National Centre of Excellence
CYBERSECURITY TECHNOLOGY AND ENTREPRENEURSHIP

इलेक्ट्रॉनिकी एवं सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF ELECTRONICS AND INFORMATION TECHNOLOGY
सत्यमेव जयते

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

A Primer on

Logic Locking Techniques

Contributors

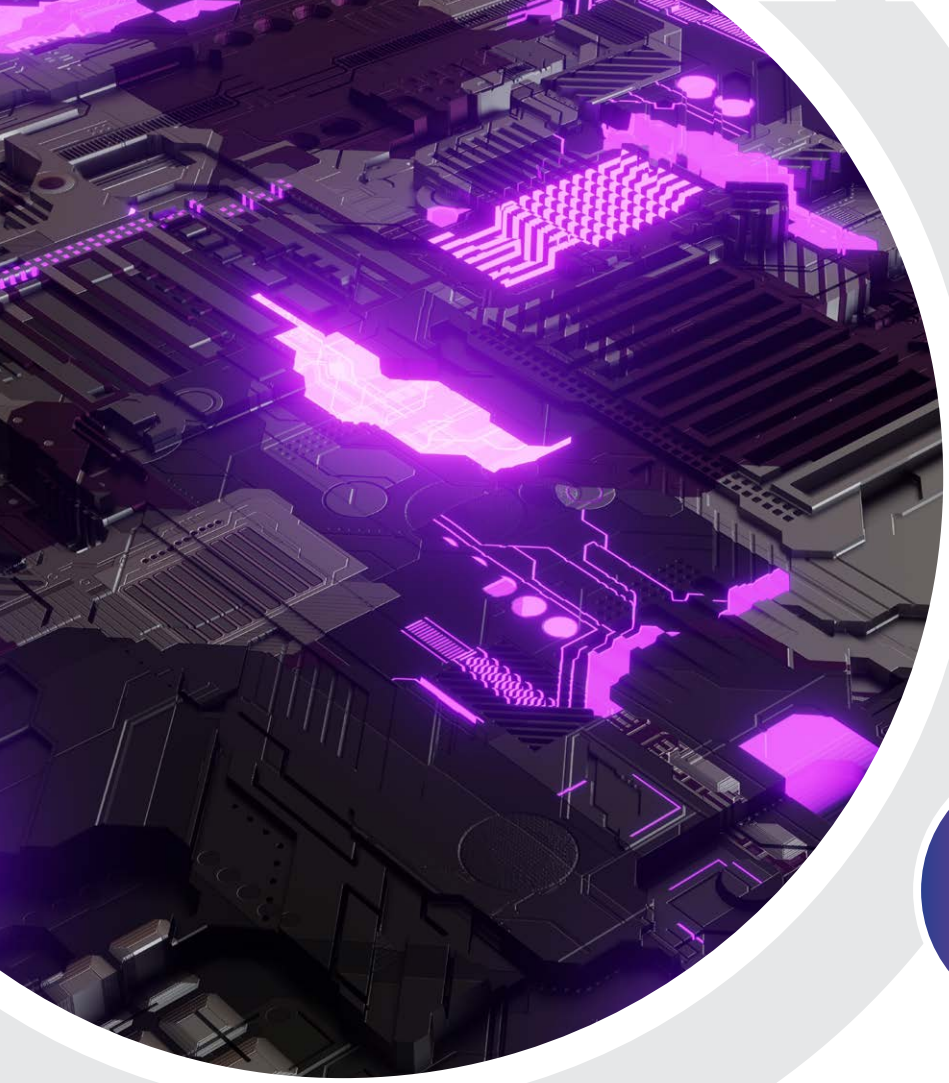
Bodhisatwa Mazumdar
Indian Institute of Technology Indore
bodhisatwa@iiti.ac.in

Soma Saha
Prestige Institute of Engineering Management and Research, Indore
ssaha@piemr.edu.in

Table of CONTENTS

Abstract

Logic locking is a hardware security mechanism for circuit design concealment, especially protecting intellectual property in modern System-on-Chip (SoC) architectures from threats such as reverse engineering, IP piracy, overproduction, and unauthorized activation through- out the integrated circuit (IC) manufacturing supply chain. Over the past decade, extensive research has been conducted on the applicability, feasibility, and efficacy of logic locking, focusing on key areas such as security metrics, abstraction levels, threat models, resiliency against physical attacks, tampering resistance, and the impact of machine learning on both attacks and defenses. Despite continuous advancements, the security of existing logic locking techniques remains a challenge, as sophisticated attacks, both logical and physical, continue to evolve alongside the proposed countermeasures. This paper provides a comprehensive classification of logic locking strategies, including their underlying principles, assumptions, and evaluation metrics. By analyzing the latest attack and defense techniques, we highlight best practices for IP protection and outline future research directions to enhance the robust- ness of logic locking. This work serves as a valuable reference for IP vendors, SoC designers, and researchers, offering insights into the latest developments and critical aspects of logic locking for hardware security.



01

Introduction

In the present day semiconductor industry, multiple phases of VLSI design supply chain are outsourced for manufacturing complexity and cost reasons. In addition, huge recurring costs of fabrication house maintenance and troubleshooting, accelerating IC supply chain flow along with aggressive time-to-market, and emergence of cutting edge technologies have been added on to the cause of outsourcing which led to ushering of separate entities fulfilling various stages of IC design flow, such as IC fabrication, chip testing and packaging, and IC integration, enforcing a globally distributed chain [15]. Over the past two decades, most semiconductor companies have transformed to fabless mode, whereas chip manufacturing, testing, and assembly are performed at specialized providers across the globe. While preventing the substantial costs of maintaining and upgrading own foundries, new threats arise when designs are sent to offshore fabrication houses. Such outsourcing introduces new risks, since the foundry naturally sees the full details of the

design, including the netlist (a description of the hardware nodes and their connectivity) and physical layout information. An unscrupulous foundry can use this information to steal the design's intellectual property (IP), overproduce unauthorized chips, or introduce hardware Trojans. This exposure of sensitive design details makes hardware security a critical concern in modern semiconductor manufacturing. Various countermeasures, such as logic obfuscation, split manufacturing, and hardware watermarking, have been proposed to mitigate these risks and protect IC designs from malicious foundry activities. Integrated circuits (ICs) become susceptible to overproduction, counterfeit, and reverse engineering [5].

