ソフトウェア不正アップロード者追跡のための実行形式ファイルへの電子透かし挿入

Watermarking in Executable Files for Tracking Illegal Uploaders.

壷内 将之[1,a)]　岡部 寿男[2]

Masayuki Tsubouchi[1,a)]　Yasuo Okabe[2]

概要：近年デジタルコンテンツの不正アップロードが問題となっている。ソフトウェアの不正コピー防止にはサーバ認証などが有効であるが、正規ユーザには不便な仕様である。そこで本研究では、実行形式ファイルへ電子透かしを挿入することにより、ソフトウェアの不正アップロード者を追跡できるようにすることで、正規ユーザの利便性を損ねることなく不正コピーの流通を抑制することを提案する。

キーワード：電子透かし, 耐タンパソフトウェア, リバースエンジニアリング

,

Abstract: Nowadays illegal uploading of digital contents is a serious problem. Some software venders have solved this issue by authenticating each purchaser via the Internet, but it has a drawback in inconvenience to legitimate users. We propose watermarking in executable files, which enables us to track illegal uploaders. This suppresses illegal distribution of unauthorized files without restricting an innocent user from making his own private copy.

Keywords: Watermarking, Tamper Resistant Software, Reverse Engineering

1. はじめに

近年ソフトウェアの不正コピーが問題となっている。中でも携帯ゲーム機業界では、マジコンと呼ばれる機器が流通しておりゲームソフトの売り上げを妨害している。これは本来はセーブデータのバックアップ等を目的としたものであるが、不正に流通しているゲームソフトのコピーを動作させる機能も併せ持つため、ゲームソフトの不正コピーが深刻な状態となっており、損害賠償額は 9562 万 5 千円にも上った [1]。

現在における不正コピー対策は、「マジコン」対策のように、正規に販売されるデバイス以外では起動できないようなプロテクトをかける手法である。この方法だと確かに不正ソフトは起動できないが、正規に利用しているユーザには不便なシステムである。たとえば、個人目的でデータのバックアップを取るなどの行為もできなくなってしまう。ただし電子書籍の場合では、すでにコンテンツをダウンロードした端末にあるファイルを消去することで新たに別の端末でダウンロードできるようにするサービスや、もう読まなくなったものは端末から消去して、買い取ってもらえるサービスも登場してきている。

2. 研究目標

本研究の目標は、「実行形式ファイルへの静的解析に耐性のある、不正アップロード者追跡のための電子透かしの挿入」とする。

[1] 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University
[2] 京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University
[a)] tsubouchi@net.ist.i.kyoto-u.ac.jp

2.1 実行形式ファイルを対象とする

本研究では画像や音声ではなく、実行ファイルを対象とする。この理由は、画像と音声に関してはハードウェア上で録音やスクリーンショットを撮ることにより、容易にコピーが得られてしまうため、電子書籍などの専用端末を用いなければコンテンツを保護することができないと考えるからである。

2.2 静的解析に耐性

本来であれば動的解析にも耐性のある電子透かしを提案すべきではあるが、本研究ではまず静的解析に耐性のある電子透かしを提案することからはじめる。

2.3 不正アップロード者追跡のための電子透かし

本研究では、不正に流通しているソフトウェアに対し、それが非正規のユーザには実行できないようにすることを目的とはしていない。その理由は、今後プロテクトの技術も進歩していくことが考えられるが、同時に「マジコン」に見られる解析技術も進歩していくことも容易に想像できる。そうなれば現在使われているプロテクトには意味がなくなるため、不正に流通してしまったソフトウェアを実行不可にするという方針は取らないことにする。

例えば、スマートフォンでは現在は端末ごとにインストール可能な OS (Operating System) は固定されているが、今後仮想化や汎用スマートフォンが出現するなど、オープンなプラットフォームがある限りエミュレーションは原理的に可能である。

本研究ではそのような技術の「いたちごっこ」をするのではなく、バイナリを販売時にユーザに配るときに違うバイナリを配り、そこに電子透かしを埋め込んでおく。本人が不特定多数に公開したときに、販売側がその流通しているバイナリを入手することができれば、もともと誰に配ったものかを特定できるようになる。このために一人一人別々のバイナリにするのであり、電子透かしという形態を採用した。

2.4 対象言語

本研究で扱う言語は Java とする。その理由は、Java は JVM(Java 仮想マシン) さえあればあらゆるプラットフォームで動作する仕様であるため、Java の言語仕様やライブラリは更新されることがあっても、JVM の内部やオペコードは固定されている状態が長く続いているため解析されやすく、ソフトウェアを保護にするには最も難易度が高いからである。もし将来 Java が解析不可能になるために複雑な仕様を持つことになるとすれば、下位互換性を保てなくなり、現在使われている Java コードはほとんど再コンパイルしなければならなくなる。したがって JVM の仕様を変更することは決して簡単なことではない。

