

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5996809号
(P5996809)

(45) 発行日 平成28年9月21日(2016.9.21)

(24) 登録日 平成28年9月2日(2016.9.2)

(51) Int.Cl. F I
G O 6 F 21/14 (2013.01) G O 6 F 21/14

請求項の数 10 (全 10 頁)

(21) 出願番号	特願2015-542937 (P2015-542937)	(73) 特許権者	513192683
(86) (22) 出願日	平成25年9月30日 (2013.9.30)		スンシル ユニバーシティー リサーチ
(65) 公表番号	特表2016-503541 (P2016-503541A)		コンソルティウム テクノパーク
(43) 公表日	平成28年2月4日 (2016.2.4)		SOONGSIL UNIVERSITY
(86) 国際出願番号	PCT/KR2013/008744		RESEARCH CONSORTIUM
(87) 国際公開番号	W02015/023023		M TECHNO-PARK
(87) 国際公開日	平成27年2月19日 (2015.2.19)		大韓民国 ソウル 156-030 ドン
審査請求日	平成27年5月13日 (2015.5.13)		ジャクク サンドドン 511
(31) 優先権主張番号	10-2013-0096514	(74) 代理人	100147485
(32) 優先日	平成25年8月14日 (2013.8.14)		弁理士 杉村 憲司
(33) 優先権主張国	韓国 (KR)	(74) 代理人	100164471
			弁理士 岡野 大和
		(74) 代理人	100188307
			弁理士 太田 昌宏

最終頁に続く

(54) 【発明の名称】 コード難読化装置及びその方法

(57) 【特許請求の範囲】

【請求項 1】

アンドロイドアプリケーションの実行コードを入力される入力部と、
 前記入力された実行コードを分析して、アプリケーションの偽造・変造の攻撃から保護せねばならない重要コードと、前記重要コード以外の一般コードとに区分するコード分析部と、
 前記重要コードをCコードに変換させてネイティブコードを生成するコードコンバータ部と、
 前記ネイティブコード及び前記一般コードを難読化する難読化部と、
 前記難読化されたネイティブコードに偽造・変造探知コードを追加して暗号化した後、ローディングルーチンを追加させて自己書換え技法が適用されたネイティブコードを生成する自己コード保護部と、
 前記自己書換え技法が適用されたネイティブコードと前記難読化された一般コードとを結合するコード結合部と、
 を含むコード難読化装置。

【請求項 2】

前記コード結合部によって結合された前記自己書換え技法が適用されたネイティブコードと前記難読化された一般コードとをクライアントに伝送する伝送部をさらに含む請求項1に記載のコード難読化装置。

【請求項 3】

10

20

前記自己書換え技法が適用されたネイティブコードは、CPU環境で動作し、前記難読化された一般コードは、ダルビック仮想マシンで実行される請求項2に記載のコード難読化装置。

【請求項4】

前記コード分析部は、

既保存された重要コードモデルを用いて、前記入力された実行コードとマッチングされる実行コードを前記重要コードと判断する請求項3に記載のコード難読化装置。

【請求項5】

前記ローディングルーチンは、

前記暗号化されたネイティブコードが実行される時、前記ネイティブコードの暗号化された部分をデコーディングする請求項4に記載のコード難読化装置。

10

【請求項6】

コード難読化装置を利用したコード難読化方法において、

前記コード難読化装置はアンドロイドアプリケーションの実行コードを入力される段階と、

前記コード難読化装置は前記入力された実行コードを分析して、アプリケーションの偽造・変造の攻撃から保護せねばならない重要コードと、前記重要コード以外の一般コードとに区分する段階と、

前記コード難読化装置は前記重要コードをCコードに変換させてネイティブコードを生成する段階と、

20

前記コード難読化装置は前記ネイティブコード及び前記一般コードを難読化する段階と、

前記コード難読化装置は前記難読化されたネイティブコードに偽造・変造探知コードを追加して暗号化した後、ローディングルーチンを追加させて自己書換え技法が適用されたネイティブコードを生成する段階と、