In the post pandemic market, as the market demand of the ICs is far more than the foundry production capacity, it is resulting into tremendous shortage of IC supply, thus making the role of chip-makers such as Taiwan Semiconductor Manufacturing Company (TSMC), United Micro- electronics Corporation (UMC), and Semiconductor Manufacturing International Corporation (SMIC) even more prominent. This exposes geographically distributed VLSI design industry to threats, such as piracy through IP theft and subsequent reselling in the black market. In the post-pandemic era, such a market with an unprecedented demand results in a more panic IC design, implementation, manufacturing, and testing by original equipment manufacturers (OEM). Moreover, these steps are executed precariously to steal the market contracts. Hence, with lesser precautions taken by the OEMs in order to meet the market demand, and by facing increased globalization, the IP vendors and OEMs face a significant depletion of the control capability and monitoring efficacy over the supply chain.

In order to mitigate threats associated with IC supply chain, many variants of design-for- trust countermeasures have been proposed over the years. Some of these countermeasures com-prise watermarking [7], IC metering [2], IC camouflaging [11], and hardware obfuscation [8]. In comparison to these countermeasures, logic locking has drawn immense interest from the scientific research community as well as industry over the past two decades through design of robust solutions in different levels of abstraction in the VLSI design flow. Logic locking en- ables the IP/IC designers to provide limited post-fabrication programmability to the fabricated designs, thereby concealing the underlying functionality. This prevents unauthorized access, overproduction, and reverse engineering. By integrating additional logic gates and key-based activation mechanisms, logic locking ensures that only authorized users with the correct key can unlock the intended functionality of the design. This technique enhances hardware security by making it significantly harder for adversaries, including untrusted foundries and end-users, to infer the true behavior of the circuit without proper authentication. The functionality of the locked circuit is determined by a secret key used in logic locking, which is exclusively known to trusted entities such as IP owners or original component manufacturers (OCMs). By inputting the correct key value, the design house can successfully unlock and enable the circuit.

In the firstly proposed lock-and-key technique, a key based scheme was introduced, which added randomness to the design-for-test (DfT) circuits or chains when accessed by an unautho- rized user. Over the past two decades, the randomization and corresponding programmability is applied to combinatorial circuit [12, 10], sequential circuit [3, 4], and behavioral design [17]. With the increased adaptability of logic locking

techniques, a large variant of these techniques are now considered for academic research as well in semiconductor industries, which comprise Mentor Graphic's TrustChain platform enabled by logic locking and the latest Defense Ad- vanced Research Projects Agency (DARPA) project on Automatic Implementation of Secure Silicon (AISS) [1].

Research shows that logic locking (LL) has emerged as a promising defense against IP piracy and IC overproduction. However, the development of logic de-obfuscation attacks by white-hat researchers have led to an important role in advancing this security paradigm. These attacks help identify weaknesses in existing LL countermeasures, distinguishing between robust and vulnerable techniques while guiding future research. Over the past two decades, exten- sive studies have explored logic locking from both defensive and offensive perspectives, leading to an ongoing evolution where attacks and countermeasures continue to grow in sophistica- tion [6, 16, 9, 20]. Moreover, as time progressed, the ever-growing cutting edge technologies, for example, failure analysis (FA) equipments, emergence of machine learning algorithms [13], and undetected infiltration of the adversaries into trusted semiconductor manufacturing facil- ities [19, 14] demonstrate that there is long way to go to freeze the secure LL algorithms for industrial applications [18].

02

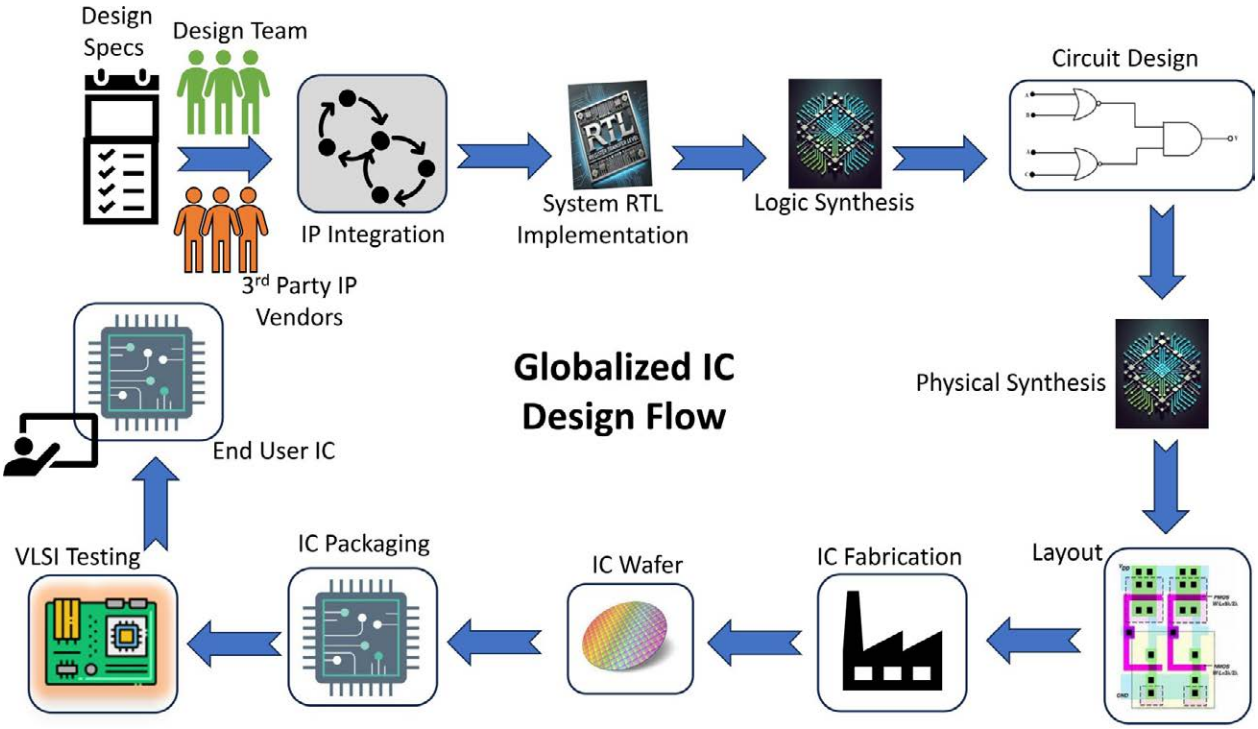
Assumptions in Logic Locking Security

The main important steps of IC design flow is shown in Figure 3, which starts from the design specifications and involves multiple parties in the process of IC manufacturing. The various stages in this process are outsourced to various 3rd party IP vendors through which the OEMs have the least reliable control on their IP circuitry resulting in introduction of contracted offshore vendors as untrusted parties.

