Authentication for Integrated Circuit and Devices Using Blockchain and Physical Unclonable Functions

Alessandro Augusto Nunes Campos[1], Tales Cleber Pimenta[2]

[1] Secretariat of Entrepreneurship and Innovation - Ministry of Science, Technology and Innovation, DF, BRAZIL
[2] Institute of Systems Engineering and Information Technologies - Federal University of Itajubá, MG, BRAZIL,
e-mail: [1] alessandro.campos@mcti.gov.br; [2] tales@unifei.edu.br

***Abstract* — Secure components and devices have always been and always will be a challenge for the electronics industry. In this sense, there is a constant and growing demand for new solutions that can allow reliability in the use and authenticity of components and devices. The end-user is not able to assess the existing risk, much less if the component or device is reliable in several aspects, mainly improper access to its information. This work presents a new integration of two technologies: Blockchains networks, which implement a kind of decentralized and inviolable database, which can increase resilience, security and guarantee against the alteration of the information registered in its structure; Physical Unclonable Functions (PUF), which allow the generation of a unique cryptographic key, since they use unique physical characteristics of each semiconductor component, considerably increasing security, the protection of industrial property and the opportunity for remote authentication of devices. The unprecedented contribution here is in the integration of existing technologies, in order to obtain an innovative solution of authentication and cyber security for the internet of things and others devices.**

***Index Terms* — Blockchain; Physics Unclonable Functions; Integrated Circuit; Authentication; Cyber Security.**

I. INTRODUCTION

Users in general, do not pay attention to whether the semiconductor components used in their daily use devices are free from safety risks, believing that the internal components perform their functions reliably [1]. Considering the possibility of improper access, deliberately created or not by a manufacturer/developer or by failures in its development, in addition to the risks found in low reliability integrated circuits (ICs), we intend to converge on a solution that increases the use of ICs safely and with a high degree of inviolability [2].

The relevance is mainly due to the considerable increase in the use of electronics in all kind of devices, in which people often trust their lives to these devices [3].

Many believe that ICs are free of the types of security flaws that are so common in software and impervious to the subversion of malicious codes [4]. This belief is supported a lot of times by the use of Trusted Platform Module (TPMs) and Hardware Secure Modules (HSMs), which are devices designed with high-security platforms for critical processes [5]. But is this faith really deserved? In recent times, the supply chain has been inundated with counterfeit components and new hardware Trojans have appeared as a threat to the reliability of ICs [6].

II. APPROACH

The idea for a security authentication system for semiconductor components (silicon) using PUF and blockchain technologies (reliable point-to-point networks) innovates in terms of decentralization and authentication without the need validation of a third party. Here it presents considerations necessary to understand this approach and description of problem in devices and integrated circuits. The proposal here is not to present a final methodology or a simulated system with your final evaluation or comparison with other security solutions, but to launch a way of integrating technologies that in next studies can be implemented and developed for safety for devices and ICs.

Assuming that the authentication process provides assurance about the user's identity, the individual (or device) whose identity will be verified, in this work can be called a "claimant". Credentials, on the other hand, are the evidence that the claimant presents to establish his/her identity. Thus, the authentication system needs to provide protection to the user against forms of attacks and unauthorized uses such as man-in-the-middle (MITM), spoofing and others, in addition to granting unicity credentials.

Finally, the idea here is to explore the main authentication paradigms and build a proposal against cyber-attacks and fraud, in addition to ensuring that the device "on the other end of the line" is who he/she claims to be.

In this work, we intend to solve the authentication paradigm called "something that is known" through passwords and PINs, when the unique identifier of a semiconductor component is generated and transformed into a secure key for authentication. We also worked on the paradigm called "something you have", since the authentication information will be published on the blockchain, and linked exclusively to the valid device. Finally, the paradigm called "something that identifies" will become evident when the extraction of physics unclonable functions is used, an exclusive characteristic of a semiconductor component, to produce the uniqueness of a cryptographic key.

Before dealing specifically with our approach, we will present the basic and specific definitions of the two technologies discussed in this work. In particular, in the section that deals with the proposal itself, the reason for the adopted option will be discussed and presented.

III. PHYSICAL UNCLONABLE FUNCTION

In the early 2000s, cyber security, or digital security, was only implemented in specific electronic devices such as ATMs, payment terminals, and bank cards. However, currently any bank transaction, in addition to several others that deal with secret or confidential information, already makes

Digital Object Identifier 10.29292/jics.v17i1.555

use of encryption techniques and uses some type of encryption algorithm to attempt protection on their transactions.

As a result of these developments, there has been a very considerable growth in the number of application-specific integrated circuits (ASICs) such as microcontrollers and chip systems (SoCs), which have embedded cryptographic accelerators in "hardware", or cryptographic libraries of "software". With this and the recent idea of the Internet of Things (IoT), and others kind of devices, pervasive cryptography emerged.

One might think that physical protection is not necessary in most cases. This is no longer true, as the automated reverse engineering associated with failure analysis techniques has made physical attacks accessible [7].

The traditional way of designing secure key storage is to store keys in non-volatile memory (OTP / ROM, EEPROM, or Flash) and to implement layout countermeasures or obfuscations, such as chip shielding, path shuffling, or creating fictitious paths [8]. A more robust solution depends on memory encryption using a master key, but the challenge is still the protection of the master key itself, which goes back to the initial challenge.

The main disadvantage of the obfuscation methods listed above is that they also require highly specialized knowledge, dominated by only a few IC designers. As a result, these solutions are not widely available and are therefore inapplicable in many cases.

However, Physical Unclonable Functions, delivered as Intellectual Properties – IPs, allow for high levels of security, even for non-security experts. A PUF is nothing more than a circuit or mathematical function that implements a unique signature of a device, exploring a specific characteristic that cannot be reproduced by any other. A fundamental difference between traditional techniques and PUF is that they are, by nature, practically immune to reverse engineering techniques.

Another challenge that the PUF solves is the need to protect the keys before writing them to the IC in the most used secure processes (smartcards and others). In traditional implementations, it is necessary to establish the keys at some stage of the manufacturing process, whereas in the implementation by PUF, that is not necessary.

A.PUF Features

Integrated circuits PUF have interesting properties for use in the generation and storage of secret keys. As the key is generated from the intrinsic randomness introduced by the inevitable variability of the manufacturing process, no explicit key programming step is necessary, which simplifies the distribution of keys. In addition, as this randomness is permanently fixed in the (sub) microscopic physical details of the chip, no conventional non-volatile key memory is needed.

During the initial generation phase, the PUF is consulted and the algorithm produces a secret key along with some additional information, often called auxiliary data. Both are stored in a safe place by the user. In the reproduction phase, the safe location itself presents the auxiliary data to the algorithm that uses them to extract the same PUF key as in the generation step. In this way, the device containing the PUF and the user himself established a shared secret key. It is possible to build these algorithms so that the key is perfectly secret, even if the auxiliary data is observed, that is, the auxiliary data can be communicated publicly to the device.

