(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2017-538217
(P2017-538217A)
(43)公表日　平成29年12月21日(2017.12.21)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/44	(2013.01)	G06F	21/44		5J104
H04L 9/32	(2006.01)	H04L	9/00	675B	
G09C 1/00	(2006.01)	G09C	1/00	640D	
G06F 21/12	(2013.01)	G06F	21/12		

審査請求 未請求　予備審査請求 未請求　（全 14 頁）

(21)出願番号　特願2017-528125 (P2017-528125)
(86)(22)出願日　平成27年11月26日 (2015.11.26)
(85)翻訳文提出日　平成29年7月24日 (2017.7.24)
(86)国際出願番号　PCT/EP2015/077836
(87)国際公開番号　WO2016/083541
(87)国際公開日　平成28年6月2日 (2016.6.2)
(31)優先権主張番号　14306920.1
(32)優先日　平成26年11月28日 (2014.11.28)
(33)優先権主張国　欧州特許庁 (EP)

(71)出願人　501263810
トムソン　ライセンシング
Ｔｈｏｍｓｏｎ　Ｌｉｃｅｎｓｉｎｇ
フランス国，　９２１３０　イツシー　レ　ムーリノー，　ル　ジヤンヌ　ダルク，　１－５
１－５，　ｒｕｅ　Ｊｅａｎｎｅ　ｄ’Ａｒｃ，　９２１３０　ＩＳＳＹ　ＬＥＳ　ＭＯＵＬＩＮＥＡＵＸ，　Ｆｒａｎｃｅ
(74)代理人　100079108
弁理士　稲葉　良幸
(74)代理人　100109346
弁理士　大貫　敏史
(74)代理人　100117189
弁理士　江口　昭彦

最終頁に続く

(54)【発明の名称】アプリケーション整合性の検証を提供する方法及びデバイス

(57)【要約】
未変更アプリケーションを変更することによって取得された変更済みアプリケーションの実行（Ｓ３０２）中、デバイス（１１０）は、未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定し（Ｓ３０４）、未変更アプリケーションの記憶されているチェックサムと比較される（Ｓ３０６）、未変更アプリケーションに対応するコードに関するチェックサムを生成して、これらが一致するかどうかを判定し、変更済みアプリケーションが未変更アプリケーションに対応するコードに対応している場合、及び未変更アプリケーションに対応するコードに関するチェックサムが未変更アプリケーションの記憶されているチェックサムと一致する場合、変更済みアプリケーションの整合性の検証に成功したものとして判定する（Ｓ３１０）。このソリューションは、Android OSを使用するデバイスに特に適しており、これは、インストール中にＤＥＸが最適化されてＯＤＥＸファイルになるか、又はＯＡＴコンパイルされてＥＬＦファイルになり、これらに対して証明済みチェックサムが存在しないためである。

