ICCAD: G: A Synergistic Framework for Hardware IP Privacy and Integrity Protection

Meng Li, meng_li@utexas.edu, ACM ID: 5955939

ECE Department, Univ. of Texas at Austin, Austin, TX 78712

1. PROBLEM AND MOTIVATION

As the technology node scales down to 45nm and beyond, the significant increase in design complexity and cost propels the globalization of the $400-billion semiconductor industry. However, such globalization comes at a cost. Although it has helped to reduce the overall cost by the worldwide distribution of integrated circuit (IC) design, fabrication, and deployment, it also introduces ever-increasing intellectual property (IP) privacy and integrity infringement. Recently, primary violations, including hardware Trojan, reverse engineering, and fault attack, have been reported by leading semiconductor companies and resulted in billions of dollars loss annually [1].

While hardware IP protection strategies are highly demanded, the researches were just initiated lately and still remain preliminary. Firstly, the lack of the mathematical abstractions for these IP violations makes it difficult to formally evaluate and guarantee the effectiveness of the protections. Secondly, the poor scalability and cost-effectiveness of the state-of-the-art protection strategies make them impractical for real-world applications. Moreover, the absence of a holistic IP protection further diminishes the chance to address these highly correlated IP violations which exploit physical clues throughout the whole IC design flow.

To protect hardware IP privacy and integrity, we propose a synergistic framework to help IP vendors from design, optimization and evaluation perspectives. Our framework consists of three algorithms, all of which are developed based on rigorous mathematical modeling for primary IP violations. The three algorithms focus on different stages of IC design and can collaborate with each other to provide a formal security guarantee and an accurate security evaluation.

2. BACKGROUND AND RELATED WORK

Recent decades have witnessed the globalization of the IC supply chain impelled by the ever-increasing design complexity and cost. As shown in Figure 1, in modern IC supply chain, the design houses take full charge of the hardware design from the register-transfer level (RTL), gate level, to physical level. Then, the IC designs are shipped across the world for fabrication, assembly, packaging, and deployment. Due to the lack of control over the supply chain after the shipment of the designs, hardware IP privacy and integrity violations emerge.

Primary IP violations can be categorized into three classes: hardware Trojan, reverse engineering, and fault attack, as shown in Figure 1. Hardware Trojans are malicious modifications made by untrusted foundries. For example in Figure 2(c), gate 6 is inserted maliciously at the output of gate 2. The detection of such malicious gate can be challenging for uninformed designers if the trigger signal t is deliberately selected. To prevent Trojan insertion, split manufacturing is proposed to hide the critical signals, including the trigger signals, e.g., signal t, and the target signals, e.g., node 2 [2,3]. The basic idea is to split a circuit layout into the front-end-of-line (FEOL) layers and back-end-of-line (BEOL) layers as shown in Figure 2(a). BEOL layers only consist of the wires in higher metal layers and are fabricated in-house, through which critical signals can be protected. For example, consider

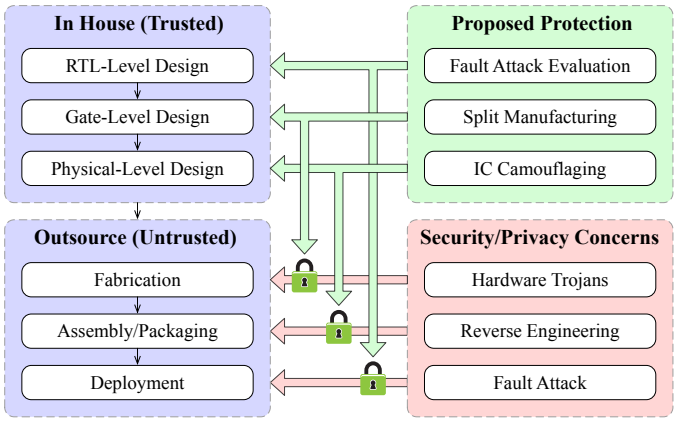


Figure 1: Security concerns of the modern globalized supply chain and the proposed protections.

