2021 IEEE International Conference on Smart Internet of Things (SmartIoT) | 978-1-6654-4511-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/SmartIoT52359.2021.00043

Exploiting Silicon Fingerprint for Device Authentication Using CMOS-PUF and ECC

Carmelo Felicetti
DIMES Department
University of Calabria
Rende, Italy
carmelo.felicetti@unical.it

Marco Lanuzza
DIMES Department
University of Calabria
Rende, Italy
lanuzza@dimes.unical.it

Antonino Rullo
DIMES Department
University of Calabria
Rende, Italy
n.rullo@dimes.unical.it

Domenico Saccà
DIMES Department
University of Calabria
Rende, Italy
sacca@unical.it

Felice Crupi
DIMES Department
University of Calabria
Rende, Italy
crupi@unical.it

***Abstract*—Device authentication is an important issue in Internet of Things (IoT) for enabling the connection of ubiquitous objects/things to the Internet. One of the emerging authentication approaches is based on some device characteristic (fingerprint) such as its type, firmware version, or signature. The usage of a Physically Unclonable Function (PUF) as device "digital fingerprint" for authentication has attracted great interest, however existing solutions present security drawbacks related to the authentication protocol, or to the poor reliability of the adopted PUF technology. The authentication protocol may require challenge-response pairs to be stored in a dependable repository, with an elevated risk of information leakage.**

To overcome the above limitations, this paper presents a reliable CMOS-PUF which produces a stable output that is used as private key in an authentication protocol based on Elliptic Curve Cryptography (ECC). The overall device architecture embeds the PUF and ECC components in a memory-less framework so that the device is resilient to cyberattacks and capable to perform authentication tasks with a stable and durable identity. The main advantages of the proposed framework are that no challenge-response pairs need to be previously stored, and no error correction mechanism is needed. A prototype implementation of the CMOS-PUF is sketched and three important key points (Randomness, Circuit Reliability and Security) of the proposed device authentication scheme are discussed as well.

***Index Terms*—Physical Unclonable Function, Elliptic Curve Cryptography, ECDSA, Authentication, Internet of Things.**

I. Introduction

The Internet of Things (IoT) [1] is changing the interactions of people with things in everyday life for it enables the connection of ubiquitous objects/things to the Internet, in order to provide innovative services in emerging scenarios of business and industrial distributed applications, such as monitoring, identification, tracking, metering, and resource management, to name a few.

Various IoT applications focus on automating different tasks and are trying to empower the inanimate physical objects to act without any human intervention. All these intelligent objects connected to the Internet need to operate quickly, safely and reliably, and to act autonomously when they are involved in processes that also concern particular entities. In this context, an object is often required to proof its identity and, therefore, device authentication [2] plays a crucial role. This role is becoming even more relevant in emerging application scenarios of distributed cyber–physical systems (CPS) where blockchain technology is used to enhance CPS in various aspects, ranging from securing the data for offline storage, to protecting key operations from cyberattacks in real time [3].

Three main authentication approaches are currently used to authenticate a device: (1) *Credentials*, e.g., a secret key (*something it knows*), (2) *Integrated authentication*, e.g., by means of a secure authentication integrated circuit (*something it has*) and (3) *Device characteristics*, such as device type, firmware version, or signature (*something it is*).

The use of a Physical Unclonable Function (PUF) as device "digital fingerprint" for authentication has attracted great interest [4]. The classical authentication approach [5] is based on a challenge-response mechanism: (i) during a preliminary enrollment phase a device P registers its challenge-response pairs at a dependable repository R; (ii) when the authenticator V wants to verify P's identity, it asks R a P's challenge-response pair and submits the challenge to P; (iii) P responds with the corresponding response generated using its internal PUF; (iv) finally, V authenticates P if the received response corresponds to the one obtained from R. A severe drawback of this authentication scheme is that challenge-response pairs must be stored at a trusted third party, which represents a point of failure due the risk of information leakage. Besides, it entails a complex enrollment phase that may involve complex difficulties in application scenarios.

Recently, a more sophisticated PUF-based approach [6] has been proposed, which provides for the use of Elliptic Curve Cryptography (ECC) as the basic component of the authentication task. In this approach, the IoT device generates its private key $\texttt{PrK}$ as $\texttt{f}(\texttt{PUF}(\texttt{c}), \texttt{H})$, that is, a function of the PUF output stimulated with an in-memory challenge $\texttt{c}$, and of an in-memory string of bit $\texttt{H}$ used to reconstruct $\texttt{PUF}(\texttt{c})$ in case the PUF generates an unstable output. The public key $\texttt{PuK}$ is then generated by means of an ECC-based mechanism as a function of $\texttt{PrK}$, and the pair $\langle \texttt{PrK}, \texttt{PuK} \rangle$ is used for authentication tasks. This approach presents two main

This work has been partially funded by the PON-MIUR Projects: ARS01_00587 "SecureOpenNet: Distributed Ledgers for Secure Open Communities – SON" and ARS01_00401 "DEMETRA: DEvelopment of MatErial and TRAcking technologies for the safety of food". Carmelo Felicetti doctoral scholarship is supported by Regione Calabria, within the Project POR Calabria - FSE/FESR 2014-2020 – PhD Students and Research Grant - POR Calabria 2014-2020 - Actions 10.5.6.