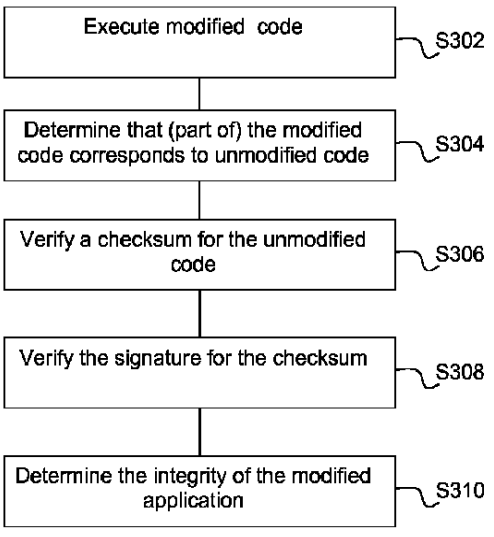


Figure 3

【特許請求の範囲】
【請求項１】
初期アプリケーションを変更することによって取得された変更済みアプリケーションの整合性を判定するデバイス（１１０）であって、
前記変更済みアプリケーションと、前記初期アプリケーションのコードに関する記憶されているチェックサムとを記憶するように構成されたメモリ（１１２）と、
前記変更済みアプリケーションの実行中に、
前記初期アプリケーションの前記コードが前記変更済みアプリケーションのコードに対応していると判定することと、
前記初期アプリケーションの前記コードに関するチェックサムを生成して、生成されたチェックサムを取得することと、
前記生成されたチェックサムと、前記初期アプリケーションの前記コードに関する前記記憶されているチェックサムとを比較して、これらが一致するかどうかを判定することと、
前記変更済みアプリケーションの前記コードが前記初期アプリケーションの前記コードに対応している場合、及び前記生成されたチェックサムが前記アプリケーションのコードに関する前記記憶されているチェックサムと一致する場合、前記変更済みアプリケーションの整合性の検証に成功したものとして判定することと
を行うように構成された処理装置（１１１）と
を含むデバイス。
【請求項２】
前記メモリ（１１２）は、前記未変更アプリケーションの前記記憶されているチェックサムの署名と、署名証明書とを記憶するように構成されており、前記処理装置は、前記署名証明書を使用して前記署名の妥当性を検証することと、前記署名の検証に成功する場合にも、前記変更済みアプリケーションの前記整合性の検証に成功したものとして判定することとを行うように構成されている、請求項１に記載のデバイス。
【請求項３】
前記プロセッサは、前記変更済みコードに対して逆変更を実施して前記未変更アプリケーションに対応するコードを取得して、前記未変更アプリケーションに対応する前記コードが前記変更済みアプリケーションにも対応していると判定するように構成されている、請求項１に記載のデバイス。
【請求項４】
前記未変更アプリケーションに対応する前記コードは、前記未変更アプリケーションであり、前記メモリは、前記未変更アプリケーションを記憶するように更に構成されている、請求項１に記載のデバイス。
【請求項５】
前記プロセッサは、前記変更済みコードと、前記未変更アプリケーションに対応する前記コードとの間の何らかの差が前記変更中に取得された正当な変換に対応しているかどうかを判定して、前記未変更アプリケーションに対応する前記コードが前記変更済みアプリケーションにも対応していると判定するように構成されている、請求項４に記載のデバイス。
【請求項６】
前記プロセッサは、前記未変更アプリケーションに対応する前記コードに対して前記変更を実施して、第２の変更済みコードを取得し、前記変更済みコードと前記第２の変更済みコードとを比較して、前記未変更アプリケーションに対応する前記コードが前記変更済みアプリケーションにも対応していると判定するように構成されている、請求項４に記載のデバイス。
【請求項７】
前記未変更アプリケーションは、インタプリタ型コードとして実装され、及び前記変更済みアプリケーションは、最適化されたインタプリタ型コードとして又はネイティブコー

ドとして実装される、請求項１に記載のデバイス。
【請求項８】
スマートフォン又はタブレットである、請求項１に記載のデバイス。
【請求項９】
未変更アプリケーションを変更することによって取得された変更済みアプリケーションの整合性を判定する方法であって、前記変更済みアプリケーションの実行（Ｓ３０２）中、デバイス（１１０）において、
前記未変更アプリケーションに対応するコードが前記変更済みアプリケーションにも対応していると判定すること（Ｓ３０４）と、
前記未変更アプリケーションに対応する前記コードに関するチェックサムを生成することと、
前記未変更アプリケーションに対応する前記コードに関する前記チェックサムと、前記未変更アプリケーションの記憶されているチェックサムとを比較して（Ｓ３０６）、これらが一致するかどうかを判定することと、
前記変更済みアプリケーションが前記未変更アプリケーションに対応する前記コードに対応している場合、及び前記未変更アプリケーションに対応する前記コードに関する前記チェックサムが前記未変更アプリケーションの前記記憶されているチェックサムと一致する場合、前記変更済みアプリケーションの整合性の検証に成功したものとして判定すること（Ｓ３１０）と
を含む方法。
【請求項１０】
署名証明書を使用して署名の妥当性を検証することと、前記署名の検証に成功する場合にも、前記変更済みアプリケーションの前記整合性の検証に成功したものとして判定することとを更に含む、請求項９に記載の方法。
【請求項１１】
前記未変更アプリケーションに対応するコードが前記変更済みアプリケーションにも対応していると判定することは、前記変更済みコードに対して逆変更を実施して、前記未変更アプリケーションに対応する前記コードを取得することを含む、請求項９に記載の方法。
【請求項１２】
前記未変更アプリケーションに対応するコードが前記変更済みアプリケーションにも対応していると判定することは、前記変更済みコードと、前記未変更アプリケーションに対応する前記コードとの間の何らかの差が前記変更中に取得された正当な変換に対応しているかどうかを判定することを含む、請求項９に記載の方法。
【請求項１３】
前記未変更アプリケーションに対応するコードが前記変更済みアプリケーションにも対応していると判定することは、前記未変更アプリケーションに対応する前記コードに対して前記変更を実施して、第２の変更済みコードを取得することと、前記変更済みコードと前記第２の変更済みコードとを比較することとを含む、請求項９に記載の方法。
【請求項１４】
プロセッサ（１１０）によって実行されると、請求項９に記載の方法を前記プロセッサに実施させる命令を含む、コンピュータで実行可能なプログラム（２２０）。

【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、概して、コンピュータシステムに関し、特に、そのようなシステムにおけるソフトウェアコードの整合性に関する。
【背景技術】
【０００２】

本セクションは、以下で説明及び／又は特許請求される本開示の様々な態様に関連し得る当該技術分野の様々な態様を読者に紹介することを意図している。この説明は、本開示の様々な態様のよりよい理解を促進するための背景情報を読者に提供することに役立つと考えられる。従って、これらの記載がこの観点から読まれるべきであり、先行技術の容認として読まれるべきでないことが理解されるべきである。

