Lightweight Secured Split Test Technique for Integrated Circuits

Dr. Sudeendra Kumar K, Akshay Girish Kaushik, Adithya B Shetty, Ashutosh Rao, Akshay S
Department of Electronics and Communication Engineering
PES University
Bengaluru, India
sudeendrakumark@pesu.pes.edu, agkblr2001@gmail.com, adithyabshetty@gmail.com, ashutosh.rao242@gmail.com,
akshaybhat456@gmail.com

***Abstract*- Counterfeits coming from untrusted foundries and OSAT centres can prove to be a detrimental factor affecting the IC Supply Chain. Secured Split Test (SST) is a technique used to mitigate the counterfeits coming out of the above mentioned areas. Usage of lightweight encryption mechanisms can optimise the existing SSTF technique by reducing the resources consumed along with making the technique more secured.**

***Keywords*- Lightweight, Supply Chain, Counterfeiting, Split-Test, Design-for-Testability, Ciphers, Encryption, Cryptography**

I. Introduction

In today's scenario, the concept of reverse engineering, counterfeiting and overproduction of ICs has taken a toll on the chip design industry. There are lots of fabless semiconductor companies on the rise and this has led to an all time high value of counterfeit goods as indicated by the survey conducted by OECD. The overall value of fake and pirated goods was estimated to be around $1.7 trillion in 2015, and it is anticipated to increase to around $3 trillion in 2022, according to survey data on counterfeiting and international trade. Magnitude of counterfeiting is growing due to globalization of the supply chain. Both the fab-owned and fabless semiconductor companies depend upon service providers to remain competitive in the market.

The major sources of counterfeiting are extraction of ICs from obsolete electronic equipments, overproduction of ICs from untrusted foundries, defective and faulty chips coming out of OSAT centres, etc.

The design owners should ensure to that the IC can provide good controllability and observability to increase the fault coverage during testing without having to leak sensitive information. Scan flip flops are inserted in the design to provide the ability to observe the internal states of the design which increases fault coverage. This may lead to leakage of sensitive information, which is the need to encrypt scan data.

There are several split test techniques proposed to protect against counterfeiting from untrusted foundries and out of spec chips coming out of OSAT centre. However, as these techniques employs the usage of blocks such as RSA, which are known to be computationally intensive, improvements can be made to further optimise the mechanism with the help of lightweight ciphers. The proposed mechanism not only improves the efficiency and power consumption, but does so while ensuring increased security and reliability by preventing scan chain attacks as well.

The paper begins by first discussing the prior work done on this topic, in section II. Section III explains the existing mechanism that is currently the only one of its kind. This is the same mechanism that this paper aims to improve and section IV encompasses the details about the proposed mechanism with all the optimal replacements and improvements. Section V shows the results obtained and analysis conducted on the new method that is being proposed. Finally, section VI concludes the paper.

II. Prior Work

The authors in [1] proposed a Secured Split Test Technique with Functional Testing which supported both structural and functional testing (SSTF) at wafer sorting and final production but suffers from scan chain related attacks. The technique makes use of a Scrambling block to disturb the test responses as shown in Fig.1 . However, this is a very ad-hoc technique and quite bulky as well. This is susceptible to scan chain attacks, which could lead to the leakage of sensitive information. Few modifications to this existing model will help in making it more secure and efficient and this can be achieved with the help of light weight ciphers.

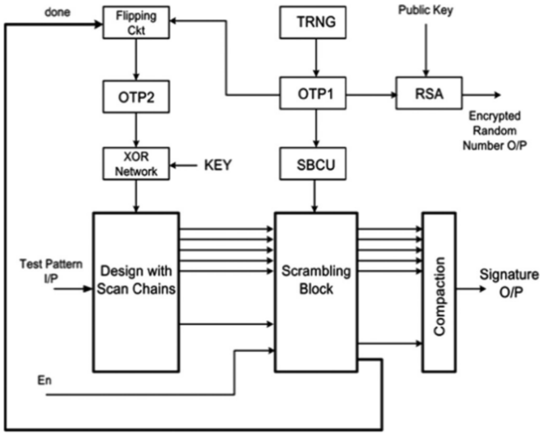

Fig.1

Existing SSTF Mechanism

Secured Split Test was proposed in [2], which supported only structural tests but did not offer the support for functional testing. This prevented counterfeits coming out of untrusted foundries and the faulty chips coming out of OSAT centres. The CSST[3] was a modification of the SST[2] which reduced the communication complexity between the design house and the foundry but still offered no support for functional testing.