978-1-6654-4511-5/21/$31.00 ©2021 IEEE
DOI 10.1109/SmartIoT52359.2021.00043

drawbacks: first, it relies on an unreliable PUF, thus it needs an error correction mechanism to make the PUF behaving deterministically; second, both H and c are stored in memory, that has the disadvantage of revealing information related to the behavior of the PUF in the case memory-leakage attacks succeed.

To overcome the above limitations, in this paper we present a novel PUF scheme, a PUF/ECC-based authentication protocol which exploit the characteristics of the proposed PUF, and a device architecture which host the two components and allows a device to perform authentication tasks with a stable and durable identity. The proposed PUF is able to reliably produce a stable output even in adverse environmental conditions, that is, that does not change due to the effect of variations in the supply voltage, operating temperature, and circuit aging.

The PUF output (*response*) is used as the device private key PrK, from which an internal circuit generates a public key PuK. The PUF input c (*challenge*) is hardwired so as to generate always the same private key, in order for a device to keep the same identity (i.e. a public-private key pair) throughout its life cycle. Further on, another internal circuit, implementing the Elliptic Curve Digital Signature Algorithm (ECDSA), signs any input message m using the private key PrK, thus returning the signed input $(\mathtt{m})_{\mathtt{PrK}}$ together with the public key PuK, so that the authenticator can eventually authenticate the device by checking the signature with its public key.

Overall, our authentication scheme gives the device itself an intrinsic, stable and durable identity, which allows to achieve fast authentication, accountability, non-repudiation, and no risk of impersonation attacks. This feature avoids any enrollment phase for a device to set up its initial identity, thus no challenge-response pairs nor further device's related data need to be stored. In addition, as the device is memory-less, any information leakage attack would be addressed only to the device hardware with the sole result of making it unusable.

Finally, we shall elaborate upon three relevant issues of our device authentication scheme: (1) *Circuit Reliability* (the PUF implemented on a specific chip instance needs to generate the same private key under different operational conditions), (2) *Randomness* (the same PUF implemented on different chips must generate different private keys in a random way to effectively ensure unpredictability and unclonability), and (3) *Security* (the device must be resistant to possible attacks and resilient to key leakage).

Our contribution can be summarized in the following three points:

- a novel PUF design which output does not change due to the effect of variations in the supply voltage, operating temperature, and circuit aging, and which exploits the intrinsic manufacturing imperfections to generate a device unique physical identity, suitably used by built-in Elliptic-Curve Cryptography mechanisms;
- an ECC-based authentication protocol which exploits the reliability of the proposed PUF to allow an IoT device to perform secure authentication tasks with a stable and durable identity, without the need of any error correction mechanism nor of any preliminary enrollment phase;
- a memory-less, authentication-oriented device architecture which hosts the above two components, and permits the implementation of a reliable authentication mechanism without incurring in memory-leakage issues.

The paper is organized as follows. In Section II, we introduce basic notions on PUF, ECC and device authentication, and discuss related work. We describe the architecture and the functionalities of our PUF/ECC authenticated device in Section III and we discuss randomness, reliability, and security issues in Section IV. Finally, we draw the conclusion and discuss further work in Section V, in particular the possible usage of our devices as smart tags in product tracking applications based on blockchain.

II. Background and Related Work

A. Physically Unclonable Functions (PUF)

The secure authentication of devices and the generation of cryptographic keys through the use of Physically Unclonable Functions (*PUF*) is an emerging technology and is proving to be of fundamental importance for many application scenarios. PUF technology natively generates a digital fingerprint for its associated security integrated circuit (IC), which can be utilized as a unique key/secret to support cryptographic algorithms and services including encryption/decryption, authentication, and digital signature. Except during the cryptographic operation, the PUF key value never exists in digital form within the IC. Further, since the key is derived and produced on-demand from physical characteristics of electronics transistors and instantaneously erased once used, it is never present in the non-volatile memory of the device. Any attempt to discover the key through micro-probing or other invasive techniques will disrupt the sensitive circuitry used to construct the key, thus making the current output useless. For these reasons, PUFs provide the desired security level for today's embedded systems.

Compared to traditional solutions that store keys in a digital memory, the PUF technology appears much more secure as the generated keys strictly depend on the physical characteristics of the circuit from which they derive. The level of security necessary for this type of applications is guaranteed by the fact that the generation of the keys is related to a complex correlation between the physical quantities involved that cannot be controlled during the production process and cannot even be external measured. This randomness originates from multiple unpredictable and uncontrollable factors: oxide variation, device-to-device mismatch in threshold voltage, interconnect impedances, and variation that exists within wafer manufacturing through imperfect or non-uniform deposition and etching steps. In general, silicon-based PUFs are circuits that implement a unique challenge-response mechanism for each chip which produces unpredictable and instantiation dependent outcomes. Due to their physical nature, PUF responses are generally not perfectly reproducible (noisy) and not perfectly random [4].

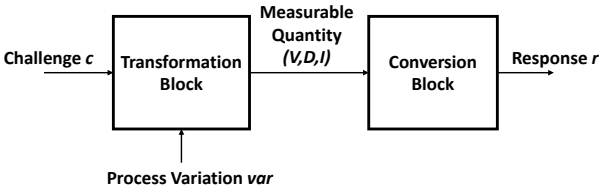

Fig. 1. Silicon based PUF Architecture.

