Theoretical framework for physical unclonable functions, including quantum readout

Giulio Gianfelici ⬤,* Hermann Kampermann, and Dagmar Bruß ⬤

Institut für Theoretische Physik III, Heinrich-Heine-Universität Düsseldorf, D-40225 Düsseldorf, Germany

We propose a theoretical framework to quantitatively describe physical unclonable functions (PUFs), including extensions to quantum protocols, so-called quantum readout PUFs (QR-PUFs). (QR-) PUFs are physical systems with challenge-response behavior intended to be hard to clone or simulate. Their use has been proposed in several cryptographic protocols, with particular emphasis on authentication. Here, we provide theoretical assumptions and definitions behind the intuitive ideas of (QR-) PUFs. This allows us to quantitatively characterize the security of such devices in cryptographic protocols. First, by generalizing previous ideas, we design a general authentication scheme which is applicable to different physical implementations of both classical PUFs and (QR-) PUFs. Then, we define the *robustness* and the *unclonability*, which allows us to derive security thresholds for (QR-) PUF authentication and paves the way to develop further new authentication protocols.

I. INTRODUCTION

Authentication is a major task of both classical and quantum cryptography. To achieve secure communication between two parties Alice and Bob, it is necessary to ensure that no intruder may participate in the communication, pretending to be one of the legitimate parties, e.g., by a so-called *man-in-the-middle attack* [1]. Authentication is ultimately classical, even in quantum protocols like quantum key distribution (QKD) [2].

The main ingredient of an authentication protocol is a shared secret between the legitimate parties: during any authenticated communication Alice and Bob must prove the possession of this secret to confirm their identity. One has to distinguish two types of authentication [1]. *Message authentication* is the assurance that a given entity was the original source of the received data. This type of authentication can be achieved by unconditionally secure protocols [3]. *Entity authentication* is the assurance that a given entity can prove its identity and its involvement in the communication session to another entity.

Entity authentication is particularly important if there is an asymmetry between the parties, e.g., when one party, namely, Alice, is a trusted institution and the other one, namely, Bob, is an untrusted user. The communication between Alice and Bob may happen on an authenticated channel owned by Alice, where Bob interacts through a remote terminal. In that case, a one-way entity authentication protocol will be used by Alice to authenticate Bob and to allow him to use her channel. Such protocols are usually based on a *challenge-response authentication*, a type of authentication where Alice presents a *challenge* and Bob provides a valid *response*, based on the common secret, to be authenticated. For instance, Alice can ask for a password (challenge) and Bob will provide the correct one (response).

In the case of asymmetric communication, it is useful to design authentication protocols based on something the parties possess. The trusted Alice can still be required to have secret knowledge since she is able to conceal information from an adversary, but Bob is required only to protect a given token from theft. A crucial condition of this approach is that the object has to be unique and an adversary, namely, Eve, should not be able to copy it easily.

A *physical unclonable function* (PUF) [4] is a physical system which can interact in a very complex way with an external signal (which can serve as a challenge) to give an unpredictable output (which can serve as a response). Its internal disorder is exploited to make it unique, hard to clone, or simulate. PUFs are particularly suited for entity authentication because their internal structure plays the role of the shared secret. They can also be used in other protocols, like oblivious transfer [5], bit commitment [6], or classical key distribution [7]. There is a large variety of PUFs, such as the *optical PUF* [8], the *arbiter PUF* [9], the *SRAM PUF* [10], the *coating PUF* [11], the *magnetic PUF* [12], the *ring oscillator PUF* [13], and so on. A more detailed description of the whole family of PUFs is given in [14] and in [15].

To ensure reliability and security it is required to postprocess the PUFs' outputs [16,17]. The most common way to do it is by using the so-called *fuzzy extractor* [18], a tool which combines error correction and privacy amplification. Error correction is necessary because the PUF's output can be different each time the PUF interacts with the same challenge, even when the authentication involves the real Bob with the original PUF. This can be due to an erroneous implementation of the challenge or to noise in the physical process. Privacy amplification is important since the outcomes of a PUF are generally nonuniform, i.e., there exist correlations between different responses that can be used by an adversary to undermine the PUF's security. Furthermore, the response, once it is mapped into a uniform key, can, in principle, be used in different protocols other than entity authentication.

*giulio.gianfelici@uni-duesseldorf.de

However, even when dealing with noise and nonuniformity, there are some issues with PUFs because it has been shown that many of them can be actually cloned or simulated [19–21], compromising their use in secure authentication schemes.

To solve these problems, an extension of PUFs to quantum protocols was suggested, the so-called *quantum readout PUFs* (QR-PUFs) [22]. Such PUFs encode challenges and responses in quantum states, and thus they are expected to be more secure and reliable than classical PUFs, as they add a layer of complexity given by the unclonability of the involved quantum states [23]. Moreover, if such quantum states are nonorthogonal, an adversary cannot perfectly distinguish them, and an attempt to do it would introduce disturbances, thus exposing the presence of an intruder to the legitimate parties.

It is desirable to establish a theoretical framework in which one can perform a rigorous, quantitative analysis of the security properties of (QR-) PUFs. Several efforts have been made to formalize the intuitive ideas of PUF [24–28], and they all capture some aspects of them, but a well-defined agreement about theoretical assumptions and definitions is still lacking. Moreover, the previous approaches are devoted to classical PUFs only.

In this article we propose a common theoretical framework by quantitatively characterizing the (QR-) PUF properties, particularly the *robustness* [25] against noise and the *unclonability*. This is done by generalizing ideas from previous approaches (in particular from [25]) to encompass both classical and QR-PUFs. Moreover, we introduce a generic scheme for authentication protocols with (QR-) PUFs, for which security thresholds can be calculated once an experimental implementation is specified. This scheme provides an abstract formalization of existing protocols, together with ideas such as the difference between a *physical layer* and a *mathematical layer* (see Sec. II) or the concept of the *shifter* (see Secs. IV A and V A). This framework is designed to be independent of the specific experimental implementation such that a comparison of different types of PUFs and QR-PUFs becomes possible. In particular, all implementations use a fuzzy extractor for postprocessing. We expect that this analysis supports both theoretical and experimental research on (QR-) PUFs by promoting the implementation of such devices in existing and new secure authentication schemes.

The paper is organized as follows. In Sec. II we give an introduction on entity authentication protocols with (QR-) PUFs. Section III contains the notation we will use in the paper, in Sec. IV we describe a protocol with a generic classical PUF, and in Sec. V we generalize this to a generic QR-PUF. The shared formalization of the theoretical properties of (QR-) PUFs is stated in Sec. VI and the formalism is applied in some examples in Sec. VII. Some final remarks and the outlook of the work are given in the Conclusion.

II. AUTHENTICATION PROTOCOLS

In the following, we will always call Alice the party that has to authenticate Bob. Mutual authentication can be achieved by repeating the protocol swapping the roles of Alice and Bob. Moreover, we stated in the Introduction that the raw
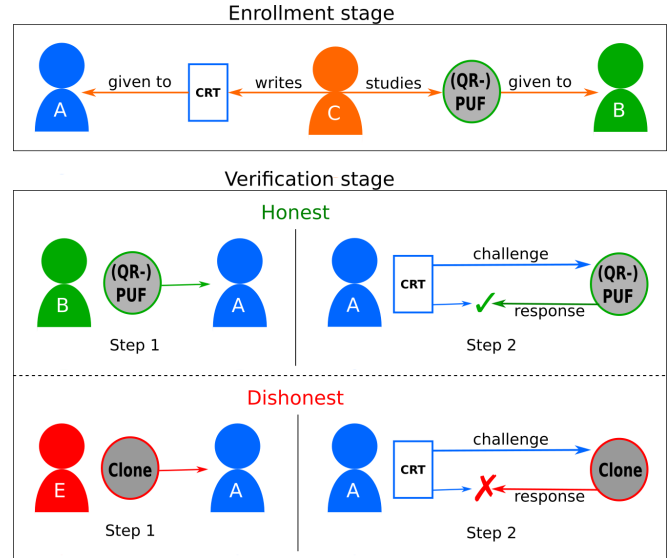

FIG. 1. A schematic description of the authentication scheme. Top: Enrollment stage. The certifier (C, orange) studies the (QR-) PUF's properties and generates the challenge-response table (CRT). Then the CRT is given to Alice (A, blue) and the (QR-) PUF is given to Bob (B, green). Bottom: Verification stage. In the honest case, Bob lets Alice interact with his (QR-) PUF through a terminal and she remotely verifies his identity with the CRT, thus authenticating him. In the dishonest case, an adversary Eve (E, red) claims to be Bob, letting Alice interact with a clone of the (QR-) PUF, and the protocol should lead to an abortion.

output of a (QR-) PUF has to be postprocessed to be used in secure cryptographic protocols. Therefore, for the sake of clarity, we call *outcome* the raw output while we mean with *response* only the postprocessed uniform key.

Entity authentication protocols with (QR-) PUFs consist of two phases [29], the *enrollment stage* and the *verification stage* (see Fig. 1).

The enrollment stage is a part of the protocol which happens only once at the beginning, after the manufacture of the (QR-) PUF and before any communications between Alice and Bob. An entity or group of entities called the *(QR-) PUF certifier* [which may be the (QR-) PUF manufacturer, Alice itself, a third trusted party, or a combination of all of them] studies the (QR-) PUF's properties and evaluates the parameters needed for the implementation and postprocessing. In particular, the certifier selects a certain number N of challenges and records the corresponding responses. Challenges and responses form the so-called challenge-response pairs (CRPs) and they are stored as a challenge-response table (CRT), together with additional information needed in the remaining part of the protocol. After the end of this stage, the certifier gives the CRT to Alice (which then *knows* the secret) and the (QR-) PUF to Bob (which then *has* the secret).

