Cube Stripping Function and Logic Restoration Based on Hardware Security

D.Keerthana, PG Scholar /ECE

Mr.K.Rajesh, M.E,ECE

SSM Institute of Engineering And Technology, Dindigul

ABSTRACT Logic locking conceived as a promising proactive defense strategy against intellectual property (IP) piracy, counterfeiting, hardware Trojans, reverse engineering, and overbuilding attacks. Various attacks that use a working chip as an oracle launched on logic locking to successfully retrieve its secret key, undermining the defense of all existing locking techniques. In this project, propose a stripped-functionality logic locking (SFLL), which strips some of the functionality of the design and hides it in the form of a secret key(s), thereby rendering on-chip implementation functionally different from the original one. When loaded onto an on-chip memory, the secret keys restore the original functionality of the design. Through security-aware synthesis that creates a controllable mismatch between the reverse-engineered net list and original design, SFLL provides a quantable and provable resilience trade-off between all known and anticipated attacks.

I. INTRODUCTION

Hardware has long been viewed as a trusted party supporting the whole computer system and is often treated as an abstract layer running instructions passed from the software layer. Therefore, hardware-related security research is often referred to hardware implementations of cryptographic algorithms where hardware is used to improve the calculation performance and efficiency for cryptographic applications. Hardware copyright protections are also categorized as hardware related security research where watermarking is widely used to solve the copyright issues. However, researchers from these areas do not consider the protection on the hardware itself. For a long time, cyber security researchers believed that the integrated circuit (IC) supply chain was well-protected with high barriers such that attackers could not easily compromise the fabricated chips. With the high cost of cutting-edge foundries and increasing design complexity of modern system-on-chip (SoC) platforms, the IC supply chain, which was once located in one country or even in one company, has been spread around the globe. Following this trend, third-party resources in hardware circuit designs, mostly in the format of third-party fabrication services and third-party soft/hard IP cores for SoC development, are prevailingly used in modern circuit designs and fabrications. The availability of those resources largely alleviates the design workload, lowers the fabrication cost, and shortens the time-to-market (TTM). However, the heavy reliance on third-party resources/services also breeds security concerns and invalidates the illusion that attackers cannot easily access the isolated integrated circuit (IC) supply chain. For example, a malicious foundry may insert hardware Trojans into fabricated chips.

The delivered IP cores may contain malicious logic and/or design flaws which could be exploited by attackers after the IP cores are integrated into SOC platforms.Besides the scope of hardware Trojan detection. While formal methods have been widely used in software program security assurance, they have also been proven to be effective in security verification on hardware code, which is often written in a hardware description language (HDL). The development of these methods helps provide high-level security assurance to hardware designs even in the circumstance that attackers may have the access to the original designs. These formal methods also help overcome the limitations of the requirement of golden models within many other hardware Trojan detection methods.

The evolution of hardware security research recently moved away from the hardware Trojan detection and now leans towards trustworthy hardware development for the construction of the root-of-trust. The intrinsic properties of hardware devices which have a negative impact on circuit performance are leveraged for security applications. One leading example is the development of physical-unclonable functions (PUFs) which rely on device process variation to generate chip-specific fingerprints in the format of challenge-response pairs. Looking beyond MOSFETs, researchers are investigating the use of emerging transistors, such as spin-transfer torque (STT) device, memristor, and spontronic domain wall, leveraging their special properties for hardware security applications.

II. RELATED WORKS

Yingjie Lao et al presented a novel approach to design obfuscated circuits for digital signal processing (DSP) applications using high-level transformations, a key-based obfuscating finite-state machine (FSM), and a reconfigurator. The goal is to design DSP circuits that are harder to reverse engineer. High-level transformations of iterative data-flow graphs have been exploited for area-speed- power tradeoffs. This is the first attempt to develop a design flow to apply high- level transformations that not only meet these tradeoffs but also simultaneously obfuscate the architectures both structurally and functionally. Several modes of operations are introduced for obfuscation where the outputs are meaningful from a signal processing point of view, but are functionally incorrect. Examples of such modes include a third-order digital filter that can also implement a sixth-order or ninth-order filter in a time-multiplexed manner. The latter two modes are meaningful but represent functionally incorrect modes. Multiple meaningful modes can be exploited to reconfigure the filter order for different applications. Other modes may correspond to non meaningful modes. A correct key input to an FSM activates a reconfigurator. The configure data controls various modes of the circuit operation. Functional obfuscation is accomplished by requiring use of the correct initialization key, and configures data. Wrong initialization key fails to enable the reconfigurator, and a wrong configure data activates either a meaningful but nonfunctional or non meaningful mode. Probability of activating the correct mode is significantly reduced leading to an obfuscated DSP circuit. Structural obfuscation is also achieved by the proposed methodology via high-level transformations.