A generic PUF is based on two essential elements: *(i)* the intrinsic presence of a physical disorder, and *(ii)* capability to measure and quantify the physical disorder itself. Relating to integrated circuits, process variations that are naturally introduced during manufacturing already match the first requirement, however, the measurement of the entropy represent an issue. Typically, a semiconductor device can be characterized by three fundamental parameters which are respectively voltage, current, and circuit response delay, whereas other measures can be derived from these mentioned above. Once the desired measure have been obtained, it is necessary to carry out a transformation of the entropy measurement into a discrete form in order to encode it with a binary digit that represent the PUF response. In order to have a random generation of bits, the intrinsic property represented by the mismatch between the same components that make up a circuit is often exploited.

Based on the above, it is possible to model a generic architecture for the realization of IC as in Figure 1 [7], where the left block turns the challenge and the process variation of the implementation technology into a measurable quantity (i.e. the voltage), and the right one turns the measure into a binary value [8]. Following the basic scheme, there are different approaches described in the literature for the realization of PUF bit-cells, some of which are based on taking account circuits delay [9], while others exploit the memory structures [10]. More robust silicon-PUF are based on static and mono-stable analog circuits that exploit the mismatch between current mirror branches and between the outputs of pairs of voltage generators. Other approaches exploit the mismatch of capacitor ratios, however, in these cases it is quite difficult to realize devices able to offer a robust response under temperature and power supply voltage fluctuations. Many applications, especially those based on digital circuits, require the presence of auxiliary systems, such as error correction mechanisms [11] in order to mitigate the effects caused by the stability problems of the outputs provided by the PUFs, which are the main cause of errors that can occur during the generation of the responses.

B. Elliptic Curve Cryptography

The Elliptic Curve Cryptography (ECC) consists of a series of public-key cryptosystems based on the structures of the elliptic curves over finite fields and on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP) [12]. ECC uses smaller keys and signatures than RSA for the same level of security and provides very fast key generation, fast key agreement, and fast signatures [13]. The key generation in the ECC cryptography is as simple as securely generating a random integer in a certain range: any number within the range $[1, \mathtt{n}-1]$ is a valid private key, where `n` is the *order* of the elliptic curve, and the public key `PuK` is a point on the elliptic curve, calculated with the EC point multiplication, that is $\mathtt{PuK} = \mathtt{PrK} \cdot \mathtt{G}$, where `G` is called the *Generator point*, used for scalar multiplication on the curve. An effective random number generator is thus at the basis of the secure functioning of any ECC-based cryptosystem, since predictable generators may be exploited by attackers to infer the private key. This is relevant, in particular, in light of the security vulnerabilities that were identified in pseudorandom generators used by many systems [14]–[16].

The Elliptic Curve Digital Signature Algorithm (ECDSA) works as follows [17]. Bob knows the public key `PuK` of Alice and wants to check her identity by asking to encrypt a message m using the private key PrK, Alice does the following:

1) compute $e = \mathrm{HASH}(m)$;
2) let z be the L_n leftmost bits of e, where L_n is the bit length of the group order n;
3) select a random integer k from $[1, n-1]$;
4) compute the curve point $(x, y) = k \times G$;
5) compute $r = x \ mod \ n$, if $r = 0$ go back to step 3;
6) compute $s = k^{-1}(z + r \cdot PrK) \ mod \ n$, if $s = 0$ go back to step 3.

Bob receives from Alice the pair (r, s) as her signature for the message m and authenticates the signature as follows:

1) Verify that r and s are integers in $[1, n-1]$, if not, the signature is invalid;
2) compute $e = \mathrm{HASH}(m)$;
3) let z be the L_n leftmost bits of e;
4) compute $a = z \cdot s^{-1} \ mod \ n$, and $b = r \cdot s^{-1} \ mod \ n$;
5) compute the curve point $(x, y) = a \times G + b \times PuK$, if $(x, y) = 0$ the signature is invalid;
6) if $r = x \ mod \ n$, the signature is valid, invalid otherwise.

A typical elliptic curve used in practice (e.g., in Bitcoin's public-key cryptography [18]) is denoted as "secp256k1" and is called a Koblitz curve [19]. Solinas [20] showed how one can compute vP very efficiently for arbitrary v where P is a point on a Koblitz curve. Since performing such scalar multiplications is the dominant computational step, Koblitz curves allow for fast computation and, therefore, are very suitable for using ECDSA on smart devices. A critical security issue on ECDSA implementation concerns the choice of the random nonce k that should not be duplicated for different signatures – also this issue arises in the bitcoin blockchain [21]. As described in Section III-C, we adopt secp256k1 elliptic curve in our implementation and use both the private key `PrK` and the input message m as source for the random generation of k.

