様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２９　年　　６　月　１２　日現在

機関番号：　３３８０３
研究種目：　基盤研究(C)（一般）
研究期間：　2014～2016
課題番号：　２６３３０１６５

研究課題名（和文）組込みシステムに適用可能なアルゴリズム公開型耐タンパーソフトウェアの研究

研究課題名（英文）A study on tamper resistant software that its creation algorithm can be public and be applicable to embedded systems

研究代表者

大石　和臣（OISHI, Kazuomi）

静岡理工科大学・総合情報学部・准教授

研究者番号：２０６３５２１３

交付決定額（研究期間全体）：（直接経費）　　3,400,000円

研究成果の概要（和文）：アルゴリズム公開型の，機能改変困難性を持つ耐タンパーソフトウェア（TRS）を作成する方法に関して，その実用性の客観的な評価を明らかにし，その適用範囲を従来のPCから組込みシステムへと広げて発展させた．
1．組込みシステムに適用可能な耐タンパー化技術として，間接ジャンプとReturn-Oriented Programmingに基づく，データメモリを利用して自己破壊的なタンパー応答を発生する方法を提案した．
2．PCを対象とする実装評価として，PC用のTRS作成ツールを開発して実証実験を行い，取得したデータを用いて実証評価した．想定した通りに動作し，機能改変困難性を有することを実証的に確認した．

研究成果の概要（英文）：A study was conducted on tamper resistant software creation method such that its algorithm can be public and be applicable to embedded systems. Its objective evaluation on practical usefulness became clear and its applicability was extended to not only PC but also embedded systems.
1. As tamper resistance technology applicable to embedded systems, we proposed a method to create tamper resistant software that can generate self destructive tamper response using data memory, based on indirect jump (jump destination is dynamically determined) and ROP (Return-Oriented Programming).
2. As implementation evaluation for personal computer, we developed a tool to create tamper resistant software of PC, conducted experiments using the tool, and evaluated the data obtained through experiments. It was confirmed that the tool worked as intended and created self destructive tamper resistant software has resistance to modification.

研究分野：　情報セキュリティ

キーワード：　耐タンパー　自己書換え　自己インテグリティ検証　自己破壊　機能改変　間接ジャンプ　ROP　ハーバード・アーキテクチャ

１．研究開始当初の背景

コンピュータと通信の発展と普及に伴い，暗号技術等に基づくセキュリティ機能がソフトウェアに実装されユーザのパソコンやスマートフォン等で利用されることが一般的になった．ところが，悪意を持つユーザ（攻撃者）がそれらのソフトウェアを入手し解析することも容易になったので，ソフトウェアの保護は今まで以上に重要な課題になっている．耐タンパーソフトウェアは，実装されたデータやアルゴリズムを不正に読むことが困難である性質（秘密情報守秘性）や改変することが困難である性質（機能改変困難性）を保つことができるソフトウェアであり，信頼できない環境で動作するソフトウェアを保護するための重要な技術である（引用文献[1][2]）．

２．研究の目的

耐タンパーソフトウェアの従来の作成方法は，パソコンのソフトウェアを対象とする自己書換え手法に基づく方法であった．本研究は，組込みシステムに適用可能なアルゴリズム公開型耐タンパーソフトウェアの実現を目的とする．組込みシステムはパソコンと異なるメモリアーキテクチャを持つ．その特徴を踏まえて，（１）組込みシステムに適用可能な耐タンパー化要素技術および作成方法を探索・検討して，仕様と実装を公開しても安全性を損なうことが無いアルゴリズム公開型の耐タンパーソフトウェアを実現することを目指した．また，（２）パソコン向け作成方法の実証評価環境を開発し，その実用性に関する客観的な評価を明らかにすることも目的とした．

３．研究の方法

研究は，（方法１）組込みシステムに適用可能な耐タンパー化技術の提案と，（方法２）パソコンのプログラムを耐タンパーソフトウェアに自動的あるいは半自動的に変換できる環境（ツール）の構築およびそれを用いた実証評価から構成される．方法１では，具体的な組込みシステムを対象とする実験に基づく検討と，理論に基づく探索・検討とを行いながら，既存の方法を発展させるアプローチ A と，自己書換えに限らずに組込みシステムに適する新たな耐タンパー化要素技術を探索するアプローチ B の 2 アプローチを並行して実施した．方法 2 では，パソコンのアセンブリ言語を変換して耐タンパー化を行うソフトウェアを開発した．研究代表者の旧所属の研究グループと連携しながら研究を進め，かつ研究代表者の研究室に所属する学生が卒業研究として取り組み，実験と開発の実務作業を分担して実施した．