Considering a given IC representation which may comprise integrated design, synthesized netlist, chip layout, or an IC under test, its functionality and design can be reverse-engineered. A set of malicious end users can resort to physical reverse engineering which entails circuit reconstruction. In case of a successful reconstruction, the malicious untrusted entity can steal IP design or illegally overproduce it to mount losses on the design house. To secure a VLSI circuit design against malicious players in the fabrication

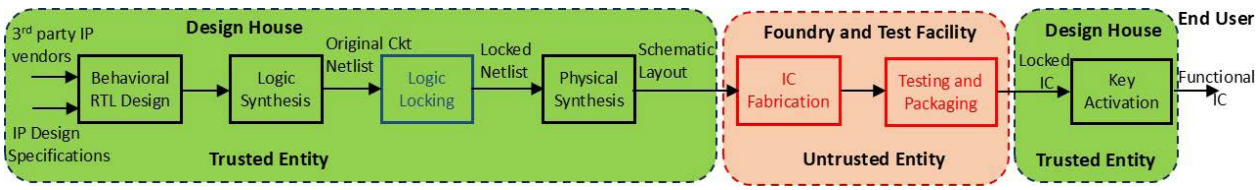
chain, protection mechanisms such as logic locking have been proposed. Logic locking is a mechanism to protect digital circuits from unauthorized access, reverse engineering circuit design, and intellectual property theft. It involves inserting additional logic gates or combinatorial blocks, referred to as "key gates", into a synthesized circuit. These key gates prevent the circuit from functioning correctly unless the gate is sensitized with the correct key value is provided. Logic locking has gained considerable emphasis in recent years due to its efficacy in enhancing trust for hardware designs, particularly with the advent of the growing concerns over IP piracy and IC overproduction.

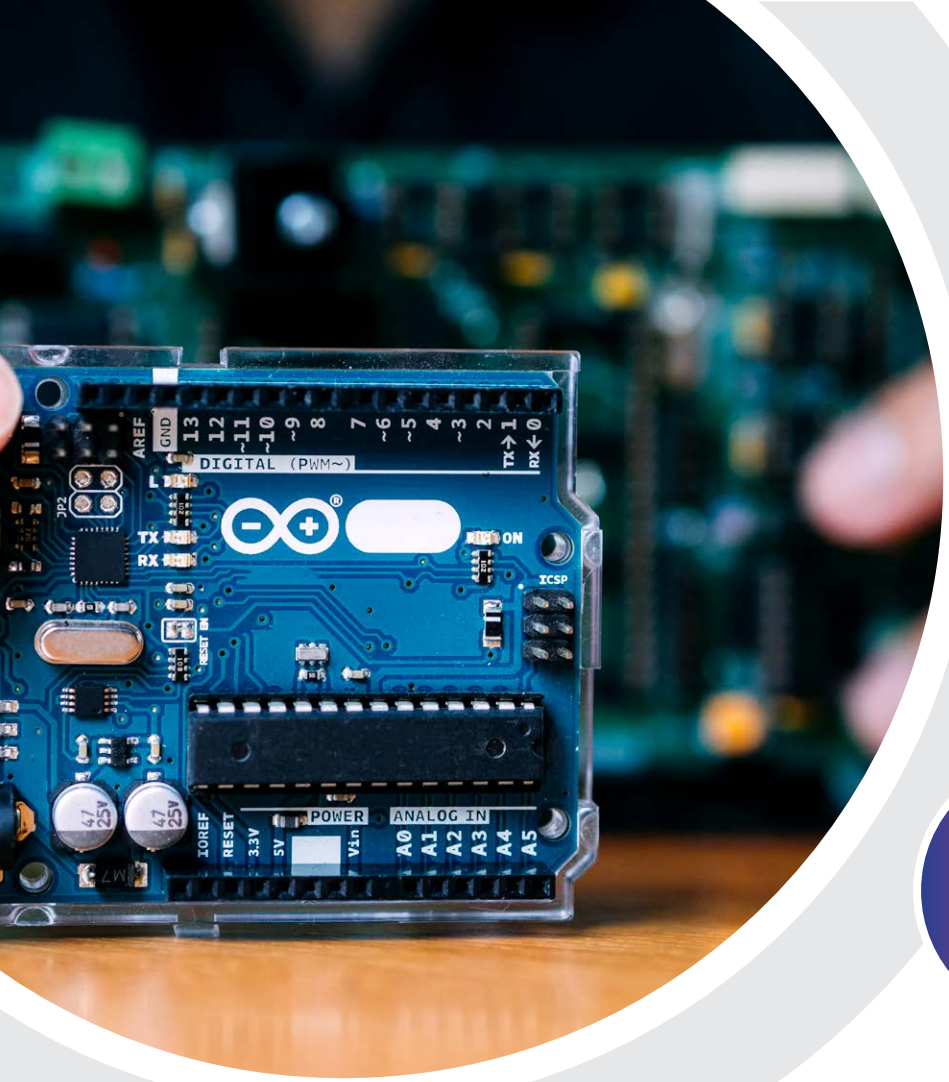
Figure 1: Conventional IC Design flow across the world.



This write-up explores the concept, implementation, types, challenges, and future directions of logic locking. With increasing complexity of digital integrated circuits (ICs), along with the rise in hard- ware piracy and malicious tampering, hardware trust has become a critical issue in the circuit design community and authentic end users. Logic locking, which is one form of hardware obfus- cation, is a promising countermeasure to prevent unauthorized access and reverse engineering of digital circuits. The idea behind logic locking is to modify a circuit in such a way that it cannot function as intended unless the correct key is applied to the key gate. This technique aims to protect intellectual property (IP), secure cryptographic designs, and prevent hardware-based attacks like those exploiting vulnerabilities in the manufacturing or supply chain process. The incorporation of logic locking mechanism in the IC design flow is shown in Fig. 2.

Figure 2: Embedding logic locking in IC Design flow.