前記コード難読化装置は前記自己書換え技法が適用されたネイティブコードと前記難読化された一般コードとを結合する段階と、

を含むコード難読化方法。

【請求項7】

前記コード難読化装置はコード結合部によって結合された前記自己書換え技法が適用されたネイティブコードと前記難読化された一般コードとをクライアントに伝送する段階をさらに含む請求項6に記載のコード難読化方法。

30

【請求項8】

前記自己書換え技法が適用されたネイティブコードは、CPU環境で動作し、前記難読化された一般コードは、ダルビック仮想マシンで実行される請求項7に記載のコード難読化方法。

【請求項9】

前記実行コードを分析して区分する段階は、

既保存された重要コードモデルを用いて、前記入力された実行コードとマッチングされる実行コードを前記重要コードと判断する請求項8に記載のコード難読化方法。

40

【請求項10】

前記ローディングルーチンは、

前記暗号化されたネイティブコードが実行される時、前記ネイティブコードの暗号化された部分をデコーディングする請求項9に記載のコード難読化方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コード難読化装置及びその方法に係り、より詳細には、アンドロイドアプリ

50

ケーションの偽造・変造の防止のための自己書換え基盤の難読化装置及びその方法に関する。

【背景技術】

【0002】

既存の難読化装置及びその方法は、ジャバ仮想マシンで動作するアプリケーションを対象にしてアプリケーションを構成するコード構造を変更することによって、難読化を適用する。ここで、難読化が適用されたアプリケーションは、難読化適用前と同様に、ジャバ仮想マシンで動作する命令語集合と目的ファイル構造とで構成される。

【0003】

しかし、既存の難読化装置または方法による場合、仮想マシンで動作する管理されるコード (Managed Code) が有する逆工学脆弱性は、難読化以後にも同様に残されるという問題点がある。

【0004】

管理されるコードが有する逆工学脆弱性は、次のようであるが、コンパイルされた目的コードにクラス名、メンバ変数名、メソッド名などソースコード情報が含まれ、コードが明示的に構造化されており、逆工学を通じて特定のロジックを探すか、分析することが容易である。

【0005】

アンドロイドアプリケーションも、前記ジャバマシンで動作するアプリケーションと類似しているようにダルビック仮想マシン (Dalvik Virtual Machine) で動作する管理されるコードで構成され、DEX (Dalvik Executable) ファイル形式で保存される。したがって、アンドロイドアプリケーションは、同一に管理されるコードの逆工学脆弱性が存在する。

【0006】

本発明の背景となる技術は、特許文献1に記載されている。

【先行技術文献】

【特許文献】

【0007】

【特許文献1】韓国登録特許第10-1234591号公報

【発明の概要】

【発明が解決しようとする課題】

【0008】

本発明が解決しようとする課題は、難読化の対象となるアプリケーションの重要コード部分をCPU命令語集合で再構成し、重要コードを、管理されるコードからネイティブコード (Native Code) に変換することによって、管理されるコードの逆工学脆弱性の1つである構造化されたコード情報を除去して、相対的に逆工学を難解に作って逆工学を防止する。

【0009】

また、ネイティブコードに変換された重要コードを保護するために、自己書換え基盤のコード保護技法と偽造・変造探知技術とを適用することによって、逆工学分析に対する抵抗性を向上させる。

【課題を解決するための手段】

【0010】

前記のような目的を果たすための本発明の一実施形態によるアプリケーションコード難読化装置は、アンドロイドアプリケーションの実行コードを入力される入力部と、前記入力された実行コードを分析して、アプリケーションの偽造・変造の攻撃から保護せねばならない重要コードと、前記重要コード以外の一般コードとに区分するコード分析部と、前記重要コードをCコードに変換させてネイティブコードを生成するコードコンバータ部と、前記ネイティブコード及び前記一般コードを難読化する難読化部と、前記難読化されたネイティブコードに偽造・変造探知コードを追加して暗号化した後、ローディングルーチ