【０００３】

様々な理由により、処理装置が実行するソフトウェアが改ざんされていないことを保証することが多くの場合で望ましい。このため、ソフトウェアイメージを改ざん攻撃から保護するために様々な技術が使用され得る。最も一般的な技術は、コードセグメントの署名又はチェックサムを計算し、後の段階でこの署名又はチェックサムを検証することである。チェックサムは、一般的に、秘密性が全くない状態で計算及び検証されるが、暗号署名の生成にはプライベート鍵及び対応する公開鍵の署名の検証が必要である。

【０００４】

チェックサム方式の保護の一例として、Windows（登録商標）オペレーティングシステムで使用されるPortable Executable（ＰＥ）フォーマットのためのＣＲＣ３２がある。ＰＥヘッダがＣＲＣ３２フィールドを含み、このフィールドは、対応するコードセクションのチェックサムを与える。この保護を首尾よく回避するために、攻撃者は、まずコードセクションを変更し、次に、元のチェックサムを、変更されたコードセクションに対して計算された新しい値に置き換える。攻撃者が秘密性を全く必要とせずに、変更されたコードセクションのチェックサムを更新できるため、このタイプの攻撃が可能である。

【０００５】

チェックサムでは弱い場合、暗号署名が好ましいソリューションである。署名の生成はコードの公開前に行われ、プライベート鍵（従って秘密鍵）を使用する。関連付けられた公開鍵はコードに付加され、その後、コードのインストール時又は実行時にコード整合性をチェックするために使用される。この場合も攻撃者はコードを変更できるが、このコードの正しい署名はプライベート鍵がないと生成できないため、攻撃は失敗する。

【０００６】

ネイティブコードで配信されて実行されるアプリケーションの整合性をチェックするソリューションが多く存在し、例えば、Arxan社（GuardIT（商標））、Metaforic社（Metafortress（商標））などから提供されるソリューションが存在する。ネイティブコードは、プロセッサが直接実行できるアセンブラ命令セットである。この命令セットは、インストール後に変化しない。即ち、インストールの前後でプログラムの整合性値は同じままである（即ち、時間に対して一定である）。この場合、署名は、あらかじめ生成され、アプリケーションパッケージとともに配信され得る。

【０００７】

これに対し、インタプリタ型コード（例えば、Java（登録商標）で書かれたコード、Android（登録商標） DEXコードなど）の形式で配布されるアプリケーションは、実行前にインタプリタを通さなければならない中間命令を含む。ネイティブコードと異なり、インタプリタ型コードは、インストール時より後に最適化のために変更され得る。コード変更は、一般に、ターゲットプラットフォームに非常に依存するので、必ずしも予測可能ではない。コードが変更された場合、インタプリタ型コードに対して生成された署名は、コードの整合性及び信憑性を実行時に動的に確認するために使用することができない。

【０００８】

前述のAndroidオペレーティングシステムにアプリケーションソフトウェアを配布してインストールするために、ＡＰＫ － Androidアプリケーションパッケージ － と呼ばれるファイルフォーマットが使用される。ＡＰＫファイルを作成するには、まず、Android用プログラムを中間言語にコンパイルし、その各部分を圧縮アーカイブファイル（ＺＩＰフォーマット）にパッケージする。このアーカイブファイルは、単一ＤＥＸ（Dalvik（登録商標）実行可能コード）ファイル内の全プログラムコード、様々なリソース（例えば、イメージファイル）、及びＡＰＫファイルのマニフェストを含む。このアーカイブフ

ファイルは、２つの追加ファイルＣＥＲＴ．ＳＦ及びＣＥＲＴ．ＲＳＡを含む。ＣＥＲＴ．ＳＦは、他の全てのアーカイブファイルの暗号ハッシュを含み、ＣＥＲＴ．ＲＳＡは、署名の検証に使用される公開鍵を含む。ＣＥＲＴ．ＳＦのみがＲＳＡプライベート鍵によって署名される。ＣＥＲＴ．ＳＦのＲＳＡ署名は、インストール中のＡＰＫファイルの全内容の妥当性検査を可能にする。実際、ＣＥＲＴ．ＳＦファイル内で言及されている全てのファイルが間接的に署名されている。なぜなら、それらのハッシュがＣＥＲＴ．ＳＦに含まれているからである。インストール前にいずれかのファイルを変更すると、ファイルダイジェストがＣＥＲＴ．ＳＦファイル内のハッシュと一致しないことをソフトウェアが検出するので、エラーが発生することになる。或いは、ＣＥＲＴ．ＳＦファイル内の暗号ハッシュ値を変更することは、（既に述べたチェックサム方式の検証に対する攻撃の場合と同様に）署名の検証中のエラーにつながることになる。