The verification stage is the part of the protocol where communication between Alice and Bob is necessary. In this stage, Bob declares his identity to Alice with his (QR-) PUF, remotely interacting with her through her terminal. To authenticate Bob, Alice sends randomly one challenge from the CRT to the (QR-) PUF and collects the outcome, which is
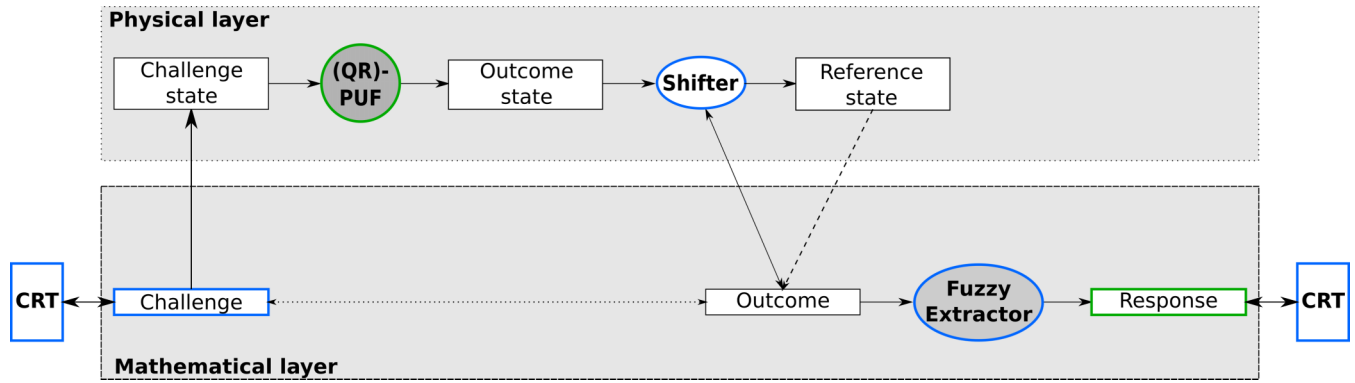
FIG. 2. A scheme of the two layers, the mathematical one (where the cryptographic protocol takes place) and the physical one (where the (QR-) PUF acts). In the physical layer a challenge state is prepared according to the information of the challenge (mathematical layer) and then the (QR-) PUF transforms it into an outcome state. The state-dependent shifter (see Secs. IV A and V A) maps the outcome state to a reference state. The outcome in the mathematical layer contains information about the implementation of the shifter and the error in the reference state and is postprocessed by the fuzzy extractor to give the response. Challenges and responses are stored into (enrollment stage) or taken from (verification stage) the challenge-response table (CRT). See Secs. IV and V for a more detailed description.

then postprocessed. The calculated response is compared with the one in the CRT, i.e., the one obtained in the enrollment stage. If they match, Alice authenticates Bob. This stage can be repeated every time Alice needs to authenticate Bob. After every round, however, the used challenge-response pair has to be eliminated from the CRT and cannot be used again.[1]

Depending on the different types of (QR-) PUFs, the challenges could be different types of physical quantities. For instance, optical PUFs are transparent materials filled with light scattering particles: a laser that interacts with one of them is turned into a unique speckle pattern. For a classical optical PUF, the challenge is the laser orientation and the outcome is the intensity of some points in the speckle pattern [8]. For a QR-PUF, the challenges and the outcomes are quantum states [22]. In both cases, however, challenges, outcomes, and responses are stored in the CRT as digital binary strings, and the responses are used as authentication keys.

There are two different layers involved in this protocol: a physical one, where the actual (QR-) PUF acts as a physical evolution from input systems to output systems, and a mathematical one, where a binary challenge string (which should represent the information on how to implement the input system) is mapped into an outcome string which is postprocessed into a response string.

To deal with the two different layers, we denote as *challenges* (*outcomes*, *responses*) the strings in the mathematical layer and as *challenge states*[2] (*outcome states*, *response states*) the implementations in the physical layer.

This configuration is schematized in Fig. 2.

---

[1]It was argued [22] that in the QR-PUF case, challenge-response pairs could be used again because an adversary is not able to gain full information about their state. Such claims need to be quantitatively proven; here we continue as if any reused CRP is insecure.

[2]This term clearly comes from quantum physics, where it is used to describe a vector in a Hilbert space. We will use the term *classical state* in this article, meaning a classical physical quantity, either scalar or vectorial.

III. NOTATION

In the article we will use the following conventions:

(i) Digital strings, like the challenges and the responses, are denoted by lowercase bold letters, for instance, $\mathbf{x_i}$ and $\mathbf{r_j}$ for the ith challenge and the jth response, respectively;

(ii) Sets of digital strings are denoted by the calligraphic uppercase letters, e.g., $\mathcal{X}$ and $\mathcal{R}$ for the set of challenges and responses, respectively;

(iii) Random variables which take values from given sets are denoted by uppercase italic letters, e.g., X and R for challenges and responses, respectively;

(iv) The physical classical states are denoted by the vector symbol (right arrow), for instance, $\vec{x}_i$ and $\vec{r}_j$ for the ith challenge state and the jth response state, respectively;

(v) The physical quantum states are denoted by the usual ket notation, for instance, $|x_i\rangle$ and $|r_j\rangle$, for the ith challenge state and the jth response state, respectively;

(vi) Maps are denoted by uppercase letters with a circumflex accent, e.g., $\hat{P}$ or $\hat{\Pi}$. In particular, the Latin letters are used for maps between strings and the Greek ones for maps between states.

IV. CLASSICAL PUF

The realization of a challenge state may involve several different steps, each of them with different experimental complexity. Each step involves devices with a limited, even though possibly large, number of different configurations, and such configurations can be used to parametrize the experimental system, resulting in our ability to formalize the challenges through discrete variables. A challenge is therefore defined as the binary string $\mathbf{x_i}$ of length n representing the configuration which realizes a given challenge state $\vec{x}_i$.

A. Enrollment

At the start of the enrollment stage, the PUF certifier selects $N \leqslant 2^n$ different challenges $\mathbf{x_i} \in \mathcal{X} \subseteq \{0, 1\}^n$, where $\mathcal{X} \subseteq \{0, 1\}^n$ is the set of all chosen challenges and $|\mathcal{X}| = N$. In fact, if a challenge consists of n bits, the total possible

number of challenges is 2^n. However, in practice, certain challenges could represent states which are impossible or hard to implement or they do not lead to a set of distinguishable responses.

For security purposes, the set of challenges $\mathcal{X}$ has to be uniform, i.e., $\hat{S}(X) = |\mathcal{X}|$, where X is the random variable defined on the set $\mathcal{X}$ and $\hat{S}(X)$ is the Shannon entropy of X. An adversary should not be able to characterize the set of challenges by studying some of them. The certifier is free to discard some challenges from $\mathcal{X}$ if he finds correlations in them. This affects the number N of challenges and has to be quantified for given experimental implementations.

Each $\mathbf{x_i} \in \mathcal{X}$ represents a challenge state $\vec{x}_i$ which can be experimentally realized and sent to the PUF, which acts as a deterministic function $\hat{\Pi}$. Due to its complex structure, any attempt to give a full description of it should be unfeasible, even for the certifier itself. For a given challenge state $\vec{x}_i$, $\hat{\Pi}(\vec{x}_i) = \vec{y}_i$, where $\vec{y}_i$ is denoted as an outcome state.

The certifier needs to map the outcome state into an outcome string, taking into account both the distribution of the outcome states and any error which may have occurred due to noise or wrong implementation of the experimental system. To do this, we introduce the notion of a shifter.

For each outcome state $\vec{y}_i$, let $\hat{\Omega}_i$ be a state-dependent operation which maps $\vec{y}_i$ into a reference state, denoted by $\vec{0}$, equal for all outcome states. For N outcome states $\vec{y}_i$ we obtain a set of N shifters $\hat{\Omega}_i$. The importance of using the shifters will be more clear when we discuss QR-PUFs. The shifters simplify the error verification process, as each expected outcome is identical.

Some devices ascribable to shifters have been used in some PUF implementations: consider, for instance, the optical PUF [8], where a laser beam (challenge state) is transformed into a complex speckle pattern (outcome state). In this scenario, it has been proposed [30] to use spatial light modulators to transform every speckle pattern into a plane wave, which then is focused into a single point (the reference state). Only if the pattern is the expected one does this happen; otherwise, the outcome state is mapped into another speckle pattern. Shifters can be designed also for other PUFs, depending on which physical quantities are implied in the outcome states. If the outcome state is already a binary value (like in the SRAM PUF [10]) the reference state can be the bit 0 and the shifters can be realized by a gate implementing either the identity or a bit-flip operation, depending on the expected outcome state. Whenever an outcome is determined by the frequency of a signal (like in a ring oscillator PUF [13]), the shifters can be passband filters.

The certifier can implement the corresponding shifter for every outcome state, since he can characterize $\hat{\Pi}(\vec{x}_i)$, possibly repeating the PUF evaluation for the same challenge state $\vec{x}_i$, to find a $\hat{\Omega}_i$ such that $\hat{\Omega}_i[\hat{\Pi}(\vec{x}_i)] = \vec{0}$.

