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

特許第6665113号
(P6665113)

(45)発行日　令和2年3月13日(2020.3.13)　(24)登録日　令和2年2月21日(2020.2.21)

(51)Int.Cl.			F I		
H04L	9/14	(2006.01)	H04L	9/00	641
H04L	9/08	(2006.01)	H04L	9/00	601A
G09C	1/00	(2006.01)	G09C	1/00	640D
H04L	9/32	(2006.01)	H04L	9/00	675B

請求項の数 11　(全 23 頁)

(21)出願番号　特願2016-566806 (P2016-566806)
(86)(22)出願日　平成27年5月4日(2015.5.4)
(65)公表番号　特表2017-515413 (P2017-515413A)
(43)公表日　平成29年6月8日(2017.6.8)
(86)国際出願番号　PCT/US2015/028991
(87)国際公開番号　WO2015/171476
(87)国際公開日　平成27年11月12日(2015.11.12)
審査請求日　平成30年4月23日(2018.4.23)
(31)優先権主張番号　61/988,786
(32)優先日　平成26年5月5日(2014.5.5)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　14/481,399
(32)優先日　平成26年9月9日(2014.9.9)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　314015767
マイクロソフト　テクノロジー　ライセンシング，エルエルシー
アメリカ合衆国　ワシントン州　９８０５２　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100118902
弁理士　山本　修
(74)代理人　100106208
弁理士　宮前　徹
(74)代理人　100120112
弁理士　中西　基晴

最終頁に続く

(54)【発明の名称】継続的な所有者アクセスを伴う、暗号化された仮想マシンの安全なトランスポート

(57)【特許請求の範囲】
【請求項１】
トランスポート鍵を使用した第１の暗号化メカニズムを用いて暗号化されたデータセットを管理するためのコンピューターシステムであって、
前記トランスポート鍵の複数のコピーを、
前記データセットの所有者のために前記トランスポート鍵の第１のコピーを第２の暗号化メカニズムを用いて暗号化して、前記データセットの前記所有者は前記トランスポート鍵の前記第１のコピーを前記第２の暗号化メカニズムを使用して復号化できるが、前記所有者以外のエンティティは前記トランスポート鍵の前記第１のコピーを復号化できないようにすることと、
前記所有者以外の少なくとも１つのエンティティのために前記トランスポート鍵の第２のコピーを第３の暗号化メカニズムを用いて暗号化して、前記少なくとも１つのエンティティは前記トランスポート鍵の前記第２のコピーを前記第３の暗号化メカニズムを使用して復号化できるが、前記少なくとも１つのエンティティ以外のエンティティは前記トランスポート鍵の前記第２のコピーを復号化できないようにすることと
を実行することによって暗号化する手段と、
前記トランスポート鍵の暗号化された前記第１のコピー及び前記第２のコピーを含むパッケージを作成する手段と、
前記所有者又は前記所有者以外の前記少なくとも１つのエンティティの何れかに対応するガーディアン署名を生成するステップであって、前記ガーディアン署名は、前記パッケ

ージを暗号的にまとめて、前記データセットが無許可エンティティにさらされないことを保証する、手段と、
前記トランスポート鍵から作成されたトランスポート鍵署名を生成するステップであって、前記トランスポート鍵署名は、前記パッケージを暗号的に更にまとめ、前記データセットへのアクセスを試みる何れのエンティティも前記トランスポート鍵の知識を有していることの証明として機能する、手段と
を含むシステム。
【請求項２】
請求項１に記載のシステムであって、前記データセットは仮想マシンの部分を含む、システム。
【請求項３】
請求項２に記載のシステムであって、前記少なくとも１つのエンティティは前記仮想マシンのホスタであり、前記データセットの前記所有者は前記ホスタのテナントである、システム。
【請求項４】
請求項３に記載のシステムであって、前記トランスポート鍵は、前記仮想マシンン又は前記仮想マシンについての仮想トラステッド・プラットフォーム・モジュール（ｖＴＰＭ）に対するマスター鍵である、システム。
【請求項５】
請求項４に記載のシステムであって、前記データセットは前記仮想マシンについてのプロビジョニング情報を含む、システム。
【請求項６】
請求項５に記載のシステムであって、前記データセットはｖＴＰＭ状態を含む、システム。
【請求項７】
請求項３に記載のシステムであって、移行エージェントが、暗号化された前記データセット及び前記パッケージを前記ホスタに転送する、システム。
【請求項８】
請求項７に記載のシステムであって、暗号化された前記データセットへのアクセスを許可する前に、前記ホスタが、要求をトラステッド鍵配布サービス（ＫＤＳ）に送り、前記要求に関してアクションを起こす前に、前記ＫＤＳが、前記ホスタがある資格を満たすかを判定する、システム。
【請求項９】
請求項８に記載のシステムであって、前記要求に加えて、前記ホスタが、前記パッケージ又は前記トランスポート鍵の暗号化された前記第２のコピーの何れかも送る、システム。
【請求項１０】
請求項８に記載のシステムであって、前記ＫＤＳが、前記トランスポート鍵の暗号化された前記第２のコピーを、別のサービスから、前記要求に関してアクションを起こす前に受ける、システム。
【請求項１１】
請求項１に記載のシステムであって、前記第２の暗号化メカニズムは、前記所有者の公開鍵を使用したものであり、前記第３の暗号化メカニズムは、前記所有者以外の前記少なくとも１つのエンティティの公開鍵を使用したものである、システム。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001]コンピューティングシステムの相互接続は、いわゆる「クラウド」コンピューティングシステムなど、分散型コンピューティングシステムを推進してきた。この説明において、「クラウドコンピューティング」とは、管理労力またはサービスプロバイダーとの

相互作用を軽減した状態で準備および解放し得る、設定変更可能なコンピューティングリソースの共有プール（例：ネットワーク、サーバー、ストレージ、アプリケーション、サービスなど）への、普遍的で便利なオンデマンドネットワークアクセスを可能にするシステムまたはリソースを指すものと考えられる。クラウドモデルは様々な特徴（例：オンデマンドセルフサービス、広域ネットワークアクセス、リソースプーリング、高速伸縮性、従量制サービスなど）、サービスモデル（例：サービスとしてのソフトウェア（「ＳａａＳ」）、サービスとしてのプラットフォーム（「ＰａａＳ」）、サービスとしてのインフラストラクチャ（「ＩａａＳ」））、および展開モデル（例：プライベートクラウド、コミュニティクラウド、パブリッククラウド、ハイブリッドクラウドなど）で構成され得る。

【０００２】

[0002]クラウドおよびリモートベースのサービスアプリケーションが普及している。そのようなアプリケーションはクラウドなどパブリックおよびプライベートの遠隔システム上でホストされ、また通常、クライアントと連絡を取り合うための一連のウェブベースサービスを提供する。

【０００３】

[0003]クラウドコンピューティング環境において、テナントは、クラウドサービスプロバイダーによって運営されるデータセンターで展開される仮想マシンのうちの１つまたは複数にアクセスする権利を有するユーザー、会社、会社の部門、または他のエンティティを含む場合がある。テナントは、テナントのために仮想マシンが展開されるのを望むことがある。しかし、テナントは、その仮想マシンがデータセンターにおけるホスト上で展開される前の保管中に、または、仮想マシンがデータセンター中のホストに展開されつつあるときに、これらのマシンに他のエンティティ（他のテナント、傍受者、さらにはデータセンター管理者など）がアクセスできないようにしたいと思うことがある。これを達成するために、仮想マシンの仮想ハードドライブおよび／またはメタデータを暗号化することなどによって、仮想マシンを暗号化することができる。したがって、暗号化された仮想マシンは、その内容の機密性および整合性を保存するようにして、オフラインストレージの内外へ、かつホスト間で、移動することになる。さらに、仮想マシンの内容を暗号化するのに使用される鍵は、何らかの定期性でロールオーバー（すなわち変更）されなければならないであろう。このことは、仮想マシンの所有者がその暗号化された仮想マシンにアクセスできることに対して、困難を提示する場合がある。

【発明の概要】

【発明が解決しようとする課題】

【０００４】

[0004]しかし、仮想マシンを、ホストにおいて展開可能であるようにするとともに、ＶＭの所有者（例えばテナント）による取出しおよび実行のために利用可能であるようにすることが、いくらか望まれているであろう。これは、様々な暗号化方式の保護性質を考えると難しい可能性がある。特に、暗号化の一目標は、復号に使用される秘密を保護することによって、暗号化されたＶＭを多数のエンティティが復号できないようにすることである。したがって、暗号化されたＶＭを２つの異なるエンティティが復号できるようにすることに伴う困難がある。

【０００５】

[0005]本明細書における特許請求対象事項は、何らかの不利益に対処する実施形態、または前述のような環境のみで動作する実施形態に限定されない。むしろ、この背景は、本明細書に記載のいくつかの実施形態が実践され得る１つの模範的技術分野を例示するために提供されるに過ぎない。

【課題を解決するための手段】

【０００６】

[0006]本明細書で例示される一実施形態は、コンピューティング環境で実践され得る方法を含む。この方法は、暗号化されたデータセットを管理するための行為を含む。この方

法は、第１の復号鍵を取得することを含む。第１の復号鍵は、第１の暗号化メカニズムを使用して暗号化された暗号化済みデータセットを復号するのに使用されるように構成される。第１の暗号化メカニズムは、データセットを復号するのに使用され得る第１の復号鍵に関連する。この方法はさらに、第１の復号鍵を第２の暗号化メカニズムで暗号化することを含む。第２の暗号化メカニズムは、第１のエンティティによって使用される第２の復号鍵に関連し、したがって、第２の復号鍵は、最初に第２の復号鍵を使用して第１の鍵暗号化済み鍵を復号してからこの復号された第１の鍵を使用してデータセットを復号することによってデータセットを復号するために、第１のエンティティによって使用され得る。この方法はさらに、第１の復号鍵を第３の暗号化メカニズムで暗号化することを含む。第３の暗号化メカニズムは、第２のエンティティによって使用される第３の復号鍵に関連し、したがって、第３の復号鍵は、最初に第３の復号鍵を使用して第１の鍵暗号化済み鍵を復号してからこの復号された第１の鍵を使用してデータセットを復号することによってデータセットを復号するために、第２のエンティティによって使用され得る。この方法はさらに、第２の暗号化方法で暗号化された第１の復号鍵と、第３の暗号化方法で暗号化された第１の復号鍵とを少なくとも含むパッケージを作成することを含む。この方法はさらに、ガーディアン（ｇｕａｒｄｉａｎ）署名でパッケージに署名すること、および、第１の復号鍵から作成された署名でパッケージに署名することを含む。