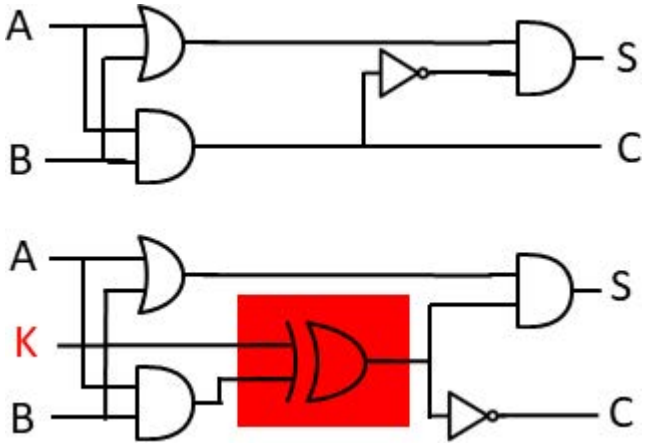
03

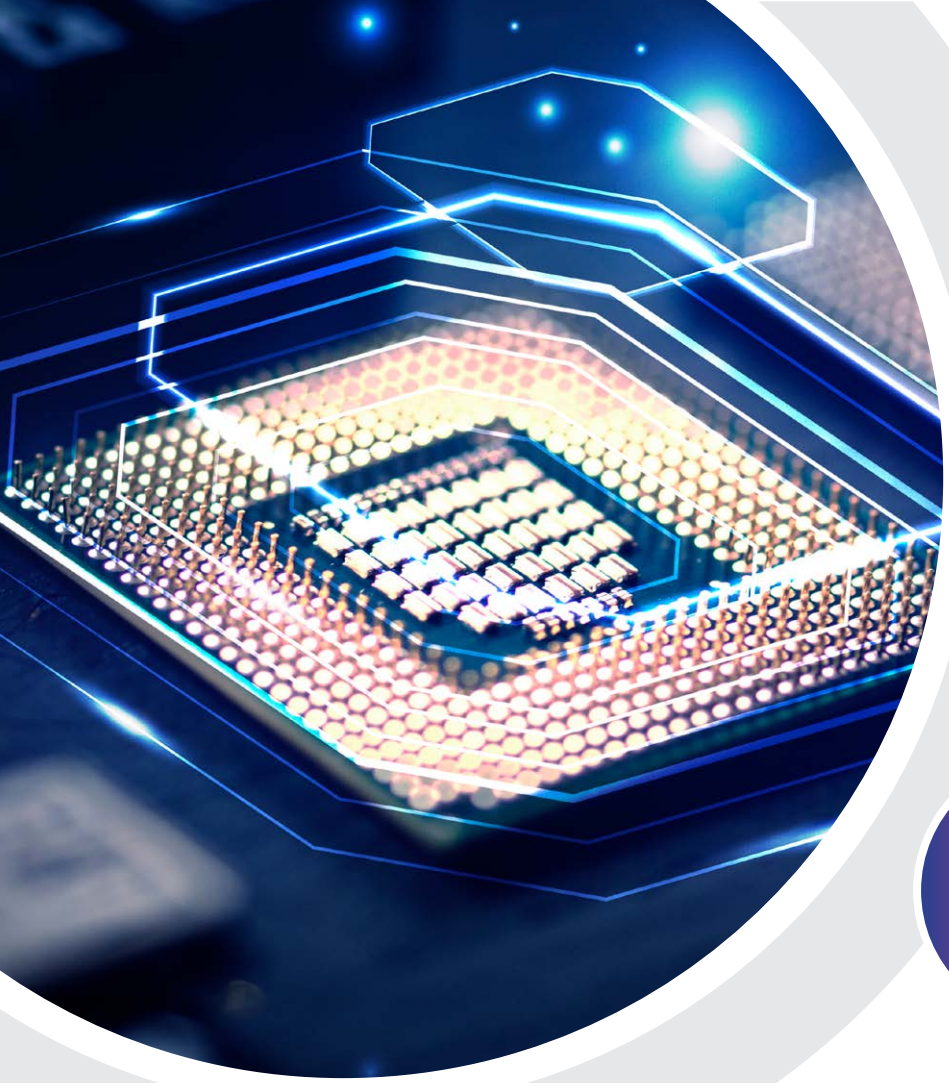
Working Principle

In its simplest form, logic locking involves the insertion of additional logic gates that require a secret key to allow the circuit to function correctly. These additional gates are typically inserted into the combinational logic portion of the design, often referred to as the "locked" portion. The correct key value unlocks the circuit, thus restoring the functionality of the circuit. Without the correct key, the circuit will output incorrect results and hence rendered useless. As shown in Figure 4, there are various techniques to implement logic locking, such as using XOR-based gates or random key bits that alter the behavior of the circuit. Typically, these inserted gates are cryptographically secure, which implies that even if an attacker attempts to reverse engineer the design, the presence of these gates and the secret key will make it computationally difficult to deduce the correct working of the circuit. Logic locking can be implemented at different abstraction layers of VLSI design cycle as shown in Table 1. An VLSI circuit implementation comprises various levels, for instance, high level synthesis (HLS), register-transfer level (RTL), gate level, transistor level, and layout level. For each of these levels, the implementation effort and

resource overhead for the specific logic locking varies. In general, propagating from layout level to RTL or HLS level mitigates the implementation effort. However, at a lower level of abstraction, proposing constructions for a logic locking countermeasure incurs lower overhead. Moreover, proposing logic locking mechanism at higher abstraction levels, such as RTL or HLS, provides immunity against a set of insider threats. In existing literature, more than 90% of the proposed logic locking schemes are implemented at the gate level, and majorly performed in the post-synthesis stage in the gate-level netlist in the IC supply chain.

Figure 3: An original design (top) and its logic locked circuit [12].





04

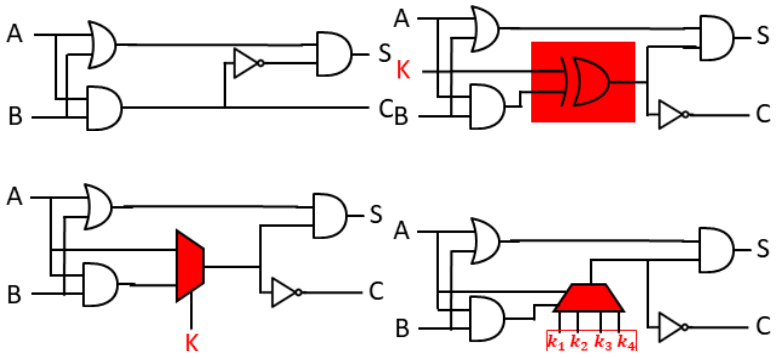
Logic Locking Techniques: Aspects and Variants

The first proposed logic locking scheme was termed EPIC. In this scheme, the locking circuitry comprising XOR (or XNOR) and inverter gates, were randomly inserted into the original combinational circuit. Incorrect key bits result in bit flips in signals, thus leading to faulty computations. XOR-based technique involves inserting XOR gates in the circuit between the primary inputs and the outputs of the circuit. The key bit values control the output of the XOR gates. If the correct key is provided, the XOR gates behave as expected, rendering the circuit to perform its intended function. If the key is incorrect, the XOR gates modify the logic and prevents producing the correct output. The secret correct key is stored in a tamper-proof memory. EPIC triggered considerable follow-up work on logic locking schemes which ushered new schemes with improved placement of the locking circuitry. Rajendran et al. discovered that if key gates are inserted at random,

different parts of the locking circuitry might influence each other and potentially interfere or cancel each other's effect, thus weakening the security of the design. As a result, they presented an algorithm that only inputs a small portion of gates at random while finding optimal positions for the remaining gates.

