明書が添付される、請求項１に記載の方法。
【請求項３】
前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれのマッピングが、前記１つまたは複数の鍵のそれぞれが前記１つまたは複数の鍵配備ポイントのいずれにマッピングされるかを示す行列を作成することを含む、請求項１に記載の方法。
【請求項４】
自動的に前記１つまたは複数の鍵を前記１つまたは複数の鍵配備ポイントに配布する前記ステップが１つまたは複数の配備パターンを使用して実行され、それぞれの配備が、配備ソースと、配備宛先と、前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布の配備要件を表す前記配備パターンとを含む、請求項１に記載の方法。
【請求項５】
前記配備パターンが、共通共用、共通固有、秘密および公開鍵共用、秘密および公開鍵固有、秘密共用、公開共用、秘密固有公開共用を含むグループからのものである、請求項４に記載の方法。
【請求項６】
前記配備パターンのうちの既存のものに新しい配備パターンを追加できるか、または前記配備パターンのうちの既存のものの合成により新しい配備パターンを作成できる、請求項４に記載の方法。
【請求項７】
前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布を１つまたは複数のルールと対照して妥当性検査するステップをさらに含む、請求項１に記載の方法。
【請求項８】
鍵配備ポイントで検出されるものが前記鍵配備ポイント内にあると期待または想定されるものであることを検証するためのチェック機能として前記配備行列を使用して、前記１つまたは複数の鍵の前記配布を妥当性検査するステップをさらに含む、請求項３に記載の方法。
【請求項９】
前記１つまたは複数の鍵のうちの少なくとも１つを修正することによるか、前記１つまたは複数の鍵配備ポイントのうちの少なくとも１つを修正することによるか、あるいは前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれの前記マッピングを修正することによるか、もしくはこれらの修正の組み合わせにより、前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布を修正するステップをさらに含む、請求項１に記載の方法。
【請求項１０】
配備行列のデルタ計算のアルゴリズムを使用して、ポリシーの変更後または管理者によ

る変更後に適切な前記鍵または証明書あるいはその両方により前記鍵配備ポイントをセットアップさせるために鍵ライフサイクル・マネージャが行わなければならない変更を決定するステップをさらに含む、請求項１に記載の方法。
【請求項１１】
暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するためのコンピュータ可読コンピュータ・プログラム・コードと、
コンピュータに方法を実装させるための命令とを含み、前記方法が、
１つまたは複数の鍵を提供するステップと、
１つまたは複数の鍵配備ポイントを提供するステップと、
前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれのマッピングに基づいて自動的に前記１つまたは複数の鍵を前記１つまたは複数の鍵配備ポイントに配布するステップと、
を含む、コンピュータ・プログラム。
【請求項１２】
前記１つまたは複数の鍵のうちの特定の鍵に証明書が添付される、請求項１１に記載のコンピュータ・プログラム。
【請求項１３】
前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれのマッピングが、前記１つまたは複数の鍵のそれぞれが前記１つまたは複数の鍵配備ポイントのいずれにマッピングされるかを示す行列を作成することを含む、請求項１１に記載のコンピュータ・プログラム。
【請求項１４】
自動的に前記１つまたは複数の鍵を前記１つまたは複数の鍵配備ポイントに配布する前記ステップが１つまたは複数の配備パターンを使用して実行され、それぞれの配備パターンが、前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布の配備パターンを表す、請求項１１に記載のコンピュータ・プログラム。
【請求項１５】
前記配備パターンが、共通共用、共通固有、秘密および公開鍵共用、秘密および公開鍵固有、秘密共用、公開共用、秘密固有公開共用を含むグループからのものである、請求項１４に記載のコンピュータ・プログラム。
【請求項１６】
前記配備パターンのうちの既存のものを合成することにより新しい配備パターンが作成される、請求項１４に記載のコンピュータ・プログラム。
【請求項１７】
前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布を１つまたは複数のルールと対照して妥当性検査するステップをさらに含む、請求項１１に記載のコンピュータ・プログラム。
【請求項１８】
前記１つまたは複数の鍵のうちの少なくとも１つを修正することによるか、前記１つまたは複数の鍵配備ポイントのうちの少なくとも１つを修正することによるか、あるいは前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれの前記マッピングを修正することによるか、もしくはこれらの修正の組み合わせにより、前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布を修正するステップをさらに含む、請求項１１に記載のコンピュータ・プログラム。
【請求項１９】
暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するためのシステムであって、
１つまたは複数のコンピュータ・メモリ記憶装置と通信状態にある処理装置を含むコンピューティング・ネットワークを含み、
前記コンピューティング・ネットワークが、暗号鍵および証明書の配備および配付の自

動妥当性検査および実行のための方法を実装するようにさらに構成され、前記方法が、
１つまたは複数の鍵を提供するステップと、
１つまたは複数の鍵配備ポイントを提供するステップと、
前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれのマッピングに基づいて自動的に前記１つまたは複数の鍵を前記１つまたは複数の鍵配備ポイントに配布するステップと、
をさらに含む、システム。
【請求項２０】
前記１つまたは複数の鍵のうちの特定の鍵に証明書が添付される、請求項１９に記載のシステム。
【請求項２１】
前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれのマッピングが、前記１つまたは複数の鍵のそれぞれが前記１つまたは複数の鍵配備ポイントのいずれにマッピングされるかを示す行列を作成することを含む、請求項１９に記載のシステム。
【請求項２２】
自動的に前記１つまたは複数の鍵を前記１つまたは複数の鍵配備ポイントに配布する前記ステップが１つまたは複数の配備パターンを使用して実行され、それぞれの配備パターンが、前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布の配備パターンを表す、請求項１９に記載のシステム。
【請求項２３】
前記パターンが、共通共用、共通固有、秘密および公開鍵共用、秘密および公開鍵固有、秘密共用、公開共用、秘密固有公開共用を含むグループからのものである、請求項２２に記載のシステム。
【請求項２４】
前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布を１つまたは複数のルールと対照して妥当性検査するステップをさらに含む、請求項１９に記載のシステム。
【請求項２５】
前記１つまたは複数の鍵のうちの少なくとも１つを修正することによるか、前記１つまたは複数の鍵配備ポイントのうちの少なくとも１つを修正することによるか、あるいは前記１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき前記１つまたは複数の鍵のそれぞれの前記マッピングを修正することによるか、もしくはこれらの修正の組み合わせにより、前記１つまたは複数の鍵配備ポイントへの前記１つまたは複数の鍵の前記配布を修正するステップをさらに含む、請求項１９に記載のシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、暗号化に関し、詳細には、暗号鍵および証明書の配備および配付の自動妥当性検査および実行のためのシステムおよび方法に関する。
【背景技術】
【０００２】
鍵管理は、暗号鍵が適切なポリシーに応じて作成され、災害に備えてバックアップされ、時間通りにそれを必要とするシステムに配信され、適切な担当者の制御下に置かれ、おそらくそのライフサイクルの終了時に削除されるプロセスである。誤りがちな手動操作を回避するために、すべての鍵は好ましくは自動的に管理される。
【０００３】
しかし、この時点まで、鍵の配備および配布は、誤りがちな人間によってほとんど制御されてきた。具体的には、すべての必要な鍵が、必要とされる正確な位置に適切に配備され配布されるという保証はまったくなく、既存の鍵の配備がいかなるリスクも露呈しない

という保証もまったくない。これは、典型的なエンタプライズ鍵管理システム（enterprise key management system）が、何千もの鍵配備ポイント（key deployment point）ともに、何千もの鍵または証明書あるいはその両方を伴う可能性があり、その多くが規則正しく更新またはリフレッシュされるからである。このようなシステムでは、適時に効率よく鍵を配布するという管理作業は本質的に複雑で、誤りがちなものである。

【０００４】

公開鍵および証明書の管理には、周知の公開鍵インフラストラクチャ（ＰＫＩ：public key infrastructure）が使用される。これは、クライアントとＰＫＩサーバとのフェッチタイプの対話を含む。しかし、ＰＫＩサーバは、クライアントの位置における証明書配備を制御しない。これまで、鍵配備および配布プロセスを妥当性検査し実行するための自動方法は言うまでもなく、コンピュータなどのマシンが鍵配備を理解できるように鍵配備を表すための方法はまったく知られていない。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