【０００７】

[0007]この「発明の概要」は、下記の「発明を実施するための形態」で詳述される一連の概念を簡略化された形で紹介する目的で提示される。この「発明の概要」は、特許請求対象事項の主要な特徴または本質的特徴を特定することを意図するものではなく、また特許請求対象事項の範囲を判断する際の支援手段として使用されることも意図していない。

【０００８】

[0008]付加的な特徴および優位性は後述の説明に記載され、また部分的に説明から明白となるか、または本明細書に記載の教示の実践によって学習され得る。本発明の特徴および優位性は、添付の請求項において特に指摘される機器および組み合わせを手段として実現および獲得され得る。本発明の特徴は、後述の説明および添付の請求項から、より全面的に明らかとなるか、または後述する通り、本発明の実践によって学習され得る。

【０００９】

[0009]上記および他の優位性や特徴を獲得可能な方法を記述するため、上記にて簡潔に記載の対象事項に関する、より詳しい記述が、添付の図面において例示される特定の実施形態を参照することにより表現される。これらの図面は典型的な実施形態だけを描写するものであるため、範囲を制限するものと解釈してはならないことを理解しつつ、実施形態は付加的な特異性および詳細と併せて、下記の添付図面の使用を通じて記述および説明される。

【図面の簡単な説明】

【００１０】

【図１】[0010]暗号化されたデータセットと、暗号化されたデータセットを復号するための暗号化された鍵を含むパッケージと、を示す図である。

【図２】[0011]暗号化された仮想マシンを展開するのを示す図である。

【図３】[0012]ホストおよびターゲットについてのアテステーション（ａｔｔｅｓｔａｔｉｏｎ；認証）動作を示す図である。

【図４】[0013]仮想マシンをホスト上で起動するためのフローを示す図である。

【図５Ａ】[0014]仮想マシンを移行するためのフローを示す図である。

【図５Ｂ】[0014]仮想マシンを移行するためのフローを示す図である。

【図６】[0015]暗号化されたデータセットを管理する方法を示す図である。

【発明を実施するための形態】

【００１１】

[0016]本発明の一実施形態の単純な例示が、図１に関して示される。図１は、暗号化されたデータセット１０１を示す。暗号化されたデータセット１０１は、トランスポート鍵

１０２を使用して復号されることが可能である。示される例では、トランスポート鍵１０２の複数のコピーが、特定のエンティティに特有の暗号化方式を使用して暗号化される。したがって、例えば、第１のコピー１０２－０は所有者に対して暗号化され、それにより、データセット１０１の所有者はコピーを復号することができるが他のエンティティは復号することができない。図１はさらに、ｎ個の追加コピーを示すが、ｎ個の追加コピー１０２－１～１０２－ｎのそれぞれは、カストディアン（ｃｕｓｔｏｄｉａｎ）（所有者のためにデータセットをホストするかまたは記憶することができるエンティティなど）に対して暗号化され、それにより、各カストディアンはそのコピーを復号することができるが他のエンティティは復号することができない。いくつかの実施形態では、これは、非対称暗号化技法を使用して達成され得る。したがって、例えば、第１のコピー１０２－０は、所有者に対する公開鍵を使用して暗号化され、それにより、所有者は、その秘密鍵を使用して第１のコピー１０２－０を復号することができる。同様に、コピー１０２－１は、第１のカストディアンに対する公開鍵を使用して暗号化されてよく、それにより、第１のカストディアンは、カストディアンの秘密鍵を使用してコピー１０２－１を復号することができる。残りのカストディアンコピーについても、同様の暗号化および復号が行われてよい。

【００１２】

[0017]コピーは、共にパッケージされ、暗号鍵付きハッシュまたはディジタル署名を使用するガーディアン署名１０３で署名されて、パッケージが改ざんされていないことが保証される。コピーに署名するエンティティであるガーディアンは、例えば、所有者またはカストディアンであってよく、暗号化されたデータセット１０１が転送および／または再暗号化されるのに伴って変化する可能性がある。ガーディアン署名は、パッケージを暗号的にまとめて、データセット１０１が無許可エンティティにさらされる可能性がないことを保証することができる。パッケージはまた、トランスポート鍵署名１０４も含む。トランスポート鍵署名１０４は、ガーディアンがトランスポート鍵を知っていることの証明として使用されることが可能な、トランスポート鍵の暗号ハッシュまたは他の関数（メッセージ認証コード（ＭＡＣ）関数など）を実施することによって作成される認証の一形式である。

【００１３】

[0018]次に、より詳細な例を示すが、本明細書で示されるいくつかの実施形態は、暗号化済み仮想マシン（ＶＭ）をデータセンター中のホストに安全に展開することを対象とし、これは、ホストが暗号化済みＶＭを復号することと、ＶＭのテナント所有者が暗号化済みＶＭを復号できることと、の両方を可能にするようにして行われる。これは、システムがＶＭを暗号化すること、および、ＶＭに対するＶＭ復号鍵の２つの（または望まれるならより多くの）コピーが作成されること、によって達成され得る。ＶＭ復号鍵の一方のコピーは、テナント公開鍵を使用して暗号化され（これにより、テナントは、テナント秘密鍵を使用してＶＭ復号鍵を復号し、次いでＶＭ復号鍵を使用してＶＭを復号することができる）、ＶＭ復号鍵の第２のコピーは、鍵配布サービス（ＫＤＳ）公開鍵を使用して暗号化される（これにより、ホストは、ＫＤＳからの助けによってＶＭ復号鍵を取得することができる）。加えて、２つの暗号化された鍵は、前述のようにガーディアン署名および復号鍵署名を使用するなどして適切にラップされてよく、それにより、改ざんが行われていないことが保証される。

【００１４】

[0019]上の要約は、仮想マシンをデータセンター環境で展開するコンテキストにおけるものだが、実施形態は他のシナリオでも実施され得ることを理解されたい。

[0020]図２に仮想マシン例が示されており、この例は、テナント２００、クラウドサービス２０１、仮想マシンマネージャー２０２、仮想マシンストレージ２０３、仮想マシン２０４、移行エージェント２０５、ＫＤＳ２０６、およびホスト２０７を示す。示される例では、仮想マシン２０４は、仮想マシンストレージ２０３中の仮想マシンのクロスハッチングによって示されるように、暗号化されている。仮想マシンストレージ２０３は、ク

ラウドサービスプロバイダー２０１によって維持されるストレージシステムである。テナント２００は、ＶＭをクラウドサービスプロバイダー２０１に提供し、クラウドサービスプロバイダー２０１は、それらをマシンストレージ２０３に記憶することができる。テナント２００がＶＭを暗号化してもよく、または、ＶＭは他のエンティティによって暗号化されてもよい。

【００１５】

[0021]移行エージェント２０５（または別の適切なエンティティ、例えばテナント２００）は、暗号化された仮想マシン２０４を復号するのに必要な鍵の、２つのコピーを提供する。鍵は、テナント２００もしくはＫＤＳ２０６、または他の何らかの信頼されるエンティティによって、移行エージェントに提供されてよい。鍵２０８の一方のコピー２０９は、テナント２００の公開鍵を使用して暗号化され、鍵２０８の他方のコピー２１０は、ＫＤＳ２０６の公開鍵を使用して暗号化される。様々な代替形態が実施されてもよい。

【００１６】

[0022]鍵２０８を使用して、ＶＭ２０４自体を復号することができる。別法として、鍵２０８を使用して、仮想トラステッド・プラットフォーム・モジュール（ｖＴＰＭ）（後でより詳細に説明される）を復号することができ、このｖＴＰＭを使用してＶＭ２０４を復号することができる。しかし、これもなお、ＶＭが暗号化されていると記述されるものの範囲内に入る。いくつかの実施形態では、ＶＭ２０４は、ＶＭ２０４の、仮想ハードドライブ（ＶＨＤ）および／またはメタデータを暗号化することによって暗号化されてよい。

【００１７】

[0023]テナント２００が暗号化済みＶＭ２０４を取り出して読み取りたいと望む場合、テナントは単に、暗号化済みＶＭ２０４を仮想マシンストレージ２０３に再び要求し、その秘密鍵を使用して鍵２０８の第１のコピー２０９を復号し、次いで鍵２０８を使用してＶＭ２０４にアクセスすればよい。

【００１８】

[0024]ＶＭ２０４をホスト２０７に展開するには、暗号化済みＶＭは、ホスト２０７に送られる。加えて、鍵２０８の暗号化済みコピー２０９および２１０を含むパッケージ２１２が、ホスト２０７に送られる。

【００１９】

[0025]パッケージ２１２は、ガーディアン署名と、鍵２０８に対する署名との両方で署名されている。

[0026]ホスト２０７は、要求２１１およびパッケージ２１２を、ＫＤＳ２０６に送る。別法として、要求２１１をＫＤＳ２０６に送ることに加えて、ホスト２０７は、鍵２０８の第２のコピー２１０をＫＤＳに送ることもできる。別法として、ＫＤＳ２０６は、第２のコピー２１０を別のサービスから受け取り、要求２１１を見越して第２のコピー２１０を記憶しておくこともできる。

【００２０】

[0027]いくつかの実施形態では、ホスト２０７が何らかの資格を満たすことが決定されてよく、ホスト２０７が資格を満たす場合は、鍵２０８がＫＤＳ２０６によってアクセスされてホスト２０７に送り返されてよい。例えば、実施形態は、ホストが何らかのソフトウェア（またはソフトウェアの何らかのバージョン）を実行していること、何らかの構成を有すること、適切なブートレコードを有することが必要な場合があるなど、ホスト２０７が何らかの正常性要件を満たすことを必要とする場合がある。次いで、ホストは、この鍵２０８を使用して、ＶＭ２０４をアンロックすることができ、ＶＭ２０４がホスト２０７上で展開されるのを可能にすることができる（ＶＭ２０４のクロスハッチングが付いていないバージョンによって示されるように）。

