SRLL: Improving Security and Reliability with User-Defined Constraint-Aware Logic Locking

MONA HASHEMI, School of Electrical and Computer Eng., College of Eng., University of Tehran, Iran and School of Computing, National University of Singapore, Singapore
SIAMAK MOHAMMADI, School of Electrical and Computer Eng., College of Eng., University of Tehran, Iran and School of Computing Science, Institue for Research in Fundamental Sciences (IPM), Iran
TREVOR E. CARLSON, School of Computing, National University of Singapore, Singapore

As chip fabrication costs rise, designers have shifted to a fabless and outsourced development model which opens up the possibility for IP piracy. To address these challenges, logic locking methods modify designs to limit functionality to authorized users that present a valid secret key. However, existing techniques often face limitations in resilience against advanced attacks and do not provide solutions to achieve user-defined constraints and goals. In this paper, we propose SRLL, a user-defined constraint-aware logic locking technique that aims to improve the security and reliability of hardware designs. SRLL bridges the gap between exact and approximate attacks and allows the user to balance the resiliency against satisfiability-based, machine-learning-based, and constant propagation attacks while securing design constraints provided by the user. To enable this, we limit the locking functions to the non-critical path components and insert key gates at specific nodes, introducing a new set of critical parameters specifically designed to prevent target attacks. Finally, we obfuscate the netlist to hide inserted key gates and locking functions. Results show that SRLL maintains strong resiliency by exponentially increasing the required number of distinguishing input patterns, the complexity of finding these patterns, and adding sufficient structural complexity to the design. We evaluate SRLL using ISCAS'85, MCNC'91, and ITC'99 benchmarks, demonstrating resiliency with low overhead against modern attacks, including SAT, AppSAT, OMLA, SAIL, and SCOPE.

CCS Concepts: • **Security and privacy → Malicious design modifications**; **Hardware reverse engineering**.

Additional Key Words and Phrases: IP Piracy, Logic Locking, Hardware Obfuscation.

1 INTRODUCTION

The traditional end-to-end design process for building ASICs is complex and time-consuming. Despite the availability of commercial as well as open-source tools to streamline the generation and validation process, many designers rely on outsourcing to reduce the time to market of their products. However, due to the lack of control in this process, relying on external design services or third-party components introduces a host of security, quality, and reliability concerns. In some cases, electronic circuit design companies may try to steal or replicate commercial designs in order to reduce the time it takes to create a competing product. This is often achieved through reverse engineering, a process where the original design is analyzed and reconstructed to replicate its functionality or extract proprietary information. The result is a compromised hardware design that would lead to device cloning, overproduction, or unauthorized integration into new products. One promising approach to

Authors' addresses: M.Hashemi, School of Electrical and Computer Eng., College of Eng., University of Tehran, Tehran, Iran: hashemi.mona@ut.ac.ir; S.Mohammadi, School of Electrical and Computer Eng., College of Eng., University of Tehran, Tehran, Iran: smohamadi@ut.ac.ir; T.E.Carlson, School of Computing, National University of Singapore: tcarlson@comp.nus.edu.sg.

ACM 1550-4840/2024/12-ART
https://doi.org/10.1145/3709139

prevent the use of unauthorized or duplicate copies of integrated circuits (ICs) is the use of custom hardware that restricts system access. This method, known as logic locking, enables functionality only when a valid private key is provided. Logic locking mechanisms are implemented by inserting key-controlled modules into the netlist. Only when the correct key is interested at runtime will the original circuit behavior be preserved, while incorrect keys will lead to erroneous outputs [54].

Logic locking can be divided into two closely related concepts: logic encryption and obfuscation. Logic encryption, known as functional locking, is a technique that requires the correct key for the circuit to function properly and makes it difficult for unauthorized users to determine the key by analyzing the design alone. Obfuscation, or structural locking, involves both hiding the design's structure to prevent its extraction through structural analysis and ensuring that only the correct key can unlock the design's functionality. As such, logic locking is used primarily to protect intellectual property against reverse engineering, IP piracy and theft, overproduction, and unauthorized activation.

