DATE: 2023/12/1

Department of Electrical and Electronic Information Engineering	ID	M203216
Name	Mikiya Ogura	

Supervisor	Shuichi Ichikawa

Abstract

Title	Hardware obfuscation using Obfuscator-LLVM and Bambu

Control system and embedded system software contains confidential information, and it is crucial to protect intellectual property in the system. This research addresses the methods for converting software into hardware to protect intellectual property using high-level synthesis (HLS), which automatically generates a hardware description language from software. In conversion of software into hardware, hardware obfuscation techniques can be applied to prevent the analysis by the attackers.

Yamada et al. employed the Obfuscator-LLVM (OLLVM) software obfuscation tool in conjunction with the HLS tool LegUp to generate obfuscated hardware. LLVM is a set of compiler and toolchain technologies that is widely used to develop HLS tools in recent years. By using LLVM intermediate representation (LLVM-IR) with HLS system, the effort to generate obfuscated hardware is drastically reduced. The problem is that LegUp is no longer available for research. This paper adopts the HLS tool Bambu to generate obfuscated hardware from LLVM-IR obfuscated by OLLVM. OLLVM is an LLVM middle end and may also work with other HLS tools developed with LLVM.

The proposed method was evaluated using CHStone benchmarks developed by Hara et al. The obfuscation methods available in OLLVM are Bogus Control Flow (BCF), Control Flow Flattening (CFF) and Instructions Substitution (ISub). First, I performed logic synthesis and simulation on the LLVM-IR obfuscated by a single method. The results of CHStone benchmarks are summarized by the geometric means of the number of Lookup tables (LUTs) and Wall Clock Time (WCT) of the generated obfuscated hardware. In LUT usage, the average of the proposed method was 58% larger than that of Yamada's method. Regarding WCT, Yamada's method was 13.6% faster than the proposed method with no obfuscation, whereas the proposed method was 7.3% faster than Yamada's method with obfuscation.

Next, I evaluated the hardware that was applied multiple OLLVM obfuscations, which Yamada could not achieve. The examined combinations of methods were BCF-CFF, CFF-BCF and CFF-ISub, which were recommended by Bansescu et al. Compared to the number of LUTs used in CFF, BCF-CFF increased by 45.1%, CFF-BCF increased by 0.3% and CFF-ISub increased by 0.7%. Comparing WCT with CFF, BCF-CFF was 74.3% slower, CFF-BCF was 20.6% slower and CFF-ISub was 0.4% slower. According to the result using a simple program, it is thought that the number of branches of switch instructions in the LLVM-IR significantly contributes to an increase in the number of LUTs and obfuscation of the LLVM-IR small contributes to an increase.

Our future work includes to conduct attacks using attack tools and confirm the effectiveness of the proposed method as a countermeasure against reverse engineering.