鍵および証明書が、データを処理または保護するために使用される任意の場所に配備または配布され、必要とされないかまたは追跡されない場所には送られず、間違った配備または配布に関連するリスクを回避するためにもはや存在する必要がない場所から除去されるような、暗号鍵または証明書あるいはその両方の配備および配布の自動管理（すなわち、妥当性検査および実行）のためのシステムおよび方法が必要である。

【課題を解決するための手段】

【０００６】

従来技術の上述の欠点および不備は、模範的な一実施形態において、暗号鍵および証明書の配備および配付の自動妥当性検査および実行のための方法によって克服または軽減され、この方法は、１つまたは複数の鍵を提供するステップと、１つまたは複数の鍵配備ポイントを提供するステップと、１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき１つまたは複数の鍵のそれぞれのマッピングに基づいて自動的に１つまたは複数の鍵を１つまたは複数の鍵配備ポイントに配布するステップとを含む。

【０００７】

他の実施形態では、コンピュータ・プログラム（computer program product）は、暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するためのコンピュータ可読コンピュータ・プログラム・コードと、コンピュータに方法を実装させるための命令とを含み、この方法は、１つまたは複数の鍵を提供するステップと、１つまたは複数の鍵配備ポイントを提供するステップと、１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき１つまたは複数の鍵のそれぞれのマッピングに基づいて自動的に１つまたは複数の鍵を１つまたは複数の鍵配備ポイントに配布するステップとをさらに含む。

【０００８】

暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するためのシステムは、１つまたは複数のコンピュータ・メモリ記憶装置と通信状態にある処理装置を含むコンピューティング・ネットワークを含み、このコンピューティング・ネットワークは、暗号鍵および証明書の配備および配付の自動妥当性検査および実行のための方法を実装するようにさらに構成され、この方法は、１つまたは複数の鍵を提供するか、あるいは配備中にオンデマンドで１つまたは複数の鍵を自動的に生成するステップと、１つまたは複数の鍵配備ポイントを提供するステップと、１つまたは複数の鍵配備ポイントのそれぞれに対して配布すべき１つまたは複数の鍵のそれぞれのパターン・ベース・マッピングに基づいて自動的に１つまたは複数の鍵を１つまたは複数の鍵配備ポイントに配布するステップとをさらに含む。

【０００９】

模範的な図面を参照すると、同様の要素は複数の図面において同様の番号が付けられている。

【発明を実施するための最良の形態】
【００１０】
本明細書では、暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するためのシステムおよび方法が開示されている。手短に言えば、本発明は、エンタプライズ暗号鍵／証明書システムにおける自動鍵配備管理を伴う。これは、多数の鍵配備ポイント（ＫＤＰ：key deployment point）を有する集中鍵管理システム（ＣＫＭＳ：centralizedkey management system）と呼ばれる集中制御点を有する典型的な配布鍵／証明書管理システムに適用することができる。ＫＤＰは、たとえば、鍵を消費する任意の暗号化／暗号化解除装置などの鍵使用エンティティまたは証明書を必要とするサーバ上のアプリケーションを含む、多くのものにすることができる。また、ＫＤＰは、１つまたは複数の鍵使用エンティティ（ＫＵＥ：key use entity）用のエージェントとして供する軽量鍵管理ポイント（lightweightkey-management point）にすることもできる。鍵使用エンティティは、暗号動作（暗号化、暗号化解除、署名、検証）を実行するために鍵を使用する。したがって、本発明は、特定の鍵管理アーキテクチャにかかわらず、ある位置から１つまたは複数の他の位置への鍵の配備または配布の管理を伴う、多種多様なシナリオに適用される。証明書／鍵配布機能は、正しい場所で正しい時期に正しい鍵を有することを保証するための任意の鍵管理システム内の重大なコンポーネントである。本発明の証明書／鍵配備および配布システムおよび方法は証明書／鍵配備の集合を定義し、そのそれぞれは、証明書／鍵の集合から鍵配備ポイントの集合へのマッピングを表す。
【００１１】
本発明の自動鍵および証明書配備システムおよび方法は、サーバ・グループまたはアプリケーション・グループなどの鍵配備ポイントへの個々の鍵配備に取り組むために必要な管理労力を低減する。その代わりに、これらの配備は、１つの事前定義配備パターンまたは複数配備パターンの合成（composition）によって指定され、そのそれぞれ（パターン）はアプリケーション用の典型的な配備ユース・ケース（use-case）を表す。これは、管理者の手動対話なしに配備ポリシーに沿ったインフラストラクチャのセットアップおよび更新を容易にする。その利点は、分散アプリケーションの管理によって、配備ポイントまたは鍵の実際の数とは無関係の１つの配備パターンまたは複数配備パターンの合成として鍵および証明書に関するその必要性を表現できることである。配備エンジンは鍵配布のプロセスを自動化するだけでなく、配備に含まれる鍵／証明書がシステム内にまだ存在しない場合に実際の鍵生成を自動的に開始することもできる。加えて、既存の配備は、変化（たとえば、配備ポイントの追加および除去、鍵の追加および除去およびリフレッシュ）に対して自動的に反応する。典型的なエンタプライズ鍵管理システムが数十から数百の鍵使用エンティティと、潜在的に数千または数百万の鍵を含有できることを考慮すると、鍵配備および配布プロセスは、好ましくは、本発明により自動的に管理される。
【００１２】
以下の定義は本発明全体を通して適用される。
【００１３】
Ｋｅｙ（Ｋ）は、有効期間（「暗号期間（cryptoperiod）」）を有する物理的な鍵を意味する。鍵は、共通鍵（secret key）（対称鍵（symmetric key））、秘密鍵（private key）、公開鍵（public key）、または秘密鍵／公開鍵のペアにすることができる。
【００１４】
Ｃｅｒｔｉｆｉｃａｔｅ（Ｃ）は、公開鍵を含み、認証局の秘密鍵によって署名された証明書を意味する。
【００１５】
ＫｅｙＡｎｄＣｅｒｔｉｆｉｃａｔｅ（Ｋ｜Ｃ）は、鍵と証明書のペアを意味する。
【００１６】
ＫｅｙＳｅｔ（K^S）は、共通特性を有する関連する（物理的または論理的な）鍵の集合を意味する。
【００１７】

ＣｅｒｔｉｆｉｃａｔｅＳｅｔ（C^S）は、共通特性を有する関連する証明書の集合を意味する。

【００１８】

ＫｅｙＡｎｄＣｅｒｔｉｆｉｃａｔｅＳｅｔ（（Ｋ｜Ｃ）^{S}）は、鍵と証明書の関連バンドルの集合を意味する。

【００１９】

ＫｅｙＤｅｐｌｏｙｍｅｎｔＰｏｉｎｔ（ＫＤＰ）は、鍵を要求するアプリケーションが鍵にアクセスする際に使用するインターフェースを意味する。

【００２０】

ＫｅｙＤｅｐｌｏｙｍｅｎｔＰｏｉｎｔＳｅｔ（KDP^S）は、関連するＫＤＰの集合またはバンドルを意味する。

【００２１】

直観的には、鍵配備は、Ｋｅｙ、Ｃｅｒｔｉｆｉｃａｔｅ、ＫｅｙＡｎｄＣｅｒｔｉｆｉｃａｔｅ、ＫｅｙＳｅｔ、ＣｅｒｔｉｆｉｃａｔｅＳｅｔ、またはＫｅｙＡｎｄＣｅｒｔｉｆｉｃａｔｅＳｅｔと、ＫＤＰまたはKDP^Sとの関連付けまたはマッピングと見なすことができる。より総称的なものにするために、鍵配備は、証明書／鍵エンティティ（CK^Eによって示す）の配列と、ＫＤＰエンティティ（KDP^Eによって示す）の配列とのマッピング（Ｍ）として定義される。証明書／鍵エンティティ（CK^E）は、Ｋ、K^S、Ｃ、C^S、Ｋ｜Ｃ、または（Ｋ｜Ｃ）^{S}にすることができ、ＫＤＰエンティティ（KDP^E）は、ＫＤＰまたはKDP^Sにすることができる。鍵配備マッピングＭは以下のように定義することができる。