We define $\vec{o}_i := \hat{\Omega}_i[\hat{\Pi}(\vec{x}_i)]$. While in the enrollment stage, or in a noiseless verification stage, $\vec{o}_i = \vec{0}$ by definition, in reality $\vec{o}_i$ will contain errors. This error is mapped into the Hamming weight, i.e., the number of bits that are different from 0, of a classical string $\mathbf{o_i}$, i.e., $\mathbf{o_i} = \mathbf{0}_{l_o} = 00\ldots0$ if and only if $\vec{o}_i = \vec{0}$. The string has a length l_o, dependent on the experimental implementation of the shifter. In the aforementioned example of an optical PUF, the plane wave

is focused onto an analyzer plane with a pinhole. If $\vec{o}_i = \vec{0}$ the light passes through this pinhole and a detector will click. Therefore the intensity of the light on the analyzer plane outside the pinhole can be used to find $\mathbf{o_i}$, and the resolution of the analyzer plane determines the length l_o.

The shifters convey information about the distribution of the outcome states (as they are designed on them) and therefore indirectly about the PUF. We can represent this information in terms of binary strings in the mathematical layer, just as we did for challenge states. The shifters are implemented by an experimental device (or a collection of them) with a limited number of configurations, each one of them implementing a different $\hat{\Omega}_i$. Parametrizing such configurations, we map each shifter $\hat{\Omega}_i$ in a string $\mathbf{w_i} \in \mathcal{W} \subseteq \{0, 1\}^{l_w}$. This string is exact, because it represents only the correct implementation of the shifter, without taking into account any noise. The length l_w depends on the entropy of the shifters and, consequently, on the outcome states (for some implementations, methods to analyze such an entropy have been derived [31,32]). The entropy of $\mathcal{W}$ has to be studied also to verify the presence of nonuniformity, i.e., correlations between different outcomes or between challenges and corresponding outcomes. This entropy affects the unclonability of the PUF (see Sec. VI).

The two strings $\mathbf{o_i}$ and $\mathbf{w_i}$ convey two different aspects of the outcome state. In fact, $\mathbf{o_i}$ gives information about the error only, without distinguishing different outcomes. Instead, $\mathbf{w_i}$ gives information about the distribution of the outcome states but not about errors (even a single bit-flip of $\mathbf{w_i}$ changes it into $\mathbf{w_{j\neq i}}$).

We combine $\mathbf{o_i}$ and $\mathbf{w_i}$ by defining as *outcome* a string $\mathbf{y_i}$ of length $l = l_w + l_o$, such that

$$\mathbf{y_i} = \mathbf{w_i} \parallel \mathbf{o_i}, \tag{1}$$

where $\parallel$ is the concatenation of strings. We designate $\mathcal{Y} \subseteq \{0, 1\}^l$ as the set of all outcomes, including all possible noisy versions. Explicitly,

$$\mathcal{Y} = \{\mathbf{y_i} = \mathbf{w_i} \parallel \mathbf{o_i}, \mathbf{w_i} \in \mathcal{W}, \mathbf{o_i} \in \{0, 1\}^{l_o}\}, \tag{2}$$

and $|\mathcal{Y}| = 2^{l_o} N$ (see Fig. 3 for a graphic representation of the set $\mathcal{Y}$). Moreover, we define a function $\hat{P}: \mathcal{X} \to \mathcal{Y}$, associating each challenge with the corresponding outcome, i.e., $\hat{P}(\mathbf{x_i}) = \mathbf{y_i}$.

The outcome string, being noisy and not uniformly distributed, cannot be used directly as a response. The most common way to postprocess it is through a *fuzzy extractor* [18], which is a combined error correction and privacy amplification scheme.

Definition IV.1. Let $\{0, 1\}^\star$ be the *star closure* of $\{0, 1\}$, i.e., the set of strings of arbitrary length:

$$\{0, 1\}^\star = \bigcup_{i \geqslant 0} \{0, 1\}^i, \tag{3}$$

where $\{0, 1\}^0 = \emptyset$ is the empty set. Let $\hat{H}(\mathbf{y_i}, \mathbf{y'_i})$ be the Hamming distance between $\mathbf{y_i}$ and $\mathbf{y'_i}$, i.e., the Hamming weight of $\mathbf{y_i} + \mathbf{y'_i}$, and $s := -\log_2(\max_k p_k)$ be the min-entropy of a probability distribution $p = \{p_k\}$. Furthermore, given two probability distributions p_A, p_B, associated to discrete random variables A, B with the same domain $\mathcal{C}$, let $\hat{D}_S(p_A, p_B)$ be the
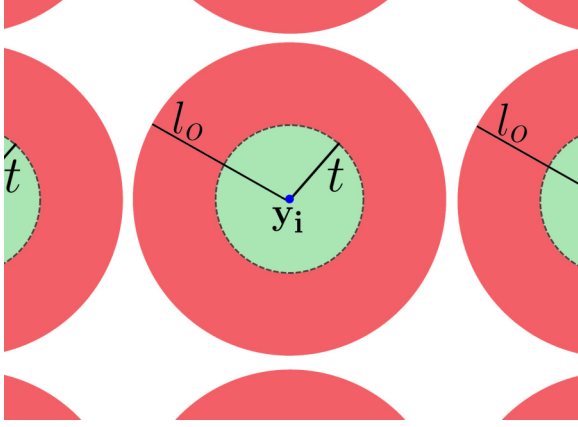
FIG. 3. Graphic representation of the set $\mathcal{Y}$, according to Eq. (2). The centers of the circles represent the noiseless outcomes $\mathbf{y_i} = \mathbf{w_i} \| \mathbf{0}_{l_o}$ for different $\mathbf{w_i} \in \mathcal{W}$, while every point in the corresponding outer circles, of radius l_o, represents a noisy version of them. Between different outcomes, including the noisy versions, there is no overlap, because $\mathbf{w_i} \neq \mathbf{w_j}$ for $i \neq j$. A fuzzy extractor can correct $t < l_o$ bit errors, i.e., the outcomes inside the inner circles.

statistical distance between p_A and p_B, i.e.,

$$\hat{D}_S(p_A, p_B) := \frac{1}{2} \sum_{c \in \mathcal{C}} |P(A = c) - P(B = c)|. \tag{4}$$

A $(\mathcal{Y}, s, m, t, \epsilon)$ fuzzy extractor is a pair of random functions, the generation function $\hat{G}$, and the reproduction function $\hat{R}$, with the following properties:

(i) $\hat{G} : \mathcal{Y} \to \{0, 1\}^m \times \{0, 1\}^\star$ on input $\mathbf{y_i} \in \mathcal{Y}$ outputs an extracted string $\mathbf{r_i} \in \mathcal{R} \subseteq \{0, 1\}^m$ and a *helper data* $\mathbf{h_i} \in \mathcal{H} \subseteq \{0, 1\}^\star$. While $\mathbf{r_i}$ has to be kept secret, $\mathbf{h_i}$ can be made public (it can even be physically attached to the PUF);

(ii) $\hat{R} : \mathcal{Y} \times \mathcal{H} \to \{0, 1\}^m$ takes an element $\mathbf{y_i'} \in \mathcal{Y}$ and a helper string $\mathbf{h_i} \in \mathcal{H}$ as inputs. The *correctness property* of a fuzzy extractor guarantees that if $\hat{H}(\mathbf{y_i}, \mathbf{y_i'}) \leqslant t$ and $(\mathbf{r_i}, \mathbf{h_i}) = \hat{G}(\mathbf{y_i})$, then $\hat{R}(\mathbf{y_i'}) = \mathbf{r_i}$;

(iii) The *security property* guarantees that for any probability distribution on $\mathcal{Y}$ of min-entropy s, the string $\mathbf{r_i}$ is nearly uniform even for those who observe $\mathbf{h_i}$: i.e., if $(\mathbf{r_i}, \mathbf{h_i}) = \hat{G}(\mathbf{y_i})$, then

$$\hat{D}_S(p_{RH}, p_{UH}) \leqslant \epsilon, \tag{5}$$

where p_{RH} (p_{UH}) is a joint probability distribution for $\mathbf{r_i} \in \mathcal{R}$ (for a uniformly distributed variable on m-bit binary strings) and $\mathbf{h_i} \in \mathcal{H}$.

The generation function of a fuzzy extractor is used, in the enrollment stage, to transform the outcome $\mathbf{y_j}$ into the uniformly distributed $\mathbf{r_i}$, that is the final response. We will see later that, in the verification stage, the reproduction function is used on a noisy version of the outcome to generate the same response.

The certifier selects a fuzzy extractor by knowing $\mathcal{Y}$ and its min-entropy s, and choosing t such that the fuzzy extractor uniquely maps a given outcome into a response, without collisions: due to noise or an erroneous experimental setup, a challenge state $\vec{x}_i$ can be implemented as a state which is closer to $\vec{x}_j$ for $i \neq j$. The error $\mathbf{o_i}^{(j)}$ associated to $\hat{\Omega}_i[\hat{\Pi}(\vec{x}_j)]$
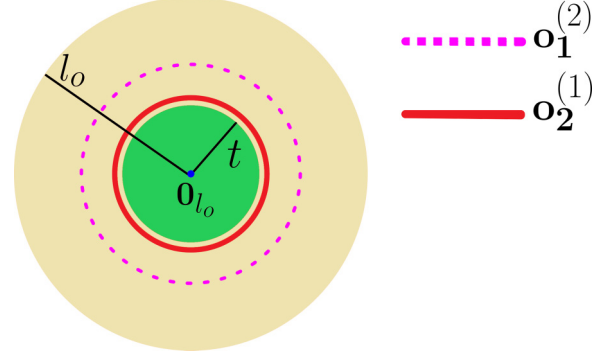

FIG. 4. Graphic representation of the choice of t for $N = 2$ challenge-response pairs. The circle represents both $\mathbf{o_1}$ and $\mathbf{o_2}$, independently from $\mathbf{w_1}$ and $\mathbf{w_2}$. The center of the circle represent the noiseless cases $\mathbf{o_1} = \mathbf{o_2} = \mathbf{0}_{l_o}$, and all the noisy cases lie in a circle of radius l_o. The errors $\mathbf{o_1}^{(2)}$ and $\mathbf{o_2}^{(1)}$ define two rings, and t is chosen smaller than the radius of the smaller one (in our case $\mathbf{o_2}^{(1)}$).

for $i \neq j$ must be uncorrectable; the certifier has to choose a maximum allowed error $t < l_o$ smaller than the minimum Hamming weight of $\mathbf{o_i}^{(j)}$, over all $i \neq j$ (see Fig. 4).

