Indian Institute of Technology Madras

Patanjali SLPSK | CS12D024

443, Bramhaputra Hostel, IIT Madras
✉ patanjali.slpsk@gmail.com • 🏠 patanjali.github.io
Bitbucket repo: https://bitbucket.org/patanjali/

Education

Program	Institution	%/CGPA	Year
PhD. (CSE)	IIT Madras	**8.73**	ongoing
MS (CSE)	IIT Madras	**8.6**	converted to PhD
B.Tech. (CSE)	Pondicherry Engineering College (PEC)	**8.32**	2011
XII	Petit Seimnaire Higher Secondary School, Pondicherry	**94.17**	2007
X	Petit Seimnaire Higher Secondary School, Pondicherry	**89.74**	2005

Publications

Journals

- **Patanjali SLPSK**, Milan Patnaik, Seetal Potluri, and V. Kamakoti. "MLTimer: Leakage Power Minimization in Digital Circuits Using Machine Learning and Adaptive Lazy Timing Analysis." Journal of Low Power Electronics 14, no. 2 (2018): 285-301.
- Krishnakumar, Gnanambikai, **Patanjali SLPSK**, Prasanna Karthik Vairam, Chester Rebeiro, and Kamakoti Veezhinathan. "GANDALF: A fine-grained hardware-software co-design for preventing memory attacks." IEEE Embedded Systems Letters (2018).

Conferences

- Gautham, Ashok, Kunal Korgaonkar, **Patanjali SLPSK**, Shankar Balachandran, and Kamakoti Veezhinathan. "The Implications of Shared Data Synchronization Techniques on Multi-Core Energy Efficiency." In HotPower. 2012.
- **Patanjali SLPSK**,Prasanna Karthik, Chester Rebeiro, Kamakoti Veezhinathan."Karna: A gate-sizing based Security Aware EDA Flow for Improved Power Side-Channel Attack Protection". ICCAD 2019 (under review).

Posters

- **Patanjali SLPSK**, Prasanna Karthik .V, Chester Rebeiro, Kamakoti .V. "Karna: A Security Aware EDA Flow for Improved Side-Channel Attack Protection". Design Automation Conference(2019) under Work-In-Progress section
- **Patanjali SLPSK**, Seetal Potluri, and Kamakoti Veezhinathan. "FastReplace: Efficient Vt Replacement Technique for Leakage Power Minimization". In Proceedings of the 2014 Design Automation Conference (DAC 2014)under Work-In-Progress section
- **Patanjali SLPSK**, Seetal Potluri and Kamakoti Veezhinathan. "HALTimer: A Fast Vt Replacement Heuristic for. Leakage Power Minimization". In Proceedings of 2015 Design Automation conference (DAC'15) under Work-In-Progress section.

Miscellaneous

- Gargi Mitra, Prasanna Karthik, **Patanjali SLPSK**, Nitin Chandrachoodan, Kamakoti.V. "White Mirror: Leaking Sensitive Information from Interactive Netflix Movies using Encrypted Traffic Analysis". arxiv March 2019.

Ongoing Research Work

Samaritan: An automated framework for detecting fault attacks in hardware **Jan 2019-ongoing**
Python,Verilog

Automatic detection of fault vulnerabilities is a challenging problem because of the complicated attack surface. An interesting observation is that, out of the large number of faults that can occur in the design, very few faults are actually exploitable. However, there are no automated tools that can identify these exploitable fault locations in a hardware device. Hence current countermeasures cannot target these exploitable locations and incur significant overhead. Another challenge is that the designers have to ensure that the security guarantees of these countermeasures are met at every stage of the manufacturing process. Formal verification is a tool that can be leveraged to ensure that these guarantees are met without incurring significant overheads. We propose Samaritan, a formal framework that addresses these problems using user-defined specifications and the implemented design. Given a design implementation, either at RTL or gate level, SAMARITAN is capable of pinpointing the vulnerable regions in the design, thereby enabling targeted countermeasures to be deployed. We demonstrate the efficiency of this framework using three ciphers namely, AES, SIMON, and CAMELLIA.

Privacy aware Deep Learning **June 2018-ongoing**
PyTorch

The advent of IoT has led to massive data generation due to the widespread deployment of sensors, networks, and processing devices. Deep Learning has emerged as a popular computing paradigm that is used to process these large scale data and gather useful information. Of late, companies like Amazon and Google have provided cloud-based Deep Learning solutions. These solutions are referred to as Deep Learning as a Service (DLaaS). Recent works have demonstrated that these DLaaS systems are quite vulnerable to adversarial attacks. In this work, we explore three different adversarial scenarios i) naive adversary, ii) countermeasure-aware adversary and iii) Untrusted server. We also explore countermeasures inspired from information theory such as adding noise and data-shuffling. We quantify the impact of these countermeasures on ResNet and GoogleNet architectures using the Imagenet dataset. We demonstrate that we can outperform the adversary in all three scenarios.