Ｍ：CK^E［　］→KDP^E［　］　　式１

ここで、CK^Eは、｛Ｋ，K^S，Ｃ，C^S，Ｋ｜Ｃ，（Ｋ｜Ｃ）^{S}｝の集合内からのものであり、KDP^Eは、｛ＫＤＰ，KDP^S｝の集合内からのものである。

【００２２】

図１を参照すると、Ｗｅｂサーバ（たとえば、アプリケーション・サーバ、プロキシ・サーバ、スタティックＷｅｂコンテンツ・サーバ）の複合体が複数のコンピューティング・システムにわたって配備されている。これらのサーバには何らかのアルゴリズムに応じて要求が送信されるが、本発明のためには、すべてのサーバは、それに送信された要求に対して同じように見える／動作する／応答することが期待される。さらに、各システムは、他のシステムの１つまたは複数が動作していない場合でも動作できる必要がある。Ｗｅｂサーバ（ＨＴＴＰプロトコルにより要求を供する）は、ＨＴＴＰＳ（すなわち、ＳＳＬ／ＴＳＬ保護）通信をサポートすることが期待される。少なくとも、サーバ側認証が必要である。したがって、各Ｗｅｂサーバごとに、そのＷｅｂサーバが使用するための秘密鍵（およびそれに対応する添付証明書または公開鍵を保持する自己署名証明書）が構成される。すべてのサーバは同じサーバと見なされることが期待されるので、各Ｗｅｂサーバ内に同じ秘密鍵＋証明書を構成しなければならない。このシナリオでは、秘密鍵＋証明書はＫ｜Ｃ１００であり、複数のＷｅｂサーバはＫＤＰ（ＫＤＰ１　１０２、ＫＤＰ２　１０４、・・・ＫＤＰｎ　１０６）であり、ＫＤＰ　１０２～１０６のそれぞれにＫ｜Ｃ１００が配備されることが必要である。

【００２３】

複数暗号鍵マネージャ（ＥＫＭ：Encryption-KeyManager）は、鍵／証明書をテープ・ドライブに供するＩＢＭモジュールであり、テープ・ドライブのバンクをサポートするためまたはＥＫＭインスタンス間のフェイルオーバを提供するために、クライアント・インストールにおいて一般的なものである。ＩＢＭ　ＴＳ１０４０（ＬＴＯ－４）ドライブは、鍵ストア（keystore）に保管されたＡＥＳまたは３ＤＥＳ対称鍵を使用する。これらの鍵は、テープ暗号化の読み取りまたは書き込み動作中にＥＫＭによってアクセスされる。ＥＫＭは、キー・マテリアル（key material）を保管するために鍵ストアの幅広い配列を使用することができる。その場合、ＫＤＰは、ＥＫＭが把握している鍵ストアの位置であり、そこからＥＫＭは適切な鍵を取得する。

【００２４】
図２を参照すると、２つの鍵集合が存在し、一方の集合２００はＡＥＳ鍵で構成され、もう一方の集合２０２は３ＤＥＳ鍵を含む。テープ・ドライブに供するための４つのＥＫＭが存在し、それぞれが１つの鍵ストアを有する。４つのＥＫＭはペアになって動作し、各ペアは同じ鍵集合を使用し、同じテープ・ドライブに供する（典型的なフェイルオーバ構成）。鍵配備を記述するために複数の方法が存在し、たとえば、１つの方法は４つのＫＤＰ２０４～２１０をセットアップすることであり、各鍵集合２００、２０２は２つのＫＤＰ２０４～２１０に進む。この方式の欠点は、どのＫＤＰ２０４～２１０がグループ化されるかについて明示的ではないことである。したがって、図２に例示されている通り、より適切な解決策は２つのＫＤＰ集合２１２、２１４を使用することであり、そのそれぞれがフェイルオーバ関係を表す。

【００２５】
鍵配備方式は、（１）証明書／鍵エンティティの配列によって実施される鍵配備ソース（key deployment source）と、（２）ＫＤＰエンティティの配列によって実施される鍵配備宛先（key deployment destination）と、（３）ソースから宛先への鍵／証明書の配付を指定する配備パターン（deployment pattern）という３つのコンポーネントを含む。配備パターンは以下のように定義される。上述の通り、鍵配備は、配備ソースと配備宛先とのマッピングを含む。配備ソースは、Ｋ、K^S、Ｃ、C^S、Ｋ｜Ｃ、$(K|C)^S$などの証明書／鍵エンティティの配列であり、配備宛先は、ＫＤＰまたはＫＤＰ集合などのＫＤＰエンティティの配列である。配備ソースがｍ個のエンティティを有し、配備宛先がｎ個のエンティティを有する場合、そのマッピングはｍ×ｎの０－１行列によって表すことができ、「１」は「そのＫＤＰ内に鍵が存在しなければならないかまたは配備されなければならない」ことを示し、「０」は「そのＫＤＰに鍵が存在しないかまたは配備されていない」ことを示す。この行列は、配備ソースおよび配備宛先内のエンティティを接続するグラフのエッジを表す隣接生起行列（adjacent incidence matrix）である。より具体的には、配備行列内の各列は配備ソース内の各エンティティに対応し、各行は配備宛先内の対応するＫＤＰまたはＫＤＰ集合に対して「イエス／ノー（yes-or-no）」の配備判断（deploying decision）を指定する。したがって、ｉ番目の行およびｊ番目の列にある１は、配備ソース内のｉ番目の鍵（または鍵集合）が配備宛先内のｊ番目のＫＤＰ（またはＫＤＰ集合）に配備されなければならないことを規定する。

【００２６】
第１の例では、単一のエンティティが配備ソース内にあり、ｎ個のエンティティが配備宛先内にある。したがって、１×ｎの鍵配備行列は［１　１　・・・　１］になり、これはｎ個のＫＤＰのそれぞれにその鍵が配備されることを意味する。第２の例では、２つのエンティティが配備ソース内にあって、一方のエンティティがＡＥＳ鍵であり、もう一方のエンティティが３ＤＥＳ鍵であり、それぞれＡＥＳ鍵および３ＤＥＳ鍵をホストとして処理する２つのエンティティ、すなわち、ＫＤＰ集合が存在する。したがって、２×２の配備行列は以下のようになる。

【数１】

$$D=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

フェイルオーバ用のＫＤＰのペアをグループ化する代わりに、ＫＤＰ集合を４つのエンティティとして定義する場合、２×４の配備行列は以下のようになる。

【数２】

$$D=\begin{pmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{pmatrix}$$

この表現は前の表現より比較的複雑であり、フェイルオーバのためのＫＤＰのペアリングに関する情報が失われている。これにより、配備宛先内のエンティティとしてＫＤＰ集合を使用する必要性が正当化される。

【００２７】

集中暗号化鍵管理システムでは、多くの鍵配備インスタンスが存在することになり、そのそれぞれは専用の配備ソースおよび配備宛先を有する特定のアプリケーションに対応する。集約された配備ソースは、システムによって管理されるすべての鍵／鍵集合を含む大規模鍵集合と見なされ、集約された配備宛先は、システムにとって既知の各ＫＤＰから構成される大規模ＫＤＰ集合と見なされる。ＫＤＰは、鍵ライフサイクル・マネージャ（ＫＬＭ：key lifecycle manager）の制御下にある鍵ストアにすることができる。その結果行われるシステム全体の配備の視覚化は、大規模行列、最も可能性の高いものとして、超疎行列（very sparse matrix）に基づいて行うことができる。

【００２８】

配備用の総称行列表現では、典型的なアプリケーション・シナリオに関する配備パターンを定義することができる。集中鍵管理システムに関するユース・ケースの詳細な検討に基づいて、以下の一般的な配備パターンを識別することができ、そのそれぞれを本明細書では「配備パターン（deployment pattern）」と呼ぶ。

