2015 IEEE Symposium on Security and Privacy

Virtual Proofs of Reality and their Physical Implementation

Ulrich Rührmair[*], J.L. Martinez-Hurtado [†], Xiaolin Xu[‡], Christian Kraeh[§],
Christian Hilgers[¶], Dima Kononchuk[||], Jonathan J. Finley[**], and Wayne P. Burleson[††]

[*]*Horst Görtz Institute for IT-Security, Ruhr Universität Bochum, 44801 Bochum, Germany. Email: ruehrmair@ilo.de*
[†]*Walter Schottky Institut, TU München, 85748 Garching, Germany. Email: leo.martinez@wsi.tum.de*
[‡]*ECE Department, UMAss Amherst, Amherst, MA 01003, USA. E-mail: xu@ecs.umass.edu*
[§] *Walter Schottky Institut, TU München, 85748 Garching, Germany. Email: christian.kraeh@wsi.tum.de*
[¶] *ZAE Bayern, Walther-Meißner-Straße 6, 85748 Garching, Germany. Email: hilgers@muc.zae-bayern.de*
[||] *CS Department, Delft University of Technology, 2628 CN Delft, Netherlands. Email: kononchukdmitry@gmail.com*
[**] *Walter Schottky Institut, TU München, 85748 Garching, Germany. E-mail: finley@wsi.tum.de*
[††] *ECE Department, UMAss Amherst, Amherst, MA 01003, USA. Email: burleson@ecs.umass.edu*

***Abstract*—We discuss the question of how *physical statements* can be proven over *digital communication channels* between two parties (a *"prover"* and a *"verifier"*) residing in two separate local systems. Examples include: *(i)* "a certain object in the prover's system has temperature $X°$C", *(ii)* "two certain objects in the prover's system are positioned at distance X", or *(iii)* "a certain object in the prover's system has been irreversibly altered or destroyed". As illustrated by these examples, our treatment goes beyond classical security sensors in considering more general physical statements. Another distinctive aspect is the underlying security model: We neither assume secret keys in the prover's system, nor do we suppose classical sensor hardware in his system which is tamper-resistant and trusted by the verifier. Without an established name, we call this new type of security protocol a *"virtual proof of reality"* or simply a *"virtual proof"* (VP).**

In order to illustrate our novel concept, we give example VPs based on temperature sensitive integrated circuits, disordered optical scattering media, and quantum systems. The corresponding protocols prove the temperature, relative position, or destruction/modification of certain physical objects in the prover's system to the verifier. These objects (so-called *"witness objects"*) are prepared by the verifier and handed over to the prover prior to the VP. Furthermore, we verify the practical validity of our method for all our optical and circuit-based VPs in detailed proof-of-concept experiments.

Our work touches upon, and partly extends, several established concepts in cryptography and security, including physical unclonable functions, quantum cryptography, interactive proof systems, and, most recently, physical zero-knowledge proofs. We also discuss potential advancements of our method, for example *"public virtual proofs"* that function without exchanging witness objects between the verifier and the prover.

***Keywords*-Virtual Proofs (VPs) of Reality, Physical Unclonable Functions (PUFs), Interactive Proof Systems, Quantum Cryptography, Physical Cryptography, Keyless Security Sensors, Physical Zero-Knowledge Proofs**

I. Introduction and Overview

The archetypical cryptographic setting consists of two or more remote parties who are connected via a digital channel. By using the latter, they want to accomplish a certain cryptographic or security task. Popular examples include the secure exchange of a secret key; confidential or authenticated communication; or secure mutual identification. All of these tasks usually have a logical or mathematical nature, and can be expressed in a purely mathematical framework. A second aspect they have in common is that their practical realization requires secret keys. However, recent years have shown that these keys can potentially be attacked by a host of malware techniques and physical approaches [1], and often represent the achilles heel of modern cryptographic hardware. As Ron Rivest put it in a keynote speech at Crypto 2011, *"calling a key 'secret' does not make it so, but rather identifies it as an interesting target for the adversary"* [31]. This suggests that classical secret keys should be avoided whenever possible.

In this paper, we thus investigate a twofold extension of the above classical security setting. We consider the following questions:

(i) How can one party *(the "prover")* prove *physical* statements over digital communication lines to another party *(the "verifier")*?
(ii) How can such proofs be led *without* classical secret keys and tamper-resistant security hardware at the location of the prover?

Following these two questions, we unfold a new method in cryptography and security in this paper, so-called *"virtual proofs of reality"* (VPs). We provide a general description, protocols, and also detailed experimental verification over the next sections.

Related Work: VPs relate to, and extend, several known concepts in cryptography and security. Firstly, they advance classical security sensors in several ways: They do not use secret keys in the "sensors" or trusted, tamper-resistant sensor hardware, i.e., they methodologically differ from classical sensors. They also extend the range of physical statements that are proven in comparison with classical

DOI 10.1109/SP.2015.12

IEEE computer society

sensors, proving, for example, the irreversible modification or destruction of a certain object.

They also generalize interactive proofs [19] from the mathematical into the physical domain. Obvious ties furthermore exist to physical unclonable functions (PUFs). Some of our example VPs use electrical and optical structures reminiscent of PUFs. However, we stress that these "PUFs" have never been used in a comparable manner before, i.e., for proving complex physical statements. Specifically, our work on VPs of sensor data and VPs of temperature (see Section III) has some links to the Sensor PUFs of Rosenfeld et al. [32]. However, our approach in this paper is much more general and takes a broader perspective on proving physical statements or phenomena over digital communication lines: For example, our optical and quantum-based VPs of destruction (Section V) clearly are distinct from classical sensors and concern more general physical phenomena. Furthermore, our paper has the novelty of presenting the first actual proofs-of-concepts in said direction, for example a first physical implementation of a circuit-based VP of temperature.

Furthermore, VPs are linked to quantum cryptography in two ways: Firstly, we exploit quantum techniques in one of our VPs of destruction. Secondly, position-based quantum cryptography [9] has some ties to our VPs of distance. One advantage of VPs here is that several negative findings and impossibility results have been proven on position-based quantum crypto [9], while we present some positive results on VPs of distance in this paper.

Our VPs are also related to very recent work by Fisch et al. [14] on physical zero-knowledge protocols. While both papers have strong similarities in their language and their general topic when taking a first look, they indeed treat quite different subjects on closer inspection: Fisch et al. concentrate on proving certain physical statements without revealing additional knowledge about the concerned physical objects. They primarily deal with the theory behind this approach, presenting no implementations. They also assume a different adversarial model, where the verifier and the prover may be in the same place, each possessing their own, trusted and unmanipulated detector or measurement device (see Section 4.1 of [14]), etc. Our VPs instead deal with the prover and verifier being spatially separated, both possessing no trusted sensors or detectors. We develop protocols for this different and novel setting, and present full proof-of-concept implementations for these new protocols. It seems fair to say that both works present two distinct, completely independent, and complementary approaches.

Organization of this Paper: Section II introduces the general setting and terminology of VPs. VPs of sensor data, location, and destruction, are treated in Sections III to V, respectively, together with their experimental proof-of-concept realizations. Public VPs are discussed in theory in Section VI. We summarize our work in Section VII.

II. General Setting and Terminology

We assume in a VP that two parties are located in two different physical systems S_1 and S_2, and can communicate with each other over a digital channel. The party in S_1 (the *"prover"*), wants to prove a physical statment to the party in S_2 (the *"verifier"*), over the digital channel. The statement describes some physical feature or phenomenon in the prover's system S_1. The proof shall achieve completeness in the sense that the prover can indeed convince the verifier with high probability if the claimed statement is true. It shall also achieve soundness in the sense that the verifier will notice with high probability if the prover tries to convince him of a false statement. Apart from the digital content of the sent digital messages, also the timing by which they arrive at the two parties may be exploited in the proof.

Even though we make no such general assumption, we may optionally assume in some of our arguments that S_1 is a "closed" physical system, i.e., that S_1 has no physical exchange of any sort with the outside, apart from the (abstract and idealized) digital channel. This reflects practical situations where the prover sits in a closed and controlled environment, for example where the role of the prover is played by a bank card inside an automated teller machine (ATM). Similar closedness assumptions can in principle also be made on the verifier's system S_2 whenever appropriate.

Two different types of virtual proofs must be distinguished. In a VP with a private set-up phase (also called *"private VP"*), we allow the verifier to prepare k physical objects $O_1, \dots, O_k$ prior to the start of the actual proof. In this phase, he can measure some characteristics of these objects and store them privately, without the prover knowing what was stored. After the set-up phase, the objects O_i are transfered to the prover's system S_i, and are being used in the VP later on. In a so-called *"public virtual proof (public VP)"*, the prover may still use a number of objects O_i in the proof, but no secure set-up phase or transfer of objects prior to the proof is assumed. The prover is indeed allowed to fabricate all objects O_i by himself. Both in a private and a public VP, the objects O_i are termed *"witness objects (WOs)"*. As mentioned earlier, these WOs shall not contain any classical secret keys nor be assumed actively tamper-resistant. The prover is allowed to open and inspect them, only being limited in his efforts by current technology.

The situation is summarized in Figure 1.