In Fig. 1, it can be observed that different measurement sequences at different time points in the same PUF, can produce different signatures from each other, for this, it is necessary to understand how close two signatures can be. This result is associated with the intra-class distance (µintra) of the PUF (proximity to the outgoing responses), which analyzes how similar the responses are for the same challenge of the same PUF (PUF A case). The metric used to calculate the distances between two signatures is the normalized absolute error between these signatures, that is, two signatures are subtracted from each other and the absolute value is calculated and then this difference value is normalized by the value of the elements of one of the signatures. The smaller this difference the better the PUF.

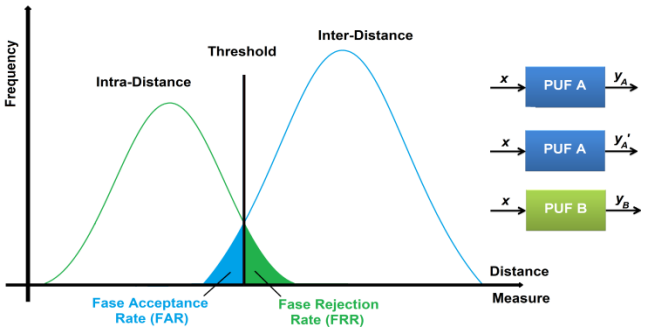

Fig.1 Details of identification in a PUF

In the case of different PUF on the same silicon, the challenge-response must be as different as possible and this is called the inter-class distance (µinter) of the PUF (distance from the outgoing responses). In this case, the comparison of the output between PUF A and B must be as far away as possible, and the calculation process is identical to the measurement of the intra-class.

It is important to point out that it is possible to obtain a value of intra-class distance greater than that of inter-class and inter-class less than that of intra-class, as seen in Fig. 1, when working with analysis at low frequencies. Those areas of interpolations are called the false rejection rate (FRR) and false acceptance rate (FAR), and should be avoided.

B. Consideration about PUFs

It is known that there are several solutions of possible physical unclonable functions that are very easy to implement in a small intellectual property of an integrated circuit, but it is worth discussing which solution could be more viable for this work.

This question pointed us to the following dilemma: being a very well-known and studied technology, where practically infinite possibilities of implementation have already been discussed, addressing technical and safety issues, what is the best solution we could propose?

Considering some implementations of PUFs from a microelectronics point of view, we can have: PUFs based on

intrinsic delay (Fig.2); PUFs based on intrinsic memory states (Fig.3), and others. It is important to remember that the implementation of a memory cell bank is more expensive and difficult to implement, since the solution of delay loops becomes simpler, but not more efficient.

PUFs of intrinsic delay start with an analog measurement of a random physical parameter, which is later quantized and can be used as a system-wide identifier. Although the non-formal definition of a PUF based on intrinsic delay is provided in the literature, it is necessary to distinguish two prerequisites for the PUF to be called intrinsic.

• PUF, including the measurement block, must be fully integrated into the on-board device;

• The complete construction of the PUF must consist of primitives that are naturally available to the manufacturing process of the embedded device

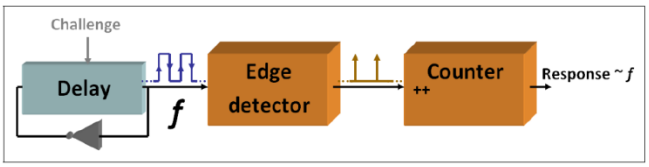

Fig.2 Basic Operation of a Ring Oscillator PUF.

PUFs of intrinsic state of memory can be obtained by considering that a digital memory cell is typically a digital circuit with more than one logically stable state. By residing in one of its stable states, it can store information, for example a binary digit in the case of two possible stable states. Now, considering the stored data, if the element goes into an unstable state, it's not clear what will happen. It may start to oscillate between unstable states or it may converge back to one of its stable states.

In the latter case, it is observed that certain cells strongly prefer certain stable states over others. Also, this effect cannot usually be explained by the cell's logical implementation, but it turns out that instability can be caused by manufacturing variation. For this reason, the steady state of a destabilized memory cell is a good candidate for a PUF response. There are different proposals in the literature, based on different types of memory cells, such as SRAM cells, latches and flip-flops.

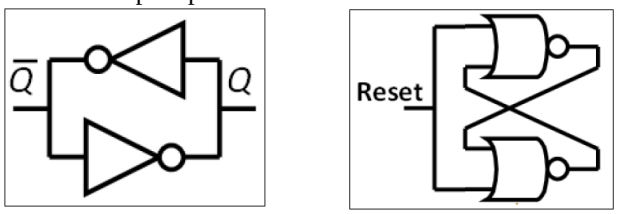

Fig.3 Logic circuit of a SRAM cell and a latch (PUFs).

By evaluating the proposal of this work, it is understood that the greatest merit here is not in implementing the best possible PUF solution, since response speed, silicon area or others most desired, as it has already been tirelessly explored by the research and public literature in recent times, but rather to present in a simple and clear way how to extract these implicit characteristics of integrated circuits and use them in a new solution.

IV. BLOCKCHAIN (DECENTRALIZED PROTOCOL)

Blockchain is a decentralized database, controlled and verified by all actors who need to transact digital assets on the internet. The blockchain allows one entity to transact directly with another, without the need for a centralized transaction authenticating authority.

On the other hand, Bitcoin is a digital currency that was created and structured on the blockchain to allow attributing value to transactions in this digital environment. Bitcoin is not the only digital asset existing on the internet today that makes use of blockchain, but certainly, it is the most valuable and also most famous.

Blockchain is the essence of the Bitcoin protocol, proposed by Satoshi Nakamoto [9], which came into operation in 2009. The initially proposed article describes a point-to-point (P2P) network where transactions with the Bitcoin cryptocurrency are received by decentralized servers (miners), which through real-time processing, will validate this operation. The process is carried out through a specific consensus protocol based on cryptographic challenges (HASH), test the validity of each transaction and the order in which they will be permanently stored in a chain of blocks (blockchain) replicated on the internet on each node or server that collaborated to process it after the consensus of validation of all.

In the evolution of the blockchain project, three phases are highlighted [10]: blockchain 1.0 corresponds to the launch of Bitcoin in 2008, with the first implementations of cryptocurrencies, and an ecosystem of applications and payments with digital assets. Blockchain 2.0 started with the innovative proposal for smart contracts in 2013 (Ethereum Assent), and the full range of possible financial applications. Blockchain 3.0, on the other hand, is characterized by the adoption of blockchain technology to benefit applications in several areas, in addition to finance, digital health is one of the main applications.