【０００９】

ＤＥＸファイルヘッダは、ＤＥＸファイルの内容のグローバルチェックサムも含む。アプリケーションの最初の実行時に、Androidシステムはオプティマイザを使用する。オプティマイザは、実行の直前に、ＤＥＸインタプリタ型バイトコードを、ＯＤＥＸ（最適化ＤＥＸ）と呼ばれる最適化機械語命令シーケンスに変更する。オプティマイザはチェックサムの更新も行う。その後、ＯＤＥＸファイルは、後の使用のためにAndroidファイルシステム内の特定のリポジトリに格納される。その後、ＯＤＥＸファイルは、アプリケーションソフトウェアにとっての基準になり、これが存在する場合、元のＤＥＸファイルはもはや使用されない。

【００１０】

実行時、本システムは、ＯＤＥＸチェックサムを使用して、アプリケーションの整合性を検証することができる。しかしながら、この選択肢は、Androidオペレーティングシステムではデフォルトで設定されておらず、Dalvikマシンは、ＯＤＥＸコードを実行するために使用されているが、常にＯＤＥＸチェックサムを確認するわけではなく、これは、チェックサムの検証が実行のパフォーマンスに無視できない影響を有するためである。

【００１１】

Androidバージョン５．０以上では、Dalvikマシンに取って代わるAndroidランタイム（ＡＲＴ）が導入されている。アプリケーションは引き続きＤＥＸコードで配備されるが、ＤＥＸコードは、インストール時に事前（ＡＯＴ）コンパイル機能によりネイティブコードにコンパイルされる。ＤＥＸファイルに対するＡＯＴコンパイルの結果として、バイナリのExecutable Linkable Format（ＥＬＦ）のファイルが得られる。アプリケーションのＤＥＸコードは、その後、いったんコンパイルされ、後にアプリケーションが実行されるたびにＥＬＦコードが起動される。ＡＲＴは、ネイティブコード（ＥＬＦコード）を直接実行するので、アプリケーションの実行を高速化し、全体の電力消費を改善する。

【００１２】

従って、Androidシステムでは、ＡＰＫ署名が検証されるのはインストール時のみであることがわかる。更に、信頼されないソースからのアプリケーションのインストールをユーザが許可すれば、中央機関の署名がないＡＰＫでもAndroidデバイスにインストールすることができる。従って、アプリケーション開発者は、いずれの信頼されている機関にもリンクしていない独自の自己署名証明書を使用する。その場合、Androidデバイスの所有者が知らないうちに、そのデバイス上で、改ざんされたアプリケーションの再署名及び再インストールが何らかのハッカーによって行われる可能性がある。

【００１３】

既に述べたように、Androidアプリケーションではインタプリタポータブルフォーマット（ＤＥＸ）を使用する。このポータブルフォーマットは、ＡＲＭ、ｘ８６、ＭＩＰＳ、リトル／ビッグエンディアンなどの様々なアーキテクチャ及び特性を有する大規模デバイス群において実行することができる。性能を向上させるために、ＤＥＸコードはインストール時又はアプリケーションの最初の使用時に変更されて、ターゲットデバイスに合わせて最適化されたＯＤＥＸ又はＥＬＦバイナリが生成される。最適化又はＯＡＴコンパイル

中、コード内の様々なものが変更され得る。即ち、命令を他の命令で置き換えることができ、命令の並びを変更することができ、バイト順を入れ替えることができるなどである。

【００１４】

従って、最適化及びＯＡＴコンパイルは、セキュリティの問題を提起する。ＤＥＸファイルの署名は、依然としてＣＥＲＴ．ＳＦ及びＣＥＲＴ．ＲＳＡにより検証され得るが、これはＯＤＥＸファイル及びＥＬＦファイルに当てはまらず、なぜなら、これらが変更されており、その整合性がもはや元のＤＥＸ署名にリンクしていないためである。換言すると、整合性及び信憑性はインストール時にのみ検証することができ、実行時には不可能である。これは、攻撃者がＯＤＥＸ及びＥＬＦのコードを変更し、これに応じてヘッダ内の最終的なチェックサムを更新することが可能なためである。

【００１５】

従って、このシステムは少なくとも２種類の攻撃、即ち、リモート攻撃及びルート攻撃に対して脆弱である。リモート攻撃では、ダウンロードされた悪意のあるアプリケーションがその特権を高め、システムパーミッションを入手する。その後、その悪意のあるアプリケーションは、内部記憶装置のキャッシュリポジトリに格納されているＯＤＥＸファイル及びＥＬＦファイルを改ざんする可能性がある。ルート攻撃では、攻撃者はAndroidデバイスを手に入れる。これは、例えば、デバイスを盗むか、又は所有者がデバイスのセッションをロックせずにいなくなった時点でデバイスにアクセスすることによって行われる。攻撃者は、ＵＳＢリンクを介してデバイスの内部記憶装置からインストール済みアプリケーションを取り出し、そのアプリケーションを変更し、その後、変更したアプリケーションをプッシュして内部記憶装置に戻すことができる。ルート攻撃が成功するには、デバイスを「ルート化」しなければならない（即ち、デバイスのAndroidシステムを制御するために「ルートアクセス権」が必要である）。