While logic locking has been previously demonstrated to be an effective solution to protect against these concerns [54], there is a constant struggle between the logic locking methods to prevent disclosure of the hardware, and the attacks that are used to reverse engineer the hardware even when the adversary does not have the private key. Modern attacks have been shown to successfully determine the correct encryption keys, even when the function of the hardware is unknown. Among these, the SAT attack [44] and its variations [10, 60, 65] are particularly effective at extracting the correct key from locked designs. The ongoing introduction of advanced countermeasures is often followed by the development of newer and more sophisticated attacks against them, illustrating the nature of this cat-and-mouse game since the introduction of logic locking. Hence, an investigation of known attacks (introduced in Section 2) and the limitations of existing locking methods reveals the need for a user-defined, constraint-aware locking technique that offers a balanced approach to security and reliability against these attacks.

To the best of our knowledge, this work presents the first defense to simultaneously counter a wide range of attacks, including SAT, machine learning-based (ML), and constant-propagation attacks. Our proposed method, Secure and Reliable Logic Locking (SRLL), is a user-driven, constraint-aware solution to improve the security and performance trade-off and enhance the ability to withstand state-of-the-art attacks. In this regard, SRLL is composed of five steps that analyze and integrate countermeasures into the target circuit. First, SRLL extracts a sub-circuit of the netlist by identifying specific gate-level parameters that can cause important effects on its functionality upon applying incorrect keys (a modified mechanism form [20, 48, 51]). Then, it applies a locking mechanism to increase the complexity and duration of each iteration of SAT-based attacks (a modified mechanism form [3, 4, 51, 79, 83]). Subsequently, it employs an obfuscation technique aiming to further increase the duration of each iteration of SAT and hide inserted key gates to overcome structural and ML attacks (a modified mechanism form [18, 51]). To concatenate the locked sub-circuit in the main design, SRLL uses methods [34] to conceal both the functionality and implementation details of the locked block. As the final step, an alternative block circuit [85] is employed, which plays a significant role in exponentially increasing the number of iterations required to mount SAT-based attacks. This block is designed to be resilient against detection by constant propagation [3] and removal attacks [77]. More specifically, SRLL makes the following contributions:

(1) To the best of our knowledge, it is the first scheme that can mitigate a wide range of ML (SAIL, OMLA), functional (SAT), constant-propagation (SCOPE), and approximate (AppSAT) attacks. The goal of this work is to provide a robust method that can prevent attacks from a wide range of methods;
(2) This work addresses a critical gap in the field by proposing a locking mechanism that is robust against both exact and approximate SAT attacks. This dual resilience represents a significant advancement over existing methods, which often fail to defend effectively against exact and approximate attacks;
(3) It introduces a user-defined constraint-aware solution. This process can be controlled by the user using two different specifications: security and reliability. The primary focus of security is to protect systems

provides configurable resistance against AppSAT with no negative effect on resiliency against SAT and learning-based attacks while having low overhead on large designs. The results that are evaluated on ISCAS'85, MCNC'91, and ITC'99 benchmarks, demonstrate that all targeted attacks were unsuccessful in decrypting SRLL-locked designs (apart from 6 small circuits) while reporting 2.8% performance overhead, 21.9% power overhead, and 29.9% area overhead on average in case of defining performance as the main constraint. Finally, our experimental evaluation demonstrates extremely low overheads in performance, power, and area for large netlists (ITC'99).

ACKNOWLEDGMENTS

The authors are grateful to Prabuddha Chakraborty for providing access to the SAIL framework [18].
This research was in part supported by grants from the Institute for Research in Fundamental Sciences (IPM) (CS1402-4-222) and the National Research Foundation (NRF) of Singapore (NRF2018NCR-NCR002).