In 2018, the authors in [4] came up with another mechanism called as Dynamically Obfuscated wrapper for Split Test (DOST). DOST was capable of both structural and functional tests and differed with SSTF[1] in the way, the Functional key was handled. Functional key was made invalid based on a aging mechanism, which is a flawed technique as the key remains valid if the chip is stored in the warehouse without powering on.

TABLE 1.
Comparison of Various Anti-Counterfeiting Techniques

Technique	Description
EPIC[5]	It prevents unauthorised production of chips, by ensuring only the external key which is held by the owner by IP rights can unlock the chip.
Split Manufacturing[6]	The chip is fabricated in two separate phases, and then connected in trusted foundry. It is expensive and does not prevent faulty chips coming from OSAT centres.
SST[2]	Supports structural tests at both wafer sorting and final testing, but does not support functional testing.
CSST[3]	Reducing communication overhead between foundry and design house, but does not provide functional testing.
SSTF[1]	Supports both structural and functional testing. It uses RSA, scrambling block and PUF based key generator, which adds to the area overhead. Scrambling block is not secure enough as it is vulnerable to Scan Chain attacks.
DOST[4]	It supports both structural and functional testing like SSTF. It only differs in the way key is managed. It makes use of an ageing mechanism which is flawed and built on weak foundations.

The concept of the usage of block ciphers for encrypting scan data was proposed in [14]. This technique presented a solution to the problem of scan-chain related attacks while preserving the efficiency in test and diagnosis provided by DFT techniques. Authors of [14] continued their work in [15] and compared the usage of block and stream ciphers for the purpose of scan chain encryption. It was found out that both have their pros and cons, but the choice can be made based on the requirement. Stream Ciphers were found to be efficient in terms of area and power consumption and is preferred if there already exists a TRNG in the design. Using Stream Ciphers has another advantage as compared to using Block Ciphers because it does not involve off-chip decryption of test responses. Whereas with the use of block ciphers, off-chip decryption is required due to confusion and diffusion properties of the block cipher.

By comparing these techniques, it is safe to say that the SSTF [1] with a few modifications is the most reliable and optimal technique to prevent counterfeit ICs coming out of untrusted foundries and also to prevent defective chips coming out of OSAT centres. Therefore, this paper aims at making these said modifications which would make it more secure and efficient. This is elaborated upon in the upcoming sections.

Authors in [9] compared the asymmetric public key cryptographic algorithms such as RSA, ECC and El-Gamal and provided the comparison in terms of the required key sizes for the necessary Security Bit Levels. It was shown that ECC is very secure when compared to RSA for smaller keysizes. ECC is also lightweight when compared to RSA, which helps in resource constrained devices.

III. Proposed Mechanism

As discussed in the previous section, the existing SSTF[1], although functional, does leave room for improvements. This section details the proposed improvements to the existing mechanism as shown in Fig. 2. These improvements aim to bring reduction in power usage, computation and area. It improves security and efficiency of the system.

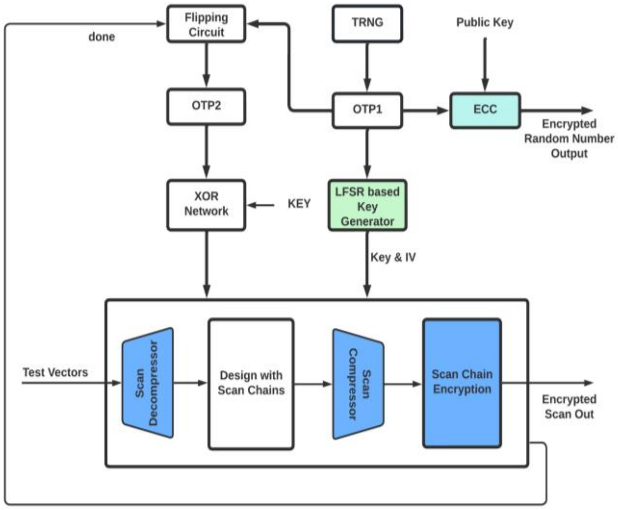

Fig.2. Proposed LW-SSTF

The description of the functionality of the proposed Lightweight SSTF is explained in the paragraphs to follow.

