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

特許第6479758号
(P6479758)

(45)発行日　平成31年3月6日(2019.3.6)　(24)登録日　平成31年2月15日(2019.2.15)

(51)Int.Cl.		F I		
H04L 9/32	(2006.01)	H04L	9/00	675A
H04L 9/08	(2006.01)	H04L	9/00	601A
G06F 21/54	(2013.01)	H04L	9/00	601E
		G06F	21/54	

請求項の数 12　(全 37 頁)

(21)出願番号	特願2016-503153 (P2016-503153)
(86)(22)出願日	平成26年3月14日 (2014.3.14)
(65)公表番号	特表2016-513945 (P2016-513945A)
(43)公表日	平成28年5月16日 (2016.5.16)
(86)国際出願番号	PCT/US2014/029586
(87)国際公開番号	WO2014/144961
(87)国際公開日	平成26年9月18日 (2014.9.18)
審査請求日	平成29年3月14日 (2017.3.14)
(31)優先権主張番号	13/841,498
(32)優先日	平成25年3月15日 (2013.3.15)
(33)優先権主張国	米国(US)
(31)優先権主張番号	14/213,244
(32)優先日	平成26年3月14日 (2014.3.14)
(33)優先権主張国	米国(US)

(73)特許権者　502303739
オラクル・インターナショナル・コーポレイション
アメリカ合衆国カリフォルニア州９４０６５レッドウッド・シティー，オラクル・パークウェイ５００
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　ボワイエ，ジョン・ジュール・アレクサンダー
アメリカ合衆国、９４０６５　カリフォルニア州、レッドウッド・シティー、オラクル・パークウェイ、５００

最終頁に続く

(54)【発明の名称】コンピュータ上におけるアプリケーション間の信頼性の確立

(57)【特許請求の範囲】

【請求項１】

単一の演算装置上におけるソフトウェアアプリケーション間の信頼性を確立するための方法であって、

各々がオリジナルのコンパイル後のオブジェクトコードを有するソフトウェアアプリケーションを演算装置上に提供するステップと、

前記演算装置上にセキュリティマネージャアプリケーションを提供するステップと、

各ソフトウェアアプリケーションのための置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いて各ソフトウェアアプリケーションのオリジナルのオブジェクトコードを変更するステップとを含み、前記置換オブジェクトコードは、各ソフトウェアアプリケーションを前記セキュリティマネージャアプリケーションと通信するように変更するとともに、前記ソフトウェアアプリケーション間の共有の秘密にアクセスすることにより、前記ソフトウェアアプリケーション間に信頼関係を確立するように構成され、

各ソフトウェアアプリケーションの前記置換オブジェクトコードによって、前記セキュリティマネージャアプリケーションから前記共有の秘密を取得するステップを含む、方法。

【請求項２】

前記変更するステップの前に各々のソフトウェアアプリケーションの前記オリジナルのオブジェクトコードを署名ありの形式から署名なしの形式に変換するステップと、

前記変更するステップの後に各々のソフトウェアアプリケーションの前記置換オブジェクトコードを署名なしの形式から署名ありの形式に変換するステップとをさらに含む、請求項１に記載の方法。

【請求項３】

前記ソフトウェアアプリケーションの第１のアプリケーションの前記置換オブジェクトコードが前記共有の秘密にアクセスすることを可能にする前に、前記第１のアプリケーションの署名を前記セキュリティマネージャアプリケーションによって検証するステップをさらに含む、請求項２に記載の方法。

【請求項４】

前記署名は公開キーインフラストラクチャ（ＰＫＩ）キーのプロダクトである、請求項３に記載の方法。

【請求項５】

前記ソフトウェアアプリケーションの第１のアプリケーションからの登録要求を前記セキュリティマネージャアプリケーションで受取るステップと、

前記第１のアプリケーションが登録されると、前記セキュリティマネージャアプリケーションによってユーザにパスコードを促すステップと、

前記パスコードを用いて、前記共有の秘密であるキーを生成するステップと、

前記第１のアプリケーションに前記キーを提供するステップとをさらに含む、請求項１から４のいずれか１項に記載の方法。

【請求項６】

前記方法はさらに、

前記セキュリティマネージャアプリケーションによってアプリケーションデータ保護キーを生成するステップと、

前記セキュリティマネージャアプリケーションによって、対応するデータ保護ルートキーを検索するステップと、

前記セキュリティマネージャアプリケーションによって、前記対応するデータ保護ルートキーを用いて前記アプリケーションデータ保護キーを暗号化するステップとを含む、請求項１から５のいずれか１項に記載の方法。

【請求項７】

コンピュータによって実行されると、前記コンピュータに請求項１から６のいずれか１項に記載の動作を実行させるコンピュータプログラム。

【請求項８】

コンピュータプログラムであって、

各々がオリジナルのコンパイル後のオブジェクトコードを有するソフトウェアアプリケーションを単一の演算装置上に提供する動作と、

前記演算装置上にセキュリティマネージャアプリケーションを提供する動作と、

各ソフトウェアアプリケーションのための置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いて各ソフトウェアアプリケーションのオリジナルのオブジェクトコードを変更する動作とを、１つ以上の機械に実行させるための命令を示す情報を格納しており、前記置換オブジェクトコードは、各ソフトウェアアプリケーションを前記セキュリティマネージャアプリケーションと通信するように変更するとともに、前記ソフトウェアアプリケーション間の共有の秘密にアクセスすることにより、前記ソフトウェアアプリケーション間に信頼関係を確立するように構成され、

各ソフトウェアアプリケーションの前記置換オブジェクトコードによって、前記セキュリティマネージャアプリケーションから前記共有の秘密を取得する動作を、前記１つ以上の機械にさらに実行させるための命令を示す情報を格納している、コンピュータプログラム。

【請求項９】

コンピュータプログラムにおける命令を実行するコンピュータシステムであって、コン

ピュータプログラム命令は、
各々がオリジナルのコンパイル後のオブジェクトコードを有するソフトウェアアプリケーションを単一の演算装置上に提供するためのプログラムコードと、
前記演算装置上にセキュリティマネージャアプリケーションを提供するためのプログラムコードと、
各ソフトウェアアプリケーションのための置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いて各ソフトウェアアプリケーションのオリジナルのオブジェクトコードを変更するためのプログラムコードとを含み、前記置換オブジェクトコードは、各ソフトウェアアプリケーションを前記セキュリティマネージャアプリケーションと通信するように変更するとともに、前記ソフトウェアアプリケーション間の共有の秘密にアクセスすることにより、前記ソフトウェアアプリケーション間に信頼関係を確立するように構成され、
各ソフトウェアアプリケーションの前記置換オブジェクトコードによって、前記セキュリティマネージャアプリケーションから前記共有の秘密を取得するためのプログラムコードとを含む、コンピュータシステム。
【請求項１０】
変更前に、各々のソフトウェアアプリケーションの前記オリジナルのオブジェクトコードを署名ありの形式から署名なしの形式に変換するためのプログラムコードと、
変更後に、各々のソフトウェアアプリケーションの前記置換オブジェクトコードを署名なしの形式から署名ありの形式に変換するためのプログラムコードとをさらに含む、請求項９に記載のコンピュータシステム。
【請求項１１】
装置であって、
各々がオリジナルのコンパイル後のオブジェクトコードを有するソフトウェアアプリケーションを単一の演算装置上に提供するための手段と、
前記演算装置上にセキュリティマネージャアプリケーションを提供するための手段と、
各ソフトウェアアプリケーションのための置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いて各々のソフトウェアアプリケーションのオリジナルのオブジェクトコードを変更するための手段とを含み、前記置換オブジェクトコードは、各ソフトウェアアプリケーションを前記セキュリティマネージャアプリケーションと通信するように変更するとともに、前記ソフトウェアアプリケーション間の共有の秘密にアクセスすることにより、前記ソフトウェアアプリケーション間に信頼関係を確立するように構成され、
各ソフトウェアアプリケーションの前記置換オブジェクトコードによって、前記セキュリティマネージャアプリケーションから前記共有の秘密を取得する手段を含む、装置。
【請求項１２】
変更前に、各々のソフトウェアアプリケーションの前記オリジナルのオブジェクトコードを署名ありの形式から署名なしの形式に変換するための手段と、
変更後に、各々のソフトウェアアプリケーションの前記置換オブジェクトコードを署名なしの形式から署名ありの形式に変換するための手段とをさらに含む、請求項１１に記載の装置。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照
本願は、２０１３年３月１５日に提出された米国特許出願第１３／８４１，４９８号の一部継続出願である、２０１４年３月１４日に提出された米国特許出願第１４／２１３，２４４号の継続出願であり、その利益を主張するものであって、これら全体があらゆる目的で引用によりこの明細書中に援用される。
【背景技術】

【０００２】
背景
本開示は、概して、セキュリティサービスを提供するためのシステム、方法および機械読取り可能媒体に関する。より特定的には、この開示は、中でも、認証、承認、監査、シングルサインオン、セキュリティポリシーの実施、キーの管理および配布、セキュア通信、セキュアデータの記憶、およびセキュアデータの共有を含むセキュリティサービスをソフトウェアアプリケーションに提供するためのシステム、方法および機械読取り可能媒体に関する。
【発明の概要】
【課題を解決するための手段】
【０００３】
概要
セキュリティサービスを提供するためのシステム、方法および機械読取り可能媒体を開示する。本開示の特徴に従うと、システムはメモリおよびプロセッサを含む。メモリは、複数のアプリケーションデータを格納するために用いられてもよく、各々のアプリケーションデータは、ソフトウェアアプリケーションに関連付けられ、アプリケーションオブジェクトコードを含み得る。プロセッサは、セキュリティマネージャアプリケーションモジュールを含み得る。セキュリティマネージャアプリケーションモジュールは、オブジェクトコード変更モジュールによって変更されたアプリケーションオブジェクトコードからのセキュリティサービスについての要求に、ソフトウェアアプリケーションを介して応答し得る。変更されたアプリケーションオブジェクトコードにより、セキュリティマネージャアプリケーションモジュールに対するセキュリティサービスについての要求の送信が容易になる。
【０００４】
一実施形態においては、オブジェクトコード変更モジュールは、動的ライブラリまたは静的ライブラリの導入、ロードコマンドの追加、シンボルの置換、スウィズリング（swizzling）および挿入のうち少なくとも１つによってアプリケーションオブジェクトコードを変更するために用いられてもよい。別の実施形態においては、セキュリティマネージャアプリケーションモジュールは、認証トークン、認証キーおよびセキュア通信チャネルからなる群から選択されるセキュリティアーティファクトを生成し得る。セキュリティマネージャアプリケーションモジュールは、セキュリティサービスについての要求を受取ったことに応じて、ソフトウェアアプリケーションにセキュリティアーティファクトを送信し得る。オブジェクト変更モジュールによって受取られて変更されたアプリケーションオブジェクトコードは、署名なしの形式であってもよい。
【０００５】
本開示の特徴に従うと、アプリケーションオブジェクトコードは、オブジェクトコード変更モジュールによるアプリケーションオブジェクトコードの変更前に、オブジェクトコード署名変換モジュールによって署名ありの形式から署名なしの形式に変換され、オブジェクトコード変更モジュールによるアプリケーションオブジェクトコードの変更後に、署名なしの形式から署名ありの形式に変換され得る。一実施形態においては、オブジェクトコード変更モジュールは、アプリケーションオブジェクトコードの変更前に、アプリケーションオブジェクトコードを署名ありの形式から署名なしの形式に変換し、アプリケーションオブジェクトコードの変更後に、アプリケーションオブジェクトコードを署名なしの形式から署名ありの形式に変換してもよい。
【０００６】
一実施形態においては、アプリケーションオブジェクトコードの変更は、ソフトウェアアプリケーションによって利用されるプログラマティックインターフェイス、クラス、オブジェクトおよびファンクションのうちの少なくとも１つの変更を含んでもよい。アプリケーションオブジェクトコードの変更は、セキュリティポリシーに対するコンプライアンスを確実にするためにポリシーエンジンを導入することを含んでもよい。セキュリティポ

リシーは、データ漏洩防止ポリシーおよびアクセス制御ポリシーからなる群から選択されてもよい。

【０００７】

本開示の特徴に従うと、セキュリティマネージャアプリケーションモジュールは、セキュリティポリシーに対するソフトウェアアプリケーションのコンプライアンスを確実にするためのポリシーエンジンを含み得る。セキュリティマネージャアプリケーションモジュールは、セキュリティポリシーをソフトウェアアプリケーションに送信して実行してもよい。セキュリティポリシーは、ソフトウェアアプリケーションの一部、単一のソフトウェアアプリケーションおよび複数のソフトウェアアプリケーションのうち少なくとも１つに適用され得る。一実施形態においては、ポリシーエンジンは動的なポリシーエンジンであり、セキュリティポリシーは、実行コンテキスト、外部イベント、明確なポリシー再定義、ならびにグループおよびロールメンバーシップの変更からなる群から選択される少なくとも１つのファクタに基づく。別の実施形態においては、セキュリティポリシーはリモートポリシーサーバから検索される。さらに別の実施形態においては、第１のソフトウェアアプリケーションがリモートポリシーサーバからセキュリティポリシーを検索し、第２のソフトウェアアプリケーションが第１のソフトウェアアプリケーションからセキュリティポリシーを検索する。ポリシーエンジンの実行によって生じるデータは、セキュリティマネージャアプリケーションモジュールおよび／またはポリシーサーバに送信されてもよい。

【０００８】

本開示の特徴に従うと、ソフトウェアアプリケーションを介してセキュリティサービスについての要求に応答するセキュリティマネージャアプリケーションモジュールは、ソフトウェアアプリケーションが公開キーインフラストラクチャ（ＰＫＩ：public key infrastructure）キーで署名される場合、セキュリティサービスを促進し得る。セキュリティマネージャアプリケーションモジュールは、ソフトウェアアプリケーションおよび関連付けられたコンピューティングプラットフォームが損なわれていないことを確認し、ソフトウェアアプリケーションの署名を検証し得る。一実施形態においては、複数のアプリケーションデータは共通のキーを含む。別の実施形態においては、セキュリティマネージャアプリケーションモジュールは、アプリケーション信頼性を確立するためにソフトウェアアプリケーションからの登録要求に応答してもよい。セキュリティマネージャアプリケーションモジュールは、ユーザにアプリケーション登録パスコードを入力するよう促し、アプリケーション登録パスコードを用いてアプリケーションの信頼性を確立するようキーを生成して、セキュリティマネージャアプリケーションモジュールによってソフトウェアアプリケーションに対して提供されるセキュリティサービスを促進し得る。

【０００９】

別の実施形態においては、セキュリティマネージャアプリケーションモジュールは、ソフトウェアアプリケーションにキーを提供して、ソフトウェアアプリケーションが、セキュリティサービスについての要求をセキュリティマネージャアプリケーションモジュールに送信することを可能にし得る。アプリケーションデータは、ソフトウェアアプリケーションまたはセキュリティマネージャアプリケーションモジュールによって生成されるアプリケーションデータ保護キーを含み得る。アプリケーションデータ保護キーは、セキュリティマネージャアプリケーションモジュールによって維持される対応するデータ保護ルートキーによって暗号化および復号化されてもよい。一実施形態においては、アプリケーションデータ保護キーは、オブジェクトデータ保護キーの暗号化および復号化のうち少なくとも１つを実行して、第１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションへのデータオブジェクトの転送を容易にし得る。セキュリティマネージャアプリケーションモジュールは、第１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションへデータオブジェクトおよびオブジェクトデータ保護キーを転送する前に、オブジェクトデータ保護キーを暗号化するためのデータ共有キーを生成してもよい。データ共有キーはまた、データオブジェクトおよびオブジェクトデータ保護キーを第

１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションに転送した後に、オブジェクトデータ保護キーを復号化するために用いられてもよい。本開示の実施形態に従うと、セキュリティマネージャアプリケーションモジュールおよび第１のソフトウェアアプリケーションのうち少なくとも１つはまた、第１のソフトウェアアプリケーションのアプリケーションデータ保護キーでデータオブジェクトのためのオブジェクトデータ保護キーを復号化し、第２のソフトウェアアプリケーションのアプリケーションデータ保護キーでデータオブジェクトのためのオブジェクトデータ保護キーを暗号化し得る。

【００１０】

本開示の特徴に従うと、セキュリティサービスを提供するための、コンピュータによって実現される方法が提供される。当該方法は、複数のアプリケーションデータを有するメモリと通信するプロセッサ上で実現されてもよく、各々のアプリケーションデータは、ソフトウェアアプリケーションに関連付けられ、アプリケーションオブジェクトコードを含み得る。当該方法は、ソフトウェアアプリケーションに対応するアプリケーションオブジェクトコードをメモリから検索するステップと、ソフトウェアアプリケーションがセキュアサービスについての要求を送信することを可能にするためにアプリケーションオブジェクトコードの変更を受取るステップとを含み得る。この変更は、動的ライブラリまたは静的ライブラリの導入、ロードコマンドの追加、シンボルの置換、スウィズリングおよび挿入のうち少なくとも１つを含み得る。当該方法はさらに、セキュリティサービスについての要求をプロセッサを介して受取るステップと、プロセッサを介してセキュリティサービスを提供するステップとを含み得る。