C. Device Authentication

Device identification requires a reliable authentication mechanism. In order to prove its identity, a device (the prover) receives a value, computes the result and sends it to the verifier. This in turn, checks if the response satisfies certain requirements, and eventually the device is authenticated. Different approaches have been proposed that use a PUF, possibly coupled with a cryptographic authentication mechanism, to compute a response. The methods proposed in [5], [22] require a trusted third party (TTP) to store challenge-response pairs in a database, collected during an enrollment phase, for future authentication operations. To check the authenticity of a device, the verifier asks the TTP a challenge (that has never been used before) to submit to the device's PUF. Then the verifier submits the PUF's response to the TTP which checks if the response matches, and if so the device is authenticated. This approach has the drawback of having challenge-response pairs to be stored somewhere, which represents a means of attack since attackers can get hold of them with the aim of stealing the device identity. In [23] the author proposes a PUF-based authentication mechanism, whose security relies on the discrete logarithm problem over a 256-bit modulus, that corresponds to a 94-bit ECC modulus [24]. This method has been later improved by the same author [6] by presenting a 384-bit ECC modulus authentication mechanism (described in Section I). However, both protocols have been implemented on an FPGA development board, which, by its nature, provides an unreliable output that needs to be reconstructed using an error correction mechanism (called "helper"). Moreover, both the helper and the challenge are stored in memory, that has the disadvantage of revealing information related to the behavior of the PUF in the case memory-leakage attacks succeed, that is equivalent to introducing security holes in the secret's custody mechanism.

In [25], it has been proposed to store the challenge-response pairs in a node of a blockchain, called Accountability Node, which acts as a bridge between the identification system based on the use of the PUFs and the distributed ledger related to the blockchain.

We finally mention that Huth et al. [26] proposed to combine two sources of entropy for the generation of cryptographyc keys: the PUF and physical properties of the communications channel. Also this approach requires a server for the enrollment phase and for the storage of challenge-response pairs.

III. Architecture and Features of the PUF/ECC Device

A. Device Architecture

The architecture of the device, depicted in Figure 2, consists of three components: (1) CMOS PUF, (2) ECC and (3) I/O. Details on the first two components are described next, while no further details on the I/O component are reported as they are rather standard.

The block (1) is an ad-hoc designed hardware comprising the PUF and the PUF Trigger modules used to generate

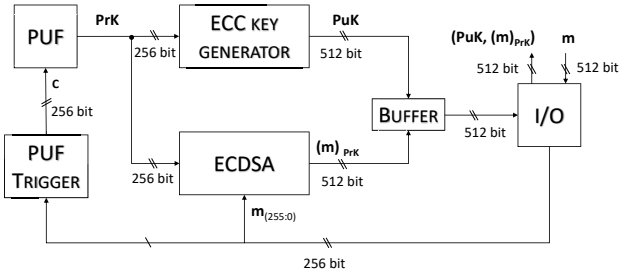

Fig. 2. Architecture of the PUF/ECC device

the private key PrK. When the device receives the message m through the I/O component, a trigger signal enables the hardwired 256-bit challenge c to be given as input to the PUF. As a result of the input c a response is pulled out of the CMOS PUF component. Notice that no challenge other than c is used in order for a device to keep the same identity (i.e. a public-private key pair) throughout its lifecycle. We point out that the key generation protocol may be prone to failure as not all possible strings of 256 bits are valid for ECC-based cryptography, as discussed in Section IV. When an out-of-range value occurs (a very low probability around 10^{-63}, indeed), the device is set to a failure state and must be discarded.

The private key PrK is given as input to the ECC (Elliptic Curve Cryptography) component together with the message m. The ECC component consists of two modules that are currently implemented by suitable firmware: the ECC public key generator and the ECDSA (Elliptic Curve Digital Signature Algorithm). The former module takes PrK as input and outputs the public key PuK – in case of out-of-range private key, the firmware operates a kind of bypass bringing its output to 0, that means a procedure failure.

The ECDSA module takes PrK and m and returns the signature $(\mathtt{m})_{\mathtt{PrK}}$, that is m signed with the private key PrK. Then $(\mathtt{m})_{\mathtt{PrK}}$ and PuK are collected into a buffer and, thereafter, the pair $<\mathtt{PuK}, (\mathtt{m})_{\mathtt{PrK}}>$ is given as output through the the I/O component.

The proposed device design does not provide for the use of any memory, thus allowing for the protection against non-volatile memory readout, invasive volatile memory probing attacks [27], and the side-channel attack [28] in which the adversary is able to learn a noisy version of the memory with the aim of inferring secret data. Rather, we hard wire the challenge c which enables the PUF to dynamically regenerate the response every time it is required. This reduces the exposure of private data, as it only exists when needed for a cryptographic operation.

B. CMOS-PUF Component

The key component of the proposed device scheme is the CMOS-PUF (or simply PUF). The PUF contains an element modeled according to the classical scheme of Figure 1, repli-

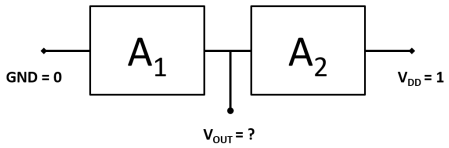

Fig. 3. The proposed PUF model Transformation Block.

cated in a matrix structure of 256 components, all identical to each other, in order to provide an output of 256 bits.

