Defense in Depth approach on AES Cryptographic Decryption core to Enhance Reliability

Gayatri Yendamury
Robert Bosch Engineering and Business Solutions Private Limited
Coimbatore, India
Gayatri.Yendamury@in.bosch.com

N Mohankumar
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India
n_mohankumar@cb.amrita.edu

2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) | 978-1-6654-4067-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEMTRONICS52119.2021.9422567

***Abstract*— Security is need of the hour in today's world since the cyber physical systems are prone to malicious attacks. Advanced Encryption Standard is a cryptographic algorithm which is utilized extensively but is sensitive to dangerous attacks due to advances in technology. This paper administers Defense in Depth approach at system level on AES cryptographic core using an effective logic locking technique. AES lock block is inserted judiciously in subprocess of AES decryption algorithm. This approach achieves output corruption of 70% when incorrect password is provided at input due to which the probability of guessing the information and reverse engineering the architecture is reduced to greater extent. AES Lock block is highly efficient to secure the AES cryptographic decryption core. This work successfully shields AES cryptographic decryption core using Defense in Depth approach.**

Keywords— Defense in Depth, Hardware Security, Design for security, Logic locking, AES Decryption Core, AES Lock Block.

I. INTRODUCTION

Security is a major concern these days. Amidst the era of fabless companies, for the cost saved in manufacturing a significant effort has to be put in to curb variety of security threats like IP Piracy, Reverse Engineering and Hardware Trojan and a steep increase in the number of untrustworthy electronics which are encountered due to deployment of IC fabrication in the global design flow. Hardware security is an essential tool being worked upon recently to restrain the huge number of losses caused to the semiconductor industry due to the above-mentioned problems to mitigate malicious threats from attacker's end, Defense in Depth approach stands tall as it implements mechanisms in layered fashion. Design for Trust techniques (DfTr) [1] are also utilized which are broadly classified into active and passive. IC metering, Fingerprinting, Watermarking, Logic locking, IC camouflaging are examples of DfTr techniques. Out of them, logic locking is one the various active techniques which has been garnering more attention from the research world recently because of versatility, lesser limitations than others and its ability to protect against an attack in anywhere in the supply chain.

Logic locking method involves insertion of additional hardware and numerous extra key-inputs, in addition to the original inputs, into the design which are driven by an on-chip tamper proof memory. The added locking hardware makes sure that the design details cannot be recovered with the help of reverse engineering and ensures that the output is incorrect if the correct key is not provided at the input. There are various sequential and combinational logic locking techniques. The former ones' result in incorrect output since it is corrupted due to the presence of incorrect keys. Various combinational logic locking techniques implement XOR/XNOR key gates, AND/OR gates, multiplexers, or combinations of these gates [2]. Kundi et al. AES encryption core is implemented on Spartan-3 FPGA. The design includes MUX in the intermediate states to fed back data into encryption core [3]. The implementation provided fast encryption core. Santoosh et al. elaborates that AES encryption and decryption algorithm was carried out by implementing more than one round parallelly [4]. It increased throughput and offered high security.

In this work, Defense in Depth approach is administered to prevent malicious attacks like key sensitization, brute force attack and reverse engineering and secure critical data using lock locking technique on AES decryption core. Section II provides brief on the state of art technology. Section III elaborates on the AES decryption core, AES lock block core and Performance metrics. Section IV presents the results justifying all our claims. Section V presents the conclusion of this work.