There is a tradeoff between t and the entropy of the shifters: a high entropy, associated to a longer length l_w of $\mathbf{w_i}$, is equivalent to similar states with a small error in case of a wrong implementation, and t has to be chosen low. The certifier may decide to delete challenge-response pairs from the challenge-response table in order to choose a higher t and increase the resistance of the PUF against the noise.

For practical purposes we define two functions $\hat{G}_R$ and $\hat{G}_H$ such that

$$\hat{G}(\cdot) = (\hat{G}_R(\cdot), \hat{G}_H(\cdot)), \tag{6}$$

and therefore $\mathbf{r_i} = \hat{G}_R(\mathbf{y_i})$ and $\mathbf{h_i} = \hat{G}_H(\mathbf{y_i})$ for $\mathbf{y_i} \in \mathcal{Y}$. Moreover, we define the function $\hat{F}_E$ to be the function mapping each challenge to the respective response in the enrollment stage, i.e.,

$$\hat{F}_E(\cdot) := \hat{G}_R(\hat{P}(\cdot)), \tag{7}$$

for $\mathbf{x_i} \in \mathcal{X}$ and therefore $\mathbf{r_i} = \hat{F}_E(\mathbf{x_i})$.

Summarizing, during the enrollment stage the certifier creates a set of N challenges $\mathcal{X} \in \{0, 1\}^n$ and a set of N responses $\mathcal{R} \subseteq \{0, 1\}^m$:

$$\mathcal{R} = \{ \mathbf{r_i} \in \{0, 1\}^m \mid \mathbf{r_i} = \hat{F}_E(\mathbf{x_i}); \quad \mathbf{x_i} \in \mathcal{X} \}. \tag{8}$$

They are stored into the CRT together with

(i) the set of N strings $\mathbf{w_i}$ representing how to set the shifter operator to get the correct outcome;

(ii) the parameters of the fuzzy extractor;

(iii) the (possibly public) set of helper data $\mathcal{H} \subseteq \{0, 1\}^\star$, i.e.,

$$\mathcal{H} = \{ \mathbf{h_i} \in \{0, 1\}^\star \mid \mathbf{h_i} = \hat{G}_H[\hat{P}(\mathbf{x_i})]; \quad \mathbf{x_i} \in \mathcal{X} \}. \tag{9}$$

The CRT is given to Alice and the PUF to Bob, concluding the enrollment stage.

B. Verification

In the verification stage, Bob declares his identity and allows Alice to (remotely) interact with his PUF. Alice, equipped with the CRT, retraces the steps made by the certifier in the enrollment stage.

She picks up a randomly selected challenge $\mathbf{x_j} \in \mathcal{X}$ [for which she knows the response $\mathbf{r_j} = \hat{F}_E(\mathbf{x_j})$] and prepares the challenge state $\vec{x}_j$. The PUF transforms $\vec{x}_j$ into the outcome state $\hat{\Pi}(\vec{x}_j)$. At this point, Alice tunes the shifter $\hat{\Omega}_j$, according to the CRT and evaluates $\hat{\Omega}_j[\hat{\Pi}(\vec{x}_j)]$.

After the use of the PUF and the shifter, she may obtain a noisy version of $\vec{y}_j$ because of noise or a wrong preparation of the challenge state. Moreover, the noise could come from the PUF not being the original one if an adversary Eve is impersonating Bob.

We call this noisy version $\vec{y}'_j = \hat{\Pi}^{(e)}(\vec{x}_j)$. In that case $\hat{\Omega}_j(\vec{y}'_j) \neq \vec{0}$, which leads to $\mathbf{o}'_j \neq \mathbf{0}_{l_o}$ such that $\mathbf{y}'_j = \mathbf{w_j} \parallel \mathbf{o}'_j = \hat{P}^{(e)}(\mathbf{x_j})$ is different from the $\mathbf{y_j}$ obtained by the certifier in the enrollment stage.

The outcome is then postprocessed by the reproduction function of the fuzzy extractor that was used in the enrollment stage, so Alice collects $\mathbf{z_j} := \hat{F}_V(\mathbf{x_j})$, where the function $\hat{F}_V$ represents the map between the challenges and the corresponding responses in the verification stage, i.e.,

$$\hat{F}_V := \hat{R}[\hat{P}^{(e)}(\cdot), \hat{G}_H(\hat{P}(\cdot))], \qquad (10)$$

for $\mathbf{x_j} \in \mathcal{X}$.

The claimed response $\mathbf{z_j}$ is compared with the one in the CRT: if $\mathbf{z_j} = \mathbf{r_j}$, Bob is authenticated, otherwise the protocol fails.

V. QR-PUF

The authentication scheme for quantum readout PUFs follows the structure of the classical scheme (see Sec. IV) and still uses classical challenges, responses, and fuzzy extractors in the mathematical layer. However, the implementation of the challenge states and outcome states in the physical layer is done via quantum states. At the moment, the only classical PUF which was extended to a QR-PUF is an optical PUF [22,30], for which there are some studies on side-channel attacks [33–35].

In this work we study discrete qubit states, but our approach could also be generalized to continuous-variable (QR-)PUFs [36,37]. Let us assume to work with λ qubits, so challenge states are elements of the Hilbert space $\mathbb{C}^{2^\lambda}$. We also assume that each qubit can be in a finite number of states. Like in the classical case, we can parametrize the configurations of the experimental system that implements the challenge states to obtain a set $\mathcal{X}$ of classical challenges. Let us denote the length of such strings by n to match the case of classical PUFs.

Here the challenge states are quantum; therefore challenge states will be represented by $|x_i\rangle$. Our QR-PUF will be described in an idealized way, as a unitary operation acting on a pure state to produce another pure state. In reality, this process will introduce noise: in our framework, this will be taken into account in the transition from the outcome state to the outcome string.

A. Enrollment

Since not all states are implementable, or they do not lead to distinguishable responses, the certifier selects $N \leqslant 2^n$ challenges $\mathbf{x_i} \in \mathcal{X} \subseteq \{0, 1\}^n$, where $\mathcal{X}$ is implemented by a set of nonorthogonal states $\{|x_1\rangle, \ldots, |x_N\rangle\} \in \mathbb{C}^{2^\lambda}$.

The nonorthogonality is expected to be a crucial condition, since, as a consequence of the no-cloning theorem [23], there does not exist a measurement which perfectly distinguishes nonorthogonal states. We expect that this enhances the security of QR-PUFs compared to classical PUFs, since an adversary could gain only a limited amount of information about the challenge and the outcome states.

In this work we consider separable challenge states $|x_i\rangle$, so $|x_i\rangle = \bigotimes_{k=1}^\lambda |x_{ik}\rangle$ and we can deal with single-qubit states $|x_{ik}\rangle$. The procedure can be generalized to other challenge states. The qubit states can be written in terms of some complete orthonormal basis, which we denote as $\{|0\rangle, |1\rangle\}$

$$|x_{ik}\rangle = \cos \theta_{ik} |0\rangle + e^{i\varphi_{ik}} \sin \theta_{ik} |1\rangle, \qquad (11)$$

where $\theta_{ik} \in [0, \pi]$ and $\varphi_{ik} \in [0, 2\pi]$.

The certifier sends all states to the QR-PUF, collecting the outcome states. The QR-PUF is formalized as a λ-fold tensor product of single-qubit unitary gates $\hat{\Phi} = \bigotimes_{k=1}^\lambda \hat{\Phi}_k$. Despite its form being unknown, it can be parametrized by [38]

$$\hat{\Phi}_k(\omega_k, \psi_k, \chi_k) = \begin{pmatrix} e^{i\psi_k} \cos \omega_k & e^{i\chi_k} \sin \omega_k \\ -e^{-i\chi_k} \sin \omega_k & e^{-i\psi_k} \cos \omega_k \end{pmatrix}, \quad (12)$$

with random parameters $\psi_k, \chi_k \in [0, 2\pi]$ and $\omega_k \in [0, \frac{\pi}{2}]$. The outcome state is then $|y_i\rangle = \bigotimes_{k=1}^\lambda |y_{ik}\rangle$, where

$$
\begin{aligned}
|y_{ik}\rangle &= \hat{\Phi}_k |x_{ik}\rangle \\
&= \begin{pmatrix} e^{i\psi_k} \cos \omega_k \cos \theta_{ik} + e^{i(\chi_k + \varphi_{ik})} \sin \omega_k \sin \theta_{ik} \\ -e^{-i\chi_k} \sin \omega_k \cos \theta_{ik} + e^{i(\varphi_{ik} - \psi_k)} \cos \omega_k \sin \theta_{ik} \end{pmatrix}.
\end{aligned}
$$
$$(13)$$

Like in the classical case, the certifier can design a state-dependent shifter that performs a tensor product of unitary transformations, $\hat{\Omega}_i = \bigotimes_{k=1}^\lambda \hat{\Omega}_{ik}$, each one of them mapping a specific qubit state to the reference state $|0\rangle = (1, 0)^T$. This operation is indeed unitary, because for $|y_{ik}\rangle = \cos \alpha_{ik}|0\rangle + e^{i\beta_{ik}} \sin \alpha_{ik} |1\rangle$, it holds that $\hat{\Omega}_{ik} |y_{ik}\rangle = |0\rangle$ for

$$\hat{\Omega}_{ik} = \begin{pmatrix} \cos \alpha_{ik} & e^{-i\beta_{ik}} \sin \alpha_{ik} \\ e^{i\beta_{ik}} \sin \alpha_{ik} & -\cos \alpha_{ik} \end{pmatrix}, \qquad (14)$$

which verifies $\hat{\Omega}_{ik} \hat{\Omega}_{ik}^\dagger = \hat{\Omega}_{ik}^\dagger \hat{\Omega}_{ik} = \mathbb{I}$, where $\mathbb{I}$ is the identity operator. The certifier can implement $\hat{\Omega}_i$ for each $\hat{\Phi}|x_i\rangle$ because he can repeat the experiment and characterize each outcome state by performing quantum state tomography or, as we work with pure states, compressed sensing [39].

