Fakultät für Elektrotechnik und Informationstechnik
TU München

Disorder-Based Security Hardware

Ulrich Rührmair

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. W. Kellerer

Prüfer der Dissertation:

1. Univ.-Prof. Dr. U. Schlichtmann

2. Prof. M. van Dijk, Ph.D., University of Connecticut, Storrs/USA

Die Dissertation wurde am 18. 07. 2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Eletrotechnik und Informationstechnik am 22. 12. 2014 angenommen.

Disorder-Based Security Hardware

Ulrich Rührmair

March 16, 2015

To my parents
and family

Habe nun, ach! Philosophie,
Juristerei und Medizin,
Und leider auch Theologie
Durchaus studiert, mit heißem Bemühn.
Da steh' ich nun, ich armer Tor,
Und bin so klug als wie zuvor!
Heiße Magister, heiße Doktor gar,
Und ziehe schon an die zehen Jahr'
Herauf, herab und quer und krumm
Meine Schüler an der Nase herum -
Und sehe, daß wir nichts wissen können!
Das will mir schier das Herz verbrennen.
Zwar bin ich gescheiter als alle die Laffen,
Doktoren, Magister, Schreiber und Pfaffen;
Mich plagen keine Skrupel noch Zweifel,
Fürchte mich weder vor Hölle noch Teufel –
Dafür ist mir auch alle Freud' entrissen,
Bilde mir nicht ein, was Rechts zu wissen,
Bilde mir nicht ein, ich könnte was lehren,
Die Menschen zu bessern und zu bekehren.
Auch hab' ich weder Gut noch Geld,
Noch Ehr' und Herrlichkeit der Welt;
Es möchte kein Hund so länger leben!

Goethe, Faust I

STAATSANWALT Es ist kalt hier.

ELSA Du bist übermüdet, Martin, das ist alles. Du bist nervös.
Ein Prozeß nach dem anderen! Und ein Mensch wie du,
der alles so ordentlich nimmt, so gewissenhaft –

STAATSANWALT Ich weiß.

ELSA Warum machst Du keine Ferien?

STAATSANWALT Ferien in Spanien.

ELSA Der Mensch braucht das, Martin.

STAATSANWALT Vielleicht.
Er blättert in den Akten.
Vielleicht auch nicht... Hoffnung auf den Feierabend,
Hoffnung auf das Wochenende, all diese lebenslängliche
Hoffnung auf Ersatz [...], vielleicht genügte es schon, wenn man den
Millionen angestellter Seelen, die Tag für Tag an ihren
Pulten hocken, diese Art von Hoffnung nehmen würde: —
groß wäre das Entsetzen, groß die Verwandlung. Wer
weiß! Die Tat, die wir Verbrechen nennen, am Ende ist sie
nichts anderes als eine blutige Klage, die das Leben selbst
erhebt. Gegen die Hoffnung, ja, gegen den Ersatz, gegen
den Aufschub ...

Max Frisch, Graf Öderland

Wo aber Gefahr ist, wächst
Das Rettende auch.

Hölderlin, Patmos

Contents

Part I

Introduction and Overview

Chapter 1

Introduction and Overview

1.1 General Context

The need to protect sensitive information presumably is as old as the human ability to write. This arguably makes cryptography one of the oldest technical disciplines. Two of the earliest documented examples are owed to the Greek historian Herodotus. Firstly, he describes an ancient case of a steganographic technique in the conflict between Persia and Greece around 500 BC: In order to communicate sensitive information, the Greek tyrant Histiaeus shaved the head of a slave and tattoed a confidential message onto the scalp. Once the hair had re-grown, the slave could serve as a secret message carrier, passing adversarial territory unrecognizedly [27]. Secondly, Herodotus reports that around the same time, the Spartans encoded their military messages by use of a wooden stick of a well-defined diameter. A leather belt was wrapped around the stick, and the message was written across the bends. Without the stick, the symbols appeared randomly distributed over the belt; but by winding it around a stick with the same diameter, the message could be recovered [183]. Said two techniques are perhaps the first documented methods that explicitly use physical and even biological phenomena for information protection. Nothing else can be said about this thesis — albeit we now focus on physical phenomena at much smaller length scales.

2500 years after Herodotus, security and cryptography have turned into mainly civil disciplines, and have gained increasing relevance for our civil society at large. While this sentence is in writing, millions of cryptographic protocols are executed world-wide, acting as hidden, but omnipresent companions. They safeguard the internet, private and corporate communication, and the banking system. Thereby a steady shift has taken place: Cryptographic services do not only protect the secrecy of sensitive information, but also guarantee the operability and functionality of large, safety-critical, interconnected systems. In other words: Not only the citizens' privacy is at stake, but their health and well-being, if cryptographic and security schemes fail. To say this in the words of Swiss cryptographer Ueli Mauer: In the future, *"security breaches and system failures will not only be a nuisance and a cost factor, but will be intolerable, possibly major disasters"* [92].

This turns the provision of reliable and efficient security mechanisms into a central task for our society as a whole. In this context, this thesis discusses a recent and alternative approach in cryptography and security: It investigates to what extent physical disorder and unclonability can be exploited in the construction of security hardware with novel and improved features.

1.2 Why Investigate Alternative Approaches?

Before discussing any details of our new approach, let us first clarify the motivation behind it: Why should one investigate alternative approaches in hardware security at all? The reasons actually lie in a few known drawbacks of current cryptographic and security practice.

Vulnerability of Secret Keys. In agreement with Kerckhoffs' principle [63], most current security methods rest on the concept of a secret key. This forces security hardware to permanently store a digital string that is, and remains, unknown to the adversary. This requirement can be difficult to realize: On the physical level, invasive techniques, semi-invasive methods and side channel attacks may extract valuable key information [2]. On the software side, malware like Trojan horses or viruses can read out and transfer keys, even without the notice of users [2].

Three aspects play into the hands of attackers here. Firstly, secret keys stored in non-volatile memory (NVM) are permanently present in the hardware in a relatively easily accessible digital form [2]. Permanent storage can even leave traces in the memory that allow recoverage of the key *after* it has been erased [51, 168, 184]. Secondly, keys are typically strings with high entropy, allowing their identification within other, less entropic data in computer memory [161]. Finally, the requirement that modern hardware should be lightweight, mobile, and inexpensive often leaves little room for dedicated and effective key protection. Functionality and cost aspects frequently dominate security requirements in commercial scenarios [2].

Ron Rivest subsumed the situation in a keynote talk at Crypto 2011 by commenting that *"calling a bit string a secret key does not make it secret, but rather identifies it as an interesting target for the adversary"* [107]. This makes effective key protecting mechanisms — or better: methods to avoid classical keys in vulnerable hardware — an important research topic.

Practicality/Cost Aspects. There is one second potential issue of classical methods. As Pappu et al. put it in a seminal article in Science magazine [101]: *"Cryptosystems don't protect information if they're not used."* Indeed, the implementation of classical schemes in hardware makes two implicit assumptions: Firstly, that the security hardware contains non-volatile memory (NVM) cells, in which a key can be stored. Secondly, that it has sufficient computational capacities to implement the cryptographic schemes which process the key.

Both assumptions are not met in certain situations. Fistly, not all security-relevant hardware contains NVM cells. This includes central processing units (CPUs), several types of field programmable gate arrays (FPGAs), certain lightweight security systems,

etc. If the keys are stored in a second, accompanying piece of hardware (for example the computer's hard disk), the transfer of the key to units where the key is needed (CPU, FPGA, etc.) creates an explicit attack point. A self-contained security solution, which circumvents such transfer, would be preferable.

Secondly, in some low-cost scenarios, the security hardware does not possess extensive computational capacities. As an example, consider the forgery-proof tagging or "labeling" of valuable objects, such as branded products, electronic components, valuable documents, and the like. There is no computational capacity in a Rolex watch, a Nike shirt, or a paper document. Adding such capacity by RFID tags may be too expensive, apart from the obvious privacy problems it creates. Still, as pointed out by Kirovski, 7% to 10% of the world trade consists of forged products, causing an overall economic loss of the order of hundreds of billions of dollars [67].

The examples illustrate highly relevant security problems that are difficult to address by standard techniques, motivating the search for alternative approaches.

1.3 Physical Disorder and Its Useful Features

This thesis tries to address these and other problems by an alternative approach: It explores *how random disorder and imperfections in physical systems* can be exploited advantageously in security. This special focus distinguishes our efforts from other non-standard cryptography and security approaches, such as quantum cryptography [9], noise-based crypto [28], or the bounded storage model [91, 5].

It is interesting to observe that said *"physical disorder"* is omnipresent in our everyday world: Almost all physical systems exhibit it intrinsically and "for free" on small enough length scales. Four illustrating examples are given in Figure 1.1: Firstly, many biological structures show fascinating small-scale irregularities, in our example pollen of *Lilium auratum* (top left). Also customary paper exhibits notable three-dimensional randomness, for example in its interwoven *"paper fibers"*; our image shows a close-up of customary filter paper (bottom left). Thirdly, modern integrated circuits are subject to complex, random manufacturing variations as well. These variations do not affect their digital functionality, but still notably influence their exact analog properties, for example the runtime delays in their individual components (bottom right). Finally, also storage media such as compact discs are subject to imperfections, for example in the exact shape and length of their information-carrying indentations. The deviations are too small to affect the stored content, but still constitute a unique sub-structure of each disc (top right).

What are the features that makes physical disorder useful in cryptographic and security applications? We discuss four particularly important ones below.

Omnipresence. We already emphasized that almost all physical objects exhibit a certain amount of random disorder at sufficiently small length scales, or, in other words, *"if only one takes a close enough look"*. [1] This phenomenon is not limited to the

[1] The only macroscopic or mesoscopic counterexamples known to the author are highly regular crystal structures, but even they can exhibit defects or surface roughness. In addition, there are certain *microscopic* objects like photons or electrons which appear to be the same for every specimen (compare [185] for an

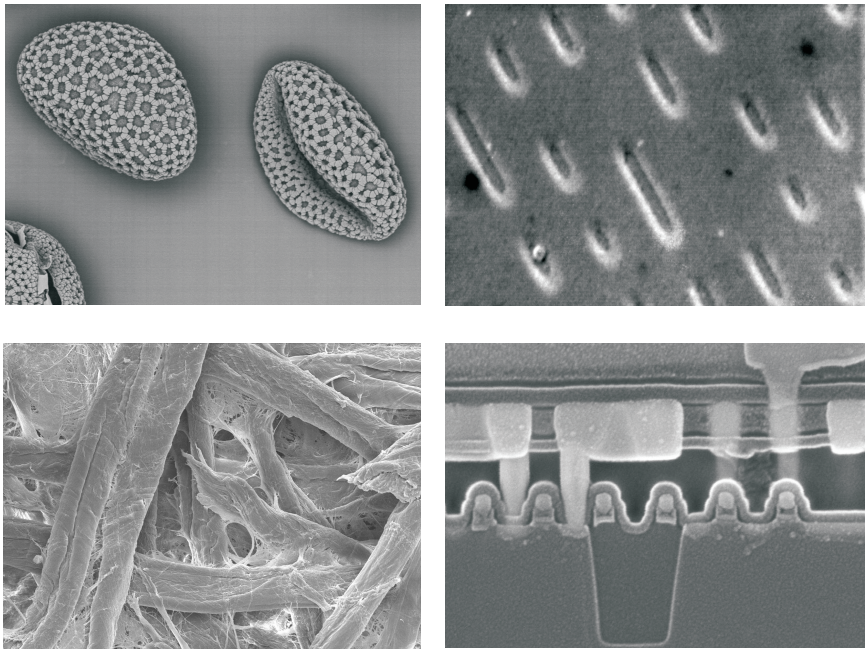

Figure 1.1: Microscopic images of several everyday objects: Pollen of *Lilium auratum* (top left) [156], ordinary (filter) paper (bottom left) [157], a customary CD (top right) [158], and a cross section through Apple's A5 chip (bottom right) [159].

examples of Figure 1.1. Readers may take a short virtual tour through their offices: Any chairs, tables, walls, windows, pens, etc. exhibit small-scale disorder and imperfections, be it due to their production process, wear and tear, or both.

In fact, it is very difficult to imagine a macroscopic system or production process that is disorder-free. Usually, this is regarded as disadvantageous, for example in the context of semiconductor manufacturing or nanofabrication. We show in this thesis, though, that the occurence of disorder can actually be turned into an advantage in the context of security.

Hard to Clone. A second central feature of physical disorder is that it is impossible to perfectly clone it with current fabrication technology.[2] This is often referred to as *"(physical) unclonability"* in the literature [101, 44] and this thesis. Interestingly, the

amusing assessment of the simlarities of all electrons by two great physicists). But such *microscopic* objects or even elementary particles are not our topic in this thesis.

[2] Please note that this type of unclonability differs from another well-known type of unclonability, namely quantum unclonability. The latter is based on inherent features of quantum mechanics, the former on the technological limitations of available two- and three-dimensional fabrication techniques.

unclonability of a system may still hold even if its entire structure is known down to every single atom to an attacker. In the physical world, *knowing* the structure of a system and *rebuilding* it accurately are actually not the same. Consider the paper surface as of Figure 1.1 as an example: Even if the exact position of all paper fibers would be known, it would still remain prohibitively difficult to refabricate it perfectly. This physical feature stands in sharp contrast to the circumstances in a mathematical or Turing machine world: If you know a bitstring exactly, it is trivial to copy it with perfect accuracy. This physical phenomenon could be termed *(re-)fabrication complexity*, in analogy to the well-known term computational complexity that underlies mathematical cryptography.

Hard to Fully Characterize. Disordered systems can possess a very large entropy or random information content. As an example, consider the random information contained in the random microscopic structure of a A4-sized sheet of paper (see again Fig. 1.1). It is infeasible to completely measure (i.e., to "characterize" in physical parlance) this information with current technology in short time. At the same time, the generation of this disorder is very inexpensive, occuring as a natural byproduct in the fabrication process. This points to a certain *asymmetry* in the physical world between *generating* randomness and *measuring* it. Again, this asymmetry has no direct analog in the Turing world: On a Turing machine, reading a bit from the tape and generating a random bit on the tape take essentially the same effort, namely one Turing step. The associated physical phenomenon could be termed *measurement complexity* or *characterization complexity*, again analog to the well-known computational complexity.

Hard to Simulate on a Turing Machine. Simulating the input-output behavior of complex, disordered physical structures on a Turing machine can be laborious. A straightforward example are the interference patterns created by disordered optical systems upon laser illumination (see [100, 101] and Section 1.4.3), but also electrical and quantum systems with similar properties exist [111, 115, 8, 30, 42, 36]. Interactions in physical systems are usually inherently parallel and analog, in contrast to digital, sequential computers. This usually makes the simulation of a complex system notably slower than the systems' real-time behavior, and can even render such simulation *practically infeasible* at all (compare [42]).

The associated phenomenon could be called the *simulation complexity* of physical structures. The occurence of physical disorder is no necessary prerequisite of simulation complexity, since also quantum systems may be hard to emulate. But the occurence of disorder can increase a system's internal complexity and the simulation overhead. As before, simulation complexity has no direct counterpart in mathematical cryptography. In our sense, it only emerges when two different worlds, for exammple the physical and the Turing world, and their "computational speeds" are compared to each other. [3]

[3] It is interesting to comment that any physical action can in principle be interpreted as a computation and, vice versa, that computation can be understood as an inherently physical process. This view has been expressed by Deutsch and others [36, 37, 170], and, in a non-scientific context, even a few years before Deutsch by novelist Douglas Adams [1]. In this sense, it appears legitimate to talk about "computational speed" also when one is actually referring to physical interactions, as we do above.

Simulation complexity is used in two different ways in this thesis. Firstly, it may render the simulation of certain disordered structures too complex to be practically feasible at all; one example are the optical scattering strucures of Pappu et al. [101] treated in Section 1.4.3. Or, simulation may be possible in practice, but notably more time consuming than the real-world behavior of the disordered structure. The latter is explicitly exploited by the so-called "SIMPL systems" treated in Part III of this thesis (see Chapters 7 and 8).

Given the above discussion of physical disorder, it seems almost straightforward to apply this phenomenon in the context of security. Why not derive unforgeable *"fingerprints"* for all everyday objects from their individual surfaces? Why not derive internal secret keys from the disorder in silicon hardware? But, as usual, the problems and challenges lie in the details. Yes, all everyday objects exhibit disorder on small length scales, ultimately when being scanned with an (expensive) atomic force microscope. But which features can be measured particularly inexpensively, are stable over time, and are still most difficult to forge or imitate? Which nanostructures and materials lead to particularly secure and practical fingerprints? How can honest users know the "correct" fingerprints of authentic objects, as opposed to the fingerprints of unauthentic objects? Etc.

Around such questions, a rich research landscape has emerged in recent years [88, 122]. It spans from nanophysics and electrical engineering to theoretical computer science and mathematics, and is concerned with implementational questions as well as with the theory behind disorder-based security. This thesis constitutes one new contribution in this field.

1.4 Examples of Disorder-Based Security Methods and Hardware

We will now illustrate the practical usability of physical disorder by three concrete examples. Among other things, our discussion details the concrete security advantages of the examples over classical techniques.

1.4.1 Certificates of Authenticity from Paper Irregularities

According to Kirovski [67], it is estimated that 7% to 8% of world trade, 10% of the pharmaceutical market, and 36% of the software market consist of counterfeit products, causing a loss of hundreds of billions of US-Dollars every year [67]. This calls for inexpensive and effective methods that verify the authenticity of products and other objects of value. Ideally, one would like to set up a system where certain *"certification authorities"*, for example product manufacturers or state authorities, can create unforgeable *"certificates of authenticity (COAs)"* for valuable objects [67]. The COAs should be machine readable, and should be verifiable by a large number of widespread *"testing devices"* [67]. Ideally, but not necessarily, the latter might be handheld and owned by security-aware consumers themselves.

Since paper is a very widespread material, it seems suggestive to utilize the random and unclonable structure of paper in this context (compare Figure 1.1). Recall that the latter induces an individual fingerprint of any paper medium, including paper documents, paper packages, and paper banknotes. Approaches in this direction have indeed been suggested by a number of researchers in the past [49, 52, 167, 14, 67]. We describe their technique by the example of paper banknotes below.

Protocol 1: CERTIFICATES OF AUTHENTICITY (COAS) FOR PAPER BANKNOTES

Set-Up Assumptions:

1. The banknote manufacturer (BM) holds a secret signing key SK from some cryptographic digital signature scheme.

2. All testing devices (TDs) hold the public verification key VK that corresponds to SK.

3. The BM has implemented a physical method to measure the random structure of a given paper surface S. The method produces a short bitstring describing the unique features of S. [4] We term this bitstring $\mathsf{UF}(\mathsf{S})$.

4. All TDs have implemented a similar method and can reproduce the measurement results of the BM in a reliable fashion. I.e., given the same piece of paper S as the BM, each TD will derive the same description $\mathsf{UF}(\mathsf{S})$, within some error thresholds.

   This presumes that the measured paper features are sufficently stable against wear-and-tear and aging.

COA Generation:

1. The BM fabricates a paper banknote. It measures the random paper structure in a selected, marked subregion S of the banknote, producing a digital string $\mathsf{UF}(\mathsf{S})$ that describes the structure.

2. The BM creates a digital signature $\mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$, and prints the information

   $$\mathsf{UF}(\mathsf{S}),\ \mathsf{I},\ \mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$$

   onto the bank note, for example via a two-dimensional barcode.

   Thereby I can be an arbitrary accompanying information, for example the banknote's value, its printing date and place, etc.

The unit consisting of an unclonable physical structure S, a digital string $\mathsf{UF}(\mathsf{S})$ that describes the unclonable features of S, and a digital signature $\mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$, is then termed a "COA" in our sense [67].

[4] One advantageous approach is shining a laser beam at the structure and measuring the resulting reflective interference pattern [14], but there are also other suitable techniques [105, 106, 52].

COA Verification:

1. The TD reads the information $\mathsf{UF}(\mathsf{S})$, I, $\mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$ from the banknote.
2. The TD verifies the validity of the digital signature $\mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$ by use of its verification key VK.
3. The TD measures the random paper structure of the banknote, and checks if the results match the information $\mathsf{UF}(\mathsf{S})$ printed on the banknote, again within some error thresholds.
4. If the tests in step 2 and 3 are passed, the TD regards the banknote as genuine.

Security Discussion. The above scheme is secure under the following assumptions:

- The adversary cannot gain access to the secret signing key stored at the manufacturer.
- The digital signature scheme is secure.
- The adversary cannot clone the paper structure, i.e., he cannot fabricate any physical system that "looks" like the original paper within the accuracy limits of the applied measurement method.

It is not too difficult to see that all of these assumptions are also necessary: If the adversary has access to the signing key, he can create COAs by himself. The same holds if the digital signature scheme is insecure, and if the adverary can forge signatures for any given plaintext. Thirdly, if the adversary can clone the paper in the above sense, he can fake notes by (i) copying the paper structure of a given banknote, and by (ii) using the very same digital signature from this note on the new, forged note.

Potential Advantages and Drawbacks. What are the advantages and disadvantages of the above approach? One notable upside lies in the way it treats secret keys. Astonishingly, there is no secret key or other security-critical secret information on the banknotes/COAs. One could allow an adversary to inspect every atom of the banknote, and still the COAs could not be forged: Knowing the paper structure and physically reproducing it are two different things. Even the testing devices do not need to contain any security-critical information, since the adversary does not benefit from learning the public key that is stored in them! Both features particularly shine in applications where adversaries can easily gain long-term access the COAs (including banknotes), or whenever there are many, widespread testing devices that can potentially be accessed by adversaries (for example widespread devices in retail stores, supermarkets, pharmacies, etc).

The only secret key of the entire scheme rests in the hands of the manufacturer, where it can usually be very well protected. There is one further noteworthy aspect in this context: Think about a scenario with many decentral fabrication sites, all of which manufacture products that need to be equipped with COAs. It seems that all of these

sites would then need to be equipped with a secret signing key, potentially creating security gaps. However, some thinking shows that indeed all necessary digital signatures could even be created centrally by one authority, and later be distributed to the sites. The sites merely send the strings UF and perhaps I to the central authority, which returns the signature $\mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$. Without going into the details, we remark that such an approach could be well applied against of gray-market IC overproduction, in particular whenever IC fabrication is outsourced. While the design is given to external manufacturers abroad, the certificatation process and signing key remains under the full control of the IP owner.

There is a third security upside of the above COAs. Standard security features of banknotes can be mass produced by the right printing equipment. In other words, the technology to produce many identical specimen exists already; if the adversary gains access to it, he will succeed. The secuity of current banknotes hence rests on the assumption that fraudsters will not gain access to a certain, existing technology. To the contrary, currently no technology exists that could exactly clone the complex, three-dimensional structure of paper. Even if developed some day, it would likely not immediately lend itself to cost-efficient mass fabrication. This creates an extra security margin against fraudsters, some sort of *"technological security"*, as opposed to the access security assumption of traditional money printing.

Readers well-versed in mathematical cryptography may object at this point: Can digital signatures provide the long-term security required in banknotes? Recall that schemes with fixed key length may become insecure after a few decades [15]. However, there are a few counterarguments to this objection. Firstly, banknotes are steadily exchanged in relatively short intervals. According to information provided by the Deutsche Bundesbank and Giesecke&Devrient [143], for example, all German banknotes are exchanged every one to five years. The newly printed notes could use longer signature keylengths, steadily adjusting security. Similar considerations hold for archival uses of COAs where long-term security is a necessity, such as birth certificates: The digital signatures could be "refreshed" by techniques well-known in the community [175]. Finally, careful choice of keylengths and elliptic curve schemes may already in itself provide strong long-term guarantees, as detailed in [76].

Let us conclude this discussion by looking at some practical aspects. While our approach offers strong security advantages, the cost and practicality aspects are rather mixed. On the upside, it becomes unnecessary to attach dedicated labels to the banknotes, creating some cost advantages. On the other hand, extra costs in the production process are generated: The random paper structure needs to be measured, the signature must be generated, and individualized information has to be printed on each note. Furthermore, verification potentially requires a costly measurement device, for example in order to position the banknote very accuractely and re-generate the original measurement value. This can partially eat up the cost savings gained by avoiding dedicated labels.

Variants. The above COA-technique based on digital signatures and unclonable structures has manifold variations. Firstly, one can attach dedicated, tailor-made unclonable structures ("labels") to the valuable objects, instead of exploiting intrinsic features of

the objects themselves. This partly creates extra cost. But at the same time, it can increase unforgeability, and may make the measurement process more efficient and inexpensive. The reason is that dedicated, tailormade labels can have extra secure and more easily measurable unique features. Various dedicated unclonable structures have been suggested to this end; see, e.g., [14, 22, 65, 66, 33, 181, 53, 164, 24, 26], and references therein.

Secondly, COAs can be used for content protection [181, 53, 60, 49]. The key observation here is that storage media may have random, unclonable features, too. For example, the small-scale structure S of the lands and pits of a CD is subject to manufacturing variations, and thus exhibits unique features $\mathsf{UF}(\mathsf{S})$ [181, 53]. Creating a digital signature $\mathsf{DigSig}_{\mathsf{SK}}(\mathsf{UF}(\mathsf{S}), \mathsf{I})$, where I contains a hash value of the digital content stored on the CD, links this very content to its unique storage medium, thus certifying it. Copying the content onto another data carrier invalidates this certificate. Similar considerations hold for content printed on paper, such as business contracts, as discussed in [49, 24, 164].

1.4.2 Secret Cryptographic Keys from SRAM Power-Up States

Secret keys are at the heart of most modern cryptographic and security schemes. But as mentioned earlier, storing them in hardware can be non-trivial: On the security side, powerful attacks have been developed, ranging from invasive to side channel techniques [2]. On the cost/practicality side, not every hardware system contains NVM cells for storing secret keys.

An alternative approach, which can potentially improve both on security and practicality, is to exploit physical disorder, more precisely the random and individual manufacturing variations in each hardware system, as a secret key source. Perhaps the most prominent example for this technique are SRAM cells: Upon power-up, each cell contains either a zero or one, depending on the random manufacturing variations present in the cells [57, 58, 50]. The power-up states are relatively well repeatable upon multiple power-ups for each single cell, but they statistically vary almost uniformly from cell to cell in an SRAM array. The k cells in an SRAM array thus together create an individual power-up state "fingerprint", allowing derivation of an individual key. In the parlance of the field, an SRAM cell can act as a so-called *"physical unclonable function"* or *"PUF"*, leading to the widespread terminology "SRAM PUF" for the above phenomenon [50].