10

20

30

40

50

ンを追加させて自己書換え技法が適用されたネイティブコードを生成する自己コード保護部と、前記自己書換え技法が適用されたネイティブコードと前記難読化された一般コードとを結合するコード結合部と、を含む。

【0011】

また、前記コード結合部によって結合された前記自己書換え技法が適用されたネイティブコードと前記難読化された一般コードとをクライアントに伝送する伝送部をさらに含む。

【0012】

また、前記自己書換え技法が適用されたネイティブコードは、CPU環境で動作し、前記難読化された一般コードは、ダルビック仮想マシンで実行可能である。

10

【0013】

また、前記コード分析部は、既保存された重要コードモデルを用いて、前記入力された実行コードとマッチングされる実行コードを前記重要コードと判断する。

【0014】

また、前記ローディングルーチンは、前記暗号化されたネイティブコードが実行される時、前記コードの暗号化された部分をデコーディングすることができる。

【0015】

また、前記偽造・変造探知コードは、コードの偽造ないし変造の行為からネイティブコードを保護することができる。

【0016】

20

本発明の一実施形態によるコード難読化方法は、アンドロイドアプリケーションの実行コードを入力される段階と、前記入力された実行コードを分析して、アプリケーションの偽造・変造の攻撃から保護せねばならない重要コードと、前記重要コード以外の一般コードとに区分する段階と、前記重要コードをCコードに変換させてネイティブコードを生成する段階と、前記ネイティブコード及び前記一般コードを難読化する段階と、前記難読化されたネイティブコードに偽造・変造探知コードを追加して暗号化した後、ローディングルーチンを追加して自己書換えネイティブコードを生成する段階と、前記自己書換えネイティブコードと前記難読化された一般コードとを結合する段階と、を含む。

【発明の効果】

【0017】

30

本発明のコード難読化装置及びその方法によれば、難読化の対象となるアプリケーションの重要コード部分をCPU命令語集合で再構成し、重要コードを、管理されるコードからネイティブコードに変換することによって、管理されるコードの逆工学脆弱性の1つである構造化されたコード情報を除去して、相対的に逆工学を難解にして逆工学を防止することができる。

【0018】

また、ネイティブコードに変換された重要コードを保護するために、自己書換え基盤のコード保護技法と偽造・変造探知技術とを適用することによって、逆工学分析に対する抵抗性を向上させる。

【図面の簡単な説明】

40

【0019】

【図1】本発明の実施形態によるコード難読化装置の構成図である。

【図2】本発明の実施形態によるコード難読化方法のフローチャートである。

【図3】本発明の実施形態による難読化及び暗号化されたアプリケーションファイルの構成図である。

【発明を実施するための形態】

【0020】

以下、添付した図面を参照して、当業者が容易に実施できるように本発明の実施形態を詳しく説明する。しかし、本発明は、さまざまな異なる形態で具現され、ここで説明する実施形態に限定されるものではない。そして、図面で本発明を明確に説明するために、説

50

明と関係のない部分は省略し、明細書全体を通じて類似した部分については、類似した図面符号を付する。

【0021】

まず、本発明のコード難読化装置について説明する。

【0022】

図1は、本発明の実施形態によるコード難読化装置の構成図である。

【0023】

図1を参照すれば、本発明の実施形態によるコード難読化装置100は、入力部110、コード分析部120、コードコンバータ部130、難読化部140、自己コード保護部150、及びコード結合部160を含み、伝送部170をさらに含む。 10

【0024】

入力部110は、アンドロイドアプリケーションの実行コードを入力される。ここで、入力部は、キーパッド形態として入力インターフェース部が備えられうる。

【0025】

コード分析部(Code Structure Analyzer)120は、APK(Android Package)ファイル内に圧縮されている実行ファイルであるDEXファイルを分析して、ユーザ入力によって重要コードを既存コードから分離する。