The operation of the blockchain is structured as follows: each node connected to the blockchain network has a copy of the original transaction database of that network since the first operation. As seen in Fig. 4, each new transaction is authenticated by the hash, registered, and linked to the previous record, thus, it becomes computationally impossible for a third party tries to change any record before the current block since, in addition to changing it, it will need to change each node on the network. Recalling that each record in the blockchain is created approximately every 10 minutes. Finally, the “past” of a blockchain record is computationally immutable, or very hard to do, only the “present” remains recorded and, at any time, it is possible to see the entire “history”.

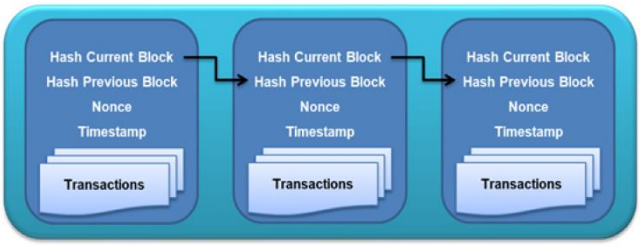

Fig.4 Graphical representation of a blockchain

Thus, the integrity of the information is guaranteed, since it is impossible to change old blocks without changing the entire subsequent blocks, which would be perceived by the network nodes. A blockchain can be understood as an initial state followed by several of transactions grouped in a block.

A. Current applications

The Blockchain is characterized by the adoption of the benefit of use in diverse applications, the potential for transformation is immense and applications are emerging from this technology in numerous sectors mainly on finances and health in addition to computing itself (network protocols, cloud, and fog, IoT, etc.). See some of these innovations;

Smart Contracts: Automated contracts that are incorporated as an "if-this-then-that" code, which gives them self-execution.

Identity Management: For online transactions, a secure identity is important and it is very difficult to find, in this case, there are huge opportunities in this area.

Protection of Intellectual Property Rights: Plagiarism and the exploitation of intellectual property are particularly problematic situations in the digital world. Blockchain can be a great help in this regard, knowing that copying and redistribution can be avoided.

International Payments and Transfers: The money goes from one bank to another and this process continues until reaching the final accounts. The problem is that each bank has its own cash book, making it necessary to have reconciliation at the end of each day, making the process expensive and very slow. Blockchain will transform that.

Education: Many institutions grant certificates of various types. A blockchain could be used as a credential repository and any institution could add its credentials to the blockchain, and anyone who needed it could verify it.

Digital Voting: When multiple parties are involved in taking care of the voting mechanism with blockchain, it becomes quite implausible to tamper with such a system.

Health: The blockchain can be used as a mechanism to control access to medical records and can provide an audit trail for accessing those records. Another advantage is its ability to improve interoperability between clinics, hospitals and other agencies in the healthcare sector. They can also provide a reliable method to track pharmaceutical products throughout the manufacturing and distribution process, reducing the problem of drug counterfeiting. It could be used to fight medical insurance fraud, among others. In conjunction with IoT devices, used to measure factors such as temperature, blockchain technology can also be used to verify proper conditions for storage and transport of products or to authenticate the quality of medicines. Remember that this last case could be directly applied as a result of this work.

B. Ethereum

Ethereum [11] refers to an open-source software platform based on blockchain technology, which allows developers to create decentralized applications that are also called dApps. However, the word Ethereum is also used to refer to the Ether currency (ETH), a cryptocurrency created on the Ethereum platform.

Ethereum's history begins with Vitalik Buterin, in 2011. Buterin became aware of Bitcoin's shortcomings and created Ethereum as superior blockchain technology. With this approach, the founder of Ethereum presented a list of descriptions, such as, for example, the opportunity to register and execute on his blockchain automatic contract execution actions, which anticipated several innovations in the cryptocurrency ecosystem.

Among the tools that have become popular on the network are decentralized applications, non-expendable tokens, and decentralized finance (DeFi).

Smart Contracts (SC) - For the implementation of these contracts, the creators of Ethereum introduced the programming language Solidity [12]. "A complete Turing language that makes it easy to program a computer to perform a variety of operations". In this way, the language allows anyone to create smart contracts on the blockchain. To do this, you only need to write the logic in a few lines of code.

Decentralized applications (dApps) - These are pieces of code written in smart contracts. The dApps communicate with the blockchain and are programmed to control various actions. They process, for example, the external information they receive, while the codes are executed on a P2P network. The functioning of a dApp depends on two elements: a network like Bitcoin or Ethereum and an execution environment. The blockchain allows the application to have a decentralized network infrastructure.

C. Consideration about Blockchains.

It is important to stress that the blockchain technology of the Ethereum network that enables the implementation of smart contracts is not the only one available today. There are several other recently created blockchain networks that implement solutions that are identical and even better than Ethereum, but they have not yet reached a critical maturity that allows the safe operation of a platform whose main objective is the authentication and security for integrated circuits and devices.

It is important to note that every decentralized network created after Bitcoin came to try to solve the blockchain trilemma that is based on three fundamental aspects: Security, Decentralization and Scalability. Bitcoin currently has the highest computational power among networks, but the amount of transactions performed per second (scalability) at layer 1 specified in the protocol is not enough for it to be used as a means of payment in real time. As for security (strength against network attacks) and decentralization (ability to keep transaction authentication as distributed as possible), there is no other blockchain to match its capacity.

With the intention of presenting a discussion in advance about which network could efficiently serve best our proposal, it is first necessary to understand that it is only possible to implement smart contracts in blockchain networks that allow such a feature, thus drastically reducing the number of protocols that we could use. An example would be the exclusion of Bitcoin itself for not implementing such contracts on its blockchain.

Within these protocols, the second option that we must consider is how secure this protocol is, after all in our work

the focus is on guaranteeing authentication, that is, a network with low capacity to suffer attacks and with a greater history of vulnerability resolution must be the preferred choice. On the other hand, the decentralization factor, which greatly supports the up-time capacity and operational resilience of the network, must be considered. Finally, the characteristic related to scalability, although important, may not be a priority at this time. Bearing in mind that due to the existential characteristic of a trilemma, we will always meet 2 requirements, to the detriment of a third, on the lack of the trilemma's own ambiguity.

To assess which are the best solutions for blockchain networks that implement smart contracts, please observe Fig 5, which presents a list of the main networks and their functionality. They are classified according to the trilemma of blockchain networks.