The TRNG produces a 80-bit number upon power on during wafer sorting in the OSAT centre and the value is written onto OTP1. This value is encrypted using Elliptic Curve

Cryptography(ECC) and the encrypted random number is sent to the design house. Initially, OTP1 and OTP2 have all logic 0's and logic 1's. A Functional Key(FKEY) to unlock the design is given to the OSAT centre by the design house. OTP1 is passed onto the LFSR based key generator, which uses this value as a seed to produce a 80-bit key and IV needed by the Trivium cipher to perform scan encryption.

The stream cipher has to be allowed to initialise itself with a known number of clock cycles before sending in the scan data. It is 1162 clock cycles in case of our Trivium implementation, to initialise the key and IV into the registers. Once the cipher is initialised, the scan chain is enabled and the ATPG test vectors can be passed onto the DUT and the scan responses are encrypted and sent back to the design house. Once all the test vectors have been run, a done signal is generated which flips the bits of OTP1 and stores it in OTP2 because of which, the FKEY given to the OSAT centre becomes invalid. This will prevent the unauthorised selling of chips from the OSAT centres as well, as they wont have the new functional key needed to unlock the design.

The design house is the only one who has the knowledge of the LFSR based key generator, and they will know the seed to the LFSR by decrypting the value received after the ECC-encryption. They use this value to obtain the key and IV used by the stream cipher during encryption and decrypt the test responses, and send the list of good dies with the assembly for packaging.

The usage of a encryption mechanism in place of the scrambling block, which was used in the existing SSTF[1] mechanism increases the security by ensuring that scan attacks are not possible and prevents sensitive internal information from being leaked. The use of ECC over RSA not only reduces the area consumed on the chip, but also increases the level of security offered for the same key size.

IV. Implementation

All the implementations discussed under this section are coded using Verilog HDL language.

A. *Elliptic Curve Cryptography*

Elliptic Curve Cryptography is an asymmetric/public key cryptography based on the structure of Elliptic Curves over finite fields. The equation of Elliptic curve is given as, $y^2 = x^3 + ax + b$.

We have designed a 16-bit ECC as shown in Fig. 3. Message Mapping algorithm proposed in [10] is implemented for message encoding. Point addition and Double and Add algorithm proposed in [11] is implemented to obtain the cipher text. The received cipher text will be in form of two coordinates i.e. kG and P_M+kP_B, where G is the generator point on the curve, P_B is the public key, P_M is a mapped point on the curve and k is a random integer.

ECC is used to encrypt the 80-bit value generated by the TRNG which can then be used by the design house to decrypt the test responses to identify dies which are structurally correct to then send it to the assembly. Design specifications of the ECC implemented is discussed in the results section.

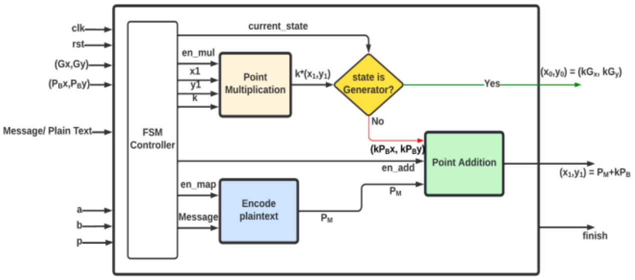

Fig.3. Hardware Implentation of ECC Encryption Module

B. *Scan Chain Encryption*

The Scan Chain Encryption technique showed in Fig.4 is used for the purpose of LW-SSTF.

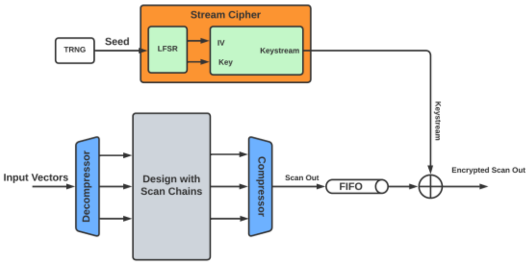

Fig.4. Proposed LW-SSTF

The TRNG produces a seed needed by the LFSR, which then generates the 80-bit key and IV value required by the stream cipher. The cipher produces a keystream which has a period of 2^{64} bits which is XOR'ed with the scan out coming out of the FIFO to get the encrypted scan responses. The design of the custom-made LFSR is only known to the design house. Trivium stream cipher is used for encrypting the test data. The test pattern is allowed to enter the design only after the initialisation phase of the cipher is completed.