Instead of having to change the single-qubit measurement basis for each qubit and each challenge, by applying the suitable shifter it is now possible to use the basis $\{|0\rangle, |1\rangle\}$ for all qubits of all challenges.

By definition of $\hat{\Omega}_{ik}$, if there is no error, we will measure for every qubit the state $|0\rangle$, and the results of the measurement form a string of length λ made by all zeros, $\mathbf{o_i} = \mathbf{0} = 00 \ldots 0$. If there is some error, which in the quantum case is introduced

by either the environment or an adversary, the Hamming weight of $\mathbf{o_i}$ will give us an estimate of it.

Like in the classical case, we can parametrize the experimental system that implements the shifters in terms of the (discrete) configuration it must assume to implement a specific $\hat{\Omega}_i$. Therefore, a given $\hat{\Omega}_i$ is represented by a classical string $\mathbf{w_i} \in \mathcal{W}$ of length l_w.

We again define as *outcome* a classical string $\mathbf{y_i}$ of length $l = l_w + \lambda$, given by

$$\mathbf{y_i} = \mathbf{w_i} \, \| \, \mathbf{o_i}, \tag{15}$$

where $\|$ is the concatenation of strings.

We also define a set $\mathcal{Y}$ like in Eq. (2) and a function $\hat{P} : \mathcal{X} \to \mathcal{Y}$ mapping every challenge to the corresponding outcome. At this point, like for classical PUFs, the certifier fixes the correctable amount of noise $t < l_o$ and selects a fuzzy extractor $(\hat{G}, \hat{R})$, able to correct t errors and to generate a uniformly distributed response, according to the distribution of the outcome states and the entropy of the set of outcomes. The nonorthogonality of the challenge states affects t: when a wrong challenge state is implemented, its *fidelity* with the correct one is preserved by the QR-PUF and the shifter, since they are unitary maps, and influences the results of the measurement. The maximum correctable error t has to be chosen lower than the error produced by wrong implementations, which becomes small for highly nonorthogonal challenges. The certifier may decide to delete challenge-response pairs from the challenge-response table in order to choose a higher t and increase the resistance of the QR-PUF against the noise. However, this reduces the overall nonorthogonality of the quantum states, thus improving Eve's ability to distinguish them. Such a tradeoff will be discussed again in the following sections.

The generation function of a fuzzy extractor generates a uniformly distributed response $\mathbf{r_i} \in \mathcal{R}$, together with a public helper data $\mathbf{h_i} \in \mathcal{H}$. Again, we have

$$\hat{G}(\cdot) = (\hat{G}_R(\cdot), \hat{G}_H(\cdot)), \tag{16}$$

and

$$\mathbf{r_i} = \hat{G}_R(\mathbf{y_i}), \quad \forall \, \mathbf{y_i} \in \mathcal{Y}. \tag{17}$$

We define a function $\hat{F}_E(\cdot) := \hat{G}_R(\hat{P}(\cdot)) : \mathcal{X} \to \mathcal{R}$, mapping each challenge to the corresponding response, representing the action of the QR-PUF in the enrollment stage. Like for classical PUFs, challenges, responses, and other information are stored in the challenge-response table, which is given to Alice, while the QR-PUF is given to Bob.

B. Verification

In the verification stage Bob allows Alice to (remotely) interact with his QR-PUF. She selects randomly a challenge $\mathbf{x_j} \in \mathcal{X}$ and prepares $|x_j\rangle$.

Using the QR-PUF with the challenge state $|x_j\rangle$, Alice may obtain $|y_j'\rangle$, different from the expected $|y_j\rangle$, because of noise or an erroneous implementation of the system or the action of a malicious intruder. Then Alice applies $\hat{\Omega}_j$ and measures each qubit state in the basis $\{ |0\rangle, |1\rangle \}$, obtaining $\mathbf{o_j'}$ and hence the outcome $\mathbf{y_j'} = \mathbf{w_j} \, \| \, \mathbf{o_j'}$. While in the ideal noiseless case $\mathbf{o_j'} = \mathbf{0}_{l_o}$, here we may measure some state $|1\rangle$ for some qubits; therefore $\mathbf{y_j'}$ could be different from the $\mathbf{y_j}$ obtained by the certifier in the enrollment stage.

The outcome is then postprocessed by the reproduction function of the fuzzy extractor that was used in the enrollment stage, so Alice collects $\mathbf{z_j} := \hat{F}_V(\mathbf{x_j})$, where the function $\hat{F}_V$ is defined like in the classical case, $\hat{F}_V(\cdot) := \hat{R}[\hat{P}^{(e)}(\cdot), \hat{G}_H(\hat{P}(\cdot))]$. Authentication succeeds if $F_E(\mathbf{x_j}) = F_V(\mathbf{x_j})$. The verification stage is schematized in Fig. 5.

VI. PROPERTIES AND FORMALIZATION

In this section, we will analyze the properties of (QR-)PUFs. As we have seen, both PUFs and QR-PUFs can be represented by a classical pair of functions $\hat{F} = (\hat{F}_E, \hat{F}_V)$ that describe the map between challenges and responses in the enrollment [$\hat{F}_E$, see Eq. (7)] or verification [$\hat{F}_V$, see Eq. (10)] stage. We will keep the same formalism for both PUFs and QR-PUFs to allow our framework to compare them, but we will also specify the practical differences.

We have seen that the noise can be a problem which can lead to false rejection in the protocols. Therefore it is important to characterize and quantify the amount of noise of a (QR-) PUF which is connected to the *robustness* of a (QR-) PUF. We take the definition of this concept from [25], adapting it to our framework and our formalism.

Definition VI.1. Let us consider a (QR-) PUF $\hat{F}$ with a set of challenges $\mathcal{X}$, where $|\mathcal{X}| = N$. $\hat{F}$ is ρ-*robust* with respect to $\mathcal{X}$ if $\rho \in [0, 1]$ is the greatest number for which

$$\frac{1}{N} \sum_{i=1}^{N} P\{\hat{F}_V(\mathbf{x_i}) = \hat{F}_E(\mathbf{x_i})\} \geqslant \rho, \tag{18}$$

where ρ is called the *robustness* of the (QR-) PUF with respect to $\mathcal{X}$.

The robustness represents the average probability that the (QR-) PUF in the verification stage outputs the correct response such that the authentication succeeds. So it represents the (QR-) PUF's ability to avoid false rejections and depends on many factors, e.g., on the average noise of the specific implementation and the parameters of the fuzzy extractor.

Regarding the robustness, we do not expect a significant advantage of QR-PUFs compared to classical PUFs. Actually, there is the possibility to have a disadvantage, because of the fragility of quantum states and of the necessity of having a low error threshold t, as the noise can originate from a possible interaction of an adversary. Any implementation with QR-PUFs has to pay special care to this issue.

Now we will discuss unclonability, which is the main parameter involved in attacks from an adversary Eve. This concept is also mildly inspired by [25] but with marked differences, mainly caused by the need of taking into account QR-PUFs. In the context of entity authentication with (QR-) PUF, the purpose of an adversary Eve is to create a clone of a (QR-) PUF such that Alice can verify with it a challenge-response pair, falsely authenticating her as Bob.

When we say *clone*, we need to specify whether we are talking of a physical or a mathematical one. A *physical clone* is an experimental reproduction of the (QR-) PUF. It will have the same physical properties as the original one, even in contexts not involved with the authentication protocol. The requirement of *physical unclonability* means that a physical
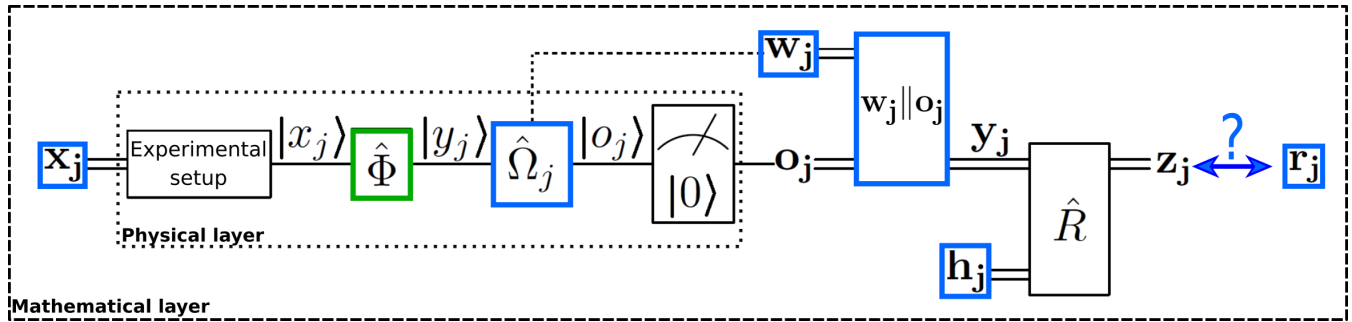
FIG. 5. A scheme for the verification stage for QR-PUFs, as described in Sec. V B. Bob provides the QR-PUF ($\hat{\Phi}$, here enclosed in a green box), and Alice uses quantities stored in the challenge-response table (here enclosed in blue boxes) to evaluate a response $\mathbf{z_j}$ for a challenge $\mathbf{x_j}$. Authentication succeeds if $\mathbf{z_j} = \mathbf{r_j}$, where $\mathbf{r_j}$ is the response stored in the CRT. The verification stage for classical PUFs (as described in Sec. IV B) can be obtained by substituting in the physical layer (the inner box) quantum states and operators with classical states and operators, and by leaving the mathematical layer (outer box) unchanged.

clone is technologically or financially unfeasible at the current state of technology.