【００２９】

共通共用（secret shared）：　１つの共通鍵または共通鍵集合を複数のＫＤＰ（たとえば、サーバ）に配備する必要がある。たとえば、ある位置が暗号化の作業を実行しており、他の位置が暗号化解除を実行しており、どちらの位置も同じ共通鍵（対称鍵）を使用する必要がある。「共通共用」パターンのグラフは、上述の第１の例とまったく同じである（鍵は共通鍵であることに留意されたい）。１×ｎの配備行列は、すべて１の行になる。

【００３０】

共通固有（secret unique）：　これは、たとえば、暗号化／暗号化解除などの作業のために、異なる共通鍵または鍵集合を異なる位置に１対１の対応で配備する必要性を指す。これは、対角行列によって特徴付けられ、上記で示した第２の例のシナリオを記述するものである。

【００３１】

秘密および公開鍵共用（private and public keyshared）：　この配備グラフ（および行列）は、図１に関して上記で示した第１の例と同じである。秘密鍵と公開鍵のペア（またはペアの集合）が複数のＫＤＰに、おそらく証明書付きで配備されることに留意されたい。

【００３２】

秘密および公開鍵固有（private and public keyunique）：　このパターンは、上記で示した第２の例と同様のものである。秘密鍵と公開鍵のペアとして各鍵集合を定義すると、配備グラフ（および行列）は、図２に関して上記で示した第２の例とまったく同じになる。

【００３３】

秘密共用（private shared）：　これは、上記の共通共用シナリオと同じであるが、その鍵は共通鍵の代わりに秘密鍵である。配備グラフ（および行列）は変化なしのままである。ユース・ケースは、ビジネス・パートナ交換データ・インポートのために１つの秘密鍵を複数のＫＤＰに配備することである。

【００３４】
公開共用（public shared）：　これは、上記の共通共用シナリオと同じであるが、その鍵は共通鍵の代わりに公開鍵（または証明書）である。配備グラフ（および行列）は変化なしのままである。ユース・ケースは、ビジネス・パートナ交換データ・エクスポートのために１つの公開鍵を複数のＫＤＰに配備することである。
【００３５】
秘密固有公開共用（private unique publicshared）：　サーバ・クラスタのために自己署名証明書付きの秘密鍵／公開鍵のペアを配備するシナリオでは、各秘密鍵はそれに対応するＫＤＰサーバのみに進むことになり、おそらく自己署名証明書付きの公開鍵はクラスタ内のすべてのＫＤＰサーバに配備されることになる。図３はこのシナリオを例示している。図３では、配備ソース３００～３１０の集合を｛秘密鍵１，秘密鍵２，秘密鍵３，公開鍵１，公開鍵２，公開鍵３｝と定義し、宛先３１２～３１６を｛ＫＤＰ１，ＫＤＰ２，ＫＤＰ３｝と定義する場合、結果として得られる配備行列は以下のようになる。
【数３】

$$D=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\\1&1&1\\1&1&1\\1&1&1\end{pmatrix}$$

この行列は、どの鍵（複数も可）がどのＫＤＰ（複数も可）に進むかに関する実際の情報を収集し、配備パターンは、鍵（複数も可）およびＫＤＰ（複数も可）の数とは無関係に総称的にマッピングを記述するだけである。
【００３６】
より多くの配備パターンをサポートするために、妥当なユース・ケースに基づいて新しいパターンを導入することにより、前述の事前定義パターンを強化することができる。また、既存のパターンから新しいパターンを作成することにより、事前定義パターンを拡張することも可能である。
【００３７】
鍵配備は、管理者またはエージェントによって手動で作成または修正することができ、あるいはアプリケーションの要求によって自動的に生成または変更することもできる。鍵管理システムは、その妥当性を保証し、以下に定義する模範的な鍵配布ポリシーに準拠するために、鍵配備を連続的にモニターする。候補鍵配備が鍵管理システムによって受け入れられる前に、その配備は、（１）配備ソースは空であってはならない、（２）配備宛先は空であってはならない、および（３）配備行列は少なくとも１つの１を含まなければならないという３つの非ヌル・ルールと対照して妥当性検査される。そうでなければ、配備指定は有効ではなく、鍵管理システムによって受け入れられない。
【００３８】
簡単にするため、配備行列内のすべてゼロの行とすべてゼロの列を除去することが推奨される。すべてゼロの各行は、対応する鍵または鍵集合を配備ソースから削除することによって除去することができ、すべてのゼロの各列は、対応するＫＤＰまたはＫＤＰ集合を削除することによって除去することができる。ある鍵配備内の同じＫＤＰに同じ鍵が複数回配備される必要性を回避するために、配備ソース内のエンティティと宛先内のＫＤＰエンティティを作成する際に注意しなければならない。特定の鍵が鍵配備ソース内に複数回現れてはならず、特定のＫＤＰが鍵配備宛先内に複数回現れてはならないことが規定されている。鍵管理システム内の実際の鍵配備は、鍵配布に関する企業または部門全域にわたるセキュリティ・ポリシーによって支配される場合が多い。鍵配布ポリシーの一例は、最

小リスク鍵配布要件を課すこと、すなわち、必要ではないＫＤＰにはけっして鍵を配備しないことである。もう一つの例は、誤ったＫＤＰにはけっして鍵を配布しないことであり、たとえば、ＫＤＰが署名を検証するためにのみ構成されている場合、共通鍵または秘密鍵をそのＫＤＰに配備することができない。鍵管理システムは、好ましくは、実際の鍵配備がそのポリシーに準拠しているかどうかを連続的にチェックする。有効な鍵配備が所定の位置になると、管理者、アプリケーション、またはポリシー執行者は、変化するビジネス要求を満たすために既存の鍵配備を修正することができる。前述の通り、鍵配備方式は、配備ソースと、配備宛先と、配備パターン（行列）という３つのコンポーネントを含み、そのそれぞれはアプリケーションの変化に対処するように変更または修正される可能性がある。その組み合わせについて０が存在する位置で見つかった鍵は、配備内のエラーと、修正すべきものを表す。したがって、配備行列に基づいて期待されるものを有するＫＤＰの実際の内容をチェックすることにより、鍵配備を妥当性検査することができる。

【００３９】

鍵配備ソース内のエンティティは、鍵または鍵集合のいずれかにすることができる。したがって、鍵配備ソースの変更に関して４通りのケース、すなわち、新しいエンティティとしての鍵（または鍵集合）の追加、鍵（または鍵集合）エンティティの削除、既存の鍵集合内への鍵の追加、既存の鍵集合からの鍵の削除が存在する。鍵配備ソース内に新しいエンティティとして鍵（または鍵集合）を追加するためには、配備行列は、配備宛先内の各ＫＤＰ（またはＫＤＰ集合）にその鍵または鍵集合を配備しなければならないかどうかなどの配備要件を指定するための追加の行を要求することになる。この新しい行は、管理者または追加動作を初期設定する他の任意のアクター（actor）のいずれかによって指定される。鍵管理システムは、新しい鍵配備表現を記録し、鍵または鍵集合の実際の配布を実装することになる。配備ソース内のエンティティである既存の鍵（または鍵集合）を削除するためには、それに応じて配備行列内の対応する行が削除されることになる。鍵管理システムは、その鍵の使用を停止し、それを除去するよう、ＫＤＰまたはＫＤＰ集合（またはＫＤＰを管理する管理者）に要求することになる。既存の鍵集合内に鍵を追加するためには、現在の配備行列を修正する必要はまったくない。それにもかかわらず、鍵管理システムは、その鍵集合に対応する行に応じて、新しい鍵をＫＤＰに配布するだけである。鍵配備ソース内に鍵集合を形成するために既存の鍵に（エンティティとして）鍵が追加されるケースにも同じ手順が適用される。既存の鍵集合から鍵を削除するためには、この場合も現在の配備行列を修正する必要はまったくない。鍵管理システムは、その鍵の使用を停止し、それをＫＤＰから除去するよう、その鍵集合に対応する行に応じてＫＤＰに要求することになる。

