様式 C-19

科学研究費補助金研究成果報告書

平成２２年　６月　７日現在

研究種目：基盤研究（C）
研究期間：2007～2009
課題番号：１９５０００５６
研究課題名（和文）　プログラム難読化適用のフレームワーク

研究課題名（英文）　A Framework for Software Obfuscation

研究代表者
門田　暁人（MONDEN　AKITO）
奈良先端科学技術大学院大学・情報科学研究科・准教授
研究者番号：８０３１１７８６

研究成果の概要（和文）：本研究では，プログラム難読化法を適材適所に用いてプログラム中の秘密情報を隠蔽するためのフレームワークを提案した．評価実験において，既存の 39 種類の自動難読化ツールを適用した場合と，難読化フレームワークを用いて難読化を行った場合を比較した結果，既存ツールではいずれも秘密情報を十分に隠蔽できていないのに対して，提案フレームワークでは秘密情報を隠蔽できていることを確認した．

研究成果の概要（英文）：To hide secrets in a computer program, this research proposes a framework to apply existing obfuscation methods to a given program. The result of an experimental evaluation showed that secret information and its clues were all hidden by the proposed framework while conventional obfuscation tools could not thoroughly hide the secrets.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	1,200,000	360,000	1,560,000
２００８年度	1,000,000	300,000	1,300,000
２００９年度	1,000,000	300,000	1,300,000
年度			
年度			
総　計	3,200,000	960,000	4,160,000

研究分野：総合領域
科研費の分科・細目：情報学・計算機システム・ネットワーク
キーワード：ソフトウェア保護，セキュリティ，プログラムの難読化，ソフトウェア開発，ゴール木，攻撃モデル，秘密情報

１．研究開始当初の背景

（１）近年，エンドユーザによるソフトウェアの解析を防止する技術への要求がこれまで以上に高まっている．例えば，デジタルコンテンツの再生，録画，移動等を行うソフトウェアは，内部に含まれるコンテンツ復号鍵の漏洩を防止することが求められる．また，携帯電話，ゲーム機，セットトップボックス等の組み込みソフトウェアもそれぞれ固有の秘密情報を含むため，解析を防ぐ必要に迫られている．
（２）そのために，従来，解析を妨げるためのプログラム難読化法が数多く提案されて

いる．例えば，名前の変換，制御構造の変換，データの変換，データ構造の変換，演算子の変換などである．
（3）これらの難読化法の多くは，汎用性が高く，それぞれプログラムのある側面（制御構造など）の解析を困難にできる．しかし，本当に保護したい秘密情報（復号鍵やサブルーチンなど）の解析防止にどの程度効果があるのかについて，客観的な評価は困難であった．難読化により制御構造などを複雑にすることと，攻撃者に秘密情報を知られにくくすることの間には大きなギャップがあったためである．