４．研究成果

（１）方法１のアプローチ A に関して，①組込みマイコンの調査・実験と，②既存の手法の検討を行った．

①組込みマイコンの調査・実験については，ARM，PIC のマイコン評価ボードを用いて自己書換えコードが実行できるか実験をした．PIC はハーバード・アーキテクチャを採用しており，実行できなかった．ARM に関しては，NXP 社の LPCXpresso Cortex-M0（LPC11C24）と LPCXpresso Cortex-M3（LPC1760）について調査したところ，Cortex-M0（LPC11C24）はフォンノイマン・アーキテクチャを，Cortex-M3（LPC1760）はハーバード・アーキテクチャを採用していることがわかった．Cortex-M0（LPC11C24）で ARM アセンブリ言語を用いて自己書換えコードを実行させたが，実行できなかった．Cortex-M3（LPC1760）では，デフォルトの設定では自己書換えコードを実行すると HardFault が発生し実行できない．しかし，特定のコード（例，関数，オブジェクト）を読み書き可能な領域（RAM）に配置することが可能で，そのコードの中では ARM アセンブリ言語で記述した自己書換えコードを実行できることがわかった．つまり，自己書換えコードの実行可能性は，アーキテクチャーだけではなくマイコンの機種や開発環境に依存することがわかった．

②既存の手法（引用文献[3]，中間コード法）の検討として，論文に記載のサンプルコードを著者から入手し，それを基に検討を行った．中間コード法において既存の耐タンパー化手法として適用可能とされていたが実装実験は行われていなかった引用文献[1][2]の方法（OM 法）について，サンプルコードに OM 法を適用した実装を作り，それが期待通りに動作することを確認した．中間データ法の改良を検討したが，引用文献[3]に挙げられた欠点の解消は容易ではないと結論した．

方法１のアプローチ B に関して，自己書換えコードを用いない方法について検討した結果，データメモリを利用し，間接ジャンプ（ indirect jump ）と Return-Oriented Programming （ROP）に基づいて自己破壊的タンパー応答を生成する耐タンパーソフトウェアを作成する方法を考案した．提案方法は，OM 法に準じ，具体的には，アイデア１：耐タンパー化対象のホストプログラムに含まれる分岐，つまり jump が indirect jump に変換され，その分岐先アドレスはインテグリティ検証で求めた値に基づいて計算される，アイデア 2：データメモリの内容はインテグリティ検証の対象にされ，プログラム（インテグリティ検証を含む）の実行に伴ってデータメモリの内容は変更され，データメモリの既定とは異なる書換えが行われた場合は正常な処理が行われない，アイデア 3：プログラムの中で用いられる定数は別の値に変換（偽装）されてコードに格納され，インテグリティ検証で求めた値に基づいて本来の値に戻されて計算に使用される，
アイデア 4：命令メモリのインテグリティ検

証とデータメモリのインテグリティ検証を相互に関連付ける，を特徴とする．ROP は Intel 社のパソコン用 CPU を対象として研究されているが，組込みマイコンの ARM に関しては ROP が可能であることがわかっているため，提案方法を ARM に対して実装することは可能である．実証実験は今後の課題である．