【００１１】

一実施形態においては、アプリケーションオブジェクトコードは、変更前に署名ありの形式から署名なしの形式に変換され、変更後に署名なしの形式から署名ありの形式に変換される。アプリケーションオブジェクトコードの変更には、セキュリティポリシーに対するコンプライアンスを確実にするためにポリシーエンジンを導入することが含まれてもよい。セキュリティポリシーは、ソフトウェアアプリケーションの一部、単一のソフトウェアアプリケーション、および複数のソフトウェアアプリケーションのうち少なくとも１つに適用され得る。セキュリティポリシーは、データ漏洩防止ポリシーおよびアクセス制御ポリシーからなる群から選択されてもよい。一実施形態においては、ポリシーエンジンは、動的なポリシーエンジンであり、セキュリティポリシーは、実行コンテキスト、外部イベント、明確なポリシー再定義、ならびにグループおよびロールメンバーシップの変更からなる群から選択される少なくとも１つのファクタに基づく。

【００１２】

一実施形態においては、コンピュータによって実現される方法は、ソフトウェアアプリケーションおよび関連付けられたコンピューティングプラットフォームが損なわれていないことをプロセッサを介して確認するステップと、ソフトウェアアプリケーションのＰＫＩキー署名をプロセッサを介して検証するステップとを含む。また、コンピュータによって実現される方法は、セキュリティサービスについての要求を受取る前にアプリケーション信頼性を確立するためにソフトウェアアプリケーションから登録要求を受取るステップと、プロセッサを介してアプリケーション登録パスコードを入力するようユーザに促すステップとを含み得る。当該方法はさらに、アプリケーション登録パスコードを用いてアプリケーション信頼性を確立するためにプロセッサを介してキーを生成するステップと、ソフトウェアアプリケーションがセキュリティサービスについての要求を送信することを可能にするためにプロセッサを介してソフトウェアアプリケーションにキーを提供するステップとを含み得る。

【００１３】

本開示の特徴に従うと、コンピュータによって実現される方法は、プロセッサによってアプリケーションデータ保護キーを生成するステップと、対応するデータ保護ルートキーをメモリから検索するステップと、対応するデータ保護ルートキーを用いて、プロセッサを介してアプリケーションデータ保護キーを暗号化するステップとを含み得る。一実施形

態においては、コンピュータによって実現される方法は、データオブジェクトのためのオブジェクトデータ保護キーを、プロセッサを介して、第１のソフトウェアアプリケーションのアプリケーションデータ保護キーで復号化するステップと、オブジェクトデータ保護キーの暗号化および復号化のうち少なくとも１つのためにデータ共有キーをプロセッサを介して生成するステップと、データ共有キーを用いて、プロセッサを介して、オブジェクトデータ保護キーを暗号化するステップと、第１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションにデータオブジェクトおよび暗号化されたオブジェクトデータ保護キーを転送するステップと、プロセッサを介して、データ共有キーを用いてオブジェクトデータ保護キーを復号化するステップと、データオブジェクトのためのオブジェクトデータ保護キーを、プロセッサを介して、第２のソフトウェアアプリケーションのアプリケーションデータ保護キーで暗号化するステップとを含み得る。

【００１４】

本開示の特徴に従うと、機械読取り可能媒体が提供される。機械読取り可能媒体は、プロセッサおよびメモリを有する機械によって読取られたときに当該機械に本開示の方法に従った動作を実行させる命令を与え得る。

【００１５】

いくつかの実施形態は、演算装置上において中心のアプリケーションから別のアプリケーションにセキュリティサービスを提供するための方法に関する。当該方法は、演算装置上にセキュリティサービスを提供するように構成された第１のアプリケーションを提供するステップと、オリジナルのコンパイル後のオブジェクトコードを有する第２のアプリケーションを演算装置上に提供するステップと、第１のアプリケーションと通信するように構成された置換オブジェクトコード（replacement object code）を作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いて第２のアプリケーションのオリジナルのオブジェクトコードを変更するステップと、第２のアプリケーションにおける置換オブジェクトコードを呼出すステップと、第２のアプリケーションにおける置換オブジェクトコードを用いて第１のアプリケーションに対してセキュリティサービスを要求するステップとを含む。

【００１６】

オリジナルのオブジェクトコードは、動的ライブラリの導入、静的ライブラリの導入、付加的なロードコマンドの導入、シンボルの置換、ポインタスウィズリングおよび挿入からなる群から選択される少なくとも１つの技術によって変更することができる。当該方法は、認証トークン、キー、セキュリティクレデンシャル、シングルサインオントークン、識別子、セキュリティポリシー、セキュリティコマンド、セキュリティ構成、セッションハンドル、セッショントークン、およびセキュア通信チャネルからなる群から選択されるセキュリティアーティファクトを第１のアプリケーションによって生成するステップと、セキュリティサービスの要求に応答して第１のアプリケーションから第２のアプリケーションにセキュリティアーティファクトを送信するステップとを含み得る。オリジナルのオブジェクトコードは署名なしのコードを含んでもよい。当該方法はさらに、変更前にオリジナルのオブジェクトコードを署名ありの形式から署名なしの形式に変換するステップと、変更後に置換オブジェクトコードを署名なしの形式から署名ありの形式に変換するステップとを含んでもよい。演算装置はモバイルデバイスであり得る。

【００１７】

変更するステップは、プログラマティックインターフェイス、クラス、オブジェクトおよびファンクションのうち少なくとも１つを変更するステップを含み得る。第２のアプリケーションのオリジナルのオブジェクトコードを変更するステップは、セキュリティポリシーに対するコンプライアンスを確実にするためにポリシーエンジンを導入するステップを含み得る。セキュリティポリシーは、データ漏洩防止ポリシーおよびアクセス制御ポリシーからなる群から選択することができる。第１のアプリケーションは、セキュリティポリシーに対する第２のアプリケーションのコンプライアンスを確実にするためのポリシーエンジンを含み得る。第１のアプリケーションは、第２のアプリケーションにセキュリテ

ィポリシーを送信して実行させ得る。セキュリティポリシーは、第２のアプリケーションの一部、単一のソフトウェアアプリケーション、および複数のソフトウェアアプリケーションのうち少なくとも１つに適用することができる。セキュリティポリシーは、実行コンテキスト、外部イベント、明確なポリシー再定義、ならびにグループおよびロールメンバーシップの変更からなる群から選択される少なくとも１つのファクタに基づき得る。セキュリティポリシーは、演算装置から離れたサーバから検索される。

【００１８】

当該方法はさらに、第３のアプリケーションのオブジェクトコードを変更することによってポリシーエンジンを第２のアプリケーションから第３のアプリケーションに導入するステップを含み得る。置換オブジェクトコードの実行によって生じるデータは、第１のアプリケーションまたはリモートサーバに送信することができる。ソフトウェアアプリケーションが公開キーインフラストラクチャ（ＰＫＩ）キーで署名される場合、セキュリティサービスについての要求に応答する第１のアプリケーションは、セキュリティサービスを促進することができる。当該方法はさらに、第２のアプリケーションおよび演算装置が損なわれていないことを第１のアプリケーションによって確認するステップと、第１のアプリケーションの署名を検証するステップとを含み得る。セキュリティサービスは、第２のアプリケーションおよび第３のアプリケーションに共通のキーを提供することができる。第１のアプリケーションは、アプリケーション信頼性を確立するために第２のアプリケーションからの登録要求に応答し得る。第１のアプリケーションは、アプリケーション登録パスコードを入力するようユーザを促し、アプリケーション登録パスコードを用いてアプリケーション信頼性を確立するためのキーを生成して、第１のアプリケーションによって第２のアプリケーションに提供されるセキュリティサービスを促進し得る。第１のアプリケーションは、第２のアプリケーションがセキュリティサービスについての要求を送信することを可能にするために第２のアプリケーションにキーを提供することができる。

【００１９】

当該方法はさらに、アプリケーションデータ保護キーを生成するステップを含み得る。アプリケーションデータ保護キーは、第１のアプリケーションによって維持される対応するデータ保護ルートキーによって暗号化および復号化され得る。アプリケーションデータ保護キーは、オブジェクトデータ保護キーの暗号化および復号化のうち少なくとも１つを実行して、第１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションへのデータオブジェクトの転送を促進することができる。当該方法はさらに、データオブジェクトおよびオブジェクトデータ保護キーを第１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションに転送する前にオブジェクトデータ保護キーを暗号化するためのデータ共有キーを生成するステップと、データオブジェクトおよびオブジェクトデータ保護キーを第１のソフトウェアアプリケーションから第２のソフトウェアアプリケーションに転送した後にオブジェクトデータ保護キーを復号化するためのデータ共有キーを生成するステップとを含み得る。セキュリティマネージャアプリケーションモジュールおよび第１のソフトウェアアプリケーションのうち少なくとも１つは、第１のソフトウェアアプリケーションのアプリケーションデータ保護キーで、データオブジェクトのためのオブジェクトデータ保護キーを復号化することができ、第２のソフトウェアアプリケーションのアプリケーションデータ保護キーで、データオブジェクトのためのオブジェクトデータ保護キーを暗号化することができる。

【００２０】

いくつかの実施形態は、ソフトウェアアプリケーションのセキュリティポリシーを動的に更新するための方法に関する。当該方法は、オリジナルのコンパイル後のオブジェクトコードを有するアプリケーションを演算装置上に提供するステップと、セキュリティポリシーを実施するように構成された置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いてアプリケーションのオリジナルのオブジェクトコードを変更するステップと、第１のセキュリティポリシーを検索するステップと、置換オブジェクトコードを用いてセキュリティポリシーを実施するステップと

、第１のセキュリティポリシーを第２のセキュリティポリシーと置換するステップと、置換オブジェクトコードを用いて第２のセキュリティポリシーを実施するステップとを含み、第１および第２のセキュリティポリシーは、実行コンテキスト、外部イベント、明確なポリシー再定義、ならびにグループおよびロールメンバーシップの変更からなる群から選択される少なくとも１つのファクタに基づく。

【００２１】

第１または第２のセキュリティポリシーは、演算装置から離れたサーバから検索することができる。第１のアプリケーションは、リモートサーバから第１または第２のセキュリティポリシーを検索することができ、第２のアプリケーションは、第１のアプリケーションから第１または第２のセキュリティポリシーを検索する。置換オブジェクトコードの実行によって生じるデータはリモートサーバに送信することができる。演算装置はモバイルデバイスであり得る。

【００２２】

いくつかの実施形態は、演算装置上におけるアプリケーション間の信頼性を確立するための方法に関する。当該方法は、各々がオリジナルのコンパイル後のオブジェクトコードを有するソフトウェアアプリケーションを演算装置上に提供するステップと、各々のアプリケーションのための置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いて各々のアプリケーションのオリジナルのオブジェクトコードを変更するステップとを含み、置換オブジェクトコードは、アプリケーション間の共有の秘密にアクセスすることにより、アプリケーション間に信頼関係を確立するように構成される。

【００２３】

当該方法はさらに、変更前に各々のソフトウェアアプリケーションのオリジナルのオブジェクトコードを署名ありの形式から署名なしの形式に変換するステップと、変更後に各々のソフトウェアアプリケーションの置換オブジェクトコードを署名なしの形式から署名ありの形式に変換するステップとを含み得る。当該方法は、演算装置上にセキュリティマネージャアプリケーションを提供するステップをさらに含み得る。各々のアプリケーションのための置換オブジェクトコードは、セキュリティマネージャアプリケーションと通信するよう変更されて、各々のソフトウェアアプリケーションの置換オブジェクトコードによってセキュリティマネージャアプリケーションから共有の秘密を取得する。当該方法はさらに、ソフトウェアアプリケーションの第１のアプリケーションの置換オブジェクトコードが共有の秘密にアクセスすることを可能にする前に、当該第１のアプリケーションの署名をセキュリティマネージャアプリケーションによって検証するステップを含み得る。署名は公開キーインフラストラクチャ（ＰＫＩ）キーのプロダクトであってもよい。

【００２４】

当該方法はさらに、ソフトウェアアプリケーションの第１のアプリケーションからの登録要求をセキュリティマネージャアプリケーションで受取るステップと、第１のアプリケーションが登録されると、セキュリティマネージャアプリケーションによってユーザにパスコードを促すステップと、パスコードを用いて、共有の秘密であるキーを生成するステップと、第１のアプリケーションにキーを提供するステップとを含み得る。当該方法はさらに、セキュリティマネージャアプリケーションによってアプリケーションデータ保護キーを生成するステップと、セキュリティマネージャアプリケーションによって、対応するデータ保護ルートキーを検索するステップと、セキュリティマネージャアプリケーションによって、対応するデータ保護ルートキーを用いてアプリケーションデータ保護キーを暗号化するステップとを含み得る。

【００２５】

いくつかの実施形態は、演算装置上でデータオブジェクトをソースアプリケーションからデスティネーションアプリケーションにセキュアに転送するための方法に関する。当該方法は、演算装置上にソースアプリケーションおよびデスティネーションアプリケーションを提供するステップを含む。ソースアプリケーションおよびデスティネーションアプリ

ケーションは各々、オリジナルのコンパイル後のオブジェクトコードを有する。当該方法はさらに、ソースアプリケーションのための第１の置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いてソースアプリケーションのオリジナルのオブジェクトコードを変更するステップと、デスティネーションアプリケーションのための第２の置換オブジェクトコードを作成するように、メモリと動作可能に結合された少なくとも１つのプロセッサを用いてデスティネーションアプリケーションのオリジナルのオブジェクトコードを変更するステップと、オブジェクトデータ保護キーで暗号化されるデータオブジェクトをソースアプリケーションからデスティネーションアプリケーションに転送するステップと、ソースアプリケーションに関連付けられたソースアプリケーションキーをソースアプリケーションの第１の置換コードによって検索するステップと、ソースアプリケーションの第１の置換コードによって、検索されたソースアプリケーションキーでオブジェクトデータ保護キーを復号化するステップと、ソースアプリケーションの第１の置換コードによって、データ共有キーで、またはデスティネーションアプリケーションに関連付けられたデスティネーションアプリケーションキーで、オブジェクトデータ保護キーを暗号化するステップと、データ共有キーまたはデスティネーションアプリケーションキーで暗号化されたオブジェクトデータ保護キーをデスティネーションアプリケーションと共有するステップと、データ共有キーまたはデスティネーションアプリケーションキーで暗号化されたオブジェクトデータ保護キーをデスティネーションアプリケーションの第２の置換コードで復号化するステップと、暗号化されていない（unencrypted）オブジェクトデータ保護キーを用いて、オブジェクトデータをデスティネーションアプリケーションの第２の置換コードで復号化するステップとを含む。

【００２６】

当該方法は、データ共有キーまたはデスティネーションアプリケーションキーで暗号化されたオブジェクトデータ保護キーをデスティネーションアプリケーションに転送するステップを含み得る。検索するステップは、演算装置上のセキュリティマネージャアプリケーションを用い得る。当該方法はさらに、ソースアプリケーションとデスティネーションアプリケーションとの間のデータオブジェクトの転送が制限されているかどうかをセキュリティマネージャアプリケーションによって判断するステップと、当該判断に基づいて検索を促進するステップとを含み得る。当該方法はさらに、検索を促進するステップの前に、中心のセキュリティアプリケーションによって、ソースアプリケーションまたはデスティネーションアプリケーションの署名を検証するステップを含み得る。署名は公開キーインフラストラクチャ（ＰＫＩ）キーのプロダクトであってもよい。検索するステップは、暗号化されたソースアプリケーションキーを復号化するようにとの要求を含む暗号化されたソースアプリケーションキーをソースアプリケーションからセキュリティマネージャアプリケーションに送信するステップと、セキュリティマネージャアプリケーションによって、データ保護ルートキーを用いてソースアプリケーションキーを復号化するステップと、さらに、ソースアプリケーションキーをセキュリティマネージャアプリケーションからソースアプリケーションに転送するステップとを含み得る。検索するステップは、ソースアプリケーションキーについての要求をソースアプリケーションからセキュリティマネージャアプリケーションに送信するステップと、セキュリティマネージャアプリケーションによって、データ保護ルートキーを用いてソースアプリケーションキーを復号化するステップと、さらに、ソースアプリケーションキーをセキュリティマネージャアプリケーションからソースアプリケーションに転送するステップとを含み得る。検索するステップは、ソースアプリケーションからセキュリティマネージャアプリケーションに対してデータ保護ルートキーを要求するステップと、ソースアプリケーションによってデータ保護ルートキーを受取るステップと、ソースアプリケーションによって、データ保護ルートキーを用いてソースアプリケーションキーを復号化するステップとを含み得る。