【００２１】

[0028]次に、さらに追加の詳細を以下に示す。鍵配布サービスの使用を必要とするフローでは、「保護記述子（ｐｒｏｔｅｃｔｉｏｎ　ｄｅｓｃｒｉｐｔｏｒ）」またはＰＤと

呼ばれるデータ構造が使用される。ＰＤの主要な機能は、トランスポート鍵と呼ばれる暗号化鍵（例えば鍵２０８）の暗号的ラッピングである。このラッピングは、鍵へのアクセスが許可エンティティのみに与えられることを保証する。ＫＤＳ２０６は、トランスポート鍵がどんなデータを保護するかを知らないかまたは気にかけない。

【００２２】

[0029]例示として、仮想マシン（ＶＭ）の所有者が、ＶＭをホスタに展開したい場合がある。ＶＭは、２つのデータセット、すなわち、メタデータセクションと仮想ハードドライブ（ＶＨＤ）のセットとを含む。ＶＨＤは、ワシントン州レドモンドのＭｉｃｒｏｓｏｆｔ　Ｃｏｒｐｏｒａｔｉｏｎから入手可能なＢｉｔＬｏｃｋｅｒなど、適切な暗号化技術を使用して暗号化される。プライマリＶＨＤを復号するのに使用されるフルボリューム暗号化鍵（ＦＶＥＫ）は、仮想トラステッド・プラットフォーム・モジュール（ｖＴＰＭ）によって保護され、ｖＴＰＭの状態は、暗号化されて、ＰＤと共にメタデータの一部として記憶される。ｖＴＰＭ状態自体は、ＰＤによってラップされた鍵を使用して暗号化される。これにより、所有者は、ＶＭへの望まれないアクセスからＶＭを保護することができる。

【００２３】

[0030]ホスタにおけるホストがＶＭを起動する必要があるとき、ホストは、ｖＴＰＭに対するＰＤをメタデータから抽出し、これをＫＤＳに送る。ホストがｖＴＰＭ鍵に対して許可される場合、ＫＤＳは、ｖＴＰＭを復号するのに使用できる鍵を、ホストのセキュアサブシステムなど、トラステッド実行環境（ｔｒｕｓｔｅｄ　ｅｘｅｃｕｔｉｏｎ　ｅｎｖｉｒｏｎｍｅｎｔ）（ＴＥＥ）に返すことになる。様々な異なるセキュアサブシステムが、合同でまたは代替的に使用されてよい。一実施形態では、このサブシステムは、ホストＶＭのカーネル中で稼働している機能として実施されてよい。別の実施では、このサブシステムは、ハイパーバイザー中で稼働していてよい。他の実施形態では、このサブシステムは、プロセッサーのメモリーマッピング能力を使用してハイパーバイザーによって強制される別個のアドレス空間として実施されてよい（本明細書では仮想保護モード（ＶＳＭ）と称される場合がある）。他の実施形態では、このサブシステムは、プロセッサーによって強制される別個の実行エリアとして実施されてよい（カリフォルニア州サンタクララのＩｎｔｅｌ　Ｃｏｒｐｏｒａｔｉｏｎによって記述される新興のＳＧＸ能力である、ＡＲＭアーキテクチャのＴｒｕｓｔＺｏｎｅ、または、トラステッド・プラットフォーム・モジュール（ＴＰＭ）技術など）。これらの異なる実施は、暗号操作を行い、証明書を記憶し、コードまたはデータの整合性を検証し、秘密を保護する能力など、同様の機能を提供することができる。しかし、これらは、これらが提供するセキュリティプロパティが異なる場合がある。

【００２４】

[0031]別法として、いくつかの実施形態では、ＶＭがＶＭ所有者の環境でアクセスされる場合、ＫＤＳは鍵の解放に関与しない。というのは、ＰＤは、ＶＭの所有者による直接アクセスを可能にするトランスポート鍵のラッピングを含むからである。とりわけ、ＫＤＳは、いくつかの実施形態ではクラウドサービス２０１によって維持されてよいが、他の環境ではサードパーティによって維持されてよい。

【００２５】

[0032]鍵の受取りを許可されるエンティティは、「所有者」であるか、または０以上の「カストディアン」である。この２つの間の主要な区別は、いくつかの実施形態では、所有者は元のＰＤを作成できることである。また、いくつかの実施形態では、所有者のみが、自己署名された証明書を使用してＰＤ中で表明されることが可能である。ＰＤ設計は、詐欺師検出に注意を払う。すなわち、無許可エンティティがＰＤの所有者またはカストディアンを装うことができないことを保証するよう留意される。最後に、鍵が危険にさらされることからの回復のためのプロビジョンが行われる。例えば、これは、異なる寿命の鍵を使用して達成されてよく、この場合、所有者は、より強く保護される長期間の鍵によって表され、この鍵は、同程度の保護を提供されない可能性のあるより短期間の所有者鍵の

受取りに署名するのに使用される。短期間の鍵は、実質的に、カストディアン（すなわち、カストディアンとしての所有者）になる。

【００２６】

[0033]実施形態はまた、自動の鍵ローリングを含むこともできる。ＫＤＳ２０６がＰＤを解読して開くよう求められる場合、それは常に、何らかのエンティティ（ファブリックホストなど）が暗号化済みデータを読み取ろうとしているからである。このような動作は、ＶＭを内部に移行することや、新しい保護付きＶＭを何らかの暗号化済み「カスタマイゼーションデータ」から作成することなどの、「イングレス（ｉｎｇｒｅｓｓ）」フローに関連する。ＫＤＳ２０６は、要求された鍵ならびに別のＰＤでこのような要求に応答するが、この別のＰＤは、後続の「イーグレス（ｅｇｒｅｓｓ）」フロー中で使用されるものであり、イーグレスフローは通常、ＶＭを別のホスト２１３またはオフラインストレージ（例えばマシンストレージ２０３）に移行することを意味する。このセットアップは、ＫＤＳ２０６がイングレス時に１回だけ接触されれば済むことを保証し、このイングレス時は、ホスト正常性が評価される時である。ＫＤＳが２回以上接触されない場合、追加のホスト評価がイーグレス時に行われることはない。しかし、実施形態は、ＫＤＳ２０６への複数のラウンドトリップを可能にしてもよい。というのは、それが行われることに害はないからである。ホスト評価は、実施形態がＫＤＳに接触するたびに実施されてよい。したがって、ＫＤＳがイーグレス時に接触されない場合は、イーグレス時のホスト評価もない。

【００２７】

[0034]いくつかの実施形態では、任意の時点で、所有者とカストディアンのどちらかの、厳密に１つのエンティティが、ＰＤの「ガーディアン」として指定される。これは、ＰＤを作成してそれに署名したエンティティである（例えば１０３に示されるように）。ＰＤが所有者からカストディアンに、またあるカストディアンから別のカストディアンに移動するのに伴い、後見は変化することになる。しかし、同じＫＤＳ２０６がＰＤを扱う限り、後見は同じのままであることになる。任意のカストディアンまたは所有者が、現在のガーディアンが関与することなしに、既存のＰＤを「引き継ぐ」（そのガーディアンになる）ことができる。

【００２８】

[0035]以下に、一例の数学的に厳格な例示を示す。このセクションでは、以下の表記を利用することになる。

・　K_0、K_i、ＳＴ、ＴＴ　－　フローに関与するエンティティ（テナントK_0、カストディアンK_1、ソースＴＥＥ　ＳＴ、ターゲットＴＥＥ　ＴＴ）。

【００２９】

・　N_{EPub}、N_{EPri}　－　何らかのエンティティＮの公開暗号化鍵および秘密暗号化鍵。N_Eは、N_{EPub}の省略表現である。

・　N_{SPub}、N_{SPri}　－　何らかのエンティティＮの公開署名鍵および秘密署名鍵。N_Sは、N_{SPub}の省略表現である。

【００３０】

・

【００３１】

【数１】

$$\sum_{M_S}^{SN} N_K$$

【００３２】

－ エンティティMによって、その秘密鍵M_{SPri}で署名することによって発行された、エンティティN（所有者またはカストディアン）の証明書。エンティティのサブジェクト名SNと、その公開鍵（署名または暗号化）N_Kとを指定する。（署名鍵は、自己署名された証明書に対する対応する公開鍵に署名した可能性がある。）

・ TK_i、TK_e － 対称トランスポート鍵（例えば、TK_iはイングレス鍵であってよく、TK_eはイーグレス鍵であってよい）。いくつかの実施形態では、トランスポート鍵は、直接に使用されるように意図されるのではなく、保護付きVMペイロード全体の様々な部分に対する暗号化鍵および認証鍵を作製するための鍵導出関数への入力として使用されるように意図される。しかし、種々の鍵導出方式が使用されてよい。

【００３３】

・ ε（K）［M］ － 鍵Kを使用して暗号化されたメッセージM。Kは、コンテキストに応じて対称または非対称である場合がある。

・ α（K）［M］ － 対称鍵Kを使用して認証されたメッセージM。

【００３４】

・ σ（K）［M］ － 非対称鍵Kを使用して署名されたメッセージM。

・ M１｜｜M２ － メッセージM１とM２との連結。

[0036]さらに、フロー記述における省略表現は、より複雑なメッセージを表すために変数を使用することになる。特にホストへのイングレス時、下記のメッセージを扱う可能性が高い。

【００３５】

・ TK－e：＝KDF（TK，”e”） － 導出された暗号化鍵（KDFは「key derivation function（鍵導出関数）」を表し、「e」は、暗号化鍵の導出を署名鍵（例えば）の導出と区別するのに例えば使用される、KDFへの入力である。しかし、これは一例に過ぎず、他の任意の適切な入力も同様に使用されてよいことに留意されたい。）