SMART CONTRACT PLATFORMS COMPARISON (MAINLY)

	ETHEREUM	CARDANO	BINANCE	POLKADOT	SOLANA	FANTOM
Architecture	Single-chain (synchronous)	Single-chain	Single-chain (synchronous)	Mulchain (parachains)	Single-chain (synchronous)	Single-chain (synchronous)
Security	Global	Blockchain-specific	Shared	Shared (if parachain connected)	Global	Global
Concensus	Proof-of-Work	Proof-of-Stake	Proof-of-Stake (Autority)	Nominated Proof-of-Stake	Proof-of-History	Proof-of-Stake
VM/Development	EVM (Solidity, Vyper)	K(EVM)	EVM (Solidity, Vyper)	WebAssembly, Substrate	Sealevel (Rust)	EVM (Solidity, Vyper)
Validator/Miners	301250	2076	21	297	1044	44
Economics	Variable transaction fees	Variable transaction fees	Variable transaction fees	Market cost for parachain slot	Variable transaction fees	Variable transaction fees
Governance	Off-chain	On-chain (not yet)	On-chain	On-chain	On-chain	On-chain
Ecosystem	5000 + projects	200+ Projetcts	857 + projects	495 + projects	338 projects	140+ projects
Market Cap	397B	76B	63B	$38,7 B	47,7B	31B
Live Network	jul/15	jul/20	ago/20	ago/20	mar/20	ago/20

Better Feature Medium Feature Worst Feature

Fig.5 Platforms Comparison Blockchain Trilema.

The main point in this proposal, regarding the best blockchain network to use at this time, has an important direction.

V. Consideration about other authentication and Security solutions

Before discussing a new proposal on authentication of integrated circuits and devices, it is necessary to know some other solutions that could solve this same issue or increase their security classification.

Considering the case of Untrusted Manufactured Integrated Circuit, the user like the United States Department of Defense, that intends to carry out the development of an application that involves secrecy or sensitive information, necessarily needs that the semiconductor components used are free from intentional risks of unauthorized access created in the design of the component. Thus, the following solutions try to cover such gaps and guarantee the reliability of the components with the proposed techniques.

We know that the implementations of Hardware Trojan Detector and Counterfeit ICs measures can effectively present a good security solution, but we will briefly present some of its main features.

Hardware Trojans Detector have been developed in recent years classified or categorized as methods that make use of techniques of Side-channel Analysis or Trojan Activation, mainly for chip-level solutions and other architecture-level detection.

Side-channel Signals, including timing and power, can be used to detect Trojans. Trojans often change the parametric characteristics of a design, for example, degrading performance, changing power characteristics, or introducing on-chip reliability issues. That influences the power and/or delay characteristics of the wires and ports in the affected circuit. Power-based side channel signals provide visibility of internal structure and activities within the IC, allowing detection of Trojans without fully activating them. Timing-based side channels can detect the presence of a Trojan if the chip is tested using efficient delay tests that are sensitive to small changes in circuit timing along the affected paths.

On the other hand, Trojan Activation strategies can speed up the Trojan detection process in cases that were combined with Power Analysis during implementation. If a portion of the Trojan circuit is activated, the Trojan circuit will consume more dynamic power, which will help further differentiate the power traces in Trojan-inserted and/or Trojan-free circuits.

Counterfeit ICs containment measures are promoted not as integrated electronics solutions, but as reports produced by the industry itself and trusted institutions, on the discovery and identification of counterfeit components. In recent years the five most common components found to be counterfeit are: Analog ICs, Microprocessor ICs, Memory ICs, Programmable Logic ICs and Transistors. With this steady increase in reported incidents, there is a need for effective methods of testing parts and maintaining proper records as components pass through the supply chain. The committee responsible for many standards on component certification and counterfeits is the G-19 Counterfeit Electronic Parts Committee, established by SAE International. Its standards target three different industry sectors: distributors, users and testing service providers (i.e. test labs).

A collection of the standards they have written or are currently working on are:

• AS6081—Counterfeit Electronic Parts Avoidance, Distributors • ARP6178—Counterfeit Electronic Parts; Tool for Risk Assessment of Distributors, Distributors & Users

• AS5553—Counterfeit Electronic Parts; Avoidance, Detection, Mitigation, and Disposition, Users

• AS6171—Test Methods Standard; Counterfeit Electronic Parts, Test Providers

As previously stated, we still need solutions that not only address these issues, but that can go further and support authentication in the daily use of semiconductor components.

VI. The proposal

With the Internet of Things (IoT) or in the more specific type, like IoT for medical devices or personal Identify, at the same time that we have the possibility to develop a range of interesting applications (traceability, sensitive data protection, remote device authentication, etc.) there are new challenges, especially regarding security and privacy. In this sense, the matching of IoT and blockchain can be a differ-

entiator for providing a new computational layer for sharing and securely analyzing data, with greater guarantees of privacy and security. In this way, this layer can be used to authenticate, authorize, control and audit the information generated by those smart devices.

Currently, there is no still solution established in the market or standard for the use of certification and authentication for integrated circuits or devices that make use of blockchain and remote access. Even without the internal implementation of cryptographic solutions, the possibility of registering and subsequently consulting on the authenticity of the ICs manufacturing origin is a fully plausible process using others approaches.

In this way, generating a randomized code with the highest probability of uniqueness possible, blockchain can be used to register, certify and guarantee, with a high degree of reliability, the authenticity of a semiconductor component.

By the way, one can deal with the authentication of any hardware device, both in the traceability of the manufacture of its components, as well as in the constitution of a new device registered and validated on the blockchain.

This article presents an integrated solution that, through specific implementations, will be able to control the activation, make authentication, in addition to other functions in an IC and will protect it against use, copying, and other improper violations, in addition to providing its authenticity and certification of use. Consequently, the device that makes use of the IC, also be secure.

A. PUF proposed

In order to advance the process of demonstrating this idea, we present here a previous solution already developed by other researchers [13] who implemented a PUF based on ring oscillators (ROs) in programmable logic circuits (Field Programmable Gate Array - FPGA). The implementation of this type of PUF clarifies that such structures are a type of delay-based PUF since the value of their outputs varies according to the delay of the circuit as a whole.

Ring oscillators can be easily implemented using digital ports. This can be achieved by connecting an odd number of inverter gates and feeding it back, as shown in Fig 6. Thus, the output changes its logic value after the time corresponding to the delay of all gates, as well the output is connected to the input, this process repeats itself, producing the oscillator circuit.