Interpreting Certain PUF-Protocols as VPs: It is perhaps worth noting that some popular PUF-protocols can be interpreted as special cases of VPs. For example, the classic PUF-based identification protocol by Pappu et al. [29], [30] could be seen as a *private VP of possession*: Any prover who is sitting in a *closed* system S_1 can show to a verifier, who fabricated a PUF and holds a private CRP-database of it, that he is now in possession of this PUF, i.e., that the PUF is located within S_1.

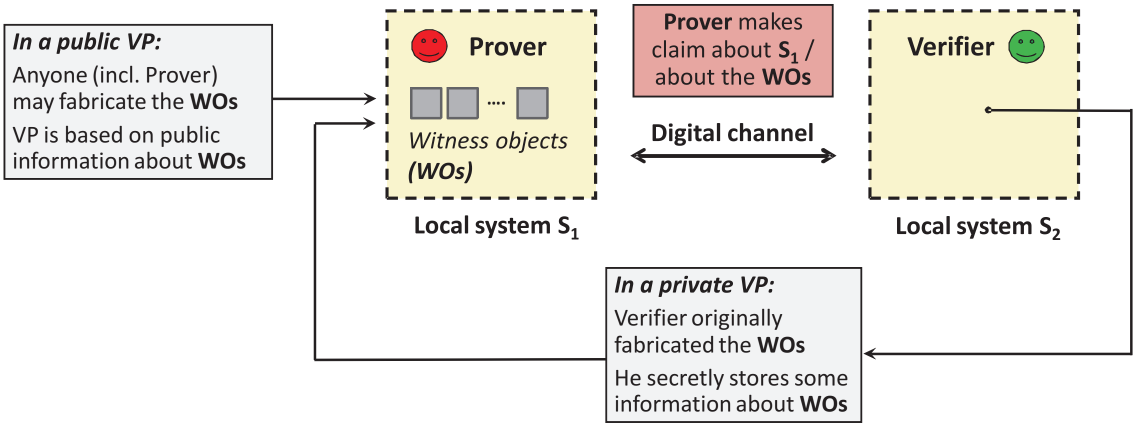
In a public VP:
Anyone (incl. Prover) may fabricate the WOs
VP is based on public information about WOs
Prover
Witness objects (WOs)
Local system S1
Prover makes claim about S1 / about the WOs
Digital channel
Verifier
Local system S2
In a private VP:
Verifier originally fabricated the WOs
He secretly stores some information about WOs

Figure 1. The general setting of public and private VPs based on witness objects (WOs). The WOs shall neither contain secret keys nor be assumed as tamper-resistant. Properties of the system S_1 are typically proven indirectly via the employed WOs.

A closely related example are the identification protocols based on so-called *"SIMPL systems"* [33], [34] or *"public PUFs (PPUFs)"* [2]. These protocols could be regarded as a *public* VP of possession: A prover sitting in a closed physical system S_1, who fabricated a SIMPL system by himself, shows to a verifier over a digital communication channel that he indeed holds this very SIMPL system. The SIMPL is thereby identified via some public data that characterizes its individual features, which is published and provided to the verifier by the prover. This data which allows the verifier to simulate the SIMPL system's input-output behavior and to check the prover's responses for correctness and speed. The proof explicitly exploits the timinig by which the messages arrive over the digital channel.

The fact that several known PUF-protocols can be regarded as special cases of VPs could be seen as positive, indicating the generality of our new concept.

III. Virtual Proofs of Sensor Data

We start our treatment by so-called *VPs of sensor data*, in which the prover shows that some sensor data measured in his system S_1 is correct and accurate. While related suggestions have been made in [32] regarding cameras, we are the first to explicitly discuss VPs of temperature, and the first to carry out a full proof-of-concept implementation. We also discuss generalizations of our approach to other one-dimensional physical variables like temperature, pressure, humidity, current, etc. Our techniques interestingly lead to a novel type of *"keyless security sensor"*, since no classical keys need to be stored in the hardware.

A. Virtual Proofs of Temperature

In a VP of temperature, the prover's aim is to show the temperature of witness objects in S_1 to the verifier. Depending on the exact circumstances of the application, this will obviously allow conclusions on the temperature of the system S_1 itself. The choice of a suitable WO is obviously decisive. We suggest employing a circuit-based Strong PUF as WO whose input-output behavior is relatively stable against voltage variations, but at the same time strongly temperature-dependent. Recall here for completeness that Strong PUFs are a PUF variant which, by definition, possess (i) a particularly complex input-output behavior, (ii) a very large number of possible challenges, and (iii) a publicly accessible input-ouptut interface, meaning that everyone who holds possession of the PUF or the PUF-embedding hardware can apply arbitrary challenges and measure the corresponding PUF-responses [37].

Interestingly, one traditional goal of Strong PUF design has always been to *minimize* temperature dependencies, for example in order to guarantee stable Strong PUF based identification schemes [30], [37]. Notable temperature dependencies have been regarded as a nuisance in this context. We demonstrate in this section, however, that they can be turned into an advantage and may be exploited usefully, allowing our VPs of temperature.

The following Protocol 1 describes our approach in general, assuming that a suitable Strong PUF has been identified. Section III-B subsequently details a proof-of-concept implementation in silicon circuits.

Protocol 1: Electrical VP of Temperature

Assumptions:

- The prover wants to show the temperature of one specific witness object, in this case one particular Strong PUF, within a temperature range $\mathbf{R_T}$. This temperature range is discretized at a certain resolution, resulting in k discrete temperature levels $t_1, \ldots, t_k \in \mathbf{R_T}$.
- The used Strong PUF is assumed to be temperature dependent in its behavior. I.e., its responses R^i_j are a function not only of the applied challenges C_j, but also

of the current (discretized) temperature t_i of the PUF: $R^i_j = F_{\mathsf{PUF}}(C_j, t_i)$.

- The used Strong PUF is suffieciently stable against other variations than temperature, e.g., voltage, aging, etc.
- The behavior of the Strong PUF varies unpredictably for the above discrete temperature levels $t_1, \ldots, t_k$. Knowing many outputs $R^i_j = F_{\mathsf{PUF}}(C_j, t_i)$ for various challenges C_j and temperatures t_i does not allow to predict unmeasured PUF-responses R^s_r for new temperatures $t_r \neq t_i$ or new challenges $C_s \neq C_j$.

Set-Up Phase:

- The verifier prepares an electrical, temperature-dependent Strong PUF with the above properties.
- He determines a private CRP-list $\mathcal{L}$ for this PUFs as follows:
  - For all considered temperature levels $t_1, \ldots, t_k$, he iterates the following procedure:
    - He puts the PUF at temperature t_i.
    - For $j = 1, \ldots, m$, he randomly chooses challenges C^i_j and applies it to the PUF (at temperature t_i). He measures the resulting response R^i_j.
  - The list $\mathcal{L}$ is then defined as $\mathcal{L} = (C^i_j, R^i_j, t_i)$ for $i = 1, \ldots, k$ and $j = 1, \ldots, m$.
- The verifier privately stores $\mathcal{L}$ and transfers the PUF to the prover.

Virtual Proof:

1) The prover claims to the verifier that the PUF is at a temperature $T \in \{t_1, \ldots, t_k\}$.
2) For $v = 1, \ldots, n$, the verifier randomly selects a tuple (C_v, R_v, T) from the list $\mathcal{L}$, and sends the value C_v to the prover.
3) For $v = 1, \ldots, n$, the prover applies the challenge C_v to the PUF, measures the response R^*_v, and sends this response to the verifier.
4) For $v = 1, \ldots, n$, the verifier compares the received value R^*_v to the values R_v in his list $\mathcal{L}$. If all values match [1], he accepts the virtual proof. Otherwise, he rejects.
5) For $v = 1, \ldots, n$, the verifier erases the tuple (C_v, R_v, T) from the list $\mathcal{L}$.

Discussion: The parameter n determines the security of the scheme. Assuming that the VP shall be executed w times, the value m should be set to $m = wn$, resulting in a list $\mathcal{L}$ of size $\Theta(wnk)$. Even though $\mathcal{L}$ may be relatively large in practice, it is still low-degree polynomial. The above technique allows proving the temperature in discrete levels of a certain step-width, which is sufficient for any practical purposes. In general, smaller step-widths have to be paid for by larger lists $\mathcal{L}$ and by a more careful design of the underlying Strong PUF.

It is interesting to ask what the above scheme exactly proves. Actually, the verifier can conclude that within the period between the times at which the last value C_v has been sent away by him in Step 2 and the last value R^*_v has been received by him in Step 3 of the protocol, the witness object was at temperature T at k points in time. In addition, the sensor/witness object by which this measurement was made is uniquely identified in the protocol. The verifier can conclude that the responses R^*_v have been obtained from this very sensor within the above time period. Finally, if the system S_1 is assumed to be closed, then also the location of the sensor is proven to lie within S_1.

The above approach generalizes easily to other simple, one-dimensional physical variables Φ, provided that Strong PUFs can be designed whose output $R_i = F_{\mathsf{PUF}}(C_i, \Phi)$ depends on Φ in a suitable manner. In these cases, Protocol 1 applies with only minor modifications. Along these lines, VPs of the current or voltage at an electrical component seem possible, or VPs of altitude, humidity, pressure, etc., provided that suitable witness objects can found. The design of such WOs appears as an interesting future research task.