（2）方法 2 に関して，OM 法を自動化した試験実装とそれを改良したプロトタイプ実装を開発した．どちらも C ソースプログラムをコンパイルする際に耐タンパー化を自動的に適用できるプログラムである．最初に開発した試験実装では，OM 法の中核部分である動的自己書換えに基づく自己破壊的タンパー応答と相互依存型自己インテグリティ検証を自動的に適用することを優先した．試験実装を基にして開発したプロトタイプ実装では，アンチデバッギング技術を追加実装し，相互依存型自己インテグリティ検証にディジタル署名を適用する機能拡張を行って OM 法の基本機能を網羅した．これらにより作成される自己破壊的耐タンパーソフトウェアを SDTRS と呼ぶ．
①試験実装の仕様は以下の通りである．開発プラットフォームは Microsoft Windows 7 Professional，耐タンパー化されるソースプログラム（ホストプログラム）の言語は C，耐タンパー化を行うプログラミング言語は Python，言語処理系は Microsoft Visual Studio 2013 である．OM 法の実現のためには，ホストプログラムのアセンブリプログラム内の被偽装命令とそれを自己書換えする自己書換えルーチンの実行順序が適切になるようにしなければならない．そのための理想的な方法は，ホストプログラムの開始点から終了点までのすべてのパスを含む，アセンブリプログラムの 1 命令を一つのノードとみなした制御フローグラフの構築と解析を含む．これは大きな開発コストを要する．そこで，試験実装ではベーシックブロックとネストしていない if-else 文とネストしていない if 文を含む領域を対象とする局所的な解析に基づいて自己書換えルーチンの挿入位置を決めた．試験実装を開発した結果，同じホストプログラムに対して偽装命令の個数と保護領域の個数を変えた SDTRS を自動的に作成する実験が実施可能となった．ホストプログラムとしてパスワードに基づく MAC（メッセージ認証コード）生成プログラムを耐タンパー化した場合について得たデータのいくつかを以下に示す．図 1 から，プログラムサイズは偽装命令の個数に比例しており，保護領域の個数には依存しない．これは偽装命令が増えると自己書換えルーチンが増えるためだと考えられる．偽装命令 1 個当たりの増加量は約 100 バイトである．図 2 から，保護領域の改ざんの有無を検出するためのハッシュ関数として SAH-256 を用いる場合，偽装命令の個数が一定のときは保護領域の個数

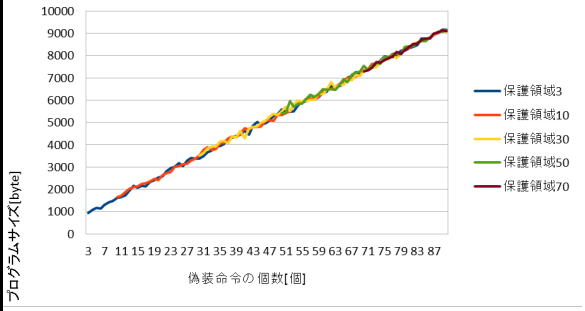

図 1：偽装命令の個数とプログラムサイズ

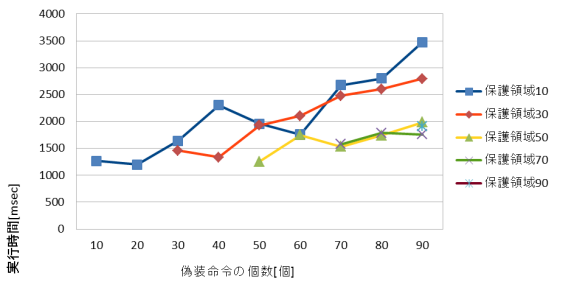

図 2：偽装命令の個数と実行時間（SHA-256）

が少ないほど実行時間が長い傾向を見て取ることができる．これらの実証評価結果は OM 法に関する初めての客観的なデータであり，学術的にも実用的にも意義があり，学会発表⑥⑤で発表した．学会発表⑤は ICSS 研究賞を受賞した．

②プロトタイプ実装に関して，その仕様は試験実装と同じであるが，機能を増やした．試験実装の評価によって判明した欠点を考慮して自己書換えルーチンの挿入位置を決めるアルゴリズムを改良し，for 文，while 文等のループを認識してその内部に自己書換えルーチンを挿入しないようにした．このプロトタイプ実装を用いて実験した結果，OM 法が想定した通りに動作することが確認された．特に，自己インテグリティ検証方式に対する汎用的な攻撃であるページ複製攻撃に対抗することができること，つまり，保護領域内に偽装命令と被偽装命令が混在する状態で自己インテグリティ検証が行われる SDTRS が作成されたことは世界的に初めての成果だと考えられる．試験実装とプロトタイプ実装の両方とも，ホストプログラムの解析と自己書換えルーチンの挿入位置の決定はアセンブリプログラムあるいはソースプログラムの静的解析に基づいており，動的解析に基づく実装の開発が今後の課題の一つである．

（3）情報収集活動として，いくつかの国際会議に参加した．2014 年 8 月に開催された 23rd USENIX Security Symposium では，機能改変を検証する技術として有名な CFI（Control Flow Integrity）の製品向け研究開発等について知見を得た．2016 年 8 月に開催された WhibOx 2016 White-Box