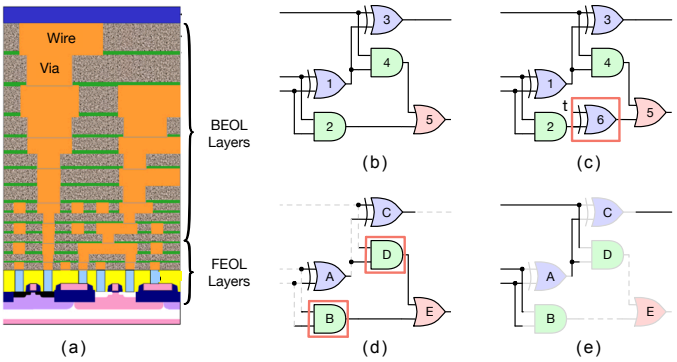


Figure 2: Example of split manufacturing: (a) definition of the FEOL and BEOL layers [4], (b) the original netlist, (c) Trojan inserted netlist, (d) the FEOL layers of the design and (e) the BEOL of the design.

the FEOL and BEOL layers shown in Figure 2(d) and 2(e). By hiding the wires in the BEOL layers, the attackers cannot distinguish gate B and D, and thus, cannot determine which gate in the FEOL layers implements node 2.

Although split manufacturing demonstrates promising protection against Trojan insertion, a critical problem is that the security cannot be guaranteed for all the signals simply by hiding the BEOL layers. For example in Figure 2, regardless of the wires hidden in the BEOL layers, node 5 can always be identified as it has a different gate type compared with the other nodes. Besides the lack of security guarantee, the state-of-the-art algorithms also suffer from poor scalability. This mainly originates from the repetitive security evaluation, which requires computation-intensive operations on large graphs [2].

The second class of IP violations comes from reverse engineering. After the fabrication, the chips are sent out for assembly, packaging and eventually go to the open market. The attackers can now strip the chip layer by layer to reconstruct the gate-level netlist and re-distribute it without the authorization from the IP vendors. To prevent reverse engineering, IC camouflag-

ing emerges as a remedy [5, 6]. By leveraging fabrication-level techniques, camouflaging cells are first designed to implement different functionalities, even though their layouts look the same to the attackers. They are then inserted into the circuit for obfuscating the circuit functionality.

Extensive researches have been conducted to insert the camouflaging cells so that the security can be maximized under the overhead constraints [5–7]. Meanwhile, different attack strategies are also proposed to resolve the correct circuit functionality [8]. Although the arms race between the attack and protection inspires better defense against reverse engineering, the key shortcoming is that due to the lack of a formal security metric, the security of all the camouflaging strategies is only evaluated with the existing attack methods. The empirical nature of such evaluation leads to an overestimation of the security level. Moreover, because of the absence of a cost-effective strategy for the design and insertion of camouflaging cells, existing methods also suffer from a large overhead on area, power, and timing.

The third class of IP integrity issues comes from fault attacks after the chips are deployed [9–11]. Fault attacks target at obstructing the normal system execution by injecting errors into the hardware. By radiating the critical circuit components with high energy particle strikes, voltage transients are created in order to make the circuit malfunction temporarily. Fault attacks have demonstrated a great capability of leaking critical system information, e.g., the cryptographic keys, and nullifying the entire system security mechanisms [9–11].

To protect against fault attacks, an accurate evaluation of the system vulnerability is highly demanded to identify critical circuit components and guide the design optimization. However, it is challenging to quantitatively evaluate the system vulnerability due to the probabilistic nature of the fault attack process. The randomness of the attack process mainly comes from the uncertainty of fault injection techniques. Our study on commercial processors reveals that failing to account for the attack uncertainty can result in over-pessimistic estimation of the system vulnerability. Nonetheless, it is not easy to capture the attack uncertainty efficiently. Though statistical metrics have been proposed [10, 11], an explicit enumeration of the system state space is usually required for an accurate evaluation, whose complexity increases exponentially with the system size.