If this approach is used in practice, there must not be a single bit flip in the derived secret key. Error correction (EC) therefore is vital, since not all states are perfectly stable upon multiple power ups. Most EC techniques thereby have in common that some public, non-secret "helper data" or "error-correcting data" is provided to the hardware system, allowing derivation of a stable key from the noisy power-up states [50]. it is well-known that this error-correcting helper data can be constructed in such a way that the helper data alone — i.e., without knowledge of the power-up values of the SRAM cells — does not leak any knowledge about the derived key. It is interesting to observe that the use of helper data shifts the problem of storing data permanently: Instead of a binary key, now the helper data needs to be stored in NVM. One difference is, though, that the helper data can be stored publicly, since it does not leak information about the

key. It needs to be provided to the secure hardware only whenever key derivation is necessary.

An illustrative example application of the above phenomenon and of *"SRAM PUFs"* is the protection of intellectual property (IP) of FPGA designs [50]. Many FPGA types do not contain non-volatile memory cells and hence cannot store application designs permanently [50]. The designs are thus put in external memory and uploaded onto the FPGA when needed. This has the disadvantage that fraudsters can intercept the uploaded bitstream and learn the designs, which often represent a very substantial IP value. In principle, the bitstream could be encrypted, but the lack of NVM on the FPGA prevents the storing of classical secret keys on the FPGA. At the same time, however, SRAM cells are present on many FPGAs, and their power-up states can be used to derive a key. This enables IP protection schemes between the manufacturer, the FPGA, and an external memory device storing the design [50]. We give one basic example of such a scheme below [50]; other, more involved techniques are described in the same reference [50].

Protocol 2: IP PROTECTION OF FPGA DESIGNS VIA SRAM PUFS [50]

Set-Up Assumptions:

1. The scheme involves four parties: The IP provider (IPP); a system integrator or designer (SYS); the FPGA-manufacturer (HWM); and a trusted third party (TPP).

2. The communication channels between HWM and TTP, and between TTP and IPP, are authenticated and confidential.

3. The TTP and the HWM are fully trusted.

4. The parties have a secure and efficient message authentication scheme MAC at their disposal.

5. The HWM can disable access to the SRAM cells after reading them out in the enrollment phase (for example by blowing some fuses). No one can access the cells anymore after this operation, including adversaries.

6. For simplicity of exposition, we do not explicitly deal with error correction in this protocol. In practice, error correcting helper data does need to be used to obtain stable responses. [5]

Initialization Phase (aka "Enrollment Protocol" [50]):

1. The HWM associates an $\mathsf{ID}_{\mathsf{HW}}$ to a given FPGA. It reads out different sets of SRAM-power up states $R_1, \ldots, R_n$ of this FPGA.

[5] Following a convention stipulated in [50], readers may interpret the protocol in such a way that C_i denotes the PUF challenge *and* the corresponding helper data required to reconstruct the PUF response R_i from a noisy version R'_i.

2. The HWM disables external access to those SRAM-cells that have provided the above response $R_1, \ldots, R_k$. Internal access for the FPGA itself to these cells must remain intact.

3. The HWM sends
$$\mathsf{ID}_{\mathsf{HW}}, R_1, \ldots, R_n$$
to the TTP.

IP Authentication Protocol:

1. SYS sends
$$\mathsf{ID}_{\mathsf{SW}}, \mathsf{ID}_{\mathsf{HW}}$$
to TTP, indicating which software $\mathsf{ID}_{\mathsf{SW}}$ shall be utilized on which FPGA hardware $\mathsf{ID}_{\mathsf{HW}}$.

2. The TTP sends $\mathsf{ID}_{\mathsf{SW}}$ to the IPP, and the IPP returns the software SW to the TTP.

3. The TTP encrypts the software with the key R_i, creating a value
$$D = \mathsf{Enc}_{R_i}(\mathsf{SW}, \mathsf{ID}_{\mathsf{SW}}).$$

4. The TTP sends the message
$$C_i, C_j, D, \mathsf{MAC}_{R_j}(C_i, C_j, D)$$
to SYS.

Design Upload and Decryption on the FPGA:

1. The FPGA uploads the encrypted bitstream created by SYS, which is stored in a (non-confidential) storage medium accompanying the FPGA.

   The bistream potentially contains k encrypted and authenticated software blocks of the above form
$$C_i^k, C_j^k, D^k, \mathsf{MAC}_{R_j^k}(C_i^k, C_j^k, D^k).$$

2. For each k, the FPGA internally reproduces the responses R_i^k and R_j^k by accessing and measuring the respective SRAM cells.

   (Please note again that in practice, error correcting helper data must be used to this end, which must be provided from and external non-volatile, but not confidential storage medium. In the case of FPGAs, the same medium can be used that stores the encrypted upload bitstream.)

   The FPGA decrypts the bitstream and verifies the authentication.

3. The FPGA is configured by the decrypted bitstream.

Security Discussion. Our security discussion below follows [50]. The protocol's aim is to achieve confidentiality and integrity of the software blocks and thus of the related IP. It achieves this aim under the following assumptions:

- The TTP and the HWM are trusted.
- The mutual communication channels TTP-HWM and TTP-IPP are confidential and authenticated.
- No adversaries can externally read-out the responses R_i and R_j after access to them has been disabled by the HWM, while the FPGA itself can still access the responses internally.

We comment that these are relatively strongly assumptions. In particular, the third hypothesis is at the least in part comparable to the standard assumption that a classical key cannot be read-out by the adversary.

Potential Advantages and Drawbacks. Perhaps the main advantage of the above scheme is that it enables security (and encryption) in an environment without NVM. Without using SRAM cells as key source, no encryption would be possible at all. This could be regarded as a practicality advantage or as a security advantage, depending on personal taste.

It has also been argued that the use of SRAM PUFs brings about general security advantages in comparison with NVM cells, i.e., even in comparison with systems that do possess NVM. Such claims require further analysis, we believe. It is true that SRAM cells allow to derive a key only whenever it is needed within the hardware. This means that the key is not present permanently in the system in more or less digital form, as in the case of NVMs. On the other hand, invasive or other access to the SRAM cells [96] does allow derivation of the key, just as in the case of NVMs. Furthermore, also cloning of SRAM PUFs has been reported recently [54]. Overall, we recommend that the exact security gains of using SRAM PUFs over NVMs should be analyzed separately and carefully for any given system and application by its users.

One potential drawback of the approach is that SRAM PUFs require error-correcting helper data for deriving a stable key. This needs to be provided either from external, non-volatile memory, or from a trusted third party during a certain protocol. On the other hand, as mentioned earlier, the helper data can be constructed in such a way that it does not leak any information (in an information-theoretic sense) about the key, as long as the power-up states of the SRAM cells are unknown to an attacker. This means that in opposition to to a classical key, the helper data at least does not need to be kept secret.

Variants. Every communication of the above scheme runs over the TTP; as described in [50], this can be resolved by additional protocol steps. We also remark that it is possible to develop other protocols in which the TTP does not have direct access to the IP and SW; interested readers are again referred to [50].

There is a second, important security use of the power-up states of SRAM cells that should not go unmentioned: Holcomb et al. show that those SRAM cells whose

power-up states are unstable (i.e., those whose power-up states flip randomly between zero and one from power-up to power-up) can be used as a hardware-internal random number generator [57, 58].

1.4.3 Remote Identification by Light Scattering in Random Media

Our last example is an identification scheme suggested by Pappu et al. [100, 101] in 2001/02, which rests on optical interference phenomena. At the heart of their method is a transparent, cuboid-shaped plastic platelet of size 1 cm $\times$ 1 cm $\times$ 2.5 mm, in which a large number of micrometer-sized glass spheres have been distributed randomly during the production process. The varying sizes, shapes and positions of the spheres induce strong disorder in the platelet, making it practically infeasible to build two specimen which are exactly the same. The platelet is *"unclonable"* by use of current technology, similar to the examples that we discussed earlier.

When a laser beam is directed at the platelet, the light is scattered multiple times in transition, creating a so-called *"speckle pattern"*, i.e., an interference pattern of dark and bright regions. This pattern can be recorded conveniently by a CCD camera placed behind the platelet. Asides from the relative positioning of the platelet and the camera, which we imagine as fixed and do not consider further here, this speckle pattern sensitively depends on:

(i) the random positions, sizes and shapes of the spheres (i.e., on the disorder inside the token), and

(ii) on the angle α and point $\vec{r}$ of incidence of the laser beam.

The latter can be varied, with each new pair of parameters $(\vec{r}, \alpha)$ leading to new patterns. Leaving aside measurement noise, or assuming perfect error correction, the input-output behavior of the token as a function f that maps measurement parameters $(\vec{r}, \alpha)$ into speckle patterns $f(\vec{r}, \alpha)$. The situation is depicted in Figure 1.2.

The function f has a number of interesting properties. First, f possesses a very large number of input-output pairs $((\vec{r}, \alpha), f(\vec{r}, \alpha))$. In the parlance of the field, these are also called *"challenge-response pairs (CRPs)"*, with $C = (\vec{r}, \alpha)$ being the challenge, and $R = f(\vec{r}, \alpha)$ being the response. Pappu et al. estimate that the above platelet size allows around $2.37 \cdot 10^{10}$ challenges/inputs which lead to computationally independent speckle patterns as responses/outputs. If an adversary has got access to the platelet merely for a limited time period on the order of days or weeks, he will find himself unable to measure all possible input-output pairs and to completely characterize the function f. Secondly, an adversary knowing only a subset of all challenge-response pairs will be unable to numerically predict the speckle pattern to a new, unknown input $\vec{r}, \alpha$ without making a physical measurement on the token. The main reason is that the input-output behavior of the object is too laborious to simulate on a computer. As analyzed by Pappu et al. [101], in the worst case every cubic subunit of the platelet whose side lenth is around the wavelength λ of the incoming laserlight would play a role in the scattering process. For a cube with side length 1 cm, this leads to one Terabit of relevant subunits whose interaction would need to be considered in a simulation, making the latter practically infeasible. Similar considerations, thirdly, hold

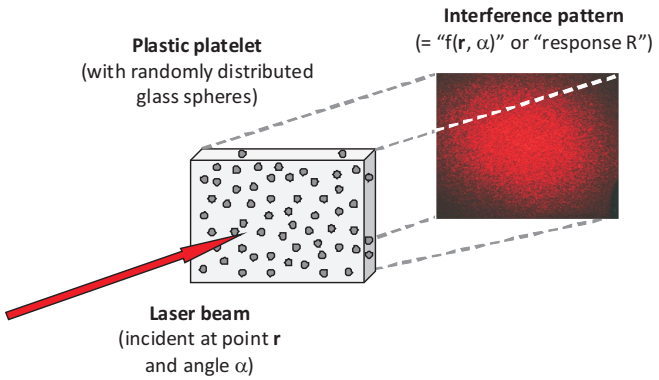

Figure 1.2: Illustration of Pappu's optical, interference-based physical one-way function [100, 101].

for the non-invertibility of f: Given a speckle pattern, it is practically impossible to determine which challenge $C = (\vec{r}, \alpha)$ created this speckle pattern, even if one has access to the token. This non-invertibility property of f originally inspired the name *"physical one-way function"* or *"POWF"* [100, 101] for the above structure; today, it is often refered to as *"optical PUF"*.

How can these properties of f, and of POWFs in general, be exploited in cryptography and security? How can we make use of their unclonability and unpredictability? Perhaps their best known security application are identification protocols, for example in a bank card scenario. In the following protocol, k is the security parameter, and l is the number of envisaged executions of the identification phase.

Protocol 3: BANK CARD IDENTIFICATION WITH LIGHT SCATTERING TOKENS

Set-Up and Security Assumptions:

1. The bank can securely store secret data on some server.
2. Each bank terminal is connected to the bank server by a non-confidential, but authenticated channel.
3. The bank can fabricate light scattering platelets or has access to a trusted manufacturer.

Initialization Phase:

1. The bank fabricates a light scattering platelet or obtains such a platelet from a trusted manufacturer. It attaches it as token to a bank card, which bears the customer identification number ID.

2. The bank chooses at random $k \cdot l$ parameters $\vec{r}_i, \alpha_i$, and applies a laser beam at position $\vec{r}_i$ and under angle α_i to the token. It measures the resulting speckle patterns, and derives from the raw data the error-corrected responses $R_i = f(\vec{r}_i, \alpha_i)$, for example by applying image transformation or error correction.

3. The bank stores the list $\mathcal{L}_{\mathsf{ID}} = (\vec{r}_1, \alpha_1, R_1), \ldots, (\vec{r}_{k \cdot l}, \alpha_{k \cdot l}, R_{k \cdot l})$ together with the identification number ID of the card on its server.

4. The bank card is released to the field.

Identification Phase (can be executed maximally l times):

1. When the card is inserted into a terminal, the terminal reads the ID from the card and sends ID to the server.

2. The server looks up the list $\mathcal{L}_{\mathsf{ID}}$. It chooses the first k entries $(\vec{r}_1, \alpha_1, R_1), \ldots, (\vec{r}_k, \alpha_k, R_k)$ from the list, and sends the parameters $(\vec{r}_1, \alpha_1), \ldots, (\vec{r}_k, \alpha_k)$ to the terminal.

3. The terminal applies laser beams with the incidence coordiantes and angles given by $(\vec{r}_1, \alpha_1), \ldots, (\vec{r}_k, \alpha_k)$ to the token, and measures the corresponding speckle patterns. It derives the responses $R'_1 = f(\vec{r}_1, \alpha_1), \ldots, R'_k = f(\vec{r}_k, \alpha_k)$ from the raw data by applying the same image transformations or error correction as the bank in the set-up phase.

4. The terminal returns $R'_1, \ldots, R'_k$ to the server.

5. The server compares the responses $R_1, \ldots, R_k$ and $R'_1, \ldots, R'_k$. If they match better than given error threshold, the server sends an "OK!" message to the terminal. Otherwise, it sends an abort message.

6. The first k entries are erased from the list $\mathcal{L}_{\mathsf{ID}}$.

Security Discussion. A meaningful discussion requires us to first fix the underlying attack model. It is reasonable to assume that an attacker will be able to physically access the plastic platelet several times between different executions of the identification phase: He could set up faked terminals, or gain possession of the bank card when the customer employs it on other occasions, for example for paying in shops or restaurants, etc. Furthermore, we should suppose that the attacker can eavesdrop the binary communication in the identification protocol and learn the used CRPs $(C_1, R_1), \ldots, (C_k, R_k)$. Under this relatively strong attack model, the security of the scheme is nevertheless upheld by the above properties of the physical one-way function f and of the token. An adversary will be (i) unable to clone the token physically, and (ii) cannot predict the unknown input-output-pairs (or CRPs) numerically, even if he knows a large number of other CRPs. This renders him unable to complete the identification protocol successfully without actual possession of the real token, guaranteeing the security of the above identification method. Interestingly, the scheme does not utilize the one-way property of f, but only the features of unclonability and unpredictability.

Potential Advantages and Drawbacks. Compared to standard identification schemes, Pappu et al.'s method exhibits a few notable advantages. First and foremost, no secret digital keys need to be stored on the bank card. Assuming that the token is too complex to simulate and rebuild, there is indeed no security-critical information at all present on the card whose disclosure would break the security of the system. Even if the adversary knew all positions of the scatterers and all irregularities of the structure, he still could not rebuild or simulate it, since both would be practically infeasible. We can allow him to possess any information in the scattering object without endangering the security of the scheme! This feature is in sharp contrast to any classical techniques, which necessitate that at least some information on the card remains secret. It is also in contrast to some PUF-based techniques, for example the SRAM PUFs presented in Section 1.4.2, where a disclosure of the SRAM power-up states to the adversary breaks the security of the system. Secondly, no potentially laborious numerical identification schemes need to be executed on the card in Pappu et al.'s scheme. The card does not even need to carry integrated circuitry, making it extremely cost effective, at least on the card side. Furthermore, it seems to potentially realize improved securiy against side channel attacks, which circuit implementations of classical identification schemes would be faced with. One last advantage is that the scheme enables remote identification (to the bank headquarters in our case) without circuitry on the card.

The scheme's main drawback is perhaps its mediocre practicality: The apparatus for measuring the speckle pattern (i.e., the response of the optical PUF) is expensive, bulky and potentially error prone. The extra costs for the measurement apparatus can partly eat up the savings from avoiding electrical ciruitry on the card.

Variants. There are a number of variants of the above scheme. Firstly, other hardware systems than the described optical PUF can be employed. Any other so-called Strong PUFs [122] can be used, for example Arbiter PUFs and variants [44, 173], as long as they are secure against modeling [134, 138] and other attacks, for example side channels. Also secure integrated optical PUFs would be an option, preferrably with non-linear scattering materials (compare [126]). Finally, the hardware of optical PUFs can be used for a number of other, more advanced cryptographic protocols, including key exchange [38, 16, 113], bit commitment [100, 16, 98, 123, 125], or oblivious transfer [112, 16, 98].

1.5 Advantages of Disorder-Based Security Hardware

Let us condense and summarize the advantages of disorder-based security hardware. Parts of them have already been addressed throughout the last Section 1.4. We thereby take a pure hardware-centered perspective, ignoring some of the specific cryptographic advantages.

Security Advantage: Better Protection/Avoidance of Keys. One central upside of disorder-based techniques is their approach to cryptographic keys. All techniques of the Section 1.4 avoid the presence of *"classical secret keys"* in vulnerable hardware, i.e.,

the presence of keys that are stored permanently in NVM. Some presented approaches go even one step further, though.

This can be seen most easily if we generalize the notion of a classical secret key: Let us call a *"security-critical information" (SCI)* any information that is present in a piece of hardware at least at one point in time, and whose disclosure to the adversary breaks the security of the system. We may then ask: Do the hardware systems of Section 1.4 contain any *SCI* in the above sense?

The answer differs for the three systems of Section 1.4. To start with, the paper structure of system 1.4.1 does not contain any security-critical information at all. Adversary would be unable to refabricate the complex paper structure, even if they knew it atom by atom. Something similar holds for the optical PUF of Section 1.4.3: It could not be cloned, and its output could not be simulated for complexity reasons, even if the entire structure would be known to the adversary in arbitrary detail. On the other hand, the SRAM PUFs of Section 1.4.2 lead to systems that do contain SCI: The power-up states of the SRAM cells constitute SCI; and so does the internal key obtained from the power-up states after error correction. In this sense, the SRAM PUFs differ from optical PUFs, or from the paper based COAs of Section 1.4.1.

We would like to stress that this distinction is not just academic, but has a direct practical relevance. For example, it eventually enables the invasive attacks on SRAM PUFs that recently have attracted considerable interest [96]. Furthermore, the delays in Arbiter PUFs also represent a form of SCI, a fact that eventually allows modeling attacks on this type of structure (see Chapter 3). In essence, the presence of SCI in a hardware system creates unwanted attack points, and well-versed adversaries will in the end exploit these. Overall, it appears preferable to construct disorder-based systems without SCI wherever possible. We think that it could be useful to pay increasing attention to this distinction in the field, and to categorize disorder-based security approaches with respect to the feature of being *"SCI-free"* or *"key-free"*.

Practicality Advantage: Hardware without NVM or ICs. A second pivotal advantage of the techniques of Section 1.4 is that they enable security features in hardware without NVM. The use of SRAM cells on FPGAs without NVM (Section 1.4.2) is one known example for the former. Partly, they even allow security mechanisms in hardware without integrated circuits (ICs); examples are the presented paper-based COAs and optical PUFs. The exploitation of surface irregularities or optical PUFs (Sections 1.4.1 and 1.4.3) are notable examples here.

Both can lead to decisive practicality and cost assets, and bring security to systems where otherwise elaborate and dedicated security measures would be impossible. Recall in this context that adding non-volatile keys to hardware systems without NVM requires significant additional production steps and extra costs. This gives the above advantages a particular relevance in a commercial context.

1.6 History of the Field and Related Work

Having given a first overview of the field and its methods, we would like to complete the picture in this introductory chapter by providing a historic overview of disorder-based

security and physical unclonable functions below. As patent writings and commercial activities are not in the direct focus of this scientific thesis, we concentrate on academic writings wherever possible. One notable exception are the origins of the field, which actually can be found in the patent sector.

Origins of the Field. It is non-trivial to trace back the field to its exact origins. To the knowledge of the author, the first publicly available source that utilizes random, uncontrollable manufacturing variations in a security context is a US-patent with priority date 1968 by Lindstrom and Schullstrom of Saab AB, Sweden [79]. It suggests that randomly, non-uniformly distributed magnetic materials could be employed for individualizing and securing *"identification documents like driver's licences and credit cards"*. It further proposes that concealed, internal layers of such materials might protect sensitive regions of identification documents, for example the picture of the card holder, against alteration, and could detect manipulation of these regions. The latter foreshadows a security feature that today is called tamper-sensitivity. The inventors also suggest that electrical or optical materials could be used to the same end.

It has also been reported that in the 1970s, Bauder and Simmons of Sandia National Laboratories, USA, exploited the optical behavior of physically disorder media for security purposes [22, 95, 68, 67]. Their goal was to conduct secure weapons inspection during the cold war era. To this end, they reportedly spray-painted epoxy onto nuclear warheads, shed light at it from a certain angle, and recorded pictures of the resulting optical patterns [22, 95, 68, 67]. These images could later be used to re-identify each single warhead in a forgery-proof manner [22, 95, 68, 67]. It seems very likely that this work was conducted independently of Lindstrom's and Schullstrom's approach.

Perhaps the first to combine modern cryptographic methods with physical disorder was Goldman of Light Signatures Inc., USA. In a patent writing with priority date 1980, he details the use of paper irregularities in connection with digital signatures for certifying documents [49]. Light Signatures commercialized this technique in order to authenticate stock certificates in the mid 1980s, but their activities were apparently not profitable and abandoned in 1988 [145]. Presumably independently of Goldman, Bauder (reportedly together with Simmons [22, 68]) suggested a similar concept at Sandia National Laboratories, also combing unique paper structures with digital signatures [67]. A Sandia-internal source that is multiply quoted in this context is by Bauder [7], dating from 1983. [6]

With some right, the three above, independent research groups could be seen as early forefathers of disorder-based security and physical unclonable functions. This would imply that the field has older roots than sometimes acknowledged.

First Presence at Scientific Conferences. The groundbreaking ideas of Lindstrom and Schullstrom, Goldman, and Bauder and Simmons, seemingly were not discussed much in public scientific conferences or journals until the 1990s. Perhaps the earliest scientific paper that points in the relevant direction is by Simmons, dating from 1991

[6] However, copies of this paper seem unavailable to a broad public. The author of this thesis has been unsuccessful in gaining access despite considerable efforts, including multiple e-mail requests to the Sandia National Laboratories. Other researchers made partly similar experiences [68].

[166]. Independently and a few years later, a number of publications by van Renesse treat similar ideas, focusing on optical product protection systems [105, 106]. Suggestions based on magnetic materials, which are in principle related to the early patent of Lindstrom and Schullstrom, have been made independently by Chu et al. [23] and Vaidya [180] at scientific venues in 1995.

In 1998, Haist et al. [52] also discuss paper and optical probing for product protection, making explicit use of digital signatures in a similar fashion as Goldman. A closely related scientific publication is by Smith et al. [167] from 1999. It uses paper irregularities with digital signatures in order to create unforgeable postal stamps.

In 2000, Lofstrom et al. [80] for the first time suggest the variations in standard circuit components for security purposes, exploiting the random threshold mismatches in transistors to identify individual circuits (compare Section 1.4.2). Their paper could be seen as a direct precursor of the modern PUF era, foreshadowing so-callled intrinsic PUFs like SRAM PUFs and Butterfly PUFs (without explicit use of the term "PUF", though).

DNA-based Steganography. Before we eventually turn to PUFs, let us quickly mention another independent research avenue. In 1999, the randomness in complex mixtures of DNA strands was suggested for use in security and steganography by Clelland et al. in Nature magazine [26]. If a secret message or other critical information is encoded in DNA strands, and if these strands are mixed with a huge number of other, random strands, an adversary would find it practically impossible to identify and isolate the "secret" strands. He would be faced with the proverbial search for the needle in the haystack. The fact that complex DNA mixtures can be generated by simple means plays into the hands of this method [26]. In follow-up work, DNA-based public and private key cryptography has been discussed, for example, in [75, 48]. The approach of Clelland et al. has even led to commercially available products [147].

DNA-based security might appear off topic and seems generally less known within the PUF community. Still, it has established its own research strand, with hundreds of citations, some presence at DNA-related venues, and a certain level of commercial activities. Furthermore, it likely is the first approach that explicitly exploits nanoscale phenomena for security. This foreshadows a recently emerging trend towards nanosecurity in the PUF area [129, 59, 109].

Physical Unclonable Functions (PUFs). Despite all above contributions, it seems fair to say that the interest of the broader security community was not sparked until 2001/02, when a few seminal works were published at major scientific venues: Firstly, Pappu in 2001 [100], and Pappu et al. in Science magazine in 2002 [101], presented the idea of so-called *"physical one-way functions"* or *"POWFs"*. Their optical implementation of POWFs (compare Section 1.4.3) has a number of novel features compared to earlier optical security works [49, 105, 106, 52]; for example, it explicitly uses random 3D media and (coherent) laser light as probing signal to facilitate maximally complex response behavior. The second seminal strand of work was by Gassend et al., who published the concept of silicon, circuit-based *"physical random functions"* at ACM CCS 2002 [44], and of *"controlled physical random functions"* at ACSAC 2002 [45].

The latter papers also use the term *"physical unclonable function"* or *"PUF"* for the first time, which today is frequently employed as a synonym for the entire research area (including parts of this thesis).

Compared to earlier works, one central innovation of Pappu et al. and Gassend et al. was to link disordered, unclonable media to more established cryptographic concepts like one-way functions or pseudo-random functions. Secondly, they used disordered media with a very large number of different input signals, whose behavior could be regarded as some sort of complex, disorder-based *"physical function"*. The mathematical properties of this function, such as unpredictability or one-wayness, could then be formally expressed and exploited in cryptographic protocols. Both aspects helped attracting the interest of the cryptographic and security community and spreading the new concepts quickly.

Other seminal PUF works in the early period from 2002 to 2007 include (but are not limited to): The AEGIS security architecture by Suh et al. [172] from 2003; first information-theoretic analyses of PUFs by Tuyls et al. in 2004/05 [178, 179]; the security use of laser illuminated paper surfaces by Buchanan et al. in Nature in 2005 [14] (compare Section 1.4.1); and the usage of disordered electrical structures as tamper sensitive coatings by Tuyls et al. [176] in 2006. A further groundbreaking idea was the use of the individual, but repeatable power-up states of SRAM cells as secret key source. This concept is particularly useful in hardware that does not carry non-volatile memory cells. It was independently put forward by Holcomb et al. [57] and Guajardo et al. [50] in 2007.