以上のことから，攻撃者の視点に立ち，攻撃方法を整理すること，および，攻撃者の行動を妨げるように難読化手法を適材適所に用いるためのフレームワークが必要である．

２．研究の目的

本研究の目的は，攻撃者の行動を整理し，その行動を妨げるように難読化手法を適用するためのフレームワークを確立することである．

３．研究の方法

（1）まず，攻撃者の行動パターンの調査，および，攻撃者の能力モデルとして記述するべき情報を整理する．例えば，攻撃対象に対する知識として，ソフトウェアの内部で用いられている暗号方式やアルゴリズムといった，ソフトウェアの仕様に関する知識の記述が必要となる．次に，攻撃者がシステムから観測できる情報を記述する必要がある．さらには，攻撃者がシステムに対して行うことの出来る行動を記述する必要がある．
（2）攻撃者のゴール木の記述方法を決定する．ゴール木は，攻撃者が獲得しようとする秘密情報が含まれるプログラムを対象として記述される．記述のフォーマットとして，FTA（Fault Tree Analysis，故障の木解析）で用いられているフォールト木の記述フォーマットを応用することを検討する．FTA は，ある故障の事象から，その原因を追跡し，対処法を見つけるための手法であり，本研究におけるゴール指向分析と類似する側面が多い．FTA におけるフォールト木は，事象，否展開事象，AND ゲート，OR ゲート，制約ゲートなどの基本要素を用いて記述される．本研究におけるゴール木においても，サブゴールを AND ゲートや OR ゲートにより接続したり，サブゴールに制約を付加したり，情報不足のためにサブゴールをそれ以上分解できない否展開事象を記述することが有用と考えられる．
（3）ゴール試行分析（ゴール分解）のガイドラインを決定する．ゴール指向分析では，ゴールおよびサブゴールを，より詳細なサブゴールへと分解していく．ここでは，サブゴールとして記述すべき内容を整理する．本研究におけるサブゴールは，秘密情報の発見に必要な「何らかの手がかりを発見する作業」であり，サブゴールの記述として，①手がかりの内容，②能力モデルと手がかりの関係（攻撃者のどの知識や能力に基づいて手がかりを探すか），③他のサブゴールとの関連，などを記述する必要があると考えられる．
（4）提案フレームワークの評価実験を行う．評価実験では，秘密情報を含むプログラムについて，ゴールの記述，ゴール指向分析，難読化法のマッピングを行う．そして，実際にプログラムの難読化を行い，その安全性を評価する．

４．研究成果

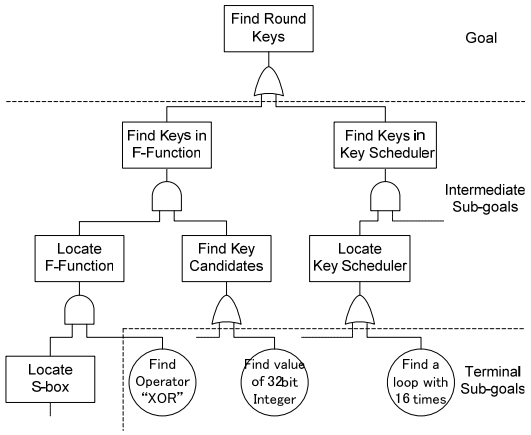
（1）現実世界の攻撃者（ハッカー，クラッカー）の行動や技術について調査を行い，調査結果に基づいて，攻撃者の能力（システムの知識，システムの観測，システムの制御）の整理・記述方法を提案した．また，鍵内蔵型プログラムにおけるケーススタディにおいて，攻撃者の能力を整理・記述した．下記に例（抜粋）を示す．

攻撃者モデルの例（抜粋）

攻撃者のゴール：
・ラウンド鍵を発見する．
システムの知識：
・10 回の繰り返しが存在し，それぞれにラウンド鍵$k_1,\ldots,k_{10}$が存在する．
・各ラウンド鍵の鍵長は 32 ビットである．
・入力のブロック長は 64 ビットであり，上位 32 ビット L と下位 32 ビット R に分割される．
・Feistel 関数には add 演算が含まれる．
・・・
システムの観測と制御：
・攻撃者は，Java の逆アセンブラや逆コンパイラを静的解析に使用する．
・攻撃者は，jasmin 形式の逆アセンブルコードを改造して再アセンブルできる．
・攻撃者は，IDA Pro などのデバッガを用いて動的解析を行うことができる．

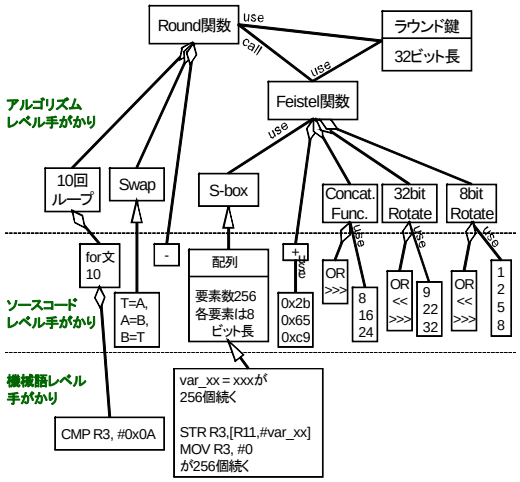
（2）FTA を応用して攻撃のゴール木（AND-OR 木）を記述する方法を提案した．ゴール木には 3 つの種類のゴール（ルートゴール，中間