Thus far, the IP violations described above are already hard to protect against by themselves. However, they can be combined together to further empower the attackers. Therefore, to defend against the IP violations, we propose a synergistic framework to help IP vendors with security evaluation and optimization. Our framework is based on rigid mathematical abstractions of IP violations and consists of the following three components that can interact with each other:

- **Practical split manufacturing algorithm:** for the first time, the insertion of dummy gates and wires is exploited in the split manufacturing process. *We extend the existing security criterion and prove the security guarantee of our algorithm.* A sufficient condition is derived to avoid security evaluation when generating the FEOL layers, which can be realized with fast relaxation algorithms and achieves 1400× speedup compared with the state-of-the-art [2].
- **Provably secure IC camouflaging:** *the equivalence between the reverse engineering attack and the boolean function learning problem is built for the first time,* based on which, a quantitative security metric is derived. The key security impacting factors are identified with different techniques proposed to achieve an exponential increase of the security level with a linear increase of the overhead.
- **Fast yet accurate fault attack evaluation:** we propose a probabilistic model and a statistical metric to capture the attack uncertainty. We also propose a Monte Carlo flow to evaluate the statistical metric efficiently, which is further empowered with an importance sampling strategy. *Our evaluation flow is prototyped by a leading IP vendor, Arm Inc., and validated on commercial processors.*

As shown in Figure 1, the proposed algorithms focus on different stages of IC design and closely collaborate with each other to protect hardware IP against the untrusted supply chain.

3. APPROACH AND UNIQUENESS

3.1 Split Manufacturing Optimization

We first introduce our split manufacturing algorithm for Trojan prevention [3, 12]. To resolve the problems of the state-of-the-art methods and provide a formal security guarantee, we propose to insert dummy gates and wires into the FEOL layers besides hiding the BEOL layers. We focus on answering the following critical questions: 1) how to define the security criterion to account for the insertion of dummy gates and wires, and 2) how to efficiently generate the FEOL layers to achieve the required security level.

Attack Model: We follow a strong attack model [2] and allow the attackers to have access to both the original netlist and the FEOL layers. Hence, the attackers can first determine the target nodes for Trojan insertion in the netlist and then, try to identify the physical gates that implement the target nodes.

Security Criterion: Split manufacturing is proposed to prevent the physical gates from being identified. Recall the example in Figure 2. We show the graph representations of the original design and the FEOL layers in Figure 3(a) and 3(b). Since the BEOL layers are hidden, the attackers cannot determine whether gate B or D implements node 2 and are forced to guess between the two gates. We denote gate B and D as the candidates for node 2. With the increase of the number of candidates, the chance for the attackers to identify the implementation of node 2 diminishes. Therefore, a security criterion, defined as k-security, is proposed [2]. The FEOL layers are k-secure if there are at least k different candidates for each node in the original netlist.

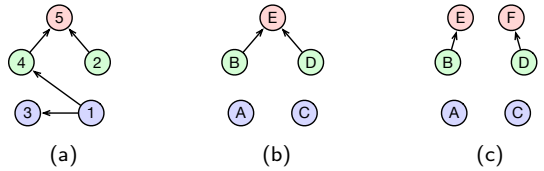


Figure 3: Graph representation of the example in Figure 2 (color represents the gate type): (a) the original netlist; (b) FEOL layers generated by hiding the BEOL layers; and (c) FEOL layers generated after the insertion of dummy wires and gates.

Proposed High-level Strategy: To achieve k-security, hiding the BEOL layers alone cannot provide any security guarantee. For the example in Figure 3, as node 5 has a different gate type compared with other nodes, despite the wires hidden in the BEOL layers, the FEOL layers are always 1-secure. Yet, we observe by inserting a dummy gate F and a dummy wire (D, F), node 5 can be protected with two candidates, i.e., gate E and F, and the FEOL layers are now 2-secure.

However, it is not straightforward to allow the insertion of dummy gates and wires as it is incompatible with the existing security definition. The existing security definition relies on the fact that the graph of the FEOL layers is a subgraph of the original netlist, which does not hold when dummy gates and wires are inserted [2]. In our work, we leverage the concepts from graph analysis and propose a more generic definition of k-security. *Moreover, we prove that by considering the insertion of dummy gates and wires, we can guarantee k-security for the generated FEOL layers, for any required security level k.*