＊ これは、ペイロードを暗号化する目的でTKから導出される対称鍵である。

【００３６】

＊ 保護付きVMペイロードの種々の部分（例えば、暗号化されたvTPM状態は、そのような部分のうちのほんの１つである）に対応するいくつかのこのような暗号化鍵が、単一のトランスポート鍵から導出されなければならない場合があることに留意されたい。

【００３７】

・ TK－a：＝KDF（TK，“a”） － 導出された認証鍵

＊ これは、HMAC関数（例えば）を介してペイロードを認証する目的でTKから導出される対称鍵である。

【００３８】

＊ 認証された暗号化の目的で、いくつかのこのような認証鍵を単一のトランスポート鍵から導出する必要がある場合があることに留意されたい。

・ A：＝ε（TK－e）［P］｜｜α（TK－a）［P］ － 暗号化され認証されたペイロードP

＊ トランスポート鍵TKに対応する、暗号化鍵およびその対になる認証鍵を使用して作成された、認証され暗号化されたペイロード（vTPM状態など）。

【００３９】

[0037]ここで示される例では別々の認証鍵と暗号化鍵が例示されるが、他の実施形態では、いくつかの暗号アルゴリズムは暗号化と認証の両方に同じ鍵が使用されるのを許容することを理解されたい。したがって、本明細書で例示されるデータ構造は、他の実施形態では単純化されることが可能である。

【００４０】

[0038]PDは、トランスポート鍵の複数のラッピングを含むことになる。１つは所有者

によるラッピング（以下、これはタイプ「Ｂ」のメッセージと呼ばれる）であり、０以上のラッピングは、所有者によって委任されたカストディアンによるラッピング（これらはタイプ「Ｃ」と称されるメッセージである）である。

【００４１】

・

【００４２】

【数２】

$$B := \sum_{K_{0S}}^{SK_0} K_{0S} \,||\, \sum_{K_{0S}}^{SK_0} K_{0E} \,||\, \varepsilon(K_{0E})[TK]$$

【００４３】

－　所有者によってラッピングされた鍵

＊　これは、ＶＭの所有者に対して暗号化されたトランスポート鍵である。これにより、ＶＭは、任意の時点でその所有者によって入手され得る。

【００４４】

＊　所有者のエントリのみが、署名鍵に対する、委任されない（自己署名された）証明書を含むことができる。

＊　このメッセージに対して３つの部分がある。すなわち、所有者の署名鍵に対する証明書と、所有者の公開暗号化鍵に対する所有者発行の証明書と、公開暗号化鍵を使用するトランスポート鍵の暗号化とである（所有者の署名鍵に対する証明書は自己署名されてよいが、自己署名される必要はなく、すべての関与者によって相互に信頼される証明書権限者によって発行されてもよい）。

【００４５】

・

【００４６】

【数３】

$$C := \sum_{K_{iS}}^{SK_j} K_{jS} \,||\, \sum_{K_{jS}}^{SK_j} K_{jE} \,||\, \varepsilon(K_{jE})[TK]$$

【００４７】

－　カストディアンによってラッピングされた鍵

＊　これは、ＶＭのカストディアンに対して暗号化された同じトランスポート鍵である。

【００４８】

＊　K_jは、トランスポート鍵がそれに対して暗号化される「現在の」カストディアンである。

＊　K_iは、カストディアン権限をK_jに委任する、チェーン中の「前の」カストディアンである（所有者の場合はｉ＝＝０）。

【００４９】

＊　ＶＭに対して、０以上のカストディアンがいる可能性がある。したがって、ＰＤ内部で、タイプＣのメッセージが０以上ある可能性がある。

[0039]トランスポート鍵の、異なるラッピングが、結合されてセットにされる。これは行列として以下に例示され、各行は、所有者またはカストディアンに対応するいくつかの連結されたエントリを含む。所有者は、そのように識別される（文字「ｏ」を使用して）。現在のガーディアンは、アステリスクによってそのようにフラグが立てられる。以下に示される例では、ＰＤは、ＫＤＳ２０６によって消費されるために所有者によって最初に作成されたときに存在する形状にある。鍵の異なるラッピングの結合は、タイプＤのメッ

セージである。単一のカストディアンの場合に所有者によって生成されたメッセージは、以下のように見えることになる。

【００５０】

【数４】

$$D_0 := \begin{pmatrix} * & o & \sum_{K_{0S}}^{SK_0} K_{0S} \,||\, \sum_{K_{0S}}^{SK_0} K_{0E} \,||\, \varepsilon(K_{0E})[TK] \\ - & - & \sum_{K_{0S}}^{SK_1} K_{1S} \,||\, \sum_{K_{1S}}^{SK_1} K_{1E} \,||\, \varepsilon(K_{1E})[TK] \end{pmatrix}$$

【００５１】

ＫＤＳ２０６は、このメッセージを消費し、それ自体をガーディアンにし、追加のカストディアンを加えることができる。この結果、以下のように見えるＰＤとなる。

【００５２】

【数５】

$$D_1 := \begin{pmatrix} - & o & \sum_{K_{0S}}^{SK_0} K_{0S} \,||\, \sum_{K_{0S}}^{SK_0} K_{0E} \,||\, \varepsilon(K_{0E})[TK] \\ * & - & \sum_{K_{0S}}^{SK_1} K_{1S} \,||\, \sum_{K_{1S}}^{SK_1} K_{1E} \,||\, \varepsilon(K_{1E})[TK] \\ - & - & \sum_{K_{1S}}^{SK_2} K_{2S} \,||\, \sum_{K_{2S}}^{SK_2} K_{2E} \,||\, \varepsilon(K_{2E})[TK] \end{pmatrix}$$

【００５３】

[0040]タイプＤのメッセージの整合性および真正性が、タイプＥのメッセージによって提供される。このメッセージは、ガーディアンがタイプＤのメッセージによって保護される鍵を実際に知っていてメッセージを実際に書いたことの証明としての働きをする。

【００５４】

・　E_i：＝α［ＴＫ－ａ］（D_i）｜｜σ（G_{SPri}）［D_i］

ここで、G_{SPri}は、「ガーディアン」としてマークされたメッセージＤ中のエントリの秘密署名鍵である。

【００５５】

[0041]最後に、タイプＦのメッセージが、実際の「ＰＤ」、すなわち、メッセージＤとＥとの連結である。

・　Ｆ：＝Ｄ｜｜Ｅ

[0042]いくつかの実施形態では、ＰＤデータフォーマットはまた、ヘッダ（ＰＤブロブ（ｂｌｏｂ）バージョン番号などの情報を通信するための）と、暗号化方式の指定（ｃｒｙｐｔｏｇｒａｐｈｉｃ　ａｇｉｌｉｔｙ）（各鍵についての、暗号文、モード、および鍵サイズの選択）のためのプロビジョンと、も含む。

【００５６】

[0043]何らかのペイロードに対するＰＤを伴う典型的なフローは、ホストがイングレス鍵TK_iのＰＤ　F_iをＫＤＳに送ること、およびＫＤＳが以下の２つの情報で応答することを含む。

【００５７】

１）ホストのＴＥＥによって消費されるための２つのトランスポート鍵（イングレスとイーグレス、すなわちTK_iとTK_e）のラッピング。このラッピングは、イングレス鍵とイーグレス鍵との連結の、認証された暗号化を含む。

【００５８】

＊　ＴＷ　－　ＫＤＳによって生成されたラッピング鍵

＊　ＴＷ－ｅ、ＴＷ－ａ　－　ＴＷから導出される暗号化鍵および認証鍵

＊　Ｈ：＝ε（ST_{EPub}）［ＴＷ］｜｜ε（ＴＷ－ｅ）［TK_i｜｜TK_e］｜｜α（ＴＷ－ａ）［TK_i｜｜TK_e］

２）イーグレスペイロードと共に含まれることになるTK_eの周りに構築されたイーグレス鍵ＰＤ　F_e。これは、ホストのＴＥＥをターゲットとせず、所有者によってまたはカストディアンのうちの１つによってしか解読され得ない。

【００５９】

[0044]場合によっては、ホストは、複数のイングレス保護記述子を提供することになる。その場合、ＫＤＳは、すべての保護記述子が厳密に同じ所有者を有することを保証することになる（委任チェーンのルートにある自己署名された証明書から証明されるように）。結果的なイーグレス保護記述子は、イングレス保護記述子からのすべてのカストディアンのスーパーセットとなり、メッセージＨは、以下のように見えることになる。

【００６０】

Ｈ：＝ε（ST_{EPub}）［ＴＷ］｜｜ε（ＴＷ－ｅ）［TK_{i1}｜｜TK_{i2}｜｜．．．｜｜TK_{in}｜｜TK_e］｜｜α（ＴＷ－ａ）［TK_{i1}｜｜TK_{i2}｜｜．．．｜｜TK_{in}｜｜TK_e］

[0045]すでに言及されたように、ＫＤＳ２０６は、タイプＦのブロブが保護するペイロードのタイプを知らない。ペイロードは、ｖＴＰＭ状態、ＶＭに関するプロビジョニング情報、または全く異なる何か、である可能性がある。いくつかの実施形態では、ペイロードは、「マスター」トランスポート鍵から導出された暗号化鍵を使用して暗号化され、対応する認証鍵を使用して認証される（タイプＡのメッセージによって例示されるように）。しかし、他の導出構成を構築して、同じまたは類似の全体的効果が達成されてもよい。

【００６１】

[0046]いくつかの実施形態では、各ホスト（例えば、ホスト２０７および２１３）は、ＶＭをホストすることまたは移行フローに参加することができるようになる前に、アテステーションを完了する。ホストアテステーションサービス（ＨＡＳ）２１４を用いてアテステーションをうまく完了すると、ホストに正常性証明書が発行される。証明書の中の鍵は、いくつかの実施形態では、ホストの信頼されるサブシステムに対するＴＥＥ公開暗号化鍵である。ホストはその後、正常性証明書をＫＤＳ２０６に提示し、ＫＤＳ２０６は、ホストのＴＥＥに対する機密データ（例えばｖＴＰＭ暗号化鍵）を暗号化することによって応答する。この場合、正常性証明書を用いた「認証」はなく、ホストを認証するために保持の証明がＫＤＳ２０６によって必要とされることはないことに留意されたい。単に、ホストは、それが提示したいどんな正常性証明書でも自由に提示することができるが、対応する秘密鍵を有さない場合は、ＫＤＳから受け取った応答を理解できないことになる。