C. *Boosted FSM*

A Finite State Machine(FSM) is a mathematical model that works on the concept of various state values that can be assigned to a variable. At any given time, the FSM can be in only one state out of the total number of finite states. The FSM will change states in response to a change in input. There are two types of widely used FSM models. They are the Mealy and Moore model. While the core principle of both of these remains the same, the difference arises in the dependency of the output on the external input and present state. In the Moore model, the output depends only on the value of the current state while in the Mealy model, the output depends on the value of the present state and the value of the input. In this instance, the project makes use of a Mealy FSM as the locking mechanism requires a key that is fed, in the form of input. A boosted FSM has been implemented for functional locking as the concept of an FSM is ideal for implementing a functional locking mechanism. Out of the various states present in the FSM, a predetermined sequence of input bits are needed to unlock the FSM by allowing it to reach the 'unlock' state. The implementation consists of an output signal named unlock that goes high only when the required sequence is entered.

V. Results and analysis

All the designs are synthesized using Cadence Genus tool. Modus DFT tool is used for ATPG generation. 90nm technology library from Synopsys is used for the purpose of this project.

A. *Area Overhead*

The design specifications of the ECC implemented is shown in Table 2. Area consumed is compared with RSA implemented in [12] and it is seen that ECC consumes lesser area when compared to RSA.

TABLE 2.
ECC Design Details

Frequency	Area of proposed ECC (μm^2)	Area of RSA[12] (μm^2)	Power Consumed (mW)
50MHz	859715.4	1213193.28	8.72

Area consumed is reduced by around 29.13% when compared to RSA while also increasing the security offered. The proposed design has a critical path of 19.47ns. Approximate time taken to encrypt 80-bits of data as needed by LW-SSTF is around 260ms at a operating frequency of 50MHz.

The implementation of ECC is a 16-bit version and it can be increased to incorporate keys of larger size to provide better security. However this will result in compromise of area, power and timing.It must also be noted that ECC can be optimised further to provide better performance with the help of multi-bit cells for synthesis, which will further reduce the area and power consumption. Low power techniques can be used to reduce the power consumption as well.

The area of Trivium cipher is 17116.877 μm^2 and that of the LFSR based key generator is found to be around 7919.30μm^2.

TABLE 3.
Scan Chain Overhead for Various Designs

Design	N_{SFFs}*	Area(μm^2)	Trivium Area Overhead (%)
s38584	1141	73364	+23.331
RISC-V	2319	191680.81	+8.929
AES	2987	429444.4	+3.985

Area overhead is calculated for few commonly known designs such as s38584 from ISCAS-89 benchmark, AES and RISC V cores. As we can see, the area overhead is more for smaller designs when compared to larger designs.Table 3 shows the computed results.

It is found that area of block ciphers are generally less when compared to that of stream ciphers, however the time overhead will be more depending on the number of flops used in the design.

B. *Timing Overhead*

For the proposed ECC encryption, time taken to encrypt 80-bits of data as needed by LW-SSTF is found to be around 260ms but as it is a one-time overhead, it is acceptable.

Trivium Stream Cipher is used to perform scan data encryption. The cipher needs 1162 clock cycles to initialise and load the values of key and IV to its internal shift registers. LFSR based key generator is used to generate the Key and IV needed by the cipher.

Timing overhead is compared for block and stream ciphers. Incase of stream ciphers, only timing overhead is with regards to the initialisation of the Trivium stream cipher which takes up 1162 clock cycles to load the key and IV. With block ciphers, the area overhead will be dependent on the number of scan flops in the design. This is explained in the following paraghgraph.

If the block size of a cipher is N, we can consider the number of scan flip flops given by F as F=SN+R, where S is the number of N-bit segments and R is given by F mod N. If the block size is not a multiple of N, then for each pattern, (N-R) bits have to be appended to match the length of test data and the block size. This requires N-R extra clock cycles per pattern.

Table 4 shows the ATPG test time needed to achieve almost 100% coverage for the designs considered.

