様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ４ 月 ２７ 日現在

機関番号：３４３０４
研究種目：若手研究(B)
研究期間：2013～2015
課題番号：２５７３００８７

研究課題名（和文）Java言語を対象とした動的多様化技術

研究課題名（英文）Dynamic Diversifying Methods for the Java Platform

研究代表者
玉田　春昭（TAMADA, Haruaki）

京都産業大学・コンピュータ理工学部・准教授

研究者番号：３０４５７１３９

交付決定額（研究期間全体）：（直接経費）　　3,300,000円

研究成果の概要（和文）：本研究では，新たに導入された機構を利用し，Javaプラットフォームにおける動的多様化技術を次の３つの観点で取り組んだ．(A)実行時の保護プログラムの隠蔽，(B)実行時間遅延の低減，そして，(C)動的解析への耐性評価である．
研究を通じて，次の手法を提案した．(A)実行時に全ての保護情報が暴露しない動的難読化手法，(B)invokedynamic命令を用いた軽量かつ動的な保護手法，(C)では(C-1)コルモゴロフ複雑度による理解困難さ評価，(C-2)パープレキシティによるステルス性評価，(C-3) 適用難読化手法の特定手法，(C-4)より大規模な集合を対象としたバースマーク手法を提案した．

研究成果の概要（英文）：In this research, we realized dynamic diversifying methods in the Java platform by the novel mechanism instrumentation introduced in Java 5.
Our research consists of three categories; (A) hiding the program at the runtime, (B) reducing the overhead from the protection methods, and (C) evaluating the tolerance against dynamic analysis.
The outcomes of our research are; (A) the self-modification technique by methods folding, (B) lightweight and dynamic obfuscation method with invokedynamic instruction, and (C) the evaluation method for robustness of the protection methods by (C-1) Kolmogorov complexity, (C-2) stealthiness by perplexity, (C-3) identifying applied obfuscation methods towards de-obfuscation, and (C-4) birthmarking methods for more large scale software search.

研究分野：ソフトウェアセキュリティ

キーワード：動的難読化　Java　invokedynamic命令　バースマーク

１．研究開始当初の背景

従来から，ソフトウェア内部の秘密情報を隠すため，ソフトウェア難読化が用いられている．難読化とは，プログラムの入出力は変えず，より理解が困難なようにプログラムを変換する技術である．これにより，プログラム中に含まれる様々な秘密情報を隠す．プログラム中のどの部分を隠すかにより，様々な難読化方法が提案されている．特に Java 言語では，バイトコードの仕様がしっかりと定められているため，バイナリの保護がより重要であり，非常に多くの難読化手法が提案されている．しかし，静的に変換するものがほとんどであり，動的解析に対する耐性は議論されてこなかった．それは，Java 言語では，一度ロードしたバイトコードは，基本的にはアンロードできず，さらに書き換えもできないという仕様のためである．従来は，Java の動的難読化の実現には，Java 仮想マシンの修正が必要であり，実現的ではないと言われていた．しかし，Java 5 で導入されたインスツルメンテーション機構により，一部ではあるが，バイトコードの実行中の再定義が可能になった．これにより，Java 言語でも動的難読化が可能になった．

一方，Java 言語の実行環境である Java 仮想マシンは，もはや Java 言語のためだけに存在するわけではない．Scala や Groovy，Jython など，新しい言語や，既存の言語であっても，Java 仮想マシン上で動作するようになってきている．そのため，Java 仮想マシンという Java プラットフォームにおける保護手法が今まで以上に重要となる．

そのため，従来から議論されている静的な難読化だけではなく，Java プラットフォーム上における動的難読化の実現が求められている．

２．研究の目的

１で述べたことを背景とし，本研究課題では，動的解析に対抗するため，Java プラットフォームにおけるプログラムの動的多様化技術(dynamic diversify)に取り組む．多様化とは，難読化を含む，プログラムを実行時に動的に変換する技術であるが，ここでは難読化を主な対象とする．ここで，従来から提案されている自己書き換えによる動的難読化手法に着目する．これは，あらかじめ，プログラムをダミーの情報に書き換えておき，実行時にプログラムがダミー情報を正規の情報に書き換えるようにする難読化手法の一つである．先行研究により実現可能であることは明らかにしている．ただし，従来手法をそのまま Java に導入することはできず，Java プラットフォームに向けての改良が必要であることも同時に明らかにした．その課題は次の２点である．

(a) 実行中のある時点で元のプログラムが暴露すること，
(b) 実行時間の大幅な増加

(a)の問題は，自己書き換え手法が，本来の命令をダミーの命令に書き換える点に起因する．ダミーの命令は，通常の命令列では存在しえない命令になる場合もある．また，実行中のどこかの時点でダミー命令は本来の命令に戻される．戻さなければ，本来の挙動とは異なる挙動になるためである．