【００２７】

当該方法は、セキュリティマネージャアプリケーションによって、ソースアプリケーションキー、デスティネーションアプリケーションキーおよびデータ共有キーのうちの少な

くとも１つを生成するステップをさらに含み得る。当該方法はさらに、セキュリティマネージャアプリケーションにソースアプリケーションが登録されるとソースアプリケーションキーを生成するステップ、または、セキュリティマネージャアプリケーションにデスティネーションアプリケーションが登録されるとデスティネーションアプリケーションキーを生成するステップを含み得る。当該方法はさらに、ユーザにパスコードを促すステップと、パスコードを用いてソースアプリケーションキーまたはデスティネーションアプリケーションキーを生成するステップを含み得る。当該方法はさらに、ソースアプリケーションによって要求されると、セキュリティマネージャアプリケーションによってデータ共有キーを生成するステップを含み得る。当該方法はさらに、デスティネーションアプリケーションの第２の置換コードによって、デスティネーションアプリケーションキーでオブジェクトデータ保護キーを暗号化するステップを含み得る。演算装置はモバイルデバイスであってもよい。

【００２８】

本開示の特徴に従うと、機械読取り可能媒体が提供される。機械読取り可能媒体は、プロセッサおよびメモリを有する機械によって読取られると、当該機械に本開示の方法に従った動作を実行させる命令を与え得る。

【００２９】

実施形態は、コンピュータによって実行されると、当該コンピュータに上記方法を実行させるコンピュータソフトウェアを含み得る。

【００３０】

図面の簡単な説明

本開示の上述の特徴および目的は、添付の図面に関連付けて読まれる以下の説明に関連付けるとより明らかになるだろう。添付の図面においては、同様の参照番号は同様の要素を示す。

【図面の簡単な説明】

【００３１】

【図１】本開示の実施形態に従った、セキュリティサービスを提供するためのシステムを示すブロック図である。

【図２】本開示の実施形態に従った、集中型セキュリティサービスを同じプラットフォーム上に存在する他のソフトウェアアプリケーションに提供するための集中型セキュリティマネージャアプリケーションモジュールを示すブロック図である。

【図３】本開示の実施形態に従ったオブジェクトコード変更を示す例示的なブロック図である。

【図４】本開示の実施形態に従った、セキュリティアーティファクトを他のソフトウェアアプリケーションに送信するセキュリティマネージャアプリケーションモジュールを示す例示的なブロック図である。

【図５】本開示の実施形態に従った、既存のアプリケーションオブジェクトコードを変更することによってソフトウェアアプリケーションの挙動を変更する方法を示す例示的なブロック図である。

【図６】本開示の実施形態に従った、オブジェクトコードにポリシーエンジンを投入することによってソフトウェアアプリケーションの挙動を変更するための方法を示す例示的なブロック図である。

【図７】本開示の実施形態に従った、オブジェクトコードにポリシーエンジンを投入することによってソフトウェアアプリケーションの挙動を変更するための方法を示す例示的なブロック図である。

【図８】本開示の実施形態に従った、オブジェクトコードにポリシーエンジンを投入することによってソフトウェアアプリケーションの挙動を変更するための方法を示す例示的なブロック図である。

【図９】本開示の実施形態に従った、オブジェクトコードにポリシーエンジンを投入することによってソフトウェアアプリケーションの挙動を変更するための方法を示す例示的な

ブロック図である。
【図１０】本開示の実施形態に従った、オブジェクトコードにポリシーエンジンを投入することによってソフトウェアアプリケーションの挙動を変更するための方法を示す例示的なブロック図である。
【図１１】本開示の実施形態に従った、オブジェクトコードにポリシーエンジンを投入することによってソフトウェアアプリケーションの挙動を変更するための方法を示す例示的なブロック図である。
【図１２】本開示の実施形態に従った、セキュリティマネージャアプリケーションモジュールを用いてソフトウェアアプリケーション間の信頼性を確立する方法を示す例示的なブロック図である。
【図１３】本開示の実施形態に従った、共通のキーを有するソフトウェアアプリケーションと通信するセキュリティマネージャアプリケーションモジュールを示す例示的なブロック図である。
【図１４】本開示の実施形態に従った、セキュリティマネージャアプリケーションモジュールに対する信頼性を確立するためのソフトウェアアプリケーションのパスコード登録を示す例示的なブロック図である。
【図１５】本開示の実施形態に従った、ソフトウェアアプリケーション内でセキュリティを維持するためのアプリケーションデータ保護キーの使用を示す例示的なブロック図である。
【図１６】本開示の実施形態に従った、データオブジェクトをソース（第１の）アプリケーションからデスティネーション（第２の）アプリケーションにセキュアに転送するための方法を示す例示的なブロック図である。
【図１７】本開示の実施形態に従った、データオブジェクトをソース（第１の）アプリケーションからデスティネーション（第２の）アプリケーションにセキュアに転送するための別の方法を示す例示的なブロック図である。
【発明を実施するための形態】
【００３２】
詳細な説明
以下の詳細な説明は、多数の特徴および教示を別個にかつ組み合わせて利用する代表的な例を含み、添付の図面を参照しつつより詳細に多数の実施形態を記載する。この詳細な説明は、単に、本教示の好ましい局面を実施するためのさらなる詳細を当業者に教示するよう意図されたものに過ぎず、添付の特許請求の範囲を限定するよう意図されたものではない。したがって、以下の詳細な説明において開示される特徴の組み合わせは、教示を最も広い意味で実施するのに必ずしも必要ではない可能性があり、代わりに、単に本教示の特に代表的な例を記載するために教示されているに過ぎない。
【００３３】
以下の詳細な説明のいくつかの部分は、コンピュータメモリ内で実行される動作のアルゴリズムおよびシーケンスに関して提示されている。これらのアルゴリズムに関する説明および表現は、データ処理技術に精通した人々によって用いられる手段であって、彼らの研究成果の本質を最も有効に他の当業者に伝えるためのものである。動作のアルゴリズムまたはシーケンスは、ここでは、そして一般的には、所望の結果に通じる矛盾のないステップのシーケンスであると考えられている。これらのステップは物理量の物理的な操作を必要とするステップである。通常、必ずしも必要ではないが、これらの量は、記憶、転送、組み合わせ、比較、および他の場合には操作、が可能な電気信号または磁気信号の形をとる。
【００３４】
しかしながら、これらおよび同様の語のすべてが適切な物理量に関連付けられるべきであって、これらの量に適用される単に都合のよい標識に過ぎないことが留意されるべきである。特に規定のない限り、以下の説明から明らかになるように、記載全体を通じて、「処理する」または「演算する」または「計算する」または「判断する」または「表示する

」などの語を用いた説明は、コンピュータシステムのレジスタおよびメモリ内の物理（電子）量として表わされるデータを処理して、電子装置のメモリもしくはレジスタまたは他のこのような情報記憶装置、送信装置もしくは表示装置内の物理量として同様に表わされる他のデータに変換するコンピュータシステムまたは同様の電子装置の動作およびプロセスを指している。

【００３５】

この明細書中に記載される方法は、本質的に如何なる特定の電子装置または他の装置にも関連していない。さまざまな汎用のシステムは、この明細書中の教示に従ってプログラムと共に用いられてもよいか、または、必要な方法ステップを実行するためにより特化された装置を構成するのに好都合であることを証明し得る。これらのさまざまなシステムのために必要な構造が以下の記載から明らかになるだろう。この明細書中に記載されるように実施形態の教示を実現するためにさまざまなプログラミング言語が用いられ得ることが認識されるだろう。

【００３６】

この特許文献は、モバイルデバイスを含む装置上のソフトウェアアプリケーションにセキュリティサービスを提供するための固有のシステム、方法および機械読取り可能媒体を記載する。これらのセキュリティサービスは、中でも、認証、承認、監査、シングルサインオン、セキュリティポリシーの実施、キーの管理および配布、セキュア通信、セキュアデータの記憶、およびセキュアデータの共有を含み得る。この目的のために、多数のプロトコルおよび基準が、この明細書中に記載される実施形態と組み合わせて説明および使用される。この明細書中に記載される実施形態は如何なるプロトコルまたは基準とも組み合わせて用いられてもよいが、以下のプロトコルおよび基準はそれら全体が引用によってこの明細書中に援用されている：ＩＥＴＦ　ＲＦＣ　２６３１（Diffie-Hellman）；ＩＥＥＥ　１３６３；ＩＥＴＦ　ＲＦＣ　３２８０（Ｘ．５０９公開キーインフラストラクチャ）；ＩＥＴＦ　ＲＦＣ　４１２０（Ｋｅｒｂｅｒｏｓ　Ｖ５）；ＩＥＴＦ　ＲＦＣ　４１７８（ＳＰＮＥＧＯ）；ＩＥＴＦ　ＲＦＣ　２６１６（ＨＴＴＰ　１．１）；ＩＥＴＦ　ＲＦＣ　４５５９；ＩＥＴＦ　ＲＦＣ　４５５６（ＫｅｒｂｅｒｏｓのためのＰＫＩＮＩＴ）；ＩＥＴＦ　ＲＦＣ　６１０１／２２４６／５２４６（ＳＳＬ／ＴＬＳ）；ＳＡＭＬ　Ｖ１．０／１．１／２．０；ＯｐｅｎＩＤ；Ｏａｕｔｈ；ＷＳ－Ｆｅｄｅｒａｔｉｏｎ；およびＯＡＴＨ　ＨＯＴＰ／ＴＯＴＰ／ＯＣＲＡ。

【００３７】

コンピュータアプリケーションのオブジェクトコードを変更することによるコンピュータアプリケーションのためのセキュリティサービス管理

当該技術においては、演算装置が一旦マルウェアに感染すると、コンピュータ上で実行されるほとんど如何なるアプリケーションも、その永続メモリをピーキングするかまたは永続メモリと他のアプリケーションとの間の通信を阻止することによってスヌープ（snooped）することができるために問題が生じる。いくつかのアプリケーションは、それらが送信したり、ディスクに保存したりするものなどすべてを暗号化することによってこれを最小限にするようにプログラムされているが、これにより、それらアプリケーションがファイルを伝達または共有するのに用いる他のすべてのアプリケーションだけでなく、それらの元来プログラムされていたソースコードに上述の特徴が必要となる。いくつかの実施形態は、暗号化および解読の追加、セキュアでないネットワークに対する呼出しの削除、セキュリティマネージャアプリケーションに対する暗号キーの要求などを行うために、演算装置にインストールされたアプリケーションのオリジナルのコンパイル後のオブジェクトコードを変更するステップを含む。たとえば、クリアテキストに電子メールを保存する電子メールクライアントアプリケーションの．ｄｙｌｉｂファイルは、暗号化を用いて電子メールを保存する別の．ｄｙｌｉｂファイルと切換えることができる。別の例においては、ビジネスインテリジェンスアプリケーションの記号表は、企業のファイアウォール外の電子メールメッセージを阻止するオリジナルの別のオブジェクトファィルとは異なるオブジェクトファィルを呼出すように変更することができる。さらに別の例においては、ポ

リシーエンジンは、新しいオブジェクトコードにコピーし、更新可能なポリシー上で実行することができる。

【００３８】

一実施形態においては、モバイルデバイスのための集中型セキュリティサービスアーキテクチャがセキュリティマネージャアプリケーションを用いて提供される。本開示のある１つの特徴は、集中型セキュリティマネージャアプリケーションおよび統合した他のアプリケーションと、コンパイル後のオブジェクトコード変更によりセキュリティマネージャアプリケーションによって提供されるサービスとの組み合わせを含む。

【００３９】

図１は、本開示の実施形態に従った、セキュリティサービスを提供するためのシステム１００のブロック図を示す。システム１００は、ネットワーク１０６を介して遠隔装置１０４にアクセスすることができる演算装置１０２を含み得る。

【００４０】

一実施形態においては、演算装置１０２はメモリ１０８およびプロセッサ１１０を含み得る。メモリ１０８は複数のアプリケーションデータを格納するのに用いられてもよく、各々のアプリケーションデータは、ソフトウェアアプリケーションに関連付けられ、アプリケーションオブジェクトコードを含み得る。

【００４１】

認識され得るように、メモリ１０８は、たとえば、エンドユーザの動作に応答してデータを格納および／または検索するために用いられてもよい。周知のように、メモリは、分割または相互参照され得るデータベースカテゴリを含み得るものであって、データベースなどの如何なる組み合わせもサーバ内から提供することができる。一実施形態においては、データベースの如何なる部分も、ネットワーク１０６を介して離れたところにまで提供することができる。外部データベースからの外部データは、デバイス１０２が理解し得るものであれば如何なる標準フォーマットでも提供することができる。たとえば、プロバイダにおける外部データベースは、たとえば名前、ユーザＩＤおよびコンピュータ同定数などのエンドユーザデータを、標準フォーマットで、サーバからの要求に応答して有利に提供することができ、エンドユーザデータブロックは、コードモジュールが理解し得る関数呼出しフォーマットに変換される。

【００４２】

認識され得るように、メモリ１０８は、機械読取り可能媒体などの記憶装置であってもよく、情報をプロセッサによって読取り可能な形式で提供（すなわち、格納および／または送信）する如何なるメカニズムであってもよい。たとえば、機械読取り可能媒体は、読取り専用メモリ（read only memory：ＲＯＭ）、ランダムアクセスメモリ（random access memory：ＲＡＭ）、キャッシュ、ハードディスクドライブ、フロッピー（登録商標）ディスクドライブ、磁気ディスク記憶媒体、光学記憶媒体、フラッシュメモリ素子、または、情報を格納することができる他の如何なるデバイスであってもよい。付加的には、機械読取り可能媒体はまた、コンピュータ記憶媒体および通信媒体を含んでもよい。機械読取り可能媒体は、コンピュータ読取り可能命令、データ構造、プログラムモジュールまたは他のデータなどの情報を格納するための如何なる方法または技術において実現される揮発性媒体および不揮発性媒体、取外し可能媒体および取外し不可能媒体を含む。機械読取り可能媒体はまた、ＲＡＭ、ＲＯＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリまたは他のソリッドステートメモリ技術、ＣＤ－ＲＯＭ、ＤＶＤまたは他の光学記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、または、所望の情報を格納するのに用いることができ、コンピュータによってアクセスすることができる他の如何なる媒体をも含むが、これらに限定されない。

【００４３】

演算装置１０２はまた、セキュリティマネージャアプリケーションモジュール１１２を含むプロセッサ１１０に電気的および／または物理的に結合された１つ以上のファンクションモジュールを含んでもよい。この明細書中において用いられる場合、モジュールとい

う語は、ハードウェアおよび／もしくはファームウェアにおいて具体化されるロジックを指すか、または、たとえばＣ＋＋などのプログラミング言語で書込まれ、エントリポイントおよびエグジットポイントを有する可能性のあるソフトウェア命令の集まりを指す。ソフトウェアモジュールは、コンパイルされて実行可能プログラムにリンクされ得るか、ダイナミックリンクライブラリにインストールされ得るか、または、ＢＡＳＩＣなどの翻訳言語で書込まれ得る。ソフトウェアモジュールが他のモジュールから呼出し可能であり得ること、および／または、検出されたイベントもしくは割込みに応答して呼出され得ることが認識されるだろう。ソフトウェア命令はＥＰＲＯＭなどのファームウェアに埋込まれてもよい。ハードウェアモジュールがゲートおよびフリップフロップなどの接続された論理ユニットで構成され得ること、ならびに／または、プログラマブルゲートアレイなどのプログラム可能なユニットで構成され得ることがさらに認識されるだろう。この明細書中に記載されるモジュールは、好ましくは、ソフトウェアモジュールとして実現されるが、ハードウェアおよび／またはファームウェアで表わされてもよい。

【００４４】

一実施形態においては、各々のモジュールはモジュラーコードオブジェクトとして提供される。この場合、コードオブジェクトは、典型的には、一組の標準化された関数呼出しを通じてインタラクトする（interact）。一実施形態においては、コードオブジェクトはＣ＋＋などの好適なソフトウェア言語で書込まれているが、コードオブジェクトは如何なる低レベル言語または高レベル言語でも書込むことができる。一実施形態においては、コードモジュールは、Ｃ＋＋で実現され、Ｗｉｎｄｏｗｓ（登録商標）プラットフォーム、ｉＯＳプラットフォーム、Ａｎｄｒｏｉｄ（登録商標）プラットフォームなどの上で実行されるコンピュータ上でコンパイルされる。当業者であれば、ハードウェアに対して直接的なコード実現例を含む実現例がいくつでも実現可能であることを認識するだろう。