ゴール，末端ゴール）があり，AND もしくは OR ゲートによって接続される．あるゴールについて，複数の下位のゴールが AND ゲートにより接続されている場合，攻撃者が全ての下位ゴールを達成する必要があることを意味する．一方，OR ゲートによる接続の場合は，攻撃者は下位ゴールの少なくとも１つを達成する必要がある．ゴール木の例を下記に示す．



（３）提案したプログラム難読化適用のフレームワークについて，ケーススタディを通して評価・改良を行った．ケーススタディでは，DRM(Digital Rights Management)ソフトウェアにおけるラウンド鍵の隠蔽を題材とした．攻撃のゴール木を作成するために，「ラウンド鍵の発見」をルートゴールとしてゴール分解を行った．まず，ルートゴールのサブゴールとして「ラウンド鍵を表す定数値を探す」，「ラウンド鍵の値を保持する変数を探す」という二つを定義し，トップダウンアプローチによってそれらをさらに小さいサブゴールへと分解した．一方，ゴール木の作成には，ボトムアップアプローチも必要なことが分かった．そこで，攻撃の手がかりとなる情報（DRM のアルゴリズムにおいて特徴のある定数値や演算子など）に着目し，それらから抽象度の高いサブゴールを導出することにより，ゴール木を完成させた．これらの知見に基づく改良方法として，トップダウンとボトムアップの２つのアプローチによって攻撃のゴール木を作成する手順を提案した．さらに，作成されたゴール木に基づいて，ゴール木の作成を妨げる４つの難読化法を選定し，DRM ソフトウェアの難読化を行った．その結果として，攻撃が困難となっていることを確認した．

（４）実用システムへの適用を想定し，難読化フレームワークにおけるゴール木の構築について，より実用的なガイドラインを提案した．また，提案したガイドラインを実用システムに適用し，その効果を評価した．提案したガイドラインでは，秘密情報とその手がかりの関係，および，手がかり間の関係を，①部分-全体，②抽象-具体，③その他，の 3 つに分類し，Unified Modeling Language のクラス図の記法により表現する．クラス図の作成にあたっては，手がりを 3 つの抽象レベル（アルゴリズム，ソースコード，機械語）に分けて記述する．これによって，手がかり間の関係が分かりやすくなるとともに，異なる抽象レベルの手がかりを網羅的に列挙しやすくなった．C2(Cryptomeria Cipher)暗号プログラムにおいてラウンド鍵を隠蔽するケースを想定し，ガイドラインに基づくゴール木の作成を行った．その結果，従来の方法では抜けていた手がかりを列挙できており，また，クラス図の記法によって手がかり間の関係をより明確にできていることを確認した．手掛かり間の関係を記述した例を下記に示す．



（５）攻撃の手掛かりを隠蔽するためには，既存の難読化法に加えて，より強力かつ利便性の高い難読化法が必要なことから，動的名前解決を用いた名前難読化法，及び，拡張プログラムカムフラージュ法を提案した．動的名前解決は，プログラム中の名前使用部分をあらかじめ暗号化しておき，実行時に名前を復号して当該処理を実行する方式である．名前使用部分は典型的な攻撃ターゲットとなるのでその防御手段が必要である．提案手法では，オブジェクト指向言語のリフレクション機構を用いて，クラスの参照，メソッド呼び出し，フィールドの参照・代入に現れる任意の名前を動的解決する方法を実現した．提案手法を Java プログラム用に実装し評価実験を行った．ある実用プログラムへの適用では，約 4 倍の性能劣化でプログラム中のすべてのクラス名，メソッド名，フィールド名を難読化できることがわかった．また，拡張プログラムカムフラージュ法では，攻撃者に知

られたくない命令の内容変更や削除を高級言語のレベルで具体的に指定可能とすることで，攻撃のターゲットを直接的に隠蔽することを可能とした．ゴール木の構築によって攻撃ターゲットを明確化した後に，拡張プログラムカムフラージュ法によってターゲットを隠蔽することで，攻撃を困難にすることが可能となった．