B. Proof-of-Concept via Integrated Circuits

Overview: We led a first proof-of-concept of the approach of Protocol 1 on FPGAs. As certain attacks on popular Strong PUFs like Arbiter PUF, XOR Arbiter PUF and Lightweight PUF have been put forward in recent works [38] [27] [39] [45] [44], we chose to employ an XOR of four Bistable Ring PUFs (BR PUFs) as the silicon witness object. The BR PUF is a relatively recent, delay-based Strong PUF architecture [10], [11]. The schematics of a single 64-stage BR PUF are depicted in Fig. 2. Each delay cell is composed of a pair of NOR gates. In addition, a pair of multiplexors (MUXs) and demultiplexors (DEMUXs) are added at the input and output of each pair of NOR gates respectively. The shared challenge bits like $C_0 C_1 \ldots C_{63}$ ensure that either the upper or the lower path of the MUX and the DEMUX will be enabled, thus only one NOR gate out of each pair is utilized in the BR chain building.

In order to fully protect the BR PUF against some recent, preliminary modeling attacks [40], [12] (which still have relatively large error rates), we used an XOR of several BR PUFs, four in our case, constructing a "4-XOR BR PUF". According to the current state of the field, such XOR BR PUFs should be secure against any modeling and physical attacks at a single, fixed temperature level, leaving alone CRP prediction over a large temperature range. Interestingly, to the best of our knowledge, XORs of several BR PUFs have never before been studied in the literature. Besides being the first implementation of a VP of temperature, our

[1] Alternatively, the verifier may accept if more values match than given by a previously specified error bound.

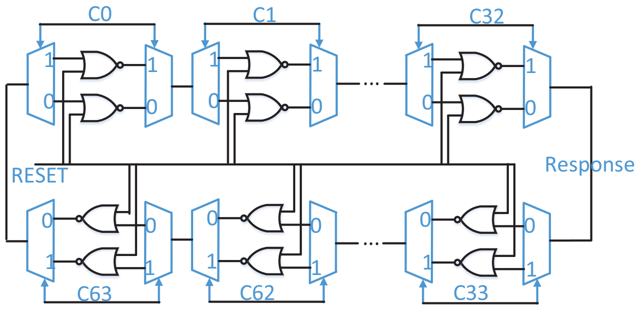

Figure 2. Schematic of a single BR-PUF with 64 stages of duplicated NOR gates. Each delay stage is composed of two parallel NOR gates, a MUX controls the output port while a DEMUX selects the input one. A shared challenge bit (select signal) ensures that only one of the two NOR gates is connected in a ring.

work therefore also constitutes the first utilization of this particular PUF design.

In our proof-of-concept, we must verify that the following two requirements are met for our 4-XOR BR PUF:

(1) There is a sufficiently large Hamming distance (HD) (or "temperature uniqueness") when a fixed CRP-set is measured at *any two* different temperature points within the considered temperature range.

(2) There is a sufficiently small HD (or "CRP-instability") when a fixed CRP-set is repeatedly measured at one fixed, single temperature.

(3) In particular, the maximal CRP-instability (caused by supply voltage, ambient noise) at each fixed temperature point should never exceed the minimal HD of any two temperature points. If this is fulfilled, the VP of temperature works; otherwise, it fails.

Please note that it was beyond our first proof-of-concept to study a statistically significant number of 4-XOR BR PUFs. All of our measurements use the same 4-XOR BR PUF implementation or "instance", and focus on its intra-die variation upon multiple measurements at the same temperature vs. its intra-die variation for measurements at different temperature levels. We stress, however, that for plain BR PUFs (i.e., without an XOR), previous works have already led detailed *inter*-die analyses [10], [11], i.e., studies on the variation between a large number of instances. These already confirmed a large *inter*-die variation of BR PUFs, implying that the variation in XOR BR PUFs can the more be regarded as non-critical. A comparably detailed statistical study on large numbers of XOR BR PUFs is left to future work.

Experiment and Results: We implemented four 64-bit BR PUFs on a Xilinx Spartan 6 FPGA. Their responses were XORed together in a post-processing step outside the FPGAs in order to obtain a 4-XOR BR PUF.

In order to study the above requirement (1), 1,000,000 pseudo-random challenges were generated with a linear feedback shift register and applied to all single BR PUFs. To mitigate the effects of noise, majority voting over eleven repetitive measurements of the response to the same challenge was performed to determine the final response. We stress that in this firt proof-of-concept experiment, this error correction step was executed *before* the XOR of the single responses. For example, if the eleven measurements resulted in at least six "0"s, the final response was set to "0". These single responses were then XORed together. The FPGA implementation was measured across several different temperatures between 27°C to 75°C with a 4°C step, whereby a Sun electronics EC12 environmental chamber was used to control the temperature.

We found that the intra-die HD between any two temperature points never drops under 3.1% and reaches a maximum of 10.7%, with most of the HDs lying in the range between 4% and 10%. Our results are depicted in Figure 3. The figure *pairwise* compares the responses at different temperature levels with each other (thin, colored zig-zag lines): Each temperature point is compared to all other temperature points, including itself (resulting in the dips that illustrate the zero-difference of each temperature point to itself).

In order to study requirement (2), the stability of the CRPs upon multiple measurement at the same temperature level had to be examined. We applied the following method: Each CRP from the above random CRP-set of size 1,000,000 was determined seven independent times under the above majority voting process (which takes the majority vote over eleven single measurements) at a fixed temperature level. Subsequently, the stability was analyzed: If a CRP was the same every seven times, it was marked as "stable"; if it was unstable at least once, it was marked as "unstable". For each fixed temperature level, the percentage of all unstable CRPs within the above CRP-set was calculated and interpreted as the intra-die HD at this level.

Following this method, we obtained the intra-die HD for all temperature level. It is depicted as the blue, thick curve at the bottom of Fig. 3. The maximal observed value is only 1.4%.

In sum, this means that the maximum HD between different measurements at the same temperatur level is notably lower than the minimum HD across all temperature points, and that also requirement (3) are fulfilled. This illustrates the basic feasibility of a VP of temperature based on electrical integrated circuits, as desired.

IV. Virtual Proofs of Location

The prover's goal in a so-called *VP of location* is to show statements about the position of one or more physical objects in his system S_1 to the verifier. Several variants are conceivable: In the most general case, the prover may try to show the absolute coordinates (within S_1) of some arbitrary objects to the prover. A more special scenarios is that the prover shows the relative location (or distance) of two witness objects. A full implementation of the latter type

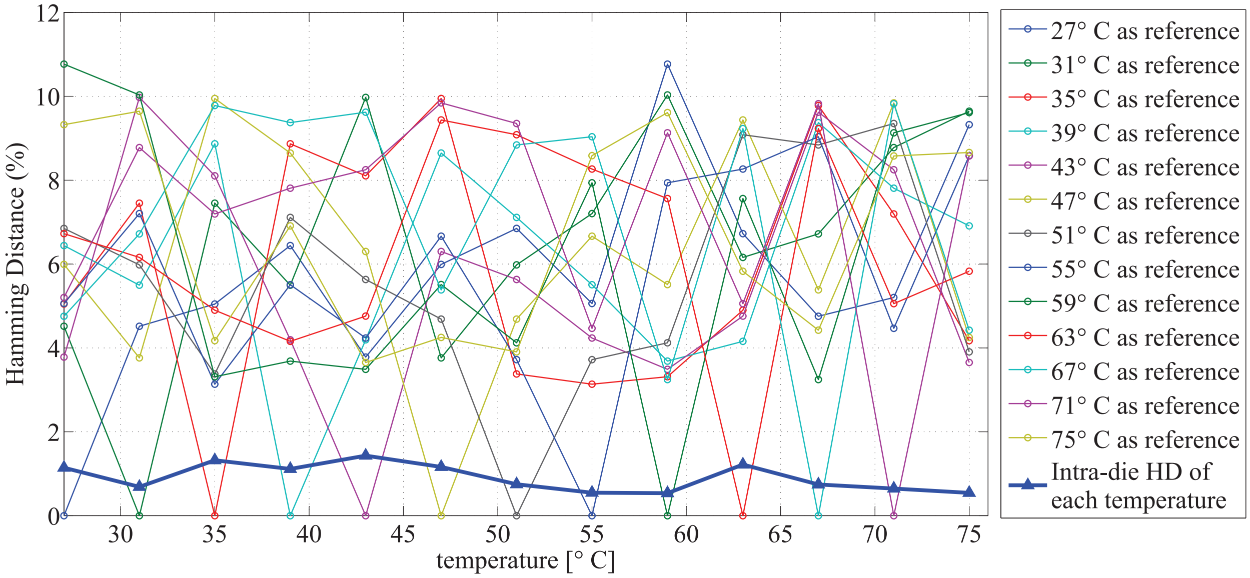

Figure 3. The figure illustrates two characterstics: Firstly, the thin, colored zig-zag lines show the HD of each temperature point to each other temperature point in a pairwise comparison. Thereby exactly one measurement was made at each temperature point. The dips to the x-axis represent the zero-different of each temperature point to itself. Secondly, the thick, blue line at the bottom of the figure illustrates the measurement stability: It shows the fraction of unstable CRPs upon multiple measurements at a fixed temperature level. This is denoted as "intra-die HD of each temperature" in the figure. Since the HD between any two different temperature levels is larger than the instability at any temperature point upon multiple measurements, the VP works.

is reported below. It is based on disordered optical systems reminiscent of Pappu's optical PUF [29], [30].