Certificates of Authenticity (COAs). Starting a few years later than PUFs, a parallel and independent strand of works helped popularizing the idea of disorder-based security. It roots quite directly in the original ideas of Lindstrom and Schullstrom, Goldman, and Bauder and Simmons, using very similar techniques: It combines the unique and unclonable features of disordered media with digital signatures to form so-called *"certificates of authenticity"* or *"COAs"* for objects of value. Example works of this strand include COA-specific error correction by Kirovski in 2004 [65, 66], optical COAs (which pick up the early ideas of Bauder and Simmons [7]) in by Chen et al. in 2005 [22], and radiowave based COAs by DeJean et al. in 2007 [33]. Also work on unique optical fingerprints of compact discs, which was independently published by DeJean et al. [181] and Hammouri et al. [53] in 2009, could be associated with this research strand. A good summary of the subarea is given by Kirovski in [67]. Most COA papers are somewhat demarked from PUFs in terms of nomenclature and scientific content. Furthermore, the COA-idea arguably dates back earlier than PUFs, being present in its full-fledged form in combination with digital signatures already in the 1980s [49, 7]. We thus found it appropriate to devote a separate paragraph to it. At the same time, we remark that the current focus of the community appears to be on PUFs, both regarding research activities, quotation numbers, and nomenclature.

Status Quo and Current Research. From 2008 onwards, a rapidly growing activity on disorder based security takes place. It is mostly, but not exlusively focused on PUFs, and often regards the two works by Pappu et al. [101] and Gassend et al. [44] as root

publications of the field. Listing all published works of the last six years is beyond the intention and scope of this section; we refer the interested reader to recent survey articles [88, 122] or PUF biobliographies [144].

Rather, we find it convenient to collect several facts that exemplarily testify the rapid establishment of the area. To start with, according to Google scholar, the two root articles of Pappu et al. [101] and Gassend et al. [44] have been quoted many hundred times to date, with increasing citation figures almost every year. Since 2008, papers on PUFs and related topics have been published at CHES [13, 87, 53, 174, 190, 61, 189, 69, 123, 74, 62, 86, 85, 11, 85], EUROCRYPT [98], ASIACRYPT [3, 32], CRYPTO [16], ACM CCS [134, 164], IEEE S&P [4, 124], IEEE T-IFS [31, 83, 138], ACM TISSEC [46], and the Journal of Cryptology [84], i.e., in all top publication channels of the general cryptography and security community. Since 2010, the two large hardware security conferences CHES and HOST continously had one or even two dedicated PUF session each year (see [141, 142]). DATE, one of the two largest international design automation conferences, in 2014 offered both a standard technical session on PUFs [148], a hot topic session on PUFs [149], and a related tutorial on counterfeiting ICs [150], illustrating how PUFs have long spread from their original field of security into neighbouring areas like circuit design. Also the first coursebooks on PUFs have appeared recently [12]. Even on the commercial side, PUFs achieved some recent breakthroughs, appearing in the product lines of major companies like NXP [152, 153] and Microsemi [154, 155]. It therefore seems justified to say that almost 15 years after its popularization in scientific circles [80, 100, 101, 44, 45], and around 45 years since its very first presence in patent writings [79], the field has developed into a central subarea of hardware security, and currently shows no signs of slowing down in its rapid progress.

1.7 Our Contributions

Below, we give a brief overview of the original contributions of this thesis.

Novel Didactic Approach. In Chapters 1 and 2 we develop a view of the field that radically puts physical disorder into the focus. This allows us to easily incorporate disorder-based security primitives other than PUFs, which arose historically before the first PUF publications, or which are to be separated from PUFs for other reasons. It allows a unified treatment of PUFs and related, disorder-based concepts.

PUF Attacks. Initially, PUFs were regarded, or at the least hoped to be, immune against many classical attacks. In recent years, this assumption has been put to the test more closely. Researchers have adjusted classical attacks to PUFs, or developed specific, tailor-made attack methods on PUFs (see [132] for an overview). In this subfield, we make the following contributions:

- Chapter 3 discusses so-called *"modeling attacks"* on Strong PUFs. In this attack form, a small subset of CRPs of a given Strong PUF is collected. Subsequently,

the adversary tries to extrapolate the behavior of the PUF on the entire CRP-space from this subset. If successful, he is able to set up a computer program that correctly simulates the input-output behavior of the PUF. In opposition to the original PUF, such program can be duplicated and distributed arbitrarly. It breaks any protocols and applications that are based on the unclonability and unpredictability of the Strong PUF.

The chapter systematically studies the performance of various, partly tailor-made machine learning techniques to achieve optimal performance, including evolution strategies (ES) and variants of logistic regression (LR). The PUF-types which we successfully attack up to a substantial level of size and complexity include standard Arbiter PUFs, XOR Arbiter PUFs, Lightweight PUFs, Feed-Forward PUFs, and certain variants of the Ring Oscillator PUF.

Modeling attacks currently represent the most effective and popular attack form on so-called Strong PUF. They have not been invented by the author (see [77] for the earliest source), but the presented material currently represents the most advanced and best performing pure modeling attacks in the literature (compare [77, 89, 78, 99, 90]).

- Chapter 4 introduces a new and extremely powerful attack form, namely power and side channels on Strong PUFs that are combined with modeling techniques. The attacked type of Arbiter PUFs and variants thereof are currently the most popular electrical Strong PUF architecture regarding citation numbers [56]. Our techniques represent the first, direct side channels on Strong PUFs that significantly improve attack performance, and also the first power and timing side channels on PUFs in the literature.

  The special asset of our techniques is that they can attack XOR-based Arbiter PUF variants with a merely polynomial attack complexity. We thus are able to demonstrate successful attacks for a bitlength of up to 512 bits and up to 16 XORs for certain XOR-based Arbiter PUF variants. All of our attacks are carried out on real silicon systems, namely FPGAs.

New PUF Designs. In response to the attacks presented in the last paragraph, we develop new approaches to PUF design. Our goal is to conduct foundational research in this area, i.e., to seek genuinely new ideas that either disable the attacks of the last chapters, or which have other substantial advantages over current approaches. Along these lines, we make the following contributions.

- Chapter 5 introduces the novel PUF-class of a *"SHIC PUF"* [7], where the acronym stands for *"Super-High Information Content"*, and suggest a first implementation of this class. SHIC PUFs are "information-theoretically" secure Strong PUFs with a particularly large information content and intrinsically slow read-out speeds. We propose high-capacity crossbar memories for their realization. The necessary high level of disorder in the system is generated by a silicon-based crystallization method known as ALILE process, where the acronym stands for

[7] Pronounce as "chique PUF".

"ALuminum-Induced Layer Exchange". This process also allows the generation of diodes with extremely high rectification rates, which are necessary to enable a functional read-out process in large scale crossbars.

One advantage of SHIC PUFs is that they are naturally immune against the modeling attacks of Chapter 3, as all of their CRPs are information-theoretically independent. Predicting unknown CRPs from known CRPs is logically impossible. Furthermore, it seems hard to imagine successful side channel attacks on SHIC PUFs: There are no straightforward attack points, comparable to the outputs of the single Arbiter PUFs within an XOR-based Arbiter PUF architecture. Since all CRPs are information-theoretically independent, all CRPs have to be read out separately by the adversary; and reading out single CRPs can already be accomplished directly via the SHIC PUF's CRP interface, without side channels. The only usefulness of a side channel could potentially lie in speeding up the read-out process, but given the design and intrinsically slow read-out mechanism of the crossbar architecture, this seems prohibitively difficult.

SHIC PUFs hence represent one possible answer to the problems rised in Chapters 3 and Chapter 4. On the downside, their realization requires a very large area consumption and cutting-edge nanotechnology. In the latter sense, they foreshadow a recent trend in the PUF-community towards nano-security [129, 122, 109, 72].

One last noteworthy aspect is that the high-rectification ALILE-produced diodes mentioned above have been of interest elsewhere, and have led to a parallel publication in Applied Physics Letters of our group [59].

- Chapter 6 suggests that the abovementioned silicon-based crystallization process (i.e., the ALILE-process) can also be used in other contexts than SHIC PUFs. One example are electrically readable certificates of authenticity (COAs), another one Weak PUFs, both of which can be based upon ALILE-fabricated diodes. One notable asset of such diodes is their extremely high fabrication variation, which reaches difference factors of up to 10^4 in their current-voltage curves. To the knowledge of the author, this superseeds any other known silicon-based fabrication process suggested in the context of PUFs to this date. Relative to their immense variation, ALILE-fabricated structures also show very good temperature stability, as has been confirmed in recent follow-up experiments at the TU München [108].

  Crystallization processes have one decisive asset in the context of PUF fabrication: At the beginning of the process, small nuclei form at positions that are determined by random, atomic roughness in the substrate. The crystallites then start to grow around these nuclei. This growth process initially *increases* the local differences in the substrate, and does *not* statistically obliterate them. This is in opposition to other statistical production processes. Our work therefore points to a new technique for generating electrical PUFs with extremely high variation, and can be seen as foundational research contribution in the PUF area.

Disorder-based Security Hardware Beyond PUFs. In the last part of the thesis, we make a step beyond PUFs, and introduce and discuss a hardware primitive that could be seen as a public key version of PUFs: So-called *"SIMPL systems"* (or just *"SIMPLs"*), where the acronym SIMPL stands for SIMulation Possible, but Laborious. SIMPLs possess a publicly known numeric description of their internal, disordered structure. This description which allows everyone to numerically simulate their challenge-response behavior, albeit with some time loss compared to the physical, real-time behavior of the SIMPL system. While everyone can thus numerically and slowly simulate a given SIMPL system, only the physical holder of the system can obtain the responses faster than a certain threshold. This allows various, public-key like protocols.

SIMPL systems possess a number of other advantages beyond their public-key functionality. One is that they allow the construction of *"keyless"* cryptographic hardware. This is hardware which does not contain or store any "security-critical" information, whose disclosure would allow the adversary to break the system. This feature constitutes a strong novelty of SIMPLs compared to electrical PUFs: For example to SRAM PUFs, where the disclosure of the cells' power up states breaks security, or to Arbiter PUFs, where the disclosure of the internal runtime delays makes the system simulatable. We stress that completely independently and around the same time, an equivalent concept was developed under the name of a *"Public PUF (PPUF)"* by Beckmann and Potkonjak [8]. [8]

- In Chapter 7, the basic concept of a SIMPL system is introducted. We discuss their fundamental properties, protocols and implementations. The chapter serves as a nice introduction to and overview of the matter.

- Chapter 8 takes a more detailed look at two implementation suggestions for circuit-based SIMPL systems. Both of them exploit two well-known bottlenecks of electrical integrated circuits (ICs) for achieving the required speed advantage of the SIMPL: Our first suggestion exploits the adversarial restrictions of increasing clock frequencies indefinitely in order to construct SRAM-based SIMPL systems. Our second suggestion is built on the inability of current IC architectures for massively parallel, analog computations, proposing cellular non-linear networks (also known as CNNs) for the construction of SIMPL systems.

- Chapter 9 treats in detail the usability of SIMPLs in cryptographic protocols, including bit commitment, coin tossing, identification, authentication, and key exchange. Also protocols of other groups are discussed and analyzed for their practical viability, for example Beckmann and Potkonjak's key exchange schemes

[8] Let us quickly compare the history of both concepts: The earliest documented source for SIMPL systems is a patent filed by the TU München and U. Rührmair with priority date March 16, 2009 [171]; a second work is an eprint publication from June 1, 2009 by Rührmair [111]. The first documented source of public PUFs to the best knowledge of the author is the conference publication of Beckmann and Potkonjak from June 8, 2009 [8], which was submitted to the conference as early as February 1, 2009; a second source is a patent by M. Potkonjak with priority date June 17, 2009 [103]. Earlier papers both on SIMPLs and PPUFs had been submitted to major conferences, but were rejected [104]. Overall, it seems fair to say that both concepts were developed completely independently and unaware of each other, with origins in both cases tracing back to as early as 2005 [104].

based on PPUFs [8]. This general analysis on the potential of SIMPLs concludes the thesis.

1.8 Formalities and Organization of this Thesis

Formalities. In the sense of the doctoral dissertation statutes of the TU München, this thesis is a so-called *"publication-based dissertation"* (also known as "cumulative dissertation"). It consists of a mandatory introductory part, followed by the cumulative inclusion of several publications, in which the candidate is the lead author. According to the statutes, each publication must be accompanied by a short summary, which subsumes the content of the publication and outlines its role within the thesis. The candidate's contributions to each publication must be detailed, and a list of all publications of the candidate should be given. To guarantee a fluent exposition, the latter two requirements are fulfilled in Appendices A and B. The one-page summaries, on the other hand, are provided directly at the beginning of each chapter. Among other things, this gives the thesis a modular character, allowing readers to easily jump between different chapters.

We stress that there is a sister dissertation by the author at the TU Berlin, which focuses on theoretical aspects of PUFs and related primitives. The focus on hardware, including hardware security and new disorder-based hardware concepts in thesis, and the concentration of the sister thesis on formal and foundational aspects of PUFs, formally distinguish these two works. Both cumulative theses use a completely disjoint set of publications.

Organization of this Thesis. The structure of this thesis is orientated towards two of the main classes of disorder-based security hardware: Physical unclonable functions (PUFs) are treated in Part II and SIMPL systems, which are a public key version of PUFs, in Part III.

Part II consists of four chapters, treating various security and implementation aspects. We start in Chapter 2 by an easily accessible overview of PUFs, which provides readers with the necessary background knowledge to assess the upcoming chapters. We then move on to one of our central topics within this thesis, namely to PUF attacks. The historically earliest attack and currently most cost effective attack form on so-called Strong PUFs, namely modeling attacks, is treated in Chapter 3. In the subsequent Chapter 4, we show how the substantial reach of modeling attacks can be improved yet further by using power and timing side channels. This drastically improves the attack complexity on XOR-based Arbiter PUFs to low-degree polynomial. In Chapters 5 and 6, we present alternative approaches to the design and implementation of PUFs: Chapter 5 introduces so-called SHIC PUFs, a new PUF class, which are implemented by nano-scale crossbar structures. Chapter 6 discusses applications of a silicon-based crystallization method known as ALILE-process for the construction of COAs and of Weak PUFs or physically obfuscated keys (POKs).

Part III puts forward a generalization and public key version of PUFs, so-called *"SIMPL systems"*. It consists of three chapters: Chapter 7 introduces the concept of SIMPL systems and gives an overview of their basic features, implementations

and applications. Chapter 8 puts the focus on hardware implementations of SIMPLs, discussing two possible implementations in greated detail. Finally, chapter 9 takes a broader perspective on SIMPLs, and discusses their applicability within cryptographic protocols, concluding this thesis.

Finally: Enjoy!

Part II

Physical Unclonable Functions (PUFs)

Chapter 2

PUFs at a Glance

As detailed in Chapter 1, PUFs have been introduced as a new security primitive around the year 2000 [80, 100, 101, 45, 44], with the roots of the field in patent writings tracing back as early as to the late 1960s [79]. Since 2002, a broad and widespread publication activity has taken place, dealing with the implementation, application, and formalization of different PUFs and PUF-like primitives.

This chapter summarizes these developments, thereby providing an easily accessible, high-level introduction to PUFs. It thereby explicitly distinguishes two central PUF types, so-called Weak PUFs and Strong PUFs, and takes readers on a tour through the characteristic features, implementations, applications, error correction mechanisms, attacks, and conceptual histories of these types. To remain easily accessible, it avoids technical details, rather focusing on the main concepts and ideas in the field. At the same time, it takes great care to provide comprehensive literature pointers for readers whose interest in a certain sub-topic was sparked, and who now want to investigate this topic more closely. The chapter also preludes some of the subjects of the upcoming chapters. It thus prepares its readers well for the following parts, making an ideal first chapter of the thesis.

The actual publication that we use in this chapter is

- U. Rührmair, D.E. Holcomb: *PUFs at a Glance*. Design, Automation & Test in Europe (DATE 2014), pp. 1-6, 2014.

It was part of the special session *"How secure are PUFs really? On the reach and limits of recent PUFs attacks"* [132], which was organized by the candidate. For a description of this session, interested readers are referred to the following article, which is explicitly not contained in this cumulative thesis:

- U. Rührmair, U. Schlichtmann, W. Burleson: *Special Session: How Secure are PUFs Really? On the Reach and Limits of Recent PUF Attacks*. Design, Automation & Test in Europe (DATE 2014), pp. 1-6, 2014.

PUFs at a Glance

Ulrich Rührmair
Technische Universität München
80333 München, Germany
E-mail: ruehrmair@ilo.de

Daniel E. Holcomb
University of Michigan
Ann Arbor, MI 48109, USA
E-mail: danholcomb@umich.edu

***Abstract*—Physical Unclonable Functions (PUFs) are a new, hardware-based security primitive, which has been introduced just about a decade ago. In this paper, we provide a brief and easily accessible overview of the area. We describe the typical security features, implementations, attacks, protocols uses, and applications of PUFs. Special focus is placed on the two most prominent PUF types, so-called *"Weak PUFs"* and *"Strong PUFs"*, and their mutual differences.**

***Keywords*—Physical Unclonable Functions, Overview, Survey, Weak PUFs, Strong PUFs**

I. Introduction

A. Motivation and Background

Electronic devices are pervasive in our everyday life. This leads to a host of security and privacy issues. Classical cryptography offers several measures against these problems, but they all rest on the concept of a secret binary key: It is assumed that the devices can permanently store a piece of digital information that is, and remains, unknown to the adversary. Unfortunately, this requirement can be quite difficult to uphold in practice. Physical attacks such as invasive, semi-invasive, or side-channel attacks, as well as software attacks like API-attacks and viruses, can lead to key exposure and security breaks [1]. One additional complication lies in the fact that the employed devices should ideally be lightweight and cost efficient, and are resource-constrained in certain commercial scenarios. For example, some security systems will not contain non-volatile memory cells due to their extra costs. This poses the question: How can medium or even high security levels be achieved in such circumstances?

B. PUFs, Role of Manufacturing Variations, Challenge-Response Formalism

The described situation was one motivation that led to the development of *physical unclonable functions (PUFs)*. Their key idea is to exploit the *"random physical disorder"* or the *"manufacturing variations"* that occur in almost all physical systems on small length scales. The shown disorder typically cannot be fully controlled during the fabrication of the system, and cannot be re-fabricated intentionally, not even by the original manufacturer. It is *unclonable*, and constitutes an individual fingerprint of each system. Usually, this phenomenon is regarded as disadvantageous, for example in the context of semiconductor fabrication. Integrated circuits commonly have to be designed in such a way that their digital behavior remains unaffected by manufacturing variations. PUFs, however, turn said variations into an advantage, and explicitly exploit them for security purposes.

More specifically, a PUF is an (at least partly) disordered physical system that can be challenged with external stimuli or so-called *"challenges"* C_i. Depending on the exact PUF type, a PUF can thereby have merely one single possible challenge, a few challenges, or even an exponential number of challenges in some system parameter (see Sections II and III). Upon being exposed to a challenge C_i, the PUF reacts by producing a corresponding response R_i. The tuples (C_i, R_i) are thereby termed the challenge-response pairs (CRPs) of the PUF.

PUFs are deliberately designed such that the response(s) R_i depend on the individual physical disorder present in the PUF. Each PUF response is hence not only a function of the applied challenge C_i, but also of the PUF's physical disorder. One consequence is that the challenge-response behavior varies between different "physical instances" or "specimen" of the same PUF, since any instance is subject to different manufacturing variations. From an abstract perspective, one could say a PUF's challenge-response mechanism converts the unique *physical* disorder of the PUF into *digital* input-output data. While the exact challenge-response mechanisms vary, most existing electrical PUFs thereby produce responses that consist of exactly one bit. If necessary, several such single-bit responses may be bundled to obtain a multi-bit identifier/key.

C. Inevitable Error Correction

Since PUF responses are based on very small manufacturing variations, PUFs usually operate more closely at their stability limits than classical, digital systems. This renders numeric error correction vital. Two basic approaches exist.

Firstly, standard error correction mechanisms may be applied to each PUF response, converting it into a stable, noise-free output. Usually this error correction is accomplished via some so-called *"helper data"*. The latter assists in the error correction process of a given response, and has been derived upon an earlier measurement of this response. The helper data must be stored in some non-volatile memory (NVM) accompanying the PUF, but not necessarily inside the PUF-carrying system (which may not be equipped with an NVM). It can be constructed in such a way that it can become known to an adversary without compromising the secrecy of the PUF response, i.e., it does not need to be kept secret [18].

978-3-9815370-2-4/DATE14/©2014 EDAA
This work was supported in part by C-FAR, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and NSF CNS-0845874. Any opinions, findings, conclusions, and recommendations expressed in these materials are those of the authors and do not necessarily reflect the views of the sponsors.

A second, less widespread possibility for error correction is the design of PUF protocols with inbuilt error tolerances. This circumvents the need for perfect error correction inside the PUF-carrying hardware, enabling more lightweight systems. One example for this method is the well-known Strong PUF identification protocol of Pappu et al. [52], [53] that we discuss in Section III-C. We remark in passing that there are notable differences between different PUF types regarding error correction, which will be discussed in Sections II-C and III-C.

D. Aspired Advantages and Some Applications

There are two benefits that users would like to gain from PUFs: Security advantages and certain forms of cost/practicality upsides. These assumed benefits have acted as drivers for PUF research in the past.

Let us start with security aspects. Due to its complex and disordered structure, a PUF can avoid some of the shortcomings associated with digital keys. For example, it is usually harder to read out, predict, or derive its responses than to obtain the values of digital keys that are permanently stored in non-volatile memory. The PUF-responses are only derived when needed, meaning that they are present in the security system in an easily accessible digital form only for very short time periods. Furthermore, many PUFs have been assumed to be tamper sensitive, meaning that invasive attacks would alter the PUF's response behavior permanently and notably. These facts have been exploited for various PUF-based security protocols. Prominent examples include schemes for identification and authentication [53], [22], key storage [76], [25], key exchange [15], [8], or digital rights management purposes [23].

On the cost/practicality side, PUFs allow the "storage" of keys in hardware systems that do not have NVM. One prominent example are FPGAs, where SRAM-based PUFs have been suggested to derive a key on the FPGA. This key can be used, for example, to encrypt/decrypt the design bitstream that is uploaded onto the FPGA [25], since this design may represent a substantial intellectual worth. Similar aspects hold for other systems without NVMs, in which PUFs can be used as an identifier or as key source.

II. Weak PUFs

The two most important subtypes of PUFs, which should be distinguished explicitly in any sound treatment of the topic, are so-called *"Weak PUFs"* and *"Strong PUFs"*. They are discussed in this and the upcoming section.

A. Characteristic Features of Weak PUFs

Weak PUFs essentially are a new form of storing secret keys in vulnerable hardware, offering an alternative to ROM, Flash or other non-volatile memories (NVMs). As all PUFs, Weak PUFs exhibit some internal, unclonable physical disorder, and possess some form of challenge-response mechanism that exploits this disorder (see Section I-B). Beyond this, their characteristic features are as follows (compare [67], [60], [2]:

1) *Few challenges:* A Weak PUF has got very few, fixed challenges, commonly only one challenge per PUF instance.
2) *Access-restricted responses:* In all but very few applications, the challenge-response interface (or the challenge-response mechanism, respectively) of a Weak PUF needs to be access-restricted. It is assumed that adversaries cannot access to the Weak PUF's responses, even if they hold physical possession of the PUF-carrying hardware.

Both features fundamentally distinguish Weak PUFs from Strong PUFs (compare Section III-A).

B. Implementation Examples

Weak PUFs can be implemented either using special purpose integrated circuits designed to be sensitive to variation, or by using the intrinsic variation present in all existing circuits. While some of the first Weak PUFs were based on special purpose circuits, the recent trend is toward instrinsic PUFs fabricated from standard CMOS logic parts, since this is more cost effective. One of the earliest Weak PUFs was a design proposed in 2000 by Lofstrom et al. [42] to leverage threshold mismatch for identifying circuits. A more involved PUF based on sensing the capacitance of specially applied protective coatings was given by Tuyls et al. [76]. Later, Su et al. [74] demonstrated a chip-ID circuit based on cross-coupled devices; to evaluate the ID, the cross coupled devices are brought to a metastable state, and then allowed to spontaneously transition to a stable state determined by process variation. In noting that their design is SRAM-like, the authors of this work foreshadow the subsequent trend of SRAM-based Weak PUFs.

The most popular implementation of intrinsic Weak PUFs are SRAM PUFs. They exploit the inherent threshold variation of the cross-coupled SRAM cells. The differential nature of the cells make them ideal for being sensitive to variation and also largely immune to common-mode noise. Furthermore, the ubiquity of SRAM in nearly all VLSI circuits gives them wide applicability as PUFs. The physical identifier is automatically generated in the cell whenever it goes from an un-powered state to a powered state, and the identifier can then be read out using the standard memory access mechanism. The earliest known mention of this phenomenon is in a patent by Layman et al [38]. The phenomenon of SRAM signatures nevertheless remained unknown to the wider research community until being later rediscovered in 2007 independenly and concurrently by Holcomb et al. [31] and Guajardo et al. [25]. An alternative formulation of an SRAM-based Weak PUFs uses the minimum data retention voltage of cells instead of the power-up state [33]. Subsequent to SRAM PUFs, a variety of other intrinsic Weak PUFs have been proposed. Memory-based PUFs are suggested for storage technologies including Flash [54], Memristors [35], and DRAM [56]. Intrinsic non-memory PUFs are also proposed, including the butterfly PUF [36] that uses cross-coupled latches in FPGAs, and a PUF based on bus-keepers [72].

All of the above PUFs have one fixed way to excite them (for example powering them up), and hence exactly one challenge.

C. Applications and Error Correction

The main application of Weak PUFs is to derive secret keys inside (lightweight) hardware systems. In principle, one

can distinguish two basic cases.

The first and by far most popular case is the derivation of an internal, but *shared* secret key from Weak PUF responses, which is known to a limited number of parties outside the PUF-carrying hardware — usually only to the manufacturer of the PUF. This approach pressumes that the manufacturer learns the key, for example by directly accessing the Weak PUF responses, in a secure set-up phase. At the same time, one commonly assumes that adversaries with physical access to the Weak PUF carrying hardware cannot access the PUF's responses, or learn the key, after the set-up phase. Even though these two assumption live in some tension, they have never been put too much in question in the general Weak PUF literature. In practice, they may be realized by disabling access to the PUF after the secure set-up phase in one way or the other. The internal, shared key can then used for any classical secret key based application. One exemplary and commercially attractive application was already named in Section I-D: Encryption of the design bitstream that is uploaded onto FPGAs. The second, but far less popular basic case is the derivation of an internal, *unshared* key that is unknown to any party outside the PUF-carrying hardware. This case partly releaves the abovementioned tension, since the key never must leave the system. It can remain forever inside a PUF-carrying, tamper-sensitive hardware, for example a hardware that is surrounded by a PUF-like coating [76]. One straightforward use for such an internal, unshared key is memory encryption [76].