（６）提案フレームワークの有効性を評価するために，既存の難読化ツールにより自動難読化を行った場合と，難読化フレームワークにより難読化すべき場所を特定して人手により難読化を行った場合の比較を行った．その結果，39 種類の既存手法による自動難読化を適用した後で攻撃を行った場合には，いずれの手法を用いても秘密情報を十分に隠蔽できないことが分かった．一方，提案フレームワークを用いた場合には，秘密情報を攻撃から保護できており，難読化による実行効率の低下も軽微であることを確認した．以上のことから，プログラム中の秘密情報を隠蔽するためには，自動化できる難読化法に頼るだけでは不十分であり，自動化できない（手作業による）難読化が不可欠であり，そのためには提案フレームワークが有用であることが分かった．

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 11 件）

(1) 武田隆之，牛窓朋義，山内寛己，門田暁人，松本健一，コーディングスタイルに基づく特徴量とソースコード盗用との関係の分析，情報処理学会報告，ソフトウェア工学研究会，No.167，2010，査読無

(2) 牛窓朋義，門田暁人，玉田春昭，松本健一，使用クラスに基づくソフトウェアの機能面からの分類，電子情報通信学会技術報告，Vol.109，No.170，31-36，2009，査読無

(3) 吉村巧朗，亀井靖高，上野秀剛，門田暁人，松本健一，ブレークポイント使用履歴に基づくデバッグ行動の分析，電子情報通信学会技術報告，Vol.109，No.307，pp.85-90，2009，査読無

(4) 神崎雄一郎，門田暁人 実行時間差に着目したコードの隠ぺい方法，第８回情報科学技術フォーラム講演論文集，第 1 分冊，361-364，2009，査読無

(5) 岡原聖，真鍋雄貴，山内寛己，門田暁人，松本健一，井上克郎，コードクローンの長さに基づくプログラム盗用確率の実験的算出，電子情報通信学会技術報告，No.SS2008-40，7-11，2008，査読無

(6) Yuichiro Kanzaki, Akito Monden, Masahide Nakamura, Ken-ichi Matsumoto, Program Camouflage: A Systematic Instruction Hiding Method for Protecting Secrets, In Proc. World Congress on Science, Engineering and Technology, Vol. 33, 557-563, 2008, 査読有

(7) Hiroki Yamauchi, Akito Monden, Masahide Nakamura, Haruaki Tamada, Yuichiro Kanzaki, and Ken-ichi Matsumoto, A Goal-Oriented Approach to Software Obfuscation, International Journal of Computer Science and Network Security, Vol.8, No.9, 59-71, 2008, 査読有

(8) Haruaki Tamada, Masahide Nakamura, Akito Monden, Ken-ichi Matsumoto, Introducing dynamic name resolution mechanism for obfuscating system-defined names in programs, Proc. IASTED International Conference on Software Engineering, 125-130, 2008, 査読有

(9) 山内寛己，門田暁人，松本健一，高級言語によって偽装内容を指定できる拡張プログラムカムフラージュ法，Information Science Technical Report, NAIST-IS-TR2009007，奈良先端科学技術大学院大学，2007，査読無

(10) 神崎雄一郎，門田暁人，中村匡秀，松本健一，高級言語によって偽装内容を指定できる拡張プログラムカムフラージュ法，Information Science Technical Report, NAIST-IS-TR2007015，奈良先端科学技術大学院大学，2007，査読無

(11) 玉田春昭，中村匡秀，門田暁人，松本健一，APIライブラリ名隠蔽のための動的名前解決を用いた名前難読化，電子情報通信学会論文誌D，Vol. J90-D，No.10，2723-2735，2007，査読有

〔学会発表〕（計 1 件）

(1) 門田暁人，ソフトウェア工学の挑戦：情報セキュリティの強化に向けて，ソフトウェアエンジニアリングシンポジウム 2008 クロージングパネル，2008.9.3，東京

〔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〕

６．研究組織

(1)研究代表者

門田　暁人（MONDEN　AKITO）
奈良先端科学技術大学院大学・情報科学研究科・准教授
研究者番号：８０３１１７８６

(2)研究分担者

（　　）

研究者番号：

(3)連携研究者

中村　匡秀（NAKAMURA　MASAHIDE）
神戸大学・工学研究科・准教授
研究者番号：３０３２４８５９

玉田　春昭（TAMADA　HARUAKI）
京都産業大学・コンピュータ理工学部・助教
研究者番号：３０４５７１３９