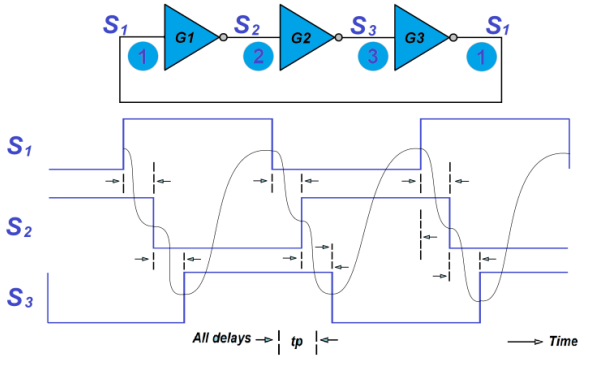

Fig.6 Ring oscillator

(With N=3 inverters and signal output from logic gates S1, S2 and S3. The signal period is T=2.N.tp, the oscillation frequency 1/(6tp))

A very important factor of this implementation is the variation in the frequencies that the oscillators may suffer naturally, mainly due to variations in the supply voltage, temperature or other external parameters. Thus, it is necessary to implement ways to compensate for this inconvenience and one way is to use a division between obtained frequency values [14]. In this model, two delay circuits are chosen to be used on the oscillator. Their frequencies are sampled and the result is the ratio between the two frequencies. In cases where the frequency of the oscillators tends to vary linearly with the temperature, and/or supply voltage, this becomes very useful, causing the ratio between two frequencies to produce a fixed value. The value obtained for the inter-class (μ_inter) and intra-class (μ_intra) distances with these methods are [15], μ_inter ≈ 10 · ⟦10⟧^(-3) and, μ_intra ≈ 0.1 · ⟦10⟧^(-3)

Functional Detailing - The implementation of the PUF based on ring oscillators was made using a programmable logic device FPGA Cyclone II from Altera embedded in the Cyclone II DSP Development Board, where its internal connections and its “truth tables” are reconfigurable. Thus, it is possible to implement hardware in different ways and test several different solutions.

On this board, we have a JTAG input connected to a USB port that allows the configuration of the FPGA. The Altera Quartus II software was used to carry out the communication and programming of the FPGA, in addition to modules such as SignalTap II, which allows the mapping of a logic analyzer within the FPGA, and the Chip Planner application, which allows visualizing the regions of the FPGA that were used in the programmed logic.

Physical Implementation - The implemented PUF contains a bank of 256 oscillators with 5, and later 10, delay cells (new oscillators) that are enabled according to the measurement need. Circuits for compensation, by comparing two frequencies, are not used in this implementation. Each oscillator (delay cell) consists of an odd number of logic gates and an enabling XOR circuit, as shown in Fig 7.

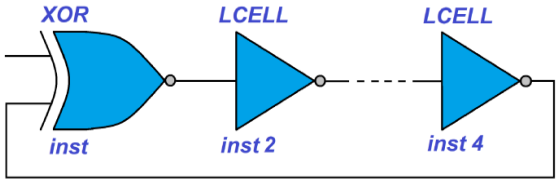

Fig.7 Implemented oscillator

The implementation of the PUF is divided into two components: one called subPUF and another called estimator, as shown in Fig 8. The subPUF is composed of the banks of oscillators and a circuit capable of selecting and driving the output at the frequency measured by the desired oscillator. This oscillator is then enabled, sends its signal to the subPUF output, which then takes it to the input of the next circuit called an estimator.

The estimator, in turn, receives the signal from the subPUF oscillator and evaluates its frequency concern to the internal FPGA clock. This frequency is then taken to its exit, which is then taken to the exit of the PUF.

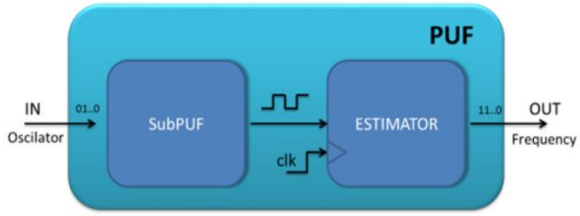

Fig.8 Schematic diagram of the implemented PUF

The subPUF consists of a decoder, a multiplexer, and a bank of oscillators. The enable signal of each Ring Oscillator (RO) is connected to a decoder output and each RO output is connected to a multiplexer input and the decoder and multiplexer selection signals are connected to the same node. In this way, the same oscillator enabled by the decoder will be selected by the multiplexer. A schematic diagram of this circuit is given in Fig 9.

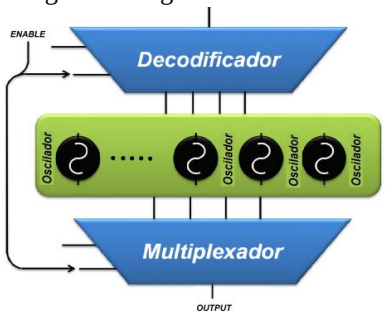

Fig.9 Schematic diagram of the subPUF

Thus, for the circuit described, given a challenge (oscillator), there is a response (frequency of the oscillator). The parameters that regulate the frequency of the oscillators, however, vary according to the internal components of the oscillator and vary according to the FPGA. Thus, for the same oscillator design, it is expected to have different oscillators in different FPGAs. This type of behavior where different devices generate different outputs given the same design is promising in the manufacture of PUF.

B. Blockchain proposed

Nothing would make sense in this work if it were not possible to introduce some innovative and possibly disruptive solution that could have the potential to transform security in access to semiconductor-based devices.

Ethereum is a platform created specifically to promote smart contracts. But these new tools should not be used in isolation; they can also form the building blocks for “decentralized applications”, and even autonomous companies that are completely decentralized.

Ethereum's infrastructure has stood out in two segments that have strong growth power, Decentralized Finance (DeFi) and Decentralized Applications (dApps).

DApps: Ethereum DApp allows the creation of decentralized, open-source applications that use the Solidity programming language, on a blockchain, to promote security. Thus, this type of application is capable of storing data and information with blockchain technology, which does not have centralized control.

Thus, we understand that, at the moment the most appropriate technology to be used in this work, as discussed above, is the Ethereum blockchain network, since it better meets the trilemma of blockchain networks, it implements smart contract solutions and it has the robustness of strength and security suitable for this proposal.

C. Operational model proposed

In this idea of smart contracts, we cover the demands and needs of the initial application, that is, functions are implemented to register a new device (ASIC/FPGA) and authenticate it remotely.

As seen in [16], it is first necessary to implement property keys and addresses to uniquely identify a component in a new ASIC or even in an already produced FPGA. This box is implemented in a programmable logic solution, as previously described, to avoid any forgery and/or authentication problems. All components are assigned public and private key pairs.

The component owner can manage keys and addresses them using a digital wallet. The digital wallet can be used to carry out any transaction. Digital wallet is a specifically designed software that contains the public-private keys and the component's address (Id). The smart contract (SC) flow can be seen in Fig 10.

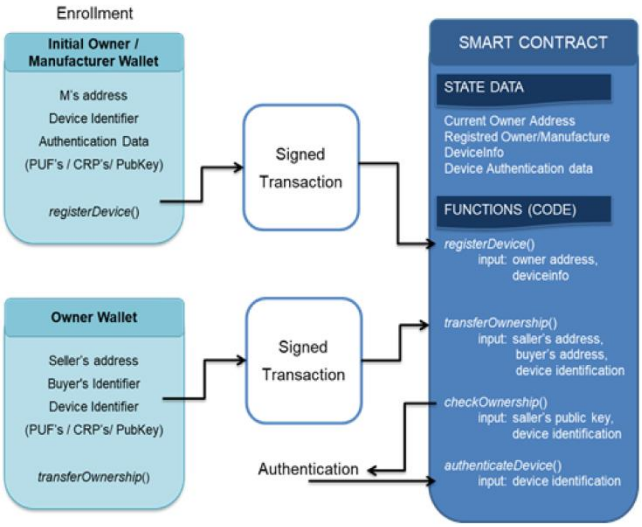

Fig.10 Smart Contract Implementation flow

It is created by the security solution owner or manufacturer, to maintain uniform applicability and usability for all devices. In this case, the smart contract provides the services of device registration, verification of information, and change of ownership and authentication.