We stress that in any non-trivial applications of Weak PUFs, perfect error correction in the derived secret keys neeeds to be achieved. Since the secret key is never released to the outside (after the set-up phase), this error correction must be carried out internally, requiring suitable resources in the Weak PUF carrying hardware. Several different approaches have been developed to this end, including [7], [43], [81], [29], [82], to which we refer the reader.

D. Attacks on Weak PUFs

If the digital responses arising from a Weak PUF are read out by invasive means, the security of the system is compromised. This is in principle comparable to the security of a secret key stored in NVM, even though the PUF-response exists in the system only for a short time. Still, this inherent attack point of Weak PUFs has been successfully exploited in recent publications by Nedospasov et al. [48]. Even if care is taken to prevent SRAM PUF values from ever being read over standard on-chip channels, attacks using laser stimulation can reveal cell states in a powered SRAM PUF [48].

Also cloning attacks have been suggested lately. One key observation is that not the entire PUF needs to be cloned in full detail; it suffices if the clone has the same challenge-response pair(s) as the original. Since Weak PUFs often have only one CRP, the clone only has to be tuned until this single CRP matches the original. It had been known for some time that the identifying tendencies of SRAM cells can indeed be shifted by directed aging using NBTI or other means [32], [5], [30]. Originally, this effect has been suggested to make the outputs of SRAM PUFs more stable. It can also be exploited by an adversary, though: In an invasive attack, he reprograms the tendency of a cell using focused ion beam circuit edit, thus effectively cloning the CRP behavior of the SRAM PUF [27].

E. History of Concept and Terminology

Historically, the concept denoted as Weak PUF in this work has been called by at least one different term: Gassend proposed the use of PUFs with a small number of fixed challenges as an internal key source under the name of a *"physically obfuscated key" (POK)* in 2003 [21]. In 2007, Guajardo et al. [25] were apparently the first to use the terms Weak and Strong PUFs, but without differentiating these two concepts in full detail. Rührmair et al. contributed to a more detailed distinction in 2009 to 2012 [69], [67], [60]. An attempt at formalizing Weak PUFs is given in 2011 by Armknecht et al. [2], who define a weak PUF as one that can be modeled from a number of challenge response pairs that fails to be exponential in any security parameter.

III. Strong PUFs

A. Characteristic Features of Strong PUFs

So-called *"Strong PUFs"* are the second major PUF type besides Weak PUFs. In opposition to the latter, they derive a more complex challenge-response behavior from the physical disorder present in the PUF. Typically, many physical components are involved in the generation of a response, and there is a very large number of possible challlenges that can be applied to the PUF. Their security features have been put down in [69], [67], [60], [70], and, more formally, in [59], [8]. In a nutshell, they can be subsumed as follows:

1) *Many challenges:* Strong PUFs have a very large number of possible challenges, ideally (but not necessarily) exponentially many challenges in some system parameter. This prevents a full read-out of all CRPs, even if an adversary holds physical possession of the PUF for considerable time.
2) *Unpredictability:* Even if an adversary knows a large subset of CRPs, he cannot extrapolate or predict the other, yet unknown CRPs.
3) *Unprotected challenge-response interface:* In all but very few applications of Strong PUFs, it is assumed that have a freely, publicly accessible challenge-response interface (or a freely accessible challenge-respons mechanism, respectively). Anyone holding physical possession of the PUF or the PUF-carrying hardware can apply arbitrary challenges to the Strong PUF and read out the corresponding responses.

Please note that all three features mark clear differences to Weak PUFs. Since the challenge-response interface of a Strong PUF is in most applications is assumed to be unprotected, no access restrictions on the PUF-responses need to be supposed. Recall from Sections II and II-D that the latter were one of the most critical assumptions in the security features of Weak PUFs. Invasive attacks on the PUF responses are therefore mostly obsolete for Strong PUFs [1]. On the other hand, the freely accessible challenge-response interface also brings about downsides: It necessarily implies that Strong PUFs must have very many CRPs to remain secure. It also enables modeling attacks on Strong PUFs, since it allows the simple collection

[1]The only exceptions are invasive attacks on internal digital signals inside the Strong PUFs itself, if such signals exist in a given Strong PUF design. Examples are XOR-based Arbiter PUFs [75], [67] or Lightweight PUFs [46].

of large subsets of CRPs. The latter attacks are irrelevant for Weak PUFs, in turn (see Sections II-D and III-D).

B. Implementation Examples

The first proposed Strong PUF is the optical PUF of Pappu et al. [53]. It consists of an optical scattering object, for example a plastic token which contains randomly distributed glass spheres. The challenge to the structure is a laser beam which is directed at the token under a selected angle and point of incidence. The resulting response is the multi-bit intereference pattern that emerges from the complex light scattering process inside the token. Pappu et al. estimate that their implementation of an optical PUF creates around 10^{10} independent CRPs [53].

The first electrical, integrated Strong PUF is the so-called Arbiter PUF [22], [75]. Its idea is to exploit the varying runtime delays in electrical components. In an Arbiter PUF architecture, electrical signals race against each other through a sequence of k stages, each of which consists of two multiplexers. The exact race path of each signal is determined by k external bits which are applied at the stages, one bit per stage. The race is called by a final arbiter element, which is implemented by a latch. Arbiter PUFs with k stages have 2^k challenges, and produce one-bit responses. Since the plain Arbiter PUF is susceptible to machine-learning based modeling attacks (see Section III-D), more sophisticated variants have been developed. They have in common that they add non-linearities in one way or the other to the standard Arbiter PUF to complicate machine learning. Examples include Feed-Forward Arbiter PUFs [39], [40], XOR Arbiter PUFs [75], [67], and the so-called Lightweight PUF [46].

Moving away from the somewhat dominant Arbiter PUF family, several alternative electrical Strong PUF designs exist, to which we would like to point interested readers: The Power Grid PUF [28]; Clock PUF [80]; Crossbar PUF [65]; and the CNN PUF, which is based on analog circuits [11], [3].

C. Applications and Error Correction

The prime application of Strong PUFs is challenge-response based identification and system authentication. The idea has been first described in a banking card scenario [53] as follows. It is assumed that the bank equips each banking card with a Strong PUF. Before the card is released to the customer, the bank applies a large number of random challenges to the PUF, and stores the resulting CRPs in a secret, internal list $\mathcal{L}$. When the card is carried by the customer to a terminal or automated teller machine, the card can identify itself by using the unique challenge-response behavior of the PUF: The bank chooses a couple of challenges from the list $\mathcal{L}$, and sends them to the terminal. The terminal applies the challenges to the Strong PUF, and returns the obtained responses to the bank. The latter compares them to the responses in the list $\mathcal{L}$; if they match, the identification was successful. Each CRP can be used only once and needs to be erased from the list subsequently.

The above identification protocol has the advantage of being extremely lightweight, and of requiring *no* resources besides the PUF on the card. Standard PUF error correction can potentially be executed outside the card (i.e., the PUF-carrying hardware), for example by the terminal or the bank itself. The protocol can also be made error tolerant by allowing a small fraction of all responses to be incorrect; in this case, no classical error correction needs to be applied at all. This constitutes an advantage compared to Weak PUFs, where perfect error correction must be accomplished inside the PUF-carrying system, making the approach less lightweight. Note that the protocol explicitly requires a Strong PUF: Since a Weak PUF only has got one (or very few) digital responses, it could be utilized in one protocol execution only.

The above protocol can be applied in any system identification scenario, and shines the most for inexpensive, lightweight systems. It can be used commercially for any forms of online identification or certification (compare [78]).

Strong PUFs have also been suggested in cryptographic applications beyond the above, basic identification scheme. Already Pappu considers a simple bit-commitment protocol that rests on the onewayness (non-invertibility) of the CRPs of his optical PUF in 2002 [53]. Van Dijk suggested a key exchange protocol based on Strong PUFs in a patent writing in 2004 [15]. The usability of Strong PUFs as a universal primitive was first demonstrated by Rührmair in 2010, who showed that oblivious transfer (and hence also any secure multi-party computatuion) can be based on Strong PUFs [61]. In 2011, Brzuska et al. [8] treated PUFs in the UC-model and lead formal proofs for the security of Strong PUF based bit commitment, oblivious transfer and key exchange. We stress, however, that the secure commercial use of plain Strong PUFs in these advanced protocols is currently under heavy research, after a number of dedicated protocol attacks has been discovered recently [62], [63], [16], [64], [17].

D. Attacks on Strong PUFs

Cloning and invasive attacks on Weak PUFs (Section II-D) appear less applicable to Strong PUFs for a number of reasons. Rather, the currently most relevant attack method for Strong PUFs are so-called *"modeling attacks"* [40], [45], [67], [70]. They assume that an adversary has collected a large number of all possible CRPs of a given Strong PUF (usually between several hundred to a few million CRPs, depending on the exact Strong PUF design). By use of numeric methods and an internal, parametric model of the PUF, the adversary then tries to extrapolate the behavior of the PUF on the other, yet unknown CRPs. Machine learning algorithms are a natural and very powerful tool to this end.

The reach of modeling attacks is surprisingly large, and a considerable number of existing electrical designs have been tackled successfully up to a certain size, including Arbiter PUFs and variants thereof [67], [70]. Only optical PUFs have resisted all modeling attacks so far. We refer the reader to existing works [67], [70] and a recent survey paper on modeling attacks [68]. Modeling attacks do not apply to Weak PUFs, since the latter have only one challenge per PUF. Therefore no extrapolation of unknown CRPs from a subset of known CRPs is applicable. One very recent trend is to combine modeling techniques with side channel information in order to boost attack performance [13], [44].

Also dedicated protocol attacks on Strong PUF schemes have been discovered recently. They differ from the above, hardware-oriented modeling attempts. We refer the interested

reader to the existing literature on this topic [62], [63], [16], [64], [50] and a recent survey paper [17].

E. History of Concept and Terminology

Historically, the structures that we call Strong PUFs today have been referred to by different names. The first Strong PUF in our sense is the optical PUF of Pappu et al. [52], [53] from 2001/02. Its input-output behavior is not just unpredictable, but also non-invertible, whence the authors originally used the term *"physical one-way function" (POWF)* for their invention. Still in 2002, Gassend et al. [22] introduced circuit-based Strong PUFs, using the names *"physical random function"* and *"physical unclonable function (PUF)"*. The term Strong PUF was then eventually suggested by Guajardo et al. [25] in 2007, but without fully detailing all its features. Rührmair et al. worked out the exact security features and the associated attack models in 2009 to 2012 [69], [67], [60]. Formal, mathematical definitions of Strong PUFs have been given by Rührmair et al. [59] in 2010 and Brzuska et al. [8] in 2011.

IV. Summary and Outlook

This survey paper presented an overview of PUFs and their applications as security primitives. The distinguishing feature of PUFs in contrast to more traditional methods is that their outputs are influenced by the random variations arising during fabrication. This new approach brings about some cost/practicality and also security upsides: PUFs allow the "storage" of keys in hardware without non-volatile memory cells, and their complex behavior promises better security against attacks. On the downside, they are generally more prone to errors and aging than classical approaches. Their intrinsically high noise levels must be compensated by dedicated error correction or protocol measures.

The two main types of PUFs are denoted Weak PUFs and Strong PUFs. Each have a variety of implementations: SRAM PUFs and variants are the most popular Weak PUF designs, while Arbiter PUFs and variants are the best investigated electrical Strong PUF architectures. The two PUF types serve distinct purposes; a Weak PUF is akin to a secret key, whereas a Strong PUF is more like a physical hash function. After almost 15 years of existence, PUFs show no signs of slowing down as a research topic, and today both Weak and Strong PUFs are already commercially available as products. The commercial and academic perspectives of the field hence appear bright.

References

[1] Ross Anderson: *Security engineering*. Wiley, 2008.

[2] F. Armknecht, R. Maes, A.-R. Sadeghi, F.-X. Standaert, C. Wachsmann: *A Formalization of the Security Features of Physical Functions*. IEEE Symposium on Security and Privacy, 2011.

[3] T. Addabbo, A. Fort, M. Di Marco, L. Pancioni, and V. Vignoli: *A 1-bit Physically Unclonable Function based on a two-neurons CNN*. IEEE International Symposium on Circuits and Systems (ISCAS'13), 2013.

[4] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, USA, 1996.

[5] M. Bhargava, C. Cakir, and K. Mai. Reliability enhancement of bistable PUFs in 65nm bulk CMOS. *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, pages 25–30, 2012.

[6] C.M. Bishop et al. *Pattern recognition and machine learning*. Springer New York:, 2006.

[7] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and Pim Tuyls. Efficient Helper Data Key Extractor on FPGAs. In Cryptographic Hardware and Embedded Systems, pages 181–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

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

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

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

[11] G. Csaba, X. Ju, Z. Ma, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, and U. Rührmair. Application of mismatched cellular nonlinear networks for physical cryptography. *IEEE CNNA*, 2010.

[12] I. Damgard, A. Scafuro: Unconditionally Secure and Universally Composable Commitments from Physical Assumptions. *Cryptology ePrint Archive, 2013:108*, 2013.

[13] J. Delvaux, I. Verbauwhede: *Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise*. HOST 2013.

[14] S. Devadas: Physical unclonable functions and secure processors. *Invited Talk, CHES 2009*.

[15] M. van Dijk: *System and method of reliable forward secret key sharing with physical random functions.* US Patent No. 7,653,197, October 2004.

[16] M. van Dijk, U. Rührmair: *Physical Unclonable Functions in Cryptographic Protocols: Security Proofs and Impossibility Results*. IACR Cryptology ePrint Archive 2012: 228 (2012)

[17] M. van Dijk, U. Rührmair: *Protocol Attacks on Advanced PUF Protocols and Countermeasures*. Design, Automation and Test in Europe (DATE'14), 2014.

[18] Y. Dodis, R. Ostrovsky, L. Reyzin, L., A. Smith: *Fuzzy extractors: How to generate strong keys from biometrics and other noisy data*. SIAM Journal on Computing, 38(1), 97-139, 2008.

[19] http://electroiq.com/chipworks_real_chips_blog/2011/03/13/apples-a5-processor-is-by-samsung-not-tsmc/

[20] https://freedom-to-tinker.com/blog/felten/fingerprinting-blank-paper-using-commodity-scanners/

[21] B. L. P. Gassend. *Physical random functions*. MSc thesis, MIT, 2003.

[22] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas. Silicon physical random functions. *ACM CCS 2002*.

[23] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Controlled physical random functions. *ACSAC 2002*.

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

[25] J. Guajardo, S. Kumar, G.J. Schrijen, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection. *CHES 2007*.

[26] C. Helfmeier, D. Nedospasov, S. Tajik, C. Boit, J.-P. Seifert: Physical Vulnerabilities of Physically Unclonable Functions. *DATE'14*.

[27] C. Helfmeier, C. Boit, D. Nedospasov, J.P. Seifert: Cloning Physically Unclonable Functions. *HOST 2013*.

[28] R. Helinski, D. Acharyya, and J. Plusquellic: A physical unclonable function defined using power distribution system equivalent resistance variations. In *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, pages 676–681. 2009.

[29] M. Hiller, D. Merli, F. Stumpf, and G. Sigl. Complementary IBS: Application specific error correction for PUFs. *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, pages 1–6, 2012.

[30] M. Hofer, C. Boehm: *An alternative to error correction for SRAM-like PUFs.* Cryptographic Hardware and Embedded Systems (CHES'10), 2010.

[31] D.E. Holcomb, W.P. Burleson, and K. Fu. Initial SRAM state as

a fingerprint and source of true random numbers for RFID tags. *Conference on RFID Security*, 2007.

[32] D. E. Holcomb, W. P. Burleson, and K. Fu. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 2009.

[33] D. E. Holcomb, A. Rahmati, M. Salajegheh, W. P. Burleson, and K. Fu. DRV-Fingerprinting: using data retention voltage of SRAM cells for chip identification. In *RFIDSec'12*, 2012.

[34] C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, and M. Stutzmann. Random p-n-junctions for physical cryptography. *Applied Physics Letters*, 96(172103), 2010.

[35] P. Koeberl, Ü. Kocabaş, and A.-R. Sadeghi. Memristor PUFs: a new generation of memory-based physically unclonable functions. In *DATE '13: Proceedings of the Conference on Design, Automation and Test in Europe*. March 2013.

[36] S.S. Kumar, J. Guajardo, R. Maes, G.J. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. *HOST 2008*.

[37] R. Kumar, W. Burleson: *Litho-aware and low power design of a secure current-based physically unclonable function.* IEEE International Symposium on Low Power Electronics and Design (ISLPED), 2013.

[38] P. Layman, S. Chaudhry, J. G. Norman, and J. R. Thomson. Electronic fingerprinting of semiconductor integrated circuits. (6,738,294), September 2002.

[39] 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. *IEEE VLSI Circuits Symposium*, 2004.

[40] D. Lim. *Extracting Secret Keys from Integrated Circuits*. Msc thesis, MIT, 2004.

[41] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration Systems*, 13(10):1200, 2005.

[42] K. Lofstrom and W.R. Daasch. IC identification circuit using device mismatch. *International Solid State Circuits Conference*, 2000.

[43] R. Maes, P. Tuyls, and I. Verbauwhede. Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs. *Cryptographic Hardware and Embedded Security*, 2009.

[44] A. Mahmoud, U. Rührmair, M. Majzoobi, F. Koushanfar: *Combined Modeling and Side Channel Attacks on Strong PUFs.* IACR Cryptology ePrint Archive 2013: 632 (2013)

[45] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Testing techniques for hardware security. In *International Test Conference (ITC)*, 2008.

[46] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Lightweight secure pufs. *IEEE/ACM Int. Conf. on Computer-Aided Design*, 2008.

[47] M. Majzoobi, M. Rostami, F. Koushanfar, D.S. Wallach, and S. Devadas: Slender PUF Protocol: A Lightweight, Robust, and Secure Authentication by Substring Matching. *IEEE S&P Workshops*, 2012.

[48] D. Nedospasov, J. P. Seifert, C. Helfmeier, and C. Boit. Invasive PUF Analysis. *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on*, pages 30–38, 2013.

[49] NXP Semiconductors. NXP Strengthens SmartMX2 Security Chips with PUF Anti-Cloning Technology, February 2013.

[50] Rafail Ostrovsky, Alessandra Scafuro, Ivan Visconti, Akshay Wadia: Universally Composable Secure Computation with (Malicious) Physically Uncloneable Functions. EUROCRYPT 2013: 702-718

[51] Erdinç Öztürk, Ghaith Hammouri, and Berk Sunar. Towards robust low cost authentication for pervasive devices. *IEEE PerCom*, 2008.

[52] R. Pappu. *Physical One-Way Functions*. Phd thesis, MIT, 2001.

[53] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026, 2002.

[54] P. Prabhu, A. Akel, L. Grupp, W.K. Yu, G. Suh, E. Kan, and S. Swanson. Extracting Device Fingerprints from Flash Memory by Exploiting Physical Variations. *Proceedings of the 4th International Conference on Trust and Trustworthy Computing*, 2011.

[55] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *IEEE international conference on neural networks*, 1993.

[56] S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, and S. S. Iyer. A Self-Authenticating Chip Architecture Using an Intrinsic Fingerprint of Embedded DRAM. *Solid-State Circuits, IEEE Journal of*, (99):1–10, 2013.

[57] U. Rührmair. Oblivious transfer based on physical unclonable functions (extended abstract). *TRUST 2010.* LNCS Vol. 6101, Springer, 2010.

[58] U. Rührmair: *PUFs at a Glance*. Design, Automation and Test in Europe (DATE'14), 2014.

[59] U. Rührmair, H. Busch, S. Katzenbeisser: Strong PUFs: Models, Constructions and Security Proofs. In A.-R. Sadeghi, P. Tuyls (Editors): *Towards Hardware Intrinsic Security: Foundation and Practice.* Springer, 2010.

[60] 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.

[61] U. Rührmair: *Oblivious Transfer based on Physical Unclonable Functions (Extended Abstract)*. TRUST 2010.

[62] U. Rührmair, M. van Dijk: *Practical Security Analysis of PUF-based Two-Player Protocols.* CHES 2012.

[63] U. Rührmair, M. van Dijk: *On the Practical Use of Physical Unclonable Functions in Oblivious Transfer and Bit Commitment Protocols*. Journal of Cryptographic Engineering (JCEN), 2013.

[64] U. Rührmair, M. van Dijk: *PUFs in Security Protocols: Attack Models and Security Evaluations*. IEEE Symposium on Security and Privacy (Oakland'13), 2013.

[65] U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, G. Csaba. Applications of high-capacity crossbar memories in cryptography. *IEEE Transactions on Nanotechnology*, 2011.

[66] U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, M. Stutzmann. Security applications of diodes with unique current-voltage characteristics. Financial Cryptography and Data Security (FC), 2010.

[67] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber. Modeling Attacks on Physical Unclonable Functions. *ACM CCS 2010*.

[68] U. Rührmair, J. Sölter: *PUF Modeling Attacks: An Introduction and Overview*. Design, Automation and Test in Europe (DATE'14), 2014.

[69] U. Rührmair, J. Sölter, F. Sehnke. On the Foundations of Physical Unclonable Functions. Cryptology ePrint Archive, 2009:277, 2009.

[70] 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 Transactions on Information Forensics and Security (IEEE T-IFS), 2013.

[71] H.P.P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc. New York, NY, USA, 1993.

[72] P. Simons, E. van der Sluis, and V. van der Leest. Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs. *HOST 2012*, pages 7–12, 2012.

[73] J. Sölter. *Cryptanalysis of Electrical PUFs via Machine Learning Algorithms*. MSc thesis, Technische Universität München, 2009.

[74] Y. Su, J. Holleman, and B. Otis. A 1.6 pj/bit 96% stable chip-id generating circuit using process variations. *International Solid State Circuits Conference*, 2007.

[75] G.E. Suh, S. Devadas. Physical unclonable functions for device authentication and secret key generation. *DAC 2007.*

[76] P. Tuyls, G. J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, R. Wolters. Read-proof hardware from protective coatings. *CHES 2006*.

[77] P. Tuyls, B. Skoric. Strong Authentication with PUFs. *In: Security, Privacy and Trust in Modern Data Management, M. Petkovic, W. Jonker (Eds.)*, Springer, 2007.

[78] http://www.verayo.com/

[79] X. Xu, W. Burleson. Hybrid Side-Channel / Machine- Learning Attacks on PUFs: A New Threat? *DATE 2014*.

[80] Y. Yao, M. Kim, J. Li, I. L. Markov, and F. Koushanfar. ClockPUF: Physical Unclonable Functions based on clock networks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*, pages 422–427. 2013.

[81] M-D. Yu and S. Devadas. Secure and Robust Error Correction for Physical Unclonable Functions. *Design & Test of Computers, IEEE*, 27(1):48–65, 2010.

[82] M.-D. Yu, D. M'Raïhi, R. Sowell, S. Devadas: Lightweight and Secure PUF Key Storage Using Limits of Machine Learning. *CHES 2011*.

Chapter 3

Modeling Attacks on Physical Unclonable Functions

As implicit already in their name, PUFs should be resistant against any form of cloning. By currently available fabrication technology, PUFs indeed cannot be cloned with perfect accuracy, for example atom by atom. Fabricating a *perfect physical clone* hence currently is infeasible. However, there could be other types of clones: Firstly *functional physical clones*, i.e., physical systems that behave like the original PUF in their challenge-response behavior, even though they do not coincide with this original atom by atom, or may actually even be totally different structures. Secondly *digital clones*, i.e., computer programms which numerically emulate the input-output behavior of the original PUF. If the simulation programm is efficient and short, and if the PUF output is digital, functional clones can usually be constructed from digital clones by simply implementing the digital clone in hardware.

This chapter is indeed concerned with constructing digital clones for several Strong PUFs by so-called *"modeling attacks"*. [1] In this attack form, the adversary collects a small fraction of all challenge-response pairs (CRPs) of a given Strong PUF. He then tries to constructs a numeric simulation model of the PUF, which can predict also previously unknown CRPs. If successful, this breaks any applications that are based on the unpredictabilty and unclonability of the PUF. The natural weapon of choice to this end are machine learning (ML) algorithms, since the above approach represents nothing else than a typical supervised ML-problem.

While PUF modeling attacks have not been invented by the author (see [47, 77] for early sources), the material presented in this chapter represents the currently most advanced results on this topic (compare [77, 99, 89, 90, 34, 35, 47, 73, 78]). The material has been published as

- U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions*. ACM Conference on Computer and Communications Security (ACM CCS), pp. 237-249, 2010.

[1] Recall from Chapter 2 that Strong PUFs are, together with Weak PUFs, one of the two main subclasses of PUFs.

Two related works, which are explicitly not used in this chapter, are

- 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 Transactions on Information Forensics and Security, Vol. 8(11), pp. 1876-1891, 2013,

  which is a journal version of the above article that is used in this chapter, and

- U. Rührmair, J. Sölter: *PUF Modeling Attacks: An Introduction and Overview.* Design, Automation & Test in Europe (DATE 2014), pp. 1-6, 2014.

His works on modeling attacks are probably the best known papers of the candidate. They arguably lead to a large number of follow-up works both on secure PUF design as well as on PUF attacks within the community.

Modeling Attacks on Physical Unclonable Functions

Ulrich Rührmair
Computer Science Departm.
TU München
80333 München, Germany
ruehrmair@in.tum.de

Frank Sehnke
Computer Science Departm.
TU München
80333 München, Germany
sehnke@in.tum.de

Jan Sölter
Computer Science Departm.
TU München
80333 München, Germany
jan_soelter@yahoo.com

Gideon Dror
The Academic College of
Tel-Aviv-Jaffa
Tel-Aviv 61083, Israel
gideon@mta.ac.il

Srinivas Devadas
Department of EECS
MIT
Cambrige, MA, USA
devadas@mit.edu

Jürgen Schmidhuber
Computer Science Departm.
TU München
80333 München, Germany
juergen@idsia.ch

ABSTRACT

We show in this paper how several proposed Physical Unclonable Functions (PUFs) can be broken by numerical modeling attacks. Given a set of challenge-response pairs (CRPs) of a PUF, our attacks construct a computer algorithm which behaves indistinguishably from the original PUF on almost all CRPs. This algorithm can subsequently impersonate the PUF, and can be cloned and distributed arbitrarily. This breaks the security of essentially all applications and protocols that are based on the respective PUF.

The PUFs we attacked successfully include standard Arbiter PUFs and Ring Oscillator PUFs of arbitrary sizes, and XOR Arbiter PUFs, Lightweight Secure PUFs, and Feed-Forward Arbiter PUFs of up to a given size and complexity. Our attacks are based upon various machine learning techniques, including Logistic Regression and Evolution Strategies. Our work leads to new design requirements for secure electrical PUFs, and will be useful to PUF designers and attackers alike.

Categories and Subject Descriptors

C.3 [**Special Purpose and Application-Based Systems**]: Smartcards; B.7.m [**Integrated Circuits**]: Miscellaneous; E.3 [**Data Encryption**]: Code breaking

General Terms

Security, Theory, Design

Keywords

Physical Unclonable Functions, Machine Learning, Cryptanalysis, Physical Cryptography

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.