Proposed k-security Realization: To realize k-security, we simultaneously insert dummy gates/wires to the FEOL layers and search for wires to hide in the BEOL layers. We derive a sufficient condition for k-security to avoid the security evaluation and the computation-intensive graph operations: if the graph of the FEOL layers consists of k disjoint isomorphic subgraphs, k-security is guaranteed. For example, in Figure 3(c), the graph consists of 2 disjoint isomorphic subgraphs, i.e., $\{A, B, E\}$ and $\{C, D, F\}$, and thus, 2-security can be achieved. To achieve the sufficient condition, we come up with a secure-by-construction approach based on a mixed-integer linear programming (MILP) formulation to construct the FEOL layers. We further propose a Lagrangian Relaxation (LR)-based algorithm to iteratively solve the MILP formulation and a minimum-cost flow transformation to speed up each iteration.

Overall Contribution: In our split manufacturing flow, for the first time, the insertion of dummy gates and wires is considered besides hiding the BEOL layers. We extend the existing security criterion to accommodate the inserted dummy gates/wires and provide a formal guarantee of achieving the required security level. To realize k-security, we first derive a sufficient condition to avoid the security evaluation when generating the FEOL layers and then, propose an efficient algorithm, which achieves more than 1400× speedup compared with the state-of-the-art [2].

3.2 Provably Secure IC Camouflaging

To prevent the second class of attack, i.e., reverse engineering, we propose a provably secure IC camouflaging strategy. In contrast to the empirical nature of the existing methods, we focus on the following two aspects: 1) how to quantitatively measure the security against reverse engineering, and 2) how to enhance the security for general circuits with a formal guarantee [7, 13, 14].

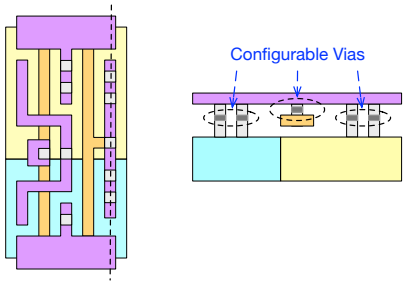
Attack Model: We assume the attackers to have access to the camouflaged netlist and a functional circuit [5, 6]. The attackers cannot directly determine the functionality of the camouflaging cells but can query the functional circuit as a black box with selected input vectors and then, resolve the functionality of these cells based on the input-output pairs.

Proposed Security Criterion: To formally evaluate the security of the camouflaging strategies, we establish the equivalence between reverse engineering and the boolean function learning problem. We build the one-to-one mapping between the concepts in reverse engineering and those in the learning problem. For example, the set of possible functionalities for the camouflaged netlist corresponds to the set of boolean functions in the learning problem. Meanwhile, the input-output pairs correspond to the samples. Based on the equivalence, we propose a new security criterion, termed as attack complexity, which corresponds to the sample complexity and is defined as the number of input-output pairs required to determine the correct circuit functionality. We leverage the recent advance of the learning theory and formally derive the attack complexity N as

$$N \sim \mathcal{O}(\theta d \log \frac{1}{\epsilon}),$$

where d characterizes the total number of functionalities for the camouflaged netlist, and θ is inversely proportional to the number of incorrect functionalities pruned by each input-output pair. ϵ represents the output error probability for the resolved circuit. *This is the first time a quantitative security metric is derived.*

Proposed Camouflaging Strategies: To enhance the attack complexity, we propose two different camouflaging strategies. The first strategy targets at increasing d by inserting more camouflaging cells under the overhead constraints. We propose a new camouflaging cell design, with an example shown in Figure 4. Depending on the connectivity of the vias, the cell can work as an inverter or a buffer. The key feature of the cell is its overhead depends on its functionality as shown in Table 4(c). Hence, we can insert a large number of such cells into the design and make most of the cells function as buffers without introducing any overhead. As the attackers cannot determine the functionality for each cell, the number of possible functionalities for the netlist becomes large.



Func.	BUF	INV
Timing	1.0X	2.0X
Area	1.0X	1.5X
Power	1.0X	1.5X

Figure 4: Example of the camouflaging cell design: (a) top view; (b) cross section; and (c) overhead for different functionalities.