【００４０】

配備宛先内のエンティティは、ＫＤＰまたはＫＤＰ集合のいずれかにすることができる。したがって、鍵配備宛先の変更に関して４通りのケース、すなわち、新しいエンティティとしてのＫＤＰ（またはＫＤＰ集合）の追加、エンティティであるＫＤＰ（またはＫＤＰ集合）の削除、既存のＫＤＰエンティティ内へのＫＤＰの追加、既存のＫＤＰエンティティからのＫＤＰの削除が存在する。鍵配備宛先内に新しいエンティティとしてＫＤＰ（またはＫＤＰ集合）を追加するためには、配備行列は、既存の鍵または鍵集合をさらにそれに配備しなければならないかどうかなどの配備要件を指定するための追加の列を要求することになる。この新しい列は、配備パターンに応じて自動的に生成するか、または管理者によって指定することができる。鍵管理システムは、修正された鍵配備表現を記録し、鍵または鍵集合の実際の配布を実装することになる。配備宛先内のエンティティである既存のＫＤＰ（またはＫＤＰ集合）を削除するためには、それに応じて鍵行列表現内の対応する列が削除されることになる。削除された列内の１に対応するすべての鍵（鍵集合）はＫＤＰ（ＫＤＰ集合）から除去され、次にそのＫＤＰ（またはＫＤＰ集合）は配備宛先から除去されることになる。既存のＫＤＰ集合内にＫＤＰを追加するためには、現在の鍵配備行列を修正する必要はまったくない。それにもかかわらず、鍵管理システムは、そのＫＤＰ集合に対応する列に応じて、新しいＫＤＰに鍵を配布しなければならない。鍵配備宛

先内にＫＤＰ集合を形成するために既存のＫＤＰ（エレメントである）とともにＫＤＰがグループ化されるケースにも同じ手順が適用される。既存のＫＤＰ集合からＫＤＰを削除するためには、この場合も現在の鍵配備行列を修正する必要はまったくない。しかし、ＫＤＰ集合に一致する列内の１に対応する鍵は削除すべきＫＤＰについて除去され、次にそのＫＤＰはＫＤＰ集合から除去されることになる。ＫＤＰ集合が空である場合、配備宛先内のエンティティを削除するためのプロセスを開始しなければならない。

【００４１】

配備内の配備ソースおよび配備宛先の変更とは別に、配備行列自体も変更することができる。たとえば、前に配備されていなかったＫＤＰ（またはＫＤＰ集合）に鍵または鍵集合を配備する必要がある場合もある。これは、配備行列内の正しい位置で０を１に変更し、関連の鍵または鍵集合を正しいＫＤＰ（またはＫＤＰ集合内の各ＫＤＰ）に配布することに対応し、また、ＫＤＰ（またはＫＤＰ集合）への鍵または鍵集合の配備を取り消す必要がある場合もある。これは、配備行列内の正しい位置で１を０に変更し、ＫＤＰ（またはＫＤＰ集合）に対して関連の鍵または鍵集合の削除を要求することを伴う。

【００４２】

配備パターンによっては、ＫＤＰの追加により、生成し、そのＫＤＰに配備しなければならない新しい鍵のために、追加列の作成だけでなく、追加行の作成も必要になるであろう。「固有（unique）」というキーワードを有するすべてのパターンの場合、配備への新しいＫＤＰの追加は、そのＫＤＰに進む新しい鍵ソースも追加することを意味する。このため、この新しい鍵配備への変更は、２つのアクション、すなわち、鍵配備ソースおよび配備宛先の変更を伴う。管理者は新たに追加されたＫＤＰを指定し、システムは自動的に、必要な数の鍵を算出して生成し、配備パターンに応じて鍵配備を変更することになる。同様に、鍵配備パターンをあるパターンから他のパターンに変更することもできる。

【００４３】

行列演算に基づいて定義された鍵配備計算の公式化により、コンピュータ（またはコンピュータ・プログラム）は、本発明のシステムおよび方法により、保管された現在の鍵配布および所望の鍵配布を理解し、鍵配備の自動配備妥当性検査および実行を実行することができる。この作業を実施するために、現在の鍵配備から所望の鍵配備に変換するために必要なデルタ変化を計算するための２進セル単位マイナス演算を特徴とする汎用手順が提示される。鍵配備の変更を実行するための手順は以下のように定義することができる。

●提案された変更（たとえば、管理者による）を鍵配布ポリシーと対照して妥当性検査し、準拠している場合、その変更を受け入れる。

●新しい配備表現と元の配備表現を比較し、結果として得られる配備アクションを識別する（たとえば、ＫＤＰから鍵／鍵集合を除去するかまたはＫＤＰに鍵／鍵集合を追加する）。

１．所望の配備ソースと現在の配備ソースの和集合（union）を作成する。

２．所望の配備宛先と現在の配備宛先の和集合を作成する。

３．それぞれの新しい項目に０を充填し、和集合に応じて所望の配備行列と現在の配備行列を拡張する。

４．所望の拡張行列から現在の拡張配備行列を減算することによりデルタ行列を求める。

結果として得られるデルタ行列は「１，０，－１」という値の行列であり、「それを正しいものにする（make it right）」ために必要な「追加、変更なし、削除」動作を示す。ｉ番目の行とｊ番目の列においてデルタ行列内に１がある場合、システムはｊ番目のＫＤＰ（またはＫＤＰ集合）に「ｉ番目の鍵／鍵集合を追加する（add i-th key/keyset）」コマンドを発行し、ｓ番目の行とｋ番目の列に－１があるたびに、システムはｋ番目のＫＤＰ（またはＫＤＰ集合）に「ｓ番目の鍵／鍵集合を削除する（delete s-th key/keyset）」コマンドを発行する。

配備が新たに作成された場合、元の配備ソース、配備宛先、および配備行列が空の場合でも上記の手順が適用される。

【００４４】

上記のアルゴリズムを例示するために、模範的な現在の配備は、｛鍵１，鍵２，鍵３｝として定義される配備ソースと、｛ＫＤＰ１，ＫＤＰ２，ＫＤＰ３｝として定義される配備宛先と、以下のように定義される配備行列とを含む。

【数４】

$$D_{cur} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

何らかの理由で、管理者は、ＫＤＰ４と呼ばれる新しいＫＤＰをその配備に追加しようと決定することができ、その場合、所望の配備宛先集合は｛ＫＤＰ１，ＫＤＰ２，ＫＤＰ３，ＫＤＰ４｝になり、管理者は以下のように配備行列を調整する。

【数５】

$$D_{des} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

アルゴリズムは以下のように進行する。

ａ）｛鍵１，鍵２，鍵３｝である、現在の配備ソースと所望の配備ソースの和集合を計算する。

ｂ）｛ＫＤＰ１，ＫＤＰ２，ＫＤＰ３，ＫＤＰ４｝である、現在の配備宛先と所望の配備宛先の和集合を計算する。

ｃ）２つの和集合の集合に基づいて、拡張された現在の配備行列と所望の配備行列をそれぞれ以下のように求める。

【数６】

$$D'_{cur} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix},$$

および

【数７】

$$D'_{des} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

ｄ）最終的に上記の２つの行列のデルタを計算すると、以下のようになる。

【数８】

$$Delta = D'_{des} - D_{cur}' = \begin{pmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

したがって、アルゴリズムは、ＫＤＰ２から鍵１を除去する（removekey1 from KDP2）コマンドと、ＫＤＰ４に鍵１を追加する（add key1 to KDP4）コマンドと、ＫＤＰ３から鍵２を除去する（remove key2 from KDP3）コマンドと、ＫＤＰ４に鍵２を追加する（addkey2 to KDP4）コマンドと、ＫＤＰ１から鍵３を除去する（remove key3 from KDP1）コマ