REFERENCES

[1] Abdulrahman Alaql and Swarup Bhunia. 2021. SARO: Scalable Attack-Resistant Logic Locking. TIFS 16 (2021), 3724–3739.
[2] Abdulrahman Alaql, Md Moshiur Rahman, and Swarup Bhunia. 2020. SCOPE Tool. https://github.com/alaql89/SCOPE
[3] Abdulrahman Alaql, Md Moshiur Rahman, and Swarup Bhunia. 2021. SCOPE: Synthesis-Based Constant Propagation Attack on Logic Locking. TVLSI 29, 8 (2021), 1529–1542.
[4] Lilas Alrahis, Satwik Patnaik, Muhammad Abdullah Hanif, Muhammad Shafique, and Ozgur Sinanoglu. 2021. UNTANGLE: Unlocking Routing and Logic Obfuscation Using Graph Neural Networks-based Link Prediction. In ICCAD. Munich, Germany, 1–9.
[5] Lilas Alrahis, Satwik Patnaik, Faiq Khalid, Muhammad Abdullah Hanif, Hani Saleh, Muhammad Shafique, and Ozgur Sinanoglu. 2021. GNNUnlock: Graph Neural Networks-Based Oracle-Less Unlocking Scheme For Provably Secure Logic Locking. In DATE. Grenoble, France, 780–785.
[6] Lilas Alrahis, Satwik Patnaik, Johann Knechtel, Hani Saleh, Baker Mohammad, Mahmoud Al-Qutayri, and Ozgur Sinanoglu. 2021. UNSAIL: Thwarting Oracle-Less Machine Learning Attacks on Logic Locking. TIFS 16 (2021), 2508–2523.
[7] Lilas Alrahis, Satwik Patnaik, Muhammad Shafique, and Ozgur Sinanoglu. 2021. OMLA: An Oracle-Less Machine Learning-Based Attack on Logic Locking. TCAS-II 69, 3 (2021), 1602–1606.
[8] Lilas Alrahis, Satwik Patnaik, Muhammad Shafique, and Ozgur Sinanoglu. 2021. OMLA Tool. https://github.com/DfX-NYUAD/OMLA
[9] Lilas Alrahis, Muhammad Yasin, Hani Saleh, Baker Mohammad, and Mahmoud Al-Qutayri. 2019. Functional Reverse Engineering on SAT-Attack Resilient Logic Locking. In ISCAS. Sapporo, Japan, 1–5.
[10] Kimia Zamiri Azar, Hadi Mardani Kamali, Houman Homayoun, and Avesta Sasan. 2018. SMT Attack: Next Generation Attack on Obfuscated Circuits with Capabilities and Performance Beyond the SAT Attacks. TCHES 2019, 1 (2018), 97–122. https://doi.org/10.13154/tches.v2019.i1.97-122
[11] Kimia Zamiri Azar, Hadi Mardani Kamali, Houman Homayoun, and Avesta Sasan. 2021. From Cryptography to Logic Locking: A Survey on the Architecture Evolution of Secure Scan Chains. IEEE Access 9 (2021), 73133–73151.
[12] Alex Baumgarten, Akhilesh Tyagi, and Joseph Zambreno. 2010. Preventing IC Piracy Using Reconfigurable Logic Barriers. D&T 27, 1 (2010), 66–75.
[13] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. 2008. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In PACT (Toronto, Ontario, Canada). 72–81. https://doi.org/10.1145/1454115.1454128
[14] Robert Brayton and Alan Mishchenko. 2010. ABC: An Academic Industrial-Strength Verification Tool. In CAV. 24–40.
[15] Franc Brglez. 1985. A Neural Netlist of 10 Combinational Benchmark Circuits. ISCAS (1985), 151–158.
[16] Franc Brglez, David Bryan, and Krzysztof Kozminski. 1989. Combinational Profiles of Sequential Benchmark Circuits. In ISCAS. 1929–1934.
[17] Abhishek Chakraborty, Yuntao Liu, and Ankur Srivastava. 2018. TimingSAT: Timing Profile Embedded SAT Attack. In ICCAD. 1–6.
[18] Prabuddha Chakraborty, Jonathan Cruz, Abdulrahman Alaql, and Swarup Bhunia. 2021. SAIL: Analyzing Structural Artifacts of Logic Locking using Machine Learning. TIFS 16 (2021), 3828–3842.
[19] Prabuddha Chakraborty, Jonathan Cruz, and Swarup Bhunia. 2019. SURF: Joint Structural Functional Attack on Logic Locking. In HOST. McLean, VA, USA, 181–190.
[20] Rajat Subhra Chakraborty and Swarup Bhunia. 2009. HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection. TCAD 28, 10 (2009), 1493–1502.
[21] Jianqi Chen, Monir Zaman, Yiorgos Makris, RD Shawn Blanton, Subhasish Mitra, and Benjamin Carrion Schafer. 2020. Decoy: Deflection-Driven HLS-Based Computation Partitioning for Obfuscating Intellectual Property. In DAC. 1–6.