Our second strategy targets at increasing θ by reducing the number of incorrect functionalities pruned by each input-output pair. We propose a novel AND-tree structure as shown in Figure 5(a), whose input pins are camouflaged with camouflaging cells shown in Figure 4(a). Empirically, we find that for such an AND-tree structure, the number of incorrect functionalities pruned by each input-output pair is very close to 1 and remains unchanged with the tree size, which leads to an exponential increase of θ with the tree size. We formally derive the necessary conditions required to achieve the exponential increase of θ and propose a novel strategy to camouflage a general circuit with the AND-tree as shown in Figure 5(b). One input of the XOR gate is fixed to 0 to maintain the correct circuit functionality. The proposed strategy is proved to achieve an exponential increase of the attack complexity and thus, provide a provably secure guarantee.

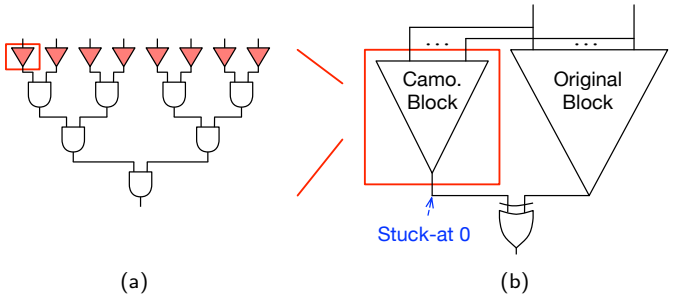


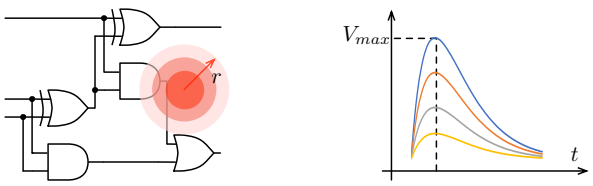
Figure 5: AND-tree based camouflaging strategy.

Overall Contribution: In our camouflaging flow, we formally define the security criterion and derive the key security impacting factors, i.e., θ and d. Two strategies are proposed to enhance the security, which achieve an exponential increase of the attack complexity and provide a formal guarantee on the security.

3.3 Fault Attack Evaluation

Another serious violation of the IP integrity is the fault attack, whose target is to alter the normal system execution. To prevent fault attack, we propose an efficient yet accurate evaluation flow to determine the system vulnerability and guide further design optimization [15, 16].

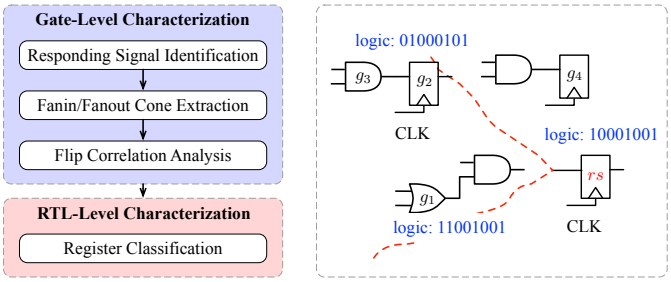
Attack Model: We follow the widely used attack model [10, 11] and assume the attackers to have physical access to the system. The attackers also know the physical implementation of the system and can choose the workload program. A common flow for the attackers is to first select a malicious program and then, inject faults into the system to compromise the security mechanisms.



(a) (b)

Figure 6: Illustration of the parameters in $\boldsymbol{p}$ for the radiation-based attack: (a) spatial uncertainty and (b) amplitude of the injected voltage transients.

Proposed Probabilistic Attack Modeling: To accurately evaluate the system vulnerability, we need to capture the intrinsic uncertainty in the attack process. We use the fault injection cycle t and the technique parameter vector $\boldsymbol{p}$ to model the attack process and regard them as random variables following a distribution $f_{t,\boldsymbol{p}}$. The randomness of t captures the temporal uncertainty of the attack process. $\boldsymbol{p}$ varies depending on the attack techniques. For the radiation-based attacks, as the fault may not be injected to the targeted location, $\boldsymbol{p}$ consists of the spatial uncertainty as shown in Figure 6(a). $\boldsymbol{p}$ can also include the amplitude of the injected voltage transients as shown in Figure 6(b). Based on the probabilistic model, we define the system security factor (SSF) to measure the probability of a successful fault attack and characterize the system vulnerability against fault attacks.