A mathematical clone, instead, is an object that *simulates* the challenge-response behavior of a (QR-) PUF. In this case, we cannot just state that a mathematical clone is unfeasible, because if there are some correlations between the outcome states, in principle they can be exploited to predict new challenge-response pairs. As mentioned in the Introduction, several PUFs have been successfully mathematically cloned. We need to formalize this notion in order to quantify it for different (QR-) PUFs.

We assume that Eve cannot directly access the internal structure of the (QR-) PUF [24,40] but only interact with the challenge and the outcome states. An attack consists of two phases, both carried out during the verification stage of the protocol. We require that the enrollment stage is inaccessible to Eve, since this part is performed in the certifier's laboratory and it involves the study of the inner structure of the (QR-) PUF. During the *passive phase*, Eve observes a certain number of successful authentications with the real (QR-) PUF, collecting as much information as she can. Then, during the *active phase* she designs a clone and gives it to Alice, claiming to be Bob. The attack succeeds if she is authenticated as Bob.

Each interaction affects one challenge-response pair. In this context, there is a crucial difference between PUFs and QR-PUFs. Classical states can be measured without introducing disturbances and can be copied perfectly. Therefore for $q \leqslant N$ interactions, we can assume that Eve would know exactly q challenge and outcome states, possibly using this information to create a mathematical clone of the PUF.

Instead, a quantum state cannot be copied. Moreover, a quantum measurement cannot perfectly distinguish the states (since they are nonorthogonal), and any measurement can in principle introduce errors, thus potentially making a passive eavesdrop a detectable action. After q interactions, Eve would know less than q challenge and outcome states. This is the main reason for which QR-PUFs have been introduced: we expect that, concerning unclonability, they can be superior, even far superior, than classical PUFs.[3]

Definition VI.2. Let $\hat{F}$ be a (QR-) PUF with a set of challenges $\mathcal{X}$, where $|\mathcal{X}| = N$. Let us suppose that an adversary Eve has q interactions with a (QR-) PUF in the passive stage of an attack, by observing an authentication protocol between Alice and Bob. With the information she can extract, she prepares a clone $\hat{E}_q$, defined as [see Eq. (10) for a comparison]

$$\hat{E}_q(\cdot) := \hat{R}[\hat{P}_E(\cdot), \hat{G}_H(\hat{P}(\cdot))], \quad (19)$$

and gives it to Alice, who selects a challenge $\mathbf{x_i} \in \mathcal{X}$ and evaluates $\hat{E}_q(\mathbf{x_i})$.

Then $\hat{E}_q$ is a (γ, q)-*(mathematical) clone* of $\hat{F}$ if $\gamma \in [0, 1]$ is the greatest number for which

$$\frac{1}{N} \sum_{i=1}^{N} P[\hat{E}_q(\mathbf{x_i}) = \hat{F}_E(\mathbf{x_i})] \geqslant \gamma. \quad (20)$$

Definition VI.3. A (QR-) PUF $\hat{F}$ is called (γ, q)-*(mathematical) clonable* if $\gamma \in [0, 1]$ is the smallest number for which it is not possible to generate a $(\bar{\gamma}, q)$ clone of the (QR-) PUF for any $\bar{\gamma} > \gamma$. Conversely, a (QR-) PUF $\hat{F}$ is denoted as (δ, q)-*(mathematical) unclonable* if it is $(1 - \delta, q)$ clonable.

The unclonability of a (QR-) PUF is therefore related to the average probability of false acceptance. We could expect to find a relation between the number of interactions q and the unclonability; with a higher knowledge of CRP, it could be expected that Eve will be able to build a more and more sophisticated reproduction of the (QR-) PUF. Increasing q increases the know-how for making $(1 - \delta, q)$ clones with a lower δ. Therefore, fixing the maximum number of uses $q = q^*$ we fix the minimum $\delta = \delta^*$. So we ensure that for $q < q^*$, the (QR-) PUF is at least (δ^*, q) unclonable.

Definition VI.4. A (ρ, δ^*, q^*)-*secure* (QR-) PUF $\hat{F}$ is ρ-robust, physically unclonable, and at least (δ^*, q)-mathematically unclonable up to q^* uses.

When manufacturing (QR-) PUFs, several properties, that are typically implementation dependent, are important [15]. We believe that the above theoretical definitions of robustness

---

[3]As we mentioned in Sec. V A, highly nonorthogonal challenge states require a fuzzy extractor with a low correctable error, under-

mining the robustness of the QR-PUF. Therefore this feature of QR-PUFs must be used carefully, balancing robustness and unclonability.

and unclonability are, from a theoretical point of view, the main and most general properties involved in a (QR-) PUF. They are directly related to the probabilities of false rejection and false acceptance, hence describing the efficiency and the security of the entity authentication protocol. They also describe all (QR-) PUFs independently from their implementation.

VII. EXAMPLES

Explicit calculation of the robustness and the unclonability for particular (QR-) PUFs strongly depends on its implementation. In this section, we illustrate the analysis for simplified examples, starting from idealized, extreme cases.

(i) Consider a physically unclonable device implementing a true random number generator. An example of that is a QR-PUF based on the shot noise of an integrated circuit. This device is extremely difficult to simulate (Eve has to try a random guess), but also not robust at all (since it will not generate the same number in the enrollment and the verification). For this device, it holds that

$$
\begin{aligned}
\frac{1}{N} \sum_{i=1}^{N} P\{\hat{F}_V(\mathbf{x_i}) = \hat{F}_E(\mathbf{x_i})\} &= \frac{1}{N} ; \\
\frac{1}{N} \sum_{i=1}^{N} P\{\hat{E}_{q^*}(\mathbf{x_i}) = \hat{F}_E(\mathbf{x_i})\} &= \frac{1}{N} .
\end{aligned}
\tag{21}
$$

Therefore it is a $(1/N, 1 - 1/N, q^*)$ (QR-) PUF for any q^*.

(ii) Consider a physically unclonable device that outputs a fixed signal ($\vec{0}$ for classical PUFs or $|0\rangle$ for QR-PUFs) for any input. An example is an optical QR-PUF, based on the polarization of light for which a fixed polarizer is used as a shifter: for all outcome states only light waves of a specific polarization would pass though. This device is perfectly robust, but also clonable. It holds that

$$
\begin{aligned}
\frac{1}{N} \sum_{i=1}^{N} P\{\hat{F}_V(\mathbf{x_i}) = \hat{F}_E(\mathbf{x_i})\} &= 1 ; \\
\frac{1}{N} \sum_{i=1}^{N} P\{\hat{E}_{q^*}(\mathbf{x_i}) = \hat{F}_E(\mathbf{x_i})\} &= 1 .
\end{aligned}
\tag{22}
$$

Therefore the (QR-) PUF is a $(1, 0, q^*)$ (QR-) PUF, for any q^*.

These examples are extreme cases, while all (QR-) PUFs will be somewhere in between. We now focus on an example of QR-PUF to point out some features of QR-PUFs and some open points.

Let $\hat{F}$ be a QR-PUF implemented by a unitary transformation $\hat{\Phi}$, acting on λ qubits, parametrized according to Eq. (12), with $\psi_k = \chi_k = 0$, i.e.,

$$
\hat{\Phi} = \bigotimes_{k=1}^{\lambda} \hat{\Phi}_k = \bigotimes_{k=1}^{\lambda} \begin{pmatrix} \cos \omega_k & \sin \omega_k \\ -\sin \omega_k & \cos \omega_k \end{pmatrix} .
\tag{23}
$$

Consider a scenario in which each challenge state is a separable state of λ qubits, $|x_i\rangle = \bigotimes_{k=1}^{\lambda} |x_{ik}\rangle$, and each qubit is in

one of four possible states:

$$
|x_{ik}\rangle = |x_{ik}^{(\ell)}\rangle := \cos\left(\frac{\phi^{(\ell)}}{2}\right)|0\rangle + \sin\left(\frac{\phi^{(\ell)}}{2}\right)|1\rangle,
\tag{24}
$$

where

$$
\begin{aligned}
\phi^{(1)} &= \phi, & \phi^{(2)} &= -\phi, \\
\phi^{(3)} &= \phi - \pi, & \phi^{(4)} &= \pi - \phi,
\end{aligned}
\tag{25}
$$

for a fixed angle ϕ. Such challenge states can be parametrized by challenge strings of length $n = 2\lambda$; for each qubit, the four possibilities are represented by two bits.

For simplicity of notation, from now on, we drop the indices i and k, e.g., we write $|x^{(\ell)}\rangle := |x_{ik}^{(\ell)}\rangle$. The pairs $\{|x^{(1)}\rangle, |x^{(3)}\rangle\}$ and $\{|x^{(2)}\rangle, |x^{(4)}\rangle\}$ are orthogonal, but the overall set is nonorthogonal.