TABLE 4.
ATPG Test Time

Design	N_{SFFs}*	Fault Coverage	ATPG Test Time (ns)
s38584	1141	99.99%	35440
RISC-V	2319	99.98%	200400
AES	2987	99.99%	74160

For example, if we consider the s38584 circuit, there are 1141 Flipflops. If we choose PRESENT or SKINNY as the block cipher to encrypt the data, their block sizes are 64 i.e. N=64. Now, F=17*64+53 which implies that 11bits of data needs to be padded for each test pattern i.e. 11 extra clock cycles per test pattern. If there are 150 patterns needed to achieve 100% fault coverage, then it will result in an overhead of more than 1500 clock cycles. Whereas if Trivium stream cipher had been used, there wouldve only been a delay to initialise the cipher irrespective of the block size.

C. *Power Reduction*

ECC proposed in this paper is found to consume a power of around 8.72mW. However this can be reduced by making use of techniques such as clock gating.

Stream ciphers consume more power when compared to block ciphers, but this power consumption can be reduced by making use of low power techniques such as clock gating. Clock gating is enabled during the synthesis using Cadence Genus tool and power consumption before and after clock gating are observed as shown in Table 5.

TABLE. 5
Clock Gating on Trivium

Before Clock Gating		After Clock Gating	
Cells Count	Power (uW)	Cells Count	Power(uW)
770	255.34	847	195.79

It can be noticed that the total power consumption is reduced by around 23.32% which is a significant improvement.

D. *Communication Complexity*

The design house sends the GDSII file to the foundry to begin the wafer slicing process. The wafers are fabricated after the foundry creates masks from the GDSII file provided. The IP owner will then provide the OSAT centre with the test patterns to perform the testing. For each die, a True Random Number will be generated and stored in OTP1 which is encrypted using ECC and sent back to the design house. Functional Key is given by the design house to the OSAT centre to perform functional testing. The OSAT centre sends the compressed test responses back to the design house, where the design house creates a list of good dies and their corresponding ECIDs and sends it to the assembly for packaging. After packaging, the design house shared the new FKEY required to unlock the chip only with genuine customers.

E. *Attack Analysis*

An attacker in the foundry or OSAT centre might try to crack the proposed LW-SSTF architecture. Even with the full knowledge of the proposed mechanism, it will be really difficult for the adversary to break the system. The scan encryption mechanism involves the use of key and IV being generated with the help of the same LFSR. However, the seed to the LFSR is being generated from the TRNG.

Trivium keystream is known to have a period of 2^{64} bits which makes it almost impossible to crack as even if a scan chain is run at 1GHz, which is already way too high for scan chains, it would take around 317 years to reach this period.

Secondly, the flipping circuit logic is only known to the design house and hence they cannot decode the value of OTP2 and deduce the FKEY. Obtaining the value of OTP1 will be as good as breaking ECC which is proven to be difficult.

Hence the proposed LW-SSTF technique can be considered secure enough to be implemented on chips to prevent counterfeits out of untrusted foundries and out-of-spec chips coming out of OSAT centres.

VI. Conclusions

Comparison between the existing SSTF and the proposed light weight SSTF can be done by comparing the performance against various factors.

TABLE. 6
Comparison between Existing SSTF and Proposed LW SSTF

Existing SSTF	Proposed Lightweight SSTF
RSA requires larger key size for better security and adds to the overhead.	ECC is lightweight and more secure with smaller key size.
Scrambling block is ad hoc, bulky and sensitive to scan related attacks.	This is replaced by a Scan Chain Encryption Mechanism which is resistant to Scan related attacks.
Uses Scan Compaction.	Uses Scan Compression Logic to reduce test cost.

As mentioned earlier, replacing RSA with ECC makes the proposed technique more secure as ECC's key is tougher to crack when compared to RSA. Earlier SSTF technique was vulnerable to Scan Chain attacks which exposed the test responses to the attackers. This is prevented in the proposed mechanism with the help of Scan Chain Encryption. Area overhead in SSTF is more when compared to Light Weight SSTF.

VII. References

[1] K. Sudeendra Kumar, G. Hanumanta Rao, Sauvagya Sahoo, K.K. Mahapatra, *Secure split test techniques to prevent IC piracy for IoT devices*,Integration, Volume 58, 2017, Pages 390-400, ISSN 0167-9260,https://doi.org/10.1016/j.vlsi.2016.09.004.

[2] G. K. Contreras, M. T. Rahman and M. Tehranipoor, "*Secure Split-Test for preventing IC piracy by untrusted foundry and assembly,*"2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2013, pp. 196-203, doi: 10.1109/DFT.2013.6653606.