【００４５】

セキュリティマネージャアプリケーションモジュール１１２は、オブジェクトコード変更モジュール１１４および／またはオブジェクトコード署名変換モジュール１１６によって変更されたアプリケーションオブジェクトコードに動作可能に結合され得る。セキュリティマネージャアプリケーションモジュール１１２は、セキュリティサービス、たとえばネットワーク１０６上でのセキュア通信、についての要求にソフトウェアアプリケーションを介して応答し得る。オブジェクトコード変更モジュール１１４および／またはオブジェクトコード署名変換モジュール１１６は、セキュリティマネージャアプリケーションモジュール１１２に対するセキュリティサービスについての要求の送信を容易にするようにアプリケーションオブジェクトコードを変更し得る。各々のアプリケーションデータのための変更済みアプリケーションオブジェクトコードは、メモリ１０８に格納されてもよい。

【００４６】

図２は、本開示の実施形態に従った、集中型セキュリティサービス１１７を同じプラットフォーム上に存在する他のソフトウェアアプリケーション１１８～１２２に提供するための集中型セキュリティマネージャアプリケーションモジュール１１２のブロック図を示す。認識され得るように、他のソフトウェアアプリケーション１１８～１２２は、他のソフトウェアアプリケーション１１８～１２２におけるソースコードを変更することによってではなく、既存のアプリケーションオブジェクトコード１２４～１２８を変更することによって集中型セキュリティサービス１１７にリンクされてもよい。一実施形態においては、オブジェクトコード変更１２４～１２８は、他の公知のオブジェクトコード変更技術の中でも、動的ライブラリまたは静的ライブラリの投入、ロードコマンドの追加、シンボルの置換、スウィズリングおよび挿入を含み得る。当業者であれば、変更という語が追加、置換および／または削除を含み得ることを認識するだろう。図３は、本開示の実施形態に従った、オブジェクトコード変更の例示的なブロック図を示す。

【００４７】

一実施形態においては、認証トークン、キー、クレデンシャル、シングルサインオント

ークン、識別子、セキュリティポリシー、セキュリティコマンド、セキュリティ構成、セッションハンドル、セッショントークンおよびセキュア通信チャネルなどのセキュリティアーティファクトは、セキュリティマネージャアプリケーションモジュール１１２によって生成され、他のソフトウェアアプリケーション１１８～１２２に対して要求に応じて分配されてもよい。図４は、本開示の実施形態に従った、他のソフトウェアアプリケーション１１８～１２２にセキュリティアーティファクト１３０～１３４を送信するセキュリティマネージャアプリケーションモジュール１１２を示す例示的なブロック図である。

【００４８】

認識され得るように、セキュリティマネージャアプリケーションモジュール１１２は、特にその目的のために構築された特定のソフトウェアアプリケーション（たとえば、セキュアコンテナアプリケーション）であってもよい。別の実施形態においては、セキュリティマネージャアプリケーションモジュール１１２は、変更されたモバイルアプリケーションであってもよく、たとえば、所与のデバイスにインストールされるかまたは所与のデバイス上で起動される第１のアプリケーションであってもよい。この場合、セキュリティマネージャ機能は上述されるオブジェクトコード変更の一部をなしている。一実施形態においては、特殊用途のセキュリティマネージャアプリケーションモジュールを備えたシステムは多数のアプリケーションにわたるコードおよびファンクションの重複を最小限にすることが好ましい可能性があるが、これは不要である。

【００４９】

当業者であれば、オブジェクトコードを変更するためのいくつかの方法があることを認識するだろう。一実施形態においては、オブジェクトコード変更１２４～１２８の段階が、ソフトウェアアプリケーション１１８～１２２の実行時に動的に実行されてもよい。これにより、オブジェクトコード変更１２４～１２８および結果として生じるアプリケーション挙動変更を、その実行時の特定の情況においてその時点で利用可能となるデータに基づいて決定することが可能となり得る。

【００５０】

図５は、本開示の実施形態に従った、既存のアプリケーションオブジェクトコードを変更することによってソフトウェアアプリケーション１１８～１２２の挙動を変更する方法を示す例示的なブロック図である。変更プロセスは２段階で実行されてもよい。ソフトウェアアプリケーション１１８～１２２がそれが実施されるであろうプラットフォームにインストールされる前に、オブジェクトコード変更プロセスの第１段階が実行されてもよく、結果として、中間の変更済みアプリケーションオブジェクトコード１３０が得られることとなる。ソフトウェアアプリケーション１１８～１２２がそれが実施されるであろうプラットフォームにインストールされた後に、オブジェクトコード変更プロセスの第２段階が実行されてもよく、結果として、最終的な変更済みアプリケーションオブジェクトコード１３１が得られることとなる。

【００５１】

一実施形態においては、オブジェクトコード変更プロセスの第２段階は、中間の変更済みアプリケーションオブジェクトコード１３０によってそれ自体で実行されてもよい。別の実施形態においては、オブジェクトコード変更プロセスの第１段階および第２段階は、構成および／またはポリシーに基づいて別のやり方で実行されてもよい。認識され得るように、オブジェクトコード変更プロセスの第２段階は、アプリケーションの実行が開始されるたびに違ったやり方で実行されてもよく、ならびに／または、実行コンテキスト、外部イベント、明確なポリシー再定義、さまざまなカウンタ、およびモバイルデバイスを所有するユーザのグループおよびロールメンバーシップの変更を含むがこれらに限定されないさまざまなファクタに基づいて実行する間に違ったやり方で実行されてもよい。別の実施形態においては、オブジェクトコード変更プロセスの第２段階では、外部ソースから何らかの新しいオブジェクトコードをロードしてもよい。

【００５２】

本開示の実施形態に従うと、オブジェクトコード変更１２４～１２６は、既存のアプリ

ケーション機能を停止させないようにソフトウェアアプリケーション１１８～１２２の挙動を変更するために用いられてもよい。これは、オリジナルの未変更のアプリケーションコードがプラットフォームまたはオペレーティングシステムによって実行される前に、変更済みのオブジェクトコードをプラットフォームまたはオペレーティングシステムによって実行させることによって達成されてもよい。

【００５３】

一実施形態においては、（アプリケーションソースコードを変更することによってではなく）既存のアプリケーションオブジェクトコードを変更することによってソフトウェアアプリケーション１１８～１２２の挙動を分析および変更する方法が提供される。当該方法は、未変更のアプリケーションオブジェクトコードおよびいずれかの関連付けられた構成情報を分析して、アプリケーション実行プロファイルを抽出するために実行するよう意図されているプラットフォームまたはオペレーティングシステムによって如何に実行されるかを判断するステップと、もはやプラットフォームまたはオペレーティングシステムによって直接用いられることのないように未変更のアプリケーションオブジェクトコードおよびいずれかの関連付けられた構成情報を変更するステップと、アプリケーション実行プロファイルを用いて、未変更のアプリケーションが、任意には新しい関連付けられた構成情報を含む新しいオブジェクトコードでプラットフォームまたはオペレーティングシステムによって如何に実行されるかを再現するステップと、新しいオブジェクトコードを未変更のアプリケーションオブジェクトコードと組み合わせて、結果として変更済みのアプリケーションオブジェクトコード１２４～１２８を得るステップと含み得る。

【００５４】

一実施形態においては、変更済みのオブジェクトコード１２４～１２８は、ソフトウェアアプリケーション１１８～１２２に投入される動的ライブラリを含み得る。認識され得るように、動的ライブラリを参照する新しいロードコマンドが、未変更のアプリケーションオブジェクトコードに存在する既存のロードコマンドのリストに追加されてもよい。

【００５５】

認識され得るように、オブジェクトコード変更プロセスに対する入力は、署名なしの形式の未変更のモバイルアプリケーションオブジェクトコードであってもよく、オブジェクトコード変更プロセスからの出力は、署名なしの形式または署名ありの形式の変更済みのモバイルオブジェクトコードであってもよい。一実施形態においては、未変更のモバイルアプリケーションオブジェクトコードの署名ありの形式を未変更のモバイルアプリケーションオブジェクトコードの署名なしの形式に変換するプロセスは、オブジェクトコード変更プロセスの前に実行されてもよい。別の実施形態においては、変更済みのモバイルアプリケーションオブジェクトコードの署名なしの形式を変更済みのモバイルアプリケーションオブジェクトコードの署名ありの形式に変換するプロセスは、オブジェクトコード変更プロセスの後に実行されてもよい。さらに別の実施形態においては、中間の変更済みのモバイルアプリケーションオブジェクトコードの署名なしの形式を中間の変更済みのモバイルアプリケーションオブジェクトコードの署名ありの形式に変換するプロセスは、これまでに説明した２段階のオブジェクトコード変更プロセスの第１段階の後に実行されてもよい。

【００５６】

一実施形態においては、図１に示されるオブジェクトコード署名変換モジュール１１６は、オブジェクトコード変更モジュール１１４によるアプリケーションオブジェクトコードの変更前にアプリケーションオブジェクトコードを署名ありの形式から署名なしの形式に変換するために用いられてもよく、オブジェクトコード変更モジュール１１４によるアプリケーションオブジェクトコードの変更後にアプリケーションオブジェクトコードを署名なしの形式から署名ありの形式に変換するために用いられてもよい。

【００５７】

認識され得るように、未変更のアプリケーションオブジェクトコードを署名ありの形式から署名なしの形式に変換すること、および／または、変更済みのアプリケーションオブ

ジェクトコードを署名なしの形式から署名ありの形式に変換することは、オブジェクトコード変更プロセスの一環として実行されてもよい。変更済みのオブジェクトコード１２４～１２８は、アプリケーション自体内におけるおよび／またはアプリケーションが使用する既存のプログラマティックインターフェイス、クラス、オブジェクトおよび／またはファンクションの挙動に対する変更を含み得る。プログラマティックインターフェイス、クラス、オブジェクトおよび／またはファンクションはモバイルデバイスプラットフォームによって提供されてもよい。

【００５８】

一実施形態においては、当該プロセスにより、結果として、既存のプログラマティックインターフェイス、クラス、オブジェクトもしくはファンクションがブロックされ、取除かれ、代替的な実現例と置換される可能性および／または部分的もしくは全体的に変更される可能性がある。別の実施形態においては、当該プロセスにより、結果として、既存のプログラマティックインターフェイス、クラス、オブジェクトもしくはファンクションの使用前および／または使用後に新しいプログラム機能が実行される可能性がある。さらに別の実施形態においては、たとえ既存のプログラマティックインターフェイス、クラス、オブジェクトまたはファンクションがオブジェクトコードに依然として存在していたとしても、当該プロセスにより、結果として、既存のプログラマティックインターフェイス、クラス、オブジェクトまたはファンクションの代わりに新しいプログラム機能が実行される可能性がある。

【００５９】

認識され得るように、オブジェクトコード変更１２４～１２８はモジュールで構成されてもよく、この場合、各々のモジュールがオブジェクトコード変更１２４～１２８の一部を実現し、オブジェクトコード変更プロセス中にアプリケーションに適用されるモジュールの組が構成および／またはポリシーによって制御されてもよい。オブジェクトコード変更プロセス中にソフトウェアアプリケーション１１８～１２２に適用されるモジュールの組は、先に説明した２段階のオブジェクトコード変更プロセスの第１段階および／または第２段階中に決定されてもよい。オブジェクトコード変更プロセス中にソフトウェアアプリケーション１１８～１２２に適用されるモジュールの組はまた、アプリケーションによって配送される構成ファイルによって決定されてもよい。

【００６０】

本開示の実施形態に従うと、（アプリケーションソースコードを変更することによってではなく）既存のアプリケーションオブジェクトコードを変更することによってソフトウェアアプリケーション１１８～１２２の記憶挙動を変更する方法が提供される。当該方法は、未変更のアプリケーションがデータの記憶を直接要求する既存のプログラマティックインターフェイスまたはファンクションを、既存の挿入またはスウィズリング技術を用いて、新しいインターフェイスまたはファンクションと置換するステップを含み得る。新しいプログラマティックインターフェイスまたはファンクションは、データをその書込み時に暗号化するのに用いられてもよく、および／または、データをその読取り時に復号化するのに用いられてもよい。新しいプログラマティックインターフェイスまたはファンクションはまた、既存のプログラマティックインターフェイスまたはファンクションを呼出してもよい。認識され得るように、既存のプログラマティックインターフェイスまたはファンクションはモバイルデバイスプラットフォームによって提供され得る。既存のプログラマティックインターフェイスまたはファンクションはＰＯＳＩＸ　Ｉ／Ｏ　ＡＰＩの一部であってもよい。

【００６１】

本開示の実施形態に従うと、既存のアプリケーションオブジェクトコードを変更することによってソフトウェアアプリケーション１１８～１２２の通信挙動を変更する方法が提供される。当該方法は、変更済みのアプリケーションオブジェクトコードを用いてソフトウェアアプリケーション１１８～１２２からの通信要求を停止させるステップを含み得る。変更済みのアプリケーションオブジェクトコードは、必要なセキュリティアーティファ

クトが通信要求に存在しているかどうかをチェックするために用いられてもよい。一実施形態においては、必要なセキュリティアーティファクトが通信要求に存在していない場合、変更済みのアプリケーションオブジェクトコードが必要なセキュリティアーティファクトを検索してもよい。必要なセキュリティアーティファクトを検索した後、変更済みのアプリケーションオブジェクトコードがこれらセキュリティアーティファクトを通信要求に追加し、通信要求の継続を可能にし得る。認識され得るように、通信要求はネットワーク通信要求であってもよい。

【００６２】

セキュリティアーティファクトは、認証トークン、キー、クレデンシャル、シングルサインオントークン、識別子、セキュリティポリシー、セキュリティコマンド、セキュリティ構成、セッションハンドル、セッショントークンまたはセキュア通信チャネルであってもよい。セキュリティアーティファクトは、サーバおよび／またはセキュリティマネージャアプリケーションモジュール１１２から検索されてもよい。

【００６３】

コンピュータアプリケーションセキュリティポリシーのオブジェクトコードを変更することによるコンピュータアプリケーションセキュリティポリシーの変更

当該技術においては、コンピュータアプリケーションがコンピュータにインストールされると、そのセキュリティ手順またはルールが典型的には変更できなくなるため問題となる。いくつかの実施形態は、セキュリティポリシーを更新することができるようにアプリケーションのオブジェクトコードを変更するステップを含む。アプリケーションのオブジェクトコードは、追加、削除、置換、編集、または変更が可能である。たとえば、モバイルデバイスが世界中を移動して回っていると判断された場合、リモートサーバからセキュリティポリシーテキストファイルを検索するためのステップを追加するものと置換されたデータの保存および検索に関するそのオリジナルの．ｄｙｌｉｂ（または．ｓｏまたは．ｄｌｌ）ファイルを有してデバイス上で実行される電子メールクライアントが、リモートサーバから新しいポリシーをダウンロードすることができる。新しいポリシーは、メッセージ通信なし、チャットなしまたは印刷なしなどの海外旅行のための新しいルールを導入することができる。別の例として、新しいポリシーは、ビジネスインテリジェンスアプリケーションによって保存または送信されたものすべての暗号化を規定し得るだろう。

【００６４】

図６～図１１は、本開示の実施形態に従った、オブジェクトコードにポリシーエンジン１３２を投入することによってソフトウェアアプリケーション１１８～１２２の挙動を変更するための方法の例示的なブロック図を示す。当該方法は、ソフトウェアアプリケーション１１８～１２２がデータ漏洩防止ポリシー、アクセス制御ポリシーなどを含むセキュリティポリシーに準拠することを確実にするポリシーエンジン１３２～１３６をソフトウェアアプリケーション１１８～１２２に投入するステップを含み得る。

【００６５】

認識され得るように、ソフトウェアアプリケーション１３２～１３６は、図７～図９に示されるように、セキュリティマネージャアプリケーションモジュール１１２によって提供されるセキュリティサービスにリンクされてもよい。図８に示されるように、ポリシーエンジン１３２～１３６によって実施されるポリシーはリアルタイムに変化し得る（動的である）。これは、実行コンテキスト、さまざまなカウンタ、外部イベント、明確なポリシー再定義、ならびにデバイスのユーザのグループおよびロールメンバーシップの変更を含むがこれらに限定されないさまざまなファクタに基づいてもよい。

【００６６】

図７～図１１に示されるように、セキュリティマネージャアプリケーションモジュール１１２は任意にはポリシーエンジン１３７を含んでもよい。ポリシーエンジン１３２～１３６は、ソフトウェアアプリケーション１１８～１２２に送信されリモートロック、ワイプ、無効化などを含むコマンドの処理をサポートしてもよい。ワイプコマンドを処理することにより、結果として、ソフトウェアアプリケーション１１８～１２２がその初期（未

使用）状態に設定し直される可能性がある。個々のポリシー（およびコマンド）は、すべてのソフトウェアアプリケーション１１８～１２２、ソフトウェアアプリケーション１１８～１２２の一部、単一のソフトウェアアプリケーション１１８、またはソフトウェアアプリケーション１１８の一部に適用されてもよい。