【００６２】

[0047]ここで図３が参照されるが、図３は、非常に具体的な一例に関するフロー３００を示す。図３を参照すると、以下の通りである。

１．ホストが、アテステーションサービスに接触することによってアテステーションを開始する。

２．アテステーションサービスが要求を発行する。ＴＰＭ技術を使用する実施形態では、これは、プラットフォーム構成レジスタ（ＰＣＲ）リード要求であってよい。図３ではこれは単一の要求／応答交換として示されているが、実際には２つの工程を要する可能性が高いであろう。すなわち、セッションを確立するための１つの工程と、要求を満たすためのもう１つの工程である。

・　ＴＰＭ２．０デバイスを使用する実施形態では、２つの異なるアテステーションモードが可能である。すなわち、一方は、サービスによって提供されるノンスに対する従来のトラステッド・コンピューティング・グループ（ＴＣＧ）クォート（ｑｕｏｔｅ）であり、もう一方は、認証されたソルト付き（ｓａｌｔｅｄ）セッションに対する「直接ＰＣＲリード」である。これは一例に過ぎないことに留意されたい。他の実施形態では、例えばＴＰＭ１．２が使用されてもよいが、直接ＰＣＲリード機能は利用可能でないことに

なる。
３．デバイスが、要求されたＰＣＲの値をＴＣＧログと共に供給することによって、ＰＣＲリード要求を満たす。
４．デバイスが、ＰＣＲリード要求に対する応答を含む「アテステーション要求」を組み立てる。
５．サービスが、供給されたＴＣＧログやＰＣＲ値を調べ、任意選択でポリシーを参照して、ホストによって供給されたＴＣＧログの内容が正常である（単にＰＣＲ値にマッチすることに加えて）かどうかを決定する。
６．サービスが、正常性証明書を作製する。この正常性証明書は、ＴＥＥ公開鍵と、正常性アテステーションサービス２１４によって決定されたホスト正常性の測定値のセットとを、内的にエンコードする。
７．この正常性証明書は、後でＶＭ起動フローおよび移行フロー中で使用されるように、ホストによって記憶される。

【００６３】
[0048]仮想マシンのホスティングおよび移行
[0049]ＶＭ移行は、いくつかの移動性部分を含む複雑なフローである。これは、ソースホストサーバーと、ターゲットホストサーバーと、移動を調整する管理サービスとを含む。このセクションでは主に、ＶＭ移行に関係する鍵管理に焦点を合わせる。ＶＭ移行は、例えば、以下のケースのうちの１つで行われることがある。

【００６４】
・　同じカストディアンの権限下で、あるホストから別のホストへ。
・　所有者からカストディアンへ。
・　あるカストディアンから別のカストディアンへ。
・　カストディアンから再び所有者へ。

【００６５】
[0050]各ケースで、ＶＭは、悪意のある区域（ネットワーク、ストレージ、アドレス空間）を横断すると仮定される。前述の、ペイロードおよびＰＤ自体を保護する方式は、ＶＭおよびその鍵材料の安全なエンドツーエンド・トランスポートを達成する。

【００６６】
[0051]各ＶＭ移行シナリオで、保護付きＶＭは、その全体が、あるホストから別のホストに、またはストレージに／ストレージから、動く。一実施形態では、ＶＭは、以下の構成ブロックを含む。

【００６７】
・　ｖＴＰＭ暗号化鍵
・　ｖＴＰＭ状態
・　ＶＭメタデータ
・　１つまたは複数のＶＨＤ
・　ＶＭメモリー状態
・　ＶＭデバイス状態

【００６８】
[0052]ｖＴＰＭ暗号化鍵（例えば、トランスポート鍵、またはトランスポート鍵から導出された鍵）は、ＶＭの所有者またはカストディアンのみがそれらへのアクセスを得ることができるように、暗号化される。次いで、トランスポート鍵は、それらを見ることを許可されたホストのＴＥＥに渡される。

【００６９】
[0053]例示的な一例では、ｖＴＰＭ状態は、トランスポート鍵、またはトランスポート鍵から導出された暗号化鍵を使用して暗号化される。暗号化はさらに、やはり同じトランスポート鍵から導出された認証鍵を使用して認証される。ｖＴＰＭ状態、およびそれを保護するトランスポート鍵は、保護されない状態でホストのＴＥＥを離れることはない。しかし、これは一例に過ぎないこと、および、鍵を使用して機密性および整合性を保護でき

る方式として、本発明の実施形態の範囲内に入るほぼ無限の数の方式があることを理解されたい。

【００７０】

[0054]メタデータの機密部分もまた、やはりトランスポート鍵から導出される異なる認証鍵を使用して認証されてよい（いくつかの実施形態では、適切な暗号実践のためには、ｖＴＰＭ暗号化済み状態を検証するのに使用される認証鍵はＴＥＥを離れるべきでなく、したがって別個の鍵が使用される）。より一般的には、示される例では、これらの秘密鍵を伴うすべての暗号は、ＴＥＥ、またはＴＥＥエクステンションの１つを使用して行われる（このようなエクステンションは、ハイパーバイザーによるコード整合性の強制を使用して保護される、ホストＯＳのカーネルであってよい）。トランスポート鍵をルートとする鍵階層が構築されてよく、ＶＭ状態の様々な部分が、この階層中の鍵を使用して暗号化される。

【００７１】

[0055]ＶＨＤは、ｖＴＰＭによって保護されたＦＶＥＫを使用して暗号化されると想定され、したがって、移行アーキテクチャは、ＶＨＤを保護するための追加の試みは行わない。加えて、テナント２００が、このテナントに属するＶＭをサービスプロバイダー２０１からダウンロードして、これを直接に、すなわちＫＤＳ２０６を必要とせずに、実行することも可能である。

【００７２】

[0056]最後に、ＶＭメモリー状態およびデバイス状態は、マスタートランスポート鍵をから導出された鍵を使用して暗号化される。これらも同様に、対応する認証鍵を使用して認証されてよい。

【００７３】

[0057]保護付きゲストＶＭを含むフローが開始できるようになる前に、いくつかの必要条件が満たされる。

１．（任意選択－ＶＭＭ２０２が移行フローを調整する場合のみ）ワシントン州レドモンドのＭｉｃｒｏｓｏｆｔ　Ｃｏｒｐｏｒａｔｉｏｎから入手可能なＳＣＶＭＭなどの管理サービスＶＭＭが、稼働しており、ソースサーバーとターゲットサーバーの両方に利用可能である。

２．鍵配布サービス２０６およびホストアテステーションサービス２１４が起動状態である。

３．ソースホストサーバー（例えばホスト２０７）が、稼働しているゲスト仮想マシン２０４を有する。

４．移行の場合、ターゲットホストサーバー（例えばホスト２１３）が、ＴＥＥ対応であり、稼働している。

【００７４】

[0058]いずれかの実際の移行が行われる前に、一連のステップがとられる。いくつかの実施形態で存在する場合のある極端な時間制約のせいで、特にホット移行は、実際に移行が試みられる前に（時として十分に前もって）これらのステップがとられることを必要とする。以下のリストは、新しいプロビジョンのみを意図的にリストしており、周知のステップ（ソースホストが、新しいゲストＶＭをホストするためのリソースの利用可能性をターゲットに照会するなど）は省略していることに留意されたい。

【００７５】

１．ソースホスト（例えばホスト２０７）とターゲットホスト（例えばホスト２１３）の両方が、アテステーション（例えば前述のような）を完了済みである。

２．ターゲットホスト（例えばホスト２１３）が、イングレス移行ポリシーを満たす。そうでない場合は、ターゲットホストはＶＭ状態を受け取る資格がない。これを早くから確立するのは有用だが、実際の移行時に、ホストが移行ポリシーに準拠することがＫＤＳ２０６によってチェックされるべきである。

３．アテステーションが成功すれば、エンティティ間のデータの暗号化のおかげで、ソ

ースホストおよびターゲットホストのルートＴＥＥ（具体的には、ルートＴＥＥ内部で稼働している移行エージェントトラストレット）とＫＤＳ２０６との間で安全な通信を生み出すことができる。

【００７６】

[0059]既存のＶＭを保護付きＶＭに変換する

[0060]テナントの環境におけるＶＭをｖＴＰＭに関連付けることによって、このＶＭは、「保護付き」ＶＭステータスに変換されることが可能である。この後には、通常、ＶＭをサービスプロバイダーのファブリックに移行することが続くと予想される。いくつかの実施形態では、通常のＶＭから保護付きＶＭを作成するには、以下のことが行われる。

【００７７】

１．ＶＭに対するＰＤが作成され、正しい所有者およびカストディアンでポピュレートされる。

２．ｖＴＰＭが作成され、任意選択で、適切な権限者（例えば、テナント自体の証明書権限者など）によって署名された保証証明書を発行することによって証明される。

３．ＢｉｔＬｏｃｋｅｒなどの暗号化技術がＶＭに対して有効化され、そのＶＨＤが完全に暗号化される。

【００７８】

[0061]ＶＭのＶＨＤが完全に暗号化されると、ＶＭは、サービスプロバイダーのファブリックに移行されるのに「十分に安全」である。

【００７９】

[0062]新品のＰＤの作成（既存のＰＤの再暗号化とは対照的に）は、他のフローとは異なる。というのは、ＰＤを伴う他の場合では、ホストは、ＫＤＳ２０６に既存のＰＤをうまく解読させることによってＫＤＳ２０６を認証するからである。新品のＰＤ作成の場合、この元のＰＤはまだ存在しない。