【0026】

コードコンバータ部(Dalvik to C Code Converter)130は、一般コードと分離された重要コードとを分析して、Cコードで再構成してマネージドコード(Managed Code)をネイティブコードに変換する。 20

【0027】

難読化部(Obfuscator)140は、ネイティブコード難読化部(Native Code Obfuscator)141とマネージドコード難読化部(Managed Code Obfuscator)142とを含む。ネイティブコード難読化部141は、ネイティブコードに変換された重要コードに対してネイティブ環境で難読化技法を適用する。また、マネージドコード難読化部142は、一般コードに対してマネージド環境で難読化技法を適用する。

【0028】

自己コード保護部(Self Code Protector)150は、難読化されたネイティブコードを静的分析から保護するために偽造・変造探知コード(Tamper-Detection Code)を追加し、自己書換え基盤のコード保護技法を適用してコードを暗号化する。そして、自己コード保護部150は、前記暗号化されたネイティブコードにローディングルーチンを追加して自己書換えネイティブコードを生成する。 30

【0029】

コード結合部160は、難読化された一般コードと自己書換えネイティブコードとを再びAPKファイルでパッケージングする。

【0030】

伝送部170は、コード結合部160によってパッケージングされたコード(自己書換えネイティブコードと難読化された一般コード)をクライアントに伝送する。この際、伝送部170は、パッケージングされたコードを直接クライアントに伝達することもでき、アンドロイドマーケットなどにアップロードを行って、クライアントがダウンロードされるようにする。 40

【0031】

以下、図2及び図3を通じて本発明の実施形態によるコード難読化方法について説明する。

【0032】

図2は、本発明の実施形態によるコード難読化方法のフローチャートであり、図3は、本発明の実施形態による難読化されて結合されたアプリケーションファイルの構成図である。 50

【0033】

図2を参照すれば、入力部110は、アンドロイドアプリケーションの実行コードを入力される（ステップS210）。ここで、実行コードが入力される過程は、入力インターフェースを通じてユーザがアンドロイドアプリケーションファイルであるAPKファイルを指定する行為、指定されたAPKファイルの圧縮を解除する行為、アンドロイドアプリケーションの実行ファイルであるDEXファイルを指定する行為を含む。

【0034】

アンドロイドアプリケーションは、拡張子がapk形態であるファイルで配布される。ここで、apkは、Android Packageを意味するが、アンドロイドアプリケーションのビルド過程でさまざまなファイルのコンパイルまたはリンク過程を経ながら、圧縮ファイル形態の拡張子がapkであるファイル（以下、APKファイル）が生成される。実行コードは、アンドロイドアプリケーションファイルであるAPKファイル内に圧縮されているdex拡張子を有する実行ファイル（以下、DEXファイル）のコードを意味する。

10

【0035】

コード分析部120は、入力部110に入力された実行コードを分析して、アプリケーションの偽造・変造の攻撃から保護せねばならない重要コードと、前記重要コードを除く一般コードとに区分する（ステップS220）。

【0036】

コード区分のためにコード分析部120は、既保存された重要コードモデルを用いて、前記入力された実行コードで重要コードモデルとマッチングされるコードを重要コードと判断する。ここで、入力された実行コードは、DEXファイルを構成するコードである。また、重要コードとは、ユーザの観点で攻撃者の偽造・変造の行為から保護せねばならないコードを言う。重要コードと反対される概念として重要コード以外のコードを一般コードと称する。

20

【0037】

コードコンバータ部130は、一般コードと分離された重要コードとを分析して、Cコードで再構成してマネージドコードをネイティブコードに変換する（ステップS230）。

【0038】

変換されたネイティブコードは、ELF（Executable and Linkable Format）形態で保存される。ELF形態は、コードの構造が明示的に区分されず、DEX形態に比べて、コードの偽造・変造のための動的分析または静的分析に相対的に容易に露出されない長所がある。また、ELF形態は、命令語構成がダルビック（Dalvik）命令語に比べて、機械語に近い低いレベルのCPU命令語で構成されており、攻撃者の動的分析及び静的分析を難しくする。