Specifically, we use CMOS technology transistors that allow to obtain the desired effects related to the amplification of process variations. Our goal is to have, for a give input, different outputs for different devices, that do not change due to the effect of variations in the supply voltage, operating temperature, and circuit aging. A device will produce an output with sufficient number of bits that are different from the output produced by a different device. Each output, however, will stay constant over the specified voltage and temperature range. To this end, we propose a bistable silicon-based PUF based on a voltage divider. More in detail, the transformation block of each PUF bit-cell is designed by coupling identical circuit elements as in Figure 3: $\mathtt{A}_1$ and $\mathtt{A}_2$ represents two identical blocks, each consisting of a pair of transistors whose respective terminals are connected according to the description below, in order to amplify the desired behaviors; $\mathtt{V}_{\mathtt{OUT}}$ is the unpredictable output that depends on the predominant branch (i.e., $\mathtt{A}_1$ or $\mathtt{A}_2$), that is the one that conduct to `GND` (logic value `0`), or the one that goes to `VDD` (logic value `1`).

The circuit design of the PUF bit-cell consists of a four transistors (4T) voltage divider operating in the deep sub-threshold region implemented by using low-voltage threshold nMOS transistors in 65-nm TSMC technology, where the body terminal of the top transistor of each block is connected to the source terminal of the bottom transistor. This body connection allows to introduce a positive feedback. This means that for the less conductive block, where the voltage drop is higher, the body effect is higher, which further reduces the block conductance. The maximum positive feedback is obtained by using a short channel bottom transistor and a long channel top transistor. In a range between 60 and 200 nm, the optimum was found by selecting a channel length of 80 nm for the bottom transistor and a channel length of 200 nm for the top transistor.

To analyze the distributions of the output voltage, we performed Monte Carlo simulation on 1000 samples, considering only transistor mismatch. During our intermediate test, executed by the Cadence simulation software, the transformation block confirmed a high level of bistability, therefore it was sufficient to insert a simple inverter (`NOT` gate) at the output of $\mathtt{V}_{\mathtt{OUT}}$ to act as a Conversion Block in order to obtain all the values close to `GND` or `VDD` as shown in Figure 4. The ideal bistability guarantees a strong robustness against interference and a small number of required conversion stages. The intrinsic advantages of this solution are a perfect

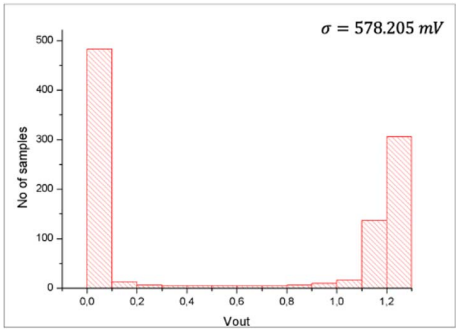

Fig. 4. Statistical distribution of the output voltage for 4T voltage divider at 1.2 volt bias voltage.

balance between the two logic states, and a strong robustness against temperature variations, voltage variations, inter-die variability, and aging. Further tests have been conducted to check robustness in different working conditions, that can be expressed as the number of bits that flip due to voltage and temperature variations. During two independent simulation rounds of 2500 samples each, performed by varying the operating temperature between 0 and 100 °C in the former, and by oscillating the supply voltage of about $\pm$ 0.2 volts around the bias voltage in the latter, only 0.2% of bits flipped due to temperature variation, whereas we did not observed bit flip events in the other case. Finally, comparing this PUF solution with our previous proposal [29] and the state-of-the-art PUF designs presented in [30], we can observe advantage in terms of variability since we have 0.2% of bit flips due to the temperature variations compared with 1.7% and 2.21% of the above works, respectively. The same applies to bit flip events due to fluctuations of the bias voltage: compared to our result (0.0%), in the two other cases we observed 0.23% and 0.66%, respectively, of bit flips under similar test conditions. In addition, due to the low number of transistors (four in total), our solution presents advantages in terms of silicon area occupation and lower power consumption, compared with the majority of proposals that can be found in literature that use more than four transistors, or less then four but with poor performances in terms of variability. As future work we point to scale technology to obtain a smaller PUF by keeping a stable behavior.

Algorithm 1

procedure AUTHENTICATE(m)
  $PrK \leftarrow PUF(c)$ ▷ generate private key
  $PuK \leftarrow G \cdot PrK$ ▷ generate public key
  $(m)_{PrK} \leftarrow$ **Sign**(m, PrK) ▷ sign input m
  return $\langle PuK, (m)_{PrK} \rangle$
end procedure

procedure SIGN(m, PrK)
  $n \leftarrow HMAC\_DRBG(m, PrK)$ ▷ generate nonce n
  $(m)_{PrK} \leftarrow ECDSA(m, PrK, n)$ ▷ sign m
  return $(m)_{PrK}$
end procedure

C. ECC Component

Our protocol uses elliptic curve cryptography which is ideal for devices with limited resources, as it provides the same level of security as other cryptographic frameworks, but keeping storage and computational requirements low. We adopt the secp256k1 elliptic curve with a field order of length 512, which has been proven to be as secure as an RSA/DSA modulus of size 3072, and recommended by several standards [13]. The ECC component includes a module for generating the public key from a given private key, and a module for signing. Currently the two modules are being implemented by two ad-hoc firmware but a possible hardwired implementation is envisioned in the future.