【００８０】

[0063]いくつかの実施形態は、「ヌル」ＰＤ（所有者およびあらゆる許容されたテナントによってラップされた既知のイングレス鍵）で開始することによって、新規ＰＤ作成の周りのフローをブートストラップすることができる。このヌルＰＤのガーディアンは、所有者自体である。ＰＤは、ＶＭ作成のタスクを課せられた、テナントの任意のホストによって利用可能である。新しいＶＭに対するＰＤを作成するには、ホストは、ヌルＰＤをＫＤＳにサブミットし、ＫＤＳは、２つの鍵（イングレスのための既知の鍵、および新しいイーグレス鍵）、ならびにイーグレス鍵の周りのＰＤを返す。このイーグレス鍵は、ＶＭをマシンストレージ２０３（もしくは他のストレージ）またはサービスプロバイダーに外部移行するために、ホストによって使用される。

【００８１】

[0064]イーグレス鍵の作成に続いて、ホストは、ｖＴＰＭを作成し証明することができる。ｖＴＰＭが作成され証明されると、ｖＴＰＭは、ＶＭのメタデータに添付されてよい。次いで、ＶＭのＯＳが再開され、新しいｖＴＰＭ仮想デバイスは次のブート時にＯＳにさらされるようになる。この時点で、ＯＳは、ＢｉｔＬｏｃｋｅｒを使用するなどして、そのＶＨＤを自由に暗号化することができる。

【００８２】

[0065]ホスト上でのＶＭ起動

[0066]ホストがＶＭを外部移行できるようになる前に、ホストは最初に、ＶＭをロードして実行する必要がある。これは、ＶＭをオフラインストレージからダウンロードすることによって達成される。ＶＭのＶＨＤは暗号化されている（例えばＢｉｔＬｏｃｋｅｒを用いて）と想定され、鍵はＶＭのｖＴＰＭの内部に封印されている。ｖＴＰＭ状態は、トランスポート鍵TK_i（メッセージA_i）と、ＫＤＳがホストのために鍵TK_iをアンラップするのを助けるための関連データ（ＰＤ　F_i）とを使用してラップされる。A_iとF_iは両方とも、ＶＭメタデータの一部である。図４を参照すると、段階的に、ＶＭ起動フロー４００は以下の通りである。

【００８３】
１．管理サービスが、ホストに接触して、起動されるべきＶＭへのリンクをホストに提供することによって、ホスト上でのＶＭ起動を開始する。
２．ホストが要求を受け取る。
３．ホストがオフラインストレージに手を伸ばしてＶＭをダウンロードする。
４．ホストが、ＶＭをダウンロードした後、ＶＭメタデータの内部を見て、ＫＤＳへの要求の構築に進む。
・　ブロブA_i（ラップされたｖＴＰＭ状態）およびF_i（ＫＤＳによってラップされたｖＴＰＭ暗号化鍵に対するＰＤ）を抽出する。
・　ホストの正常性証明書が付随したＰＤ　F_iをＫＤＳに送る。これは、そのＴＥＥ公開鍵ST_{EPub}を含む。
５．ＫＤＳが、要求を受け取り、正常性証明書を認証する。
・　証明書が失効していない。
・　証明書上の署名が有効である（証明書が自己署名されている場合は、署名がアテステーションサービスの署名とマッチする。そうでない場合は、証明書が、ＫＤＳが認識する証明書権限者（ＣＡ）からくるものである）。
・　（任意選択）証明書中でエンコードされた発行ポリシーが、現在のホスト正常性ポリシーによる正常なホストに対応する。
・　注：ＫＤＳは、要求が、正常性証明書に対応する秘密鍵を保持するホストからきたことを検証しない。その応答は、要求元の公開ＴＥＥ鍵を使用して暗号化されることになり、したがって攻撃者には無用であることになる。
６．ＫＤＳが、ブロブF_iを解読し、応答を計算する。
・　イングレスＰＤを処理してイングレストランスポート鍵を抽出する。
＊　F_iからD_iおよびE_iを抽出する。
＊　D_iから、B_i、ならびに、０以上のタイプC_iのメッセージを抽出する。
＊　メッセージB_iおよびC_iのうちで、ＫＤＳに対応する行を突き止める。
＊　ガーディアンの署名証明書から始まる証明書チェーンを構築して、チェーン中のすべての署名鍵証明書が所有者の署名証明書にロールアップすることを確実にする（所有者の署名証明書は、自己署名され得る唯一のものであり、自己署名されていない場合は、例えばバージニア州レストンのＶｅｒｉｓｉｇｎなど、信頼される権限者によって署名されていることを確認する）。
＊　メッセージB_iおよびC_iのすべてを確認するのが有益であろう。
＊　ＫＤＳに対応するメッセージB_iおよびC_iから、TK_iを復号する。
＊　TK_iからTK_i－ａを導出する。
＊　TK_i－ａを使用して、E_i内部のＨＭＡＣと、ＰＤに対するガーディアンの署名とを確認する。
・　イーグレストランスポート鍵TK_eを生成する。
・　TK_eに対するＰＤを生成する。
＊　T_{EPub}（B_iから得られる）およびTK_eから、B_eを生成する。
＊　K_{EPub}およびTK_eからC_eを生成する（これは、すべてのＣメッセージについて実施され、Ｃメッセージは０以上存在する場合がある）。
＊　B_eおよびC_eからD_eを生成する。ガーディアンにある間、それ自体をガーディアンとしてマークする。
＊　D_eをTK_e－ａで認証し、その結果、メッセージE_eが得られる。
＊　その署名鍵を使用してD_eに署名する。
＊　D_eとE_eとを連結してイーグレスＰＤ　F_eにする。
・　TK_iおよびTK_eをホストに送り返す準備をする。
＊　ＴＷを生成し、このＴＷからＴＷ－ｅおよびＴＷ－ａを生成する。
＊　ＴＷ、ＴＷ－ａ、ＴＷ－ｅ、TK_i、TK_e、およびST_{EPub}から、Ｈを生成する。

・　ホストへの応答は、連結されたメッセージH｜｜F_eを含む。
７．ホストが、ＫＤＳから応答を受け取り、メッセージHをA_iと共にＴＥＥに渡す。
８．ＴＥＥが、ＫＤＳからの応答を扱う。
・　Hから、ST_{Epri}を使用してＴＷを復号する。
・　ＴＷからＴＷ－ｅおよびＴＷ－ａを導出する。
・　ＴＷ－ｅを使用してTK_iおよびTK_eを復号する。
・　ＴＷ－ａを使用してTK_iおよびTK_eの暗号化を認証し、したがって、ＰＤを解読するＫＤＳの能力（したがってＫＤＳの真正性）を認証する。
・　TK_iから暗号化鍵および認証鍵TK_i－ｅおよびTK_i－ａを導出する。
・　TK_i－ｅを使用してｖＴＰＭ状態をA_iから復号する。
・　TK_i－ａを使用してｖＴＰＭ状態暗号化を認証する。
９．ｖＴＰＭ状態が復号されると、ＴＥＥはｖＴＰＭを起動する。
・　ＴＥＥは、ｖＴＰＭを外部移行（すなわち、メッセージA_eを編成してA_e｜｜F_eをターゲットに送る）しなければならないような時まで、TK_eの値を堅持する。
１０．ホストは今や、ＶＭ起動シーケンスを完了することができる。

【００８４】

[0067]実際の移行

[0068]いくつかの実施形態では、前方秘匿性の理由で、各移行インスタンスは、その移行のみに有効な一意の鍵を必要とする。この設計によって対応される重要なセキュリティ考慮事項は、ＶＭＭ２０２サービスにほとんどまたは全く信頼が置かれないことである。

【００８５】

[0069]図５に、ＴＥＥ状態を伴う移行のための、可能な１つのメカニズムが示されている。図５に示されるフロー５００は、いくつかのサービスを利用する。すなわち、「管理サービス」（ＳＣＶＭＭなど）、「アテステーションサービス」、および「鍵配布サービス」である。アテステーションサービスおよび鍵配布サービスは、等しく信頼される（かつＳＣＶＭＭよりも信頼される）ので、同じ場所に位置してよい。

【００８６】

[0070]実際の移行は、「鍵合意」で開始し、その後に実際のデータ転送が続く。
１．管理サービスが、ソースホストに接触して、移行されることになる仮想マシンＩＤと、ターゲットホストの識別とをソースホストに提供することによって、移行を開始する。
２．ソースホストが、イーグレストランスポート鍵ＰＤ　F_e（ターゲットから見ればこれはイングレスＰＤである）と共に、移行要求をターゲットホストに回す。
３．ターゲットホストが移行要求を受け取る。
４．ターゲットホストが、ソースホストから受け取ったメッセージF_eをそのホスト正常性証明書と共にＫＤＳに送ることによって、ターゲットホストのためにトランスポート鍵をアンラップするようＫＤＳに要求する。
５．ＫＤＳが、ソースホストについて上述されたのと同様にして、ターゲットホストによって提示されたホスト正常性証明書を検証する。
６．ＫＤＳが、前述のＶＭ起動について行われたのと同様の、状態暗号化鍵の処理を行う。ただし今回は、ＫＤＳは、TK_iおよびTK_eの代わりに、TK_eおよびTK_e'を返す。
７．ターゲットホストが、この応答を受け取り、これをアンラップのためにそのＴＥＥに渡す。
８．ターゲットホストのＴＥＥが、その秘密ＴＥＥ鍵TT_{Epri}を使用して、鍵TK_eを復号する。
９．ターゲットホストが、実際の移行に進む準備ができたことの信号ソースホストに送る。

[0071]これで、鍵合意の段階が完了し、最後の１つのステップ、すなわち実際の安全なデータ転送への土台が整う。

１０．最後に、上記のすべてのステップが達成された後、２つのサーバーは、状態の転送を開始することができる。状態は、ソース上でＴＥＥ移行トラストレットによって暗号化され、ターゲットＴＥＥに渡される。

