1L-02

コード断片の不自然さの比較による保護機構の発見困難さの評価

松田 篤和 † 神崎 雄一郎 ‡ 門田 暁人 ††

† 熊本高等専門学校 電子情報システム工学専攻 ‡ 熊本高等専門学校 人間情報システム工学科

†† 奈良先端科学技術大学院大学 情報科学研究科

1 はじめに

ソフトウェア保護機構の発見困難さは，保護機構の強さ (攻撃者の目的達成に必要なコスト) を左右する重要な要素であると言われており [1]，その評価方法が求められている．そこで，本研究では，文献 [2] で提案されているアセンブリコードの「不自然さ」を測定するメトリクスを用いて，プログラムに適用された保護機構の発見困難さを評価する一方法を提案する．具体的には，プログラムを一定長に分割して得られる各コード断片の不自然さの値を N-gram モデルによって測定し，比較・分析することで発見困難さを評価する．文献 [2] では，保護適用による特定のルーチンの不自然さの変化を議論しているが，本研究ではプログラム全体を一定長に分割して得られる各コード断片の不自然さから，適用された保護機構の発見困難さを議論する．

ケーススタディでは，保護対象のプログラムに，命令のカムフラージュ [3]，AES 暗号を用いた暗号化 [2]，整数演算の複雑化 [4] といった既存の保護方法を適用し，提案方法を用いて保護機構の発見困難さの評価を行った．その結果，命令のカムフラージュ，AES 暗号を用いた暗号化については，攻撃者に発見されやすい傾向がある一方，整数演算の複雑化については，攻撃者に発見されにくい傾向があることがわかった．

2 コードの不自然さの定義

コード断片の不自然さを，文献 [2] で提案されている方法と同様，次のように定義する．まず，N-gram モデルによって，与えられたコードを構成する命令列 $i_1^n = i_1 i_2 ... i_n$ の生成確率 $P(i_1^n)$ を，次式のように近似する．

$$P(i_1^n) \approx \prod_{k=1}^{n} P(i_k \mid i_{k-N+1}^{k-1}) \quad (1)$$

すなわち，k 番目の命令 i_k の生成確率が，直前の $(N-1)$ 命令 $i_{k-N+1} ... i_{k-2} i_{k-1}$ にのみ依存すると考える．また，命令列 $i_1^n = i_1 i_2 ... i_n$ で構成されるコード C の不自然さ $A(C)$ を，生成確率 $P(i_1^n)$ を用いて次のように定義する．

$$Artificiality\ A(C) = -\log_{10} P(i_1^n) \quad (2)$$

$A(C)$ の値が大きいほど，アセンブリ言語としてもっともらしくなく，C は不自然であると考える．

3 発見困難さの評価手順

発見困難さの評価手順の概略を図 1 に示す．まず，PE 形式として与えられた評価対象のプログラムの実行可能ファイルを逆アセンブルし，アセンブリコード C を取得する．次に，C を N-gram 単位に分割し，コード断片を取得する．図 1 の例では，C を 3-gram 単位で分割すると，C は 7 命令含んでいるため C_1, C_2, C_3, C_4 の 4 つのコード断片が得られる．次に，各コード断片に対して，文献 [2] で提案されている方法によって，各コード断片の不自然さの値を測定する．最後に，測定して得られた各不自然さの値をグラフを用いて可視化し，不自然さの平均や標準偏差を用いて保護機構の発見困難さの評価を行う．

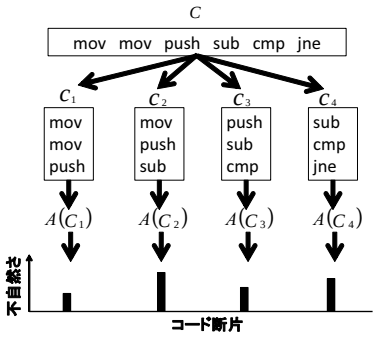


図 1 評価方法の概略

4 ケーススタディ

3 章で述べた発見困難さの評価手順に従って既存の保護機構の発見困難さの評価を行った．コードは 3-gram 単位で分割した．保護の適用対象としては，文献 [1] で示されている DRM のプログラムを評価対象とした．対象プログラムは，`main` 関数と `play` 関数で構成されており，ライセンスチェックなどの秘密情報を含む `play` 関数を保護の対象とした．適用した保護方法は，命令のカムフラージュ [3]，AES 暗号を用いた暗号化 [2]，整数演算の複雑化 [4] の 3 種類である．AES 暗号を用いた暗号化については，`main` 関数と `play` 関数に加えて，復号のために `getCode` 関数，`cryptography` 関数が追加されている．以後，元来の DRM プログラムのコードを C_o，C_o に命令のカムフラージュを適用したものを C_{camf}，C_o に AES 暗号を用いた暗号化を適用したもの

Evaluating the Stealth of Protected Code Based on the Artificiality of Its Code Fragments

Atsuto Matsuda†, Yuichiro Kanzaki‡, Akito Monden††

†Advanced Course of Electronics and Information System Engineering, Kumamoto National College of Technology

‡Department of Human-Oriented Information Systems Engineering, Kumamoto National College of Technology

††Graduate School of Information Science, Nara Institute of Science and Technology