similarity comparison method based on a new concept, longest common subsequence of semantically equivalent basic blocks, which combines rigorous program semantics with longest common subsequence based fuzzy matching. We model the semantics of a basic block by a set of symbolic formulas representing the input-output relations of the block. This way, the semantic equivalence (and similarity) of two blocks can be checked by a theorem prover. We then model the semantic similarity of two paths using the longest common subsequence with basic blocks as elements. This novel combination has resulted in strong resiliency to code obfuscation. We have developed prototypes. Marc Fyrbiak proposed two main contributions. We propose the first hybrid diversification approach for protecting embedded software and we provide statistical metrics to evaluate the protection. Diversification is achieved by combining hardware obfuscation at the microarchitecture level and the use of software-level obfuscation techniques tailored to embedded systems. Both measures are based on a compiler which generates obfuscated programs, and an embedded processor implemented in an FPGA with a randomized Instruction Set Architecture (ISA) encoding to execute the hybrid obfuscated program. We employ a fine-grained, hardware-enforced access control mechanism for information exchange with the processor and hardware-assisted booby traps to actively counteract manipulation attacks.

III. PROPOSED METHODOLOGY

A powerful attack that broke all the logic locking techniques existing then is Boolean satisfiability (SAT)-based key-pruning attack, referred to as SAT attack. Be attack is based on the notion of incorrect key elimination using distinguishing input patterns (DIPs) [44]. DIPs are computed using a miter circuit constructed using two copies of the locked netlist; the two circuits share the primary inputs but have different key inputs. A DIP is found when the two copies of the locked netlist in their outputs. A functional IC with the secret key loaded in its memory is used as an oracle to identify the incorrect keys in an iterative fashion. Be computational complexity of the attack is expressed in terms of the number of DIPs generated by the SAT attack. Be latest research works on logic locking have focused on defending against the SAT attack.

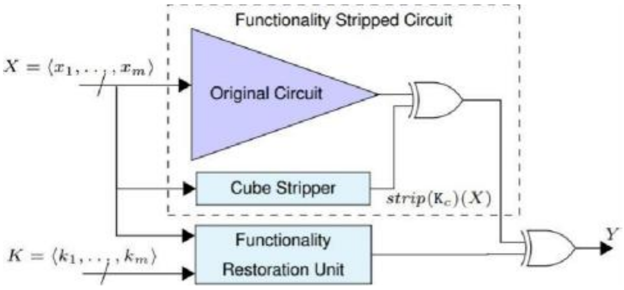

Functionality Stripped Circuit
$X = \langle x_1, \ldots, x_m \rangle$
Original Circuit
Cube Stripper
$strip(\mathrm{K}_c)(X)$
Functionality
Restoration Unit
$K = \langle k_1, \ldots, k_m \rangle$
Y

Fig.1Circuit Diagram for Proposed System

Much subsequent work has focused on SAT-attack resilient logic locking that ensures the number of equivalence classes of keys is exponential in the key length. Broadly speaking, these proposals all share the structure shown in Figure. They introduce a circuit which “flips” the output refer to this component as the cube stripping unit. This flipped output is then inverted by a key-dependent circuit that we refer to as the progammable functionality restoration unit. 20