Security Aware Gate Sizing **Dec 2018-ongoing**
Tcl,C++

Fault attacks are one of the growing physical attacks exploits that target edge devices. The attacker introduces faults at specific locations in the device during encryption thereby corrupting the result of the operation. These faults are introduced by either manipulating the system clock, the power supply or by using a laser. Traditional countermeasures rely on redundant logic, parity checking which introduce area and power overheads. In this work, we explore the idea of gate-sizing as a countermeasure for fault attacks. We observe that by modifying the parameters of the vulnerable gates, the side-channel resistance of the design improves. We leverage this observation and propose SecureSizer, a gate-sizing technique to improve the side-channel resistance of the design. We show that the overheads of the proposed technique are less when compared to traditional countermeasures.

Completed Research Work

MLTimer: Leakage Power Minimisation in Digital Circuits using Machine Learning and Adaptive Lazy Timing Analysis **Jan 2016- Dec 2017**
Journal of Low Power Electronics June 2018 *C,Verilog*

Power optimization techniques in a VLSI flow typically end up being the performance bottlenecks leading to a large turn around time for the following reasons

- **Scalability**: The design typically spans millions and millions of gates with different operating conditions leading to a large search space.
- **Portability**: The constraints vary across technology nodes hindering reusability of solutions.

ML models are inherently trained to operate on large datasets and navigate a complex search space. The contributions of our work are as follows.

- We propose a novel learning (Support Vector Machine) based classifier, which provides a good initial design configuration that guarantees leakage optimal solution.
- We use a Lazy Timing Analysis procedure that postpones the timing validation step as much as possible.
- We show the efficiency of our technique on large scale benchmark datasets (25K - 1million gates). **Our technique performs 23% better in terms of solution quality when compared with the state-of-the art technique and 50% better in terms of runtime**.

GANDALF: A fine-grained hardware-software co-design for preventing memory attacks **June 2017-November 2017**
Embedded System Letters Special Issue Feb 2017 *C,Verilog*

- **Won first place in Embedded systems challenge in CSAW 2016**.
- Developed a **lightweight** full-stack architecture to prevent memory attacks on OpenRISC processors.
- Our framework allows legacy code and protected to co-exist without any additional overheads.We demonstrate this by running Linux and execute the protected code on top of it.
- Ours is the first framework to demonstrate a full fledged system with complete hardware, OS and compiler modifications that runs Linux, legacy and protection enabled code seamlessly.
- **We achieve complete temporal safety with only 1% delay overhead 0% area overhead, 60% code size increase**.

Karna: A Security Aware EDA Flow for Improved Side-Channel Attack Protection **June 2018-November 2018**
ICCAD Nov. 2019 ***Under review*** *C,Verilog*

- Observed that different parts of the same design leak information differently .
- Developed an algorithm that identifies the vulnerable areas in a given design.
- Our framework modifies the gate parameters of the gates in these vulnerable areas to reduce information leaks.
- Ours is the first framework to achieve this at a design level.
- We achieve complete safety with **no delay, power and area** overheads on large ciphers like AES and lightweight ciphers like Simon and PRESENT.

Achievements

- **Winner, First Place** in TCS Student Research Presentation in Shaastra IIT Madras, 2018.
- **Winner, First Place** in Embedded Systems Challenge in CSAW 2016.
- **Winner, Second Place** in HackU 2013 at IIT Madras. **Click here for demo**
- Came first in the DELF A1 Certification Exam conducted by Alliance Française Pondicherry. .

Internship

Developing a distributed EDA framework **June 2015- December 2015**
IBM System Development Labs India *Spark, Scala, Python*

In a typical design flow, multiple tools and scripts are used. Each tool reads the input, its associated constraints, processes and generates data as output. At present there is no mechanism to represent this data in a database which limits the possibility of live query on the data and get a useful information and value add out of it. For example: to find a timing critical path it requires to load a timer which in turn processes the input and eventually writes out report. Instead, it can be easily queried if the database is annotated with relevant information. The project addresses the following:

- Handle Hierarchical design data
- Distribution over multiple clusters (as design is developed over multiple sites and multiple people and partitioned)
- Extensible as data can be annotated with any information on top of the base structure.

Positions of Responsibility

- **System Administrator** Manage a lab of around 60-100 students. I have helped setup a local version control system, bug tracker and a GPU cluster.
- **Teaching Assistant** for Secure Systems Engineering, Computer Organization, Digital Systems Testing and Testable Design, Digital Systems Verification, Operating Systems and Advanced Programming Lab.

Technical Skills

- **Programming Languages:** C, C++, Python, MATLAB, Scala, Bluespec, VHDL and Verilog
- **Simulation Platforms** Xilinx, Altera, Cadence and Synopsys.

Course Work

- **Machine Learning**: Machine Learning
- **VLSI**: CAD for VLSI, Digital Design Verification, Digital Systems Testing and Testable Design, Mapping Signal Processing algorithms to DSP architecures.
- **System Design**: Computer Architecture, Concurrent Programming
- **Security**: Foundations of Cryptography, Mathematical concepts in Computer science.