Device registration - To introduce a device to the blockchain, it must be registered first. As shown in Fig 11, the pseudo-code of the function that implements smart contract registration is *registerDevice*. This function registers a device if the sender of the message is the owner of the security solution or manufacturer. The so-called *deviceInfo* includes data for identification and authentication of the component. The identification data is used to search, among a collection of devices, the target device being consulted. It can be, for example, a device serial number, electronic product code, or another specific identifier. This device data is used for identification purposes and not as the primary means of authen-

tication. For authentication, the owner of the PUF data uses these dates to input to the *registerDevice* function.

```
inputs: manufacturer's address (addrManufacturer),
        and device information (deviceInfo)
if message sender is in manufacturer's list them
    specify owner of the device as addrManufacturer
    register deviceInfo on de blockchain
else
    do nothing
end
```

Fig.11 Pseudo code for registration (*registerDevice*)

Ownership verification - Ownership of the device can be verified using the *checkOwnership* smart contract function shown in Fig 12, where you can see its pseudo code. This function checks the ownership of the device against the address of the owner. If the device with the provided identifier belongs to the sender, it will return True. The function is called when the device owner wants to confirm its registration source.

```
inputs: seller's public key (sellerPubKey), and
        device identifier (deviceIdentifier)
outputs: a boolean True or False
if (hash(sellerPubKey) ==
  blockchain[deviceIdentifier].owner) them
    return True
else
    return False
end
```

Fig.12 Pseudo Code for property verification (*checkOwnership*)

Property transfer - A secure transfer of ownership is necessary if the end customer wishes to implement their authentication or data acquisition solution on their own. To transfer ownership of a device, the smart contract implements the *transferOwnership* function, as indicated by the pseudo-code in Fig 13. This function transfers ownership of the device (*deviceIdentifier*) from the seller to the buyer (*addrBuyer*). First, the function checks whether the sender of the message is the owner of the device with *deviceIdentifier*. If this is true, the function assigns the result of the *addrBuyer* function as the new owner of the device.

```
inputs: buyer's address (addrBuyer), and device
  Identifier (deviceIdentifier)
if (addrMessageSender ==
  blockchain[deviceIdentifier].onwer) them
    set blockchain[deviceIdentifier]. Owner = addrBuyer
else
    do nothing
end
```

Fig.13 Pseudo Code for transfer of ownership (*transferOwnership*)

Device authentication - Any system trying to communicate with a device must confirm that the device is authentic. This is implemented by the smart contract function *authenticateDevice* as indicated by the pseudo-code in Fig 14. This function starts the device authentication process for a given identifier. The process basically implements functions to obtain the challenge-response data for a specific *deviceIdentifier*, applying the challenge to the device, calculating the response, combining the challenge-response pairs, and verifying the authenticity of the device.

```
inputs: identifier of the device (deviceIdentifier)
outputs: a boolean True or False
set deviceChallenge = blockchain[deviceIdentifier].Challenge
get deviceResponse from the device after applying
  deviceChallenge
if (deviceResponse ==
  blockchain[deviceIdentifier].Response) them
    return True
else
    return False
end
```

Fig.14 Pseudo Code for Authentication (*authenticateDevice*)

D. Authentication protocol and secure data transmission

Here is presented the idea of a protocol for device authentication and secure information transmission as an application example, between a signals monitoring solution (IoT device) and a data acquisition software platform (dashboard). The protocol, shown in Fig 15, consists of:

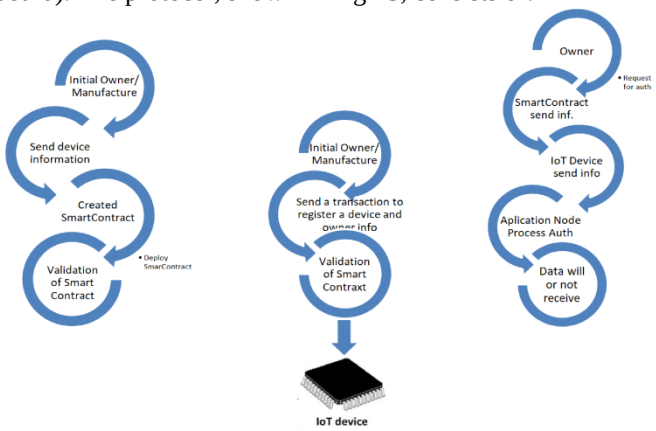

Fig. 15. Diagrams for registration, validation, authentication and data transmission

Creating a smart contract for resources and updating firmware - To automate the device authentication process, the manufacturer first creates a smart contract. The security solution owner or manufacturer sends the initial device information to the blockchain node for the purpose of creating the contract. Device information includes the owner's wallet, device ID, and authentication data (PUF key, CRP, Public Keys, etc.).

Deployment of smart contract for configuration of authentication features - The security solution owner or manufacturer node on the blockchain creates the smart contract. After creating the smart contract, the supplier's node deploys it to the blockchain. Once validated by the nodes, the smart contract is added to the blockchain as a transaction, and an address is assigned to it. After this initial transaction, the contract becomes part of the blockchain forever and its address never changes. The registry owner incorporates this address into the device's security module in Fig 16.

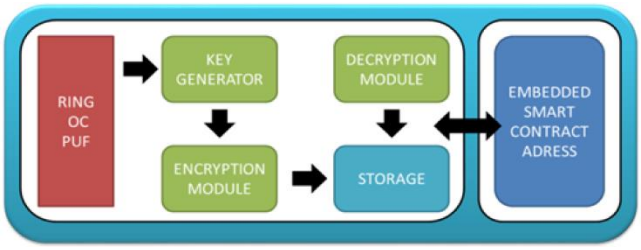

Fig. 16 Detailed diagram of the security module

Enrollment of a device by the owner or manufacturer - When the owner or manufacturer wants to register a device, he/she sends a *registerDevice* transaction to the smart contract. The manufacturer also sends the device information, such as device identifier, authentication information (PUF data) in the transaction, among others. For this, the first transaction to register a device is referred to as a "genesis" transaction. The owner can also register an N number of devices in the same transaction, including the corresponding information for all devices. This facilitates the scalability of the protocol. As all blockchain transactions are digitally signed by the actor who creates it, a possible counterfeiter cannot fraudulently claim to be a manufacturer and change the registration.