As an improvement, in fault-based logic locking (FLL), fault simulation techniques were proposed in order to model the effect that a faulty key exhibits on the overall functionality. Henceforth, depending of the position of the locking circuitry, an injected fault might not be propagated to the outputs. Hence, identifying positions with a good fault impact is beneficial for the design to be locked. This strategy has often been used to optimize different locking schemes. Subsequently, Strong Logic Locking (SLL) was proposed. In this scheme, the logic locking circuitry is inserted such that key bits cannot be computed from the observed IC output, thwarting any reduction of the complexity of brute-force attacks. With an amalgamation of the ideas presented above, Karmakar et al. proposed an algorithm to determine the optimal position for the locking circuitry which results in maximal resilience againts recovering the key. Their logic locking scheme is based on fault analysis approach, while simultaneously determining a part of the key gates were made according to the rules of SLL.

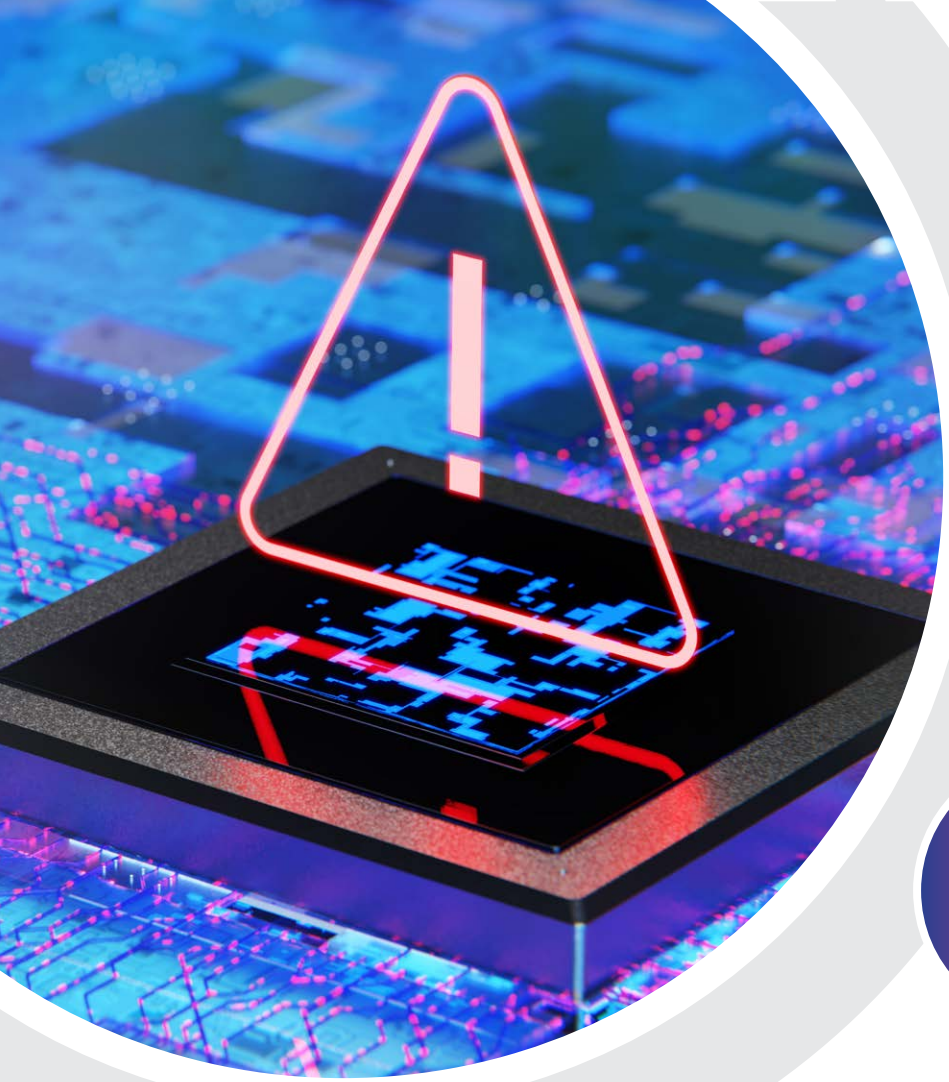
Figure 4: (a) Original circuit, (b) XOR-based logic locking, (c) MUX-based logic locking, (d) LUT based logic locking.



All these primitive techniques are mostly XOR-based and implemented at the gate level. Since a locked circuit initiated with an incorrect key corrupts the primary outputs by propagat- ing errors at primary outputs, in SLL and FLL, certain characteristics of automatic test pattern generation (ATPG) tools and testability specifications, such as controllability/observability, and faults propagation/masking have been used for selecting the location of XOR-based key gates. For instance, in SLL, specifications comprise key-gates exclusion, isolation, cascading (running), mutability, and convergence have been examined. This resulted in forming an interference graph of key gates. The locations that maximizes these characteristics are selected for key gate in- sertion, thus helping to enhance the strength of the logic locking against VLSI testing-based attacks in comparison with RLL.

In most of the logic locking techniques, all ICs of a circuit design are unlocked with the same key. As a result, if the information about the correct key is leaked, all other instances can be unlocked immediately. To thwart this single-point-of-failure, a key preprocessor can be deployed. This preprocessor module precedes the locking circuitry and derives

the key to unlock an IC from a different key that is given to the IC. Through IC's device-unique signatures, for example, derived through a physical unclonable function (PUF), the input to the key preprocessor is unique for each manufactured IC, while its output is the same for all ICs. In this way, despite the fact that a logic locking scheme possess a global internal key, every IC contains its own individual chip key that is a function of the embedded PUF response. If the correct key value of one IC is used to unlock a different IC, the key preprocessor would thus compute an incorrect internal key, forcing the design to remain locked. As a result, even if an adversary obtained the internal key, other ICs would still remain unlocked.



05

Challenges and Issues in Logic Locking

One of the most pertinent challenges in logic locking schemes are their security vulnerabilities. Despite being a robust security measure, logic locking schemes are not invulnerable. Researchers have developed methods to break or weaken many existing logic-locking techniques. For ex- ample, attacks such as SAT solvers, machine learning techniques, and brute force can be used to recover the key and bypass the locking mechanism. Moreover, effective key management is crucial for the success of logic locking. The key must be stored securely, which introduces chal- lenges in ensuring that the key is not leaked or exposed during manufacturing, distribution, or use. In addition, managing large numbers of keys in complex circuits can become cumbersome.