30

【0039】

ネイティブコード難読化部141は、ネイティブコードに変換された重要コードに対してネイティブ環境で適用可能な難読化技法を適用する（ステップS240）。

【0040】

自己コード保護部150は、難読化されたネイティブコードを静的分析から保護するために偽造・変造探知コードを追加し、自己書換え基盤のコード保護技法を適用してコードを暗号化する。そして、自己コード保護部150は、前記暗号化されたネイティブコードにローディングルーチンを追加して自己書換えネイティブコードを生成する（ステップS250）。

40

【0041】

その他に、ネイティブコードに追加された偽造・変造探知コードは、動的にアプリケーションの偽造・変造を検査することによって、攻撃者のコード追加または変造の行為を探知することができる。偽造・変造探知コードによって攻撃者の偽造・変造の行為が探知される場合、偽造・変造探知コードは、実行状態にあるアプリケーションを強制終了させる

50

ことによって、攻撃者のコード分析の行為を中止させることができる。また、偽造・変造探知コードは、自己書換え技法によって保護されて攻撃者の偽造・変造探知ルーチンの迂回可能性を減少させる。

【0042】

自己コード保護部150によって追加されたローディングルーチンは、特定コードが実行される時、特定コードの暗号化部分をデコーディングする。すなわち、暗号化された特定コードは、実行時に呼び出されるルーチンのみ一時的にデコーディングされる。したがって、暗号化されたコードが露出される時間的な範囲を減らすことによって、攻撃者によるコードの偽造・変造の行為を防止することができる。

【0043】

マネージドコード難読化部142は、マネージドコードに変換された一般コードに対してマネージド環境で適用可能な難読化技法を適用する（ステップS255）。すなわち、難読化部140は、ハードウェア的にネイティブコード難読化部141とマネージドコード難読化部142とに分離され、ネイティブ環境とマネージド環境という互いに異なる環境で重要コードまたは一般コードを分離して難読化する。

【0044】

そして、コード結合部160は、図3のように難読化された一般コード（Obfuscated General Code）に該当するマネージドコードセクション（Managed Code Section）とローディングルーチン（Loading Routine）とが追加された自己書換え重要コードに該当するネイティブコードセクション（Native Code Section）でAPKファイルをパッケージングする（ステップS260）。

【0045】

図3で難読化されたアンドロイドアプリケーションファイルのコード部分が、マネージドコードセクションとネイティブコードセクションとに区分される。マネージドコードセクションは、難読化部140によって難読化された一般コードを含み、ネイティブコードセクションは、暗号化されたコード（Encrypted Code）及びローディングルーチンを含む。

【0046】

自己コード保護部150によって暗号化されたコードは、ネイティブコードが暗号化された秘密コード（Secret Code）、ネイティブコードに含まれて伝送される秘密データ（Secret Data）、及び偽造・変造防止コード（Tamper Detection Code）を含む。

【0047】

伝送部170は、コード結合部160によって結合された自己書換えネイティブコードと難読化された一般コードとをクライアントに伝送する（ステップS270）。ここで、伝送されるファイルの形態は、クライアントからダウンロードされて設置することができる形態になり、ネイティブコードと一般コードとを含むAPKファイルであり得る。

【0048】

このように、本発明の実施形態によれば、難読化の対象となるアプリケーションの重要コード部分をCPU命令語集合で再構成し、重要コードを、管理されるコードからネイティブコードに変換することによって、管理されるコードの逆工学脆弱性の1つである構造化されたコード情報を除去して、相対的に逆工学を難解に作って逆工学を防止することができる。