【００１６】

従って、Androidアプリケーションのライフサイクル中、アプリケーション整合性の信頼が失われる可能性がある。Androidシステムにインストールされているアプリケーションを信頼することはできるが、実行中のアプリケーションを信頼することは必ずしもできない。

【００１７】

当然のことながら、インタプリタ型コードアプリケーションの整合性及び信憑性に関連する問題の少なくとも一部を克服するソリューションを有することが望ましい。本開示は、そのようなソリューションを提供する。

【発明の概要】

【００１８】

第１の態様では、本開示は、未変更アプリケーションを変更することによって取得された変更済みアプリケーションの整合性を判定するデバイスを提供する。本デバイスは、変更済みアプリケーションと、未変更アプリケーションの記憶されているチェックサムとを記憶するように構成されたメモリと、変更済みアプリケーションの実行中に、未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定することと、未変更アプリケーションに対応するコードに関するチェックサムを生成することと、未変更アプリケーションに対応するコードに関するチェックサムと、未変更アプリケーションの記憶されているチェックサムとを比較して、これらが一致するかどうかを判定することと、変更済みアプリケーションが未変更アプリケーションに対応するコードに対応している場合、及び未変更アプリケーションに対応するコードに関するチェックサムが未変更アプリケーションの記憶されているチェックサムと一致する場合、変更済みアプリケーションの整合性の検証に成功したものとして判定することとを行うように構成された処理装置とを含む。

【００１９】

第１の態様の様々な実施形態は、以下を包含する。

【００２０】

メモリは、未変更アプリケーションの記憶されているチェックサムの署名と、署名証明書とを記憶するように構成されており、処理装置は、署名証明書を使用して署名の妥当性を検証することと、署名の検証に成功する場合にも、変更済みアプリケーションの整合性の検証に成功したものとして判定することとを行うように構成されている。

【００２１】

プロセッサは、変更済みコードに対して逆変更を実施して未変更アプリケーションに対応するコードを取得して、未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定するように構成されている。

【００２２】

未変更アプリケーションに対応するコードは、未変更アプリケーションであり、メモリは、未変更アプリケーションを記憶するように更に構成されている。有利には、プロセッサは、変更済みコードと、未変更アプリケーションに対応するコードとの間の何らかの差が変更中に取得された正当な変換に対応しているかどうかを判定して、未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定するように構成されている。或いは、有利には、プロセッサは、未変更アプリケーションに対応するコードに対して変更を実施して、第２の変更済みコードを取得し、変更済みコードと第２の変更済みコードとを比較して、未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定するように構成されている。

【００２３】

未変更アプリケーションは、インタプリタ型コードとして実装され、及び変更済みアプリケーションは、最適化されたインタプリタ型コードとして又はネイティブコードとして実装される。

【００２４】

本デバイスは、スマートフォン又はタブレットである。

【００２５】

第２の態様では、本開示は、未変更アプリケーションを変更することによって取得された変更済みアプリケーションの整合性を判定する方法を提供する。変更済みアプリケーションの実行中、デバイスが、未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定し、未変更アプリケーションに対応するコードに関するチェックサムを生成し、未変更アプリケーションに対応するコードに関するチェックサムと、未変更アプリケーションの記憶されているチェックサムとを比較して、これらが一致するかどうかを判定し、変更済みアプリケーションが未変更アプリケーションに対応するコードに対応している場合、及び未変更アプリケーションに対応するコードに関するチェックサムが未変更アプリケーションの記憶されているチェックサムと一致する場合、変更済みアプリケーションの整合性の検証に成功したものとして判定する。

【００２６】

第２の態様の様々な実施形態は、以下を包含する。

【００２７】

本方法は、署名証明書を使用して署名の妥当性を検証することと、署名の検証に成功する場合にも、変更済みアプリケーションの整合性の検証に成功したものとして判定することとを更に含む。

【００２８】

未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定することは、変更済みコードに対して逆変更を実施して、未変更アプリケーションに対応するコードを取得することを含む。

【００２９】

未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定することは、変更済みコードと、未変更アプリケーションに対応するコードとの間の何らかの差が変更中に取得された正当な変換に対応しているかどうかを判定することを含む。