This latter circuit is guaranteed to have an exponential number of equivalence classes of keys and ensures SAT attack resilience.Initial proposal along these lines were Anti-SAT and SARLock. Anti-SAT was vulnerable to the signal probability skew (SPS) attack while SARLock was vulnerable to the Double DIP attack and the Approximate SAT attack. Both schemes are vulnerable to removal and bypass attacks. TTLock and Secure Function Logic Locking (SFLL) to the best of the knowledge, SFLL is the only combinational logic locking scheme resilient to all of the above attacks. The vulnerability that I identify in the state-of-the-art logic locking technique SFLL-hd. This vulnerability is an end-result of the fact that existing synthesis tools are security-oblivious. It is clear from the netlists, the SFLL authors indeed resynthesized their netlists to hide the protected patterns; the structural attack is able to identify the protected pattern in all the cases, in turn extracting the secret key. Until a truly security aware synthesis tool is developed, any defense that relies on conventional CAD tools will be vulnerable. By eliciting a vulnerability in a state-of-the-art logic locking technique that has been unbroken until now, the work emphasizes a major shortcoming, i.e., the need for the development of a security aware synthesis tool, thus identifying a very important research direction. The technique that enhances Anti-SAT increases the connectivity between the newly added block and the original logic to prevent simple removal attacks. This enhancement technique is of no use for SFLL-hd either the attack expects and searches for traces stepping from a hamming distance checker embedded into the original circuit for functionality stripping. context of computer imaging are images with only the two colors, black, and white also called bi-level or binary images.

IV. SIMULATION RESULTS

In this section, the simulation results are implemented using xilinux and the comparison results and performance charts are given below:

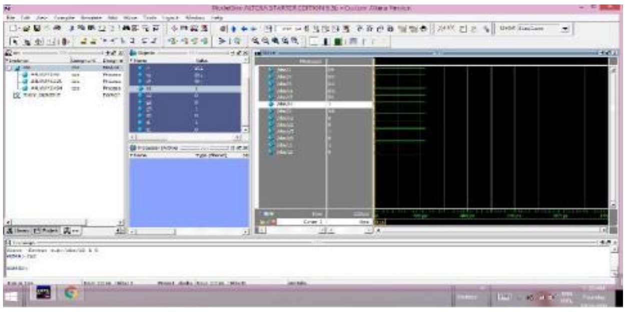


Fig .2 simulation result of c7

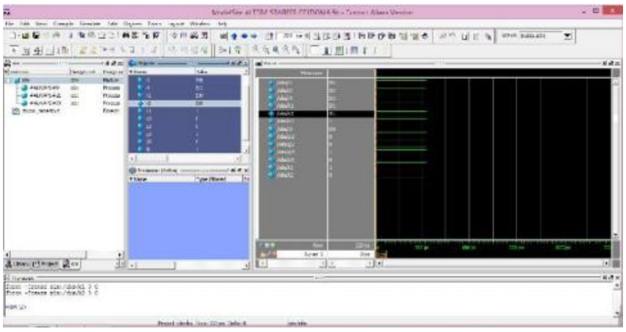


Fig. 3 simulation result of c14

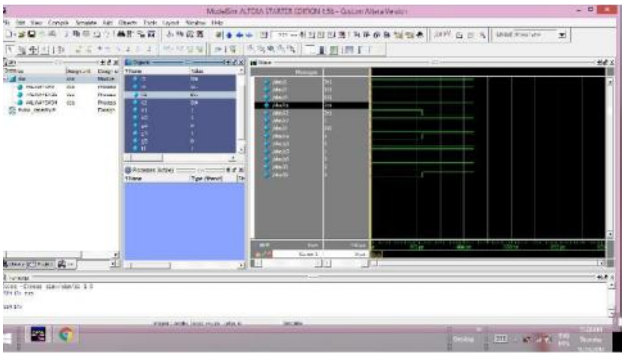


Fig.4 simulation results of c1355

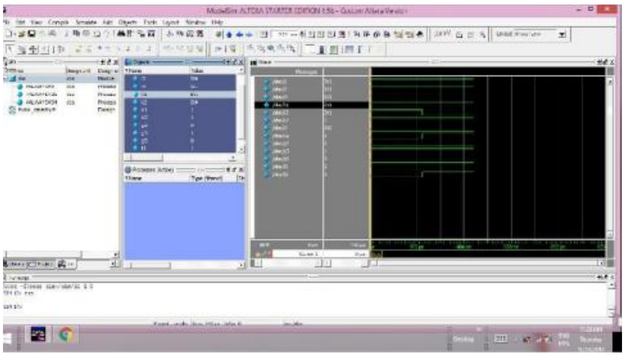


Fig.5 simulation result of c13

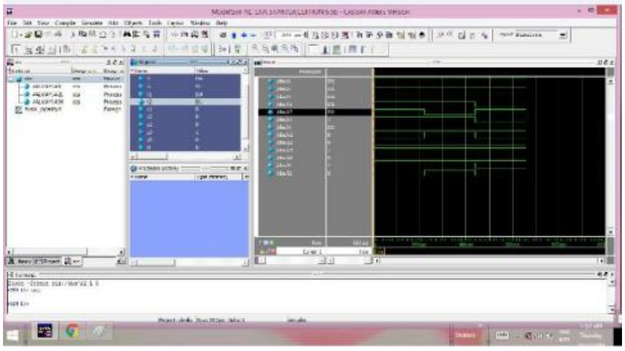
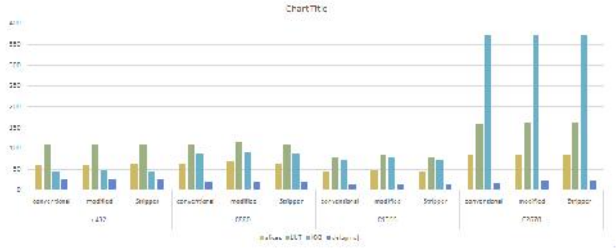


Fig. 4 simulation result of c2380

Comparsion Graph



Performance analysis

V. CONCLUSION

In this project proposed cube stripping based Functional Analysis attacks on state of the art Logic Locking algorithms (FALL attacks). The FALL attacks identified the locking key used structural and functional analysis. Experiments showed that the proposed method succeeded against existing system of bench mark circuits locked using cube strip Secure Function Logic Locking (SFLL), the only combinational locking algorithm resilient to all known attacks. The structural implementation of the functionality strip operation in the locked design tools to implement the functionality strip operation. The experimental results show that, the considerable reductions in the slices count and clock period time. The design methodology and Xilinx 14.2 is used as a simulation tool to show the performance analysis.after simulation, hardware implementation has been done using SPARTAN-3 FPGA Board.

REFERENCES

[1]. Alkabani Y.M. and Koushanfar F., (2007), "Active hardware metering for intellectual property protection and security," in Proceedings of 16thUSENIX Security Symposium on USENIX Security Symposium, ser.SS'07. Berkeley, CA, USA: USENIX Association, pp. 20:1–20:16.

[2]. Baumgarten A., Tyagi A.,(2010) and Zambreno.J "Preventing IC piracy using reconfigurable logic barriers," IEEE Design & Test of Computers,vol. 27, no. 1, pp. 66–75.

[3]. Emtena A., Garg S., Imeson F., and Tripunitara M.,(2013), "Securing computer hardware using 3D integrated circuit (IC) technology and split manufacturing for obfuscation," in Presented as part of the 22ndUSENIX Security Symposium (USENIX Security 13). Washington, D.C: USENIX, pp. 495–510.

[4]. Guin U., Huang K., DiMase D., Carulli J.M., Tehranipoor M. and Makris Y.,(2014), "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," Proceedings of the IEEE, vol. 102, no. 8 pp. 1207–1228.

[5]. Koushanfar F.,(2012), "Provably secure active IC metering techniques for piracy avoidance and digital rights management," IEEE Transactions on Information Forensics and Security, vol. 7, no. 1, pp. 51–63. 38

[6]. Rajendran J., Pino Y., Sinanoglu O. and Karri R.,(2012) "Security analysis of logic obfuscation," in Proceedings of the 49th Annual Design Automation Conference on - DAC '12.

[7]. Rajendran J.,Pino Y., Sinanoglu O. and Karri R., (2014),"Fault analysis-based logic encryption," IEEE Transactions on Computers, vol. 64, no. 2, pp. 410–424.

[8]. Tehranipoor J. and Koushanfar F.,(2010), "A survey of hardware Trojan taxonomy and detection," IEEE Design & Test of Computers, vol. 27,no. 1, pp. 10–25.

[9]. Liu B. and Wang B.,(2014), "Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks," in Design, Automation & Test in Europe Conference & Exhibition (DATE).

[10]. Subramanyan P., Ray S. and Malik S.,(2015), "Evaluating the security of logic encryption algorithms," IEEE International Symposium on Hardware Oriented Security and Trust (HOST).

[11]. Shen Y. and Zhou H.,(2017), "Double DIP," in Proceedings of the on Great Lakes Symposium on VLSI.