Requesting authenticated device information - Using the address embedded in the security module, the owner sends an authentication request through its own system, to the smart contract, as shown in Fig 17.

The blockchain node receives the request and checks the requirements in the contract that corresponds to the request received. If the requirements are met, the node sends the corresponding signaling for authentication of the requesting device. Before sending authentication, the smart contract verifies ownership of the requesting device.

The main components of the Security Module and their functions are described below:

Key generation module - During the enrollment phase on the owner's or manufacturer's system platform, the key generator in the authentication module generates a pair of public and private keys. Using the private key, a device can create a signature on a message protecting the integrity of the message and proving its authenticity. At the end of receiving the message, the authenticity of the signature can be verified using the public key corresponding to the private key.

Encryption module - The private key is encrypted with a "second encryption key" generated from the PUF and stored in a non-volatile memory. The secondary encryption key used is obtained from the PUF. The owner registers the public key on the blockchain by issuing a *registerDevice* transaction to the device as shown in Fig 17a.

Decryption module - During the authentication phase, the encryption key can be generated instantly from the PUF output. Using the encryption key, the decryption module generates the device's private key. The device can be authenticated by invoking the *authenticateDevice* function with its identifier. The smart contract sends a challenge message generated with a pseudo-random number generator to the device. The device's additional cryptographic module creates the device signature using the private key. Using the public key, the smart contract can verify the digital signature for the authenticity of the device, as shown in Fig 17b.

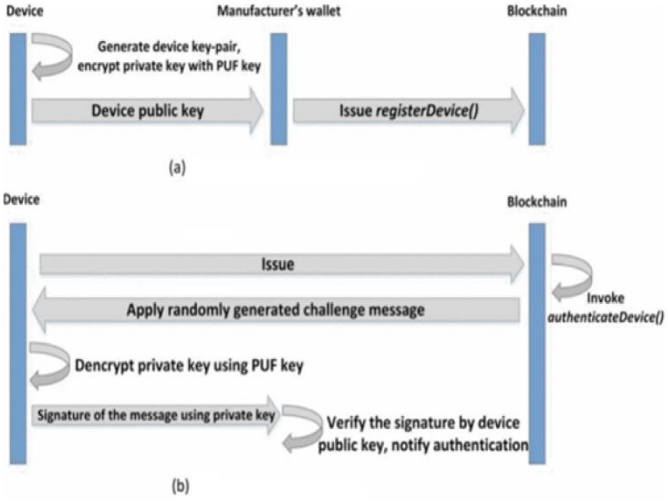

Fig. 17 Device registration process (a); Authentication process (b)

After the device's data acquisition system makes a request, the component prepares the data to meet the requested shipment. The communication infrastructure must implement the properties of the US Intelligent Agency - CIA triad, that is, confidentiality, integrity and availability. For integrity of information and authentication of the sender, the system creates a digital signature based on the uniqueness of the PUF created internally in the component.

To maintain confidentiality, the manufacturer encrypts with a symmetric key generated by the secondary key of the PUF and further encrypts the symmetric key with the public key provided in the smart contract so that only the device with the corresponding private key can decrypt the symmetric key. Then, the system that requested the information decrypts the data using the symmetric key obtained.

VII. SECURITY EVALUATION

As we can see in [17], we need to take into account the security analysis. This is the first part of evaluating whether the application meets the minimum requirements to be eligible for use. In the security analysis we have to evaluate three aspects; secure storage, tamper proof and privacy protection.

A. Secure storage

Data storage security is an important feature. Basic information and features like public encryption key and other basic information about the device is stored in the blockchain and even though it is publicly visible, it cannot be tampered with. The device manufacturer implements the initial registration through an initial hash generation operation and stores it in the blockchain through a smart contract. The device's private information is encrypted and stored locally. When an application needs and/or the device needs to send information, it asks the *smartcontract* registered in the blockchain to run the verification function, which in turn returns to the device the challenge message which, after the local validation process, allows notify the application of the validity or not of the data sent, ensuring the authenticity and confidentiality of the data source. In addition, other infor-

mation registered in the blockchain is also sent.

B. Tamper-proofing

As seen, all basic information linked to the blockchain is public, tamper-proof and recorded in chronological order. The blockchain's consensus engine makes its trust based cryptographic algorithms without relying on a third party to secure the recorded information. Once the data is written to the blockchain, it cannot be tampered with, because each block is saving in addition to the device information, the hash value of its previous block. If we want to modify the data of a block, we need to have at least 51% of the computational power of the blockchain which is, from a practical and financial point of view, impossible. The hash value linked to the initially registered devices is preserved in the blockchain, and any change to the original data will cause a change to its hash value, thus also directly guaranteeing the inelastic non-tampering of the device registration.

C. Privacy protection

First, it is necessary to remember that the dynamics of sending data from the device to the requesting (application or database) of the end user has a specific connection (wired or not, proprietary or public) and not necessarily secure, so it is necessary to try to guarantee a secure channel for the transmission of this information. Because of this, the data sent to the requesting may or may not be from the original device, thus, to validate the information received and authenticate the device that is sending it, the requesting application submits a request anonymously, searching the blockchain for the record of the device and requesting the execution of the authentication function in the related smart contract. Each data sent from the device is encrypted, which may or not generate new public-private key pairs by the requesting application (wallet). In this case, the application must update the device registration in the smart contract depending on the need, thus protecting the identity of the end-user.

Second, the distributed blockchain record does not contain the device's privacy information, only its public key and identification (Id) having its encrypted private key (PUF key) stored locally on the device itself. Third, as only the device identification information is stored in the blockchain, the unauthorized data requester cannot perform updates to the smart contract registry, thus, with a good key update policy, it is possible to prevent unauthorized access to valid data sent by the authentic device.

This way, as there is no data produced by the device on the blockchain, only authentication capabilities, it is impossible to get any real data transmitted from the device from the blockchain.

VIII. Expected Compensations

As a way of trying to evaluate the possible advantages and disadvantages of this work, it is first necessary to take into account the scope of the proposed solution and others already presented. In this way, it is possible to understand that both the Hardware Trojan Detection (HJD) and Counterfeit IC solutions, in addition to those mentioned at the beginning of this article, such as Trusted Platform Module (TPM) and Hardware Secure Modules (HSM), despite being very valid, they do not cover the proposed scope so well, especially with regard to authentication not only of the component but also of the information sent by it.

Considering that and trying to make a parallel about the compensations of the use of this proposal, we can take into account two important points; one is the effort to implement the physical solution or Intellectual Property (IP) in the semiconductor component and another, in the decentralized application solution.

It is already possible to understand that the physical implementation of a PUF, in addition to all the accessory circuits for the operation of this solution, are relatively simple to implement in silicon, and several of these blocks are already popular on shelves of Design and IP marketing. In this way, even without a thorough evaluation, it is possible to say that the cost here is much lower than in others such as TMP, HSM and HJD.