[22] Hsiao-Yu Chiang, Yung-Chih Chen, De-Xuan Ji, Xiang-Min Yang, Chia-Chun Lin, and Chun-Yao Wang. 2019. LOOPLock: Logic Optimization-Based Cyclic Logic Locking. TCAD 39, 10 (2019), 2178–2191.
[23] Animesh Basak Chowdhury, Lilas Alrahis, Luca Collini, Johann Knechtel, Ramesh Karri, Siddharth Garg, Ozgur Sinanoglu, and Benjamin Tan. 2023. ALMOST: Adversarial Learning to Mitigate Oracle-less ML Attacks via Synthesis Tuning. arXiv preprint arXiv:2303.03372 (2023), arXiv–2303.
[24] Scott Davidson. 1999. Notes on ITC'99 Benchmarks. http://cerc.utexas.edu/itc99-benchmarks/bendoc1.html (1999).
[25] Sophie Dupuis, Papa-Sidi Ba, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. 2014. A Novel Hardware Logic Encryption Technique for Thwarting Illegal Overproduction and Hardware Trojans. In IOLTS. 49–54.
[26] Task Force. 2005. High Performance Microchip Supply. DTIC (2005).
[27] Mona Hashemi, Siamak Mohammadi, and Trevor E. Carlson. 2024. LOTUS: A Scalable Framework to Lock Multi-Module Designs with One-Time Key and Self-Destructive Approaches. IEEE ESL (2024), 1–1.
[28] Mona Hashemi, Amirabbas Momeni, A Pashrashid, and Siamak Mohammadi. 2022. Graph Centrality Algorithms for Hardware Trojan Detection at Gate-Level Netlists. IJE 35, 7 (2022), 1375–1387.
[29] Rakibul Hassan, Gaurav Kolhe, Setareh Rafatirad, Houman Homayoun, and Sai Manoj Pudukotai Dinakarrao. 2022. A Neural Network-Based Cognitive Obfuscation Toward Enhanced Logic Locking. TCAD 41, 11 (2022), 4587–4599. https://doi.org/10.1109/TCAD.2021.3138686
[30] Hadi Mardani Kamali, Kimia Zamiri Azar, Farimah Farahmandi, and Mark Tehranipoor. 2022. Advances in logic locking: Past, present, and prospects. Cryptology ePrint Archive (2022).
[31] Hadi Mardani Kamali, Kimia Zamiri Azar, Kris Gaj, Houman Homayoun, and Avesta Sasan. 2018. LUT-Lock: A Novel LUT-Based Logic Obfuscation For FPGA-Bitstream And ASIC-Hardware Protection. In ISVLSI. 405–410.
[32] Hadi Mardani Kamali, Kimia Zamiri Azar, Houman Homayoun, and Avesta Sasan. 2019. Full-Lock: Hard Distributions of SAT Instances for Obfuscating Circuits Using Fully Configurable Logic and Routing Blocks. In DAC. Article 89, 6 pages. https://doi.org/10.1145/3316781.3317831
[33] Hadi Mardani Kamali, Kimia Zamiri Azar, Houman Homayoun, and Avesta Sasan. 2020. InterLock: An Intercorrelated Logic And Routing Locking. In ICCAD. 1–9.
[34] Soroush Khaleghi, Kai Da Zhao, and Wenjing Rao. 2015. IC Piracy Prevention via Design Withholding and Entanglement. In ASP-DAC. 821–826.
[35] Leon Li and Alex Orailoglu. 2019. Piercing Logic Locking Keys Through Redundancy Identification. In DATE. 540–545.