A. *Virtual Proofs of Distance*

The scattering process in Pappu et al.'s optical PUF [29], [30] and the resulting interference pattern are highly dependent on the exact relative position of the PUF, the laser source, and the recording CCD camera. While this is a disadvantage for the inexpensive practical implementation of this PUF type, it is again an advantage in our context: It can be used to prove the relative distance of two optical PUFs, which then act as witness objects, to the verifier.

The following protocol gives the details. Our proof makes the assumption that the prover wants to show small distances D to the verifier, and that the interval of possible distance has been suitably discretized.

Protocol 2: OPTICAL VP OF DISTANCE

Assumptions:

- We assume that the verifier wants to show the distance of two specific witness objects, in this case two optical PUFs à la Pappu et al. [30], [29], within a distance range $\mathbf{I_D}$. We further assume that this distance range is partitioned equally at a certain stepwidth, with k resulting discretized distances $d_1, \ldots, d_k \in \mathbf{I_D}$.

Set-Up Phase:

- The verifier prepares a first and a second optical PUF à la Pappu et al. [30], [29].
- He determines a private CRP-list $\mathcal{L}$ for these two PUFs as follows:
  - For all considered distances $d_1, \ldots, d_k$, he iterates the following procedure:
    - He places the first and the second PUF at distance d_i to each other, as in the set-up depicted in Figure 4. [2]
    - For $j = 1, \ldots, m$, he randomly chooses challenges $C^i_j = (p^i_j, \Theta^i_j)$, where p^i_j is a coordinate on the first PUF and Θ^i_j a spatial angle.
    - For $j = 1, \ldots, m$, he directs a laser beam at coordinate p^i_j and under angles Θ^i_j at the first PUF, and measures the resulting optical responses R^i_j behind the second PUF. [3]
  - The list $\mathcal{L}$ is defined as $\mathcal{L} = (C^i_j, R^i_j, d_i)$ for $i = 1, \ldots, k$ and $j = 1, \ldots, m$.
- The verifier privately stores the list $\mathcal{L}$ and transfers the two PUFs to the prover.

[2] To make this yet more precise: The two cuboid-shaped optical PUFs are positioned in such a way that their geometrical centers are on a line that is perpendicular to their largest two surfaces, and that the distance between their nearest neighbouring surfaces is d_i.

[3] Again to be precise, these responses will usually not be the raw interference patterns, but the result of an image transformation that is applied to these patterns, for example the Gabor transformation [29], [30].

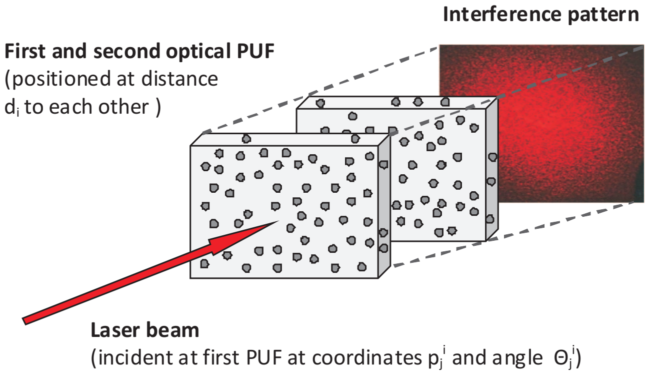
Interference pattern
First and second optical PUF
(positioned at distance
d_i to each other)
Laser beam
(incident at first PUF at coordinates p_j^i and angle Θ_j^i)

Figure 4. The basic set-up of an optical virtual proof of relative distance. Both optical PUFs participate in the interference process.

Virtual Proof:

- The prover claims to the verifier that the first and second PUF are at a distance $D \in \{d_1, \ldots, d_k\}$ in the set-up of Figure 4.
- For $v = 1, \ldots, n$, the verifier randomly selects tuples (C_v, R_v, D) from the list $\mathcal{L}$, and sends the values $C_v = (p_v, \Theta_v)$ to the prover.
- For $v = 1, \ldots, n$, the prover directs a laser beam at coordinate p_v and angle Θ_v to the first PUF in the set-up of Figure 4, measures the resulting optical response R_v^* behind the second PUF, and sends this response to the verifier.
- The verifier compares the received n values R_v^* to the values R_v in his list $\mathcal{L}$. If they match, he accepts the virtual proof, otherwise, he rejects. He erases the n used tuples (C_v, R_v, D) from the list $\mathcal{L}$.

Discussion: The above scheme is, in principle, suited to allow very small resolutions of the proved distance, down to the order of the wavelength of the employed laser light. Distance changes much smaller than that wavelength can not be resolved, however, as the optical signal will not notably change for such small differences.

One important subcase of VPs of distance are *VPs of co-locality*, where the prover wants to show that two objects are in direct neighbourhood to each other. The above schemes easily can be used for such an approach. For example, it can prove that the two witness objects have a distance smaller than a certain value d_0, for example smaller than the resolution of the VP.

We remark that also integrated circuits (ICs) could be used for VPs of distance or co-locality, even though we did not follow this route in this paper. The circuits could execute some form of “joint computation”, whose outcome of the computation should depend on their distance. In this context, it is important to see that the information exchange between the two ICs is limited: In practice, it is obviously bounded by the used interfaces; but even in theory, it is physically and fundamentally constricted by the speed of light according to Einstein, a fact that is perhaps yet more interesting for us. Let us briefly mull over the latter fundamental limit: The ICs used in a circuit-based VP of distance could operate at GHz frequencies, meaning that one clock cycle occurs every nanosecond. Within this small time period, light travels only 30cm. This could give rise to circuit-based VPs of distance at resolutions on the order of 30cm, or at least of a few meters. Such resolution would easily suffice for many security applications, for example in order to conduct certain VPs of co-locality. We would like to suggest circuit-based VPs of co-locality as an interesting future research topic.

B. Proof-of-Concept via Optical Systems

Experimental Set-Up and Methods: We used WOs similar to Pappu’s optical PUF [30] in our proof-of-concept implementation (compare Figure 4). They were fabricated according to the following the methodology: Spherical glass beads with varying diameters $80\mu m$–$840\mu m$ (Worf Glasskugeln GmbH) were mixed proportionally in a silicon polymer elastomer (poly-dimethylsiloxane, Sylgard 184, DowCorning) and deposited into a mould to fabricate the optical PUFs. After polymerization of the silicon solid hydrophobic blocks are mounted in a custom made sample holder matching the PUFs shape. Ten objects were fabricated with this method (A_1, ..., A_5 and B_1, ..., B_5). The sample holder was subsequently mounted on a x-y-z positioning stage in which a set of three coordinates and three spatial angles can be varied (p_v, Θ_v) with a goniometer. The stage is aligned in a measurement set-up in which a laser light beam is directed through the holder and sample towards a CCD. The stage allows for mounting a second sample holder with a second object at different distances along the laser path. The number of possible challenges (C_v) is in the order of 10^{11} when considering two objects. Ten challenges (C_1, ..., C_{10}) are chosen at random by varying x, y, z, θ, γ and β at random. A pair of objects A_i and B_i is positioned on the stage with distances d_1, d_2, and d_3 between the objects. The resulting speckle patterns from the laser light transmission were recorded for each pair A_i and B_i at each C_i and d_i. Additionally, the objects were unmounted separately and independently, and mounted again separately and independently, and then interrogated again by laser transmission to corroborate our mounting precision. Finally, the results are compared by calculating percentage hamming distances from Gabor transformations of the speckle pattern images, similar to the methodology introduced by Pappu et al. [30].

Results: The hamming distances before un-mounting and after re-mounting were obtained for each pair of objects at the same challenge and distance for all challenges and distances (C_i, d_i). Analogously, the objects were compared amongst themselves for all C_i and d_i, the distances among themselves for all object pairs and challenges (A_i B_i, C_i), and the challenges for all distances and an object pairs (A_i B_i, d_i). The threshold for very similar was set at $\leq$20% and completely dissimilar at $\geq$40%. Our findings show that mounting and remounting did not affect the similarity between measurements giving an average percentage variation in hamming distance of 8.7% with a 100% of the values located below the similarity threshold. Furthermore, 96% of the values fell under very similar threshold below 20%. This means mounting is very precise and speckle patterns can be replicated after mounting remounting the samples.

The object pairs are unique and should differ from one another, which is what we observed. In this case for all challenges and distances the comparison between objects yielded an average of 37.6% hamming distance variation with 98.9% of the values within the dissimilarities threshold. The challenges were also compared giving 35.7% average hamming distance with 97.2% of the values in the dissimilarities threshold, thus proving that each challenge generates unique patterns. Furthermore, in the case of distance 99.5% of the comparisons proved dissimilar with an average hamming distance of 36.3%, thus successfully demonstrating the virtual proof of distance.

We also analyzed the precision by which the measurements need to be carried out, and determined the minimum allowed variation of angles and coordinates. To this end, two coordinates and one angle were systematically altered over the whole range. The speckle patterns were collected for the minium precision variations and the results compared with one another. We found that the minimum precision given by the x-y-z stage is 10μm in 25mm range for each coordinate, and the goniometer gives 0.04° for 360° rotation range. Varying the height y for example gave 130μm precision and the rotation 0.29°, whereas varying z along the laser path gave 5.45mm precision.