As depicted in Figure 2 the authentication protocol works as follows. The verifier (another device or a human by means of a smartphone) submits a message $\mathtt{m}$ of 256 bits to the device. The 256 bits are typically constructed by the verifier by means of a hash function, e.g. SHA256. The input $\mathtt{m}$ and the private key PrK (returned by PUF activation) are the input of the ECDSA module, which generates the signature on $\mathtt{m}$ as a pair of curve coordinates $(\mathtt{m})_{\mathtt{PrK}} = (\mathtt{x}, \mathtt{y})$. As shown in Section II-B, $\mathtt{x}$ and $\mathtt{y}$ are a function of $\mathtt{m}$, PrK, and a nonce $\mathtt{k} \in [1, \mathtt{n}-1]$, where $\mathtt{n}$ is the *order* of the elliptic curve. The generation of $\mathtt{k}$ is crucial for a secure functioning of ECDSA: first, it must be adopted a random number generator free from backdoors that can be exploited by attackers with the aim of inferring the private key; second, different $\mathtt{k}$ must be generated for different pairs $<\mathtt{m}, \mathtt{PrK}>$ (i.e. <input message, device>), otherwise an adversary would be able to infer the private key of the device by comparing the pairs $<\mathtt{m}, (\mathtt{m})_{\mathtt{PrK}}>$ s/he collects; finally, one must be careful not to use the same j nonces with j different keys (i.e. the same j nonces used by one device must not be used by other $j-1$ devices), as also in this case an adversary would be able to infer the private keys by simply solving a system of j linearly independent equations and j unknowns, which is uniquely solvable [31]. In the light of this considerations, we employ the implementation [32] of the pseudorandom generator algorithm HMAC_DRBG proposed in [33], which uses a seed computed from $\mathtt{m}$ and PrK for the generation of a (pseudo)random number. Thus, since $\mathtt{k}$ is deterministically generated from the data to be signed $\mathtt{m}$ (and from PrK), the concerns described above are no longer as relevant as we always produce the same signature for the same piece of data.

The ECC key generator module takes as input the private key PrK generated by the PUF, and computes a 512-bit public key $\mathtt{PuK} = \mathtt{G} \cdot \mathtt{PrK}$, where $\mathtt{G}$ is the *base point* of the elliptic curve. Finally, the pair $<\mathtt{PuK}, (\mathtt{m})_{\mathtt{PrK}}>$ is returned to the verifier, which authenticates the device by simply verifying the device signature using its public key.

The sequence of the steps performed for authentication is shown in Algorithm 1.

IV. Relevant Issues of PUF/ECC-based Device Authentication

A. Circuit Reliability

It is very important that the circuit provides stable responses when operating conditions may change. Ideally, a circuit well designed for the purpose should have an intra-PUF Hamming Distance equal to 0%, which means that by stimulating the circuit in the same way, its output must remain constant. This result is not obvious, and often the motivation is due to the fact that in order to facilitate the implementation of the system, the choice for the PUF falls on the use of models based on digital circuits, which, by their nature perform the discretization of the analog input values, thus introducing noise that makes them assume a behavior not free from imperfections. This also leads to a significant departure from the value of 0% previously indicated. In these cases, a practical and functional stratagem, but not entirely free from defects, is often used: To increase the reliability to desirable levels, an error correction mechanism-based system (typically called "helper") is introduced. However, this solution has the disadvantage of revealing information related to the behavior of the PUF, which is equivalent to introducing security holes in the secret's custody mechanism. This may represent a reason for the loss, at least partially, of the main requirements for which this kind of devices are used as they are considered safe. The approach that involves the use of analog PUFs of the type used in this work, combined with appropriate circuit design choices, makes the mechanism much more robust, and does not require the use of a support solution in order to reconstruct the circuit responses correctly. The experimental results confirm that by making these choices, the circuit proved to be robust under the effect of significant variations in terms of supply voltage and temperature. Currently, further circuits configuration are being examined, some already produced for which physical characterization activities are underway by the use of a probe station (equipped with microneedles for laboratory circuits inspection). As future work we aim to further improve the level of reliability, limiting area occupation and energy consumption of the proposed solution.

With refer to reliability, it is worth noting that the key generation protocol proposed in Section III may be prone to failure. This is due to the fact that the private key must lie in the range $[1, \mathtt{n}-1]$ to be valid for ECC-based cryptographic operations, with $\mathtt{n} \approx 1.157 \cdot 10^{77}$, which is slightly lower than $2^{256}-1$. However, the 256-bit PUF response may be either 0 or between $\mathtt{n}$ and $2^{256}-1$. Anyway, we note that this may happen with very low probability, that is of the order of 10^{-63}. In current implementation, if out-of-range values occur, the ECC key generator module forces its output to 0 as a meaning of failure.