We assume that the noise can be parametrized as a depolarizing channel, associated to a probability of error $\tilde{p}$ and equal for all qubits. The noisy challenge state reads

$$
\begin{aligned}
\tilde{\rho}_x &:= (1 - \tilde{p})|x\rangle\langle x| + \tilde{p}\frac{\hat{I}}{2} \\
&= \left[(1 - \tilde{p})\cos^2\left(\frac{\phi^{(\ell)}}{2}\right) + \frac{\tilde{p}}{2}\right]|0\rangle\langle 0| \\
&\quad + \left[(1 - \tilde{p})\sin\left(\frac{\phi^{(\ell)}}{2}\right)\cos\left(\frac{\phi^{(\ell)}}{2}\right)\right](|0\rangle\langle 1| + |1\rangle\langle 0|) \\
&\quad + \left[(1 - \tilde{p})\sin^2\left(\frac{\phi^{(\ell)}}{2}\right) + \frac{\tilde{p}}{2}\right]|1\rangle\langle 1| .
\end{aligned}
\tag{26}
$$

The shifter needs to map the noiseless outcome state to $|0\rangle \dots |0\rangle$. According to Eq. (14) it can be chosen to be a λ-fold tensor product of single-qubit gates,

$$
\begin{aligned}
\hat{\Omega} &= \cos\left(\frac{\phi^{(\ell)}}{2} - \omega\right)|0\rangle\langle 0| + \sin\left(\frac{\phi^{(\ell)}}{2} - \omega\right)|0\rangle\langle 1| \\
&\quad + \sin\left(\frac{\phi^{(\ell)}}{2} - \omega\right)|1\rangle\langle 0| - \cos\left(\frac{\phi^{(\ell)}}{2} - \omega\right)|1\rangle\langle 1|,
\end{aligned}
\tag{27}
$$

and it follows that

$$
\tilde{\rho}_o := \hat{\Omega}\, \tilde{\rho}_y\, \hat{\Omega}^\dagger = \left(1 - \frac{\tilde{p}}{2}\right)|0\rangle\langle 0| + \left(\frac{\tilde{p}}{2}\right)|1\rangle\langle 1| .
\tag{28}
$$

For a single qubit, therefore, the probability of measuring $|1\rangle$ is $\tilde{p}/2$. For a challenge state of λ qubits, the average Hamming weight of the string $\mathbf{o_i}$ is $\lambda\, \tilde{p}/2$.

Any fuzzy extractor is defined in terms of the maximum number of errors t it can correct. With our error model, we can choose to correct the average error of the system, i.e., $t = \lceil \lambda\, \tilde{p}/2 \rceil$, where $\lceil \lambda\, \tilde{p}/2 \rceil$ is the least integer greater than or equal to $\lambda\, \tilde{p}/2$. However, t and the number N of challenge-response pairs are related since the fuzzy extractor has to uniquely map a given outcome into a unique response, without collisions.

Consider $|x^{(\ell)}\rangle$ and $|x^{(\ell')}\rangle$ ($\ell, \ell' \in \{1, 2, 3, 4\}$ and $\ell \neq \ell'$) and estimate the error if $|x^{(\ell)}\rangle$ is implemented as the state

TABLE I. Error induced by implementing the wrong challenge state. The entry in row ℓ and column ℓ' of the table is the probability of error when applying shifter ℓ to state ℓ'. The parameter ϕ is defined in Eq. (25).

	$\lvert x^{(1)}\rangle$	$\lvert x^{(2)}\rangle$	$\lvert x^{(3)}\rangle$	$\lvert x^{(4)}\rangle$
$\lvert x^{(1)}\rangle$	0	$\sin^2\phi$	1	$\cos^2\phi$
$\lvert x^{(2)}\rangle$	$\sin^2\phi$	0	$\cos^2\phi$	1
$\lvert x^{(3)}\rangle$	1	$\cos^2\phi$	0	$\sin^2\phi$
$\lvert x^{(4)}\rangle$	$\cos^2\phi$	1	$\sin^2\phi$	0

$\lvert x^{(\ell')}\rangle$, by evaluating $\hat{\Omega}_\ell\,\hat{\Phi}\lvert x^{(\ell')}\rangle$. From

$$\lvert x^{(\ell)}\rangle = \cos\left(\frac{\phi^{(\ell)}}{2}\right)\lvert 0\rangle + \sin\left(\frac{\phi^{(\ell)}}{2}\right)\lvert 1\rangle,$$

$$\lvert x^{(\ell')}\rangle = \cos\left(\frac{\phi^{(\ell')}}{2}\right)\lvert 0\rangle + \sin\left(\frac{\phi^{(\ell')}}{2}\right)\lvert 1\rangle, \qquad (29)$$

it follows that

$$\hat{\Omega}_\ell\,\hat{\Phi}\lvert x^{(\ell')}\rangle$$
$$= \cos\left(\frac{\phi^{(\ell)} - \phi^{(\ell')}}{2}\right)\lvert 0\rangle + \sin\left(\frac{\phi^{(\ell)} - \phi^{(\ell')}}{2}\right)\lvert 1\rangle. \quad (30)$$

Therefore, for this case the probability of measuring $\lvert 1\rangle$ is $\sin^2[(\phi^{(\ell)} - \phi^{(\ell')})/2]$.

In Table I, the explicit values for all the combinations of the four-qubit states are listed. In case of wrong implementation, challenges with a large overlap lead to small error weights, while orthogonal challenges lead to big ones. Thus there is a tradeoff between the robustness of the QR-PUF and the quantum advantage of using indistinguishable nonorthogonal states. For any pair of possible challenge states $\lvert x_i\rangle = \bigotimes_{k=1}^{\lambda}\lvert x_{ik}\rangle$ and $\lvert x_j\rangle = \bigotimes_{k=1}^{\lambda}\lvert x_{jk}\rangle$, the average Hamming weight of the error string $\mathbf{o_i}$, obtained by the aforementioned process, is

$$\mathrm{err}_{i,j} := (n_{12} + n_{34})\sin^2\phi + (n_{13} + n_{24})$$
$$+ (n_{14} + n_{23})\cos^2\phi, \qquad (31)$$

where n_{ab} counts how many times $\lvert x_{ik}\rangle = \lvert x^{(\ell)}\rangle$ when $\lvert x_{jk}\rangle = \lvert x^{(\ell')}\rangle$ (or vice versa).

If $\mathrm{err}_{i,j} < \lceil\lambda\,\tilde{p}/2\rceil$, then the certifier should discard one of the two challenges, either $\mathbf{x_i}$ or $\mathbf{x_j}$, thus reducing the number N of possible challenge-response pairs. After this selection is repeated for all pairs of challenges, the certifier studies the entropy of the set of shifters, determining the strings $\mathbf{w_i}$ and the outcomes $\mathbf{y_i} = \mathbf{w_i} \| \mathbf{o_i}$.

The *Canetti's reusable fuzzy extractor* [41] is able to correct up to $t = (l \ln l/m)$ bits, where l is the length of the outcomes and m the length of the responses. As $l = \lambda + l_w$ is fixed, m has to be adapted to the noise level $\lceil\lambda\,\tilde{p}/2\rceil$. The correctness property of this fuzzy extractor guarantees that an error smaller than t is corrected with probability $1 - \tilde{\varrho}$, where

$$\tilde{\varrho} = \left[1 - \left(1 - \frac{t}{l}\right)^m\right]^{\xi_1} + \xi_1\xi_2, \qquad (32)$$

with ξ_1 and ξ_2 being computational parameters of the fuzzy extractor (in [41], to which we refer for a precise explanation, they are called ℓ and γ, respectively). Then the robustness of this QR-PUF is $1 - \tilde{\varrho}$.

Concerning the unclonability, one should relate the amount of information Eve obtains from the (possibly correlated) challenge-response pairs to her ability to create a mathematical clone of the QR-PUF. Unfortunately, there is no general method known to provide this relation. We expect that, for some QR-PUFs, quantum unitary gate discrimination methods [42] could be used, but this line of research goes beyond the purposes of our work. Here, we can show that QR-PUFs prevent Eve to gain too much information about challenges and responses, thus strongly hindering her ability to learn the challenge-response table.

As the optimal global attack on the challenge states is unknown, unless knowing all challenge states, here we consider an attack that acts individually on qubits. In particular, we consider the case for which, on each qubit, Eve can apply a $1 \to 2$ cloning operator, i.e., she can intercept each qubit of a challenge state during an authentication round to produce two (imperfect) copies, one of which is given back to the legitimate parties and the other is kept for herself.

For such a set of states, the optimal cloning transformation, i.e., the transformation who keeps the highest possible fidelity between the copies and the original states, has been derived [43] and for any challenge state $\lvert x_i\rangle$ and its optimal copy ρ_i^E holds:

$$F\left(\lvert x_i\rangle\langle x_i\rvert, \varrho_i^E\right) := \prod_{k=1}^{\lambda}\langle x_{ik}\rvert\varrho_{ik}^E\lvert x_{ik}\rangle$$
$$= \left[\frac{1}{2}\left(1 + \sqrt{\sin^4\phi + \cos^4\phi}\right)\right]^{\lambda}. \quad (33)$$

For fixed λ, the minimum value of the fidelity is reached for $\phi = \pi/4$, for which, considering a single qubit, $F = (0.85)$. Already for 10 qubits the fidelity drops to $F = (0.20)$, and for 20 qubits, $F = (0.04)$.