II. OVERVIEW

Logic locking techniques are utilized to protect against malicious threats. Stripped functional behaviour logic locking technique is proposed by Yasin et al. which mentions that certain amount of functional behaviour is latent in form of secret key [5]. A stripped-functionality logic locking (SFLL) technique which strips the functionality of the design and hides it in the form of a secret key. The stripped functional behaviour is recovered only through on-chip restore process. There are a number

978-1-6654-4067-7/21/$31.00 ©2021 IEEE

of ways to attack the design. SAT attack which is a recent and a fatal one which has the ability to decipher the right key of almost all logic locking techniques. Xie et al. explains one of the effective methods to counteract this is proposed in [6]. Delay locking determines functionality and time profiles. A key with incorrect timing will end up in violation of time and malfunction of the circuit. A delay key gate which is tunable is also proposed which can overshadow both functionality and timing profile of IC design. Sweeney et al. proposes Latch-based logic locking technique [7]. This method alters data flow and logic in the circuit by inserting latches. Karmakar et al. describes Logic Encryption strategy [8]. This strategy inserts key gate in such a way that quality of security improves and enhances key-interdependence due to incorporation of key-dependency module. Torrance et al. discusses Reverse engineering techniques [9]. These techniques extract design information which is confidential. Yasin et al. introduces Strong logic locking method that has been implemented in such a way that the key gates are inserted where it is non mutable. Drawback of this method is that it is laborious to find interfering key locations and does not guarantee output corruption. The study also states that existing SAT-based attacks can be averted possibly using one-way random functions. The technique of weighted logic locking [11], [12], [13] was proposed recently in which multiple key inputs are given to every control gate, usually at locations of highest fault impact, instead of the conventional technique where a single key locks the entire circuit. To gain security, Rekha et al. has implemented logic locking concept to protect the clock line of I2C protocol by securing data [14]. When loaded onto an on-chip memory, the secret keys restore the original functionality of the design and creates mismatch between the reverse- engineered netlist and original design. Baby et al. proposes a technique known as LUT based dynamic obfuscation. It ensures that the functionality is latent from untrustworthy stages of design flow [15]. Benchmark circuits were analysed related to number of cycles and hamming distance. Power consumption and area overhead is also decreased.

III. PROPOSED METHODOLOGY

The motive behind this work is to secure the critical data and prevent malicious attacks by administering the defense in depth approach at system level. This approach surges redundancies which reduces momentum of attacker resulting in enhanced reliability. This work is divided into two phases where first phase elaborates the defense in depth approach and second phase evaluates its performance using specific parameters.

A. Defense of Depth

In this work, Defense in Depth approach is applied in two stages. In the first stage, logic locking technique is used to authenticate and grant access control to implement AES algorithm and in the second stage, security is propounded by performing AES cryptographic decryption algorithm. The attacker has to obtain access in order to access the crypto primitive algorithm (AES) before key guessing resulting in high protection of the data.At gate level, in each round, the control signals from logic lock block are connected to different logic cells. This increases random behaviour and disables the attacker from differentiating the correct and incorrect output control signals. This approach adds multiple layers of security and strengths it against malicious attacks. Figure 1 depicts the Defense in Depth approach adopted in this project.

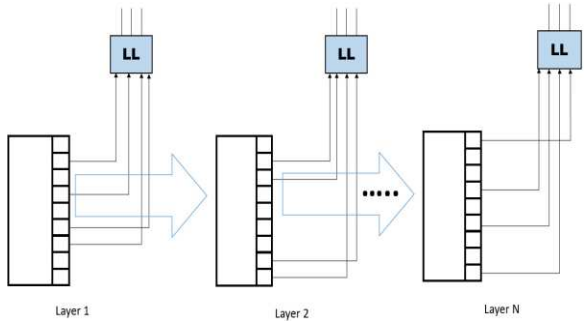

Fig. 1. Defense in Depth Approach

AES Decryption. AES algorithm is widely used symmetric cryptographic algorithm as it is more secure and has low cost for implementation. AES decryption algorithm is iterative in nature. The transformation occurs in every round of the decryption algorithm. Each round consists four subprocess. The last round is an exception as it excludes the Inverse Mix Column subprocess. AES Decryption algorithm has 4 main components. Plain text, Secret key, Cipher text and Decryption algorithm. AES Decryption algorithm consists of four subprocesses. Inverse Shift rows, Inverse Substitution Bytes, Add round key, Inverse Mix column. AES core is vulnerable to attacks. Absence of strong secure system has led to such malicious attacks and insertion of trojans. Numerous techniques are implemented to detect the presence of Trojans. But the call is for a safe and smart technique to safeguard the IC by adding extra layers of security using logic locking technique.

