SCIS 2022 2022 Symposium on Cryptography and Information Security
Osaka, Japan & Online, Jan. 18 – 21, 2022
The Institute of Electronics, Information and Communication Engineers

準同型暗号を用いたOpaque Predicateの提案
Constructing an Opaque Predicate using Homomorphic Encryption

平能 耀介 * 大瀧 保広 *
Yohsuke Hirano Yasuhiro Ohtaki

キーワード 難読化　準同型暗号　静的解析

あらまし

プログラムの難読化は、ソフトウェアのリバースエンジニアリングを困難にする手段として知られている。難読化手法の一つである Opaque Predicate は、コンパイル時に真か偽どちらを取るかすでに決まっている述語であり [1]、難読な条件分岐や到達不能コードを作り出すことにより、主に制御フローを複雑にすることで解析を困難にするものである。しかし、Opaque Predicate はすでに多くの検出手法が提案されており、実装方法によっては簡単に除去されることが知られている。[2] また、近年マルウェアなどのソフトウェア解析において CPU エミュレータやシンボリック実行を組み合わせたバイナリ解析フレームワークが利用されており [3, 4, 5]、フレームワークに含まれる機能や SMT ソルバを使用すると、逆アセンブラのみを用いた静的解析と比較して Opaque Predicate の検出が容易になる。このことから、現在では Opaque Predicate のような制御フローを複雑にする難読化手法の効果は低下している。本研究では、準同型暗号を用いた Opaque Predicate を提案する。提案手法では、Opaque Predicate に利用される数式の変数と定数を準同型暗号を用いて暗号化し、そのまま演算を行い、復号することによって、静的解析におけるパターンマッチングや SMT ソルバを用いた自動証明による検出が困難となる。提案手法で難読化を行い、実行時のオーバーヘッドを計測した結果、実行速度の低下が許容される場面においては有効な手法であることが確認できた。また、難読化の気づかれやすさの観点から考察した結果、今後はコードの「不自然さ」[6] のような指標に基づいた評価や準同型暗号そのものに対する難読化などを検討する必要がある。

* 茨城大学, 茨城県日立市中成沢町 4 丁目 12-1, Ibaraki University,4-12-1, Nakanarusawa-Cho,Hitachi,Ibaraki,316-8511,Japan

参考文献

[1] Genevieve Arboit. A method for watermarking java programs via opaque predicates.ICECR-5.2002.

[2] Lukas Zobernig, Steven D. Galbraith, and Giovanni Russello.When Are Opaque Predicates Useful?.2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering. IEEE,2019.

[3] Jonathan Salwan,Sebastien Bardin ,and Marie-Laure Potet.Symbolic deobfuscation:from virtualized code back to the original?.DIMVA 2018.

[4] Mingyue Liang, Zhoujun Li, Qiang Zeng, and Zhejun Fang.Deobfuscation of Virtualization-obfuscated Code through Symbolic Execution and Compilation Optimization.ICICS2017.

[5] Tirenna, Pietro Francesco. Techniques for malware analysis based on symbolic execution. Diss. Politecnico di Torino, 2020. https://webthesis.biblio.polito.it/15305/1/tesi.pdf.

[6] 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機構のステルシネス評価. 情報処理学会論文誌 Vol55 1005-1015. 2014.