3. 関連研究

3.1 Java クラスファイルへの電子透かし

Java クラスファイルに対する電子透かしの挿入に関しては、古くから門田ら [2] の方法が使われている。Java 中間言語の命令の中には bipush, sipush, iinc, wideiinc というオペランドの値が変更されても型とクラステーブルの整合性を満たすものが存在する。その命令のオペランドに電子透かしを挿入するという手法を取る。オペランドに対する透かしの挿入方法を開発した点では評価ができるが、コンパイル後に透かしを挿入しているため、挿入される位置が毎回同じなため、同一ソフトウェアに異なる透かしを挿入することができないという欠点がある。

3.2 埋め込み位置の難読化

門田らの研究に対し、福島ら [3] は透かしの埋め込み位置を具体的な実装手順及び埋め込んだ電子透かしの取り出し方法や攻撃耐性については議論されていない。

4. 実装要件

本研究で実装する電子透かしは以下の 2 つの性質を持っていなければならない。

- 結託攻撃耐性
- 透かしそのものの耐性

1 つ目の結託攻撃耐性は、攻撃者がユーザ 2 人分のバイナリを入手することは容易に考えられるので、その 2 つのバイナリから透かしを除去したものが作れないことが求められる。

2 つ目の透かしそのものの耐性とは、例えばすでに透かしが埋め込まれているバイナリに対して再度透かしを埋め込むことにより、元の透かしを読み取れなくしようとする攻撃に対する耐性のことである。

5. 実装方法

5.1 プログラムの分割

まずはじめに Java のクラスファイルのコード部分をおよそ 5 つのブロックに分割する。

5.2 プログラムのシャッフル

次に、分割したコードブロックをシャッフルする。ここで、シャッフルの結果は同一コンテンツでもユーザごとに異なる結果になるようにする。このシャッフルによる結果と行列ファイルが電子透かしとなる。

5.3 条件判定部分の挿入

コードブロックのシャッフルの次は、図 **2** のようにブロックとブロックの間と、コード全体のはじめとおわりに条件判定部分を挿入する。条件判定部分は次の 3 つのス

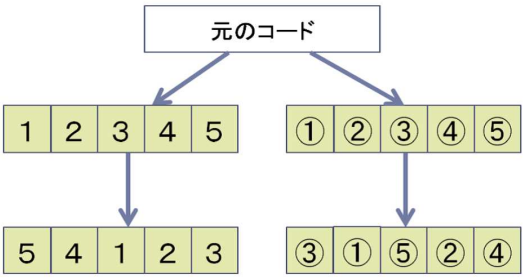

図 1　ブロック分割とシャッフル

テップに分けることができる。

(1) 行列を参照するインデックスの計算

(2) 行列ファイルの参照

(3) ジャンプ

プログラムとは別にまず行列ファイルを用意する。行列ファイルの中身は、行と列のインデックスを入力すると int 型の整数を返すものとする。(1) ではまず行列を参照するためのインデックスを計算する。その計算したインデックスの値に対応する行列の位置を参照する。(3) は、行列から得られた値をもとにジャンプ先のアドレスを求める。

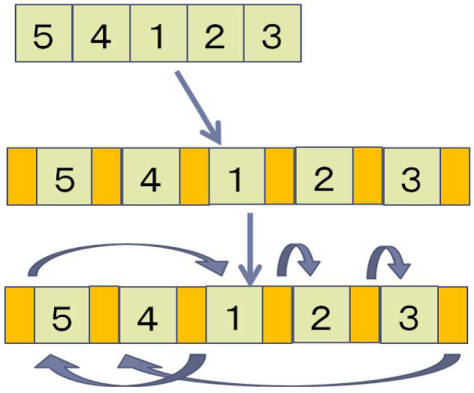

図 2　ブロック分割とシャッフル

6. 今後の課題

本論文で扱った内容では、各ユーザに配布される行列ファイルからジャンプ先アドレスを計算する方法については十分に工夫されているとはいえない。

また、今回提案した手法では静的解析については考慮されているが、実際にプログラムを実行させてみてスタック領域上に現れる呼び出し履歴からプログラムのシャッフルなどを解析するなどの動的解析に関する耐性は考慮されていない。

今後は動的解析を含む様々な攻撃に耐性のある電子透かしを提案することにより、不正アップロードの抑止力に貢献していきたい。

参考文献

[1] 任天堂ニュースリリース 2013 年 7 月 9 日付 http://www.nintendo.co.jp/corporate/release/2013/130709.html

[2] 門田 暁人, 松本 健一, 飯田 元, 井上 克郎, 鳥居 宏次 Java クラスファイルに対する電子透かし法 (¡特集¿:電子化知的財産・社会基盤) 情報処理学会論文誌 03875806 一般社団法人情報処理学会 2000-11-15 41 11 3001-3009

[3] 福島 和英, 櫻井 幸一クラスファイル変換による難読化を用いた JAVA への個人識別情報の埋め込み電子情報通信学会技術研究報告. ISEC, 情報セキュリティ 09135685 一般社団法人電子情報通信学会 2003-05-14 103 61 13-20