B. Randomness

As pointed out in Section II-B, randomness is a critical aspect when performing cryptographyc operations. The same PUF implemented on different chips must generate different

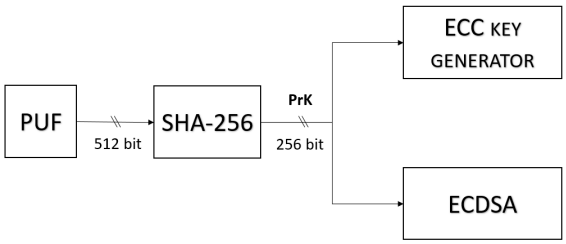

Fig. 5. Increasing randomness using hash function on 512 bits string.

private keys in a random way to effectively ensure unpredictability and unclonability. Circuit randomness property ensures the balance between 0 and 1 in the PUFs response, to make information deduction more difficult. Random bit generation is achieved through the intrinsic mismatch in circuit components. Using a bistable silicon-based PUF on a voltage divider, implemented by two identical series-connected circuits has the intrinsic advantage of perfect balance between the two logic states. Essentially, if we consider two perfectly identical circuits from the macroscopic point of view, but with intrinsically different microscopic characteristics due to process variations, and we trigger a competition between these circuits, it is evident that one prevails over the other, but this is impossible to predict (we are not able to understand in advance which of the two circuits is stronger than the other). However, at the same time the outcome of this challenge will always be reproducible over time (the circuits remain unchanged and their behavior is deterministic). This is true when the PUF designer is able to emphasize the random process fluctuations obtaining high robustness against noise, supply voltage, and temperature variations due to the effect of the high variability.

In addition to relying on the effectiveness of the PUF operation for randomness, we will use a 512-bit PUF in place of a smaller one, since longer strings are able to capture a larger number of features that distinguish a device instance from another. The subsequent reduction function to 256 bits is made by a hash function (see Figure 5), that is a deterministic, one-way (i.e., practically infeasible to invert) function which maps data of arbitrary size to fixed-size values. Hash functions are characterized by their random behavior, that is, inputs that only differ very slightly are mapped to hash values very distant with each other. In particular we use the `SHA256` hash function since its output size is functional to be used as a private key in an ECC-based cryptographic protocol with the sepc256k1 curve.

C. Security Issues

Machine learning attacks [34] digest a number of known challenge-response pairs to train a mathematical model and could be utilized to simulate the behavior of a PUF, thus predicting the responses for unknown challenges. As evidenced in Section I, as no challenge-response pairs (CRP) are stored, these attacks cannot be performed in our setting since the PUF input is hardwired and never-changing, thus the PUF cannot be challenged with other inputs, making it impossible for adversaries to collect CRPs for training purposes. It turns out then, that key leakage, including the attacks to device's memory exploiting temporary persistence when an ordinary DRAM loses power [28], is not actually a security issue in our authentication scheme.

A critical security issue in our scheme (that also arises in blockchain protocols) is the generation of a unique 256-bit private key, that must be performed using a secure source of randomness. This issue is strongly related to the randomness that has have been dealt with in the previous section. Also the well-known critical security issue concerning the choice of a random nonce in ECDSA implementation has been addressed by using both the private key `PrK` and the input m as source for the random generation of the nonce.

Typical attacks to hardware devices are the so called side-channel attacks [35], [36] that are based on the analysis of information relating to physical channels such as energy consumption, information timing, electromagnetic and acoustic emissions, etc. Since a tampering attempt to the PUF modifies its behavior, physical inspection of this component cannot be exploited (intrinsic anti-tamper system). Moreover, considering we have an architecture free of information leakage, we can consider our proposal resilient to best known side-channel attacks. Our devices directly communicate with each other only when they are in proximity, thus attacks typically carried in an Internet environment like man-in-the-middle (i.e. relaying or altering the communications between devices) and eavesdropping cannot be accomplished. Moreover, CRPs are not required to be enrolled at trusted third parties, which leaves no room for impersonation attacks. The verifier can prevent replay attacks by simply including a timestamp in the message `m` it sends to the device during the authentication protocol.

V. Conclusions

In this paper we have presented an authentication-oriented device architecture, based on a CMOS-PUF scheme able to reliably produce a stable output even in adverse environmental conditions. We have then employed the CMOS-PUF as the key component of an authentication scheme that uses ECC for the generation of asymmetric cryptographyc keys and signatures.

We have also discussed three important issues that determine the effectiveness of the proposed architecture: (1) *Randomness*, the same PUF implemented on different chips must generate different private keys in a random way to effectively ensure unpredictability and unclonability. With this in mind, we have increased the PUF output to 512 bits, as longer strings are able to capture a larger number of features that distinguish a device instance from another. (2) *Circuit Reliability*, the PUF implemented on a specific chip instance must generate always the same private key even under adverse operational conditions. To this end we have designed a PUF based on the naturally occurring random variation and mismatch of the analog characteristics of MOSFET semiconductor devices. (3) *Security*, the device must be resistant to possible attacks and key leakage. We have shown that the design of both PUF and

authentication protocol does not leave room for many known attacks, like key leakage, impersonation, side-channel, man-in-the-middle, etc.

As future works we plan to increase the private key randomness by implementing a PUF with a wider number of outputs, so as to improve the overall PUF/ECC device reliability. A further contribution will be given on using PUF/ECC authenticated devices as smart tags to verify the authenticity of goods in a product tracking scenario. The public keys of the used tags are registered on a blockchain (or some trusted authority) by the producer so that the consumer can later check the product authenticity by inquiring the blockchain about the tag, that will be eventually removed after a successful check. In this case, the I/O component of the device will be implemented by the NFC technology to enable the interaction for both the enrollment and authentication phases.

REFERENCES

[1] P. V. Paul and R. Saraswathi, "The Internet of Things — A comprehensive survey," in *2017 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC)*, 2017, pp. 421–426.