Table 1: Logic locking specifications at different abstraction layers of VLSI Design Cycle

Circuit	Granularity	Overhead	Implementation Effort
Layout level	bitwise, wiring	Close to zero	High
Transistor level	bitwise, wiring,	Small	High
Gate level	switching logical,	Variant	Medium
RTL level	bitwise behavioral,	mid-high	low
High level (HLS)	operational, bitwise behavioral, operational, bitwise	mid-high	low

Logic locking schemes incurs significant overhead in design and performance metrics of the fabricated VLSI circuits. Logic locking introduces additional gates into the design, which can lead to increased area, power consumption, and delay. The impact on the performance and size of the final design can be significant, especially when the locking scheme is complex. Thus, a trade-off must be made between security and the additional overhead imposed on the design. In terms of manufacturing variability, the insertion of locking mechanisms can introduce issues during the manufacturing process, such as faults due to variability in the production line. This can affect the reliability of the circuit and create additional concerns regarding yield and quality control.

There exists several threat models on logic locking schemes which are illustrated in Table 2. The original design house is considered trusted. The fabrication house and manufacturer with packaging , test facility. The malicious entity in the foundry aims to overproduce the ICs, i.e., fabricate more ICs than ordered or authorized by the design house and if possible sell them illegally on the black market. In the case of a design on which logic locking has been applied, the attacker's goal is to bypass the security approach. In other words, the adversary attempts to discover the value of the key for unlocking the ICs, which are based on reverse engineering and ad-hoc attacks. Logic locking prevents the recovery of the functionality with reverse engineering, whether the attacker is the SoC integrator, the manufacturer or an end- user. In general, the attacker is assumed to have an access to the logic locked netlist, possibly obtained from reverse engineering of the GDSII, masks, or from a procured IC. Moreover, the attacker in the form of an end user can get an access to a functional IC, which can be legally purchased and properly unlocked or activated. In addition, the attacker can distinguish regular and key circuit inputs in order to simulate the locked netlist with chosen data. The key bits are connected to a tamper-proof memory.

5.1 SAT Attack (Satisfiability Attack)

The SAT attack targets the satisfiability of the locked circuit's Boolean equations. To break the logic lock, the attacker needs to figure out the correct key that unlocks the circuit. The locked circuit can be represented as a Boolean satisfiability (SAT) problem, where the circuit's logic is formulated in terms of a system of equations with variables corresponding to the lock bits.

5.1.1 Process of a SAT attack

Obtain the locked netlist: The attacker gains access to the locked circuit or netlist (the network representation of the gates). Translate the netlist to a SAT problem: The attacker

then trans- lates the locked circuit into a Boolean formula. The locked gates introduce additional variables that represent the key bits. The solving process of the SAT problem is defined as follows. The attacker uses a SAT solver to solve the system of equations. The SAT solver tries to find an as- signment for the variables (key bits) that satisfies the circuit's constraints (the logic functions). The next step involves determining the key. If the SAT solver successfully finds a solution, it reveals the correct key that unlocks the circuit. The key point is that The SAT attack relies on the computational complexity of SAT solvers. Modern SAT solvers are very efficient and can solve many real-world instances quickly, making the attack a significant threat to logic locking schemes.

Table 1: Logic locking specifications at different abstraction layers of VLSI Design Cycle

Attack Scenario	Malicious entity involved				Adversary activity
	3rd Party IP Vendor	SoC Integrator	Manufacturer	End User	
1	Trusted	Defender	Adversary	Untrusted	Overproduction
2	Defender	Trusted	Adversary	Untrusted	IP Overuse
3	Defender	Adversary	Untrusted	Untrusted	IP Overuse
4	Trusted	Defender	Adversary	Untrusted	Reverse Engineering
5	Defender	Trusted	Adversary	Untrusted	Reverse Engineering
6	Defender	Adversary	Untrusted	Untrusted	Reverse Engineering
7	Trusted	Defender	Untrusted	Adversary	Reverse Engineering
8	Defender	Trusted	Untrusted	Adversary	Reverse Engineering
9	Untrusted	Untrusted	Defender	Adversary	Reverse Engineering

5.1.2 Defense Mechanisms Against SAT Attacks

Given the power of SAT attacks, several countermeasures have been developed to protect logic locking schemes. The first attempt to mitigate Boolean satisfiability (SAT) based attacks is the construction of point function that minimizes the number of available input patterns that prunes out incorrect key values. The SAT attack adopts a fast convergence approach in ruling out the incorrect key values. the point function-based constructions of logic locking techniques have been shown to be provable secure, which implies that such constructions are algorithmically resilient against any variant of input-output query based attacks. The point function-based logic locking schemes which have been shown to be resilient against SAT attacks are shown in Table 3. One of the first logic locking techniques in this category are SARLock and Anti-SAT. The main structure of point function is based on a flipping circuitry that flips or corrupts a set of primary outputs for only certain number of input patterns provided to the primary inputs. Moreover, in these circuits, a masking or restore logic block has been included that again reflips the corrupted bits

when activated with the correct key value, thereby guaranteeing correct primary output value in this case.

The next method is that of adding redundancy. Adding redundant logic or using more complex lock gates makes the SAT problem harder to solve. The next method is that of incorporating obfuscation. Introducing non-linear or obfuscated logic gates can make it difficult for SAT solvers to map the circuit's functionality. In addition, key diversification can also be used as a protection countermeasure. Using multiple keys or different key types for different parts of the circuit increases the complexity of the attack. Moreover, recent research has focused on Anti-SAT techniques. These include techniques like key encoding or modifying the Boolean formula in ways that increase the search space for the SAT solver.

Table 3: Different types of point function based logic locking techniques

Logic locking scheme	Mechanism of operation
SARLock	Adds flipping circuit to corrupt only one input pattern for all possible incorrect keys and a masking circuit for correct key.
AntiSAT	Merging of two ANDed complementary functions (g and g) as combination of the flipping and masking circuitry.
AND-Tree	Hard coded AND tree as flipping circuits and generic masking circuitry
TTLock	SARLock and stripped original circuit for one minterm
SFLL-HD	SARLock and stripped original circuit for d minterms, where d = (k), h=Hamming distance, k=Key size.
SFLL-flex	Adding the flexibility of protecting user-defined input patterns in a point function manner and a generic restore circuitry.
SFLL-rem	Removing logic for creating the corruption-based on fault insertion and a generic restore circuitry
G-AntiSAT	Merging of ANDed two toggled functions (f and g) as the flipping and masked circuit together
CASLock	Variant of AND-OR tree as corruptible circuitry under incorrect key

5.2 Differential Attack

A differential attack on logic locking is a type of side-channel attack used to break or bypass the protection mechanisms implemented in logic locking schemes. Logic locking, also known as circuit obfuscation, is a technique designed to protect intellectual property (IP) in integrated circuits (ICs) by inserting additional logic gates or obfuscating the circuit such that the correct functionality of the circuit is hidden unless a secret key is provided. In this context, a differential attack aims to exploit the behavior of the circuit under different inputs or conditions to deduce the secret key used for the locking mechanism. A differential attack on logic locking attempts to deduce the secret key by analyzing the differences in the circuit's output when various inputs are applied. By comparing the circuit's behavior with different sets of inputs, an attacker might be able to identify patterns or key-related information that reveal the locked key.