1. INTRODUCTION

1.1 Motivation and Background

Electronic devices are now pervasive in our everyday life. They are an accessible target for adversaries, which raises a host of security and privacy issues. Classical cryptography offers several measures against these problems, but they all rest on the concept of a secret binary key: It is assumed that the devices can contain a piece of information that is, and remains, unknown to the adversary. Unfortunately, it can be difficult to uphold this requirement in practice. Physical attacks such as invasive, semi-invasive, or side-channel attacks, as well as software attacks like API-attacks and viruses, can lead to key exposure and full security breaks. The fact that the devices should be inexpensive, mobile, and cross-linked obviously aggravates the problem.

The described situation was one motivation that led to the development of *Physical Unclonable Functions (PUFs).* A PUF is a (partly) disordered physical system S that can be challenged with so-called external stimuli or challenges C_i, upon which it reacts with corresponding responses termed R_{C_i}. Contrary to standard digital systems, a PUF's responses shall depend on the nanoscale structural disorder present in the PUF. This disorder cannot be cloned or reproduced exactly, not even by its original manufacturer, and is unique to each PUF. Assuming the stability of the PUF's responses, any PUF S hence implements an individual function F_S that maps challenges C_i to responses R_{C_i} of the PUF.

Due to its complex and disordered structure, a PUF can avoid some of the shortcomings associated with digital keys. For example, it is usually harder to read out, predict, or derive its responses than to obtain the values of digital keys stored in non-volatile memory. This fact has been exploited for various PUF-based security protocols. Prominent examples including schemes for identification and authentication [1, 2], key exchange or digital rights management purposes [3].

1.2 Strong PUFs, Controlled PUFs, and Weak PUFs

There are several subtypes of PUFs, each with its own applications and security features. Three major types, which must explicitly be distinguished in this paper, are *Strong*

PUFs [1, 2, 4] [1], *Controlled PUFs* [3], and *Weak PUFs* [4], initially termed *Physically Obfuscated Keys (POKs)* [5].

1.2.1 Strong PUFs

Strong PUFs are disordered physical systems with a complex challenge-response behavior and very many possible challenges. Their security features are: (i) It must be impossible to physically clone a Strong PUF, i.e., to fabricate a second system which behaves indistinguishably from the original PUF in its challenge-response behavior. This restriction shall hold even for the original manufacturer of the PUF. (ii) A complete determination/measurement of all challenge-response pairs (CRPs) within a limited time frame (such as several days or even weeks) must be impossible, even if one can challenge the PUF freely and has unrestricted access to its responses. This property is usually met by the large number of possible challenges and the finite read-out speed of a Strong PUF. (iii) It must be difficult to numerically predict the response R_C of a Strong PUF to a randomly selected challenge C, even if many other CRPs are known.

Possible applications of Strong PUFs cover key establishment [1, 7], identification [1], and authentication [2]. They also include oblivious transfer [8] and any protocols derived from it, including zero-knowledge proofs, bit commitment, and secure multi-party computation [8]. In said applications, Strong PUFs can achieve secure protocols without the usual, standard computational assumptions concerning the factoring or discrete logarithm problem (albeit their security rests on other, independent computational and physical assumptions). Currently known electrical, circuit-based candidates for Strong PUFs are described in [9, 10, 11, 12, 13].

1.2.2 Controlled PUFs

A Controlled PUF as described in [3] uses a Strong PUF as a building block, but adds control logic that surrounds the PUF. The logic prevents challenges from being applied freely to the PUF, and hinders direct read-out of its responses. This logic can be used to thwart modeling attacks. However, if the outputs of the embedded Strong PUF can be directly probed, then it may be possible to model the Strong PUF and break the Controlled PUF protocol.

1.2.3 Weak PUFs

Weak PUFs, finally, may have very few challenges — in the extreme case just one, fixed challenge. Their response(s) R_{C_i} are used to derive a standard secret key, which is subsequently processed by the embedding system in the usual fashion, e.g., as a secret input for some cryptoscheme. Contrary to Strong PUFs, the responses of a Weak PUF are never meant to be given directly to the outside world.

Weak PUFs essentially are a special form of non-volatile key storage. Their advantage is that they may be harder to read out invasively than non-volatile memory like EEPROM. Typical examples include the SRAM PUF [14, 4], Butterfly PUF [15] and Coating PUF [16]. Integrated Strong PUFs have been suggested to build Weak PUFs or Physically Obfuscated Keys (POKs), in which case only a small subset of all possible challenges is used [5, 9].

One important aspect of Weak PUFs is error correction and stability. Since their responses are processed internally as a secret key, error correction must be carried out on-chip and with perfect precision. This often requires the storage of error-correcting helper data in non-volatile memory on the chip. Strong PUFs usually allow error correction schemes that are carried out by the external recipients of their responses.

[1]Strong PUFs have also been referred to as Physical Random Functions [5], or Physical One-Way Functions [6].

1.3 Modeling Attacks on PUFs

Modeling attacks on PUFs presume that an adversary Eve has, in one way or the other, collected a subset of all CRPs of the PUF, and tries to derive a numerical model from this data, i.e., a computer algorithm which correctly predicts the PUF's responses to arbitrary challenges with high probability. If successful, this breaks the security of the PUF and of any protocols built on it. It is known from earlier work that machine learning (ML) techniques are a natural and powerful tool for such modeling attacks [5, 17, 18, 19, 20]. How the required CRPs can be collected depends on the type of PUF under attack.

Strong PUFs.

Strong PUFs usually have no protection mechanisms that restricts Eve in challenging them or in reading out their responses. Their responses are freely accessible from the outside, and are usually not post-processed on chip [1, 9, 10, 11, 12, 13]. Most electrical Strong PUFs further operate at frequencies of a few MHz [12]. Therefore even short physical access periods enable the read-out of many CRPs. Another potential CRP source is simple protocol eavesdropping, for example on standard Strong PUF-based identification protocols, where the CRPs are sent in the clear [1]. Eavesdropping on responses, as well as physical access to the PUF that allows the adversary to apply arbitrary challenges and read out their responses, is part of the established attack model for Strong PUFs.

Controlled PUFs.

For any adversary that is restricted to non-invasive CRP measurement, modeling attacks can be successfully disabled if one uses a secure one-way hash over the outputs of the PUF to create a Controlled PUF. We note that this requires error correction of the PUF outputs which are inherently noisy [3]. Successful application of our techniques to a Controlled PUF only becomes possible if Eve can probe the internal, digital response signals of the underlying Strong PUF on their way to the control logic. Even though this is a significant assumption, probing digital signals is still easier than measuring continuous analog parameters within the underlying Strong PUF, for example determining its delay values. Physical access to the PUF is part of the natural attack model on PUFs, as mentioned above.

Weak PUFs.

Weak PUFs are only susceptible to model building attacks if a Strong PUF, embedded in some hardware system, is used to derive the physically obfuscated key. This method has been suggested in [5, 9]. In this case, the internal digital response signals of the Strong PUF to injected challenges have to be probed.

We stress that purely numerical modeling attacks, as presented in this paper, are not relevant for *Weak PUFs with just one challenge* (such as the Coating PUF, SRAM PUF, or Butterfly PUF). This does not necessarily imply that

these PUFs are more secure than Strong PUFs or Controlled PUFs, however. Other attack strategies can be applied, including invasive, side-channel and virus attacks, but they are not the topic of this paper. For example, probing the output of the SRAM cell prior to storing the value in a register can break the security of the cryptographic protocol that uses these outputs as a key. We also note that attacking a Controlled PUF via modeling attacks that target the underlying Strong PUF requires substantially more signal probing than breaking a Weak PUF that possesses just one challenge.

1.4 Our Contributions and Related Work

We describe successful modeling attacks on several known electrical candidates for Strong PUFs, including Arbiter PUFs, XOR Arbiter PUFs, Feed-Forward Arbiter PUFs, Lightweight Secure PUFs, and Ring Oscillator PUFs. Our attacks work for PUFs of up to a given number of inputs (or stages) or complexity. The prediction rates of our machine learned models significantly exceed the known or derived stability of the respective PUFs in silicon in these ranges.

Our attacks are very feasible on the CRP side. They require an amount of CRPs that grows only linearly or log-linearly in the relevant structural parameters of the attacked PUFs, such as their numbers of stages, XORs, feed-forward loops, or ring oscillators. The computation times needed to derive the models (i.e., to train the employed ML algorithms) are low-degree polynomial, with one exception: The computation times for attacking XOR Arbiter and Lightweight Secure PUFs grow, in approximation for medium number of XORs and large number of stages, super-polynomial in the number of XORs. But the instability of these PUFs also increases exponentially in their number of XORs, whence this parameter cannot be raised at will in practical applications. Still, it turns out that the number of stages in these two types of PUFs can be increased without significant effect on their instability, providing a potential lever for making these PUFs more secure without destroying their practicality. Our work thus also points to design requirements by which the security of XOR Arbiter PUFs and Lightweight Secure PUFs against modeling attacks could be upheld in the near future.

Our results break the security of any Strong PUF-type protocol that is based on one of the broken PUFs. This includes any identification, authentication, key exchange or digital rights management protocols, such as the ones described in [1, 2, 6, 7, 11]. Under the assumptions and attack scenarios described in Section 1.3, our findings also restrict the use of the broken Strong PUF architectures within Controlled PUFs and as Weak PUFs, if we assume that digital values can be probed.

Related Work on Modeling Attacks.

Earlier work on PUF modeling attacks, such as [11, 17, 18, 19], described successful attacks on standard Arbiter PUFs and on Feed-Forward Arbiter PUFs with one loop. But these approaches did not generalize to Feed-Forward Arbiter PUFs with more than two loops. The XOR Arbiter PUF, Lightweight PUF, Feed-Forward Arbiter PUF with more than two Feed-Forward Loops, and Ring Oscillator PUF have not been cryptanalyzed thus far. No scalability analyses of the required CRPs and computation times had been performed in previous works.

Entropy Analysis vs. Modeling Attacks.

Another useful approach to evaluate PUF security is entropy analysis. Two variants exist: First, to analyze the internal entropy of the PUF. This is similar to the established physical entropy analysis in solid-state systems. A second option is to analyze the statistical entropy of all challenge-response pairs of a PUF; how many of them are independent?

Entropy analysis is a valuable tool for PUF analysis, but it differs from our approach in two aspects. First, it is non-constructive in the sense that it does not tell you how to break a PUF, even if the entropy score is low. Modeling attacks, to the contrary, actually break PUFs. Second, it is not clear if the internal entropy of a circuit-based Strong PUF is a good estimate for its security. Equivalently, is the entropy of an AES secret key a good estimate of the AES security? The security of a Strong PUF comes from an interplay between its random internal parameters (which can be viewed as its entropy), and its internal model or internal functionality. It is not the internal entropy alone that determines the security. As an example, compare an 8-XOR, 256-bit XOR PUF to a standard PUF with bitlength of $8 \cdot 256 = 2048$. Both have the same internal entropy, but very different security properties, as we show in the sequel.

1.5 Organization of the Paper

The paper is organized as follows. We describe the methodology of our ML experiments in Section 2. In Sections 3 to 7, we present our results for various Strong PUF candidates. They deal with Arbiter PUFs, XOR Arbiter PUFs, Lightweight Arbiter PUFs, Feed-Forward Arbiter PUFs and Ring Oscillator PUFs, in sequence. We conclude with a summary and discussion of our results in Section 8.

2. METHODOLOGY SECTION

2.1 Employed Machine Learning Methods

2.1.1 Logistic Regression

Logistic Regression (LR) is a well-investigated supervised machine learning framework, which has been described, for example, in [21]. In its application to PUFs with single-bit outputs, each challenge $C = b_1 \cdots b_k$ is assigned a probability $p\,(C, t \,|\, \vec{w})$ that it generates a output $t \in \{-1, 1\}$ (for technical reasons, one makes the convention that $t \in \{-1, 1\}$ instead of $\{0, 1\}$). The vector $\vec{w}$ thereby encodes the relevant internal parameters, for example the particular runtime delays, of the individual PUF. The probability is given by the logistic sigmoid acting on a function $f(\vec{w})$ parametrized by the vector $\vec{w}$ as $p\,(C, t \,|\, \vec{w}) = \sigma(tf) = (1 + e^{-tf})^{-1}$. Thereby f determines through $f = 0$ a decision boundary of equal output probabilities. For a given training set $\mathcal{M}$ of CRPs the boundary is positioned by choosing the parameter vector $\vec{w}$ in such a way that the likelihood of observing this set is maximal, respectively the negative log-likelihood is minimal:

$$\hat{\vec{w}} = argmin_{\vec{w}} l(\mathcal{M}, \vec{w}) = argmin_{\vec{w}} \sum_{(C,\,t) \in\, \mathcal{M}} -ln\left(\sigma\left(tf(\vec{w}, C)\right)\right) \tag{1}$$

As there is no analytical solution to determine the optimal parameter vector $\hat{\vec{w}}$, it has to be optimized iteratively, e.g.,

using the gradient information

$$\nabla l(\mathcal{M}, \vec{w}) = \sum_{(C,\, t) \in \mathcal{M}} t(\sigma(tf(\vec{w}, C)) - 1)\nabla f(\vec{w}, C) \qquad (2)$$

From the different optimization methods which we tested in our ML experiments (standard gradient descent, iterative reweighted least squares, RProp [21] [22]), RProp gradient descent performed best. Logistic regression has the asset that the examined problems need not be (approximately) linearly separable in feature space, as is required for successful application of SVMs, but merely differentiable.

In our ML experiments, we used an implementation of LR with RProp programmed in our group, which has been put online, see [23]. The iteration is continued until we reach a point of convergence, i.e., until the averaged prediction rate of two consecutive blocks of five consecutive iterations does not increase anymore for the first time. If the reached performance after convergence on the training set is not sufficient, the process is started anew. After convergence to a good solution on the training set, the prediction error is evaluated on the test set.

The whole process is similar to training an Artificial Neural Network (ANN) [21]. The model of the PUF resembles the network with the runtime delays resembling the weights of an ANN. Similar to ANNs, we found that RProp makes a very big difference in convergence speed and stability of the LR (several XOR-PUFs were only learnable with RProp). But even with RProp the delay set can end up in a region of the search space where no helpful gradient information is available (local minimum). In such a case we encounter the above described situation of converging on a not sufficiently accurate solution and have to restart the process.

2.1.2 Evolution Strategies

Evolution Strategies (ES) [24, 25] belong to an ML subfield known as population-based heuristics. They are inspired by the evolutionary adaptation of a population of individuals to certain environmental conditions. In our case, one individual in the population is given by a concrete instantiation of the runtime delays in a PUF, i.e., by a concrete instantiation of the vector $\vec{w}$ appearing in Eqns. 1 and 2. The environmental fitness of the individual is determined by how well it (re-)produces the correct CRPs of the target PUF on a fixed training set of CRPs. ES runs through several evolutionary cycles or so-called *generations*. With a growing number of generations, the challenge-response behavior of the best individuals in the population better and better approximates the target PUF. ES is a randomized method that neither requires an (approximately) linearly separable problem (like Support Vector Machines), nor a differentiable model (such as LR with gradient descent); a merely parameterizable model suffices. Since all known electrical PUFs are easily parameterizable, ES is a very well-suited attack method.

We employed an in-house implementation of ES that is available from our machine learning library PyBrain [26]. The meta-parameters in all applications of ES throughout this paper are (6,36)-selection and a global mutation operator with $\tau = \frac{1}{\sqrt{n}}$. We furthermore used a technique called Lazy Evaluation (LE). LE means that not all CRPs of the training set are used to evaluate an individual's environmental fitness; instead, only a randomly chosen subset is used for evaluation, that changes in every generation. In this paper, we always used subsets of size 2,000 CRPs, and indicated this also in the caption of the respective tables.

2.2 Employed Computational Resources

We used two hardware systems to carry out our experiments: A stand-alone, consumer INTEL Quadcore Q9300 worth less than 1,000 Euros. Experiments run on this system are marked with the term "HW ★". Secondly, a 30-node cluster of AMD Opteron Quadcores, which represents a worth of around 30,000 Euros. Results that were obtained by this hardware are indicated by the term "HW ■". All computation times are calculated for one core of one processor of the corresponding hardware.

2.3 PUF Descriptions and Models

Arbiter PUFs.

Arbiter PUFs (Arb-PUFs) were first introduced in [11] [12] [9]. They consist of a sequence of k stages, for example multiplexers. Two electrical signals race simultaneously and in parallel through these stages. Their exact paths are determined by a sequence of k external bits $b_1 \cdots b_k$ applied to the stages, whereby the i-th bit is applied at the i-th stage. After the last stage, an "arbiter element" consisting of a latch determines whether the upper or lower signal arrived first and correspondingly outputs a zero or a one. The external bits are usually regarded as the challenge C of this PUF, i.e., $C = b_1 \cdots b_k$, and the output of the arbiter element is interpreted as their response R. See [11] [12] [9] for details. The parameter k is often referred to as the bitlength of the Arbiter PUF.

It has become standard to describe the functionality of Arb-PUFs via an additive linear delay model [17] [10] [19]. The overall delays of the signals are modeled as the sum of the delays in the stages. In this model, one can express the final delay difference Δ between the upper and the lower path in a k-bit Arb-PUF as $\Delta = \vec{w}^T \vec{\Phi}$, where $\vec{w}$ and $\vec{\Phi}$ are of dimension $k+1$. The parameter vector $\vec{w}$ encodes the delays for the subcomponents in the Arb-PUF stages, whereas the feature vector $\vec{\Phi}$ is solely a function of the applied $k-$bit challenge C [17] [10] [19].

In greater detail, the following holds. We denote by $\delta_i^{0/1}$ the runtime delay in stage i for the crossed (1) respectively uncrossed (0) signal path. Then

$$\vec{w} = (w^1, w^2, \ldots, w^k, w^{k+1})^T, \qquad (3)$$

where $w^1 = \frac{\delta_1^0 - \delta_1^1}{2}$, $w^i = \frac{\delta_{i-1}^0 + \delta_{i-1}^1 + \delta_i^0 - \delta_i^1}{2}$ for all $i = 2, \ldots, k$, and $w^{k+1} = \frac{\delta_k^0 + \delta_k^1}{2}$. Furthermore,

$$\vec{\Phi}(\vec{C}) = (\Phi^1(\vec{C}), \ldots, \Phi^k(\vec{C}), 1)^T, \qquad (4)$$

where $\Phi^l(\vec{C}) = \prod_{i=l}^{k}(1 - 2b_i)$ for $l = 1, \ldots, k$.

The output t of an Arb-PUF is determined by the sign of the final delay difference Δ. We make the technical convention of saying that $t = -1$ when the Arb-PUF output is actually 0, and $t = 1$ when the Arb-PUF output is 1:

$$t = sgn(\Delta) = sgn(\vec{w}^T \vec{\Phi}). \qquad (5)$$

Eqn. 5 shows that the vector $\vec{w}$ via $\vec{w}^T \vec{\Phi} = 0$ determines a separating hyperplane in the space of all feature vectors $\vec{\Phi}$. Any challenges C that have their feature vector located on the one side of that plane give response $t = -1$, those with

feature vectors on the other side $t = 1$. Determination of this hyperplane allows prediction of the PUF.

XOR Arbiter PUFs.

One possibility to strengthen the resilience of arbiter architectures against machine learning, which has been suggested in [9], is to employ l individual Arb-PUFs in parallel, each with k stages (i.e., each with bitlength k). The same challenge C is applied to all of them, and their individual outputs t_i are XORed in order to produce a global response t_{XOR}. We denote such an architecture as l-XOR Arb-PUF.

A formal model for the XOR Arb-PUF can be derived as follows. Making the convention $t_i \in \{-1, 1\}$ as done earlier, it holds that $t_{XOR} = \prod_{i=1}^{l} t_i$. This leads with equation (5) to a parametric model of an l-XOR Arb-PUF, where $\vec{w}_i$ and $\vec{\Phi}_i$ denote the parameter and feature vector, respectively, for the i-th Arb PUF:

$$t_{XOR} = \prod_{i=1}^{l} sgn(\vec{w}_i^T \vec{\Phi}_i) = sgn(\prod_{i=1}^{l} \vec{w}_i^T \vec{\Phi}_i) \quad (6)$$

$$= sgn(\underbrace{\bigotimes_{i=1}^{l} \vec{w}_i^T}_{\vec{w}_{XOR}} \underbrace{\bigotimes_{i=1}^{l} \vec{\Phi}_i}_{\vec{\Phi}_{XOR}}) = sgn(\vec{w}_{XOR}^T \vec{\Phi}_{XOR}) \quad (7)$$

Whereas (6) gives a non-linear decision boundary with $l(k+1)$ parameters, (7) defines a linear decision boundary by a separating hyperplane $\vec{w}_{XOR}$ which is of dimension $(k+1)^l$.

Lightweight Secure PUFs.

Another type of PUF, which we term Lightweight Secure PUF or Lightweight PUF for short, has been introduced in [10]. It is similar to the XOR Arb-PUF of the last paragraph. At its heart are l individual standard Arb-PUFs arranged in parallel, each with k stages (i.e., with bitlength k), which produce l individual outputs $r_1, \ldots, r_l$. These individual outputs are XORed to produce a multi-bit response $o_1, ..., o_m$ of the Lightweight PUF, according to the formula

$$o_j = \bigoplus_{i=1,\ldots,x} r_{(j+s+i) \bmod l} \quad \text{for } j = 1, \ldots, m. \quad (8)$$

Thereby the values for m (the number of output bits of the Lightweight PUF), x (the number of values r_j that influence each single output bit) and s (the circular shift in choosing the x values r_j) are variable design parameters.

Another difference to the XOR Arb-PUFs lies in the l inputs $C_1 = b_1^1 \cdots b_k^1, C_2 = b_1^2 \cdots b_k^2, \ldots, C_l = b_1^l \cdots b_k^l$ which are applied to the l individual Arb-PUFs. Contrary to XOR Arb-PUFs, it does not hold that $C_1 = C_2 = \ldots = C_l = C$, but a more complicated input mapping that derives the individual inputs C_i from the global input C is applied. This input mapping constitutes the most significant difference between the Lightweight PUF and the XOR Arb PUF. We refer the reader to [10] for further details.

In order to predict the whole output of the Lightweight PUF, one can apply similar models and ML techniques as in the last section to predict *its single output bits* o_j. While the probability to predict the full output of course decreases exponentially in the misclassification rate of a single bit, the stability of the full output of the Lightweight PUF also decreases exponentially in the same parameters. It therefore seems fair to attack it in the described manner; in any case, our results challenge the bit security of the Lightweight PUF.

Feed Forward Arbiter PUFs.

Feed Forward Arbiter PUFs (FF Arb-PUFs) were introduced in [11] [12] [17] and further discussed in [19]. Some of their multiplexers are not switched in dependence of an external challenge bit, but as a function of the delay differences accumulated in earlier parts of the circuit. Additional arbiter components evaluate these delay differences, and their output bit is fed into said multiplexers in a "feed-forward loop" (FF-loop). We note that an FF Arb-PUF with k-bit challenges $C = b_1 \cdots b_k$ (i.e., with bitlength k) and l loops has $s = k + l$ multiplexers or stages.

The described dependency makes natural architecture models of FF Arb-PUFs no longer differentiable. Consequently, FF Arb-PUFs cannot be attacked generically with ML methods that require linearly separable or differentiable models (like SVMs or LR), even though such models can be found in special cases, for example for small numbers of non-overlapping loops.

The number of loops as well as the starting and end point of the FF-loops are variable design parameters, and a host of different architectures for an FF Arb-PUF with a moderate or even large number of loops are possible. The architecture we investigated in this paper consists of loops that are distributed at equal distances over the structure, and which just overlap each other: If the starting point of loop m lies in between stages n and $n+1$, then the previous loop $m-1$ has its end point in the immediately following stage $n+1$. This seemed the natural and straightforward architectural choice; future experiments will determine whether this is indeed the optimal (i.e., most secure) architecture.

Ring Oscillator PUFs.

Ring Oscillator PUFs (RO-PUFs) were discussed in [9]. They are based on the influence of fabrication variations on the frequency of several, identically designed ring oscillators. While [9] describes the use of Ring Oscillator PUFs in the context of Controlled PUFs and limited-count authentication, it is worth analyzing them as candidate Strong PUFs. A RO-PUF consists of k oscillators, each of which has its own, unique frequency caused by manufacturing variations. The input of a RO-PUF consists of a tuple (i, j), which selects two of the k oscillators. Their frequencies are compared, and the output of the RO-PUF is "0" if the former oscillates faster than the latter, and "1" else. A ring oscillator can be modeled in a straightforward fashion by a tuple of frequencies $(f_1, \ldots, f_k)$. Its output on input (i, j) is "0" if $f_i > f_j$, and "1" else.

2.4 CRP Generation, Prediction Error, and Number of CRPs

Given a PUF-architecture that should be examined, the challenge-response pairs (CRPs) that we used in our ML experiments were generated in the following fashion: (i) The delay values for this PUF architecture were chosen pseudo-randomly according to a standard normal distribution. We sometimes refer to this as choosing a certain PUF instance in the paper. In the language of Equ. 3, it amounts to choosing the entries w^i pseudo-randomly. (ii) If a response of this PUF instance to a given challenge is needed, it is calculated by use of the delays selected in step (i), and by application

ML Method	Bit Length	Prediction Rate	CRPs	Training Time
LR	64	95%	640	0.01 sec
		99%	2,555	0.13 sec
		99.9%	18,050	0.60 sec
LR	128	95%	1,350	0.06 sec
		99%	5,570	0.51 sec
		99.9%	39,200	2.10 sec

Table 1: LR on Arb PUFs with 64 and 128 stages, or with bitlength 64 and 128. We used HW ★.

of a linear additive delay model [13]: The delays of the two electrical signal paths are simply added up and compared.

We use the following definitions throughout the paper: The prediction error ϵ is the ratio of incorrect responses of the trained ML algorithm when evaluated on the test set. For all applications of LR, the test set each time consisted of 10,000 randomly chosen CRPs. For all applications of ES (i.e., for the Feed-Forward Arbiter PUF), the test set each time consisted of $8,000$ randomly chosen CRPs. The prediction rate is $1 - \epsilon$.

N_{CRP} (or simply "CRPs") denotes the number of CRPs employed by the attacker in his respective attack, for example in order to achieve a certain prediction rate. This nomenclature holds throughout the whole paper. Nevertheless, one subtle difference should be made explicit: In all applications of LR (i.e., in Sections 3 to 5), N_{CRP} is equal to the size of the training set of the ML algorithm, as one would usually expect. In the applications of ES (i.e., in Section 6), however, the situation is more involved. The attacker needs a test set himself in order to determine which of his many random runs was the best. The value N_{CRP} given in the tables and formulas of Section 6 hence reflects the sum of the sizes of the training set and the test set employed by the attacker.