[2] M. Trnka, T. Cerny, and N. Stickney, "Survey of Authentication and Authorization for the Internet of Things," *Security and Communication Networks*, vol. 2018, pp. 1–17, 06 2018.

[3] W. Zhao, C. Jiang, H. Gao, S. Yang, and X. Luo, "Blockchain-Enabled Cyber–Physical Systems: A Review," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4023–4034, 2021.

[4] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *2007 44th ACM/IEEE Design Automation Conference*. IEEE, 2007, pp. 9–14.

[5] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–25, 2017.

[6] J. R. Wallrabenstein, "Practical and secure IoT device authentication using physical unclonable functions," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2016, pp. 99–106.

[7] B. Halak, *Physically Unclonable Functions: Design Principles and Evaluation Metrics*. Cham: Springer International Publishing, 2018, pp. 17–52.

[8] Y. Su, J. Holleman, and B. P. Otis, "A digital 1.6 pj/bit chip identification circuit using process variations," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 69–77, 2008.

[9] S. S. Mansouri and E. Dubrova, "Ring oscillator physical unclonable function with multi level supply voltages," in *2012 IEEE 30th International Conference on Computer Design (ICCD)*, 2012, pp. 520–521.

[10] A. R. Krishna, S. Narasimhan, X. Wang, and S. Bhunia, "Mecca: A robust low-overhead puf using embedded memory array," in *Cryptographic Hardware and Embedded Systems – CHES 2011*, B. Preneel and T. Takagi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 407–420.

[11] M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 48–65, 2010.

[12] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[13] D. R. Brown, "Sec 2: Recommended elliptic curve domain parameters," *Standars for Efficient Cryptography*, 2010.

[14] L. Dorrendorf, Z. Gutterman, and B. Pinkas, "Cryptanalysis of the windows random number generator," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 476–485.

[15] Z. Gutterman, B. Pinkas, and T. Reinman, "Analysis of the linux random number generator," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE, 2006, pp. 15–pp.

[16] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage, "When private keys are public: Results from the 2008 debian openssl vulnerability," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, 2009, pp. 15–27.

[17] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.

[18] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, 2nd ed. O'Reilly Media, Inc., 2017.

[19] N. Koblitz, "CM-curves with good cryptographic properties," in *Annual international cryptology conference*. Springer, 1991, pp. 279–287.

[20] J. A. Solinas, "An improved algorithm for arithmetic on a family of elliptic curves," in *Annual International Cryptology Conference*. Springer, 1997, pp. 357–371.

[21] M. Brengel and C. Rossow, "Identifying key leakage of bitcoin users," in *Research in Attacks, Intrusions, and Defenses*, M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis, Eds. Cham: Springer International Publishing, 2018, pp. 623–643.

[22] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based" unclonable" RFID ICs for anti-counterfeiting and security applications," in *2008 IEEE international conference on RFID*. IEEE, 2008, pp. 58–64.

[23] J. R. Wallrabenstein, "Implementing authentication systems based on physical unclonable functions," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, pp. 790–796.

[24] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," *Journal of cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[25] C. Felicetti, A. Furfaro, D. Saccà, M. Vatalaro, M. Lanuzza, and F. Crupi, "Making IoT Services Accountable: A Solution Based on Blockchain and Physically Unclonable Functions," in *International Conference on Internet and Distributed Computing Systems*. Springer, 2019, pp. 294–305.

[26] C. Huth, J. Zibuschka, P. Duplys, and T. Güneysu, "Securing systems on the Internet of Things via physical properties of devices and communications," in *2015 Annual IEEE Systems Conference (SysCon) Proceedings*. IEEE, 2015, pp. 8–13.

[27] S. P. Skorobogatov, "Semi-invasive attacks – A new approach to hardware security analysis," 2005.

[28] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest we remember: cold-boot attacks on encryption keys," *Communications of the ACM*, vol. 52, no. 5, pp. 91–98, 2009.

[29] R. De Rose, F. Crupi, M. Lanuzza, and D. Albano, "A physical unclonable function based on a 2-transistor subthreshold voltage divider," *International Journal of Circuit Theory and Applications*, vol. 45, no. 2, pp. 260–273, 2017.

[30] A. Alvarez, W. Zhao, and M. Alioto, "Static physically unclonable functions for secure chip identification with 1.9-5.8% native bit instability at 0.6-1 v and 15 fj/bit in 65 nm," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 3, pp. 763–775, Mar. 2016.

[31] M. Brengel and C. Rossow, "Identifying key leakage of bitcoin users," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 623–643.

[32] T. Pornin, "Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA)," *Internet Engineering Task Force RFC*, vol. 6979, pp. 1–79, 2013.

[33] E. B. Barker and J. M. Kelsey, *Recommendation for random number generation using deterministic random bit generators (revised)*. US Department of Commerce, Technology Administration, National Institute of …, 2007.

[34] Rührmair, Ulrich and Sehnke, Frank and Sölter, Jan and Dror, Gideon and Devadas, Srinivas and Schmidhuber, Jürgen, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM conference on Computer and communications security*, 2010, pp. 237–249.

[35] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[36] F. Koeune and F.-X. Standaert, *A Tutorial on Physical Security and Side-Channel Attacks*. Berlin, Heidelberg: Springer-Verlag, 2005, p. 78–108.