The steps in a Differential Attack on Logic Locking is as follows. The first step is Circuit Analysis. In this stage, the adversary observes the circuit's behavior for various input values. This involves running the circuit with different inputs and observing the outputs, with or without the key being applied. The next step is that of key guessing. The adversary starts by guessing possible key values or using a key search strategy. It then simulate or test the locked circuit for specific inputs, comparing the outputs with the expected behavior. For each guessed key, the attacker tracks how the outputs differ when different input sets are applied. For these different inputs, the adversary observes the differential patterns at the corresponding primary outputs of the circuit.

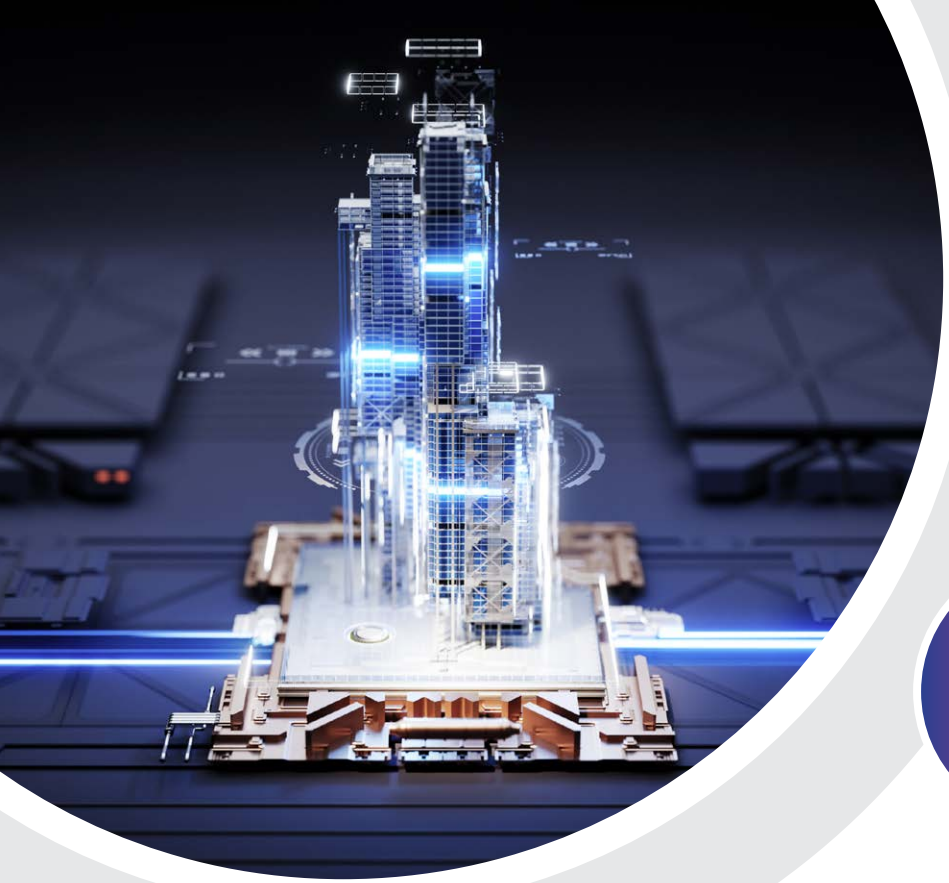
The differential attack is based on the idea that the differences in output when applying specific input combinations can reveal useful information about the secret key. The attack looks for differential patterns, i.e., the difference in output when the circuit is operated with different inputs that could indicate which parts of the logic are key-dependent. For the recovery of the embedded secret key, through careful analysis of the output differences is performed. The attacker might be able to reduce the possible key candidates or, in some cases, directly recover the entire secret key. If the attacker manages to collect enough differential output data, they can often reconstruct the key.

The mitigation strategies ask for stronger obfuscation techniques. To thwart differential attacks, stronger and more complex logic locking schemes can be used, such as inserting more diverse logic gates or employing more sophisticated key management schemes. Moreover, in-corporating error detection and correction codes into the design can prevent small differential changes from revealing key information. Using randomized or dynamic logic locking where the locked circuit behaves differently for each key and input combination can make it harder for attackers to predict patterns. In addition, designing circuits to resist power and fault analysis attacks can reduce the effectiveness of differential power or fault analysis.

06

Applications of Logic Locking

Intellectual Property Protection: One of the primary applications of logic locking is in pro- tecting the IP of digital designs. By adding an extra layer of security to the circuit, designers can prevent unauthorized replication, reverse engineering, or modification of their proprietary designs. From the perspective of cryptographic hardware, logic locking is particularly useful in securing cryptographic circuits, such as those used in secure communication devices, hardware security modules (HSMs), and trusted execution environments (TEEs). These circuits often store sensitive information, and logic locking ensures that the design remains secure even if an attacker has access to the physical hardware. For supply chain security, the hardware supply chain is increasingly becoming vulnerable to tampering and counterfeiting. Logic locking can protect circuits from being altered or cloned during the manufacturing process, ensuring that only authentic, secure hardware is used in critical systems. Preventing reverse engineering of proprietary circuits is also one of the strong applications of logic locking. It can thwart reverse engineering efforts that aim to copy and redistribute proprietary hardware designs, helping to safeguard the investments of companies that develop unique hardware solutions.



07

Future Directions

Considering the post-silicon threat mitigation, as the ability of adversaries to perform sophisticated post-silicon attacks grows, new logic locking mechanisms are being developed to resist these advanced techniques. Future work is expected to focus on designing lock schemes that can defend against new threats and attacks. Moreover, incorporating integration with other security measures, logic locking may be combined with physical unclonable functions (PUFs), encryption, and tamper detection mechanisms, to offer multi-layered protection for hardware systems. This will provide a more robust security architecture for modern ICs.

In addition, with the advent of quantum computers in future and its applicability towards breaking logic locking schemes, quantum-safe logic locking also holds a great scope of securing IC supply chain. With the advent of quantum computing, traditional cryptographic techniques could become vulnerable. Researchers are exploring quantum-safe approaches to logic locking to ensure that future hardware systems remain secure even in the face of quantum-powered attacks. In the context of performance optimization, future research work will also focus on improving the efficiency of logic locking schemes, minimizing performance overheads, and reducing the impact on power, area, and delay. Construction techniques for adaptive or selective locking, where only the most critical portions of the circuit are locked, are also being explored to mitigate the trade-offs.