There are three types of AES decryption algorithms depending on key size. AES supports key size of 128, 192 and 256 bits. The number of rounds increases with respect to key size. In AES algorithm, the

input text data size always remains 128 bits. The organization of AES decryption algorithms are illustrated in Table I.

TABLE I. ORGANIZATION OF AES

Type	Key Size	Data Size	Rounds
AES 128	128	128	10
AES 192	192	128	12
AES 256	256	128	14

Logic Locking. Logic Locking is an emerging technology in the sphere of Design for security. Purpose behind using logic locking technique in this work is to protect against vulnerable attacks. In this technique, the attacker is unaware of the presence of logic lock block and hence obtains corrupted text while performing attacks. This technique reduces the probability of the model being prone to attacks. Lock module is inserted to ensure that the correct output is obtained only in presence of correct password. The output is corrupted when incorrect password is provided at input.

Logic lock block is inserted in each subprocess of AES Decryption algorithm. It is inserted judiciously at system level in such a manner where the probability of differentiating it from the original circuit is minimal. The 16-bit lock key is flashed in tamper proof memory (D flash). The lock block consists of non-linear weighted hexadecimal component and a comparator. Comparator compares the weighted hexadecimal with the input password.

The naming convention of the block is in systematic form. In the naming convention, key size is mentioned first followed by B which indicates Block and then followed by numerical value which in turn indicates the subprocess as illustrated in the Table II. For example, 192-B3 indicates that the logic lock block is inserted in the AddRound Key subprocess of AES-192 algorithm. The schematic of AES algorithm with lock block using Defense in Depth is depicted in Figure 2.

TABLE II. NAMING CONVENTION

Value after B	Corresponding Sub-Process
1	Inverse Shift Row
2	Inverse Substitution Byte
3	AddRound Key
4	Inverse Mix Column

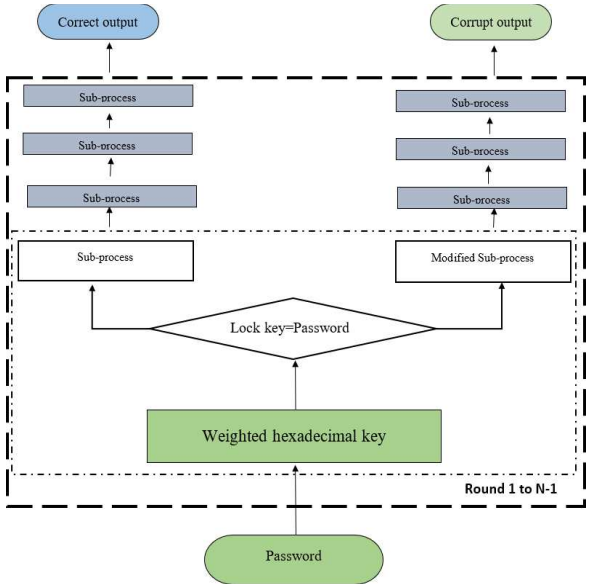

Fig. 2. AES lock block in Defense in Depth approach

1) *Insertion of Lock Block in Inverse Shift row:* The lock block accepts the password and maps the bits to weighted hexadecimal bits and compares it with the lock key in tamper proof memory. The block unlocks and executes the remaining subprocess without alteration if the password provided at input is correct. If the password provided is incorrect, the inverse shift rows sub process is altered by varying the offsets to shift the rows cyclically.

2) *Insertion of Lock Block in Inverse Substitution Byte:* The lock block accepts the password and maps the bits to weighted hexadecimal bits and compares it with the lock key in tamper proof memory. The block unlocks and executes the remaining subprocess without alteration if the password provided at input is correct. If the password provided is incorrect, the inverse substitution sub process is altered by replacing the mapping elements in the substitution box.

