Automatic Hardware Trojan Insertion in Behavioral IPs during the Obfuscation Process

Nandeesh Veeranna, The Hong Kong Polytechnic University, Hong Kong
Benjamin Carrion Schafer, The University of Texas at Dallas, USA

Design Automation and Reconfigurable Computing Laboratory (DARClab)

Introduction

• Globalization trend in integrated circuit (IC) development — In house and third party intellectual properties (3PIPs)
• To reduce the IP design time — Raised the design abstraction level to behavioral
• High-level Synthesis (HLS) — ANSI-C/C++ to Register-Transfer-Level (RTL)
• Protect the BIPs — Encryption/Obfuscation
• Obfuscation — Easy and inexpensive way to protect the IPs

Obfuscation — An Example

/*Original code*/
buffer[0] = in0;
sum = sum + buffer[0];

/*Obfuscated code*/
37661-45842+0x1FFD; zddd43c876a = zddd43c876a + 0o2563-
for (zddd43c876a = 0xEFCD-52363-0x2341; zddd43c876a <
k795f772c7c = z7929401884 [0x32D5-0x2EF5-992];
0x2FED-2801+ zddd43c876a] + 0xEFCD-52364-0x2341; }

Ta2e5f06cde = ud904d243ce;
k795f772c7c = k795f772c7c + z7929401884 [0x3ADF-0x2FED-2801]

Additional Test-Vector Generation

/*Original code*/
sop= in0+(fir*calc);
temp= buffer[0]*var;

/*Obfuscated code*/
37661-45842+0x1FFD; zddd43c876a = zddd43c876a + 0o2563-
for (zddd43c876a = 0xEFCD-52363-0x2341; zddd43c876a <
k795f772c7c = z7929401884 [0x32D5-0x2EF5-992];
0x2FED-2801+ zddd43c876a] + 0xEFCD-52364-0x2341; }

Intermediate results Extraction

Out of 8 generated Taps (Tap 1, Tap 2, Tap 3)
- Compare values in New Taps (New Tap 1, New Tap 2, Tap 3)
- If similar values are found, request new taps

New Tap 1
New Tap 2
New Tap 3

Additional Test-Vector Generation

Final Obfuscated IP with the Vendor TB

Obfuscated IP with Trojan+Test-bench

No Trojan

Conclusions

• During the obfuscation, the rogue IP vendor inserts hardware Trojans into the behavioral IP and intelligently builds the test-bench so that it never triggers the Trojan for the customer during the behavioral IP verification
• In this work, we automatically created the testbench that does not trigger the Trojan, but also a testbench that can trigger the Trojan
• We have also created a simple source code obfuscator with the obfuscation functions such as mangling integers, trimming spaces and replacing the variables
• The entire flow is automated using perl script