(a) (b)

Figure 7: System pre-characterization flow.

Fast Security Evaluation: To evaluate SSF efficiently, we introduce a Monte Carlo based approach to avoid explicit enumeration of the system state space via sampling. Instead of directly sampling from the whole sample space, we propose an importance sampling strategy. The key intuition is to increase the sampling probability for the region in the parameter space that is more likely to result in a successful attack. We propose a cross-level system pre-characterization flow shown in Figure 7(a) to determine the sampling distribution. We use the example in Figure 7(b) to illustrate the pre-characterization flow. First, we identify the responding signals (RSs) in the system that alert the processors in case of security violations, e.g., rs in Figure 7(b). For an attack to succeed, these signals must be temporarily blocked with the injected faults. Then, we extract the fan-in and fan-out cones of the RSs as only the attacks on these two regions can potentially block the RSs. Thirdly, we analyze the correlation of the signals in the fan-in cones with the RSs. The signal with a higher correlation is statistically preferred as they are more likely to alter the RSs. For example, g_1 is statistically preferred compared to g_2 as it has a higher signal correlation with RS. Lastly, we classify the registers in the fan-in cones of the RSs and propose different strategies for each class. Based on the pre-characterization, the sampling distribution can be determined analytically. To efficiently determine the attack outcome for each sample, we further develop a cross-level simulator [15].

Overall Contribution: Our evaluation flow features a probabilistic modeling for the attack process to capture its intrinsic uncertainty, based on which a statistical metric is proposed. To evaluate the metric, we propose a Monte Carlo method and develop a pre-characterization procedure to facilitate an importance sampling, which achieves superior accuracy and efficiency.

3.4 Synergistic IP Protection

As shown in Figure 1, the three algorithms in our framework can be applied to the RTL-level, gate-level, and physical-level design stages. By combining them together, better resilience against the IP privacy and integrity violations can be achieved. We use fault attack and reverse engineering as examples to demonstrate the stronger protections.

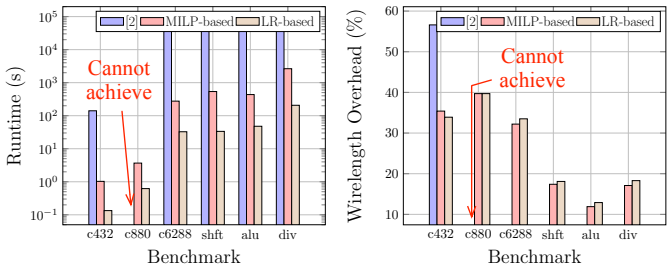
For the fault attack, currently, the attackers are assumed to have the implementation details of the design. Such assumption is made as the design details can be leaked by untrusted foundries or through unauthorized reverse engineering. With the proposed split manufacturing and IC camouflaging algorithms, such assumption can be significantly relaxed, which further impair the attackers' capability to determine the fault injection time and chip locations in the attack process.

For the reverse engineering, with the proposed IC camouflaging flow, although high security is achieved against malicious end-users, we have to rely on a trusted foundry since the IP vendors have to reveal design details to the foundries to leverage the camouflaging fabrication techniques. With split manufacturing, the reliance on a single trusted foundry diminishes. We can now split the fabrication into multiple foundries so that even though some of the foundries can be untrusted, the circuit functionality can still be protected.

4. RESULTS AND CONTRIBUTIONS

4.1 Split Manufacturing Optimization

We first prototype our split manufacturing algorithm and compare the proposed MILP-based and LR-based algorithms with the current state-of-the-art [2]. We set the security level k to be 10. The efficiency comparison is shown in Figure 8(a). As we can see, [2] only achieves 10-security for the smallest benchmark, c432. For benchmark c880, even if all the wires are hidden, 10-security cannot be achieved. For large benchmarks, due to the runtime inefficiency, [2] cannot be finished within 10^5 seconds. Both our MILP-based and LR-based algorithms can achieve 10-security for all the benchmarks, which demonstrate a much stronger security guarantee and a better efficiency. Specifically, for small benchmarks, we can achieve 1400× speedup compared with [2]. For the largest benchmark, our LR-based algorithm can be finished within 200s.