3) *Insertion of Lock Block in Add Round Key:* The lock block accepts the password and maps the bits to weighted hexadecimal bits and compares it with the lock key in tamper proof memory. The block unlocks and executes the remaining subprocess without alteration if the password provided at input is correct. If the password provided is incorrect, the AddRound key subprocess is altered by including additional XOR function to perform XOR operation with a random constant.

4) *Insertion of Lock Block in Inverse Mix Column*: The lock block accepts the password at input and maps the bits to weighted hexadecimal bits and compares it with the lock key in tamper proof memory. The block unlocks and executes the remaining sub process without alteration if the lock key provided at input is correct. If the password provided is incorrect, the inverse mix column sub process is altered by performing XOR operation on the resultant value of multiplicative inverse.

B. Performance Metrics

Performance metric such as distance metrics, power, time and Information gain are evaluated to estimate the performance of the Defense in Depth approach by examining Lock block. Distance metrics such as Hamming distance, Levenshtein distance and Jaro distance are utilized to estimate the performance of the algorithm and evaluate the extent to which the output is corrupted when incorrect password is provided at the input.

Hamming Distance. Hamming distance is the difference between the number of bits in output state matrix and input state matrix. Substitution of bits are carried out for evaluating the hamming distance.

Levenshtein Distance. Levenshtein distance is the difference between the number of bits in the input state matrix and output state matrix. Insertion, deletion and substitution operations are carried out to transform input state matrix to output state matrix for evaluating Levenshtein distance.

Jaro Distance. Jaro distance is difference between the number of bits in input state matrix and output state matrix. Transposition is carried out to transform from input state matrix to output state matrix for evaluating Jaro distance.

Power and Time. Power and time consumed is measured by performing side channel power analysis. Side channel power analysis is performed on Chip Whisperer Lite board using the measurement setup.

Information Gain. Information gain is a measure of the information obtained. Entropy is lack of order. Information gain is the difference of entropy before transformation and after transformation. The formula for Information gain is given by formula, Information Gain = [entropy(parent)] – [average entropy(children)]. Entropy is computed using the formula, Entropy = $- \sum P_i \log_2 P_i$.

IV. RESULTS AND ANALYSIS

This work is implemented on AES symmetric cryptographic decryption core with key size of 128,192 and 256 bits. Table III depicts the correct output and corrupted output values of AES crypto primitive for varying inputs when Lock block is inserted.

TABLE III. SAMPLE CASES

Case I	Correct Password	0x0C, 0x0B
	Incorrect Password	0x0D,0x0E
	Weighted hexadecimal	0x0A,0x0F
	Key	0xb267516182ea2d6aba7f518890fcf4c3
	Input	0x935248d120cd90bfcd11587991a6b023
	Output	0xdc4e3f68ddb1c6e1ddc8236e5f562aa
	Corrupted output_128AR	0xadc5c7640060ef51968bee4664ecc8b0
	Corrupted output_128MC	0x6dbe5aa938416feeb1b78e51d2784e68
	Corrupted output_128SB	0x23c7c820a97bc328756ff28a07745ac4
	Corrupted output_128SR	0xc01e90698bf037c85536405784b07fa
Case II	Correct Password	0x0A, 0x0B
	Incorrect Password	0x0C,0x0D
	Weighted hexadecimal	0x0D,0x0E
	Key	0x552061726520746865206265737420696e20776f26c6420a
	Input	0xd8f3a72fc3cdf74dfaf6c3e6b97b2fa6
	Output	0x55a33826068337bbf99165c6530d134c
	Corrupted output_192AR	0xc66d9bd03393c575350902a77fa995a7
	Corrupted output_192MC	0xe934ba0a54cd0cd0be58be710bcc1184
	Corrupted output_192SB	0x84f6ca442db2d935378695ded3b20888
	Corrupted output_192SR	0x5f15645c80789c83467c9e4d55c9f425
Case III	Correct Password	0x0C, 0x0D
	Incorrect Password	0x0A,0x0B
	Weighted hexadecimal	0x09,0x0A
	Key	0x6c6f6769636c6f636b696e6769736173656372657 4636f6e63657074206f75747
	Input	0x26f39bbca19c0fb7c72e7e3063927313
	Output	0x29f7741db1f4a824f2d6ab7d18051543
	Corrupted output_256AR	0x13e758ed901536abf0673ecf1843230e