【００３０】
未変更アプリケーションに対応するコードが変更済みアプリケーションにも対応していると判定することは、未変更アプリケーションに対応するコードに対して変更を実施して、第２の変更済みコードを取得することと、変更済みコードと第２の変更済みコードとを比較することとを含む。
【００３１】
第３の態様では、本開示は、プロセッサによって実行されると、第２の態様の方法をプロセッサに実施させる命令を含む、コンピュータで実行可能なプログラムを提供する。
【図面の簡単な説明】
【００３２】
以下では、次に示す添付図面を参照しながら、本開示の好ましい特徴を非限定的な実施例として説明する。
【００３３】
【図１】本開示が実施される一例示的システムを示す。
【図２】本例示的システムの機能的態様を示す。
【図３】本開示の好ましい一実施形態による方法の好ましい一実施形態を示す。
【発明を実施するための形態】
【００３４】
本開示によれば、ＯＤＥＸファイル又はＥＬＦファイルの整合性の検証は、対応するＤＥＸの署名を検証することにより、ＯＤＥＸファイル又はＥＬＦファイルがＤＥＸと対応することを検証することにより行われる。
【００３５】
図１は、本開示が実施される一例示的システムを示す。本システムは、デバイス１１０と、アプリケーションプロバイダ（アプリケーションストア）１２０とを含む。デバイス１１０は、Android OSが動作する任意の種類の適切なデバイス、例えば、スマートフォン又はタブレットであってよく、これは、少なくとも１つのハードウェア処理装置（「プロセッサ」）１１１と、メモリ１１２と、ユーザと対話するユーザインタフェース１１３と、インターネットなどの接続１４０を介してアプリケーションプロバイダ１２０と通信する通信インタフェース１１４とを含む。当業者であれば理解されるように、図示されているデバイスは明確さのために非常に簡略化されており、実際のデバイスは、電源及び永続記憶装置などの機能を更に含むであろう。アプリケーションプロバイダ１２０は、デバイス１１０がダウンロードできる少なくとも１つのアプリケーションＡＰＫファイル１２２を記憶しており、このＡＰＫファイルは、署名エンティティによって署名されたＡＰＫ証明書を含む。
【００３６】
図２は、本例示的システムの機能的態様を示す。アプリケーション２２０は、署名エンティティによって署名されたＡＰＫ証明書２２２と、アプリケーションコード２２４（インストール前のＤＥＸと、インストール後のＯＤＥＸファイル又はＥＬＦファイル）と、少なくとも１つの署名済みＤＥＸチェックサム（ＣＳ）２２６（場合によってはリスト形式）、並びに少なくともＡＰＫ証明書の署名に使用された鍵と異なる鍵を使用して署名された場合、署名検証鍵２２８を含む署名証明書と、ソース取得モジュール２３２及び整合性検証モジュール２３４を含むライブラリ２３０とを含む。
【００３７】
ＤＥＸチェックサムは、有利には、ＤＥＸの一部分のチェックサムであり、ＤＥＸ全体のチェックサムに対する追加として与えられてよい。ＤＥＸチェックサムは、ＡＰＫ証明書に署名した署名鍵で署名されてよいが、別の鍵で署名されてもよい。
【００３８】
アプリケーションは、署名済みチェックサムが計算されたＤＥＸのコピーを含んでもよいが、ＯＤＥＸファイル又はＥＬＦファイルを生成するために、又はアプリケーションのインストール後にＡＰＫファイルの少なくとも一部分をＤＥＸコードとともに保持するた

めにこのＤＥＸのコピーが最適化されている場合、ＯＳがこのＤＥＸのコピーを保持することも可能である。

【００３９】

ソース取得モジュール２３２及び整合性検証モジュール２３４は、ＡＰＫのネイティブライブラリに含まれており、ネイティブライブラリにより、アプリケーションとともにパッケージされ、特に署名検証を可能にする拡張ＪＮＩライブラリにアクセスすることが可能である。

【００４０】

ソース取得モジュール２３２は、ＯＤＥＸファイル又はＥＬＦファイルの少なくとも一部分と、これに対応するＤＥＸとを取得して、これらを比較するように構成されている。これは、様々な方法で行うことが可能である。

【００４１】

第１の方法では、ソース取得モジュール２３２は、ＯＤＥＸファイルに逆最適化機能を適用するか、ＥＬＦファイルに逆コンパイル機能を適用して、相当するＤＥＸコードを取得する。ＤＥＸ命令のタイプによっては、ほとんどのＯＤＥＸ及びＥＬＦのファイルコードが可逆である。典型的には、オペコードを置換するのみのＤＥＸ最適化であれば、ＤＥＸからＯＤＥＸに及びＯＤＥＸからＤＥＸに容易に実施可能である。

【００４２】

第２の方法では、ソース取得モジュール２３２は、元のＤＥＸコードを（例えば、ＡＰＫファイルから）取り出し、これをＯＤＥＸ又はＥＬＦのファイルコードと比較して、これら２つの間の差が最適化による正当な変換に対応するかどうかを判定する。正当な変換に対応すると判定されれば、ＯＤＥＸファイル又はＥＬＦファイルは元のＤＥＸに対応していると判定される。

【００４３】

第３の方法では、ソース取得モジュール２３２は、元のＤＥＸコードを（例えば、ＡＰＫファイルから）取り出し、最適化を実施して生成されたＯＤＥＸを取得するか、ＯＡＴコンパイルを実施してＥＬＦファイルを取得し、これらを記憶されているＯＤＥＸファイル又はＥＬＦファイルと比較して、これらが同じかどうかを判定する。ＤＥＸチェックサムは、元のＤＥＸコードから生成される。