一方の(b)は，難読化のトレードオフとも言える．難読化は，盗用など攻撃のない世界には不要な技術である．しかし，それらの対策を行うために，プログラムの仕様を達成するためには不要な処理を行うことになる．それにより遅延は必ず発生する．その中でも自己書き換えは重い処理であり，大きな遅延が発生する．特に Java プラットフォームでは顕著に現れる．

３．研究の方法

先に述べた問題に対応すべく，次の３つの課題を掲げ，研究に取り組んだ．

(A) 実行時の保護プログラムの隠蔽
(B) 実行時間遅延の低減
(C) 動的解析への耐性評価

(A)では，書き換え情報が完全なダミー情報である点が問題となっている．完全なダミーであるということは，実行中のある時点で正しい情報に書き換えられ，ダミー情報を含まない状態が存在しうることを指す．そのため，ダミー情報ではなく，書き換え前後の両方が実行に必要な情報であれば，常に何らかの命令が隠されていることになる．

一方，Java の自己書き換えは，(1) バイトコードのバイナリを書き換え，(2) 以前のバイトコードをアンロードし，(3) 書き換えたバイトコードをロードする，という３つのステップを経て実行される．ここで問題となるのは，それぞれが不可欠な上，大きな遅延を生むことである．これを改善するため，(B)において，堅牢性を保ちつつ自己書き換えの回数を少なくするほか，他の方法についても検討する．つまり，自己書き換え以外の動的難読化の検討である．そのために，難読化の手順の中のボトルネックを特定し，そのボトルネックの回避方法を検討する．そして，更新可能なメソッドポインタとしてのメソッド呼び出し命令 invokedynamic を利用することで動的難読化の実現を試みる．

最後の(C)は，(A), (B)で取り組んだ研究内容を客観的に評価する方法に取り組む．どの程度堅牢な手法であるのか，ステルス性がどの程度であるのか，を評価し，より効果的な難読化手法に改善できるよう取り組む．難読化の堅牢性を評価するためには，難読化の攻撃方法に対する理解も必要である．

４．研究成果

まず，(A)の取り組みでは，メソッドのたたみ込み手法を提案した[学会発表 13]．これは２つのメソッド m_1，m_2 を１つのメソッドにたたみ込むものである．たたみ込みの例を図 1

に示す．このように，m_1 と m_2 の差分をあらかじめ調査し，一方のメソッドからもう一方に書き換える処理を用意する． そして，実行時に，m_2 が必要な時は，m_2 が実行される直前に m_1 に戻す処理(changeM2toM1)を呼び出し，m_1 が必要な時は，m_2 を m_1 に書き換える処理（changeM1toM2）を呼び出す．このようにして，m_1, m_2 のどちらが実行されたとしても，一方の情報が隠されている状態となる．このように秘密情報がプログラム中に含まれていたとしても，実行中のある一点において,全ての情報が暴露しない．

次に，(B)への取り組みにおいて，メソッドの偽装難読化手法を提案した．この取り組みの中で，まず，静的難読化として提案し[雑誌論文 1, 学会発表 6, 8, 10, 11]，その手法を動的難読化に発展させた[学会発表 7]．手法の概略図を図 2 に示す．図 2 (i)に method1 を呼び出す命令がある．