3. ARBITER PUFS

3.1 Machine Learning Results

To determine the separating hyperplane $\vec{w}^T \vec{\Phi} = 0$, we applied SVMs, LR and ES. LR achieved the best results, which are shown in Table 1. We chose three different prediction rates as targets: 95% is roughly the environmental stability of a 64-bit Arbiter PUF when exposed to a temperature variation of 45C and voltage variation of ±2% [2]. The values 99% and 99.9%, respectively, represent benchmarks for optimized ML results. All figures in Table 1 were obtained by averaging over 5 different training sets. Accuracies were estimated using test sets of 10,000 CRPs.

3.2 Scalability

We also executed scalability experiments with LR, which are displayed in Fig. 1 and Fig. 2. They show that the relevant parameters – the required number of CRPs in the training set and the computational complexity, i.e., the number of basic operations – grow both linearly or low-degree polynomially in the misclassification rate ϵ and the length k of the Arb PUF. Theoretical considerations (dimension of the feature space, Vapnik-Chervonenkis dimension) suggest that the *minimal* number of CRPs N_{CRP} that is necessary to model a k-stage arbiter with a misclassification rate of ϵ should obey the relation

$$N_{CRP} = O(k/\epsilon). \quad (9)$$

This was confirmed by our experimental results.

In practical PUF applications, it is essential to know the concrete number of CRPs that may become known before the PUF-security breaks down. Assuming an approximate linear functional dependency $y = ax + c$ in the double logarithmic plot of Fig. 1 with a slope of $a = -1$, we obtained the following empirical formula (10). It gives the approximate number of CRPs N_{CRP} that is required to learn a k-stage arbiter PUF with error rate ϵ:

$$N_{CRP} \approx 0.5 \cdot \frac{k+1}{\epsilon} \quad (10)$$

Our experiments also showed that the training time of the ML algorithms, measured in the number of basic operations N_{BOP}, grows slowly. It is determined by the following two factors: (i) The evaluation of the current model's likelihood (Eqn. 1) and its gradient (Eqn. 2), and (ii) the number of iterations of the optimization procedure before convergence occurs (see section 2.1.1). The former is both a sum over a function of the feature vectors $\vec{\Phi}$ for all N_{CRP}, and therefore has complexity $O(k \cdot N_{CRP})$. On the basis of the data shown in Figure 2, we may further estimate that the numbers of iterations increases proportional to the logarithm of the number of CRPs N_{CRP}. Together, this yields an overall complexity of

$$N_{BOP} = O\left(\frac{k^2}{\epsilon} \cdot \log \frac{k}{\epsilon}\right). \quad (11)$$

4. XOR ARBITER PUFS

4.1 Machine Learning Results

In the application of SVMs and ES to XOR Arb-PUFs, we were able to break small instances, for example XOR Arb-PUFs with 2 or 3 XORs and 64 stages. LR significantly

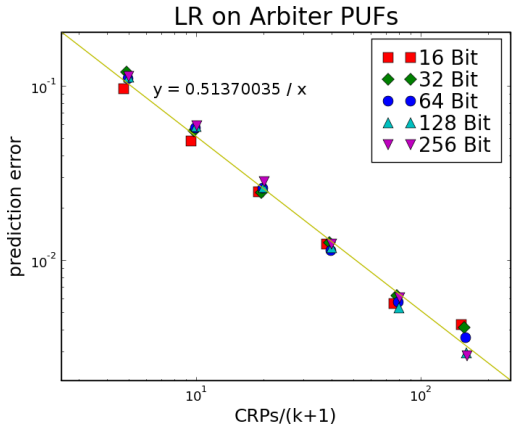

Figure 1: Double logarithmic plot of misclassification rate ϵ on the ratio of training CRPs N_{CRP} and $dim(\Phi) = k + 1$.

[2]The exact figures reported in [17] are: 4.57% CRP variation for a temperature variation of 45C, and 2.16% for a voltage variation of ±2%.

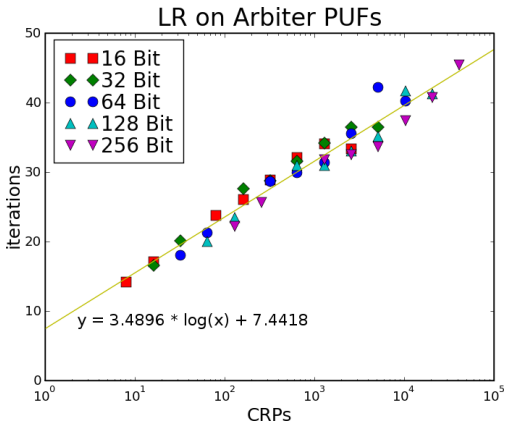

Figure 2: No. of iterations of the LR algorithm until "convergence" occurs (see section 2), plotted in dependence of the training set size N_{CRP}.

outperformed the other two methods. The key observation is that instead of determining the linear decision boundary (Eqn. 7), one can also specify the non-linear boundary (Eqn. 6). This is done by setting the LR decision boundary $f = \prod_{i=1}^{l} \vec{w}_i^T \vec{\Phi}_i$. The results are displayed in Table 2.

4.2 Performance on Error-Inflicted CRPs

The CRPs used in Section 4.1 have been generated pseudorandomly via an additive, linear delay model of the PUF. This deviates from reality in two aspects: First of all, the CRPs obtained from real PUFs are subject to noise and random errors. Secondly, the linear model matches the phenomena on a real circuit very closely [17], but not perfectly. This leads to a deviation of any real system from the linear model on a small percentage of all CRPs.

In order to mimic this situation, we investigated the ML performance when a small error is injected artificially into the training sets. A given percentage of responses in the training set were chosen randomly, and their bit values were flipped. Afterwards, the ML performance on the unaltered, error-free test sets was evaluated. The results are displayed in Tables 3 and 4. They show that LR can cope very well with errors, provided that around 3 to 4 times more CRPs are used. The required convergence times on error inflicted training sets did not change substantially compared to error free training sets of the same sizes.

ML Method	Bit Length	Pred. Rate	No. of XORs	CRPs ($\times 10^3$)	Training Time
LR	64	99%	4	12	3:42 min
			5	80	2:08 hrs
			6	200	31:01 hrs
LR	128	99%	4	24	2:52 hrs
			5	500	16:36 hrs
			6	—	—

Table 2: LR on XOR Arbiter PUFs. Training times are averaged over different PUF-instances. HW ★.

CRPs ($\times 10^3$)		Percentage of error-inflicted CRPs			
		0%	2%	5%	10%
24	Best Pr.	98.76%	92.83%	88.05%	—
	Ave. Pr.	98.62%	91.37%	88.05%	—
	Suc. Tr.	0.6%	0.8%	0.2%	0.0%
	Inst.	40.0%	25.0%	5.0%	0.0%
50	Best Pr.	99.49%	95.17%	92.67%	89.89%
	Ave. Pr.	99.37%	94.39%	91.62%	88.20%
	Suc. Tr.	12.4%	13.9%	10.0%	4.6%
	Inst.	100.0%	62.5%	50.0%	20.0%
200	Best Pr.	99.88%	97.74%	96.01%	94.61%
	Ave. Pr.	99.78%	97.34%	95.69%	93.75%
	Suc. Tr.	100.0%	87.0%	87.0%	71.4%
	Inst.	100.0%	100.0%	100.0%	100.0%

Table 3: LR on 128-bit, 4-XOR Arb PUFs with different levels of error in the training set. We show the best and average prediction rates of 40 randomly chosen instances, the percentage of successful trials over all instances, and the percentage of and instances that converged to a sufficient optimum in at least one trial. We used HW ■.

CRPs ($\times 10^3$)		Percentage of error-inflicted CRPs			
		0%	2%	5%	10%
500	Best Pr.	99.90%	97.55%	96.48%	93.12%
	Ave. Pr.	99.84%	97.33%	95.84%	93.12%
	Suc. Tr.	7.0%	2.9%	0.9%	0.7%
	Inst.	20.0%	20.0%	10.0%	5.0%

Table 4: LR on 128-bit, 5-XOR Arb PUFs with different amounts of error in the training set. Rest as in the caption of Table 3. We used HW ■.

4.3 Scalability

Figures 4 and 5 display the results of our scaling experiments with LR. Again, the smallest number of CRPs in the training set N_{CRP} needed to achieve predictions with a misclassification rate ϵ scales linearly with the number of parameters of the problem (the product of the number of stages k and the number of XORed Arb-PUFs l):

$$N_{CRP} \sim \frac{(k+1) \cdot l}{\epsilon}. \qquad (12)$$

But, in contrast to standard Arb-PUFs, optimizing the non-linear decision boundary (6) on the training set now is a non-convex problem, so that the LR algorithm is not guaranteed to find (an attractor of) the global optimum in its first trial. It needs to be iteratively restarted N_{trial} times. N_{trial} thereby can be expected to not only depend on k and l, but also on the size N_{CRP} of the employed training set.

As it is argued in greater detail in [20], the success rate ($= 1/N_{trial}$) of finding (an attractor of) the global optimum seems indeed determined by the ratio of dimensions of gradient information ($\propto N_{CRP}$ as the gradient is a linear combination of the feature vector) and the dimension d_Φ in which the problem is linear separable. The dimension d_Φ is the number of independent dimensions of $\vec{\Phi}_{XOR} = \bigotimes_{i=1}^{l} \vec{\Phi}_i = \bigotimes_{i=1}^{l} (\Phi_i^1 \ldots, \Phi_i^k, 1)^T$.

As the tensor product of several vectors consists of all possible products between their vector components, the independent dimensions are given by the number of differ-

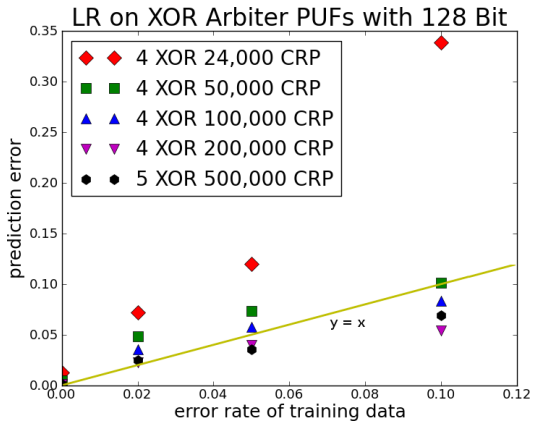

Figure 3: Graphical illustration of the effect of error on LR in the training set, with chosen data points from Tables 3 and 4. We used HW ■.

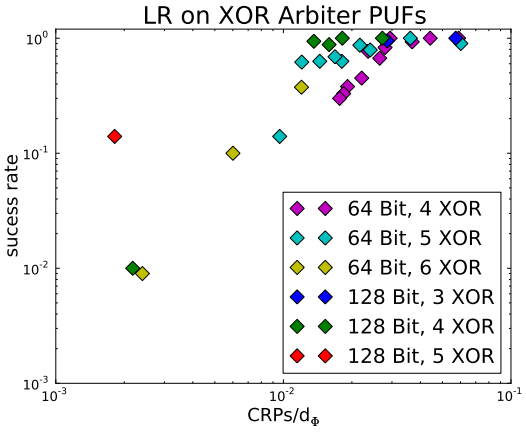

Figure 5: Average rate of success of the LR algorithm plotted in dependence of the ratio d_Φ (see Eqn. (13)) to N_{CRP}. We used HW ■.

ent products of the form $\Phi_1^{i_1} \cdot \Phi_2^{i_2} \cdot \ldots \Phi_l^{i_l}$ for $i_1, i_2, \ldots, i_l \in \{1, 2, \ldots, k+1\}$ (where we say that $\Phi_i^{k+1} = 1$ for all $i = 1, \ldots, l$). For XOR Arb-PUFs, we furthermore know that the same challenge is applied to all l internal Arbiter PUFs, which tells us that $\Phi_j^i = \Phi_{j'}^i = \Phi^i$ for all $j, j' \in \{1, \ldots, l\}$ and $i \in \{1, \ldots, k+1\}$. Since a repetition of one component does not affect the product regardless of its value (recall that $\Phi^r \cdot \Phi^r = \pm 1 \cdot \pm 1 = 1$), the number of the above products can be obtained by counting the unrepeated components. The number of different products of the above form is therefore given as the number of l-tuples without repetition, plus the number of $(l-2)$-tuples without repetition (corresponding to all l-tuples with 1 repetition), plus the number of $(l-4)$-tuples without repetition (corresponding to all l-tuples with 2 repetitions), etc.

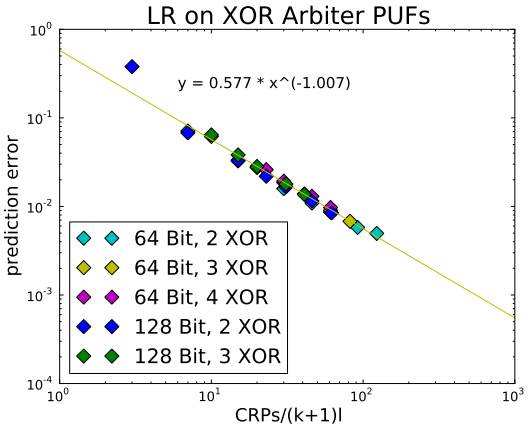

Figure 4: Double logarithmic plot of misclassification rate ϵ on the ratio of training CRPs N_{CRP} and problem size $dim(\Phi) = (k+1) \cdot l$. We used HW ■.

Writing this down more formally, d_Φ is given by

$$d_\Phi = \binom{k+1}{l} + \binom{k+1}{l-2} + \binom{k+1}{l-4} + \ldots \overset{k \gg l}{\approx} \frac{(k+1)^l}{l!}. \quad (13)$$

The approximation applies when k is considerably larger than l, which holds for the considered PUFs for stability reasons. Following [20], this seems to lead to an expected number of restarts N_{trial} to obtain a valid decision boundary on the training set (that is, a parameter set $\vec{w}$ that separates the training set), of

$$N_{trial} = O\left(\frac{d_\Phi}{N_{CRP}}\right) = O\left(\frac{(k+1)^l}{N_{CRP} \cdot l!}\right). \quad (14)$$

Furthermore, each trial has the complexity

$$T_{trial} = O\left((k+1) \cdot l \cdot N_{CRP}\right). \quad (15)$$

5. LIGHTWEIGHT SECURE PUFS

5.1 Machine Learning Results

In order to test the influence of the specific input mapping of the Lightweight PUF on its machine-learnability (see Sec. 2.3), we examined architectures with the following parameters: variable l, $m = 1$, $x = l$, and arbitrary s. We focused on LR right from the start, since this method was best in class for XOR Arb-PUFs, and obtained the results shown in Table 5. The specific design of the Lightweight PUF improves its ML resilience by a notable quantitative factor, especially with respect to the training times and CRPs. The given training times and prediction rates relate to single output bits of the Lightweight PUF.

5.2 Scalability

Some theoretical consideration [20] shows the underlying ML problem for the Lightweight PUF and the XOR Arb PUF are similar with respect to the required CRPs, but

Bit Length	Pred. Rate	No. of XORs	CRPs	Training Time
64	99%	3	6,000	8.9 sec
		4	12,000	1:28 hrs
		5	300,000	13:06 hrs
128	99%	3	15,000	40 sec
		4	500,000	59:42 min
		5	10^6	267 days

Table 5: LR on Lightweight PUFs. Prediction rate refers to single output bits. Training times were averaged over different PUF instances. HW ★.

differ quantitatively in the resulting runtimes. The asymptotic formula on N_{CRP} given for the XOR Arb PUF (Eqn. 12) analogously also holds for the Lightweight PUF. But due to the influence of the special challenge mapping of the Lightweight PUF, the number N_{trial} has a growth rate that is different from Eqn. 14. It seems to lie between $O(\frac{(k+1)^l}{N_{CRP} \cdot l!})$ and the related expression $O(\frac{(k+1)^l}{N_{CRP}})$ [20]. While these two formulas differ by factor of $l!$, we note that in our case $k \gg l$, and that l is comparatively small for stability reasons. Again, all these considerations on N_{CRP} and N_{Trial} hold for the prediction of single output bits of the Lightweight PUF.

These points were at least qualitatively confirmed by our scalability experiments. We observed in agreement with the above discussion that with the same ratio $CRPs/d_\Phi$ the LR algorithm will have a longer runtime for the Lightweight PUF than for the XOR Arb-PUF. For example, while with a training set size of $12,000$ for the 64-bit 4-XOR Arb-PUF on average about 5 trials were sufficient, for the corresponding Lightweight PUF 100 trials were necessary. The specific challenge architecture of the Lightweight PUF hence noticeably complicates the life of an attacker in practice.

6. FEED FORWARD ARBITER PUFS

6.1 Machine Learning Results

We experimented with SVMs and LR on FF Arb-PUFs, using different models and input representations, but could only break special cases with small numbers of non-overlapping FF loops, such as $l = 1, 2$. This is in agreement with earlier results reported in [19].

The application of ES finally allowed us to tackle much more complex FF-architectures with up to 8 FF-loops. All loops have equal length, and are distributed regularly over the PUF, with overlapping start- and endpoints of successive loops, as described in Section 2.3. Table 6 shows the results we obtained. The given prediction rates are the best of 40 trials on one randomly chosen PUF-instance of the respective length. The given CRP numbers are the sum of the training set and the test set employed by the attacker; a fraction of 5/6 was used as the training set, 1/6 as the test set (see Section 2.4). We note for comparison that in-silicon implementations of 64-bit FF Arb-PUFs with 7 FF-loops are known to have an environmental stability of 90.16% [17].

6.2 Results on Error-Inflicted CRPs

For the same reasons as in Section 4.2, we evaluated the performance on error-inflicted CRPs with respect to ES and FF Arb PUFs. The results are shown in Table 7 and Fig. 6. ES possesses an extremely high tolerance against the inflicted errors; its performance is hardly changed at all.

Bit Length	FF-loops	Pred. Rate Best Run	CRPs	Training Time
64	6	97.72%	50,000	07:51 min
	7	99.38%	50,000	47:07 min
	8	99.50%	50,000	47:07 min
	9	98.86%	50,000	47:07 min
	10	97.86%	50,000	47:07 min
128	6	99.11%	50,000	3:15 hrs
	7	97.43%	50,000	3:15 hrs
	8	98.97%	50,000	3:15 hrs
	9	98.78%	50,000	3:15 hrs
	10	97.31%	50,000	3:15 hrs

Table 6: ES on Feed-Forward Arbiter PUFs. Prediction rates are for the best of a total of 40 trials on a single, randomly chosen PUF instance. Training times are for a single trial. We applied Lazy Evaluation with 2,000 CRPs. We used HW ■.

6.3 Scalability

We started by empirically investigating the CRP growth as a function of the number of challenge bits, examining architectures of varying bitlength that all have 6 FF-loops. The loops are distributed as described in Section 2.3. The corresponding results are shown in Figure 7. Every data point corresponds to the averaged prediction error of 10 trials on the same, random PUF-instance.

Secondly, we investigated the CRP requirements as a function of a growing number of FF-loops, examining architectures with 64 bits. The corresponding results are depicted in Figure 8. Again, each data point shows the averaged prediction error of 10 trials on the same, random PUF instance.

In contrast to the Sections 4.3 and 5.2, it is now much more difficult to derive reliable scalability formulas from this data. The reasons are threefold. First, the structure of ES provides less theoretical footing for formal derivations. Second, the random nature of ES produces a very large variance in the data points, making also clean empirical derivations more difficult. Third, we observed an interesting effect when comparing the performance of ES vs. SVM/LR on the Arb PUF: While the supervised ML methods SVM and LR showed a linear relationship between the prediction error ϵ and the required CRPs even for very small ϵ, ES proved more CRP hungry in these extreme regions for ϵ, clearly showing a superlinear growth. The same effect can be expected for

CRPs ($\times 10^3$)		Percentage of error-inflicted CRPs			
		0%	2%	5%	10%
50	Best Pr.	98.29%	97.78%	98.33%	97.68%
	Ave. Pr.	89.94%	88.75%	89.09%	87.91%
	Suc. Tr.	42.5%	37.5%	35.0%	32.5%

Table 7: ES on 64-bit, 6 FF Arb PUFs with different levels of error in the training set. We show the best and average prediction rates from over 40 independent trials on a single, randomly chosen PUF instance, and the percentage of successful trials that converged to 90% or better. We used HW ■.

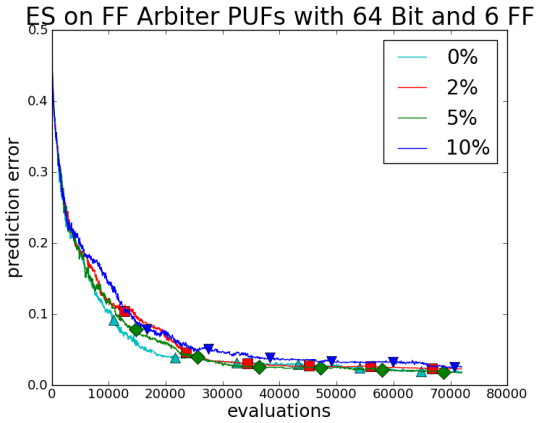

Figure 6: Graphical illustration of the tolerance of ES to errors. We show the best result of 40 independent trials on one randomly chosen PUF instance for varying error levels in the training set. The results hardly differ. We used HW ■.

FF architectures, meaning that one consistent formula for extreme values of ϵ may be difficult to obtain.

It still seems somewhat suggestive from the data points in Figures. 7 and 8 to conclude that the growth in CRPs is about linear, and that the computation time grows polynomially. For the reasons given above, however, we would like to remain conservative, and present the upcoming empirical formulas only in the status of a conjecture.

The data gathered in our experiments is best explained by assuming a qualitative relation of the form

$$N_{CRP} = O(s/\epsilon^c) \tag{16}$$

for some constant $0 < c < 1$, where s is the number of stages in the PUF. Concrete estimation from our data points leads to an approximate formula of the form

$$N_{CRP} \approx 9 \cdot \frac{s+1}{\epsilon^{3/4}}. \tag{17}$$

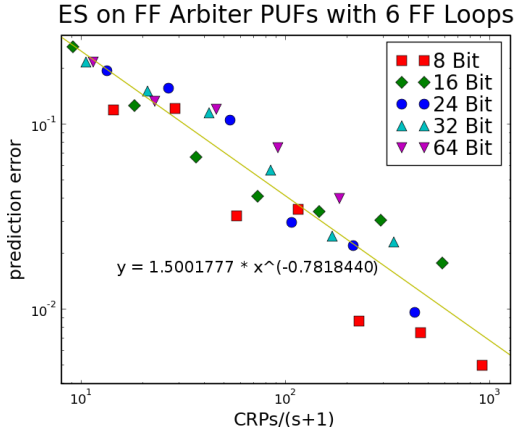

Figure 7: Results of 10 trials per data point with ES for different lengths of FF Arbiter PUFs and the hyperbola fit. HW ■.

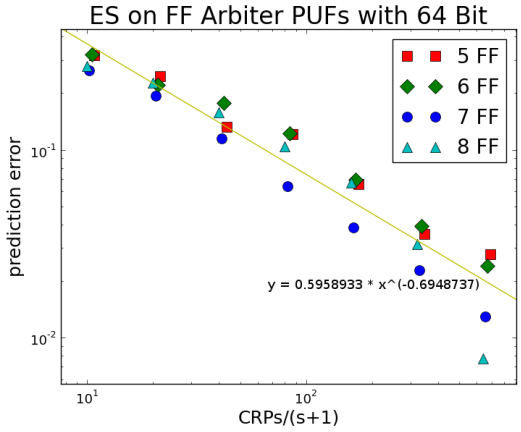

Figure 8: Results of 10 trials per data point with ES for different numbers of FF-loops and the hyperbola fit. HW ■.

The *computation time* required by ES is determined by the following factors: (i) The computation of the vector product $\vec{w}^T\vec{\Phi}$, which grows linearly with s. (ii) The evolution applied to this product, which is negligible compared to the other steps. (iii) The number of iterations or *"generations"* in ES until a small misclassification rate is achieved. We conjecture that this grows linearly with the number of multiplexers s. (iv) The number of CRPs that are used to evaluate the individuals per iteration. If Eqn. 17 is valid, then N_{CRP} is on the order of $O(s/\epsilon^c)$.

Assuming the correctness of the conjectures made in this derivation, this would lead to a polynomial growth of the computation time in terms of the relevant parameters k, l and s. It could then be conjectured that the number of basic computational operations N_{BOP} obeys

$$N_{BOP} = O(s^3/\epsilon^c) \tag{18}$$

for some constant $0 < c < 1$.

7. RING OSCILLATOR PUFS

7.1 Possible Attacks

There are several strategies to attack a RO-PUF. The most straightforward attempt is a simple read out of all CRPs. This is easy, since there are just $k(k-1)/2 = O(k^2)$ CRPs of interest, given k ring oscillators.

If Eve is able to choose the CRPs adaptively, she can employ a standard sorting algorithm to sort the RO-PUF's frequencies $(f_1, \ldots, f_k)$ in ascending order. This strategy

Method	No. of Oscill.	Pred. Rate average		CRPs	
QS	256	99%	99.9%	14,060	28,891
	512	99%	99.9%	36,062	103,986
	1024	99%	99.9%	83,941	345,834

Table 8: Quick Sort applied to the Ring Oscillator PUF. The given CRPs are averaged over 40 trials. We used HW ■.

	No. of XORs/Loops	ML Method	Bit Length	Prediction Rate	CRPs ($\times 10^3$)	Training Time
Arbiter PUF	—	LR	128	99.9%	39.2	2.10 sec
XOR Arbiter PUF	5	LR	128	99.0%	500	16:36 hrs
Lightweight PUF	5	LR	128	99.0%	1000	267 days
FF Arbiter PUF	8	ES	128	99.0%	50	3:15 hrs

Table 9: Some of our main results.

subsequently allows her to predict all outputs with 100% correctness, without knowing the exact frequencies f_i themselves. The time and CRP complexities of the respective sorting algorithms are well known [27]; for example, there are several algorithms with average- and even worst-case CRP complexity of $N_{CRP} = O(k \cdot \log k)$. Their running times are also low-degree polynomial.

The most interesting case for our investigations is when Eve cannot adaptively choose the CRPs she obtains, but still wants to achieve optimal prediction rates. This case occurs in practice whenever Eve obtains her CRPs from protocol eavesdropping, for example. We carried out experiments for this case, in which we applied Quick Sort (QS) to randomly drawn CRPs. The results are shown in Table 8. The estimated required number of CRPs is given by

$$N_{CRP} \approx \frac{k(k-1)(1-2\epsilon)}{2+\epsilon(k-1)}, \tag{19}$$

and the training times are low-degree polynomial. Eqn. 19 quantifies limited-count authentication capabilities of RO-PUFs.