１１．ソースおよびターゲット上のＴＥＥは、ｖＴＰＭ状態暗号化と、導出された鍵を使用する暗号化とを扱って、状態を一方から他方に転送する。

【００８７】

[0072]仮想マシンのバックアップおよび復元

[0073]鍵管理およびデータフローの観点からの、バックアップおよび復元フローは、ソースサーバーから何らかの保存データストレージ設備に外部移行して、それに続いて（何らかの後の時点で）同じデータをターゲットサーバーに移行することに、非常に類似する。フローはすでに、安全でない区域を横断するように設計されているので、このようにバックアップされた保存データは、どんな追加の保護も必要としない。唯一の要件は、ＶＭＩＤと、暗号化（ＫＤＳによる）された移行鍵とが、ＶＭ状態と共に記憶されることである。というのは、これらは、復元に必要な復号鍵の使用の封印を解除するために鍵配布サービスによって使用されるからである。

【００８８】

[0074]前方秘匿性のための要件は、ＶＭ状態をバックアップするホストが新しい暗号化鍵を得て、この暗号化鍵を使用してＶＭ状態を再暗号化した後でＶＭ状態をバックアップする場合に、最もよく満たされる。

【００８９】

[0075]クラスタ内部での仮想マシンのフェイルオーバー

[0076]クラスタ内部でのＶＭのフェイルオーバーもまた、移行の特別なケースと考えることができる。このケースでは、クラスタ中のすべてのノードが、クラスタによってホストされる各ＶＭ　ＩＤにつき同じＶＭメタデータを共有し、したがって、どんなフェイルオーバーにも先立って、移行鍵をＫＤＳから得ることができる。クラスタ中のノードは、ＶＭデータが位置するストレージを共有する。クラスタ中のすべてのノードは、どのラップされたＶＭ鍵を使用するかについて合意しているので、鍵合意の確立も同様に容易であり、それにより、速く安全なフェイルオーバーが保証される。特に、フェイルオーバーシナリオは前方秘匿性を必要とせず、したがって、ｖＴＰＭの暗号化に使用される鍵は、ＶＭがクラスタ内にある限り変化しない。

【００９０】

[0077]データセンター間での仮想マシンの移行

[0078]ＶＭに対するＰＤが両方のデータセンターをカストディアンとして含む限り、移行は、テナントとサービスプロバイダーとの間のケースと違わない。すなわち、受取り側のカストディアンがＰＤを解読して開き、新しいＰＤを生成し、それ自体をそのＰＤのガーディアンとして設定する。新しいＰＤ中のトランスポート鍵は、前と同様、所有者および各カストディアンに対して暗号化される。

【００９１】

[0079]次に、以下の考察では、実施され得るいくつかの方法および方法行為に言及する。方法行為は、何らかの順序で考察される場合があり、または特定の順序で行われるものとしてフローチャート中で示される場合があるが、特段の記載がない限り、または行為がその実施前に別の行為の完了に依存するという理由で必要性があるのでない限り、特定の順序付けは必要とされない。

【００９２】

[0080]ここで図６を参照すると、方法６００が示されている。方法６００は、コンピューティング環境で実践されてよい。この方法は、暗号化されたデータセットを管理するための行為を含む。この方法は、第１の復号鍵を取得すること（行為６０２）を含む。第１の復号鍵は、第１の暗号化メカニズムを使用して暗号化された暗号化済みデータセットを復号するのに使用されるように構成される。第１の暗号化メカニズムは、データセットを復号するのに使用され得る第１の復号鍵に関連する。

【００９３】
[0081]方法６００はさらに、第１の復号鍵を第２の暗号化メカニズムで暗号化すること（行為６０４）を含む。第２の暗号化メカニズムは、第１のエンティティによって使用される第２の復号鍵に関連し、したがって、第２の復号鍵は、最初に第２の復号鍵を使用して第１の鍵暗号化済み鍵を復号してからこの復号された第１の鍵を使用してデータセットを復号することによってデータセットを復号するために、第１のエンティティによって使用され得る。
【００９４】
[0082]方法６００はさらに、第１の復号鍵を第３の暗号化メカニズムで暗号化すること（行為６０６）を含む。第３の暗号化メカニズムは、第２のエンティティによって使用される第３の復号鍵に関連し、したがって、第３の復号鍵は、最初に第３の復号鍵を使用して第１の鍵暗号化済み鍵を復号してからこの復号された第１の鍵を使用してデータセットを復号することによってデータセットを復号するために、第２のエンティティによって使用され得る。
【００９５】
[0083]方法６００はさらに、第２の暗号化方法で暗号化された第１の復号鍵と、第３の暗号化方法で暗号化された第１の復号鍵とを少なくとも含むパッケージを作成すること（行為６０８）を含む。
【００９６】
[0084]この方法はさらに、ガーディアン署名でパッケージに署名すること（行為６１０）、および、第１の復号鍵から作成された署名でパッケージに署名すること（行為６１２）を含む。
【００９７】
[0085]さらに、該方法は、１つまたは複数のプロセッサーやコンピューターメモリーなどコンピューター可読媒体を含む、コンピューターシステムによって実践され得る。特に、コンピューターメモリーは、１つまたは複数のプロセッサーによって実行された場合に、実施形態にて言及の行為など様々な機能を遂行させる、コンピューター実行可能命令を保存し得る。
【００９８】
[0086]本発明の実施形態は、以下にて詳述する通り、コンピューターハードウェアを含む専用または汎用のコンピューターを含むかまたは活用し得る。本発明の範囲内の実施形態は、コンピューター実行可能な命令および／またはデータ構造を運搬または保存するための、物理的および他のコンピューター可読媒体をも含む。そのようなコンピューター可読媒体は、汎用または専用のコンピューターシステムからアクセスできる、利用可能ないかなる媒体であってもよい。コンピューター実行可能命令を保存するコンピューター可読媒体は、物理的保存媒体である。コンピューター実行可能命令を運搬するコンピューター可読媒体は、伝送媒体である。したがって、例を挙げると、制限なく、本発明の実施形態は少なくとも２種類の特徴的なコンピューター可読媒体、すなわち物理的コンピューター可読保存媒体と、コンピューター可読伝送媒体とを含み得る。
【００９９】
[0087]物理的コンピューター可読保存媒体の例として、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、ＣＤ－ＲＯＭまたは他の光学ディスクストレージ（ＣＤ、ＤＶＤなど）、磁気ディスクストレージまたは他の磁気ストレージデバイス、またはその他、所望のプログラムコード手段をコンピューター実行可能な命令またはデータ構造の形で保存するために使用可能かつ汎用または専用のコンピューターからアクセス可能な媒体が挙げられる。
【０１００】
[0088]「ネットワーク」は、コンピューターシステムおよび／またはモジュールおよび／または他の電子デバイスの間での電子データ輸送を可能にする、１つまたは複数のデータリンクとして定義される。コンピューターに至るネットワーク上または別の通信接続（有線、無線、または有線もしくは無線の組み合わせのいずれか）上で情報が転送または提

供される場合、コンピューターは適切に、接続を伝送媒体として捉える。伝送媒体の例として、所望のプログラムコード手段をコンピューター実行可能な命令またはデータ構造の形で運搬するために使用可能かつ汎用または専用のコンピューターからアクセス可能な、ネットワークおよび／またはデータリンクが挙げられる。上記を組み合わせたものも、コンピューター可読媒体の範囲に含まれる。

【０１０１】

[0089]さらに、様々なコンピューターシステムコンポーネントに到達後、コンピューター実行可能な命令またはデータ構造の形であるプログラムコード手段は、コンピューター可読伝送媒体から物理的コンピューター可読保存媒体へと（または逆方向）自動的に転送され得る。例えば、コンピューター実行な命令またはデータ構造は、ネットワーク上またはデータリンク上で受信された後、ネットワークインターフェースモジュール（例：「ＮＩＣ」）内のＲＡＭにおいてバッファー処理され、その後、最終的に、コンピューターシステムのＲＡＭおよび／またはコンピューターシステム側の低揮発性コンピューター可読物理保存媒体へと転送され得る。したがって、コンピューター可読物理保存媒体は、伝送媒体も活用（さらには主として活用）するコンピューターシステムコンポーネントに含まれ得る。

【０１０２】

[0090]コンピューター実行可能命令の例として、汎用コンピューター、専用コンピューター、または専用処理デバイスに対し、一定の機能または機能群を遂行させる命令およびデータが挙げられる。コンピューター実行可能命令の例として、アセンブリー言語など２進法、中間フォーマットの命令、さらにはソースコードも挙げられる。対象事項は構造的特徴および／または方法論的行為に特有の言語で記述されているが、添付の請求項において定義される対象事項は、記載の特徴または上記の行為に必ずしも限定されるわけではないことが、理解されなければならない。むしろ、記載の特徴および行為は、請求項の実施形態の例として開示される。

【０１０３】

[0091]当業者であれば、本発明はパーソナルコンピューター、デスクトップコンピューター、ラップトップコンピューター、メッセージプロセッサー、携帯端末、マルチプロセッサーシステム、マイクロプロセッサーベースまたはプログラム設定可能な家庭用電化製品、ネットワークＰＣ、ミニコンピューター、メインフレームコンピューター、携帯電話機、ＰＤＡ、ポケットベル、ルーター、スイッチなどを含む、多数の種類のコンピューターシステム機器構成を有するネットワークコンピューティング環境において実践され得ることを、理解することになる。本発明は、ネットワーク経由で（有線データリンク、無線データリンクまたは有線および無線の複合型データリンクにより）連動するローカルおよびリモートのコンピューターシステムがいずれも作業を遂行する、分散型システム環境においても実践され得る。分散型システム環境において、プログラムモジュールはローカルおよびリモート双方のメモリーストレージデバイスに配置され得る。

【０１０４】

[0092]代替的または付加的に、本明細書に記載の機能性は、少なくとも部分的に、１つまたは複数のハードウェアロジックコンポーネントによって遂行され得る。使用され得る典型的種類のハードウェアロジックコンポーネントの例として、制限なく、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、特定用途向け集積回路（ＡＳＩＣ）、特定用途向け標準製品（ＡＳＳＰ）、システムオンチップ（ＳＯＣ）システム、結合プログラム可能論理回路（ＣＰＬＤ）などが挙げられる。

【０１０５】

[0093]本発明は、その精神または特徴から逸脱することなく、他の特異的形態でも具現化され得る。記載の実施形態は、あらゆる態様において、単なる例示であり、制限的ではないと解釈されなければならない。したがって、本発明の範囲は、前述の説明によるのではなくむしろ、添付の請求項によって示される。請求項の同等性の意味および範囲に該当する変更はすべて、それらの範囲に含まれることになる。