ンドと、ＫＤＰ４に鍵３を追加する（add key3 to KDP4）コマンドを発行することになる。

【００４５】

暗号鍵および証明書の配備および配付の自動妥当性検査および実行のための本発明のシステムおよび方法をさらに例示するために、いくつかのユース・ケース・シナリオについて以下に説明する。具体的には、ユース・ケース・シナリオは、高レベル鍵ライフサイクル・マネージャ（ＫＬＭ）内の様々な鍵配備パターンを記述するものである。１つのユース・ケースについて、それに応じて１つまたは複数の配備パターンが存在する可能性があることに留意されたい。以下に示す模範的なシナリオでは、「Ｋ」は公開鍵または秘密鍵あるいはその両方を示し、「Ｃ」は自己署名されたかまたは秘密鍵の所有者のＩＤを検証した後で公開鍵に関する証明書を配信するエンティティである認証局（ＣＡ：certification authority）によって署名された公開鍵証明書である。これらのシナリオのうちの１つまたは複数では、１つのアプリケーションが複数の鍵配備を必要とする可能性があり、そのそれぞれが異なるＫＤＰ集合を有する。ＫＬＭからのビジネス・パートナの証明書のインポートと、ＫＬＭへの自分自身の証明書のエクスポートは、記載されている配備の範囲外であることに留意されたい。ＫＬＭ内の配備のみが収集される。また、「ｓｋ」および「ｋ／Ｋ」は鍵グループを表す。鍵グループは、少なくとも１つの特定の配備パターンを使用して配備される１つまたは複数の鍵を含む。さらに、「Ａ」、「Ｂ」、および「Ｃ」は、ＫＬＭの制御下にある鍵ストアなどの鍵配備ポイントを表す。

【００４６】

図４を参照すると、ＳＳＬ－ＴＬＳ　ＨＴＴＰＳ用のサーバ鍵および証明書の配備を伴うユース・ケース・シナリオが例示されている。これは、ＨＴＴＰＳをサポートするＷｅｂサーバ４００～４０４のクラスタを特徴とし、各サーバは、クライアントの視点から見て同じサーバであると見なされる。このシナリオでは少なくともサーバ側認証が必要である。このシナリオは、同じ秘密鍵Ｋ　４０６と公開鍵証明書Ｃ　４０８を使用し、これは構成しなければならないものである。また、満了時に鍵ベア４０６～４０８の自動リフレッシュが行われる。

【００４７】

図５および図６を参照すると、ＳＳＬ保護ＣＳＩｖ２用のサーバ鍵５００～５０４および証明書５０６～５１０を伴うユース・ケース・シナリオが例示されている。これは、ＣＳＩｖ２プロトコルを使用して通信するＪ２ＥＥサーバ５１２～５１６のグループを含む。各サーバ５１２～５１６は、固有の秘密鍵５００～５０４と、自己署名されたかまたはＣＡ署名された関連デジタル証明書５０６～５１０とを有する。図５は、秘密固有公開共用パターンを使用する鍵配備を例示しており、証明書は自己署名されている。図６は、秘密固有公開固有（private unique public unique）パターンを使用する鍵配備を例示しており、証明書はＣＡによって署名されている。

【００４８】

図７および図８を参照すると、ベンダ固有製品（vendor-specificproduct）のためにＷｅｂＳｅａｌとＷＡＳとの間の信頼関係を伴うユース・ケース・シナリオが例示されている。これは、ＷｅｂＳｅａｌ　ＷＡＳとＨＴＴＰサーバとの間でセットアップする必要があるＳＳＬ接続を伴う。結果として得られる鍵配備は図５および図６に例示されているものと同じである。すなわち、このユース・ケース・シナリオは、サーバ鍵６００～６０４と、証明書６０６～６１０と、サーバ６１２～６１６のグループとを伴う。各サーバ６１２～６１６は、固有の秘密鍵６００～６０４と、自己署名されたかまたはＣＡ署名された関連デジタル証明書６０６～６１０とを有する。図７は、秘密固有公開共用パターンを使用する鍵配備を例示しており、証明書は自己署名されている。図８は、秘密固有公開固有パターンを使用する鍵配備値を例示しており、証明書はＣＡによって署名されている。

【００４９】

図９および図１０を参照すると、ＷｅｂＳｅａｌとＷＡＳの両方に関するＬＴＰＡトークン生成および検証のために鍵がセットアップされるユース・ケース・シナリオが例示さ

れている。これは、クラスタとして動作し、ＬＴＰＡの使用によりサービスのリクエスタのＩＤを通信する複数のＷＡＳサーバ７００～７０４を含む。ＬＴＰＡトークンは、暗号化に対称暗号鍵ならびに公開鍵／秘密鍵の両方を使用し、ＬＴＰＡトークン・フォーマットを署名／検証して、暗号法で生成され保護される。図９は、共通鍵７０６がサーバ７００～７０４間で共用配備される、共通共用パターンを使用する鍵配備を例示している。図１０は、各サーバ７００～７０４が関連秘密鍵７０８～７１２を有するとともに各サーバ７００～７０４がそれぞれ公開鍵７１４～７１８のうちの１つを有する、秘密固有公開共用パターンを使用する鍵配備を例示している。

【００５０】

図１１および図１２を参照すると、ＳＣＳＩ機能ベースのコマンド・セキュリティ（ＣｂＣｓ）を伴うユース・ケース・シナリオが例示されている。ＣｂＣｓは、アプリケーション・クライアント、クライアント記憶域、およびセキュリティ・マネージャという３つのアクティブ・エンティティを伴う。記憶装置への要求には、暗号法で保護された機能、すなわち、［ＣＡＰ；Ｃｋｅｙ］というペアが付随しなければならない。Ｃｋｅｙ（Ｃａｐａｂｉｌｉｔｙ　Ｋｅｙ）は、セキュリティ・マネージャと記憶装置によって共用される対称共通鍵を使用する機能に関する暗号ハッシュ・トークンである。ＫＬＭは、それぞれのリフレッシュと、適切な記憶装置（複数も可）およびセキュリティ・マネージャへの配備を自動化することにより、作業鍵（working key）のメンテナンスを単純にする。図１１は、異なる共通鍵８００～８０４のそれぞれが対応する記憶装置８０６～８１０に配備される、共通固有パターンを使用する鍵配備を例示している。図１２は、同様の共通鍵を含む共通鍵８１２がセキュリティ・マネージャおよびそのバックアップ（複数も可）８１４～８１８に配備される、共通共用パターンを使用する鍵配備を例示している。これと同等であるのは、セキュリティ・マネージャが、すべてがすべての鍵を有する必要があるサーバのクラスタからなる場合である。

【００５１】

図１３を参照すると、たとえば、ＩＢＭのＴＳ１１２０ドライブとともに使用するためのＥＫＭサーバ９１２～９１６用の秘密鍵９００～９０４および公開鍵９０６～９１０（証明書なし）が配備されるユース・ケース・シナリオが例示されている。これは、テープ・ドライブのバンクをサポートするためまたはＥＫＭのインスタンス間のフェイルオーバを提供するために複数のＥＫＭサーバを伴い、ＲＳＡ秘密鍵および公開鍵を使用する。この鍵配備は、秘密鍵固有公開鍵固有（private key unique public key unique）パターンを使用する内部動作を含む。

【００５２】

図１４および図１５を参照すると、図１３のユース・ケース・シナリオの続きが例示されている。図１４では、公開鍵（おそらく）共用パターンを使用する誰かのビジネス・パートナにデータがエクスポートされる。この図の右側にある「Ｃ」はＫＬＭが入手したビジネス・パートナのＣＡ署名付き証明書１０００であり、Ａ、Ｂ、およびＣはＥＫＭで使用される鍵ストア１００２～１００６である。図１５では、ビジネス・パートナは、ＣＡ署名付き証明書を受信し、それによりその誰かに送られるデータを暗号化し、「ｋ」１００８はＡ、Ｂ、およびＣ　１０１０～１０１４でデータを暗号化解除するために使用される秘密鍵である。

【００５３】

図１６および図１７を参照すると、たとえば、ＬＴＯ－４ドライブとともに使用するためのＥＫＭサーバ用の対称暗号鍵を伴うユース・ケース・シナリオが例示されている。このシナリオは、ＡＥＳ鍵または３ＤＥＳ鍵を使用する。図１６は、共通鍵１１００～１１０４が関連の鍵ストアＡ、Ｂ、およびＣ　１１０６～１１１０にそれぞれ配備される、共通固有パターンを使用する内部動作を含む鍵配備を例示している。図１７は、共通（おそらく）共用パターンを使用するデータのエクスポートおよびインポートを含む鍵配備を例示している。ビジネス・パートナは誰かの公開鍵を使用して、データ暗号化のためにそのパートナの対称鍵をラップ（wrap）し、その後、その対称鍵はその誰かに送られる。その

