# 2021 年度 修士論文発表 梗概

電気電子工学専攻

| 氏　名 | 米澤颯真 | 指導教員 | 藤枝直輝 |
|---|---|---|---|
| 題　目 | RISC-V における命令レジスタファイルによる命令難読化の検討 | | |

## 1　背景と目的

近年，ソフトウェアを解析・盗用・改ざんなどといった攻撃から保護することの重要性が高まっている。ソフトウェアで保護を行う方法に自己暗号化と難読化がある [1] が，安全性に限界があり，オーバーヘッドも大きいという問題がある。そのため，リソースに限りのある組込みシステムではハードウェアを用いた保護が好まれる。ハードウェアによる保護手法の 1 つに，Instruction Register File (IRF) を用いた命令レベルの難読化手法がある [2]。プログラム中のよく使われる命令は，IRF を参照する命令へとあらかじめ変換される。この命令がフェッチされると，命令に含まれているアドレスに対応する IRF 上の命令がデコーダに送られ，元の命令が復元される。IRF は元々プロセッサの性能向上を目的としていたが，命令難読化の手法としても利用できる。

本研究では IRF を用いた命令難読化における，RISC-V に適した実装方法について検討する。先行研究では固定長命令をもつアーキテクチャを検討していた。一方，RISC-V は圧縮命令セットをもち，32 ビット通常命令と 16 ビット圧縮命令がプログラム中で混在可能である。そのため本研究では，圧縮命令を含む場合の命令の出現頻度の調査と，圧縮命令に対応した IRF の試作を行う。

## 2　カスタム命令の定義

IRF を参照する命令は，通常命令でカスタム命令用に予約されているオペコードを使用し，元の圧縮命令または通常命令と 1 対 1 で対応する。命令は，7 ビットのオペコード，1 ビットの命令長，IRF のアドレスから構成される。IRF のアドレスは，元の命令の長さに応じて 8 または 24 ビットとなる。

## 3　命令の出現頻度調査

先行研究 [2] に準じた方法で，命令の使用頻度を集計し，それをもとに IRF を構成したときの難読化の度合いを評価する。命令セットは RV64GC とし，Spike シミュレータにより命令の実行ログを取得した。ベンチマークとして MiBench から 8 つのプログラムを用いた。IRF のエントリ数ごとの難読化指標の評価結果から，先行研究 [2] と比較して，同じ IRF のエントリ数でも IRF から命令が実行される割合が高いことがわかった。実行された命令のうち平均 63.3% が圧縮命令であり，最も頻繁に実行される命令も多くが圧縮命令であった。その影響で，先行研究の結果 [2] と比べ，IRF のエントリ数が少ない場合でも，多くの命令が IRF から実行できる可能性があることがわかった。

## 4　ハードウェア実装のプロトタイプ

既存の RISC-V プロセッサである CVA6 に対し，IRF のプロトタイプを実装する。IRF は，CVA6 の命令キューと圧縮デコーダとの間の RAM として実装した。また，IRF を参照する命令が圧縮命令に対応する場合，命令長が 32 ビットではなく 16 ビットになる。そのため，プログラムカウンタを 4 ではなく 2 加算するように変更を施した。また，命令リライナーにも IRF を参照する命令が圧縮命令に対応しているかの判別を行うように変更を施した。動作検証においては，一部の命令を IRF を参照する命令に変更したプログラムを用意し，オリジナルの CVA6 と上記の変更を加えた CVA6 とでそれぞれ実行した。CVA6 は Digilent Genesys2 FPGA ボード上で動作させた。検証の結果，用意したプログラムはオリジナルの CVA6 では動作せず，変更後の CVA6 でのみ動作した。

## 5　結論

本研究では，RISC-V における命令の出現頻度の調査と，IRF のプロトタイプのハードウェア実装を行った。その結果，圧縮命令を含む場合でも，固定長命令のアーキテクチャの場合と少なくとも同程度の効率で IRF による命令難読化が実現できる見込みが得られた。

## 参考文献


[1] C.Collberg, C.Thomborson, and D.Low. A Taxonomy of Obfuscating Transformations. Technical report, Department of Computer Science The University of Auckland, 1997.

[2] N.Fujieda, T.Tanaka, S.Ichikawa. Design and Implementation of Instruction Indirection for Embedded Software Obfuscation. Microprocessor and Microsystems, vol. 45, part A, pp.115–128, 2016.