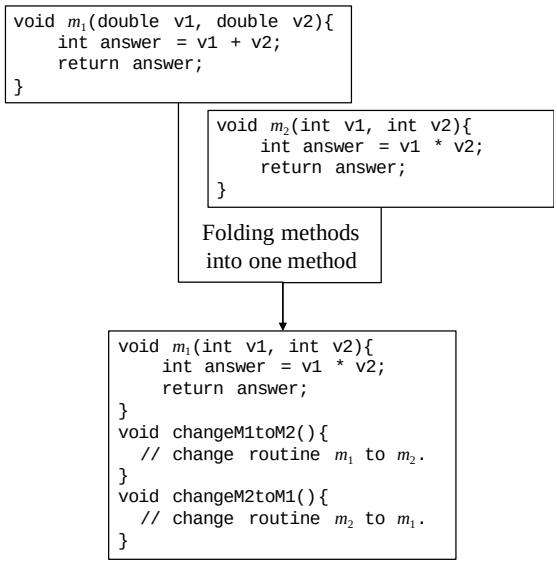

図 1. メソッドのたたみ込みの例

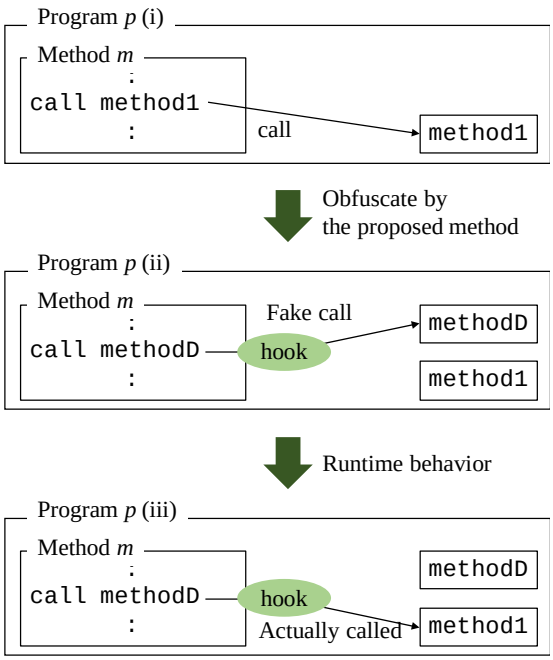

図 2. メソッド偽装難読化の手順

これに偽装難読化を施す．難読化後のプログラムは図 2 (ii)である．図 2 (i)に，メソッド呼び出しのフックと，ダミーメソッド methodD を追加する．そして，method1 の呼び出しを methodD に変更する．こうすることで, methodD が呼び出されているように見える．そして，実行時（図 2 (iii)）には，あらかじめ挿入したフックが methodD ではなく, method1 を呼び出すように呼び出し先を誘導する．このフックの実現には 2 つの方法がある．従来からの方法であるリフレクションを用いる方法と，Java SE 7 で導入された invokedynamic 命令を利用する方法である．両者の実行時間を計測し, invokedynamic 命令を利用する方法の方が実行時間の遅延が抑えられることを確認した．適用範囲は限られるものの，より軽量な難読化手法が実現できた．

取り込み(C)では, 動的解析への評価を目的に研究に取り組んだ．この取り組みではより一般的に，難読化の評価や，プログラムの読みやすさの評価を．大きく，コルモゴロフ複雑度[雑誌論文 2]，命令列の不自然さ[学会発表 9]，コードクローン[学会発表 3]に基づいて難読化の評価を行った．

さらなる研究として，難読化の攻撃に対する耐性の評価を行った．まず，逆難読化に必要なステップを洗い出し，その第 1 ステップとしての適用された難読化手法の特定を試みた[学会発表 2, 4]．そして，適用された難読化を特定する手順を明らかにした．

一方，難読化の堅牢性を評価する手法として，バースマーク手法も提案されている．この手法は，本来は，盗用されたプログラムを発見するための手法である．バースマークは, プログラム中の特徴を元にプログラム同士の類似性を算出する手法である．様々な手法が提案されているものの，類似計算に NP 困難な手法が用いられているなど，膨大な時間がかかる手法もある．大量のプログラムの中から盗用であると疑わしいプログラムを見つけ出すための技術であるにもかかわらず，検索に時間がかかるのは望ましくない．そこで，前処理として，簡易な比較方法を用いて

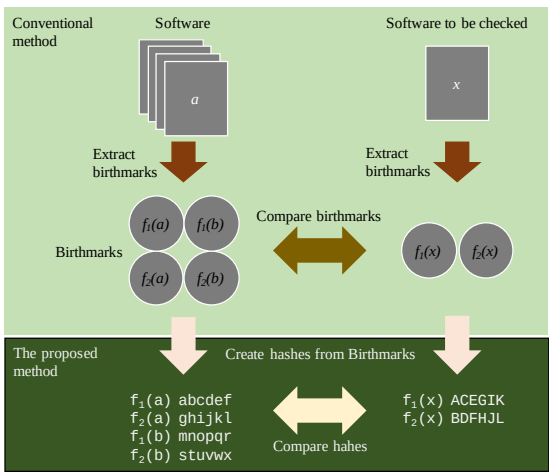

図 3. バースマーク比較の高速化

検索対象を絞り込んだ後に，従来手法で比較する手法を提案した[学会発表 5]．図 3 に手順の概要を示す．既存のバースマーク手法で，プログラムの特徴を抽出した後，特徴をファジーハッシュにかける．ファジーハッシュとは，似たデータ列からは似たデータ列が出るようなハッシュである．そして，ファジーハッシュで高速に比較し，候補を絞り込む．これにより，より高速にプログラム同士を比較できるようになる．

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計２件）

1. Kazumasa Fukuda and Haruaki Tamada, ``To Prevent Reverse-Enginnering Tools by Shuffling the Stack Status with Hook Mechanism,'' International Journal of Software Innovation (IJSI), Volume 3, Issue 3, pp.14-25, 2015（査読あり）.
2. 二村 阿美, 門田 暁人, 玉田 春昭, 神崎 雄一郎, 中村 匡秀, 松本 健一, ``命令のランダム性に基づくプログラム難読化の評価''，コンピュータソフトウェア, Vol. 30, No. 3, pp. 18-24, September 2013（査読あり）.