8. SUMMARY AND DISCUSSION

Summary.

We investigated the resilience of currently published electrical Strong PUFs against modeling attacks. To that end, we applied various machine learning techniques to challenge-response data generated pseudo-randomly via an additive delay model. Some of our main results are summarized in Table 9.

We found that all examined Strong PUF candidates under a given size could be machine learned with success rates above their in-silicon stability. The attacks require a number of CRPs that grows only linearly or log-linearly in the internal parameters of the PUFs, such as their number of stages, XORs, feed-forward loops or ring oscillators. Apart from XOR Arbiter PUFs and Lightweight PUFs (whose training times grew quasi-exponentially in their number of XORs for large bitlengths k and small to medium number of XORs l), the training times of the applied machine learning algorithms are low-degree polynomial, too.

While we have presented results only on pseudo-random CRP data generated in the additive delay model, experiments with silicon implementations [17] [28] have shown that the additive delay model achieves very high accuracy. We also showed that the stability of our results against random errors in the CRP data is high. Our approach is hence robust against some inaccuracies in the model and against measurement noise. In our opinion, it will transfer to the case where CRP data is collected from silicon PUF chips.

Our results prohibit the use of the broken architectures as Strong PUFs or in Strong-PUF based protocols. Under the assumption that digital signals can be probed, they also affect the applicability of the cryptanalyzed PUFs as building blocks in Controlled PUFs and Weak PUFs.

Discussion.

Two straightforward, but biased interpretations of our results would be the following: (i) All Strong PUFs are insecure. (ii) The long-term security of electrical Strong PUFs can be restored trivially, for example by increasing the PUF's size. Both views are simplistic, and the truth is more involved.

Starting with (i), our current attacks are indeed sufficient to break most implemented PUFs. But there are several ways how PUF designers can fight back in future implementations. First, increasing the bitlength k in an XOR Arbiter PUF or Lightweight Secure PUF with l XORs increases the effort of the presented attacks methods as a polynomial function of k with exponent l (in approximation for large k and small or medium l). At the same time, it does not worsen the PUF's stability [28]. For now, one could therefore disable attacks through choosing a strongly increased value of k and a value of l that corresponds to the stability limit of such a construction. For example, an XOR Arbiter PUF with 8 XORs and bitlength of 512 is implementable by standard fabrication processes [28], but is currently beyond the reach of our attacks. Similar considerations hold for Lightweight PUFs of these sizes. Secondly, new design elements may raise the attacker's complexity further, for example adding nonlinearity (such as AND and OR gates that correspond to MAX and MIN operators [17]). Combinations of Feed-Forward and XOR architectures could be hard to machine learn too, partly because they seem susceptible only to different and mutually-exclusive ML techniques.

Moving away from delay-based PUFs, the exploitation of the dynamic characteristics of current and voltage seems promising, for example in analog circuits [29]. Also special PUFs with a very high information content (so-called SHIC PUFs [30, 31, 32]) could be an option, but only in such applications where their slow read-out speed and their comparatively large area consumption are no too strong drawbacks. Their promise is that they are naturally immune against modeling attacks, since all of their CRPs are information-theoretically independent. Finally, optical Strong PUFs, for example systems based on light scattering and interference phenomena [1], show strong potential in creating high input-output complexity.

Regarding view (ii), PUFs are different from classical cryptoschemes like RSA in the sense that increasing their size often likewise decreases their input-output stability. For example, raising the number of XORs in an XOR Arbiter PUF has an exponentially strong effect both on the attacker's complexity and on the instability of the PUF. We are yet unable to find parameters that increase the attacker's ef-

fort exponentially while affecting the PUF's stability merely polynomially. Nevertheless, one practically viable possibility is to increase the bitlength of XOR Arbiter PUFs, as discussed above. Future work will have to show whether the described large polynomial growth can persist in the long term, or whether its high degree can be diminished by further analysis.

Future Work.

The upcoming years will presumably witness an intense competition between codemakers and codebreakers in the area of Strong PUFs. Similar to the design of classical cryptoprimitives, for example stream ciphers, this process can be expected to converge at some point to solutions that are resilient against the known attacks.

For PUF designers, it may be interesting to investigate some of the concepts that we mentioned above. For PUF breakers, a worthwhile starting point is to improve the attacks presented in this paper through optimized implementations and new ML methods. Another, qualitatively new path is to combine modeling attacks with information obtained from direct physical PUF measurements or from side channels. For example, applying the same challenge multiple times gives an indication of the noise level of a response bit. It enables conclusions about the absolute value of the final runtime difference in the PUF. Such side channel information can conceivably improve the success and convergence rates of ML methods, though we have not exploited this in this paper.

Acknowledgements

This work was partly supported by the Physical Cryptography Project of the Technische Universität München.

9. REFERENCES

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026, 2002.

[2] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, page 160. ACM, 2002.

[3] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Controlled physical random functions. In *Proceedings of 18th Annual Computer Security Applications Conference*, Silver Spring, MD, December 2002.

[4] J. Guajardo, S. Kumar, G.J. Schrijen, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection. *Cryptographic Hardware and Embedded Systems-CHES 2007*, pages 63–80, 2007.

[5] B.L.P. Gassend. *Physical random functions*. Msc thesis, MIT, 2003.

[6] R. Pappu. *Physical One-Way Functions*. Phd thesis, MIT, 2001.

[7] P. Tuyls and B. Skoric. Strong Authentication with PUFs. *In: Security, Privacy and Trust in Modern Data Management, M. Petkovic, W. Jonker (Eds.)*, Springer, 2007.

[8] Ulrich Rührmair. Oblivious transfer based on physical unclonable functions (extended abstract). In Alessandro Acquisti, Sean W. Smith, and Ahmad-Reza Sadeghi, editors, *TRUST*, volume 6101 of *Lecture Notes in Computer Science*, pages 430–440. Springer, 2010.

[9] G.E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. *Proceedings of the 44th annual Design Automation Conference*, page 14, 2007.

[10] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Lightweight secure pufs. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 670–673. IEEE Press, 2008.

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

[12] 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 *Proceedings of the IEEE VLSI Circuits Symposium*, pages 176–179, 2004.

[13] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration Systems*, 13(10):1200, 2005.

[14] Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. Initial sram state as a fingerprint and source of true random numbers for rfid tags. In *In Proceedings of the Conference on RFID Security*, 2007.

[15] S.S. Kumar, J. Guajardo, R. Maes, G.J. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *IEEE International Workshop on Hardware-Oriented Security and Trust, 2008. HOST 2008*, pages 67–70, 2008.

[16] P. Tuyls, G.J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters. Read-proof hardware from protective coatings. *Cryptographic Hardware and Embedded Systems-CHES 2006*, pages 369–383, 2006.

[17] Daihyun Lim. *Extracting Secret Keys from Integrated Circuits*. Msc thesis, MIT, 2004.

[18] Erdinç Öztürk, Ghaith Hammouri, and Berk Sunar. Towards robust low cost authentication for pervasive devices. In *PerCom*, pages 170–178. IEEE Computer Society, 2008.

[19] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Testing techniques for hardware security. In *Proceedings of the International Test Conference (ITC)*, pages 1–10, 2008.

[20] Jan Sölter. *Cryptanalysis of Electrical PUFs via Machine Learning Algorithms*. Msc thesis, Technische Universität München, 2009.

[21] C.M. Bishop et al. *Pattern recognition and machine learning*. Springer New York:, 2006.

[22] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE international conference on neural networks*, volume 1993, pages 586–591. San Francisco: IEEE, 1993.

[23] http://www.pcp.in.tum.de/code/lr.zip, 2010.

[24] T. Bäck. *Evolutionary algorithms in theory and*

practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, USA, 1996.

[25] H.P.P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc. New York, NY, USA, 1993.

[26] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 1:999–1000, 2010.

[27] C.H. Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[28] S. Devadas. Physical unclonable functions and secure processors. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES 2009)*, September 2009.

[29] G. Csaba, X. Ju, Z. Ma, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, and U. Rührmair. Application of mismatched cellular nonlinear networks for physical cryptography. In *12th IEEE CNNA - International Workshop on Cellular Nanoscale Networks and their Applications*. Berkeley, CA, USA, February 3 - 5 2010.

[30] U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, and G. Csaba. Applications of high-capacity crossbar memories in cryptography. *To appear in IEEE Transactions on Nanotechnology*, 2010.

[31] U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, and M. Stutzmann. Security applications of diodes with unique current-voltage characteristics. In *Lecture Notes in Computer Science*, volume 6052, Tenerife (Spain), January 25 - 28 2010. 14th International Conference on Financial Cryptography and Data Security, Springer.

[32] C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, and M. Stutzmann. Random p-n-junctions for physical cryptography. *Applied Physics Letters*, 96(172103), 2010.

Chapter 4

Efficient Power and Timing Side Channels for Physical Unclonable Functions

PUFs were originally believed, or at the least hoped, to be secure against many traditional physical attack forms, including invasive and side channel techniques. The modeling attacks presented in the last chapter did not diminish this belief, as they are a non-physical attack method. Indeed, physical attacks played only a little role in the first decade of PUF research.

This picture changed in recent years, when, among other things, the side channel resilience of PUFs was investigated more closely. Merli et al. [93] were the first to implement a side channel attack on a PUF's error correction module, and to discuss several other potential vulnerabilities of PUFs in theory. While their work is no direct attack on the PUF itself, it at least targets a common PUF postprocessing method. Delvaux et al. [34] implement a direct side channel attack on PUFs that is based on noisy responses, which was originally suggested in theory in [134]. The same authors in Delvaux et al. [35] advance their method, and intentionally change the PUF's temperature to improve attack efficiency, devising the first PUF fault injection attacks. While being very innovative, the attacks of Delvaux et al. [34, 35] as of yet do not perform better than standard machine-learning based modeling, i.e., better than modeling *without* side channel info (compare [134, 138]).

This chapter deals with PUF side channel attacks, too, and takes their efficiency one step further. More concretely, we present the first *power* and *timing* side channels on PUFs in the field. They combine side channel information with newly adapted machine learning algorithms. One of their central achievements is to reduce the complexity of machine learning attacks on XOR-based Arbiter PUFs not only quantitavely, but qualitatively, i.e., from *exponential* (without side channel information [134, 138]) to *polynomial*. Our method historically is the first Strong PUF side channel that strongly improves attack performance compared to non-physical, pure modeling attacks [134, 138].

The presented material has just been accepted to the venue *Cryptographic Hardware and Embedded Systems (CHES)* prior to the submission of this thesis:

- U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, W. Burleson: *Efficient Power and Timing Side Channels for Physical Unclonable Functions*. 16th Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2014. Lecture Notes in Computer Science, Springer, 2014 (to appear).

Two other related recent publications by the candidate on this topic, which are explicitly not included in this thesis, are

- A. Mahmoud, U. Rührmair, M. Majzoobi, F. Koushanfar: *Combined Modeling and Side Channel Attacks on Strong PUFs*. IACR Cryptology ePrint Archive, Report 2013/632, 2013,

as well as

- U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, W. Burleson: *Power and Timing Side Channels for PUFs and their Efficient Exploitation*. IACR Cryptology ePrint Archive, Report 2013/851, 2013.

Efficient Power and Timing Side Channels for Physical Unclonable Functions

Ulrich Rührmair [◇, ⋆], Xiaolin Xu [†, ⋆], Jan Sölter [□], Ahmed Mahmoud [◇], Mehrdad Majzoobi [‡], Farinaz Koushanfar [‡], and Wayne Burleson [†]

[◇] Technische Universität München, 80333 München, Germany
[†] University of Massachusetts Amherst, Amherst, MA 01003, USA
[□] Freie Universität Berlin, 14195 Berlin, Germany
[‡] Rice University, Houston, TX 77005, USA

ruehrmair@in.tum.de, xiaolinx@umass.edu, jan_soelter@yahoo.com
ahmed.mahmoud@tum.de, m.majzoobi@gmail.com
fk1@rice.edu, burleson@umass.edu

Abstract. One part of the original PUF promise was their improved resilience against physical attack methods, such as cloning, invasive techniques, and arguably also side channels. In recent years, however, a number of effective physical attacks on PUFs have been developed [17, 18, 20, 8, 2]. This paper continues this line of research, and introduces the first power and timing side channels (SCs) on PUFs, more specifically on Arbiter PUF variants. Concretely, we attack so-called XOR Arbiter PUFs and Lightweight PUFs, which prior to our work were considered the most secure members of the Arbiter PUF family [28, 30]. We show that both architectures can be tackled *with polynomial complexity* by a combined SC and machine learning approach.
Our strategy is demonstrated in silicon on FPGAs, where we attack the above two architectures for up to 16 XORs and 512 bits. For comparison, in earlier works XOR-based Arbiter PUF designs with only up to 5 or 6 XORs and 64 or 128 bits had been tackled successfully. Designs with 8 XORs and 512 bits had been explicitly recommended as secure for practical use [28, 30].
Together with recent modeling attacks [28, 30], our work shows that unless suitable design countermeasures are put in place, no remaining member of the Arbiter PUF family resists all currently known attacks. Our work thus motivates research on countermeasures in Arbiter PUFs, or on the development of entirely new Strong PUF designs with improved resilience.

Key words: Physical unclonable functions (PUFs), side-channel attacks, power side channel, timing side channel, modeling attacks, machine learning, hardware security

1 Introduction

One part of the original PUF promise was their improved resilience against many classical attack forms, in particular physical attacks. This included cloning, invasive techniques, and arguably also side channels (SC). Regarding the latter, recall that Strong

⋆ These two authors contributed equally.

PUF based identification schemes [22] do not require a standard key that is processed bit by bit, a fact that arguably led to hopes about improved SC resilience within the community.

Recent years have put these assumptions to the test, but sometimes with a negative outcome. Let us start with non-physical attacks: Firstly, machine learning (ML) based modeling attacks have proven a more efficient threat than originally assumed. When the first of these attacks were put forward in 2004 [9], it was supposed that they could be thwarted by adding simple non-linear elements to Arbiter PUF designs, for example XOR gates or feed-forward loops. However, by improved ML algorithms, Rührmair et al. in 2010 and 2013 [28, 30] also tackled XOR-based Arbiter PUFs up to 64 or 128 bits and 5 XORs, and Feed-Forward Arbiter PUFs up to essentially arbitrary sizes. As a second, non-physical attack form, PUF protocol attacks have been devised in recent years. Since they are not in the focus of this work, we refer interested readers to the literature on this topic [26, 25].

Also dedicated physical attacks on PUFs have been devised lately. For example, the physical unclonability of PUFs, one of their core properties, has been investigated more closely. It is obvious that complex three-dimensional objects like PUFs cannot be cloned atom by atom by current fabrication technology. Generating a *perfect clone* thus to date is infeasible. However, *functional clones* are easier to construct, i.e., PUFs that merely agree with the original in their challenge-response behavior. In a breakthrough effort, Helfmeier et al. [8] in 2013 were indeed able to functionally clone SRAM PUFs by tuning the power-up states of SRAM cells. Soon after, invasive attacks on SRAM PUFs have been presented by Nedospasov et al. [20] in 2013. The authors apply semi-invasive, single-trace, backside readout of logic states to to obtain the responses of SRAM PUFs. This compromises any secret keys that would be derived from these responses.

Around the same time, first side-channel attacks on PUFs have been investigated. In 2011, Merli et al. [17] demonstrated SC attacks on the error correcting (EC) module of PUFs. Their attack is indirect in the sense that it does not target the PUF itself, but a specific EC module of the PUF, working only for certain modules. Furthermore, Merli et al. reported electromagnetic analyses on ring oscillator PUFs in two consecutive works in 2011 and 2013 [18, 19]. Also in 2013, Delvaux et al. [2] exploited the instabilities of Arbiter PUF responses as side channel, implementing an idea originally suggested by Rührmair et al. in [28]. While the work of Delvaux et al. is quite fascinating due to the fact that it does not use any machine learning algorithms, it must be said that it performs slightly worse than pure machine-learning based modeling *without* side channels [28, 30, 2].

We continue this line of research, and introduce in this paper the first power and timing side channel attacks on PUFs. Our approach constitutes one of the first physical attacks on Strong PUFs [24, 27, 29] that can *notably increase* attack performance in comparison with existing, non-physical methods, specifically with pure modeling attacks [28, 30].

In greater detail, we devise power and timing SCs for XOR Arbiter PUFs and Lightweight PUFs that provide the adversary with information about the *cumulative* number of zeros and ones in the outputs of the k parallel Arbiter PUFs before the XOR

gate. We then adapt existing machine learning (ML) techniques to efficiently exploit this information. This "hybrid" attack form can tackle XOR Arbiter PUFs and Lightweight PUFs with a *polynomial complexity* in their number of XORs, bitlengths, and number of required CRPs, while pure modeling attacks on these two PUFs have *exponential complexity* [28, 30]. We provide a full proof of concept on FPGAs, attacking XOR Arbiter PUFs and Lightweight PUFs for up to 16 XORs and 512 bits. Comparably large sizes of these two PUFs had hence never been realized before in silicon; in earlier works, already XOR Arbiter PUFs with 8 XORs and 512 bits had been explicitly suggested as secure [28, 30].

Organization of this Paper Section 2 provides the necessary background and methodology. Sections 3 and 4 describe the design and implementation of our power and timing side channels, respectively. Section 5 details our adaptation of logistic regression to incorporate SC information. Section 6 lists silicon results on FPGA implementations and provides an asymptotic peformance analysis. We conclude the paper in Section 7.

2 Background, Methodology, and Definitions

Background on XOR Arbiter PUFs and Lightweight PUFs. Together with SRAM PUFs, the Arbiter PUF family [7, 31] is arguably the best studied PUF design, and also the most popular implementation of so-called "Strong PUFs" [24, 27]. Nevertheless, a large number of its members have been attacked successfully by so-called modeling attacks in recent works [28, 30]. The currently *only* remaining Arbiter PUF variants which partly resist modeling, since they cause exponential modeling efforts (i.e., exponential training times of the ML algorithm), were so-called XOR Arbiter PUFs [9, 31] and Lightweight PUFs [11].

In an XOR Arbiter PUF, k Arbiter PUFs are used in parallel, and the same, multi-bit challenge is applied to all of them. The final, one-bit response is defined as the XOR of all the parallel k outputs [9, 31]. In a Lightweight PUF [11, 28], again k Arbiter PUFs are used in parallel, but different challenges $C^1, \ldots, C^k$ are applied to them, all of which are generated by some "input mapping" from a single, global challenge C (see [11] for the details of the mapping). The k outputs of the single Arbiter PUFs are used (without error correction) as input to a postprocessing function, which XORs subsets of them together in order to produce an m-bit output string (see again [11] for details). From a machine learning and modeling perspective, the optimal bit security is achieved if *all* of the k outputs are XORed to produce a *single bit output* [28, 30]. Therefore earlier works [28, 30] focused exactly on this case and on this special architecture of the Lightweight PUF, and so do we in this paper. If nothing else, this evaluates the maximally achievable bit security in a Lightweight PUF architecture. Using the same Lightweight PUF variant as [28, 30] also allows a fair comparison with our results.

FPGA Implementations. We implemented the above XOR Arbiter PUFs and Lightweight PUFs on Xilinx Spartan-6 FPGAs. In order to balance FPGA routing asymmetries, a lookup table (LUT) based programmable delay line (PDL) has been implemented [13, 10, 15]. This is the standard approach for realizing Arbiter PUFs on FPGAs,

and ensures a balanced output between zeros and ones in each single Arbiter PUF. For each CRP, majority voting over five repeated measurements of the response to the same challenge was performed in order to determine the final response. The challenges were generated by an n-bit maximal-length linear feedback shift register (LFSR) with polynomial $f = 1 + x^1 + x^3 + x^4 + x^{64}$.

Machine Learning Definitions and Computational Resources. Following [28, 30], we use the following definitions throughout the paper: The prediction error ϵ is the ratio of incorrect responses of the trained ML algorithm when evaluated on the test set. The prediction rate is $1 - \epsilon$. For all ML experiments throughout this paper, each test set consisted of 10,000 randomly chosen CRPs. The term N_{CRP} (or simply *"CRPs"*) denotes the number of CRPs employed in an attack, i.e., the size of the training set. We used an Intel Xeon X5650 processor at 2.67GHz with 48 GB of RAM in all of our ML experiments, having a value of a few thousand Euros. All computation times (= "training times") are calculated for one core of one processor of this hardware.

3 Power Side Channels on XOR-based Arbiter PUFs

3.1 Basic Idea of the Power Side Channel

Currently known pure modeling attacks on XOR-based Arbiter PUFs require training times of the ML algorithm that are exponential in the number of XORs [28, 30]. This makes it difficult to tackle XOR-based Arbiter PUFs with more than five or six single parallel Arbiter PUFs, and with bitlengths longer than 128, by pure modeling attacks [28, 30]. XOR-based Arbiter PUF architectures are therefore the currently most secure designs from the Arbiter PUF family. Our side-channel attacks now take a novel route: They gain additional information from the physical implementation of XOR-based Arbiter PUFs, and use this information to improve the ML computation times (i.e., training times) from exponential to polynomial.

One straightforward power side channel is to apply power (i.e., current) tracing to determine the transition from zero to one of the latches (i.e., the arbiter elements) in the single Arbiter PUFs. The power tracing is based on measuring the amount of current drawn from the supply voltage during any latch transition to one. We implemented a first SPICE simulation to validate this approach, and to verify the power consumption of an arbiter circuit with different loading outputs. Only one latch (i.e., arbiter circuit) is used in the simulation, but with three different outputs loading scenarios (i.e., floating output, output connected to one gate, and output connected to four gates). Figure 1 illustrates the results, and shows the different amount of current drawn for the three different output loading scenarios. The reason for having different values for the different loadings is that an additional amount of charges is required to charge the capacitance of each gate. Hence, the amount of drawn charges, which is the integration of the current curve, is linearly proportional with the number of loading gates. Taking this phenomenon into consideration, the amount of charges normally drawn in case of a floating load should be subtracted.

In XOR-based architectures with k parallel single Arbiter PUFs, the current that is drawn *in sum* and *altogether* in principle tells the (cumulative) number of latches

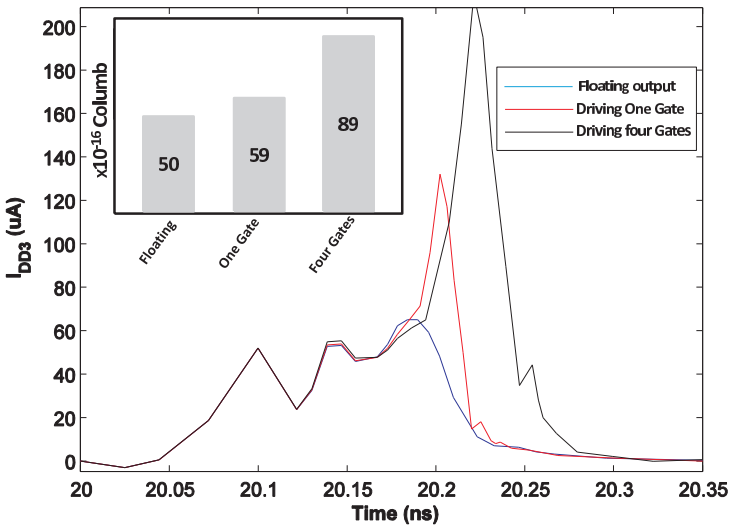

Fig. 1. The power tracking side-channel analysis for a latch that had a transition to 1, with different driving loads, in SPICE simulation. The inset is the amount of drawn charges, which is calculated from the area under each curve. The amount of charges is linearly proportional with the number of gates. The amount of charges normally drawn for a floating load should be subtracted.

that are zero, and the (cumulative) number that are equal to one. Please note, however, that it does *not* tell us *which* of the k parallel Arbiter PUFs had *which* output. If it did, CRPs from every single Arbiter PUF could be collected, and every single Arbiter PUF could be machine learned separately. As this is not possible, a more complicated strategy is required, in particular a way to exploit the cumulative number of zeros and ones beneficially in the ML process, as detailed in Section 5. But before we move on to the details of the ML process, we discuss the exact implementation of the side channels in this and the next section.

3.2 Practical Implementation of the Power Side Channel

Measurement Noise To further validate the practicality of our power SC, we had to move beyond the simplifications of SPICE simulations, most notably the absence of supply and measurement noise and real process variations. We extracted the power trace of 30 sub-response patterns from Lightweight PUFs on FPGA (see Figure 2). However, we found that the 30 power traces are difficult to be differentiated from each other (as are their power consumptions). In other words, in practical implementations, a straightforward identification of the desired power side channel information from the measured power (current) traces appears infeasible.

There are two reasons for this problem:

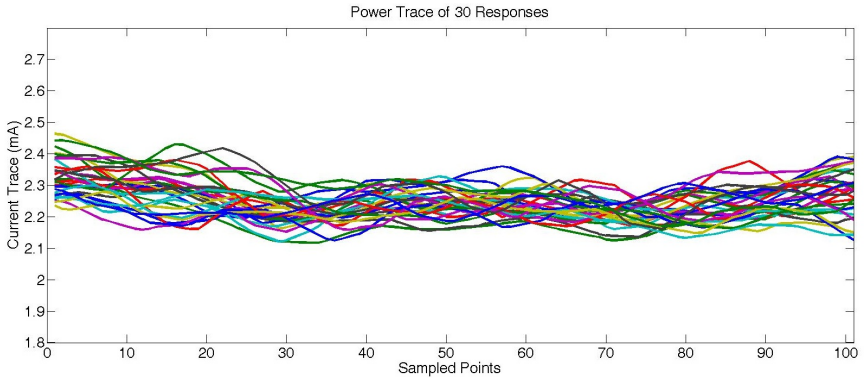

Fig. 2. Power trace of 30 different sub-responses, collected from FPGA, illustrating the difficulty of differentiating them from each other.

1. In real silicon Arbiter and Lightweight PUFs, the final XOR function usually consumes no more than 5% silicon resource of the whole design. Thus, it is difficult to extract the power consumption of XOR function, which consumes much less power compared with the whole circuits;
2. Unlike a simulated PUF, measuring real silicon PUF circuit is always impacted by the noise from supply voltage and measurement, which plays a negative role in extracting the desired power information.

To overcome this problem and maintain the feasibility of our power side channel, we developed a new, statistical signal processing strategy.

Our main objective is to extract the subtle power consumption of XOR gates and transform it into a recognizable format, which is correlated with the cumulative number of one or zero sub-responses. Even though the extra power consumed by active XOR gates is not directly extractable, it does really affect the whole power consumption. Thus, it should change the probability distribution functions (PDF) of the measured power leakage, if it can extract the probability distribution of leaked power information, the cumulative of one sub-responses can be inferred. For this purpose, we apply a "challenge-dependent responses estimation" method to calculate the PDF of every power trace collection.