【００６７】

図９に示されるように、ポリシー（およびコマンド）は、各々のソフトウェアアプリケーション１１８～１２２によってセキュリティマネージャアプリケーションモジュール１１２から検索されてもよい。代替的には、図１０に示されるように、ポリシー（およびコマンド）は、各々のソフトウェアアプリケーション１１８～１２２によってデバイス外にあるポリシーサーバ１３８から検索されてもよい。一実施形態においては、ポリシー（およびコマンド）は、各々のソフトウェアアプリケーション１１８～１２２によって、デバイス外にあるポリシーサーバ１３８からポリシー（およびコマンド）を検索するセキュリティマネージャアプリケーションモジュール１１２から検索されてもよい。図１１に示されるように、ポリシー（およびコマンド）はまた、予めポリシー（およびコマンド）を検索した別のソフトウェアアプリケーション１１８～１２２から（セキュリティマネージャアプリケーションモジュール１１２を含む）各々のソフトウェアアプリケーション１１８～１２２によって検索されてもよい。認識され得るように、（セキュリティマネージャアプリケーションモジュール１１２を含む）各々のソフトウェアアプリケーション１１８～１２２によるポリシー（およびコマンド）の検索は、プッシュメカニズム、プルメカニズム、ポーリング、コールバック関数、イベントの登録、同報通信などを含むがこれらに限定されないさまざまな方法を用いて実行されてもよい。

【００６８】

一実施形態においては、ポリシーの実施またはコマンドの実行により、結果として、実行結果が生成される可能性がある。実行結果は、セキュリティマネージャアプリケーションモジュール１１２および／またはポリシーサーバ１３８に送り返されてもよい。一実施形態においては、実行結果は監査イベントであってもよい。実行結果は、アプリケーションの使用についての統計を集める目的で用いられるデータであってもよい。実行結果はまた、１つ以上のソフトウェアアプリケーション１１８～１２２が用いられてからどれくらいの時間が経過したのかを判断する目的で用いられるデータであってもよい。認識され得るように、各々のソフトウェアアプリケーション１１８～１２２においてポリシーが実施されかつコマンドが実行され得る方法は、上述のオブジェクトコード変更プロセスを含み得る。

【００６９】

当業者であれば、いくつかのセキュリティサービスが、それらのオブジェクトコードの変更後に、セキュリティマネージャアプリケーションモジュール１１２によってソフトウェアアプリケーション１１８～１２２に提供され得ることを認識するだろう。たとえば、セキュリティサービスは、認証、承認、監査、シングルサインオン、休止時のデータの保護、移送時のデータの保護、データ漏洩保護ポリシーの実施、アクセス制御ポリシーの実施、アプリケーションコマンドの実行、キーの管理、キーの配布、プログラム間でのセキュアデータの共有、ソフトウェアアプリケーション１１８～１２２間でのセキュア通信、プロビジョニング、アプリケーションライフサイクル管理、損なわれたプラットフォームの検出、損なわれたアプリケーションの検出、などを含み得る。

【００７０】

付加的には、当業者であれば、使用され得るいくつかのタイプの認証、シングルサインオン、データ漏洩保護ポリシー、アクセス制御ポリシー、アプリケーションコマンド、およびセキュリティアーティファクトが存在することを認識するだろう。認証のタイプは、パスワード、ＰＫＩ証明、チャレンジ／レスポンス、ワンタイムパスワード、セキュリティトークン、生体認証などを含み得る。シングルサインオンのタイプは、Ｋｅｒｂｅｒｏｓ、ＮＴＬＭ、ＳＡＭＬ、ＯｐｅｎＩＤ、ＯＡｕｔｈ、ＷＳ－Ｆｅｄ、パスワード、ＨＴＴＰクッキーなどを含み得る。実施可能なデータ漏洩保護ポリシーのタイプとしては、オ

フライン記憶なし、バックアップなしであり、信頼されたアプリケーションに対するオープンイン（open-in）の制限、信頼されたアプリケーションに対するコピー／ペーストの制限を含み得るが、電子メールなし、メッセージ通信なし、チャットなし、ソーシャルシェアリングなし、印刷なし、などであり得る。実施可能なアクセス制御ポリシーのタイプは、認証強度、認証頻度、停止中のタイムアウト、認証セッション期間、使用可能にされたアプリケーションのリスト、ウェブサイト、およびウェブサービスブラックリスト／ホワイトリスト、セキュリティ侵害の検出、不活動期間、タイムフェンス、ジオフェンスなどを含み得る。実行可能なアプリケーションコマンドのタイプは、アプリケーション無効化、リモートロック、リモートワイプなどを含み得る。ソフトウェアアプリケーション１１８～１２２に分配されたセキュリティアーティファクトのタイプは、ユーザクレデンシャル、認証トークン、シングルサインオントークン、識別子、データ漏洩保護ポリシー、アプリケーションポリシー、アプリケーションコマンド、アプリケーション構成などを含み得る。

【００７１】

コンピュータ上でのアプリケーション間における信頼性の確立

当該技術においては、他のアプリケーションが損なわれている可能性があるせいで、セキュアアプリケーションが同じコンピュータ上の別のアプリケーションとデータを共有することができるかどうかを認識しないかもしれないという問題がある。他のアプリケーションはマルウェアであるかもしれず、または、マルウェアによって損なわれた正当なアプリケーションであるかもしれない。いくつかの実施形態は、デバイスがインストールされると、暗号キーまたはキーペアなどの共有の秘密のために、各アプリケーションを中心のセキュリティアプリケーションに登録させるかまたは互いに登録させることによって、モバイルデバイス上におけるアプリケーション間の信頼性を確立するステップを含む。たとえば、デバイスに新しいアプリケーションがインストールされると、ユーザは、パスワードを選択するように促され、パスワードを用いてアプリケーションのためのキーが作成される。別の信頼されたアプリケーションがインストールされたアプリケーションとデータを共有することを所望する場合、他の信頼されたアプリケーションがターゲットアプリケーションのためのキーを用いて、そのためのデータを暗号化することができる。

【００７２】

いくつかの実施形態の技術的な利点として、時間が重要視されない非緊急時に、演算装置上のアプリケーションが互いに対する信頼性を確立することができる点が挙げられる。アプリケーションの署名は、それらアプリケーションが元来インストールされていたときに存在していた署名と一致するかどうかを判断するために、後で比較することができる。

【００７３】

認識され得るように、セキュリティマネージャアプリケーションモジュール１１２は、信頼されたアプリケーションにセキュリティサービスおよび極秘データを提供する態様で、モバイルデバイスなどのデバイス上でソフトウェアアプリケーション１１８～１２２間に信頼性を確立するために用いられてもよい。図１２は、本開示の実施形態に従った、セキュリティマネージャアプリケーションモジュール１１２を用いてソフトウェアアプリケーション１１８～１２２間に信頼性を確立する方法を示す例示的なブロック図である。ソフトウェアアプリケーション１１８～１２２を介してセキュリティサービスについての要求に応答するセキュリティマネージャアプリケーションモジュール１１２は、ソフトウェアアプリケーション１１８～１２２が共通のＰＫＩキーで署名される場合に、セキュリティサービスを促進する。ソフトウェアアプリケーション１１８～１２２が、既存のコンピューティングプラットフォーム能力および／またはセキュリティマネージャアプリケーションモジュール１１２を活用することによってインストールされると、各々のソフトウェアアプリケーション１１８～１２２の署名が検証され得る。図１２に示されるように、コンピューティングプラットフォームは、プラットフォーム署名検証モジュール１４２を介して各々のソフトウェアアプリケーション１１８～１２２の署名を検証してもよく、セキュリティマネージャアプリケーションモジュール１１２は、セキュリティマネージャアプ

リケーション署名検証モジュール１４０を介して各々のソフトウェアアプリケーション１１８～１２２の署名を検証してもよい。一実施形態においては、セキュリティマネージャアプリケーションモジュール１１２は、ソフトウェアアプリケーション１１８～１２２および関連付けられたコンピューティングプラットフォームが損なわれていないことを確認し、ソフトウェアアプリケーション１１８～１２２の署名を検証するために用いられてもよい。

【００７４】

各々のソフトウェアアプリケーション１１８～１２２の署名は、既存のコンピューティングプラットフォーム能力を活用することによって、および／または、セキュリティマネージャアプリケーションモジュール１１２によって、実行時に検証されてもよい。同じＰＫＩキーで署名されたソフトウェアアプリケーション１１８～１２２にとって利用可能な共有の通信メカニズムは、セキュリティマネージャアプリケーションモジュール１１２と信頼されたソフトウェアアプリケーション１１８～１２２のうちの残りとに対する信頼性を確立するために用いられてもよい。ｉＯＳプラットフォーム上におけるこのような共有通信メカニズムの一例は、キーチェーンにデータを書込むことである。

【００７５】

一実施形態においては、ＰＫＩキーの証明書に含まれるかまたはＰＫＩキーの証明書に関連付けられる識別子は、一群のソフトウェアアプリケーション１１８～１２２を互いに信頼させるためにこれらソフトウェアアプリケーション１１８～１２２の各々に加えられてもよい。

【００７６】

別の実施形態においては、すべてのアプリケーションは、それらに埋込まれた共通のキーで構築される。このような場合、複数のアプリケーションデータは共通のキーを含む。図１３は、本開示の実施形態に従った、共通のキー１４４を有するソフトウェアアプリケーション１１８～１２２と通信するセキュリティマネージャアプリケーションモジュール１１２を示す例示的なブロック図である。ソフトウェアアプリケーション１１８～１２２が信頼されることを所望する場合、セキュリティマネージャアプリケーションモジュールの通信交換検証モジュール１４６との通信交換を開始してもよい。この通信交換は、共通のキー１４４で暗号化され得るか、任意には署名され得るか、またはＭＡＣを確認され得る。認識され得るように、多数の特定かつ周知の暗号メカニズムのうちのいずれかをこの通信交換において用いることにより、信頼されることを所望するソフトウェアアプリケーション１１８～１２２とセキュリティマネージャアプリケーションモジュール１１２とが同じ共通のキーを共有することを検証することができる。通信交換を検証する前に、セキュリティマネージャアプリケーションモジュール１１２を用いて、コンピューティングプラットフォームおよび／またはソフトウェアアプリケーション１１８～１２２が損なわれていないことをチェックしてもよい。

【００７７】

図１４は、本開示の実施形態に従った、セキュリティマネージャアプリケーションモジュール１１２に対する信頼性を確立するためのソフトウェアアプリケーション１１８～１２２のパスコード登録を示す例示的なブロック図である。一実施形態においては、ソフトウェアアプリケーション１１８～１２２が最初にインストールされるとき、それらは信頼されていない。信頼性を得るために、ソフトウェアアプリケーション１１８～１２２はセキュリティマネージャアプリケーションモジュール１１２に登録要求を行ってもよい。次いで、セキュリティマネージャアプリケーションモジュール１１２は、信頼されることを要求しているソフトウェアアプリケーション１１８～１２２の名前を示しかつアプリケーション登録パスコード１４８を提供するダイアログをユーザに提示してもよい。このダイアログを提示する前に、セキュリティマネージャアプリケーションモジュール１１２を用いて、コンピューティングプラットフォームおよび／またはソフトウェアアプリケーション１１８～１２２が損なわれていないことをチェックしてもよい。

【００７８】

一実施形態においては、アプリケーション登録パスコード１４８が、信頼されていないソフトウェアアプリケーション１１８～１２２によって表示された対応するダイアログボックスに入力されてもよく、さらに、セキュリティマネージャアプリケーションモジュール１１２と信頼されたソフトウェアアプリケーション１１８～１２２のうちの残りとに対する信頼性を確立するために用いられるキー１５０を導出するために用いられてもよい。この導出されたキー１５０は、セキュリティマネージャアプリケーションモジュール１１２の通信交換検証モジュール１４６との通信交換を開始するために用いられてもよい。上述のとおり、この通信交換は、キー１５０で暗号化されてもよく、任意には署名されてもよく、またはＭＡＣを確認されてもよい。一実施形態においては、このダイアログを提示する前に、変更済みのオブジェクトコードを用いて、コンピューティングプラットフォームおよび／またはソフトウェアアプリケーション１１８～１２２が損なわれていないことをチェックしてもよい。

【００７９】

認識され得るように、セキュリティマネージャアプリケーションモジュール１１２は、アプリケーション信頼性を確立するためにソフトウェアアプリケーション１１８～１２２からの登録要求に応答し得る。セキュリティマネージャアプリケーションモジュール１１８～１２２は、アプリケーション登録パスコード１４８を入力するようユーザに促し、アプリケーション登録パスコード１４８を用いてアプリケーション信頼性を確立するためにキー１５０を生成して、セキュリティマネージャアプリケーションモジュール１１２によってアプリケーションソフトウェア１１８～１２２に提供されるセキュリティサービスを促進し得る。

【００８０】

当該方法のうちの１つを用いて信頼性を確立した結果、キーなどの１つ以上のセキュリティアーティファクトは、セキュリティマネージャアプリケーションモジュール１１２から、新しく信頼されたソフトウェアアプリケーション１１８～１２２に分配され得る。次いで、これらのセキュリティアーティファクトは、セキュアにセキュリティサービスを要求するかまたはデータを交換するのに用いられてもよい。このようにして、セキュリティマネージャアプリケーションモジュール１１２は、ソフトウェアアプリケーション１１８～１２２にキー１５０を提供して、ソフトウェアアプリケーション１１８～１２２がセキュリティマネージャアプリケーションモジュール１１２にセキュリティサービスについての要求を送信することを可能にし得る。

【００８１】

アプリケーション間におけるコンピュータ間の保護された通信

当該技術においては、アプリケーションによって格納されたデータまたはアプリケーション間の通信が、同じデバイス上で実行されるマルウェアによってスヌープ（snoop）される可能性があるという問題がある。マルウェアがデバイスに感染してしまうと、そのマルウェアは、典型的には、メモリ内もしくはディスク上の暗号化されていないデータまたはアプリケーション間の暗号化されていない通信にアクセスすることができる。そして、これは、ユーザに知られることなくバックグランドにおいて密かに行うことができる。

【００８２】

いくつかの実施形態の技術的な利点により、デバイス上の登録済みアプリケーション間における通信および保存されたデータをすべて暗号化することが実質的に可能となる。アプリケーション内のデータの暗号化および復号化のためのキーはそれら自体が暗号化されており、攻撃しようとする者がキーを得るのを阻止する。暗号化されたデータとして１つのアプリケーションに残存するデータは、データを復号化しなくても別のアプリケーションに転送することができ、処理時間および処理電力を節約することができる。代わりに、データを暗号化するキーは、それ自体が暗号化されてラップされた状態で、アプリケーション間を転送される。マルウェアがデバイスに至る行程を見つけたとしても、アプリケーション間におけるデータの転送は比較的セキュアである。

【００８３】

認識され得るように、セキュリティマネージャアプリケーションモジュール１１２は、デバイス上のソフトウェアアプリケーション１１８～１２２間でデータを共有するように構成および／またはプログラムされて、以下の特徴を容易にし得る。すなわち、データは、信頼された一群のアプリケーション間で自由に共有することができるが、信頼されていないアプリケーションにはエクスポートされない可能性がある；データは、信頼されていないアプリケーションから信頼されたアプリケーションにインポートすることができる；信頼されたアプリケーション内に格納されたデータはいつでも暗号化され得る；信頼されたアプリケーション間で共有されるデータは、信頼されたアプリケーション間での移送時に暗号化され得る；大きなデータオブジェクトは、アプリケーション間で共有するために再度暗号化される必要がなくなり、これにより、共有中に大規模な演算が行われるのが回避され得；（ＵＩダイアログなどを受理する）ユーザインタラクションを必要とすることなくアプリケーション間でデータを共有することが可能となり得る。

【００８４】

さらに、所与のデータオブジェクトがどんなアプリケーションに公開される（共有される）べきであるかを示すリストからユーザインタラクションが選択されることが所望される場合、セキュリティマネージャアプリケーションモジュール１１２は、ユーザインターフェイスを介して、信頼されたソフトウェアアプリケーション１１８～１２２のリストを表示するように構成および／またはプログラムされてもよい。データは、ドキュメント、セキュリティアーティファクト、ポリシー、コマンド、アプリケーション間の要求／応答などを含み得る。データは、デバイス上の信頼されたソフトウェアアプリケーション１１８～１２２内で、デバイス上の信頼されていないソフトウェアアプリケーション１１８～１２２内で、またはデバイス外で、生成され得る。ユーザインタラクションは、極めて慎重な扱いを有するビジネスデータまたは機密データのためのダブルチェックとして有用であり得る。

【００８５】

いくつかの実施形態は、デバイスのための集中型キーリポジトリとして作用するセキュリティマネージャアプリケーションを用いる。セキュリティマネージャアプリケーションの利点は、それがキーを高レベルで保護するよう特化されることにより、セキュリティ意識の低い開発者によって書込まれた他のアプリケーションを他の領域において特化することが可能になることであり得る。