【図１】

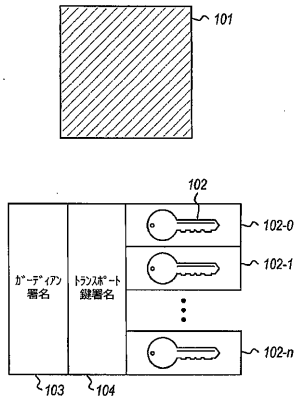

図１

【図２】

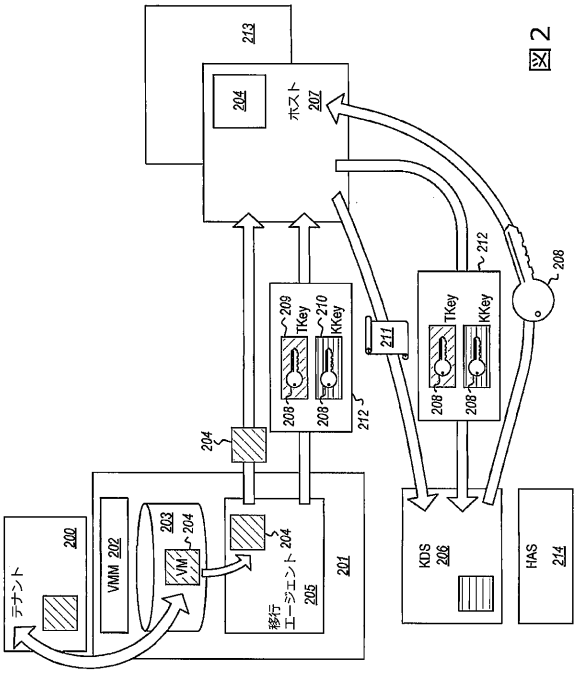

図２

【図３】

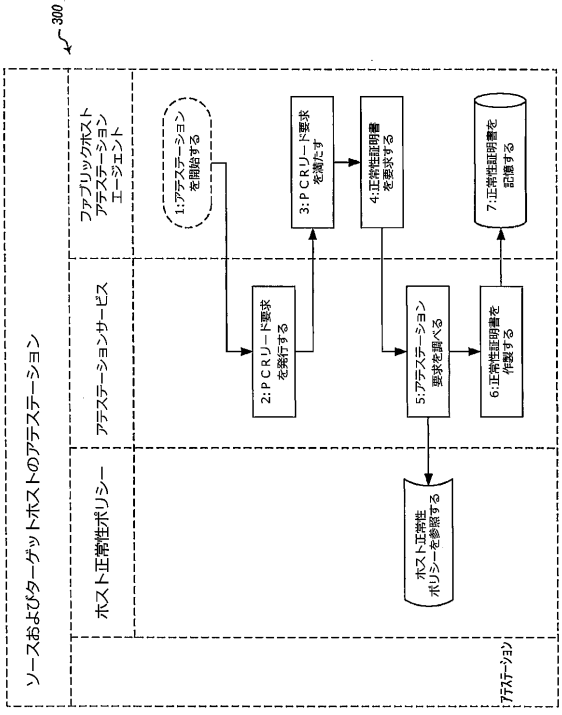

図３

【図４】

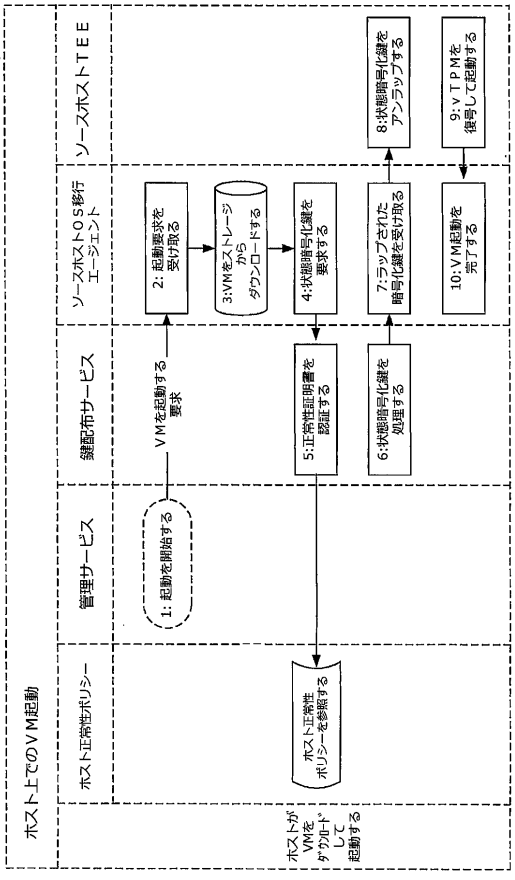

図４

【図５Ａ】

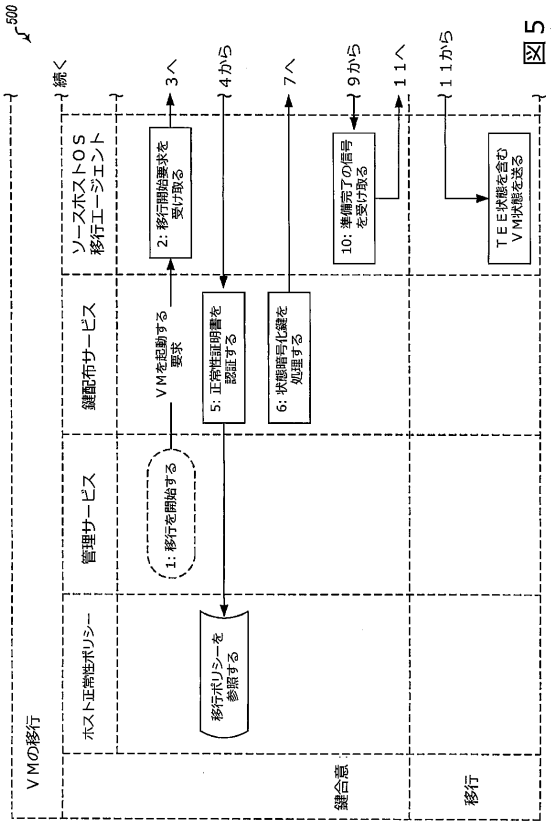

図５

【図５Ｂ】

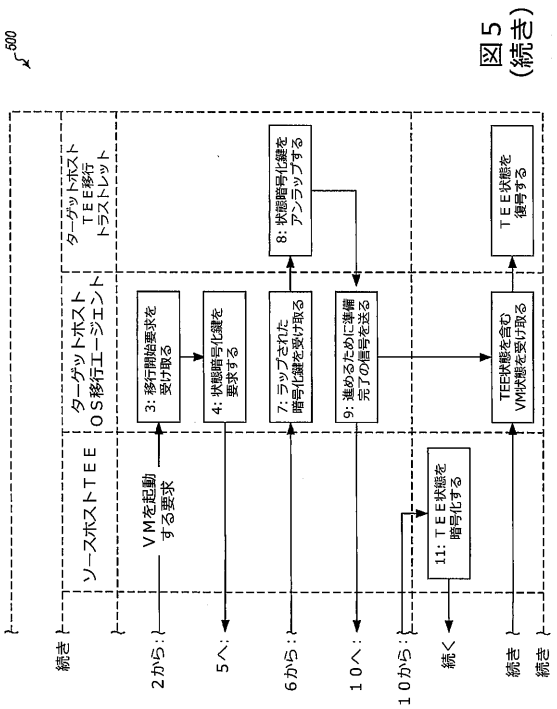

図５
(続き)

【図６】

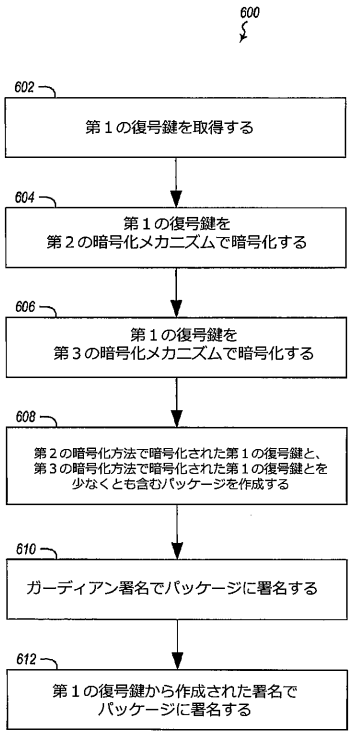

図６

フロントページの続き

前置審査

(74)代理人 100153028
弁理士 上田 忠
(72)発明者 ノヴァック，マーク・フィシェル
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，マイクロソフト コーポレーション，エルシーエイ－インターナショナル・パテンツ（８／１１７２）
(72)発明者 ベン－ズヴィ，ニル
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，マイクロソフト コーポレーション，エルシーエイ－インターナショナル・パテンツ（８／１１７２）
(72)発明者 ファーガソン，ニールズ・ティー
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，マイクロソフト コーポレーション，エルシーエイ－インターナショナル・パテンツ（８／１１７２）

審査官 青木 重徳

(56)参考文献 特開２０１１－０４８６６１（ＪＰ，Ａ）
国際公開第２００１／０６３８３１（WO，Ａ1）
米国特許出願公開第２０１１／０３０２４００（ＵＳ，Ａ1）
米国特許出願公開第２０１３／０３３９９４９（ＵＳ，Ａ1）
米国特許出願公開第２０１２／０１３７１１７（ＵＳ，Ａ1）
米国特許出願公開第２０１４／０１０８７８４（ＵＳ，Ａ1）
高橋 正和 ほか，Ｗｉｎｄｏｗｓ Ｖｉｓｔａで考える情報漏えい対策，日経コミュニケーション，日本，日経ＢＰ社，２００７年 9月１５日，第４９４号，ｐｐ．８６－９１

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ ９／１４
Ｇ０９Ｃ １／００
Ｈ０４Ｌ ９／０８
Ｈ０４Ｌ ９／３２