Conclusion: A precise mounting measurement set-up allowed us to successfully demonstrate and implement the virtual proof of distance. Accuracy experiments indicate that a resolution of 0.13μm can be achieved over a range of 25mm by our set-up.

V. Virtual Proofs of Destruction

Let us now turn to the last VPs treated in this paper, so-called VPs of destruction. They prove that a certain object in the prover's system was irreversibly modified or "destroyed". Their existence is somewhat counterintuitive: How should one prove that a physical object has been destroyed? Given a pile of ashes, say, how should the prover argue that this pile results from a certain and unambiguously identifiable original object? How could such proofs be led for arbitrary items, not just for specially designed witness objects? Some quick thoughts along these lines illustrate that such VPs of destruction in their most general form are extremely difficult to achieve, if not straightforwardly impossible. Arbitrary hopes in this direction are hence probably unrealistic.

There are certain subforms that are simpler to accomplish, however, and which suffice for many conceivable security applications. For example, one might design a VP of destruction in the following manner:

- The prover shows that a first object O_1 is in his possession.
- The prover *"destroys"* or *"irreversibly modifies"* this object to obtain a second object O_2. The nature of the second object O_2 should be such that it is unambiguously clear that O_2 can *only* be obtained by irreversibly modifying O_1.
- The prover shows that the second object O_2 is in his possession.

The challenge here is to design a suitable object O_1 and to think out a suitable physical modification on O_1 that produces O_2. In the rest of this section, we present two constructions to this end, one optical and one quantum mechanical. As discussed above, the schemes work only for a special form of "destruction", in which enough structure is left to identify the remaining object, and to establish a link with the original. Subject to personal taste, they could also be called VPs of (irreversible) modification for this reason.

A. Optical Virtual Proofs of Destruction

The following scheme realizes a VP of destruction for an optical system that is again reminiscent of Pappu's optical PUF. The idea is to design the system in two stages, with an inner and an outer layer, and to later prove when the outer layer has been removed (see also Figure 5). The following protocol has the details.

Protocol 3: Optical VP of Destruction

Assumptions:

- We assume that the prover wants to show that a certain object has been irreversibly modified or changed.

Set-Up Phase:

- The verifier prepares a first optical PUF, for example of cuboid or spherical shape.
- The verifier collects a challenge-response list $\mathcal{L}_1$ for this first PUF. I.e., he directs a laser beam under a number of randomly chosen points and angles of incidence at the first PUF and records the optical responses.
- The verifier fully encapsulates this first PUF within a second optical PUF (see Figure 5), forming a *larger, composed optical PUF*.
  He may use a different material for forming the second PUF, for example one with a different melting point or chemical solubility than the first PUF.
- The verifier collects a challenge-response list $\mathcal{L}_C$ for the composed PUF. I.e., he directs a laser beam under a number of randomly chosen points and angles of incidence at the composed PUF and records the optical responses.
- The verifier transfers the composed PUF to the prover.

Virtual Proof:

1) The prover shows to the verifier that he is still in possession of the composed PUF. To this end, the following steps are executed:
   a) For $v = 1, \dots, n$, the verifier randomly selects a tuple (C_v, R_v) from the list $\mathcal{L}_C$, and sends the value C_v to the prover.
   b) For $v = 1, \dots, n$, the prover applies the challenge C_v to the composed PUF, measures the response R_v^*, and sends this response to the verifier.
   c) The verifier compares the received n values R_v^* to the values R_v in his list $\mathcal{L}_C$. If they match, he accepts the virtual proof, otherwise, he rejects.
2) The prover removes the encapsulating second PUF from the composed PUF, setting free the first PUF. He can do so, for example, by exploiting the different melting point or solubility of the second PUF.
3) The prover shows to the verifier that he has removed the encapsulating second PUF and revealed the first PUF. This shows that he has irreversibly modified the composed PUF. To this end, the following steps are executed:
   a) For $v = 1, \dots, n$, the verifier randomly selects a tuple (C_v, R_v) from the list $\mathcal{L}_1$, and sends the value C_v to the prover.
   b) For $v = 1, \dots, n$, the prover applies the challenge C_v to the first PUF, measures the response R_v^*, and sends this response to the verifier.
   c) The verifier compares the received n values R_v^* to the values R_v in his list $\mathcal{L}_1$. If they match, he accepts the virtual proof, otherwise, he rejects.

Discussion: What does the above VP actually prove? It shows that an irreversible modification of a specific object, namely the composed PUF, has occured in the time period between the events of the verifier sending away the first challenge C_v in Step 1a and the verifier receiving the last value R_v^* in Step 3b of the protocol. The involved objects (the first PUF and the composed PUF) are uniquely identified in this process. Finally, under the additional assumption that the prover's system is closed, the verifier can conclude that the modification has taken place in S_1.

The protocol combines two standard PUF-like challenge-response protocols with several specific hardware features of the witness objects, i.e., of the composed PUF and the first PUF. These security-relevant hardware features are:

(A) The composed PUF must have a large number of challenges. Otherwise, a fraudulent prover could read out all possible CRP and falsely prove possession of the composed PUF in Step 1, while in fact the PUF has already been modified.

(B) The composed PUF must be unclonable. Otherwise, a fraudulent prover could clone it and modify or destroy the clone instead of the original composed PUF, i.e., the unamnbiguous identification of the involved objects is then no longer maintained.

(C) Physically removing the second PUF from the composed PUF must be a practically irreversible process, i.e., it must be impossible to restore the composed PUF in its original form after the removal.

(D) Given the composed PUF, it must be impossible to obtain challenge response pairs from the first PUF by any other method (such as special physical measurements or numerical simulations) than removing the second PUF.

If the proof is executed only once in practice and surely will never be re-started (for example due to channel malfunctions or similar practical issues), then the last steps of erasing the used CRPs from the lists $\mathcal{L}_1$ and $\mathcal{L}_C$ can be left away, and the lists can be made very short.

Finally, let us remark that by using several onion-like layers around the first PUF, multiple irreversible modifications of an object can be proven in a row. Of course, such a repeated proof must assume that it is impossible to remove, clone or simulate any of these layers (compare our above list of security-relevant features).

B. Proof-of-Concept via Optical Systems

Experimental Set-Up and Methods: The WOs used for this VP were fabricated using the methodology described for the VP of distance in Section IV-B. However, there are additional manufacturing steps for the "encapsulation" of the inner PUF (see Figure 5). First, five individual objects (O_1, ..., O_5) were produced and mounted independently in the interrogation. In this VP of destruction only one object was interrogated at a time by choosing ten random challenges (C_1, ..., C_{10}) formed by choosing ten random coordinates

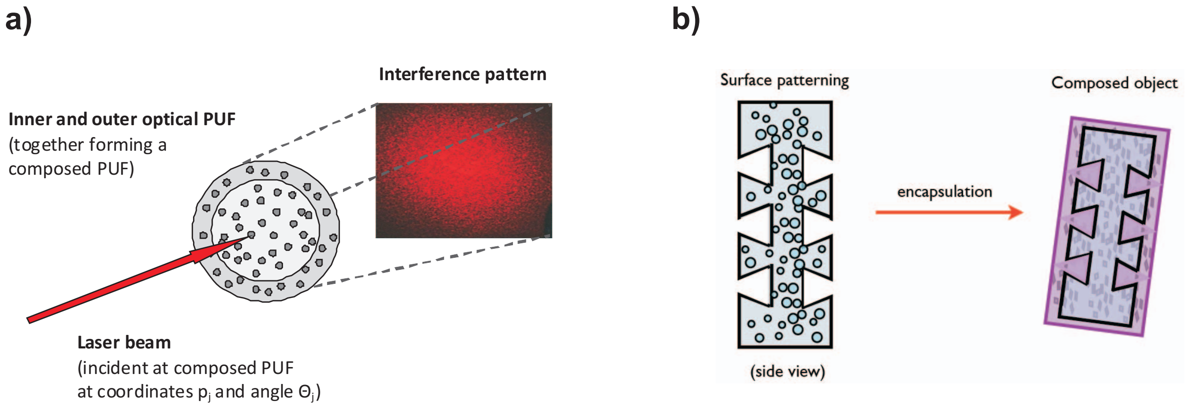

Figure 5. Part a) of the figure is a schematic illustration of our VP of destruction. It shows a system composed of two optical PUFs: A first, inner PUF, and a second, outer PUF, which encapsulates the first PUF. The outer PUF shall not be removable from the composed system without being irreversibly destroyed. Part b) illustrates the actual implementation in our proof-of-concept experiment: In order to guarantee that the outer PUF cannot be mechanically removed or separated from the inner PUF, we used pyramid-shaped indentations. They guarantee that the two PUFs cannot be mechanically separated once the matrix material has hardened. The outer PUF can only be removed by melting or chemically solving the outer matrix material. This inevitably destroys the random, unique configuration of the outer PUF.

and angles. The objects were removed from the stage mounted again and measured, mounting and remounting was preformed for every object and challenge. Then, the objects were encapsulated measured in the same fashion and then de-encapsulated and measured again as explained in the following paragraph.