Conclusion

Logic locking represents a significant advancement in securing digital circuits and protecting hardware intellectual property. While the technique provides substantial benefits, it also faces challenges, including potential vulnerabilities, key management issues, and performance over- heads. As hardware security continues to be a top priority in the industry, the development of more robust and efficient logic locking methods will be crucial for defending against evolv- ing threats. Through continued research and innovation, logic locking can evolve to meet the growing demands of modern secure hardware design.

References

[1] D.P. Affairs. DARPA Selects Teams to Increase Security of Semiconductor Supply Chain. https://www.darpa.mil/news/2020/semiconductor-supply-chain-security#:~:text=Two%20teams%20of%20academic%2C%20commercial,incorporated%20efficiently%20into%20chip%20designs., 2020. Accessed: March 12, 2025.

[2] Yousra M. Alkabani and Farinaz Koushanfar. Active hardware metering for intellectual property protection and security. In 16th USENIX Security Symposium (USENIX Security 07), Boston, MA, August 2007. USENIX Association.

[3] Rajat Subhra Chakraborty and Swarup Bhunia. Harpoon: An obfuscation-based soc design methodology for hardware protection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(10):1493–1502, 2009.

[4] Avinash R. Desai, Michael S. Hsiao, Chao Wang, Leyla Nazhandali, and Simin Hall. Inter- locking obfuscation for anti-tamper hardware. In Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, CSIIRW '13, New York, NY, USA, 2013. Association for Computing Machinery.

[5] Ujjwal Guin, Ke Huang, Daniel DiMase, John M. Carulli, Mohammad Tehranipoor, and Yiorgos Makris. Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain. Proceedings of the IEEE, 102(8):1207–1228, 2014.

[6] Clemens Helfmeier, Dmitry Nedospasov, Christopher Tarnovsky, Jan Starbug Krissler, Christian Boit, and Jean-Pierre Seifert. Breaking and entering through the silicon. In Pro- ceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS '13, page 733–744, New York, NY, USA, 2013. Association for Computing Machinery.

[7] A.B. Kahng, J. Lach, W.H. Mangione-Smith, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe. Constraint-based watermarking techniques for de- sign ip protection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20(10):1236–1252, 2001.

[8] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic. Securing scan design using lock and key technique. In 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05), pages 51–62, 2005.

[9] Bodhisatwa Mazumdar, Soma Saha, Ghanshyam Bairwa, Souvik Mandal, and Tatavarthy Venkat Nikhil. Classical Cryptanalysis Attacks on Logic Locking Techniques. J. Electron. Test., 35(5):641–654, 2019.

[10] Jeyavijayan Rajendran, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. Security analysis of logic obfuscation. In DAC Design Automation Conference 2012, pages 83–89, 2012.

[11] Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. Security Anal- ysis of Integrated Circuit Camouflaging. In Proceedings of the 2013 ACM SIGSAC Con- ference on Computer & Communications Security, CCS '13, page 709–720, New York, NY, USA, 2013. Association for Computing Machinery.

[12] Jarrod A. Roy, Farinaz Koushanfar, and Igor L. Markov. Epic: Ending piracy of integrated circuits. In 2008 Design, Automation and Test in Europe, pages 1069–1074, 2008.

[13] Dominik Sisejkovic, Farhad Merchant, Lennart M. Reimann, Harshit Srivastava, Ahmed Hallawa, and Rainer Leupers. Challenging the Security of Logic Locking Schemes in the Era of Deep Learning: A Neuroevolutionary Approach. J. Emerg. Technol. Comput. Syst., 17(3), May 2021.

[14] Andrew Stern, Huanyu Wang, Fahim Rahman, Farimah Farahmandi, and Mark Tehra- nipoor. ACED-IT: Assuring Confidential Electronic Design Against Insider Threats in a Zero-Trust Environment. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(10):3202–3215, 2022.

[15] Can Sun and Thomas Rose. Supply chain complexity in the semiconductor industry: Assessment from system view and the impact of changes. IFAC-PapersOnLine, 48(3):1210– 1215, 2015. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[16] Huanyu Wang, Domenic Forte, Mark M. Tehranipoor, and Qihang Shi. Probing Attacks on Integrated Circuits: Challenges and Research Opportunities. IEEE Design Test, 34(5):63– 71, 2017.

[17] Yang Xie and Ankur Srivastava. Delay locking: Security enhancement of logic locking against ic counterfeiting and overproduction. In Proceedings of the 54th Annual Design Automation Conference 2017, DAC '17, New York, NY, USA, 2017. Association for Com- puting Machinery.

[18] Muhammad Yasin, Bodhisatwa Mazumdar, Jeyavijayan J V Rajendran, and Ozgur Sinanoglu. Sarlock: Sat attack resistant logic locking. In 2016 IEEE International Sympo- sium on Hardware Oriented Security and Trust (HOST), pages 236–241, 2016.

[19] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. Security analysis of anti-sat. In 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), pages 342–347, 2017.

[20] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. Removal attacks on logic locking and camouflaging techniques. IEEE Transactions on Emerging Topics in Computing, 8(2):517–532, 2020.

National Centre of Excellence
CYBERSECURITY TECHNOLOGY AND ENTREPRENEURSHIP

The National Centre of Excellence (NCoE) for Cybersecurity Technology Development has been conceptualized by the Ministry of Electronics & Information Technology (MeitY), Government of India, in collaboration with the Data Security Council of India (DSCI). Its primary objective is to catalyze and accelerate cybersecurity technology development and entrepreneurship within the country. NCoE plays a crucial role in scaling and advancing the cybersecurity ecosystem, with a focus on critical and emerging areas of security.

Equipped with state-of-the-art facilities, including advanced lab infrastructure and test beds, NCoE enables research, technology development, and solution validation for adoption across government and industrial sectors. By adopting a concerted strategy, NCoE aims to translate innovations and research into market-ready, deployable solutions—contributing to the evolution of an integrated technology stack comprising cutting-edge, homegrown security products and solutions.

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

Data Security Council of India (DSCI) is a premier industry body on data protection in India, setup by nasscom, committed to making the cyberspace safe, secure and trusted by establishing best practices, standards and initiatives in cybersecurity and privacy. DSCI brings together governments and their agencies, industry sectors including ITBPM, BFSI, telecom, industry associations, data protection authorities and think-tanks for policy advocacy, thought leadership, capacity building and outreach initiatives. For more info, please visit www.dsci.in

DATA SECURITY COUNCIL OF INDIA

+91-120-4990253 | ncoe@dsci.in

https://www.n-coe.in/

4 Floor, NASSCOM Campus, Plot No. 7-10, Sector 126, Noida, UP -201303

Follow us on

@CoeNational
nationalcoe
nationalcoe
NationalCoE