Considering the point of view of the software application made available in the form of Smart Contract on the Blockchain, it is possible to understand that this would be the great collaboration, as it implements a compensation and a gain not yet available in commercial solutions, since in addition to increasing the robustness of the solution for the decentralization of authentication, it allows evolutions through updates in the Smart Contract over time. It is worth to remember that such functionalities can only be possible through the use of specific firmware that would require an individual connection with each component for updating, thus generating a risk of interception of communication and injection of malicious code into the system, besides the costs that would be much greater than blockchain solutions.

Finally, it should be understood that this work alone does not exhaust the implementation opportunities. Since PUFs are not such a new technology, possible trade-offs can be evaluated in choosing one or the other, but ring oscillation systems are the simplest to implement and therefore have the lowest cost, despite studies showing that in terms of safety, there are better ones. In terms of blockchain, it is important to remember that the proposal of this work suggest Ethereum. The more we know about the opportunities that are emerging in these ecosystems and that possibly tomorrow we will be able to have a new decentralized network that is safer, more efficient and cheap that can replace it. Thus, even without a formal accounting of gains and losses, it is understood that this is a viable, cheaper and safer solution than those available on the market today.

IX. Conclusions

This work presents an idea for implementation of an authentication system for semiconductors, sensors and others. This idea can be used not only for a specific integrated circuit, but also for a complete device solution.

We describe here a physical unclonable function (PUF) and his information extraction; the IC functional schematic for information processing; and the high-level software application that can implement this authentication solution.

The innovation here is not related to the proposition of a differentiated PUF that could increase security paradigms or be more resistant to new forms of hacking attempts. Not even in the structuring of a new intermediate hardware platform (hardware + firmware), which implements verification and validation procedures for components, or even in new high-level systems. The contribution of this work lies in the integration of recent and old technologies to achieve a new form of authentication and cyber security for semiconductors and devices in a simpler and more decentralized way.

In the end, the intention is not to present a full implementation of a solution with your entire functional platform or a system with simulation and data validation to compare with other solutions but to put a model very qualified and detailed to stimulate your implementation and future works.

For this reason, this work did not perform a complete assessment of economic costs, computational complexity, or performance. It will be necessary to implement a complete functional model, tests, and simulations. It will also be interesting to evaluate and implement new features that are still under evaluation that may turn out to be important additional features.

Furthermore, the software solution presented in this work is not implemented in a specific programming language such as Solidity for the Ethereum blockchain, because at this point we are considering a broader pseudo-code view that can be developed in other smart contracts solutions in blockchains more recent, such as Cardano, Polkadot, Binance Smart Chain, Fantom, among others.

Finally, we understand that the value is in the proposal of the model and in the integration of solutions, pointing out specific paths and analyzing its viability, thus promoting innovation in this area.

REFERENCES

[1] Grauman, B. “Cyber-security: The vexed question of global rules. Bruxelas” SDA – Security & Defense Agenda, 2013. Available from:https://www.files.ethz.ch/isn/139895/SDA_Cyber_report_FINAL.pdf

[2] Cruz Jr., Samuel César da “A segurança e a defesa cibernética no Brasil e uma revisão das estratégias dos Estados Unidos, Rússia e Índia para o espaço virtual”; IPEA, 2013. Available from: http://repositorio.ipea.gov.br/bitstream/11058/1590/1/TD_1850.pdf

[3] M. Bishop, "What is computer security?," in IEEE Security & Privacy, vol. 1, no. 1, pp. 67-69, Jan.-Feb. 2003, doi: 10.1109/MSECP.2003.1176998.

[4] Layton, T. P.; "Information Security: Design, Implementation, Measurement, and Compliance”, CRC Press, 2016 ISBN: 1420013416, 9781420013412; 2016.

[5] Benedikt Gierlichs, Axel Y. Poschmann “Cryptographic Hardware and Embedded Systems – CHES 2016: 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings”, Springer, 2016 ISBN: 3662531402, 9783662531402

[6] Kelly, S.; Zhang, X.; Tehranipoor, M.; Ferraiuolo, A.; “Detecting Hardware Trojans using On-chip Sensors in an ASIC Design”, Springer - Journal of Electronic Testing - Volume 31, Issue 1, pp 11–26; 2015.

[7] Enamulquadir, Mdshahed & Chen, Junlin & Forte, Domenic & Asadizanjani, Navid & Shahbazmohamadi, Sina & Wang, Lei & Chandy, John & Tehranipoor, Mark. (2016). A Survey on Chip to System Reverse Engineering. ACM Journal on Emerging Technologies in Computing Systems. 13. 1-34. 10.1145/2755563.

[8] H. Wang, D. Forte, M. M. Tehranipoor and Q. Shi, "Probing Attacks on Integrated Circuits: Challenges and Research Opportunities," in IEEE Design & Test, vol. 34, no. 5, pp. 63-71, Oct. 2017, doi: 10.1109/MDAT.2017.2729398

[9] Nakamoto, S. “Bitcoin: A Peer-to-Peer Electronic Cash System”. In: Bitcoin Org, 2008; Aviable from: https://bitcoin.org/bitcoin.pdf

[10] Buterin V. “A next-generation smart contract and decentralized application platform” - white paper, 2014, pp. 1 – 36; Aviable from: https://ethereum.org/en/whitepaper/

[11] Ricou, E. “What is Ethereum (ETH)?”, 2020. Available from: https://stormgain.com/blog/what-ethereum-eth

[12] Vale, S. “Solidity: a linguagem de programação para criar os smart contracts na Ethereum”, 2020. Available from: https://www.voitto.com.br/blog/artigo/linguagem-de-programacao-solidity

[13] Trevisan, G. V, “Análise de Physical Unclonable Functions baseadas em osciladores em anel em FPGA”, UNB, 2014. Available from: https://bdm.unb.br/bitstream/10483/13749/1/2014_GabrielViniciusTrevisan.pdf

[14] Gassend, B. et al. “Silicon physical random functions.” Proceedings of the ACM Conference on Computer and Communications Security. New York, NY, USA: ACM, 2002. (CCS ’02), p. 148–160. ISBN 1-58113-612-9.

[15] Katzenbeisser, S. et al. “Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (pufs) cast in silicon.” In: Cryptographic Hardware and Embedded Systems – CHES 2012. [S.l.]: Springer, 2012. p. 283–301.

[16] M. N. Islam and S. Kundu, "Remote Configuration of Integrated Circuit Features and Firmware Management via Smart Contract," 2019 IEEE International Conference on Blockchain, Atlanta, GA, USA, 2019, pp. 325-331.

[17] Sun J, Ren L, Wang S, Yao X (2020) “A blockchain-based framework for electronic medical records sharing with fine-grained access control”. PLoS ONE 15(10): e0239946. Available from: https://doi.org/10.1371/journal.pone.0239946