【0049】

また、ネイティブコードに変換された重要コードを保護するために、自己書換え基盤のコード保護技法と偽造・変造探知技術とを適用することによって、逆工学分析に対する抵抗性を向上させる。

【0050】

以上、本発明について実施形態を中心に説明した。当業者ならば、本発明の本質的な特

10

20

30

40

50

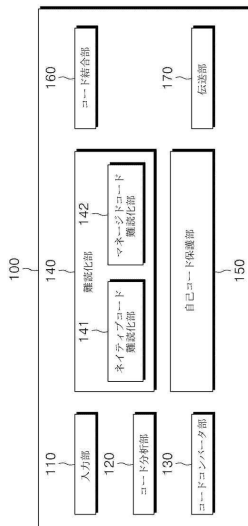
性から外れない範囲内で変形された形態として具現可能であるということを理解できるであろう。したがって、開示された実施形態は、限定的な観点ではなく、説明的な観点で考慮されなければならない。したがって、本発明の範囲は、前述した実施形態に限定されず、特許請求の範囲に記載の内容及びそれと同等な範囲内にある多様な実施形態が含まれるように解釈しなければならない。

【産業上の利用可能性】

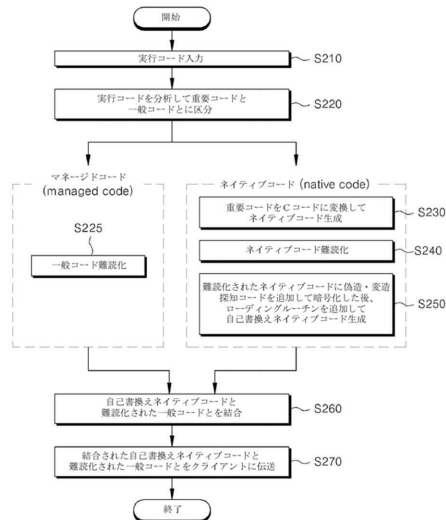
【0051】

本発明は、コード難読化装置及びその方法関連の技術分野に適用可能である。

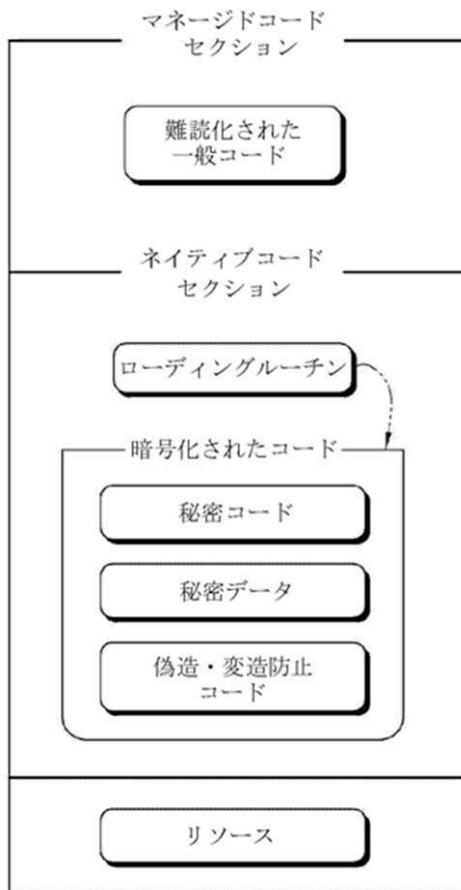
【図1】



【図2】



【図 3】



フロントページの続き

(72)発明者 イ ジョン ヒョン

大韓民国 463-781 キョンギード ソンナムーシ ブンダンーグ スナードン プレウン
マウル サンヨン アパートメント 610-602

(72)発明者 ジュン ジン ヒョク

大韓民国 156-881 ソウル ドンジャクーグ サンドードン 507-20 ナンバー4
02

審査官 脇岡 剛

(56)参考文献 特開2007-233426 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 21/14