The encapsulation was achieved by fabricating a capsule similar the ones used in the pharmaceutical industry. A hydrogel was prepared by mixing 9g of high molecular weight gelatin in 90mL boiling water. This mixture stays in liquid from until cooled and dried. Spherical glass beads were mixed with the solution in the same proportion as in Section IV-B. The mixture was poured into moulds containing the original objects O_i, then left to solidify at room temperature for 24h. The resulting composed objects O_i^* were mounted in the interrogation setup and measured at ten random challenges (C_1, ..., C_{10}), then unmounted, mounted again, and measured for comparison. In a subsequent step, the composed objects were irreversible destroyed by dissolving the encapsulating object with water. This process irreversibly destroys the outer object leaving the inner object intact. The original recovered objects O_i' were then interrogated again in the measuring setup. Finally, speckle pattern images were collected and the hamming distances from Gabor transformations compared for similarities or differences, either between objects, challenges or state of destruction. It is important to note that the original objects were fabricated in a hydrophobic material inert to water, which solvent compatibility limits its solubility in other common solvents [28]. These objects are encapsulated within another unique object from which the encapsulated object cannot be accessed without destroying the encapsulating layer. Inverted pyramid indentations prevent removal of the capsule by mechanical means without destroying the object. By selecting an encapsulating material with analogous optical properties but different solvent compatibility, it is possible to fully recover the encapsulated hydrophobic object. An image of the fabricated encapsulated object that we used in our VP of destruction is given in Figure 6.

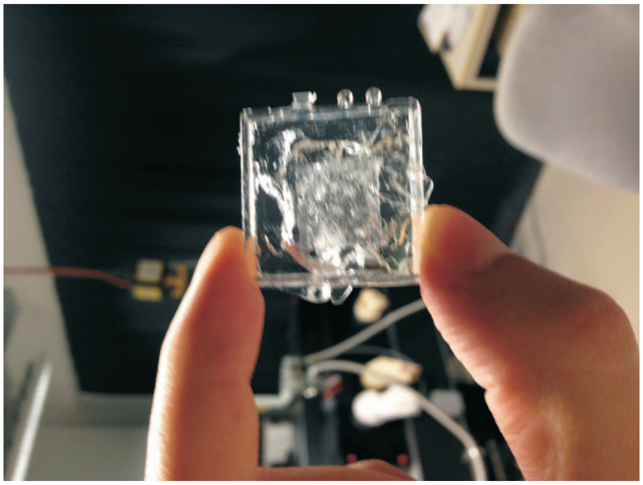

Figure 6. A picture of the encapsulated optical PUF (a “PUF-inside-a-PUF”) that we fabricated for the optical VP of destruction.

Results: The mounting precision as in the VP of distance yielded very similar results for 100% of the measurements within the similarity threshold with an average of 10.3% hamming distance variation, and 95.3% considered as very similar. This proves the mounting precision remains righteous also for one object in this VP. Similarly, comparison amongst objects and challenges before, during, and after encapsulation. The comparison amongst the original objects gave 28.5% variation for themselves and 34.7% for different challenges with 90% and 93.6% measurements within the dissimilarity threshold respectively. For the recovered objects after destruction of the capsule the values were 25.4% and 39.4% amongst themselves and amongst challenges respectively, with 92.7% and 97.8% in the dissimilarity threshold. With this results we showed that the challenges are unique as well as the objects. Finally, for the encapsulated objects the values were 25.3% and 35.2% for objects and challenges, with 71.3% and 99.3% percent within the dissimilarity threshold respectively. Varying the challenges in the encapsulated objects should lead to dissimilarities as we well observed, however not as many objects ($\leq$90%) were within the dissimilarity threshold. It is possible to explain this observation because adding an extra layer of material (i.e. encapsulation) diminishes the differences between the O_i^* due to the extra scatterers decreasing the amount of light reaching the CCD detector. Anyhow, the majority of comparisons was dissimilar and in order to perform the proof of distance one same object has to be compared with its respective encapsulated counterpart after irreversibly destroying the capsule (O_i vs O_i'). This calculation was performed for all objects at all different challenges (C_i) and also for the objects compared to their encapsulated counterpart when encapsulated. Comparing O_i to O_i^* resulted in an average 35.7% variation with 70% within the dissimilarity threshold, however no value under 20%, suggesting they are practically different. Similarly, comparing O_i^* to O_i' yielded 35.0% with 70% measurements in the dissimilarity threshold also with no values under 20%. Therefore, the encapsulated objects are different to the recovered objects. Lastly, the comparison between O_i and O_i' gave 10.9% with 98% within the similarity threshold, and most importantly 92% very similar under 20%. Thus proving that the object was fully recovered and validating the VP of destruction.

Conclusions: The virtual proof of destruction was successfully implemented proving the irreversible destruction of an encapsulating layer of material. The proof-of-concept reported here incorporates one layer of encapsulating material (i.e., one outer PUF), but several layers can be used by choosing materials with different degrees of solubility. This would allow several cascaded VPs of destruction.

C. Quantum Virtual Proofs of Destruction

Also a second well-known security technology can be exploited in VPs, namely quantum systems. The key observation in our context is that quantum systems in unknown states cannot be measured without disturbing their state. In other words: As long as the original state remains unknown, the original state cannot be rebuilt after measurement. In this sense, quantum measurements bring about some form of "irreversibly destruction" in certain situations.

It is well-known that this effect can be exploited cryptographically, for example in quantum key exchange protocols. An adversary who measures the quantum systems (e.g., polarized photons) in transmission betweeen Alice and Bob will irreversibly disturb them. This can be noticed by Alice and Bob, who may then abort transmission.

A closer look reveals that the same effect can be used to obtain a VP of destruction by quantum technology. The quantum systems (e.g., polarized photons) act as witness objects in this case. The following protocol has the details; it is very similar to the Bennett-Brassard key exchange protocol [4], and assumes some familiarity with this protocol.

Protocol 4: QUANTUM VP OF DESTRUCTION

Assumptions:

- The prover wants to show to the verifier that he has measured (and thus irreversibly altered in their state) some quantum systems $p_1, \ldots, p_k$. These quantum systems act as witness objects in the VP.
- We implicitly assume that the prover has some technology at his disposal that allows him to store the quantum systems he receives, at least for the time frames that are relevant in the context of our VP.

Set-Up Phase:

The verifier prepares k polarized photons $p_1, \ldots, p_k$ in the following fashion (compare [4]):

- He fixes two orthogonal bases $\mathcal{B}_0, \mathcal{B}_1$, for example $\mathcal{B}_0 = \{0°, 90°\}$ and $\mathcal{B}_1 = \{45°, 135°\}$
- He chooses two tuples $B = (b_1, \ldots, b_k) \in \{0,1\}^k$ (the *"bases-tuple"*) and $V = (v_1, \ldots, v_k) \in \{0,1\}^k$ (the *"value-tuple"*) at random.
- For $i = 1, \ldots, k$, he encodes the value v_i in the basis B_{b_i} in the photon p_i. He does so by suitably polarizing the photon p_i in the basis B_{b_i}, as described in [4]. For example, if he wants to encode the value "1" in the basis $\mathcal{B}_1$, he polarizes the respective photon in an angle of 90°.

The verifier sends the photons $p_1, \ldots, p_k$ to the prover.

Virtual Proof:

In order to allow the prover to show that he measures the photons (and thus irreversibly destroys their state), the prover and verifier jointly execute the following protocol:

1) The verifier chooses a tuple $T = (t_1, \ldots, t_k) \in \{0,1\}^k$ (the *"test-tuple"*) at random, and sends it to the prover.
2) For $i = 1, \ldots, k$, the prover measures the photon p_i in the basis B_{t_i}, and returns the measured value $r_i \in \{0,1\}$ to the verifier.
3) Let now $I \subseteq \{1, \ldots, k\}$ be the index set for which $t_i = b_i$. The verifier checks for all $i \in I$ that $r_i = v_i$.

If this is the case, he accepts the VP, otherwise he aborts and rejects the VP.

Discussion: The above VP is described by the example of photons, but can be carried out by other quantum systems in an analog fashion. It allows the conclusion that the measurement has taken place in the time frame between step 1, in which the verifier sends away the bistring T, and step 2, in which the values r_i are returned to the verifier.

The VP's security directly follows from the security of the Bennett-Brassard key exchange protocol [4]. The role of the external adversary in Bennett-Brassard is played by the prover in our protocol: He cannot know or derive the values encoded in the photons without knowing the bases in which they were encoded. Any measurement without knowledge of the correct bases (for example, random measurements) will both lead to wrong measured values and to a notable disturbance of the state of the photons. For example, if the measurement of the photons had taken place before the VP (i.e., in the wrong bases), the values r_i would be altered and incorrect. This would be noticed by the verifier, who would reject the VP. Furthermore, if the prover tried to present the correct answers r_i without measurement, he would fail exactly for the same reasons as an external adversary fails to derive the exchanged key in the Bennett-Brassard protocol. Again, this would be noticed in step 3, and the verifier would reject the VP. In fact, the prover's chance of measuring l photons ahead of time without being caught decrease exponentially in l (compare [4]).

Regarding our assumption of a quantum memory, we remark this very assumption implicitly underlies many quantum protocols and quantum computing proposals, without diminishing the scientific reception of these proposals. It is currently under heavy research (see [23] and references therein). The time frame for which quantum storage is required depends very strongly on the application of our protocol, and should be revisited when real applications become a topic. Our aim in this paper is different: It lies on introducing VPs, and on determining whether it is plausible that they can be realized by various technologies.