	Corrupted output_256MC Corrupted output_256SB Corrupted output_256SR	0x229a61ac9c057f356dac209279259e1a 0x077550bfae6a7741d960b22381375f03 0xf0f74252fc4d997d8ce3fcf1e34eb3a0

In figures 3,4 and 5, x-axis indicates the key size and subprocess where the lock block is judiciously inserted and the y-axis indicates the dissimilarity measure of input text and output text. The uncorrupted output is obtained by unlocking the lock block if correct password is provided at input whereas corrupted output is obtained when the lock block remains locked if incorrect password is provided at input.

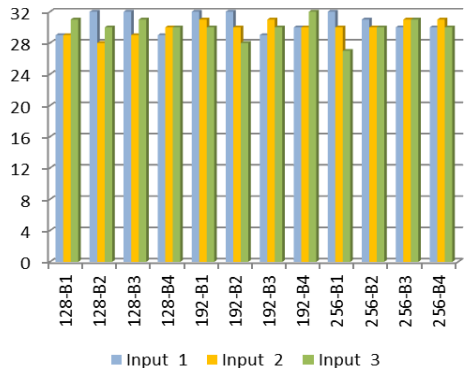

Fig. 3. Analysis of Hamming distance

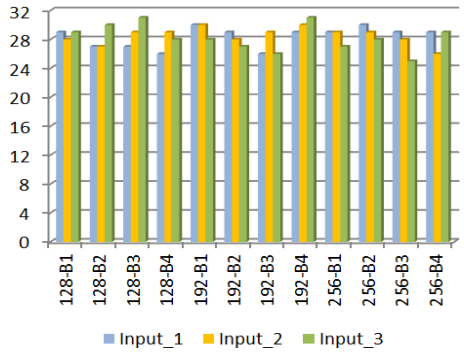

Fig. 4. Analysis of Levenshtein distance

Figure 3 depicts the analysis of Hamming distance in AES 128,192 and 256 when AES Lock block is judiciously inserted in the each of the four subprocess of AES Decryption algorithm for three different set of input cipher text and input secret key. From the figure, it can be inferred that the output text is entirely corrupted with Hamming distance greater than 80%. Figure 4 depicts the analysis of Levenshtein distance in AES 128,192 and 256 when AES Lock block is judiciously inserted in each of the four sub process of AES Decryption algorithm for three different set of input cipher text and input secret key. From the above figure, it can be inferred that the output text is entirely corrupted with Levenshtein distance greater than 80%.

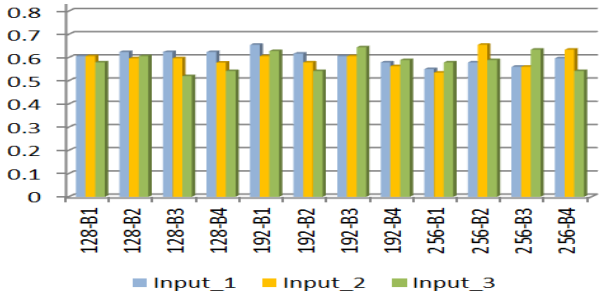

Fig. 5. Analysis of Jaro distance

Figure 5 depicts the analysis of Jaro distance in AES 128,192 and 256 when AES Lock block is judiciously inserted in each of the four subprocess of AES Decryption algorithm for three different set of input cipher text and input secret key. From the Figure, it can be inferred that the output text is entirely corrupted with Jaro distance greater than 50%.

