2023 年 12 月 1 日

電気・電子情報工学専攻	学籍番号	M203216
申請者氏名	小倉 幹也	

指導教員氏名	市川 周一

論　文　要　旨（修士）

論文題目	Obfuscator-LLVM と Bambu を用いたハードウェア難読化

制御システムや組込みシステムのソフトウェアにはノウハウ等の秘匿すべき情報が含まれており，システム内部の知的財産保護は喫緊の課題となっている．ソフトウェアは複製や解析が容易であるため，ソフトウェアからハードウェア記述言語を自動生成する技術 (高位合成) を用いて，ソフトウェアの一部をハードウェア化し知的財産を保護する手法が研究されている．ハードウェア化に加えて，ハードウェア難読化のような秘匿化技術を併用することもできる．難読化とは機能を変えずに内部構造を複雑化させて解析を困難にする技術である．

山田ら (2020) は，LLVM ベースの難読化ツール Obfuscator-LLVM (OLLVM) とオープンソース高位合成ツール LegUp を組み合わせてハードウェアを難読化した．LLVM はコンパイラ基盤であり，高位合成ツールの開発等に近年広く用いられている．難読化されたコンパイラ中間表現 (LLVM-IR) を高位合成することで，ハードウェア難読化プロセスを自動化し知的財産保護の労力を削減できる．しかしその後，LegUp の商用化により研究利用の継続が困難となった．そこで本研究では，OLLVM とオープンソース高位合成ツール Bambu を新たに組み合わせハードウェア難読化を行う．OLLVM は LLVM ミドルエンドに実装されており，LLVM を利用した様々な高位合成ツールでもハードウェア難読化に利用できる可能性がある．

評価には，原ら (2008) による C 言語ベースの高位合成用ベンチマークである CHStone を用いる．OLLVM は難読化機能として偽の制御フロー (BCF)，制御フロー平坦化 (CFF)，命令置換 (ISub) を提供しており，まずそれぞれを単独で適用した場合の評価を行った．ベンチマークプログラムを難読化・高位合成した結果について論理合成とシミュレーションを行い，生成された難読化ハードウェアの論理規模と実行時間を求めた．Lookup table (LUT) 使用数と実行時間について幾何平均を求め，提案手法と山田らの手法を比較した．提案手法は既存手法と比較し LUT 使用数が 58.5% 多く難読性が高いと考えられる．実行時間について非難読化時は既存手法が提案手法より 13.6% 高速であるが，難読化時は提案手法が既存手法より 7.3% 高速である．

次に，山田らが達成できなかった OLLVM の難読化を複数適用したハードウェアの評価を行った．評価した組み合わせは Banescu ら (2016) が難読性が高いとした BCF-CFF，CFF-BCF，CFF-ISub である．CFF 難読化単独の結果と LUT 使用数を比較し，BCF-CFF 難読化は 45.1% 増えているが，CFF-BCF 難読化は 0.3%，CFF-ISub 難読化は 0.7% と微増であった．実行時間は CFF 難読化単独と比較して，BCF-CFF 難読化で 74.3%，CFF-BCF 難読化で 20.6%，CFF-ISub 難読化で 0.4% 低速である．この結果について単純なプログラムで難読化された LLVM-IR を生成し考察した．入力した LLVM-IR の switch 命令の分岐数は LUT 使用数増加に大きく寄与し，入力した LLVM-IR の難読性は LUT 使用数増加への寄与が小さいと考えられる．

攻撃ツールによる攻撃を行い，リバースエンジニアリング対策としての有効性を確認することは今後の課題とする．