【００４４】

従って、これら３つの方法では、ＯＤＥＸファイル又はＥＬＦファイルは、記憶されているＤＥＸ又は生成されたＤＥＸに対応していると判定される。このように、これらのＤＥＸのそれぞれは、ＯＤＥＸファイル又はＥＬＦファイルに対応し、ＯＤＥＸファイル又はＥＬＦファイルを生成するために使用されたＤＥＸに対応している。

【００４５】

ソース取得モジュール２３２が、現在のＯＤＥＸファイル又はＥＬＦファイルが署名済みＤＥＸに対応するＤＥＸから生成されていると判定すると、整合性検証モジュール２３４は、現在のＤＥＸチェックサム及び署名を検証することが可能である。

【００４６】

第１の方法の場合、ソース取得モジュール２３２は、生成されたＤＥＸから現在のＤＥＸチェックサムを計算し、これを署名済みＤＥＸチェックサム２２６と比較する。一致すれば、ＯＤＥＸファイル又はＥＬＦファイルが、取得されたＤＥＸから取得されたことになる。第２及び第３の方法の場合、ソース取得モジュール２３２は、現在のＤＥＸチェックサムを元のＤＥＸから計算する。第４の方法の場合、現在のＤＥＸチェックサムは、元の最適化されていないＤＥＸコードから計算される。

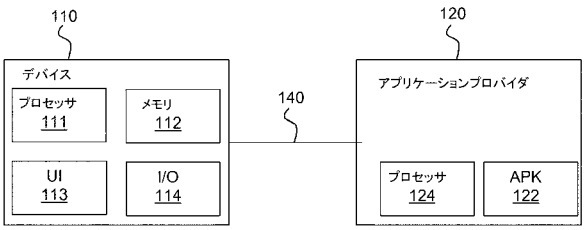
【００４７】

整合性検証モジュール２３４は、ＡＰＫ内の署名証明書（又は同じ鍵が使用された場合にはＡＰＫ証明書）から公開検証鍵２２８を取り出すように構成されている。整合性検証モジュール２３４は、また、検証鍵２２８が取り出された証明書の妥当性を検証することと、ＤＥＸの署名を検証することとを行うように構成されている。

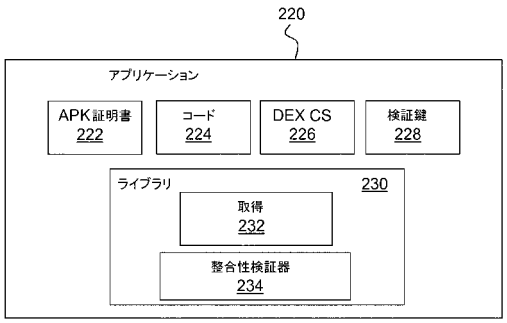
【００４８】
全ての検証が成功する場合、ＯＤＥＸファイル又はＥＬＦファイルは妥当性が確認されたと見なされる。当然のことながら、他の場合には適切な処置が行われてよい。
【００４９】
図３は、好ましい一実施形態による方法のフローチャートを示す。
【００５０】
ステップＳ３０２において、デバイス１１０は、署名が使用可能なＤＥＸ（即ち、未変更コード）を変更することによって取得されたＯＤＥＸファイル又はＥＬＦファイル（即ち、変更済みコード）を実行する。
【００５１】
ステップＳ３０４において、デバイス１１０は、ＯＤＥＸファイル又はＥＬＦファイルの少なくとも一部分がＤＥＸに対応していると判定する。ＤＥＸに対応するコードは、ＤＥＸ自体であってよいが、ＯＤＥＸファイル又はＥＬＦファイルの取得に使用されたＤＥＸのコピーであってもよい。この判定は、本明細書に記載のいずれかの方法で実施されてよい。
【００５２】
デバイス１１０が、ＯＤＥＸファイル又はＥＬＦファイルがＤＥＸに対応していると判定する場合、ステップＳ３０６においてＤＥＸチェックサムが検証される。
【００５３】
ＤＥＸチェックサムの検証に成功する場合、デバイス１１０は、ステップＳ３０８においてＤＥＸチェックサムの署名を検証する。
【００５４】
署名の検証結果に問題がない場合、ステップＳ３１０において、ＯＤＥＸファイル又はＥＬＦファイルの整合性が検証済みであると判定される。これは、ＯＤＥＸファイル又はＥＬＦファイルがＤＥＸに対応しており、ＤＥＸのチェックサムが検証された場合にそのように判定されるからである。
【００５５】
なお、ステップ３０４、３０６、及び３０８は、いかなる順序で実施されてもよい。例えば、最初にＤＥＸチェックサムの署名が検証され（ステップ３０８）、次にＤＥＸチェックサムが検証され（ステップ３０６）、最後にＯＤＥＸファイル又はＥＬＦファイルとＤＥＸとが一致するかどうかが判定される（ステップ３０４）。これらのステップの少なくとも幾つかが並行して実施されてもよい。
【００５６】
整合性の確認は、アプリケーションの実行中に複数回行われてよい。
【００５７】
なお、本ソリューションは、現在配備されているAndroidシステムを全く変更しなくてよい。
【００５８】
本明細書では、「チェックサム」という用語は、チェックサムが生成されたデータがチェックサムの生成後に変更されたかどうかの検証を可能にする値を包含することを意図されている。従って、チェックサムは、例えば、ハッシュ値、巡回冗長検査（ＣＲＣ）値、又は他の種類のダイジェストであってもよく、チェックサムからコードを取り出すことが計算では不可能であることが好ましい。更に、明確さのために単一のチェックサムを使用したが、複数のチェックサムを使用してもよく、コードの別個の部分についてチェックサムを生成してよく（これらの別個の部分は一部が重なり合ってよい）、コードの複数の別個の部分についての複数のチェックサムを使用して、比較用の単一のグローバルチェックサムが生成される。署名は、任意の適切な暗号署名であってよく、例えば、ＨＭＡＣ（ハッシュ方式のメッセージ認証コード）であってよく、又は例えば、ＲＳＡ、ＤＳＡ（デジタル署名アルゴリズム）、若しくはＥＣＤＳＡ（楕円曲線デジタル署名アルゴリズム）に基づく署名であってよい。