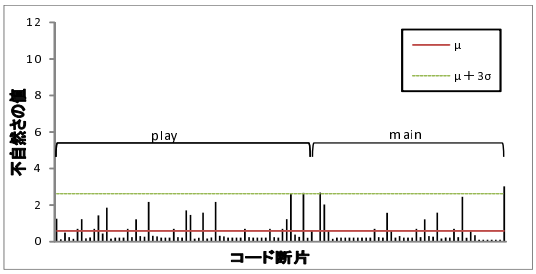
を C_{aes}，C_o に整数演算の複雑化を適用したものを C_{ea} と表す．

C_o，C_{camf}，C_{aes}，C_{ea} について，3-gram 単位で分割した各コード断片の不自然さの測定結果を図 2 に示す．縦軸は不自然さの値を示し，コードの出現順に並べている．ただし C_{aes} は，play 関数内にコーパスに存在しない命令が 2 つある．それらの命令は，生成確率を計算できないため，不自然さの値を便宜上 20 とし，グラフ (図 2(c)) 内の該当データの先頭に印をつけている．また，C_{camf} のグラフ (図 2(b)) は，議論のために，偽の命令を含むコード断片のデータの先頭に印をつけている．また，図 2 の各グラフには，各コードにおける不自然さの値の平均値 μ と，$\mu + 3\sigma$ (σ は標準偏差) を示している．ここでは，$\mu + 3\sigma$ を超える不自然さを持つものは,「プログラム内で目立つ」コード断片と考える．

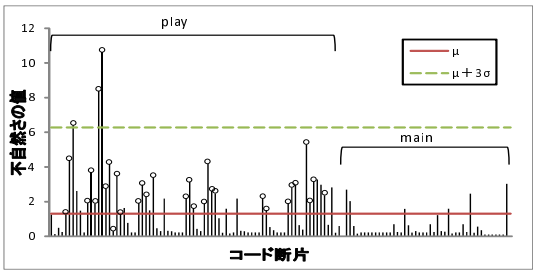
C_o は，プログラム内で目立つ断片が少し存在するものの，突出して不自然なコード断片は存在しない．C_{camf} は，play 関数内で保護を適用した部分のコード断片が C_o の play 関数と比べて不自然さの値が大きくなっており，保護の適用によって play 関数が全体的に不自然になっていることがわかる．特に，偽の命令に置き換えられたコード断片は突出して不自然になっており，保護機構は攻撃者に発見されやすいと評価できる．C_{aes} は，暗号化されている play 関数の大部分の不自然さが C_o の play 関数と比べて大きくなっており，保護の適用によって play 関数は全体的に不自然になっていることがわかる．また，C_{aes} の play 関数はプログラム内でも目立っており，保護機構は攻撃者に発見されやすいと評価できる．C_{ea} は，C_o と比べて不自然さの値は全体的に大きくなっているものの，大きな差異はなく，プログラム内で目立つコード断片も少ないことがわかる．そのため，元来のコードの自然さを維持しており，保護機構は発見されにくいと評価できる．整数演算の複雑化はコード断片の意味的なつながりを破壊せずに行われるため，保護を適用してもコード断片の自然さが維持されたものと考えられる．

5 おわりに

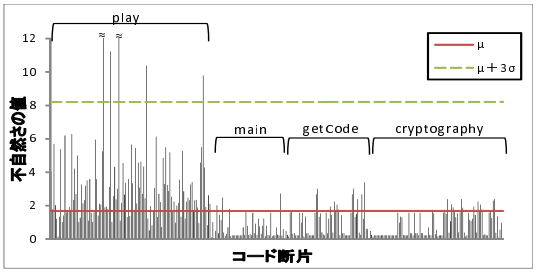
本論文では，プログラムを構成するコード断片の不自然さに基づいて，プログラムに適用された保護機構の発見困難さを評価する一方法を提案した．ケーススタディでは，既存の保護方法が適用されたプログラムに対して保護機構の発見困難さを評価する実験を行った．その結果から，命令のカムフラージュと AES を用いた暗号化は攻撃者に発見されやすく，整数演算の複雑化は発見されにくい傾向があることがわかった．今後の課題として，オペランドも含めた命令列を対象にして評価することを考えている．



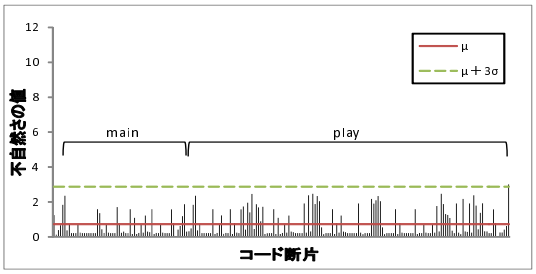
(a) C_o の不自然さの分布



(b) C_{camf} の不自然さの分布



(c) C_{aes} の不自然さの分布



(d) C_{ea} の不自然さの分布

図 2　各評価対象コードの結果

参考文献

[1] C. Collberg. and J. Nagra.: Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Program Protection, Addison-Wesley Professional, 2009.

[2] 神崎雄一郎，尾上栄浩，門田暁人：コードの「不自然さ」に基づくソフトウェア保護のステルシネス評価，情報処理学会論文誌 Vol.55，No.2，pp.1005-1015，February 2014.

[3] 神崎雄一郎，門田暁人，中村匡秀，松本健一：命令のカムフラージュによるソフトウェア保護方法，電子情報通信学会論文誌，Vol.J87-A，No.6，pp.755-767，June 2004.

[4] C. Collberg. The tigress diversifing C virtualizer. http://tigress.cs.arizona.edu/.