【００８６】

図１５は、本開示の実施形態に従った、セキュリティマネージャアプリケーションモジュール１１２でセキュリティを維持するためのアプリケーションデータ保護キー１５２～１５６の使用を示す例示的なブロック図である。図１５に示されるように、各々のソフトウェアアプリケーション１１８～１２２は１つ以上の固有のアプリケーションデータ保護キー１５２～１５６を含み得る。アプリケーションデータ保護キー１５２～１５６を含む特定のソフトウェアアプリケーション１１８～１２２が損なわれている場合、これを用いてそのソフトウェアアプリケーション１１８～１２２に対する露出が制限され得ることが利点である。これによっても、いずれかの所与のキーを使用する頻度を減らして、より長い期間にわたってセキュアに使用することを可能にし得る。セキュリティマネージャアプリケーションモジュール１１２に対して信頼性が確立されると、各々のソフトウェアアプリケーション１１８～１２２のためのアプリケーションデータ保護キー１５２～１５６が生成され得る。これらは、ソフトウェアアプリケーション１１８～１２２自体の内部で、またはセキュリティマネージャアプリケーションモジュール１１２内部で生成されてもよい。

【００８７】

ソフトウェアアプリケーション１１８～１２２のためのアプリケーションデータ保護キー１５２～１５６をセキュアに格納する一方法は、セキュリティマネージャアプリケーションモジュール１１２によって維持される複数のデータ保護ルートキー１５８のうちの１つによって暗号化されたキーを、ソフトウェアアプリケーション１１８～１２２自体の内

部に暗号化された形式で残存させるためのものである。ソフトウェアアプリケーション１１８～１２２は、アプリケーションデータ保護キー１５２～１５６を用いる必要がある場合、セキュリティマネージャアプリケーションモジュール１１２との要求／応答交換を開始して、対応するデータ保護ルートキー１５８を用いてアプリケーションデータ保護キー１５２～１５６を復号化してもよい。要求は、暗号化されたアプリケーションデータ保護キー１５２～１５６を含んでもよく、応答は復号化されたアプリケーションデータ保護キー１５２～１５６を含んでもよい。アプリケーションに関連付けられたアプリケーションデータ保護キーがアプリケーションによって維持されるので、必要な場合にのみロードされることが利点であり得る。

【００８８】

ソフトウェアアプリケーション１１８～１２２のためのアプリケーションデータ保護キー１５２～１５６をセキュアに格納する別の方法は、セキュリティマネージャアプリケーションモジュール１１２によって維持される複数のデータ保護ルートキー１５８のうちの１つによって暗号化されたキーをセキュリティマネージャアプリケーションモジュール１１２内に残存させるためのものである。アプリケーションは、アプリケーションデータ保護キー１５２～１５６を用いる必要がある場合、セキュリティマネージャアプリケーションモジュール１１２との要求／応答交換を開始して、対応するデータ保護ルートキー１５８を用いてアプリケーションデータ保護キー１５２～１５６を復号化し得る。応答は、復号化されたアプリケーションデータ保護キー１５２～１５６を含んでもよい。機密のアプリケーションデータ保護キーがともに格納され、それらのメモリ空間への侵入がよりうまく検出され得ることが利点であり得る。

【００８９】

認識され得るように、これらの２つの上述の方法のいずれかの変更例では、セキュリティマネージャアプリケーションモジュール１１２から、ソフトウェアアプリケーション１１８～１２２に対応するデータ保護ルートキー１５８をソフトウェアアプリケーション１１８～１２２に提供させて、アプリケーションデータ保護キー１５２～１５６自体を復号化してもよい。要求時にアプリケーションデータ保護キーを伝達する必要がないことが利点であり得る。

【００９０】

認識され得るように、アプリケーションデータは、ソフトウェアアプリケーション１１８～１２２およびセキュリティマネージャアプリケーションモジュール１１２のうち少なくとも１つによって生成されるアプリケーションデータ保護キー１５２～１５６を含み得る。アプリケーションデータ保護キー１５２～１５６は、セキュリティマネージャアプリケーションモジュール１１２によって維持される対応するデータ保護ルートキー１５８によって暗号化および復号化されてもよい。

【００９１】

一実施形態においては、セキュリティマネージャアプリケーションモジュール１１２は、ソフトウェアアプリケーション１１８～１２２間における暗号化されたデータオブジェクト１７２～１８２の転送を容易にするために用いられてもよい。データオブジェクト１７２～１８２は、それらデータオブジェクト１７２～１８２または関連するデータオブジェクト１７２～１８２の組に固有であるオブジェクトデータ保護キー１６０～１７０で暗号化されてもよい。これらのオブジェクトデータ保護キー１６０～１７０はさらに、それらが属するソフトウェアアプリケーション１１８～１２２のアプリケーションデータ保護キー１５２～１５６のうち１つ以上で暗号化されてもよい。これにより、転送中にデータオブジェクト１７２～１８２を復号化する必要なしに、または、データオブジェクト１７２～１８２を再度暗号化する必要なしに、ソフトウェアアプリケーション１１８～１２２間でデータオブジェクト１７２～１８２を共有することが可能となり得る。一般に、オブジェクトデータ保護キー１６０～１７０の暗号化および復号化は、データオブジェクト１７２～１８２自体を暗号化および復号化するよりもはるかに速く行われる。というのも、ほとんどのデータオブジェクト１７２～１８２がそれらのオブジェクトデータ保護キー１

６０～１７０よりも有意に大きくなるからである。

【００９２】

図１６は、本開示の実施形態に従った、ソース（第１の）アプリケーション１８４からデスティネーション（第２の）アプリケーション１８６にセキュアにデータオブジェクト１７２～１８２を転送するための方法を示す例示的なブロック図である。当該方法は、ソース（第１の）アプリケーション１８４のアプリケーションデータ保護キー１５２でデータオブジェクト１７２のためのオブジェクトデータ保護キー１６０を復号化するステップと、セキュリティマネージャアプリケーションモジュール１１２によって生成されたデータ共有キー１８８でデータオブジェクト１７２のためのオブジェクトデータ保護キー１６０を暗号化するステップとを含み得る。信頼性が確立されると、データ共有キー１８８は、セキュリティマネージャアプリケーションモジュール１１２からソース（第１の）アプリケーション１８４およびデスティネーション（第２の）アプリケーション１８６に転送され得る。ソース（第１の）アプリケーション１８４およびデスティネーション（第２の）アプリケーション１８６によってセキュリティサービス要求に応答して暗号化および復号化するために新しいデータオブジェクト１７２がセキュリティマネージャアプリケーションモジュール１１２によって共有または使用される必要がある場合、データ共有キー１８８はまた、要求に応じてこれらのソフトウェアアプリケーション１８４および１８６に転送されてもよい。

【００９３】

当該方法はさらに、データオブジェクト１７２と、データオブジェクト１７２のための暗号化されたデータ保護キー１６０とをデスティネーション（第２の）アプリケーション１８６に転送するステップと、データ共有キー１８８でデータオブジェクト１７２のためのオブジェクトデータ保護キー１６０を復号化するステップと、デスティネーション（第２の）アプリケーション１８６のアプリケーションデータ保護キー１５４でデータオブジェクト１７２のためのオブジェクトデータ保護キー１６０を暗号化するステップとを含み得る。

【００９４】

認識され得るように、セキュリティマネージャアプリケーションモジュール１１２は、データオブジェクト１７２およびオブジェクトデータ保護キー１６９をソース（第１の）ソフトウェアアプリケーション１８４からデスティネーション（第２の）ソフトウェアアプリケーション１８６に転送する前にオブジェクトデータ保護キー１６９を暗号化し、データオブジェクト１７２およびオブジェクトデータ保護キー１６０をソース（第１の）ソフトウェアアプリケーション１８４からデスティネーション（第２の）ソフトウェアアプリケーション１８６に転送した後にオブジェクトデータ保護キー１６０を復号化するためのデータ共有キー１８８を生成し得る。

【００９５】

図１７は、本開示の実施形態に従った、データオブジェクト１７２～１８２をソース（第１の）アプリケーション１８４からデスティネーション（第２の）アプリケーション１８６にセキュアに転送するための別の方法を示す例示的なブロック図である。当該方法は、ソース（第１の）アプリケーション１８４のアプリケーションデータ保護キー１５２でデータオブジェクト１７２のためのオブジェクトデータ保護キー１６０を復号化するステップと、デスティネーション（第２の）アプリケーション１８６のアプリケーションデータ保護キー１５４でデータオブジェクト１７２のためのオブジェクトデータ保護キー１６０を暗号化するステップとを含み得る。信頼性が確立されると、デスティネーション（第２の）アプリケーション１８６のアプリケーションデータ保護キー１５４は、セキュリティマネージャアプリケーションモジュール１１２からソース（第１の）アプリケーション１８４に転送されてもよい。ソース（第１の）アプリケーション１８４によってセキュリティサービス要求に応答して暗号化および復号化するために新しいデータオブジェクト１７２がセキュリティマネージャアプリケーションモジュール１１２によって共有または使用される必要がある場合、デスティネーション（第２の）アプリケーション１８６のアプ

リケーションデータ保護キー１５４はまた、要求に応じて転送されてもよい。当該方法はさらに、データオブジェクト１７２とデータオブジェクト１７２のための暗号化されたオブジェクトデータ保護キー１６０とをデスティネーション（第２の）アプリケーション１８６に転送するステップを含み得る。

【００９６】

本開示の実施形態に従うと、セキュリティマネージャアプリケーションモジュール１１２は、制約されたプラットフォーム上でアプリケーション間におけるセキュアデータの共有を促進するために用いられてもよい。デバイスプラットフォームは、ソフトウェアアプリケーション１１８～１２２間でデータが如何に共有され得るかについて制約を課す可能性がある。各々のプラットフォームは、ソフトウェアアプリケーション１１８～１２２間でデータを共有するためのさまざまなメカニズムを有し得るが、各々のメカニズムは特定の制限を有する可能性があり、これにより、セキュアデータの共有のためにそれ単独で使用するのが不適切となるかもしれない。各々のデータ共有メカニズムは、最大データサイズを有し得るかまたはメモリ１０８に全データオブジェクトを配置し得るので、大きなデータオブジェクトを共有するために用いることができない。これは、すべてのソフトウェアアプリケーション１１８～１２２に公然とアクセス可能であるため、極秘データを暗号化されていない形式で共有するために用いることができない。これは、必要なユーザインタラクションをプログラムによって開始することをサポートしない可能性があるので、データオブジェクトを自動的にプログラムによって共有するために用いることができない。これはまた、デスティネーションアプリケーションに対して制御を転送しない可能性があるので、要求／応答処理のために用いることができない。付加的には、これは、アプリケーションがフォアグラウンドにない場合に非アクティブになる可能性があるので、別のアプリケーションからデータを受取るのに用いることができない。

【００９７】

当業者であれば、いくつかのクラスのデータ共有メカニズムがあることを認識するだろう。たとえば：

クラス１：デスティネーションアプリケーションに制御を転送するメカニズムは、ユーザインタラクションを必要とせず、ソースアプリケーションおよびデスティネーションアプリケーションにのみアクセス可能であるが、大きなデータオブジェクトを共有するために用いることはできない。このクラスのメカニズムの一例として、ｉＯＳプラットフォーム上でのカスタムＵＲＬスキーム処理がある。

【００９８】

クラス２：大きなデータオブジェクトを共有するために用いることができるメカニズムは、ユーザインタラクションを必要とせず、限られた一組の信頼されたアプリケーションにのみアクセス可能であるが、デスティネーションアプリケーションには制御を転送しない。このクラスのメカニズムの一例として、ｉＯＳプラットフォーム上のキーチェーンがある。

【００９９】

クラス３：大きなデータオブジェクトを共有するために用いることができるメカニズムは、ユーザインタラクションを必要としないが、デスティネーションアプリケーションに制御を転送せず、すべてのアプリケーションに対して公然とアクセス可能である。このクラスのメカニズムの一例として、ｉＯＳプラットフォーム上のペーストボードがある。

【０１００】

クラス４：デスティネーションアプリケーションに制御を転送するメカニズムは、ユーザインタラクションを必要とせず、大きなデータオブジェクトを共有するために用いることができ、ソースアプリケーションおよびデスティネーションアプリケーションのみにアクセス可能であるが、アプリケーションがバックグラウンドに移行するときに短期間だけアクティブになり、アプリケーションが完全にバックグラウンドにある場合には非アクティブとなる。このクラスのメカニズムの一例として、ｉＯＳプラットフォーム上のローカルの受信ソケットがある。

【０１０１】
クラス５：デスティネーションアプリケーションに制御を転送するメカニズムは、大きなデータオブジェクトを共有するために用いることができ、ソースアプリケーションおよびデスティネーションアプリケーションにのみアクセス可能であるが、ユーザインタラクションがデスティネーションアプリケーションを選択することを必要とする。このクラスのメカニズムの一例として、ｉＯＳプラットフォーム上の登録済みファイルタイプのためのオープン・イン（open-in）ファンクションがある。
【０１０２】
一実施形態においては、ユーザインタラクションを必要とすることなく、制約付きのプラットフォーム上でデータをソースアプリケーション１８４からデスティネーションアプリケーション１８６までセキュアに共有するための方法が提供される。当該方法は、クラス２のデータ共有メカニズムを用いてデータオブジェクト１７２を書込むようにソースアプリケーション１８４に命令を送信するステップと、ソースアプリケーション１８４によってデータオブジェクト１７２を暗号化するステップとを含む。当該方法はさらに、クラス１のデータ共有メカニズムを用いて制御をデスティネーションアプリケーション１８６に転送するようにソースアプリケーション１８４に命令を送信するステップを含んでもよく、クラス２のデータ共有メカニズムを用いて書込まれたデータオブジェクト１７２を識別するために十分な情報が含まれている。クラス１のデータ共有メカニズムは、任意には、クラス２のデータ共有メカニズムを用いて書込まれたデータオブジェクト１７２を暗号化するオブジェクトデータ保護キー１６０を含み得る。さらに、当該方法は、クラス２のデータ共有メカニズムを用いてデータオブジェクト１７２を読出し、データオブジェクト１７２を復号化するようにデスティネーションアプリケーション１８６に命令を送信するステップを含み得る。
【０１０３】
ユーザインタラクションを必要とすることなく、要求アプリケーションと応答アプリケーションとの間でセキュア要求／応答インタラクションを実行するために用いられ得る別の方法が提供される。要求アプリケーションは、クラス２のデータ共有メカニズムを用いて要求を書込み得る。この場合、要求のうち少なくともいくらかの部分がソースアプリケーションによって暗号化される。要求アプリケーションは、応答アプリケーションに制御を転送するためにクラス１のデータ共有メカニズムを用いてもよい。応答すべき要求アプリケーションおよびクラス２のデータ共有メカニズムを用いて書込まれた要求を識別するために十分な情報が含まれている。クラス１のデータ共有メカニズムは、任意には、クラス２のデータ共有メカニズムを用いて書込まれた要求を暗号化するキーを含んでもよい。応答アプリケーションは、クラス２のデータ共有メカニズムを用いて要求を読出し、要求の暗号化された部分を復号化し得る。応答アプリケーションはまた、クラス２のデータ共有メカニズムを用いて要求を処理し、応答を書込んでもよく、この場合、応答のうち少なくともいくらかの部分が応答アプリケーションによって暗号化される。さらに、応答アプリケーションは、要求アプリケーションに制御を転送し返すためにクラス１のデータ共有メカニズムを用いてもよい。クラス２のデータ共有メカニズムを用いて書込まれた応答を識別するために十分な情報が含まれている。クラス１のデータ共有メカニズムは、任意には、クラス２のデータ共有メカニズムを用いて書込まれた応答を暗号化するキーを含んでもよい。要求アプリケーションは、クラス２のデータ共有メカニズムを用いて応答を読出し、応答の暗号化された部分を復号化し得る。
【０１０４】
別の実施形態においては、要求および応答がさほど大きくない限り、要求の位置および任意のキーが上述の方法とは逆になり、結果として下記の方法が得られる可能性がある。要求アプリケーションはクラス２のデータ共有メカニズムを用いてキーを書込んでもよい。次いで、要求アプリケーションは、クラス１のデータ共有メカニズムを用いて応答アプリケーションに制御を転送し、少なくともいくらかの部分がキーによって暗号化されている要求を送信し得る。クラス２のデータ共有メカニズムと、応答すべき要求アプリケーシ