誰かは、自分の秘密鍵を使用して、ビジネス対称鍵ｓｋ　１１１２をアンラップ（unwrap）することができ、その対称鍵はＡ、Ｂ、およびＣ　１１１４～１１１８に配備される。

【００５４】

他に可能なユース・ケース・シナリオ（図示せず）は、ＣＣＡ　ＰＩＮ鍵を伴い、その鍵はハードウェアで保護されるので実際の鍵生成および保管はＫＬＭでローカルに管理される。ＫＬＭは、鍵の生成およびライフサイクル管理のためのポリシーを定義するための管理インターフェースを提供する。

【００５５】

さらに他に可能なユース・ケース・シナリオ（図示せず）は、効果的な保護および回復のための集中鍵管理を伴う。このシナリオの特徴としては、複数のアプリケーションにわたるとともにインフラストラクチャ・スタックの複数の層にわたって集中方式で暗号鍵を管理することを含む。

【００５６】

また、ユース・ケース・シナリオ（図示せず）は、規制準拠をサポートするための暗号化鍵の管理を伴う場合もある。このシナリオの特徴としては、データが安全に暗号化され、すべての関連鍵が中央で正しく管理されることを証明するためにＫＬＭが監査報告を配信することを含む。

【００５７】

さらに他に可能なユース・ケース・シナリオ（図示せず）は、自動証明書リフレッシュによりＷｅｂサイトの可用性を維持することを伴う。このシナリオの特徴としては、証明書の満了によるＷｅｂサーバの停止時間を回避するために証明書の満了付近で自動リフレッシュすることを含む。

【００５８】

一般に、暗号鍵および証明書の配備および配付の自動妥当性検査および実行を実装するための方法実施形態は、汎用コンピュータによって実施することができ、その方法は、汎用コンピュータによる使用のための取り外し可能媒体またはハード媒体上の１組の命令としてコード化することができる。図１８は、本発明の諸実施形態を実施するために適した汎用コンピュータの概略ブロック図である。図１８では、コンピュータ・システム１２００は、少なくとも１つのマイクロプロセッサまたは中央演算処理装置（ＣＰＵ）１２０５を有する。ＣＰＵ１２０５は、システム・バス１２１０を介して、ランダム・アクセス・メモリ（ＲＡＭ）１２１５、読み取り専用メモリ（ＲＯＭ）１２２０、取り外し可能データ／プログラム記憶装置１２３０および大容量データ／プログラム記憶装置１２３５を接続するための入出力（Ｉ／Ｏ）アダプタ１２２５、キーボード１２４５およびマウス１２５０を接続するためのユーザ・インターフェース・アダプタ１２４０、データ・ポート１２６０を接続するためのポート・アダプタ１２５５、ならびにディスプレイ装置１２７０を接続するためのディスプレイ・アダプタ１２６５に相互接続されている。

【００５９】

ＲＯＭ１２２０は、コンピュータ・システム１２００用の基本オペレーティング・システムを収容している。オペレーティング・システムは、代わって、ＲＡＭ１２１５または当技術分野で知られている他の場所に常駐することもできる。取り外し可能データ／プログラム記憶装置１２３０の例としては、フレキシブル・ディスク・ドライブおよびテープ・ドライブなどの磁気媒体と、ＣＤ－ＲＯＭドライブなどの光メディアを含む。大容量データ／プログラム記憶装置１２３５の例としては、ハード・ディスク・ドライブと、フラッシュ・メモリなどの不揮発性メモリを含む。キーボード１２４５およびマウス１２５０に加えて、トラックボール、ライティング・タブレット、圧力パッド、マイクロホン、ライト・ペン、および位置感知スクリーン・ディスプレイなどのその他のユーザ入力装置もユーザ・インターフェース１２４０に接続することができる。ディスプレイ装置の例としては、陰極線管（ＣＲＴ）および液晶ディスプレイ（ＬＣＤ）を含む。

【００６０】

本発明の実施を単純にするために、適切なアプリケーション・インターフェースを有す

るコンピュータ・プログラムが当業者によって作成され、システム上またはデータ／プログラム記憶装置上に保管される場合がある。動作時に、本発明を実行するために作成されたコンピュータ・プログラムに関する情報は、適切な取り外し可能データ／プログラム記憶装置１２３０上にロードされるか、データ・ポート１２６０を介して供給されるか、またはキーボード１２４５を使用してタイプ入力される。

【００６１】

したがって、上記を考慮して、本発明の方法実施形態は、コンピュータまたはコントローラで実装されるプロセスおよびこれらのプロセスを実施するための装置の形を取ることができる。また、本発明は、フレキシブル・ディスク、ＣＤ－ＲＯＭ、ハード・ディスク・ドライブ、または任意のその他のコンピュータ可読記憶媒体などの有形媒体に実施される命令を含むコンピュータ・プログラム・コードの形で実施することもでき、コンピュータ・プログラム・コードがコンピュータまたはコントローラにロードされ、コンピュータまたはコントローラによって実行されたときに、そのコンピュータは本発明を実施するための装置になる。また、本発明は、たとえば、記憶媒体に保管されるか、コンピュータまたはコントローラにロードされるかあるいはコンピュータまたはコントローラによって実行されるかもしくはその両方であるか、電線またはケーブルによるか、光ファイバによるか、電磁放射によるなどの何らかの伝送媒体により伝送されるかにかかわらず、コンピュータ・プログラム・コードまたは信号の形で実施することもでき、コンピュータ・プログラム・コードがコンピュータにロードされ、コンピュータによって実行されたときに、そのコンピュータは本発明を実施するための装置になる。汎用マイクロプロセッサ上で実装される場合、コンピュータ・プログラム・コード・セグメントは、特定の論理回路を作成するようにマイクロプロセッサを構成する。実行命令の技術的効果は、上記の模範的な方法を実装することである。

【００６２】

１つまたは複数の好ましい実施形態に関して本発明を説明してきたが、当業者であれば、本発明の範囲を逸脱せずに、様々な変更が可能であり、その諸要素の代わりに同等のものを使用できることが理解されるであろう。加えて、本発明の本質的な範囲を逸脱せずに、特定の状況または材料を本発明の教示に適合させるために多くの修正を行うこともできる。したがって、本発明は、本発明を実行するために企図された最良の態様として開示されている特定の実施形態に限定されず、特許請求の範囲内に含まれるすべての実施形態を含むことになることが意図されている。

【図面の簡単な説明】

【００６３】

【図１】複数の鍵配備ポイントに配備される秘密鍵＋証明書を有する配備パターンを例示する図である。

【図２】それぞれが１対の鍵配備ポイントに配備された２通りの鍵集合を有する配備パターンを例示する図である。

【図３】秘密固有公開共用配備パターンを例示する図である。

【図４】ＳＳＬ－ＴＬＳ　ＨＴＴＰＳ用のサーバ鍵および証明書の配備に使用できる配備パターンを例示する図である。

【図５】ＳＳＬ保護ＣＳＩｖ２用のサーバ鍵および証明書の配備に使用できる配備パターンを例示する図である。

【図６】ＳＳＬ保護ＣＳＩｖ２用のサーバ鍵および証明書の配備に使用できる配備パターンを例示する図である。

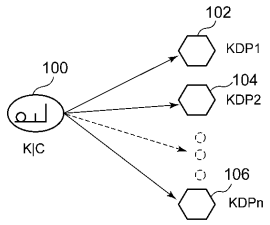
【図７】ＷｅｂＳｅａｌとＷＡＳとの間の信頼関係を提供できる配備パターンを例示する図である。

【図８】ＷｅｂＳｅａｌとＷＡＳとの間の信頼関係を提供できる配備パターンを例示する図である。

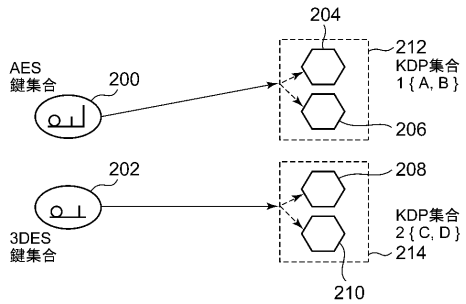
【図９】ＷｅｂＳｅａｌとＷＡＳの両方に関するＬＴＰＡトークン生成および検証のために鍵がセットアップされるシナリオで使用できる配備パターンを例示する図である。