〔学会発表〕（計１３件）

1. 川端 盛志, 玉田 春昭, ``初学者を対象としたプログラム中の識別子の適切性評価,'' 情報処理学会第 78 回全国大会, 10-12 March 2016，慶応義塾大学 矢上キャンパス（神奈川県横浜市）.
2. 匂坂 勇仁, 玉田 春昭, ``逆難読化に向けた適用された難読化手法の特定,'' 情報処理学会第 78 回全国大会, 10-12 March 2016, 慶応義塾大学 矢上キャンパス（神奈川県横浜市）.
3. 若林 洸太, 門田 暁人, 伊原 彰紀, 玉田 春昭, ``コードクローンと使用ライブラリに着目したオープンソースソフトウェアの進化の定量化''，研究報告ソフトウェア工学（SE）, Vol.2015-SE-190, No.2, pp.1–6, 15, 16 December 2015，JR 博多シティ会議室（福岡県福岡市）.
4. 匂坂 勇仁，玉田 春昭，``適用保護手法特定の試み −不自然さ評価方法を用いて−''，信学技報, Vol. 115, No. 153, SS2015-21, pp. 63-68, 22-24 July 2015，教育文化会館（北海道札幌市）.
5. 山本 照明，玉田 春昭，門田 暁人，``大量のプログラムを対象としたファジーハッシュを用いたバースマーク手法''，2015 年暗号と情報セキュリティシンポジウム予稿集（SCIS2015），3B4-4，20-23 January 2015，リーガロイヤルホテル小倉（福岡県北九州市）.
6. 福田 収真，稲垣 賢一，玉田 春昭，``メタプログラミング技法を用いた偽装難読化手法''，2015 年暗号と情報セキュリティシンポジウム予稿集（SCIS2015），3B4-3，20-23 January 2015，リーガロイヤルホテル小倉（福岡県北九州市）.
7. 稲垣 賢一，福田 収真，玉田 春昭，``メソッド呼び出しのフックを用いた動的コールフローグラフ偽装の試み''，2015 年暗号と情報セキュリティシンポジウム予稿集（SCIS2015），3B4-2，20-23 January 2015，リーガロイヤルホテル小倉（福岡県北九州市）.
8. 福田 収真，玉田 春昭，``フックを用いた変数アクセス偽装難読化に向けて''，ソフトウェア工学の基礎 XXI，日本ソフトウェア科学会 FOSE2014 (FOSE2014), pp.81–86, 11-13 December 2014，霧島国際ホテル（鹿児島県霧島市）.
9. 大滝 隆貴，大堂 哲也，玉田 春昭，神崎 雄一郎，門田 暁人，``Java バイトコード命令のオペコード、オペランドを用いた難読化手法のステルシネス評価''，2014 年暗号と情報セキュリティシンポジウム予稿集 (SCIS2014), 2D2-2, 21-24 January 2014，城山観光ホテル（鹿児島県鹿児島市）.
10. 福田 収真，玉田 春昭，``メソッド呼び出し関係隠蔽のための引数順序の入れ替えによる難読化''，2014 年暗号と情報セキュリティシンポジウム予稿集 (SCIS2014), 2D2-3, 21-24 January 2014，城山観光ホテル（鹿児島県鹿児島市）.
11. 福田 収真，玉田 春昭，``Java 7 における API 名隠ぺいのための invokedynamic 命令を用いた難読化の試み''，コンピュータセキュリティシンポジウム 2013 (CSS 2013), pp.1050-1057，No.3D4-2，21-23 October 2013，香川国際会議場サンポートホール高松（香川県高松市）.
12. Kazumasa Fukuda, Haruaki Tamada, ``A Dynamic Birthmark from Analyzing Operand Stack Runtime Behavior to Detect Copied Software,'' In Proc. 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2013), pp.505-510, 1-3 July 2013, Sheraton Princess Kaiulani (Honolulu, Hawaii, U.S.A.).
13. Tetsuya Ohdo, Haruaki Tamada, Yuichiro Kanzaki, Akito Monden, ``An Instruction Folding Method to Prevent Reverse Engineering in Java Platform,'' In Proc. 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2013), pp.517-522, 1-3 July 2013, Sheraton Princess Kaiulani (Honolulu, Hawaii, U.S.A.).

〔その他〕
ホームページ等
1．静的バースマーク抽出・比較ツール
https://github.com/tamada/stigmata/

６．研究組織
(1)研究代表者
玉田　春昭（TAMADA, Haruaki）
京都産業大学・コンピュータ理工学部・准教授
研究者番号：３０４５７１３９