Let us have a final word on variants of the protocol. In principle, it would be possible that the prover chooses the "test-tuple" $T = (t_1, \ldots, t_k)$ by himself, and measures the quantum systems in the bases stipulated (by himself!) in the test tuple. This method saves one round of communication. However, it would only prove that the prover has measured the photons *before* a certain point in time. It would not allow the conclusion that the prover has executed the measurement *after* a certain point. In fact, he could have made the measurement a very long time ago, and simply kept the results. In this sense, Protocol 4 is a more exact method, one that allows a very close determination of the point of the measurement, i.e., of the destruction.

VI. Public Virtual Proofs

In order to complement the main contributions of the paper, let us very briefly discuss one important extension of our techniques, namely public VPs. The idea of a public VP is to avoid the set-up phase of the VP, in which the verifier needs to prepare the witness objects, and in which he needs to measure some private information about the WOs. Instead, a public VP shall be based on publicly available information only. This has the advantage that the role of the verifier can be played by an arbitrary party who has obtained this public information. In practice, this would lead to yet further efficiency improvements, comparable to the advantages of public key cryptography.

Building on the concepts presented in this paper, such public VPs appear generally possible by the use of public PUFs or SIMPL systems [33], [2]. These are PUF versions with a public simulation model, which allows the simulation of the PUF-responses under time loss compared to the real-time behavior of the system. Imagine, for example, a temperature-sensitive public PUF whose outputs depend on the applied challenge and the ambient temperature, and, in addition, can be simulated numerically under some time loss by a publicly available simulation code. Using similar techniques as in the identification protocols for SIMPL systems [33], this leads to public VPs of Temperature. The verifier is convinced of the temperature if the prover can present the correct responses of the temperature-sensitive public PUF quick enough. The correctness can be checked by the verifier by simulation, and the quickness condition simply by measuring the response time of the prover to the verifier's randomly chosen challenges. Exploiting this approach, public VPs appear feasible and realistic, further advancing the application range of VPs.

VII. Summary

Summary: We introduced a new security concept in this paper, so-called *"virtual proofs of reality"* (VPs). Figure 7 illustrates their underlying idea in its most general form: Physical systems or processes shall be converted into digital data in a way that enables a later proof that the digital data is "correct" and "authentic", i.e., that it adequately describes some features of a really existing physical system or process. So-called *"witness objects"* (WOs) may play a central role in the proofs, and may aid in transforming physical reality into digital data in an authenticatable manner.

More concretely, VPs are usually carried out between a *"prover"* and a *"verifier"*, who are situated in two separate locations/systems S_1 and S_2, and are solely connected via a digital communication channel. The prover claims that some physical statement, which concerns his general system S_1 or the WOs in S_1, holds true. He tries to prove this statement to the verifier, using the digital communication channel. The proof shall achieve completeness, i.e., the prover can indeed convince the verifier with high probability if the claimed statement is true. It shall also achieve soundness, i.e., the verifier will notice with high probability if the prover claims a false statement.

One may thereby assume that the WOs have been prepared by the verifier and were handed over in a preparation phase prior to the proof; this was referred to as a *"private VP"* by us. If the prover prepared the witness objects himself, we called the resulting scheme a *"public VP"*. We comment that all VPs realized in this paper are private VPs, but we observed that public VPs could be possible by use of techniques similar to SIMPL Systems [33] or public PUFs [2].

The concrete VPs that we studied in greater detail in our paper were:

- VPs of sensor data and temperature, where the prover claims some sensor data (for example temperature) to the verifier.
- VPs of distance and co-locality, where the prover claims the distance of two witness objects, or the fact that they are at least in close proximity (closer than some distance threshold).
- VPs of destruction, where the prover claims that a certain object has been irreversibly modified or "destroyed" within a certain time period.

The above schemes either have general novelty, such as VPs of destruction, which have not been considered in the literature before to our knowledge. Or they allow new, advantageous solutions to known problems, for example the construction of temperature sensors without classical secret keys.

Experimental proofs-of-concept have been led for all three above types of VPs. The used hardware were integrated circuits (for the VPs of temperature) and optical systems (for our VPs of distance, co-locality and destruction). The used structures are reminiscent of PUFs, and/or could be considered as certain, novel variants of PUFs. For example, our VP of temperature employs an XORed version of the Bistable Ring PUF. It advantageously exploits the known high temperature variations of this PUF type, turning them into an advantage in our context. Our VPs distance and destruction, on the other hand, employ novel variants of optical PUFs, for example a PUF that is put inside another PUF. A proof-of-concept realization for our quantum VP of destruction has been skipped, since our protocol builds on the same mechanisms as the Bennet-Brassard quantum key exchange [4]. This mechanism has been verified multiple times in previous experiments in the literature. Even without such a proof-of-concept, the observation that quantum techniques can be useful in VPs is very important, so we believe, as it shows the reach of our new concept beyond PUFs and related techniques.

We stress once more that the security model underlying our VPs notably differs from traditional settings: The prover's system and the WOs shall not contain any secret keys in the classical sense, nor shall the WOs be assumed tamper resistant in the usual manner. We wanted to avoid classical keys since they often represent the Achilles heel of modern cryptographic hardware, potentially being attackable by a host of physical and malware techniques. Furthermore, means to protect keys in mobile hardware are usually costly. Avoiding classical keys could hence lead to safer, more cost effective and more compact hardware, for example to particularly small and lightweight secure temperature sensors.

Applications: Our main concern in this work were not potential applications of VPs; rather, we focused on the introduction and plausibilization of this new concept. Still, several such applications lie at hand, and will be briefly discussed below for completeness.

To start with, VPs of distance could be used to prove that one or more objects were at a particular place in a particular point in time. Conceivable applications lie in the context of bank cards and automated teller machines (ATMs), missile tagging, or weapons inspections. They can also be employed for a verifiable joint authentication of some action via the use of two or more security tokens in the same place, one textbook example being the two security tokens of the US president and vice president required to launch an atomic weapon. Another application of the joint interference patterns arising from two different scattering objects could be (i) encryption and decryption schemes that depend on the cooperation of two parties holding the two objects, or (ii) location-dependent encryption and decryption, especially in the case that one of the scattering objects has been immobilized and is bound to one particular place (for example a terminal or ATM).

To name another example, VPs of destruction have immediate applications to the digital rights management problem: The rights to play a certain content may be linked to the existence of a certain object; if the customer no longer wants to maintain these rights (and no longer wants to pay for them), the object may be provably destroyed by the customer. He could prove the destruction to the company granting the rights, ending his period of payment. Yet other applications of VPs of destruction could lie in the field of provable and secure data deletion. Finally, VPs of temperature and, more generally, sensor data obviously have straightforward applications in secure sensor networks. Many other examples are conceivable, and are for now left to the readers and to upcoming papers.

Future Work: We believe that a host of research opportunities arises from the presented material. A first obvious next step is optimization of our proof-of-concept experiments: Which temperature resolution and distance resolutions can be achieved maximally in practice? How can we, for example, design Bistable Ring PUFs (or other electrical PUFs) for maximally finegrained VPs of temperature? This required a particularly high temperature sensitivity, but

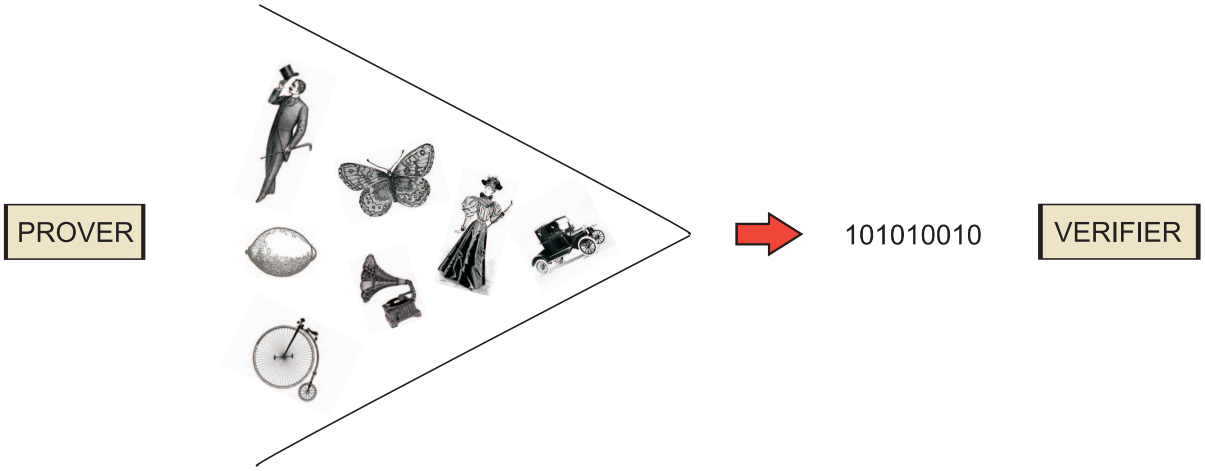

Figure 7. The idea behind virtual proofs in its most general form: Complex physical systems are converted into digital data in a way that allows proving that the digital data is "correct" and "authentic", i.e., that it corresponds to a real, actual physical system or physical process with the claimed properties. This conversion is accomplished via so-called *"witness objects"* (red arrow), without using classical secret keys or tamper proof hardware.

at the same time high stability against any other variations and aging, representing a new design goal for the circuit community. Likewise: How could optimal mechanical setups for our optical VPs look like, and which distance resolutions can be achieved? How closely can we approach the wavelength of the employed laser light in practice?