[3] M. T. Rahman, D. Forte, Q. Shi, G. K. Contreras and M. Tehranipoor,"*CSST: Preventing distribution of unlicensed and rejected ICs by untrusted foundry and assembly*," 2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT),2014, pp. 46-51, doi: 10.1109/DFT.2014.6962096.

[4] D. Zhang, X. Wang, M. T. Rahman and M. Tehranipoor, "*An On-Chip Dynamically Obfuscated Wrapper for Protecting Supply Chain Against IP and IC Piracies*," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2456-2469, Nov. 2018, doi: 10.1109/TVLSI.2018.2850807.

[5] J. A. Roy, F. Koushanfar and I. L. Markov, "*EPIC: Ending Piracy of Integrated Circuits*," 2008 Design, Automation and Test in Europe, 2008, pp. 1069-1074, doi: 10.1109/DATE.2008.4484823.

[6] K. Vaidyanathan, B. P. Das, E. Sumbul, R. Liu and L. Pileggi, "*Building trusted ICs using split fabrication,*" 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, pp. 1-6, doi: 10.1109/HST.2014.6855559.

[7] N. Limaye, C. Wachsmann, M. Nabeel, M. Ashraf, A. Kanuparthi and O. Sinanoglu, "*AntiDOTE: Protecting Debug Against Outsourced Test Entities*," in IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 3, pp. 1507-1518, 1 July-Sept. 2022, doi: 10.1109/TETC.2021.3102832.

[8] R.S. Chakraborty, S. Bhunia, *RTL hardware IP protection using key-based control and data flow obfuscation*, in: Proceedings of the 23rd IEEE International Conference on VLSI Design (VLSID), 2010.

[9] F. Mallouli, A. Hellal, N. Sharief Saeed and F. Abdulraheem Alzahrani, "*A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms*," 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2019, pp. 173-176, doi: 10.1109/CSCloud/EdgeCom.2019.00022.

[10] Sengupta, A., and Ray, U. K. (2016) *Message mapping and reverse mapping*

in elliptic curve cryptosystem. *Security Comm. Networks*, 9: 5363– 5375. doi: 10.1002/sec.1702.

[11] M. Jaiswal and K. Lata, "*Hardware Implementation of Text Encryption using Elliptic Curve Cryptography over 192 bit Prime Field*," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 343-349, doi: 10.1109/ICACCI.2018.8554410.

[12] Sheba Diamond Thabah, Mridupawan Sonowal, Rekib Uddin Ahmed, Prabir Saha, *Fast and Area Efficient Implementation of RSA Algorithm*, Procedia Computer Science, Volume 165, 2019, Pages525-531,ISSN1877-0509, https://doi.org/10.1016/j.procs.2020.01.024.

[13] N. A. Mohandas, A. Swathi, A. R., A. Nazar and G. Sharath, "*A4: A Lightweight Stream Cipher*," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 573-577, doi: 10.1109/ICCES48766.2020.9138048.

[14] M. Da Silva, M. -L. Flottes, G. Di Natale and B. Rouzeyre, "*Preventing Scan Attacks on Secure Circuits Through Scan Chain Encryption*," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 3, pp. 538-550, March 2019, doi: 10.1109/TCAD.2018.2818722.

[15] Emanuele Valea, Mathieu Da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre, "*Stream vs block ciphers for scan encryption*," Microelectronics Journal, Volume 86, 2019, Pages 65-76, ISSN 0026-2692, https://doi.org/10.1016/j.mejo.2019.02.019.

[16] C. Bharathi, K. Y. Annapurna, D. Koppad and K. Sudeendra Kumar, "*An Analysis of Stream and Block Ciphers for Scan Encryption*," 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), 2022, pp. 1-5, doi: 10.1109/PARC52418.2022.9726687.

[17] K. -J. Lee, C. -A. Liu and C. -C. Wu, "*A Dynamic-Key Based Secure Scan Architecture for Manufacturing and In-Field IC Testing*," in IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 1, pp. 373-385, 1 Jan.-March 2022, doi: 10.1109/TETC.2020.3021820.

[18] De Cannière, C. (2006). TRIVIUM: *A Stream Cipher Construction Inspired by Block Cipher Design Principles*. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds) Information Security. ISC 2006. Lecture Notes in Computer Science, vol 4176. Springer, Berlin, Heidelberg. https://doi.org/10.1007/1183681013