ョンとを用いて書込まれたキーを識別するために十分な情報が含まれている。次いで、応答アプリケーションは、クラス２のデータ共有メカニズムを用いてキーを読出し、要求の暗号化された部分を復号化し得る。さらに、応答アプリケーションは要求を処理し、任意には、クラス２のデータ共有メカニズムを用いて新しいキーを書込み得る。応答アプリケーションは、クラス１のデータ共有メカニズムを用いて、要求アプリケーションに制御を転送し返し、キー（要求と同じキーまたは新しい応答キーのいずれか）によって少なくともいくらかの部分が暗号化されている応答を送信し得る。クラス２のデータ共有メカニズムを用いて書込まれたキーを識別するために十分な情報が含まれている。最終的に、要求アプリケーションは、クラス２のデータ共有メカニズムを用いてキーを読出し、応答の暗号化された部分を復号化し得る。

【０１０５】

当業者であれば、上述の記載に照らして、制約されたプラットフォーム上においてアプリケーション間でセキュアデータを共有するための他の代替的な方法を認識するだろう。たとえば、要求のさまざまな部分、応答のさまざまな部分、およびキーのさまざまな部分は、クラス１のデータ共有メカニズムとクラス２のデータ共有メカニズムとの間で多くの方法で分割されてもよい。代替的には、たとえば、クラス３のデータ共有メカニズムを用いて書込まれたいずれかのデータが上述のとおりアプリケーションデータ保護キーを用いて暗号化される場合、上述の方法ではクラス２のデータ共有メカニズムの代わりにクラス３のデータ共有メカニズムが用いられてもよい。別の代替例には、（バックグラウンドへと進むにつれて、各々のソフトウェアアプリケーション１１８～１２２において一時的に利用可能となる）クラス４のデータ共有メカニズムが含まれる。クラス４のデータ共有メカニズムは、たとえば、バックグラウンドに移行するアプリケーションが非アクティブになる前に各々のデータ転送を短期間で実行することができる場合、上述の方法でクラス２のデータ共有メカニズムの代わりに用いられてもよい。

【０１０６】

一実施形態においては、ユーザインタラクションがデスティネーションアプリケーション１８６を選択することを必要とすることなく、信頼されたソースアプリケーション１８４から信頼されたデスティネーションアプリケーション１８６までセキュアにデータを共有するのに用いることができる方法であって、信頼されていないアプリケーションへのデータのエクスポートを防ぐ方法が、提供される。この例示的な方法においては、ソースアプリケーション１８４が、クラス５のデータ共有メカニズムを用いてデータオブジェクト１７２を書込み得る。データオブジェクト１７２は、ソースアプリケーション１８６によって暗号化されてもよく、所与のデータまたはファイルタイプをサポートするアプリケーションのリストから選択する場合に、信頼されたアプリケーションのリストだけが表示されるように、固有のデータまたはファイルタイプとして書込まれてもよい。次いで、デスティネーションアプリケーション１８６は、クラス５のデータ共有メカニズムを用いてデータオブジェクト１７２を読出し、データオブジェクト１７２を復号化してもよい。認識され得るように、ソースアプリケーション１８４からデスティネーションアプリケーション１８６にセキュアにデータオブジェクト１７２を転送するための上述の方法はいずれも、データオブジェクト１７２を暗号化および復号化するために用いられてもよい。

【０１０７】

本開示の実施形態に従うと、場合によっては周知の技術を用いた場合に損なわれる可能性のあるプラットフォームレベルの保護に対する依存性を最小限にするために、上述のようにアプリケーションデータ保護キーを用いて、暗号化されたデータオブジェクトを転送することが好ましいかもしれない。セキュリティのために、アプリケーション外にデータを維持するデータ共有メカニズムを用いて書込まれた如何なる一時データも、それが読出された後には削除されてもよい。

【０１０８】

技術的な利点はアプリケーションと同様に多く存在する。たとえば、ある実施形態においては、スマートフォン上のユーザは、ビジネスインテリジェンスアプリケーション内の

機密データのチャートを準備することができ、ビジネスインテリジェンスアプリケーションから新しい電子メールを開始させることができる。ビジネスインテリジェンスアプリケーションは、ユーザに電子メールを「構成させる」ためにスマートフォンの電子メールクライアントを開く。データオブジェクトにおけるチャートは、ビジネスインテリジェンスアプリケーションにおいて暗号化された形式で残り、オブジェクトデータ保護キーによって暗号化される。オブジェクトデータ保護キーはビジネスインテリジェンスアプリケーション内に残って、それ自体はビジネスインテリジェンスアプリケーション自体のキーで暗号化される。ビジネスインテリジェンスアプリケーションは、スマートフォン上の中心的なセキュリティマネージャアプリケーションに対してそのキーを要求し、中心的なセキュリティマネージャは、これを暗号化されていない形式でビジネスインテリジェンスアプリケーションに提供する。ビジネスインテリジェンスアプリケーションはまた、セキュリティマネージャアプリケーションに対してデータ共有キーを要求し、セキュリティマネージャアプリケーションがデータ共有キーを提供する。ビジネスインテリジェンスアプリケーションは次いで、それ自体のキーでオブジェクトデータ保護キーを復号化し、さらに、データ共有キーでオブジェクトデータ保護キーを暗号化する。次いで、データ共有キーで暗号化されるオブジェクトデータ保護キーは、（既に暗号化された）チャートと共に電子メールクライアントに転送される。電子メールクライアントにおいては、オブジェクトデータ保護キーが、セキュリティマネージャアプリケーションに対して要求されたデータ共有キーで解読される。チャートは、オブジェクトデータ保護キーで解読され、次いで、ユーザが構成することのできる新しい電子メールに挿入される。

【０１０９】

認識され得るように、上述の方法はいずれも、セキュア通信のための方法およびシステム（Methods and Apparatuses for Secure Communication）と題された係属中の米国特許出願連続番号第１３／４０５，３５７号と、ウェブアプリケーションおよびウェブアプリケーションデータとのインタラクションのための方法および装置（Methods and Apparatuses for Interaction with Web Applications and Web Application Data）と題された米国特許出願連続番号第１３／２１５，１７８号とにおいて開示された方法およびシステムの１つ以上とまとめられ、組合わされ、および／または利用され得るものであって、これら特許出願はともに全体が引用によりこの明細書中に援用されている。

【０１１０】

本開示の実施形態に従うと、プロセッサ１１０およびメモリ１０８を有する機械によって読取られたときに当該機械に上述の方法のいずれかに従った動作を実行させる命令を与える機械読取り可能媒体がまた提供される。

【０１１１】

システム、方法および機械読取り可能媒体は、現在最も実用的かつ好ましい実施形態であるとみなされるものに関して説明されてきたが、その開示が必ずしも開示された実施形態に限定されるわけではないことが理解されるべきである。請求項の精神および範囲内に含まれるさまざまな変更例および同様の構成例をカバーするように意図されており、その範囲には、このような変更および同様の構造をすべて包含するように最も広範な解釈が与えられるべきである。本開示は、添付の特許請求の範囲のいずれかの実施形態およびすべての実施形態を含む。

【０１１２】

本発明の本質から逸脱することなくさまざまな変更がなされ得ることも理解されるはずである。このような変更も記載に暗黙的に含まれている。これらの変更は依然として本発明の範囲内にある。この開示が、独立して、かつ全体的なシステムである機械読取り可能媒体として、方法モードおよび装置モードの両方で、発明の多数の局面をカバーする特許をもたらすよう意図されたものであることが理解されるべきである。

【０１１３】

さらに、本発明および請求項のさまざまな要素の各々がまた、さまざまな態様で実現され得る。この開示は、いずれかの装置の実施形態、方法、機械読取り可能媒体もしくはプ

ロセスの実施形態についてのこのような各々の変形例、または単にこれらのいずれかの要素の変形例をも包含するものと理解されるはずである。

【０１１４】

特に、開示が本発明の構成要素に関するものであるので、機能または結果だけが同じあったとしても、各々の構成要素についての用語が方法用語の相当する装置用語によって表わされ得ることが理解されるはずである。このような同等の、より広範囲の、またはさらにより多くの一般名称は、各々の要素または動作の説明に含まれるものとみなされるべきである。このような用語は、この発明が受ける暗黙的に広範な保護範囲を明らかにすることが望まれる場合に、代用することができる。

【０１１５】

すべての動作が、その動作を行うための手段として、または、その動作を引起す要素として表わされ得ることが理解されるべきである。同様に、開示された各々の物理的要素は、その物理的要素が容易にする動作の開示を包含するものと理解されるべきである。

【０１１６】

この特許出願において言及されている如何なる特許、刊行物または他の引用物も、引用によりここに援用されている。加えて、用いられる各々の用語に関して、本願におけるその利用がこのような解釈と一致しない場合を除き、共通の辞書的な定義が、当業者によって認識される標準的な専門辞書のうち少なくとも１つに含まれるような、この明細書中に引用により援用されている各々の用語およびすべての定義、代替的な用語、ならびに同義語として援用されるものと理解されるべきである。

【０１１７】

さらに、出願人に法律上許容可能な最も広範な保護範囲を与えるように、すべての請求項の用語がそれらの最も広範な形で解釈されるべきである。実施形態を添付の図面および具体例に関連付けて説明してきたが、この明細書中に記載されたプロセスおよび装置についての多くの変更例および適応例が、本発明の精神および範囲から逸脱することなく実現可能であることが当業者に容易に認識されるだろう。このため、この記載が単なる例示としてのみなされるものであって、添付の特許請求の範囲において主張されている実施形態の範囲を限定するものとしてなされるわけではないことが明確に理解されるはずである。