【図１０】ＷｅｂＳｅａｌとＷＡＳの両方に関するＬＴＰＡトークン生成および検証のために鍵がセットアップされるシナリオで使用できる配備パターンを例示する図である。
【図１１】ＳＣＳＩ機能ベースのコマンド・セキュリティ（ＣｂＣｓ）で使用できる配備パターンを例示する図である。
【図１２】ＳＣＳＩ機能ベースのコマンド・セキュリティ（ＣｂＣｓ）で使用できる配備パターンを例示する図である。
【図１３】たとえば、ＩＢＭのＴＳ１１２０ドライブとともに使用するためのＥＫＭサーバ用の公開鍵および秘密鍵の配備に使用できる配備パターンを例示する図である。
【図１４】図１３の続きを例示する図である。
【図１５】図１３の続きを例示する図である。
【図１６】たとえば、ＬＴＯ－４ドライブとともに使用するためのＥＫＭサーバ用の対称暗号鍵の配備に使用できる配備パターンを例示する図である。
【図１７】たとえば、ＬＴＯ－４ドライブとともに使用するためのＥＫＭサーバ用の対称暗号鍵の配備に使用できる配備パターンを例示する図である。
【図１８】本発明の諸実施形態を実施するために適した汎用コンピュータの概略ブロック図である。
【符号の説明】
【００６４】
１００　秘密鍵＋証明書（Ｋ｜Ｃ）
１０２　Ｗｅｂサーバ（ＫＤＰ）
１０４　Ｗｅｂサーバ（ＫＤＰ）
１０６　Ｗｅｂサーバ（ＫＤＰ）

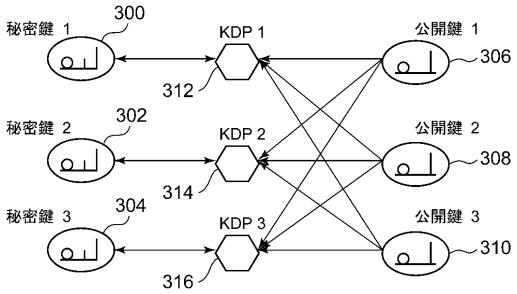
【図１】



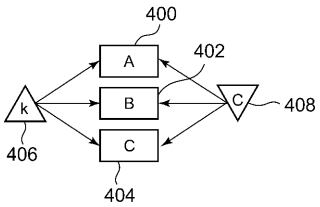
【図２】



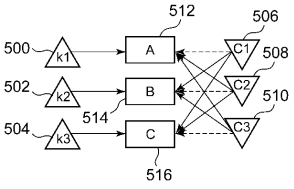
【図３】



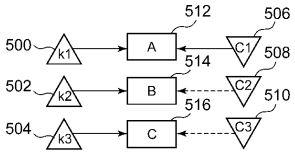
【図４】



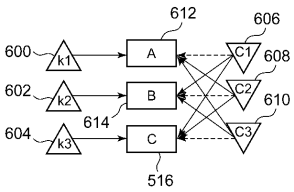
【図５】



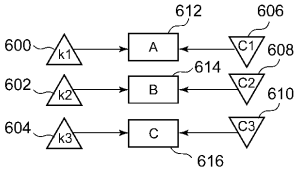
【図６】



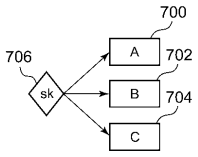
【図７】



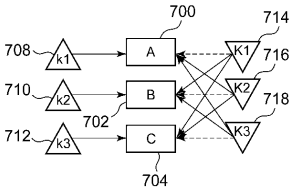
【図８】



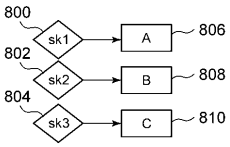
【図９】



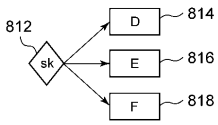
【図１０】



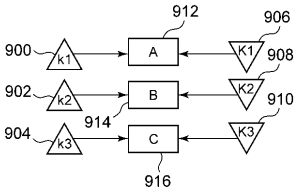
【図１１】



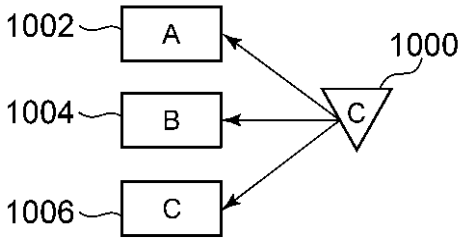
【図１２】



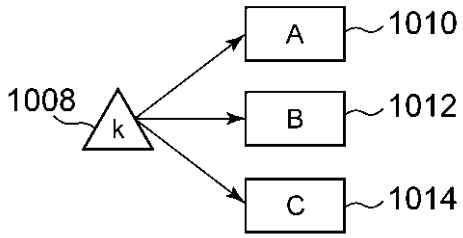
【図１３】



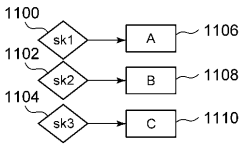
【図１４】



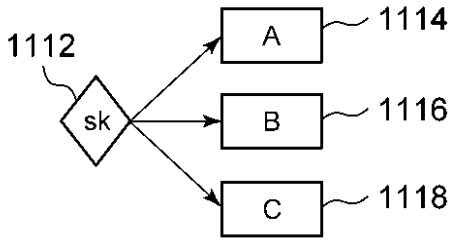
【図１５】



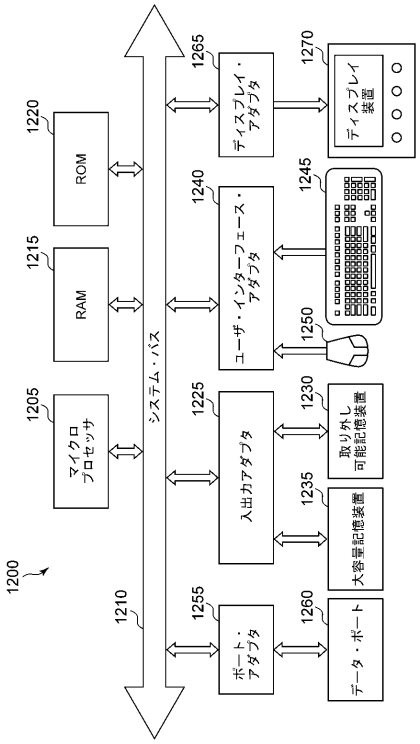
【図１６】



【図１７】


1112
sk
A
B
C
1114
1116
1118

【図１８】


1200
1210
システム・バス
1205
マイクロプロセッサ
1215
RAM
1220
ROM
1225
入出力アダプタ
1230
取り外し可能記憶装置
1235
大容量記憶装置
1240
ユーザ・インターフェース・アダプタ
1245
1250
1255
ポート・アダプタ
1260
データ・ポート
1265
ディスプレイ・アダプタ
1270
ディスプレイ装置

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 シャオユウ・フウ
スイス国シーエイチ－８８１０ ホーゲン シュライフェトベルベク ７
(72)発明者 ロバート・ハース
スイス国シーエイチ－８１３４ アドリスビル スードシュトラッセ １３
(72)発明者 イリアス・イリアディス
スイス国シーエイチ－８８０３ ルーシュリコン シュロス－シュトラッセ ２９
(72)発明者 クリスチャン・カシャン
スイス国シーエイチ－８８００ タルヴィル イム・マールバッハ １７
(72)発明者 レネ・アー・パブリツェク
スイス国シーエイチ－８８０２ キルヒベルク シュロスベルクシュトラッセ １２
(72)発明者 ティモシー・ジェイムス・ハーン
アメリカ合衆国２７５１９ ノースカロライナ州ケアリー リトルフォード・レーン ２０７
(72)発明者 ジョン・ティー・ペック
アメリカ合衆国７８６４２－４５４２ テキサス州リバティー・ヒル チャパラル・ドライヴ ３００

Ｆターム(参考) 5J104 AA16 EA04 EA15 EA16 JA03 JA21 MA05 MA07 NA02 NA27
NA37 NA38