A related topic is the development of entirely new VPs: Which novel types of VPs can be imagined, and how would the corresponding witness objects have to look like?

A third possible strand of future activities concerns the logic and computational complexity aspects behind our new concept. Is there a "universal" VP, to which any other VP can be reduced, similar to the existence of universal Turing machines? Is there a "hierarchy" of physical statements that can be proven by VPs with different computational resources, communication complexities, or numbers of witness objects? Given the relationship of VPs to interactive proof systems, it seems natural to consider such issues. An extension of the Turing machine model, some sort of "physical Turing machines", could be necessary to address them with full formal rigor. Some first steps to this end have already been made in [35].

Interestingly, the above theoretical questions do not solely lie in the realm of mathematics, but overlap with physics. They follow a general recent trend of linking information theory, physics and computation. Among the many works relevant to this emerging area, we would like to exemplarily mention the arguments of G. 't Hooft [21], L. Susskind [43], R. Bousso [5], [6] and others on the *provably* limited information storage capacity of physical systems (see [3] for an easily accessible overview). These works originated in physics, but also have immediate consequences for the areas of physical compuation and information theory. For example, the authors of [21], [43], [5], [6] establish a theoretical upper bound on the information that can be stored in a given spatial volume, i.e., they show that it is impossibile to store more information in bits in a physical system than given by a low-degree polynomial bound in the system's volume. It seems quite thrilling to extend such *physical* impossibility arguments to *computational* and *information-theoretic* questions. VPs and the open theoretical issues associated with them seem to naturally fit into this young, emerging area.

REFERENCES

[1] R. J. Anderson: *Security Engineering: A guide to building dependable distributed systems*. Wiley, 2010.

[2] N. Beckmann, M. Potkonjak: *Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions*. Information Hiding 2009.

[3] J.D. Bekenstein: *How does the entropy/information bound work?* Foundations of Physics 35.11 (2005): 1805-1823.

[4] C.H. Bennett, G. Brassard: *Quantum cryptography: Public key distribution and coin tossing.* IEEE International Conference on Computers, Systems and Signal Processing, 1984.

[5] R. Bousso: *A covariant entropy conjecture.* Journal of High Energy Physics 1999.07 (1999): 004.

[6] R. Bousso: *The holographic principle.* Reviews of Modern Physics 74.3 (2002): 825.

[7] C. Brzuska, M. Fischlin, H. Schröder, S. Katzenbeisser: *Physical Unclonable Functions in the Universal Composition Framework.* CRYPTO 2011.

[8] J. Buchanan, R. Cowburn, A. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. Allwood, and M. Bryan: *Forgery: Fingerprinting documents and packaging.* Nature, vol. 436, no. 7050, p. 475, 2005.

[9] Buhrman, H., Chandran, N., Fehr, S., Gelles, R., Goyal, V., Ostrovsky, R., and Schaffner, C. *Position-based quantum cryptography: Impossibility and constructions*. CRYPTO 2011, pp. 429-446, Springer.

[10] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Rührmair: *The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions*. HOST 2011.

[11] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Rührmair: *Characterization of the Bistable Ring PUF*. DATE 2012.

[12] Y. Dai, M. Takenaka, K. Sakiyama, and N. Torii: *Security evaluation of bistable ring PUFs on FPGAs using differential and linear analysis.* Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on. IEEE, 2014.

[13] M. van Dijk, U. Rührmair: *Physical Unclonable Functions in Cryptographic Protocols: Security Proofs and Impossibility Results.* Cryptology ePrint Archive, Report 228/2012, 2012.

[14] Ben Fisch, Daniel Freund, Moni Naor: *Physical Zero-Knowledge Proofs of Physical Properties*. CRYPTO (2) 2014: 313-336

[15] B. Gassend, *Physical Random Functions*, MSc Thesis, MIT, 2003.

[16] B. Gassend, D. E. Clarke, M. van Dijk, S. Devadas: *Silicon physical random functions*. ACM Conference on Computer and Communications Security 2002, pp. 148-160, 2002

[17] B. Gassend, D. Lim, D. Clarke, M. van Dijk, S. Devadas: *Identification and authentication of integrated circuits.* Concurrency and Computation: Practice & Experience, Vol. 16(11), pp. 1077 - 1098, 2004.

[18] O. Goldreich, S. Micali, A. Wigderson: *Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems.* Journal of the ACM (JACM) 38.3 (1991): 690-728.

[19] S. Goldwasser, S. Micali, C. Rackoff: *The knowledge complexity of interactive proof systems.* SIAM Journal on computing 18.1 (1989): 186-208.

[20] J. Guajardo, S. S. Kumar, G. J. Schrijen, P. Tuyls: *FPGA Intrinsic PUFs and Their Use for IP Protection*. CHES 2007: 63-80

[21] G. 't Hooft: *Dimensional reduction in quantum gravity.* Arxiv preprint gr-qc/9310026, 1993.

[22] http://swissquantum.idquantique.com/?-Quantum-Cryptography-

[23] Khodjasteh, K., Sastrawan, J., Hayes, D., Green, T. J., Biercuk, M. J., and Viola, L. Designing a practical high-fidelity long-time quantum memory. Nature Communications, 4, 2013

[24] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, P. Tuyls: *The Butterfly PUF: Protecting IP on every FPGA.* HOST 2008: 67-70

[25] Kurtsiefer, C., Zarda, P., Halder, M., Weinfurter, H., Gorman, P. M., Tapster, P. R., and Rarity, J. G.. Quantum cryptography: A step towards global key distribution. Nature, 419(6906), 450-450.

[26] 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 with identification and authentication applications.* In Proceedings of the IEEE VLSI Circuits Symposium, June 2004.

[27] Ahmed Mahmoud, Ulrich Rührmair, Mehrdad Majzoobi, Farinaz Koushanfar: *Combined Modeling and Side Channel Attacks on Strong PUFs*. IACR Cryptology ePrint Archive 2013: 632 (2013)

[28] J. NgLee and Ch. Park and G. Whitesides: *Solvent Compatibility of Poly(dimethylsiloxane)-Based Microfluidic Devices*. Anal. Chem., Vol. 75, pp. 6544-6554, 2003.

[29] R. Pappu: *Physical One-Way Functions*. PhD Thesis, Massachusetts Institute of Technology, 2001.

[30] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld: *Physical One-Way Functions*, Science, vol. 297, pp. 2026-2030, 20 September 2002.

[31] R. Rivest: *Illegitimi non carborundum*. Invited keynote talk, CRYPTO 2011.

[32] K. Rosenfeld, E. Gavas, R. Karri: *Sensor Physical Unclonable Functions*. HOST 2010, pp. 112-117, 2010.

[33] U. Rührmair: *SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions*. Cryptology ePrint Archive, Report 2009/255, 2009.

[34] U. Rührmair: *SIMPL Systems, Or: Can We Design Cryptographic Hardware without Secret Key Information?* SOFSEM 2011.

[35] U. Rührmair: *Physical Turing Machines and the Formalization of Physical Cryptography*. Cryptology ePrint Archive, Report 2011/188, 2011.

[36] U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems.* WISTP 2010.

[37] U. Rührmair, S. Devadas, F. Koushanfar: *Security based on Physical Unclonability and Disorder.* In M. Tehranipoor and C. Wang (Editors): Introduction to Hardware Security and Trust. Springer, 2011

[38] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, S. Devadas: PUF Modeling Attacks on Simulated and Silicon Data. IEEE T-IFS, 2013.

[39] Ulrich Rührmair, Xiaolin Xu, Jan Sölter, Ahmed Mahmoud, Mehrdad Majzoobi, Farinaz Koushanfar, Wayne P. Burleson: *Efficient Power and Timing Side Channels for Physical Unclonable Functions*. CHES 2014, pp. 476-492.

[40] D. Schuster, R. Hesselbarth: *Evaluation of Bistable Ring PUFs Using Single Layer Neural Networks.* Trust and Trustworthy Computing. Springer International Publishing, 2014. 101-109.

[41] Sun Electronic Systems, Inc.: *Model EC1X Environmental Chamber User and Repair Manual*, 2011.

[42] G. E. Suh, S. Devadas: *Physical Unclonable Functions for Device Authentication and Secret Key Generation.* DAC 2007: 9-14

[43] L. Susskind: *The World as a Hologram.* J. Math. Phys. 36, pp. 6377-6396, 1995.

[44] Shahin Tajik, Enrico Dietz, Sven Frohmann, Jean-Pierre Seifert, Dmitry Nedospasov, Clemens Helfmeier, Christian Boit, Helmar Dittrich: *Physical Characterization of Arbiter PUFs*. CHES 2014, pp. 493-509.

[45] X. Xu, W. Burleson: *Hybrid side-channel/machine-learning attacks on PUFs: a new threat?*. In Proceedings of the conference on Design, Automation & Test in Europe. European Design and Automation Association, 2014.