(a) (b)

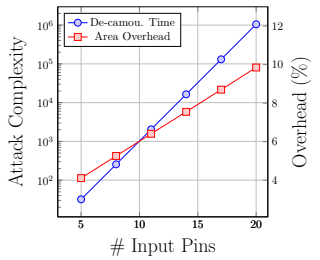
Figure 8: Comparing the traditional method and the proposed MILP-based, LR-based methods: (a) runtime, and (b) overhead.

We also compare the introduced wirelength overhead. As shown in Figure 8(b), for benchmark c432, we achieve 37% wirelength overhead reduction compared with the state-of-the-art [2]. Our

LR-based algorithm introduces on average less than 2% overhead increase compared with the MILP-based algorithm, which is negligible especially considering the significant speedup.

4.2 Provably Secure IC Camouflaging

We prototype our IC camouflaging algorithm and verify the security leveraging the state-of-the-art attack strategy [8]. We first verify the security and evaluate the overhead of the AND-tree based strategy. As shown in Figure 9(a), our strategy achieves an exponential increase of the security level with a linear increase of overhead with the tree size.



(a)

	Security			Overhead
	Worst	Avg	Best	
Ours	2^{64}	2^{64}	2^{64}	1.90%
[5]	2^{64}	2^{64}	2^{64}	3.67%
[6]	1	2^{64}	2^{64}	11.1%

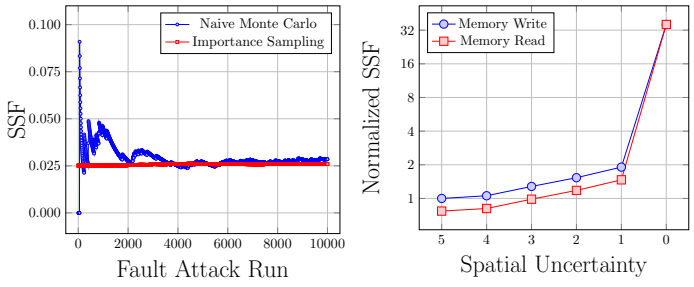
(b)

Figure 9: (a) Overhead and security trade-off and (b) comparison with state-of-the-art when a 64-bit AND tree is inserted.

We then compare our camouflaging strategy with other methods that also leverage AND-tree [5, 6]. After inserting a 64-bit AND-tree into the benchmark, i.e., div, as shown in Table 9(b), our strategy achieves the best security level in the worst, average, and best cases. The overhead introduced by our method is less than 2%, which is much less than the other two methods.

4.3 Fault Attack Evaluation

We prototype our fault attack evaluation algorithm using the internal security evaluation framework of a leading IP vendor, Arm Inc., and apply our algorithm to commercial processors to check the memory isolation mechanism. We first compare the efficiency of the proposed importance sampling strategy with the strategy that directly samples from the whole space on commercial benchmarks. As shown in Figure 10(a), we can reduce the sample variance from 0.0261 to 9.70×10^{-05}, which indicates a 2500× increase of the convergence rate.



(a) (b)

Figure 10: (a) Convergence plot of SSF estimation and (b) the impact of spatial uncertainty on SSF.

We then verify the importance to capture the uncertainty in the attack process with a probabilistic model. We use memory read and write benchmarks and evaluate the impact of the spatial uncertainty. We assume the actual attack location follows a uniform distribution over the neighboring region of the target location. As shown in Figure 10(b), by reducing the radius of the neighboring region (measured by the logic depth), SSF increases by up to 40×. This indicates if one ignores the uncertainty in the attack process, the system vulnerability for different attack techniques can be significantly over-estimated.

4.4 Research Impact

My Ph.D. study so far has led to **7 first-authored publications** [3, 7, 12, 13, 15–17] in premier EDA/CAD journals and conferences. The three proposed algorithms are built upon rigorous mathematical modeling and can collaborate with each other to prevent cross-stage violations. Our split manufacturing algorithm can harness the fabrication technology advances for Trojan prevention, which achieves much better security guarantee and efficiency. The importance of IP protection against reverse engineering is recognized by both leading industrial companies, e.g., Mentor Graphics, and academia. We pioneer the research in the area and get the **Best Paper Award** in the premier hardware security conference, International Symposium on Hardware Oriented Security and Trust in 2017. Our fault evaluation algorithm has been prototyped by a leading IP vendor, **Arm Inc.**, and applied on commercial products. Our endeavor on IP protection is well recognized by the academia and we receive the **gold medal** in ICCAD student research competition, the **best poster award** in ASP-DAC student research forum and the **best poster award** in Florida Institute for Cybersecurity Research (FICS) Conference.