【図１】

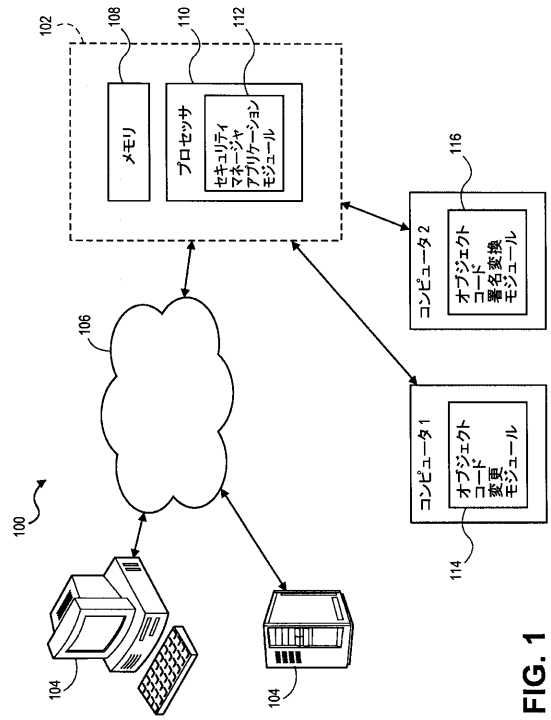

FIG. 1

【図２】

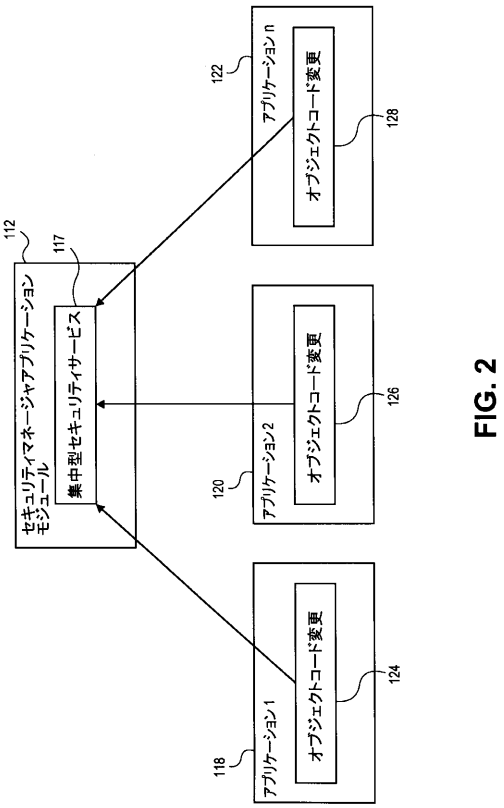

FIG. 2

【図３】

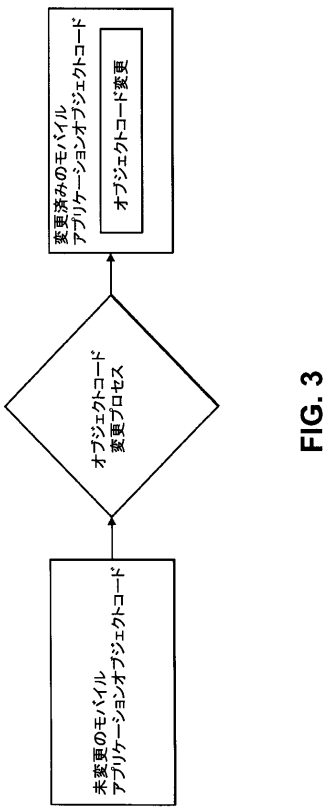

FIG. 3

【図４】

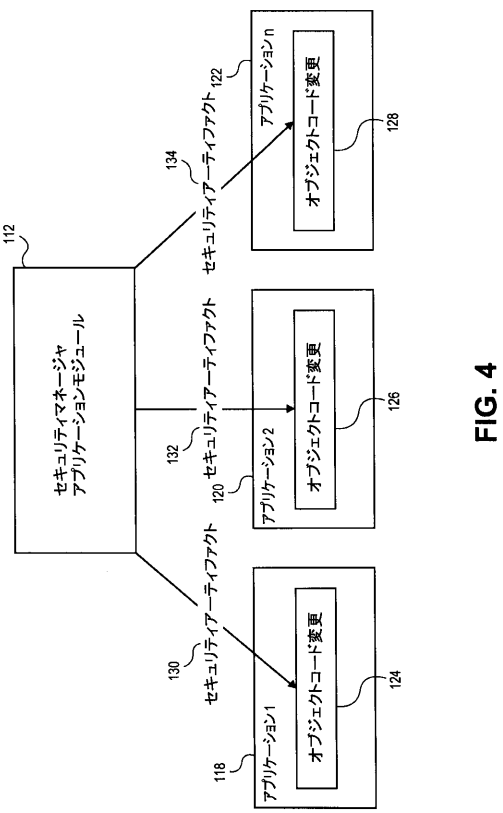

FIG. 4

【図５】

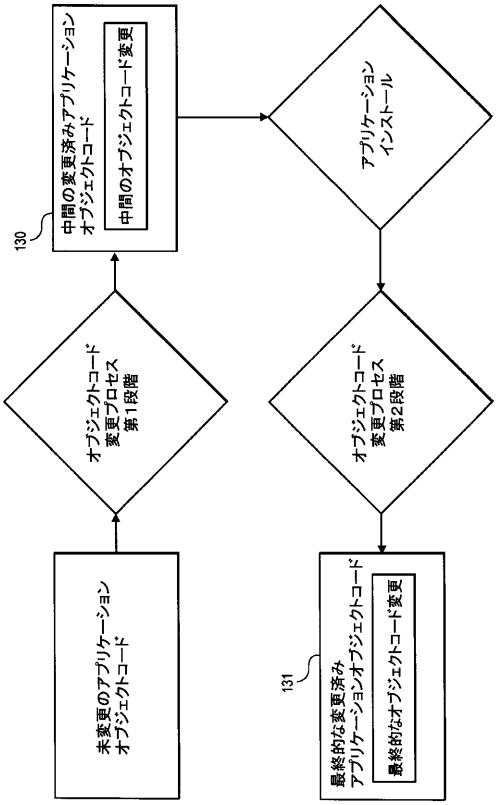

FIG. 5

【図６】

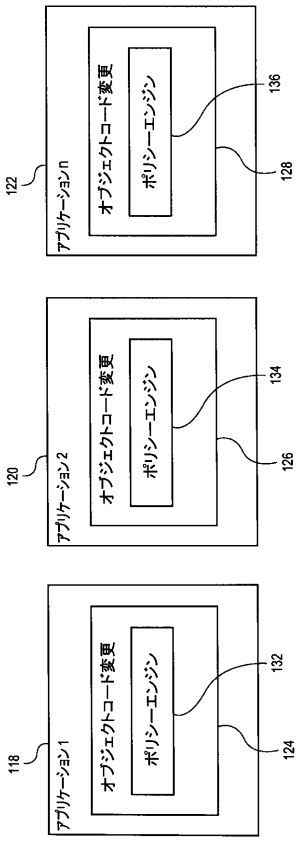

FIG. 6

【図７】

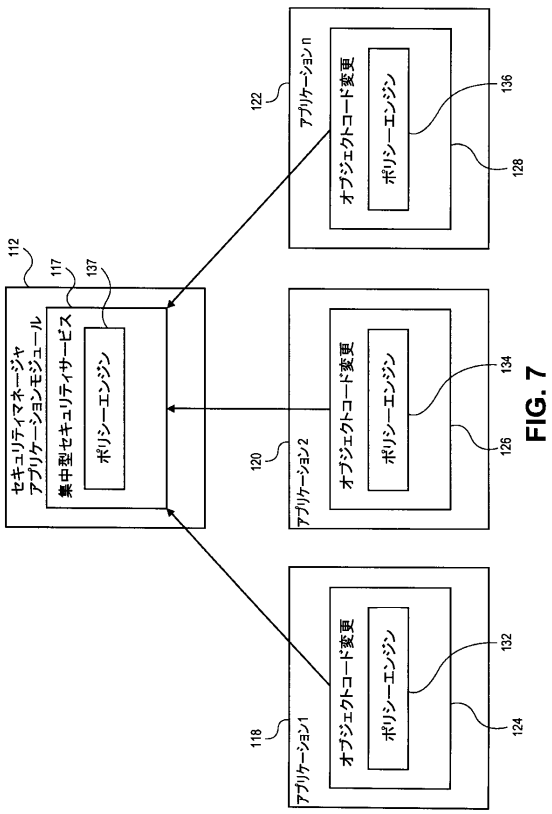

FIG. 7

【図８】

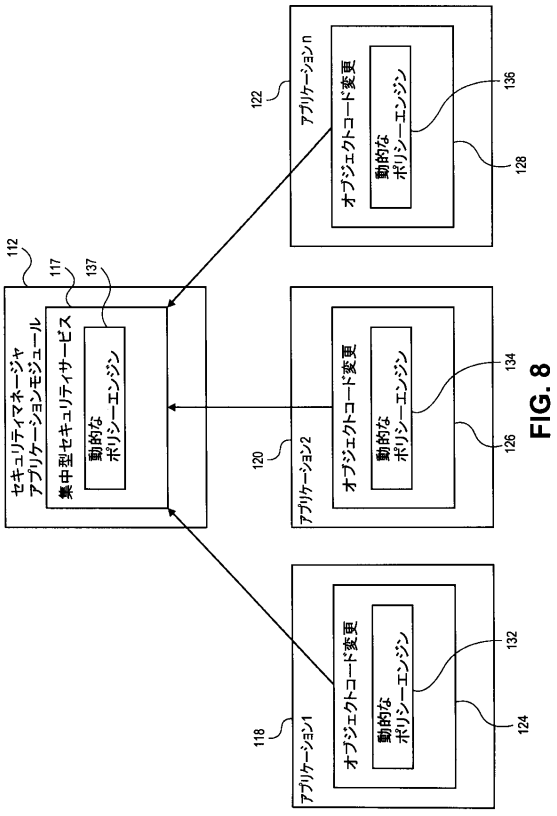

FIG. 8

【図９】

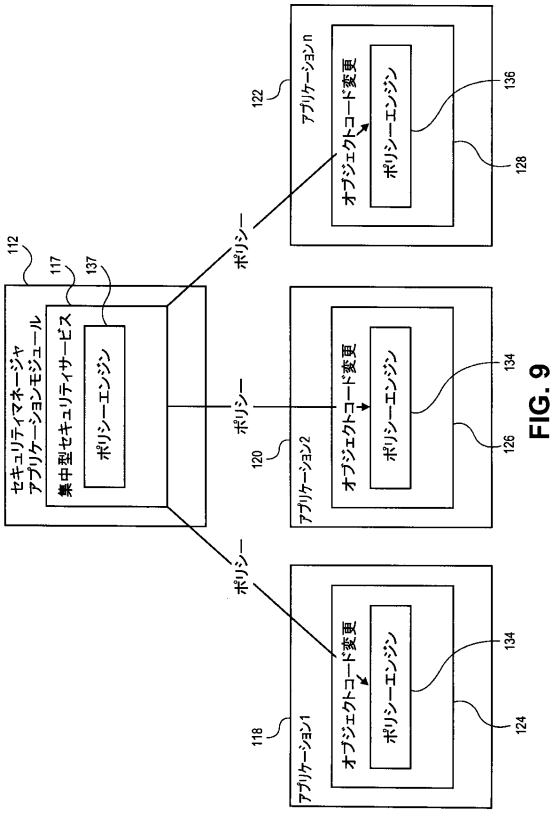

FIG. 9

【図１０】

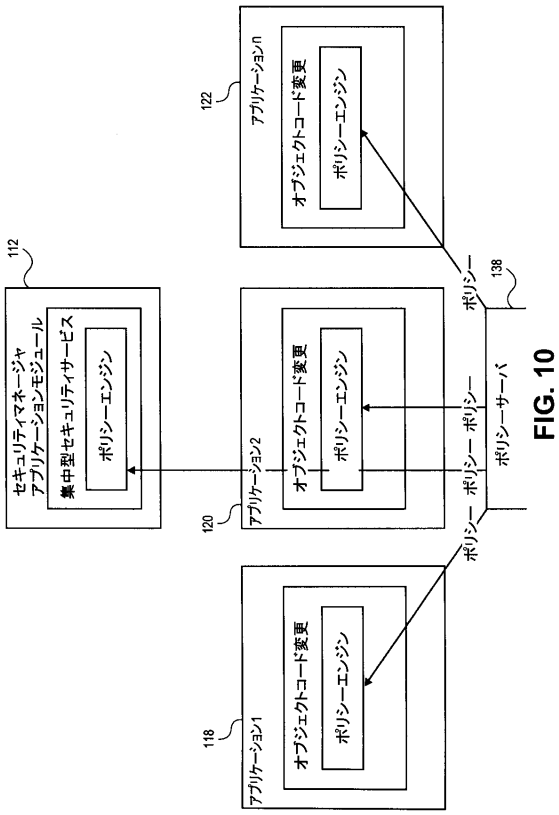

FIG. 10

【図１１】

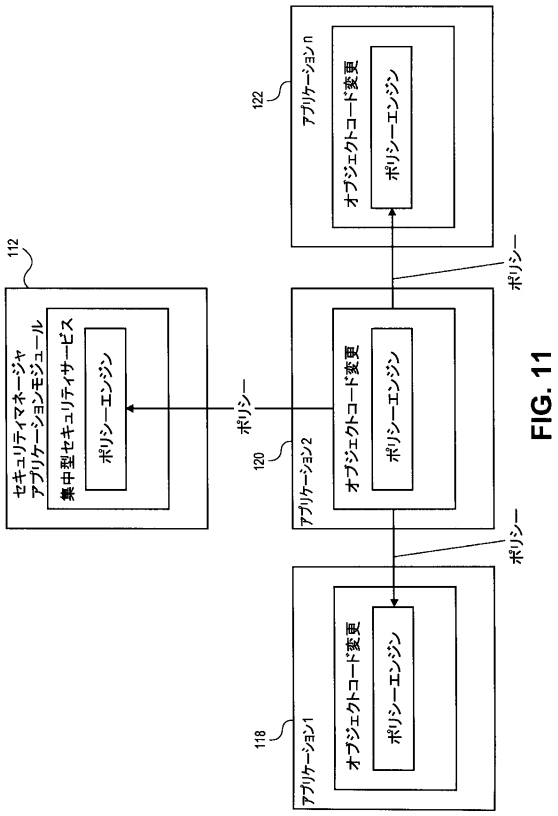

FIG. 11

【図１２】

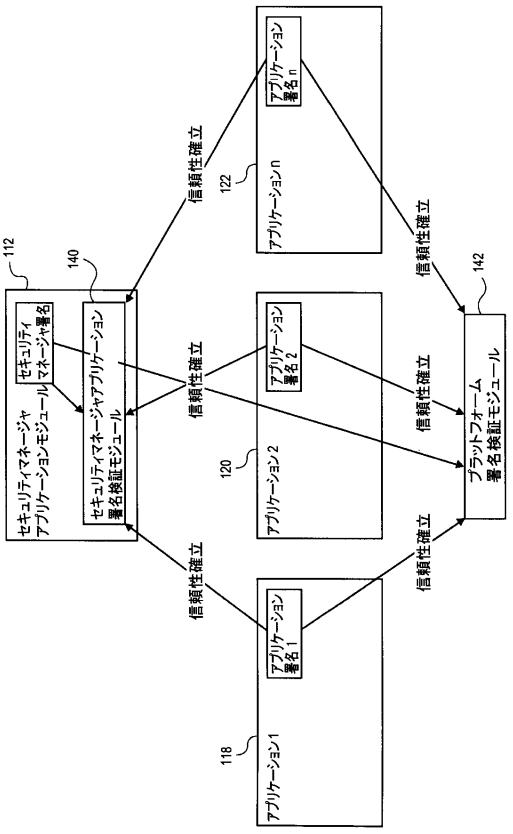

FIG. 12

【図１３】

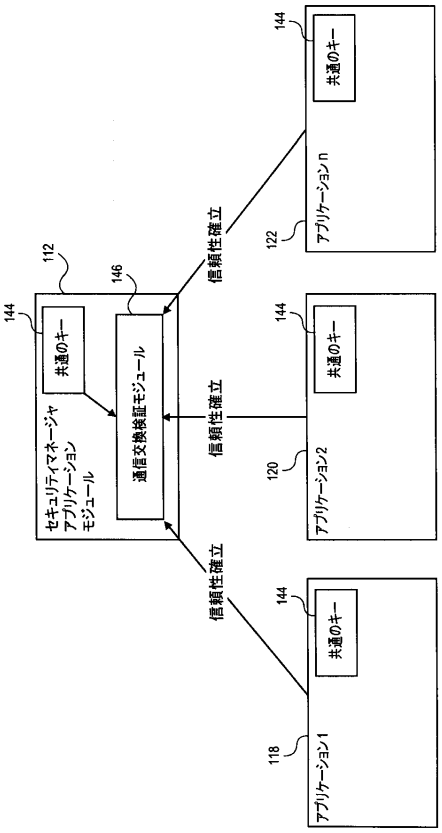

FIG. 13

【図１４】

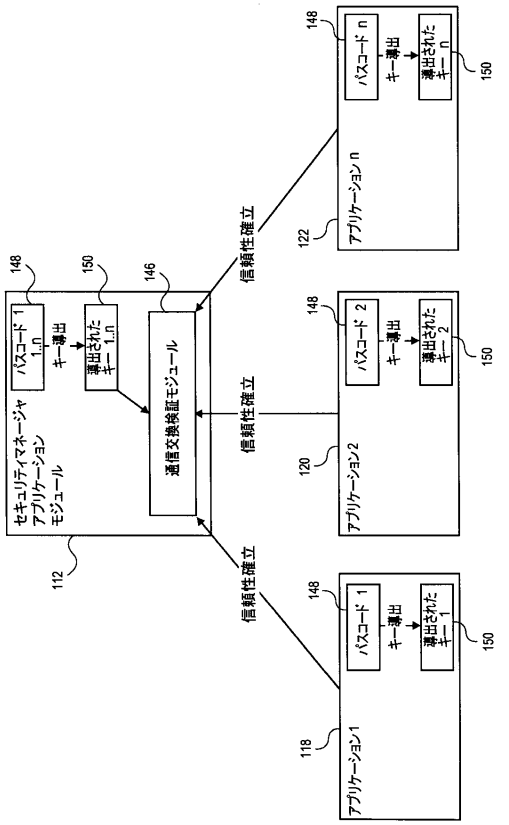

FIG. 14

【図１５】

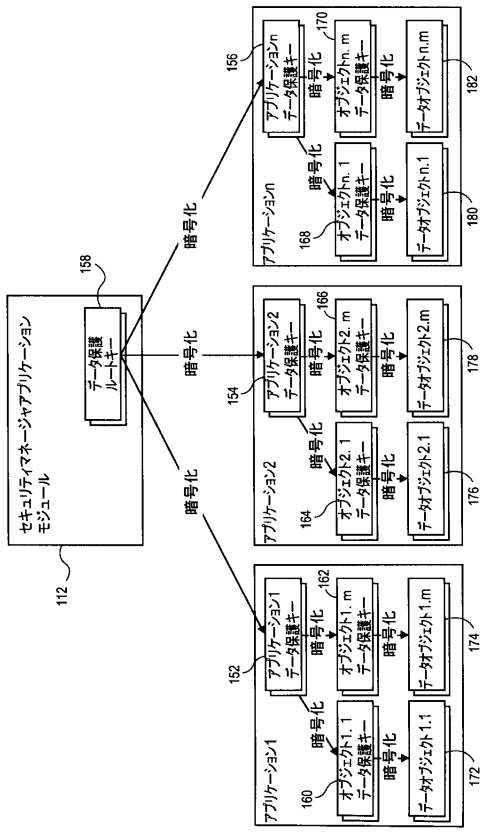

FIG. 15

【図１６】

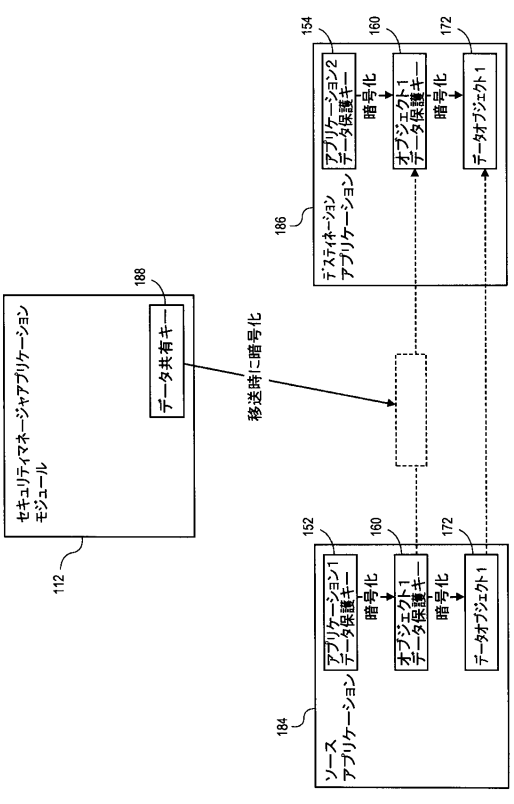

FIG. 16

【図１７】

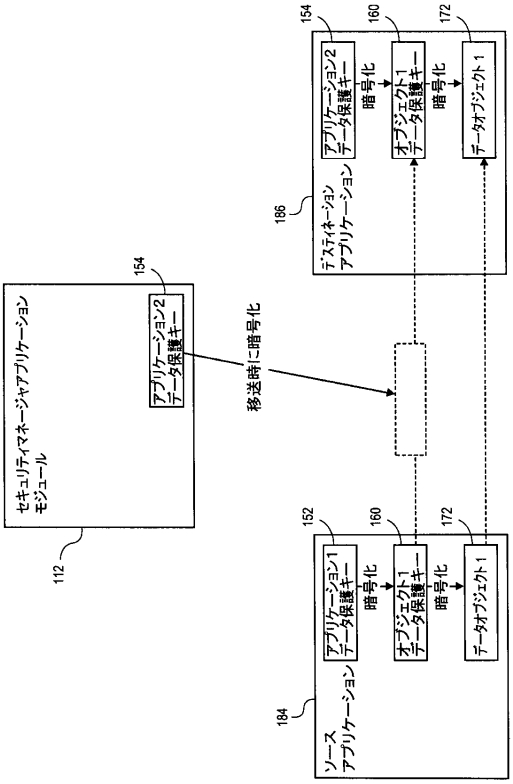
セキュリティマネージャアプリケーション
モジュール
112
154
アプリケーション2
データ保護キー
移送時に暗号化
154
160
172
アプリケーション2
データ保護キー
暗号化
オブジェクト1
データ保護キー
暗号化
データオブジェクト1
デスティネーション
アプリケーション
186
152
160
172
アプリケーション1
データ保護キー
暗号化
オブジェクト1
データ保護キー
暗号化
データオブジェクト1
ソース
アプリケーション
184

FIG. 17

フロントページの続き

(72)発明者　アーメド，アリ・カムラン
アメリカ合衆国、９４０６５　カリフォルニア州、レッドウッド・シティー、オラクル・パークウェイ、５００
(72)発明者　シェファード，ティム
アメリカ合衆国、９４０６５　カリフォルニア州、レッドウッド・シティー、オラクル・パークウェイ、５００
(72)発明者　プラブ，ビナイ
アメリカ合衆国、９４０６５　カリフォルニア州、レッドウッド・シティー、オラクル・パークウェイ、５００
(72)発明者　テワリ，ルチル
アメリカ合衆国、９４０６５　カリフォルニア州、レッドウッド・シティー、オラクル・パークウェイ、５００

審査官　平井　誠

(56)参考文献　米国特許第０６３１７８６８（ＵＳ，Ｂ１）
米国特許出願公開第２０１２／０２１０４４３（ＵＳ，Ａ１）
米国特許出願公開第２００６／０２９１６６４（ＵＳ，Ａ１）
特開平０９－１４８９９３（ＪＰ，Ａ）
米国特許出願公開第２００３／００１８９０６（ＵＳ，Ａ１）
米国特許出願公開第２００２／０１９９１１５（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／
Ｇ０６Ｆ　２１／