[36] Leon Li and Alex Orailoglu. 2019. Shielding Logic Locking From Redundancy Attacks. In VTS. 1–6.
[37] Nimisha Limaye, Emmanouil Kalligeros, Nikolaos Karousos, Irene G Karybali, and Ozgur Sinanoglu. 2020. Thwarting All Logic Locking Attacks: Dishonest Oracle with Truly Random Logic Locking. TCADS 40, 9 (2020), 1740–1753.
[38] Nimisha Limaye, Satwik Patnaik, and Ozgur Sinanoglu. 2021. Fa-SAT: Fault-Aided SAT-Based Attack on Compound Logic Locking Techniques. In DATE. 1166–1171.
[39] Yuntao Liu, Michael Zuzak, Yang Xie, Abhishek Chakraborty, and Ankur Srivastava. 2020. Strong Anti-SAT: Secure and Effective Logic Locking. In ISQED. 199–205.
[40] Yuntao Liu, Michael Zuzak, Yang Xie, Abhishek Chakraborty, and Ankur Srivastava. 2021. Robust and Attack Resilient Logic Locking With a High Application-Level Impact. JETC 17, 3 (2021), 1–22.
[41] Tyler McDonnell, Sari Andoni, Elmira Bonab, Sheila Cheng, Jun-Hwan Choi, Jimmie Goode, Keith Moore, Gavin Sellers, and Jacob Schrum. 2018. Divide and Conquer: Neuroevolution for Multiclass Classification. In GECCO. 474–481.
[42] Assistant Secretary of Defense for Research and Engineering. 2019. Common Evaluation Platform. https://github.com/mit-ll/CEP
[43] Hammond Pearce, Ramesh Karri, and Benjamin Tan. 2023. High-Level Approaches to Hardware Security: A Tutorial. TECS (2023).
[44] Subramanyan Pramod, Sayak Ray, and Sharad Malik. 2015. Evaluating the Security of Logic Encryption Algorithms. In HOST. 137–143.
[45] Subramanyan Pramod, Sayak Ray, and Sharad Malik. 2015. SAT Attack Tool. https://bitbucket.org/spramod/host15-logic-encryption
[46] M Tanjidur Rahman, M Sazadur Rahman, Huanyu Wang, Shahin Tajik, Waleed Khalil, Farimah Farahmandi, Domenic Forte, Navid Asadizanjani, and Mark Tehranipoor. 2020. Defense-in-Depth: A Recipe for Logic Locking to Prevail. Integration 72 (2020), 39–57.
[47] Jeyavijayan Rajendran, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. 2012. Security Analysis Of Logic Obfuscation. In DAC. 83–89.
[48] Jeyavijayan Rajendran, Huan Zhang, Chi Zhang, Garrett S Rose, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. 2013. Fault Analysis-Based Logic Encryption. TC 64, 2 (2013), 410–424.
[49] Amin Rezaei, You Li, Yuanqi Shen, Shuyu Kong, and Hai Zhou. 2019. CycSAT-Unresolvable Cyclic Logic Encryption Using Unreachable States. In ASP-DAC. 358–363.
[50] Amin Rezaei, Yuanqi Shen, Shuyu Kong, Jie Gu, and Hai Zhou. 2018. Cyclic Locking And Memristor-Based Obfuscation Against CycSAT And Inside Foundry Attacks. In DATE. 85–90.
[51] Amin Rezaei, Yuanqi Shen, and Hai Zhou. 2020. Rescuing Logic Encryption in Post-SAT Era by Locking & Obfuscation. In DATE. 13–18.