Thus, Eve is not able to successfully simulate the challenge-response behavior, as she cannot even reconstruct the challenge and outcome states. Moreover, as the fidelity is preserved by unitary matrices, this result holds also for the expected outcome state $\lvert y_i\rangle$ and the actual outcome state Alice obtains after challenging the QR-PUF with her (unwittingly altered by the cloning process) challenge state. The noise is too high to be corrected by the fuzzy extractor, thus aborting the authentication protocol and exposing the presence of an intruder.

For classical PUFs, instead, Eve could perfectly read the challenge and outcome states, without being noticed. This provides an advantage of QR-PUFs compared to classical PUFs in terms of unclonability. However, we also noticed that a high nonorthogonality of the challenges can, in principle, undermine the robustness. The tradeoff between the advantages and disadvantages of QR-PUFs (see Table II) has to be studied in order to find secure applications of them.

TABLE II. Advantages and disadvantages of QR-PUFs compared to classical PUFs.

Advantages of QR-PUFs	Disadvantages of QR-PUFs
An adversary cannot copy or distinguish nonorthogonal states.	Highly nonorthogonal states reduce the robustness.
Adversarial measurements on the states introduce detectable disturbances.	Quantum states are more fragile than classical states.

VIII. CONCLUSION

In this article we proposed a theoretical framework for the quantitative characterization of both PUFs and QR-PUFs. After developing an authentication protocol common to both typologies, with the same error correction and privacy amplification scheme, we formalized the (QR-) PUFs in term of two main properties, the *robustness* (connected to false rejection) and the *unclonability* (connected to false acceptance). Finally, we studied some examples, motivating the possible advantages and disadvantages of QR-PUFs compared to classical PUFs.

Our framework is useful to study and to compare different implementations of (QR-) PUFs and to develop new authentication schemes. An important application would be to strictly prove the superiority of QR-PUFs over classical PUFs. The next step towards that goal would be the development of new methods to estimate the unclonability of (QR-) PUFs for different implementations. This could open an interesting line of theoretical and experimental research about (QR-) PUFs. Furthermore, our framework can be employed to determine the level of security of using (QR-) PUFs in other cryptographic protocols, like QKD, where a quantitatively secure (QR-) PUF can be used as authentication and reduces the number of necessary preshared key bits.

Note added. Recently we became aware of a preprint on a related topic [44].

[1] K. M. Martin, *Everyday Cryptography: Fundamental Principles and Applications* (Oxford University Press, Oxford, England, 2012).

[2] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, The security of practical quantum key distribution, Rev. Mod. Phys. **81**, 1301 (2009).

[3] M. N. Wegman and J. L. Carter, New hash functions and their use in authentication and set equality, J. Comput. Syst. Sci. **22**, 265 (1981).

[4] R. Pappu, Physical one-way functions, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[5] U. Rührmair, Oblivious transfer based on physical unclonable functions, in *International Conference on Trust and Trustworthy Computing* (Springer, New York, 2010), pp. 430–440.

[6] U. Rührmair and M. van Dijk, On the practical use of physical unclonable functions in oblivious transfer and bit commitment protocols, J. Cryptogr. Eng. **3**, 17 (2013).

[7] C. Brzuska, M. Fischlin, H. Schröder, and S. Katzenbeisser, Physically unclonable functions in the universal composition framework, in *Annual Cryptology Conference* (Springer, New York, 2011), pp. 51–70.

[8] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, Physical one-way functions, Science **297**, 2026 (2002).

[9] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, A technique to build a secret key in integrated circuits for identification and authentication applications, in *2004 Symposium on VLSI Circuits, Digest of Technical Papers (IEEE Cat. No. 04CH37525)* (IEEE, New York, 2004), pp. 176–179.

[10] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, FPGA intrinsic PUFs and their use for IP protection, in *International Workshop on Cryptographic Hardware and Embedded Systems* (Springer, New York, 2007), pp. 63–80.

[11] P. Tuyls, G.-J. Schrijen, B. Škorić, J. Van Geloven, N. Verhaegh, and R. Wolters, Read-proof hardware from protective coatings, in *International Workshop on Cryptographic Hardware and Embedded Systems* (Springer, New York, 2006), pp. 369–383.

[12] R. S. Indeck and M. W. Muller, Method and apparatus for fingerprinting magnetic media. US Patent 5,365,586 (1994).

[13] L. Bossuet, X. T. Ngo, Z. Cherif, and V. Fischer, A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon, IEEE Trans. Emerg. Topics Comput. **2**, 30 (2013).

[14] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, A PUF taxonomy, Appl. Phys. Rev. **6**, 011303 (2019).

[15] R. Maes and I. Verbauwhede, Physically unclonable functions: A study on the state of the art and future research directions, in *Towards Hardware-Intrinsic Security* (Springer, New York, 2010), pp. 3–37.

[16] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, Helper data algorithms for PUF-based key generation: Overview and analysis, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **34**, 889 (2014).

[17] S. Puchinger, S. Müelich, M. Bossert, M. Hiller, and G. Sigl, On error correction for physical unclonable functions, in *SCC 2015, 10th International ITG Conference on Systems, Communications and Coding* (VDE, German Society for Biomedical Engineering, 2015), pp. 1–6.

[18] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, SIAM J. Comput. **38**, 97 (2008).

[19] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, Cloning physically unclonable functions, in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (IEEE, New York, 2013), pp. 1–6.

[20] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, Modeling attacks on physical unclonable functions, in *Proceedings of the 17th ACM Conference on Computer and Communications Security* (Association for Computing Machinery, New York, 2010), pp. 237–249.

[21] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, PUF modeling attacks on simulated and silicon data, IEEE Trans. Inf. Forensics Secur. **8**, 1876 (2013).

[22] B. Škorić, Quantum readout of physical unclonable functions, Int. J. Quantum Inf. **10**, 1250001 (2012).

[23] W. K. Wootters and W. H. Zurek, A single quantum cannot be cloned, Nature (London) **299**, 802 (1982).

[24] U. Rührmair, J. Sölter, and F. Sehnke, On the foundations of physical unclonable functions, IACR Cryptology ePrint Archive **2009**, 277 (2009).

[25] F. Armknecht, R. Maes, A.-R. Sadeghi, F.-X. Standaert, and C. Wachsmann, A formalization of the security features of physical functions, in *2011 IEEE Symposium on Security and Privacy* (IEEE, New York, 2011), pp. 397–412.

[26] R. Plaga and F. Koob, A formal definition and a new security mechanism of physical unclonable functions, in *International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance* (Springer, New York, 2012), pp. 288–301.

[27] R. Plaga and D. Merli, A new definition and classification of physical unclonable functions, in *Proceedings of the Second Workshop on Cryptography and Security in Computing Systems* (Association for Computing Machinery, New York, 2015), p. 7.

[28] J. Delvaux, Security analysis of PUF-based key generation and entity authentication, Ph.D. thesis, Katholieke Universiteit Leuven, Belgium, 2017.

[29] B. Škorić, P. Tuyls, and W. Ophey, Robust key extraction from physical uncloneable functions, in *International Conference on Applied Cryptography and Network Security* (Springer, New York, 2005), pp. 407–422.

[30] S. A. Goorden, M. Horstmann, A. P. Mosk, B. Škorić, and P. W. Pinkse, Quantum-secure authentication of a physical unclonable key, Optica **1**, 421 (2014).

[31] P. Tuyls, B. Škorić, S. Stallinga, A. H. Akkermans, and W. Ophey, Information-theoretic security analysis of physical uncloneable functions, in *International Conference on Financial Cryptography and Data Security* (Springer, New York, 2005), pp. 141–155.

[32] O. Rioul, P. Solé, S. Guilley, and J.-L. Danger, On the entropy of physically unclonable functions, in *2016 IEEE International Symposium on Information Theory (ISIT)* (IEEE, New York, 2016), pp. 2928–2932.

[33] B. Škorić, A. P. Mosk, and P. W. Pinkse, Security of quantum-readout PUFs against quadrature-based challenge-estimation attacks, Int. J. Quantum Inf. **11**, 1350041 (2013).

[34] B. Škoric, Security analysis of quantum-readout PUFs in the case of challenge-estimation attacks, Quantum Inf. Comput. **16**, 50 (2016).

[35] Y. Yao, M. Gao, M. Li, and J. Zhang, Quantum cloning attacks against PUF-based quantum authentication systems, Quantum Inf. Process. **15**, 3311 (2016).

[36] G. M. Nikolopoulos and E. Diamanti, Continuous-variable quantum authentication of physical unclonable keys, Sci. Rep. **7**, 46047 (2017).

[37] G. M. Nikolopoulos, Continuous-variable quantum authentication of physical unclonable keys: Security against an emulation attack, Phys. Rev. A **97**, 012324 (2018).

[38] K. Zyczkowski and M. Kus, Random unitary matrices, J. Phys. A: Math. Gen. **27**, 4235 (1994).

[39] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Quantum State Tomography Via Compressed Sensing, Phys. Rev. Lett. **105**, 150401 (2010).

[40] U. Rührmair, H. Busch, and S. Katzenbeisser, Strong PUFs: Models, constructions, and security proofs, in *Towards Hardware-Intrinsic Security* (Springer, New York, 2010), pp. 79–96.

[41] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. Smith, Reusable fuzzy extractors for low-entropy distributions, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (Springer, New York, 2016), pp. 117–146.

[42] C. W. Helstrom, Quantum detection and estimation theory, J. Stat. Phys. **1**, 231 (1969).

[43] D. Bruß and C. Macchiavello, Optimal cloning for two pairs of orthogonal states, J. Phys. A: Math. Gen. **34**, 6815 (2001).

[44] M. Arapinis, M. Delavar, M. Doosti, and E. Kashefi, Quantum physical unclonable functions: Possibilities and impossibilities, arXiv:1910.02126v1.