It can be inferred that the hamming distance, Levenshtein distance and Jaro distance is above 50%. Logic locking technique is resilient to brute force attack as this technique is over shadowing the vulnerabilities due to added security control access layer. Further this technique is sturdy enough to resist key sensitization attack since the conversion of input password to non-linear weighted hexadecimal tightens the security by making it difficult for the attacker to identify correlation. between input password and weighted hexadecimal. Also, Reverse engineering is inhibited by mystifying the attacker through layered structure of defense in depth approach.

Further, power consumption and execution time are measured using ChipWhisperer Lite board. The power consumed and execution time while running conventional AES Decryption core and AES Decryption

core with AES lock block is depicted in Figure 6 and Figure 7 respectively.

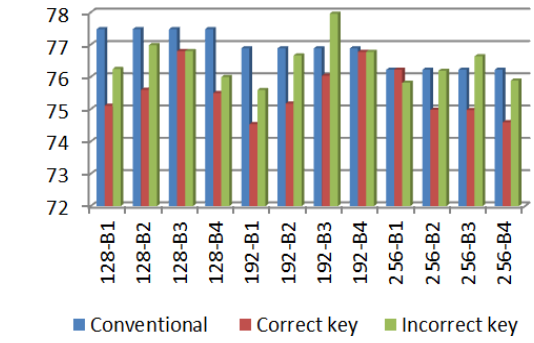

Fig. 6. Analysis of Power Measurements

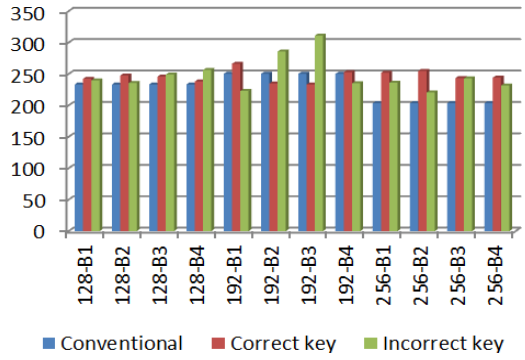

Fig. 7. Analysis of Execution time

Figure 6 depicts the analysis of Power when AES lock block is judiciously inserted in each of the four subprocess of AES 128, 192 and 256 when correct password and incorrect password are provided at input. The x-axis indicates the key length and subprocess where it is inserted and the y-axis indicates the power consumption. From the plot, it can be clearly inferred that the difference between power consumed by conventional AES core and AES core with AES lock block is less than 4 % (negligible differences of µW). Figure 7 depicts the analysis of execution time when AES lock block is judiciously inserted in each of the four subprocess of AES 128,192 and 256 when correct password and incorrect password are provided at input. The x-axis indicates the key length and subprocess where it is inserted and the y-axis indicates the execution time of the circuit.

The difference between execution time of conventional AES core is less than AES core with AES lock block core by 6% for key sizes 128 and 192, and 20% for key size of 256 (negligible differences in ms). Hence, AES core with AES lock block does not drastically increase the power consumption and execution time. So, it is a difficult task for attackers to obtain the password using side channel attacks.

Information gain has been evaluated considering the power consumption attribute. Information gain for AES 128,192 and 256 has been evaluated which has depicted in Table III. It can be inferred from the table that the information gain obtained is less. It is extremely difficult to rely on the information obtained from power consumed to differentiate the correct output and incorrect output when Lock block is inserted.

TABLE IV. INFORMATION GAIN

	Entropy	**Average Entropy**	**Information Gain**
AES 128	0.992774	0.853473	0.139301
AES 192	0.896038	0.775885	0.120153
AES 256	0.746234	0.651909	0.094326

V. CONCLUSION

In this work, the AES lock block has been successfully designed and judiciously inserted into AES cryptographic decryption core in each of the four subprocess of AES decryption algorithm. Power consumed and execution time was measured in XMEGA microcontroller with difference of less than 6% and 20% respectively. Information gain evaluated is also minimal to perform attacks. This technique achieved 80% Hamming distance, 80% Levenshtein distance and 50% Jaro distance. This technique is resilient to Key sensitization attack, reverse engineering and brute force attack. Hence, it is validated that the proposed method is an efficient defense in depth approach which uses logic locking technique. This approach is an effective security measure to avoid unauthorized access on AES Decryption core.