[52] Shervin Roshanisefat, Hadi Mardani Kamali, Houman Homayoun, and Avesta Sasan. 2020. SAT-Hard Cyclic Logic Obfuscation For Protecting The IP In The Manufacturing Supply Chain. TVLSI 28, 4 (2020), 954–967.
[53] Shervin Roshanisefat, Hadi Mardani Kamali, and Avesta Sasan. 2018. SRCLock: SAT-Resistant Cyclic Logic Locking For Protecting The Hardware. In GLSVLSI. 153–158.
[54] Jarrod A. Roy, Farinaz Koushanfar, and Igor L. Markov. 2008. EPIC: Ending Piracy of Integrated Circuits. In DATE. 1069–1074.
[55] Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov. 2010. Ending Piracy of Integrated Circuits. Computer 43, 10 (2010), 30–38. https://doi.org/10.1109/MC.2010.284
[56] Abhrajit Sengupta, Nimisha Limaye, and Ozgur Sinanoglu. 2021. Breaking CAS-Lock and Its Variants By Exploiting Structural Traces. Cryptology ePrint Archive (2021).
[57] Abhrajit Sengupta, Mohammed Nabeel, Nimisha Limaye, Mohammed Ashraf, and Ozgur Sinanoglu. 2020. Truly Stripping Functionality for Logic Locking: A Fault-Based Perspective. TCADS 39, 12 (2020), 4439–4452.
[58] Abhrajit Sengupta, Mohammed Nabeel, Muhammad Yasin, and Ozgur Sinanoglu. 2018. ATPG-Based Cost-Effective, Secure Logic Locking. In VTS. 1–6.
[59] Bicky Shakya, Xiaolin Xu, Mark Tehranipoor, and Domenic Forte. 2020. CAS-Lock: A Security-Corruptibility Trade-off Resilient Logic Locking Scheme. TCHES 2020, 1 (2020), 175–202. https://doi.org/10.13154/tches.v2020.i1.175-202
[60] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin. 2017. AppSAT: Approximately Deobfuscating Integrated Circuits. In HOST. 95–100.
[61] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin. 2017. Cyclic Obfuscation For Creating SAT-Unresolvable Circuits. In GLSVLSI. 173–178.
[62] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin. 2021. AppSAT Attack Tool. https://bitbucket.org/kavehshm/neos/src/master/
[63] Kaveh Shamsi, Meng Li, David Z Pan, and Yier Jin. 2018. Cross-Lock: Dense Layout-Level Interconnect Locking Using Cross-Bar Architectures. In GLSVLSI. 147–152.
[64] Yuanqi Shen, You Li, Amin Rezaei, Shuyu Kong, David Dlott, and Hai Zhou. 2019. BeSAT: Behavioral SAT-Based Attack On Cyclic Logic Encryption. In ASP-DAC. 657–662.
[65] Yuanqi Shen and Hai Zhou. 2017. Double DIP: Re-Evaluating Security of Logic Encryption Algorithms. In GLSVLSI. 179–184.
[66] Deepak Sirone and Pramod Subramanyan. 2020. Functional Analysis Attacks on Logic Locking. TIFS 15 (2020), 2514–2527.
[67] Dominik Šišejković, Farhad Merchant, Rainer Leupers, Gerd Ascheid, and Sascha Kegreiss. 2019. Inter-Lock: Logic Encryption for Processor Cores Beyond Module Boundaries. In ETS. 1–6.
[68] Joseph Sweeney, Marijn JH Heule, and Lawrence Pileggi. 2020. Modeling Techniques For Logic Locking. In ICCAD. 1–9.
[69] Inc. Synopsys. 2016. Synopsys Design Compiler. https://www.synopsys.com/
[70] Yang Xie and Ankur Srivastava. 2016. Mitigating SAT attack on Logic Locking. In CHES. 127–146.
[71] Yang Xie and Ankur Srivastava. 2017. Delay Locking: Security Enhancement of Logic Locking Against IC Counterfeiting And Overproduction. In DAC. 1–6.
[72] Yang Xie and Ankur Srivastava. 2018. Anti-SAT: Mitigating SAT Attack on Logic Locking. TCAD 38, 2 (2018), 199–207.
[73] Xiaolin Xu, Bicky Shakya, Mark M Tehranipoor, and Domenic Forte. 2017. Novel Bypass Attack and BDD-Based Tradeoff Analysis Against All Known Logic Locking Attacks. In CHES. 189–210.
[74] Fangfei Yang, Ming Tang, and Ozgur Sinanoglu. 2019. Stripped Functionality Logic Locking With Hamming Distance-Based Restore Unit (SFLL-hd)–Unlocked. TIFS 14, 10 (2019), 2778–2786.
[75] Saeyang Yang. 1991. Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0. Citeseer.
[76] Muhammad Yasin, Bodhisatwa Mazumdar, Jeyavijayan JV Rajendran, and Ozgur Sinanoglu. 2016. SARLock: SAT Attack Resistant Logic Locking. In HOST. 236–241. https://doi.org/10.1109/HST.2016.7495588
[77] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. 2017. Removal Attacks on Logic Locking and Camouflaging Techniques. TETC 8, 2 (2017), 517–532.
[78] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. 2017. Security Analysis of Anti-SAT. In ASP-DAC. 342–347.
[79] Muhammad Yasin, Abhrajit Sengupta, Mohammed Thari Nabeel, Mohammed Ashraf, Jeyavijayan Rajendran, and Ozgur Sinanoglu. 2017. Provably-Secure Logic Locking: From Theory to Practice. In ACM CCS. 1601–1618.
[80] Muhammad Yasin, Abhrajit Sengupta, Benjamin Carrion Schafer, Yiorgos Makris, Ozgur Sinanoglu, and Jeyavijayan Rajendran. 2017. What to lock? Functional and Parametric Locking. In GLSVLSI. 351–356.
[81] Muhammad Yasin and Ozgur Sinanoglu. 2017. Evolution of Logic Locking. In VLSI-SoC. 1–6.
[82] Michael Yue and Sara Tehranipoor. 2021. A Novel Probability-Based Logic-Locking Technique: ProbLock. Sensors 21, 23 (2021), 8126.
[83] Hai Zhou. 2017. A Humble Theory and Application for Logic Encryption. Cryptology ePrint Archive (2017).
[84] Hai Zhou, Ruifeng Jiang, and Shuyu Kong. 2017. CycSAT: SAT-Based Attack on Cyclic Logic Encryptions. In ICCAD. 49–56.
[85] Jingbo Zhou and Xinmiao Zhang. 2021. Generalized SAT-Attack-Resistant Logic Locking. TIFS 16 (2021), 2581–2592.