Cryptography and Obfuscation では，鍵内蔵型の暗号プログラムに秘密情報守秘性を持たせる方法である White-Box Cryptography（WBC）と，その理論的な研究である Obfuscation に関する最新の研究開発状況が発表され，動向を把握できた．新しい WBC の提案と WBC に対する新しい攻撃（Differential Computation Analysis），従来は否定的な成果が目立っていた Obfuscation に関して，肯定的かつ汎用性のある成果として indistinguishability Obfuscation（iO）が提案されたことを知った．iO は最近の暗号理論研究において非常に注目を浴びている技術であり，耐タンパーソフトウェアの研究に関連が深く重要であるため，この知見を得たことは今後の研究に有用である．2016 年 10 月に開催された ACM CCS 2016 の併設ワークショップ Second Workshop on Software PROtection (SPRO 2016)では，ヨーロッパにおける耐タンパーソフトウェア開発プロジェクトと思われる ASPIRE consortium の存在と動向を知ることができ，日本より欧米の方が耐タンパーソフトウェアに関する研究開発が活発で進んでいることを知ることができた．

＜引用文献＞

[1] 大石和臣，松本勉，“自己破壊的タンパー応答を発生する耐タンパーソフトウェア”，電子情報通信学会論文誌 A，査読あり，Vol.J94-A，No.3，pp.192-205，2011-03-01.
[2] K. Oishi and T. Matsumoto, “Self destructive tamper response for software protection,” ASIACCS ’11, pp.490-496, 2011.
[3] 吉田 直樹，吉岡 克成，松本 勉，“可変な中間コードとして振る舞うデータ部とそれを実行するインタプリタ部からなる 2 部構成の耐タンパーソフトウェア作成法”，情報処理学会論文誌 Vol．55，No．2，pp.1100-1109，2014.

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０件）

〔学会発表〕（計６件）

①山本 幸二，大石 和臣，櫻井 幸一，須崎 有康，千葉 大紀，松本 晋一，森 達哉，吉岡 克成，”第 25 回 USENIX Security Symposium 調査報告”，2017 年暗号と情報セキュリティシンポジウム（SCIS2017），2A2-4，2017 年 1 月 25 日，ロワジールホテル那覇（沖縄県那覇市）．
②大石 和臣，”WhibOx 2016 White-Box Cryptography and Obfuscation 参加報告”，第 36 回情報通信システムセキュリティ研究会(ICSS)，信学技報 ICSS2016-43，Vol.116，No. 328，pp. 27-29，2016 年 11 月 25 日，情報セキュリティ大学院大学（神奈川県横浜市）．
③大石 和臣，”データメモリを利用する耐タンパーソフトウェア”，情報セキュリティ研究会（ISEC），信学技報 ISEC2016-45，Vol.116，No. 207，pp. 43-48，2016 年 9 月 2 日，機械振興会館（東京都港区）．
④松本 晋一，大石 和臣，須崎 有康，”第 23 回 USENIX Security Symposium 参加報告”，第 30 回情報通信システムセキュリティ研究会(ICSS)，信学技報 ICSS2015-8，Vol.115，No. 81，pp. 39-44，2015 年 6 月 11 日，九州工業大学百周年中村記念館（福岡県北九州市）．
⑤大石 和臣，吉田 直樹，渡邉 直紀，坂本 純一，松本 勉，”自己破壊的耐タンパーソフトウェアの試験実装と評価”，第 29 回情報通信システムセキュリティ研究会（ICSS），信学技報 ICSS2014-99，Vol. 114，No. 489，pp. 217-222，2015 年 3 月 4 日，名桜大学（沖縄県名護市）．
⑥大石 和臣，吉田 直樹，渡邉 直紀，坂本 純一，松本 勉，”自己破壊的耐タンパーソフトウェアの試験実装”，2015 年暗号と情報セキュリティシンポジウム(SCIS2015)，2B1-4，2015 年 1 月 21 日，リーガロイヤルホテル小倉（福岡県北九州市小倉北区）．

〔図書〕（計０件）

〔産業財産権〕
○出願状況（計０件）
○取得状況（計０件）

〔その他〕
ホームページ等

６．研究組織

(1)研究代表者
大石　和臣（OISHI，Kazuomi）
静岡理工科大学・総合情報学部・准教授
研究者番号：２０６３５２１３

(2)研究分担者
（　　）
研究者番号：

(3)連携研究者
（　　）
研究者番号：

(4)研究協力者
吉田 直樹（YOSHIDA，Naoki）
横浜国立大学・環境情報学府・大学院生

渡邉 直紀（WATANABE，Naoki）
横浜国立大学・環境情報学府・大学院生

坂本 純一（SAKAMOTO, Jun’ichi）
横浜国立大学・環境情報学府・大学院生

松本 勉（MATSUMOTO, Tsutomu）
横浜国立大学・環境情報研究院・教授