REFERENCES

[1] Yasin, Muhammad & Mazumdar, Bodhisatwa & Rajendran, Jeyavijayan & Sinanoglu, Ozgur. (2019). Hardware Security and Trust: Logic Locking as a Design-for-Trust Solution: Design and Implementation. 10.1007/978-3-319-93100-5_20.

[2] M. Yasin and O. Sinanoglu, "Evolution of logic locking," 2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Abu Dhabi, 2017, pp. 1-6, doi: 10.1109/VLSI- SoC.2017.8203496.

[3] Kundi, Dur-e-Shahwar & Zaka, Saleha & Qurat-Ul-Ain, & Aziz, Arshad. (2009). A compact AES encryption core on Xilinx FPGA. 1 - 4. 10.1109/IC4.2009.4909251.

[4] S. K. R, S. R, M. A. M, P. K. M. S and R. M, "Design of High Speed AES System for Efficient Data Encryption and Decryption System using FPGA," 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization

Techniques (ICEECCOT), Msyuru, India, 2018, pp. 1279-1282, doi: 10.1109/ICEECCOT43722.2018.9001535.

[5] Muhammad Yasin, Abhrajit Sengupta, Mohammed Thari Nabeel, Mohammed Ashraf, Jeyavijayan (JV) Rajendran, and Ozgur Sinanoglu. 2017. Provably-Secure Logic Locking: From Theory to Practice. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 1601–1618.

[6] Y. Xie and A. Srivastava, "Delay locking: Security enhancement of logic locking against IC counterfeiting and overproduction," 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2017, pp. 1-6, doi:10.1145/3061639.3062226.

[7] J. Sweeney, V. Mohammed Zackriya, S. Pagliarini and L. Pileggi, "Latch-Based Logic Locking," 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), San Jose, CA, USA, 2020, pp. 132-141, doi:10.1109/HOST45689.2020.9300256.

[8] R. Karmakar, N. Prasad, S. Chattopadhyay, R. Kapur and I. Sengupta, "A New Logic Encryption Strategy Ensuring Key Interdependency," 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), Hyderabad,2017, pp. 429-434, doi: 10.1109/VLSID.2017.29.

[9] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC), New York, NY, 2011, pp. 333-338.

[10] M. Yasin, J. J. Rajendran, O. Sinanoglu and R. Karri, "On Improving the Security of Logic Locking," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 9, pp. 1411-1424, Sept. 2016, doi: 10.1109/TCAD.2015.2511144.

[11] J. Rajendran, Y. Pino, O. Sinanoglu and R. Karri, "Logic encryption: A fault analysis perspective," 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2012, pp. 953-958, doi: 10.1109/DATE.2012.6176634.

[12] N. Karousos, K. Pexaras, I. G. Karybali and E. Kalligeros, "Weighted logic locking: A new approach for IC piracy protection," 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), Thessaloniki, 2017, pp. 221-226, doi: 10.1109/IOLTS.2017.8046226.

[13] S. Krishnan, M. K. N. and N. D. M., "Weighted Logic Locking to Increase Hamming Distance against Key Sensitization Attack," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 29-33, doi: 10.1109/ICECA.2019.8821880.

[14] Rekha, S. & Reshma, B. & Dilipkumar, N. & Crocier, A. & N., Mohankumar. (2020). Logically Locked I2C Protocol for Improved Security.10.1007/978-981-15-2612-1_67.

[15] Baby J., Mohankumar N., Nirmala Devi M. (2020). Reconfigurable LUT-Based Dynamic Obfuscation for Hardware Security. In: Sengodan T., Murugappan M., Misra S. (eds) Advances in Electrical and Computer Technologies. Lecture Notes in Electrical Engineering, vol 672 Springer, Singapore. https://doi.org/10.1007/978-981-15-5558-9_81.