【００５９】
当然のことながら、本ソリューションは、ルート攻撃及びリモート攻撃の両方に問題なく対処することが可能である。
【００６０】
ここまで本ソリューションをAndroid環境に関して説明してきたが、本ソリューションは、インストール時にコードを変更し、インストールされたアプリケーションの安全な整合性検証を実行時に可能にしない他のオペレーティングシステムにも適応され得る。
【００６１】
従って、当然のことながら、本開示は、Androidデバイス上でのアプリケーションの実行時整合性を有効にすることができるソリューションを提供する。
【００６２】
本明細書並びに（該当する場合には）特許請求の範囲及び図面に開示される各特徴は、単独で又は何らかの適切な組み合わせで提供されてよい。ハードウェアで実施されるように説明された特徴がソフトウェアで実施されてもよく、その逆であってもよい。請求項中に記載される参照符号は例示に過ぎず、請求項の範囲を限定する効果を有するものではない。

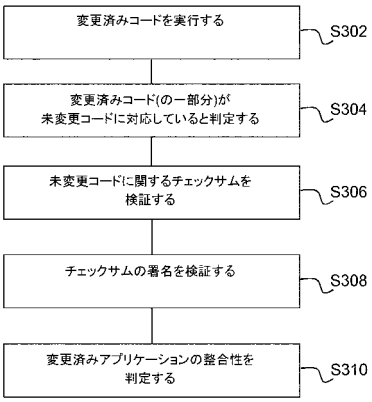
【図１】



【図２】



【図３】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/077836

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/10
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 2 362 314 A1 (THOMSON LICENSING [FR]) 31 August 2011 (2011-08-31) page 1, paragraph 7 - page 3, paragraph 24 page 5, paragraph 39-42; figures 2-4	1-14
X	EP 2 378 452 A1 (THOMSON LICENSING [FR]) 19 October 2011 (2011-10-19) page 3, paragraph 24 - page 6, paragraph 54; figures 2,3	1-14
X	US 2012/144279 A1 (RABELER BRYAN EUGENE [US]) 7 June 2012 (2012-06-07) page 3, paragraph 35 - page 5, paragraph 68; figures 2-4	1-14

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

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 January 2016	09/02/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Bauer, Regine

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2015/077836

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2362314 A1	31-08-2011	CN 102163268 A	24-08-2011
		EP 2362314 A1	31-08-2011
		JP 5734685 B2	17-06-2015
		JP 2011170847 A	01-09-2011
		US 2011202996 A1	18-08-2011
EP 2378452 A1	19-10-2011	BR PI1100138 A2	28-08-2012
		CN 102222196 A	19-10-2011
		EP 2378452 A1	19-10-2011
		JP 5785762 B2	30-09-2015
		JP 2011227897 A	10-11-2011
		US 2011258516 A1	20-10-2011
US 2012144279 A1	07-06-2012	KR 20120063455 A	15-06-2012
		US 2012144279 A1	07-06-2012

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

(74)代理人 100134120
弁理士 内藤 和彦
(74)代理人 100108213
弁理士 阿部 豊隆
(72)発明者 サーモン－ルガネール, チャールズ
フランス国, ３５５７６ セソン－セビニエ, セーエス１７６ １６, ザック デ シャン ブラン, アベニュー デ シャン ブラン ９７５, テクニカラー・アール・アンド・ディー フランス
(72)発明者 カロウミ, モハメド
フランス国, ３５５７６ セソン－セビニエ, セーエス１７６ １６, ザック デ シャン ブラン, アベニュー デ シャン ブラン ９７５, テクニカラー・アール・アンド・ディー フランス

Ｆターム(参考) 5J104 AA09 LA03 PA02