5. REFERENCES

[1] L. Frontier Economics Ltd, "Estimating the global economic and social impacts of counterfeiting and privacy," 2011.

[2] F. Imeson, A. Emtenan, S. Garg, and M. V. Tripunitara, "Securing computer hardware using 3D integrated circuit (IC) technology and split manufacturing for obfuscation," in *Proc. USENIX Security Symposium*, 2013, pp. 495–510.

[3] M. Li, B. Yu, Y. Lin, X. Xu, W. Li, and D. Z. Pan, "A practical split manufacturing framework for trojan prevention via simultaneous wire lifting and cell insertion," in *Proc. Asia and South Pacific Design Automation Conf.*, 2018, pp. 265–270.

[4] "International technology roadmap for semiconductor 2014," Available: https://www.itrs.net/, [Online; accessed Nov. 2014].

[5] Y. Xie and A. Srivastava, "Mitigating SAT attack on logic locking," in *Proc. Int. Conf. on Cryptographic Hardware and Embedded Systems*, 2016, pp. 127–146.

[6] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "CamoPerturb: Secure IC camouflaging for minterm protection," in *Proc. Int. Conf. on Computer Aided Design*, 2016, pp. 29:1–29:8.

[7] M. Li, K. Shamsi, T. Meade, Z. Zhao, B. Yu, Y. Jin, and D. Z. Pan, "Provably secure camouflaging strategy for IC protection," in *Proc. Int. Conf. on Computer Aided Design*, 2016, pp. 28:1–28:8.

[8] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. on Hardware Oriented Security and Trust*, 2015, pp. 137–143.

[9] M. Tunstall, D. Mukhopadhyay, and S. Ali, "Differential fault analysis of the advanced encryption standard using a single fault," in *Proc. Int. Workshop on Information Security Theory and Practices*, 2011.

[10] A. Nahiyan, K. Xiao, K. Yang, Y. Jin, D. Forte, and M. Tehranipoor, "AVFSM: a framework for identifying and mitigating vulnerabilities in FSMs," in *Proc. IEEE/ACM Design Automation Conf.*, 2016.

[11] B. Yuce, N. F. Ghalaty, and P. Schaumont, "TVVF: Estimating the vulnerability of hardware cryptosystems against timing violation attacks," in *Proc. IEEE Int. Symp. on Hardware Oriented Security and Trust*, 2015.

[12] M. Li, B. Yu, Y. Lin, X. Xu, W. Li, and D. Z. Pan, "A practical split manufacturing framework for trojan prevention via simultaneous wire lifting and cell insertion," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2018, (submitted).

[13] M. Li, K. Shamsi, T. Meade, Z. Zhao, B. Yu, Y. Jin, and D. Z. Pan, "Provably secure camouflaging strategy for IC protection," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2018, (accepted).

[14] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "AppSAT: Approximately deobfuscating integrated circuits," in *Proc. IEEE Int. Symp. on Hardware Oriented Security and Trust*, 2017, pp. 95–100.

[15] M. Li, Y. Wang, and M. Orshansky, "A monte carlo simulation flow for seu analysis of sequential circuits," in *Proc. IEEE/ACM Design Automation Conf.* ACM, 2016, p. 44.

[16] M. Li, L. Lai, V. Chandra, and D. Z. Pan, "Cross-level monte carlo framework for system vulnerability evaluation against fault attack," in *Proc. IEEE/ACM Design Automation Conf.*, 2017, pp. 17:1–17:6.

[17] M. Li, J. Miao, K. Zhong, and D. Z. Pan, "Practical public puf enabled by solving max-flow problem on chip," in *Proc. IEEE/ACM Design Automation Conf.* ACM, 2016, p. 164.