The "challenge-dependent responses estimation" is implemented by comparing the power trace just before and after the generation of response to distinguish subtle changes. In the experiment, we measure the power trace of a single PUF response for totally m times, and record all of them in parallel. If denoting the generation time of the ith PUF response R_i as t_i, we can then filter out the two adjacent sections of power trace (length of which is T_i^{before} and T_i^{after}) just before and after time t_i. Assume that T_i^{before} = T_i^{after}, then we divide each time slice into n parts with the collected power trace (current trace) data. Based on the divided current trace data, we can calculate the power consumption of each n part before and after the generation of response R_i.

By denoting power consumption of all the $2*n$ parts of the ith PUF response under the lth measurement (totally m measurements are did as described above, thus, $l \in (1...m)$) as P_{lij}^{before} and P_{lij}^{after} respectively ($j \in (1...n)$), two matrices including the power consumption information of the i_{th} response are obtained as:

$$M_i^{before} = \begin{pmatrix} P_{11}^{before} & P_{12}^{before} & P_{13}^{before} & ... & P_{1n}^{before} \\ P_{21}^{before} & P_{22}^{before} & P_{23}^{before} & ... & P_{2n}^{before} \\ ... & ... & ... & ... & ... \\ P_{m1}^{before} & P_{m2}^{before} & P_{m3}^{before} & ... & P_{mn}^{before} \end{pmatrix} \tag{1}$$

$$M_i^{after} = \begin{pmatrix} P_{11}^{after} & P_{12}^{after} & P_{13}^{after} & ... & P_{1n}^{after} \\ P_{21}^{after} & P_{22}^{after} & P_{23}^{after} & ... & P_{2n}^{after} \\ ... & ... & ... & ... & ... \\ P_{m1}^{after} & P_{m2}^{after} & P_{m3}^{after} & ... & P_{mn}^{after} \end{pmatrix} \tag{2}$$

Based on the power trace processing above, we now denote the power information of a single PUF response with two matrix: M_i^{before} and M_i^{after}. Assuming that we totally collect K response bits, then the power consumption matrix for all responses can be described as (for brevity, "b" means before and "a" means after):

$$M^{before/after} = \begin{pmatrix} M_1^{b/a} & M_2^{b/a} & M_3^{b/a} & ... & M_K^{b/a} \end{pmatrix} \tag{3}$$

Due to the existence of environmental and measurement noise, the m parallel segmentations of measured power trace (such as $P_{11}^{before}, P_{21}^{before} ... P_{m1}^{before}$ in Equation 1, and $P_{11}^{after}, P_{21}^{after} ... P_{m1}^{after}$ in Equation 2) consumption would build n PDF respectively. Since we divide power trace slice into 2 parts (before and after), thus totally $2*n$ PDF are generated for each response. As we discussed, though there is no directly leaked power information that we can extract for the XOR function, it impacts the probability distribution of the whole power trace. To convert the PDF information into the cumulative number of one and zero responses, we applied histograms method to describe the PDF, and then implement basic calculus computation to get the cumulative distribution function (CDF):

$$C_j^{before/after}(x) = \sum_{x_j<x} PDF(X = x_j) = \sum_{x_j<x} p(x_j) \qquad j \in (1..n) \tag{4}$$

Based on Equation.4, the original leaked power information can be transformed as CDF. To filter out the difference between two power trace segments: before and after time t_i, and erase the impact of environmental and measurement noise, we then calculate the mean-squared-error (MSE) following Equation 5:

$$MSE_j = E[(C_j^{before}(x) - C_j^{after}(x)^2],\ j \in (1..n) \tag{5}$$

then, all of the n MSEs are summed up for a final sub-response estimation: E_i, which reflects and amplifies the impact of active XOR gates on leaked power:

$$E_i = \sum_{j=1}^{n} MSE_j,\ j \in (1..n) \tag{6}$$

With the proposed "challenge-dependent responses estimation" method, the power trace of different challenge-dependent responses patterns are transformed into an estimated value: "E_i". Thus, we can deduce the pattern of CRPs and integrate them within our proposed ML attacks.

Determining the Generation time of PUF Response In the previous paragraph, we applied the "challenge-dependent responses estimation" method to extract the power side channel information of active XOR gates, assuming that we know the generation time of the ith PUF response R_i as t_i. However, one additional problem is that in practice, t_i is not a direct known parameter. In this last paragraph, we will now detail how we overcame this final problem.

If we randomly set a t_{i_random} as the generation time of response R_i, the power information of a certain PUF response R_i can be described as:

$$P_i = P_{i\_noise}^{before} + P_{i\_oc}^{before} + P_{i\_XOR}^{before} + P_{i\_noise}^{after} + P_{i\_oc}^{after} + P_{i\_XOR}^{after}, \tag{7}$$

where $P_{i\_noise}^{b/a}$ denotes the environmental and measurement noise (as before, "b" abbreviates before and "a" after t_{i_random} here), $P_{i\_oc}^{b/a}$ stands for the power consumption of "other circuitry", again before and after t_{i_random}, while $P_{i\_XOR}^{b/a}$ denotes the similar power information of XOR functional circuitry. Since based on the measurement, we can roughly tell the range of a PUF response generation time, we would have several choices of t_{i_random}. To determine the exact generation time of each PUF responses, we move the t_{i_random} in the approximate time range, then we will get different power side channel informative patterns.

Since the PUF circuitry are measured for multiple times, and under the same environment, we can assume that for each response, we will have:

$$P_{i\_noise}^{before} \approx P_{i\_noise}^{after} \ \ and \ \ P_{i\_oc}^{before} \approx P_{i\_oc}^{after} \tag{8}$$

thus, if we measure the power trace of a single PUF response for multiple times, we get:

$$\sum P_{i\_noise}^{before} - \sum P_{i\_noise}^{after} \approx 0 \ \ and \ \ \sum P_{i\_oc}^{before} - \sum P_{i\_oc}^{after} \approx 0 \tag{9}$$

Based on this algorithm, it is clear that only when t_i is set as the correct generation time, the E_i in Equation 6 is maximized.

4 Timing Side Channels on XOR-based Arbiter PUFs

As with our power side channel, the objective of the timing side channel is providing additional information about the individual response bits (i.e., PUF output bits) even though the response bits are XOR'ed together for providing the output. Assume that k response bits $\{r_1, \ldots, r_k\}$ are XOR'ed to form a single output bit b_{out}. (Note that a k-input XOR shall consist of several stages of smaller XOR gates. For the sake of demonstration, assume that the delay of the response bit r_i, denoted by t_{r_i} follows a certain order, say $t_{r_1} \leq t_{r_2} \cdots \leq t_{r_{k-1}} \leq t_{r_k}$). Our timing side-channel approach is based on a delay measurement circuit, which can be used to characterize the delay length of different patterns of k response bits $\{r_1, \ldots, r_k\}$.

4.1 Timing Characterization Method

Every ASIC manufactured chip undergoes a set of structural and functional tests which measure/ evaluate the IC's physical and logical properties respectively. Measuring the delay of certain combinational paths in the circuit is a part of standard structural testing. Since the internal combinational paths are typically inaccessible, the timings are indirectly inferred from the FF outputs using clock sweeping. The FF values can be set using a testing scan-chain while all the FFs are connected to the global chip clock. The pertinent chip is referred to as Circuit Under Test (CUT). The frequency of this clock is swept in a continuous monotonic fashion from a high to low value while the path under measurement is toggled using the logic at the input FF. When the frequency is higher than the path delay, the output FF does not have enough time to settle which is called a "fail". Once the frequency approaches the path delay, the output FF sets to the correct value (from the initial reset dictated by the scan chain) which is the "pass" state. The frequency at which this transition occurs denotes the path delay and this overall testing method is called pass/fail timing test.

On our FPGA testbed, the pass/fail timing tests have to be implemented by reconfiguration. We adopt the measurement circuitry from [14, 15] that is demonstrated in Figure 3. Note that because of the timing uncertainty around the FF metastability point, the toggle between the pass/fail states appears with a certain property. Thus, error density estimation followed by smoothing methods are used for inferring the exact toggle point from a set of stochastic measurements.

To estimate the probability of error at a certain clock frequency, an error histogram accumulator is realized using two counters. The first one is an error counter whose value increments by one each time an error occurs. The second one counts the clock cycles; after 2^N clock cycles, this counter clears (resets) the error counter and then restarts again, where N is the binary counters' size. The error counter value is stored in the memory one clock cycle before it is reset. Now, the stored number of fails normalized to N would yield the error probability value for each target frequency.

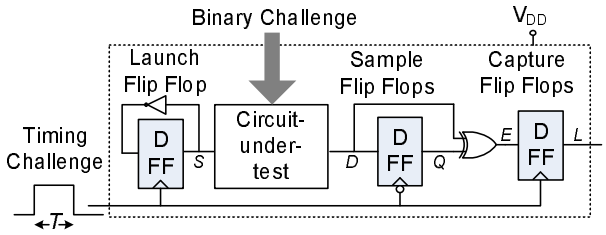

Fig. 3. The timing signature extraction circuit.

Next, we linearly and continually sweep the input clock frequency: in T_{sweep} seconds from $f_i = \frac{1}{2T_i}$ to $f_t = \frac{1}{2T_t}$, where $T_t < t_p < T_i$. For each frequency sweep, a separate set of registers count the number of clock pulses. We use this counter as an accurate timer which records the frequency of the timing errors. This counter value is

retrieved every time the content of the error counter is written into memory. The system described above can be configured and utilized for extracting the delays of any CUT implemented on FPGA. We use this adaptation of pass/fail timing test to measure the delay between the FF storing the challenge input, to the output of the PUF which shall be stored in an output register. To prevent attacks, this output is measured after XOR-ing the arbiter values. Note that the scanning for extracting delay values could also be performed in parallel to reduce the characterization time [14, 15].

4.2 Characterization Accuracy

The resolution of the delay measurement, i.e., the measured delay's accuracy, is a function of a few factors: (i) the clock noise and skew, (ii) the sweeping frequency resolution, and (iii) the number of pulses at each frequency. The output of the characterization circuit is a binary zero/one (pass/fail) value. A real-valued output can be measured by repeating several (same width) clock pulses to the circuitry and accumulating the number of ones at the output. The resulting value, when normalized, shows the probability at which the timing errors occur for each input clock's pulse width. The more the input clock pulse is repeated, a higher sampling resolution and accuracy can be achieved.

For now assume that the clock pulse (of width T) is sent to the CUT for M times. Because of clock skew and phase noise, the characterization circuitry receives a clock pulse with width $T_{eff} = T + T_j$, where T_j is the additive jitter. Suppose that T_j is a random variable with a zero mean and symmetric distribution around its mean. The output probability is a continuous and smooth function of T_{eff}; thus, approximating the probability by averaging shall be an asymptotically unbiased estimator as $M \to \infty$. Lastly, the minimum measurable timing is a function of the maximum clock speed at which the FFs can be run (maximum clock frequency). During a linear frequency sweep, a longer sweep time increases both items (ii) and (iii) and thus the characterization accuracy.

4.3 Parameter Extraction

Thus far, we have described a system that measures the probability of timing errors for various clock pulse widths. The error probability can be fully represented by a set of few parameters; the parameters are directly related to the CUT delay and FF setup and hold times. It can be shown that the probability of timing errors shall be written as the sum of shifted Gaussian CDFs [14, 15]. The central limit theorem can determine the Gaussian nature of the error probabilities which can be explained by Equation 10 showing the parameterized error probability function.

$$f_{\mathbf{D},\Sigma}(t) = 1 + 0.5 \sum_{i=1}^{|\Sigma|-1} -1^{\lceil i/2 \rceil} \left[Q(\frac{t - d_i}{\sigma_i}) \right] \tag{10}$$

where $Q(x)$=$\frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right)$ and $d_{i+1} > d_i$. To estimate the timing parameters, f is fit to the set of measured data points (t_i,e_i), where e_i is the error value recorded when the pulse width is t_i.

4.4 Side Channel Timing Analysis of XOR'ed Outputs

The pass/fail timing measurement above is able to estimate the delay of the overall PUF path (after XOR'ing). As we sweep the clock, we eventually get to a stable regime, i.e., the regime where the overall output does not change any more. However, before getting to this stable regime, there are clock periods for which only a few XOR inputs (i.e., response bits) change. Sweeping the clock frequency could yield the information about the approximate timing of the XOR inputs: every time one of the inputs to the XOR network, i.e., an arbiter output, changes, there will be a toggle. Even though it is not possible to distinguish the response bit that has changed, it is possible to estimate the number of flipping XOR inputs with a good probability. This number shall be vague if the timings of two or more response bits coincide. Since the probability of such a coincidence is rather low, in most instances clock sweeping shall yield an approximation of the number of flipped XOR inputs, i.e., the cumulative number of zeros and ones among the single Arbiter PUF responses $r_1, \ldots, r_k$.

5 Adapting Machine Learning Algorithms to Side Channel Information

The question how (and if at all) SC information on the cumulative number of zeros and ones can be efficiently exploited in PUF modeling turned out to be highly non-trivial. Eventually, we found a gradient based optimization similar to the logistic regression (LR) algorithm of [28, 30]. The following treatment assumes some familiarity with this algorithm and with the work in [28, 30].

Let $r_i(C) \in \{0, 1\}$ be the output of the i^{th} Arbiter PUF within a k-XOR Arbiter PUF (or within a Lightweight PUF with k parallel Arbiter PUFs) to a challenge C. The side-channel information then yields the number n of individual Arbiter PUFs with output one: $n = \sum_i r_i(C)$. It lies in contrast to the general setting of binary outputs in LR on an interval scale. Therefore, instead of optimizing the binary class probabilities [28, 30], we rely on minimizing the squared error between a side-channel model $f(\boldsymbol{w}, C)$ and the actual outputs n:

$$l(\mathcal{M}, \boldsymbol{w}) = \sum_{(C,\, t) \in\, \mathcal{M}} (f(\boldsymbol{w}, C) - n)^2.$$

The corresponding gradient

$$\nabla l(\mathcal{M}, \boldsymbol{w}) = \sum_{(C,\, r) \in\, \mathcal{M}} 2\left(f\left(\boldsymbol{w}\right) - n\right) \nabla f(\boldsymbol{w}) \tag{11}$$

is highly similar to the gradient in LR. We again applied the RProp update scheme (as in [28, 30]) to find a solution $\hat{\boldsymbol{w}}$ with minimal error l.

Assuming the standard linear additive delay model [9, 6, 28, 30], one obtains the following model of the side-channel information:

$$f(\boldsymbol{w}, C) = \sum_i \Theta(\boldsymbol{w}_i^T \boldsymbol{\Phi}_i).$$

Note that the model only depends on the direction, but not on the length $\|\boldsymbol{w}_i\|$ of the weight vectors. That is, any two solutions $\boldsymbol{w}_i$ and $\alpha\boldsymbol{w}_i, \alpha \in \mathbb{R}^+$ are equivalent. Therefore we might substitute the Heaviside function by the differentiable logistic sigmoid $\sigma(x) = (1 + e^{-x})^{-1}$ to enable gradient based optimization. This is a reasonable substitution as $\lim_{\|\boldsymbol{w}\|\to\infty} \sigma(\boldsymbol{w}^T\boldsymbol{\Phi}) = \Theta(\boldsymbol{w}^T\boldsymbol{\Phi})$ and, as noted above, a valid solution is unaffected by scaling of $\boldsymbol{w}$.

As this substitution makes the model differentiable, we obtain the following gradient to insert in Equation 11:

$$\nabla f(\boldsymbol{w}_j) = \sigma(\boldsymbol{w}_j^T\boldsymbol{\Phi}_j)(1 - \sigma(\boldsymbol{w}_j^T\boldsymbol{\Phi}_j))\boldsymbol{\Phi}_j. \tag{12}$$

This gradient of an individual Arbiter PUF's weight vector $\boldsymbol{w}_j$ depends only on the value of the weight vector itself, being in strong contrast to the case without side-channel information [28, 30]. The decoupling of individual Arbiter PUF updates thus drastically simplifies the ML problem, provided that side-channel information is available.

In addition to the above new regression, we applied a two step optimization methodology: First we optimized the PUF model based on the above process and gradient, using the side-channel information, until a fraction of $f = 0.95$ percent of the final XOR Arbiter output was correctly reproduced. Secondly, we further refined and optimized the model with the "standard" LR algorithm applied in [28, 30] for 1000 iterations. This led to very low error rates around 2% or below. For all experiments, we used hundred times more CRPs than free parameters in the model, i.e.,

$$N_{CRP} \approx 100 \times \text{bitlength} \times \text{no. of XORs}.$$

Note that the above equation merely describes a linear CRP consumption in the problem parameters. This is in stark contrast to the exponentially growing complexities of pure ML attacks on XOR Arbiter and Lightweight PUFs [28, 30].

While our approach in the first step of the above methodology mostly converged to the global minimum, in a few cases it got stuck (i.e., the performance after 5000 iterations was worse than 5% remaining missclassifications). In this case, we restarted the algorithm with a different random initialization of $\boldsymbol{w}$.

6 Results and Asymptotic Performance Analysis

We applied our adapted ML methods (see Section 5) to CRP data and SC information gathered from FPGAs (see Sections 2, 3, and 4), both for power and timing SCs. The results are presented in Tables 1 and 2. The attacks perform extremely efficiently, as we were able to successfully attack XOR Arbiter PUFs and Lightweight PUFs for up to 16 XORs and for bitlengths of up to 512 (timing SCs) and 128 (power SCs). No implementations of comparable sizes of these two PUFs in silicon had ever been considered or reported before. Furthermore, pure modeling attacks thus far had only been able to tackle the two PUFs for up 5 or 6 XORs and bitlength 64 [28, 30]. Both facts illustrate the impact and reach of our new method.

Tables 1 and 2 already indicate that the CRP requirments and computation times grow very mildly, with the same holding for the prediction errors. In order to quantify

No. of XORs	Bit Length	CRPs ($\times 10^3$)	Prediction Rate XOR Arb. PUF	Training Time XOR Arb. PUF	Predict. Rate LW PUF	Training Time LW PUF
8	64	26	98.5%	2 min	98.5%	1 min
	128	51.6	97.5%	12 min	98.2%	9 min
	256	103	97.7%	1:35 hrs	97.8%	1:00 hrs
	512	205	97.4%	16:50 hrs	97.5%	3:30 hrs
12	64	39	98.1%	16.5 min	98.5%	2 min
	128	77.4	97.4%	38.5 min	97.9%	24.1 min
	256	154.5	97.1%	3.8 hrs	97.3%	1.75 hrs
	512	308	96.92%	56.25 hrs	97.11%	9.55 hrs
16	64	52	98%	37 min	98%	7 min
	128	103.2	97.5%	2 hrs	97.5%	51.7 min
	256	206	97.3%	15.1 hrs	96.9%	4.8 hrs
	512	410	96.5%	102 hrs	96.7%	20.2 hrs

Table 1. Effectiveness of *timing* side-channel attacks on the XOR Arbiter PUF and Lightweight PUF (LW PUF), all carried out on FPGA implementations.

No. of XORs	Bit Length	CRPs ($\times 10^3$)	Prediction Rate XOR Arb. PUF	Training Time XOR Arb. PUF	Predict. Rate LW PUF	Training Time LW PUF
8	64	26	98.1%	3 min	98.4%	1.25 min
	128	51.6	98%	13 min	98.1%	9.25 min
12	64	39	98.3%	11 min	98.2%	3.5 min
	128	77.4	97.3%	47 min	97.8%	25 min
16	64	52	98%	38 min	98%	6.5 min
	128	103.2	97.5%	2:28 hrs	97.5%	46.5 min

Table 2. Effectiveness of *power* side-channel attacks on the XOR Arbiter PUF and Lightweight PUF (LW PUF), all carried out on FPGA implementations.

this with yet more data points, we conducted comprehensive ML experiments on simulated CRPs and simulated SC data. The CRPs were generated by the linear additive delay model (LADM), similarly as in earlier ML experiments [28, 30]. We executed these simulated attacks on XOR Arbiter PUFs and Lightweight PUFs for 2, 3, ..., 16 XORs, and with 64, 128, 256 and 512 bits. This means that we treated $2 \cdot 15 \cdot 4 = 120$ different architectures in sum, investing hundreds of hours of computation time. The generated data points are shown in Figure 4, and fully confirm the suspected mild, actually cubic growth. For those cases where we also had silicon data for comparison (see Tables 1 and 2), the silicon and the simulated attacks performed very similarly, confirming both earlier conjectures [6, 28, 30] on the validity of the additive linear delay model, as well as the accuracy of our side-channel measurements. The empirically estimated computational complexity of our attacks is hence $O(n^3)$, or, in other words, low-degree polynomial, in the problem size. Furthermore, as indicated already in Section 5, the number of used/required CRPs is merely linear in the same parameter.

Two important aspect should not go unnoticed. Firstly, our power side channel is

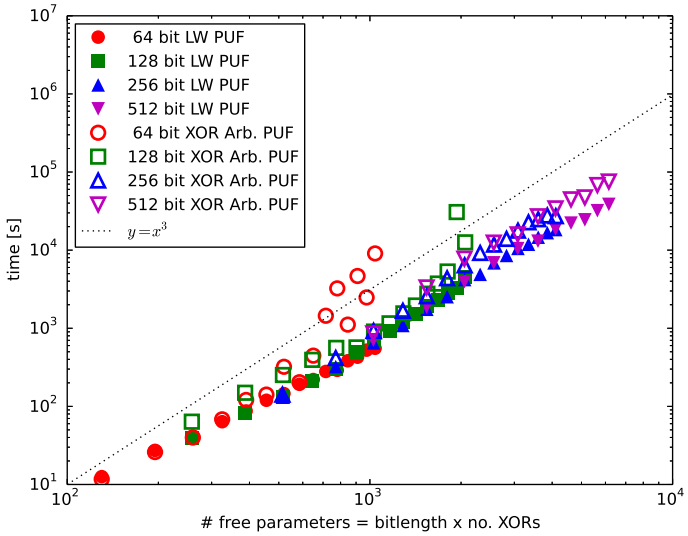

Fig. 4. The training times for our ML-algorithm on Lightweight PUFs (LW PUFs) and XOR Arbiter PUFs on a logarithmic scale. They show that the computational complexity regarding training times is cubic, i.e., $O(x^3)$.

more noisy than the timing side channel. This had the effect that we could only handle bit lengths of up to 128 by use of the power SC. Improved, less noisy versions seem possible, but also non-trivial, and are left to future work.

Secondly, in the presence of side-channel information, our ML algorithms perform slightly faster on Lightweight PUFs than on XOR Arbiter PUF. Without side channels, the converse effect has been observed [28, 30]. Intuitively, the challenge input mapping of the Lightweight PUF creates a more diverse and stable information basis for the ML algorithm, which leads to faster convergence. A full, rigorous mathematical analysis of this effect will be conducted in future work.

7 Summary and Conclusions

In this paper, we introduced and implemented the first power and timing side channels (SCs) on PUFs, more precisely on XOR Arbiter PUFs and Lightweight PUFs. These two PUF designs were chosen by us due to their particular relevance: The Arbiter PUF family is arguably the most studied electrical Strong PUF design, and said two PUFs are the most secure representatives of this family according to recent work [28, 30]. Our two SCs consisted of (i) power tracing of the arbiter element (i.e., the latch) in Arbiter PUFs, and (ii) marking different response patterns with corresponding timing signatures. Both SCs tell us the *cumulative* number of zeros and ones in the outputs of the k parallel

Arbiter PUFs within XOR-based Arbiter PUF variants, such as the XOR Arbiter PUF or the Lightweight PUF. One main obstacle in exploiting the above SCs efficiently was that the attacker does not learn *which* of the single Arbiter PUF outputs is zero or one. This makes the cumulative information worthless at first sight. However, we were able to devise adapted, tailor-made ML algorithms, which can exploit the information very efficiently.

We carried out a full silicon proof of concept on FPGAs, attacking the two above PUFs for up to 16 XORs and bitlengths of 512 bits (by timing SCs) and 128 bits (by power SCs). Their smaller noise levels made timing SCs the yet more efficient tool, even though improved future versions of the power side channels seem possible. Interestingly, XOR-based Arbiter PUF variants had never even been *implemented* (left alone attacked) for comparable sizes in the literature, since already versions with 8 XORs and 512 bits had been recommended as practically secure against known attacks in earlier works [28, 30]. This may illustrate the relevance and strength of our results. A close asymptotic analysis on simulated CRP data furthemore showed that our attacks have only *cubic complexity*. This is a drastic improvement over the exponential complexity of state-of-the-art, pure modeling attacks [28, 30].

Our methods are the first physical attacks on Strong PUFs, i.e., on PUFs with many CRPs, that can notably increase attack performance. Overall, they imply that *as long as no suitable design countermeasures are put in place*, no currently existing architecture from the Arbiter PUF family can withstand all known attacks: "Standard" Arbiter PUFs as well as Feed-Forward Arbiter PUFs have been attacked by pure modeling attacks with polynomial complexity [28, 30]; and XOR-based variants such as the XOR Arbiter PUF and the Lightweight PUF are susceptible to the methods presented in this paper, which have polynomial complexity, too.

We did not explicitly deal with design countermeasures in this paper for space reasons. However, one conceivable strategy against power SCs could consist of using two symmetric, inverted output signals with two latches. This construction could neutralize and balance power consumption, regardless of the PUF's output. Interestingly, this could even be used to detect and stabilize output errors in Arbiter PUF variants, even though we did not follow this route in in this paper. Countermeasure against our timing SCs would probably have to focus on the construction of an isochronous hardware. Implementing such strategies is left to future, follow-up works.

We believe that the PUF attacks presented in this and other papers should be interpreted in a balanced fashion. None of them "kills" the field in its entirety. In our opinion, they are part of a natural consolidation process in the PUF area, similar to the consolidation that classical security primitives have undergone already some time ago. The occurence of this process could be seen as indication that the field is becoming increasingly mature. One typical byproduct is the insight that certain aspects are not as simple as originally believed, which may be disappointing at first sight. Overall, however, a sound consolidation will be beneficial to the field, eventually creating more research opportunities than it destroys. This paper could be seen as one (of many) steps within this process.

Acknowledgements

The work at the University of Massachusetts Amherst was supported in part by SRC task 1836.074, US NSF grants 0923313 and 0964641, and US DHHS grant 90TR0003/01. The work at Rice University was supported in part by NSF CCF-1116858:SHR:Small, NSF CNS-1059416:CI-ADDO-NEW: Trust-Hub, and ONR ONR N00014-11-1-0885 grants.

References

1. Christopher M. Bishop, Nasser M. Nasrabadi: *Pattern recognition and machine learning*. Springer, New York, 2006.
2. Jeroen Delvaux, Ingrid Verbauwhede: *Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise*. HOST 2013.
3. Jeroen Delvaux, Ingrid Verbauwhede: *Attacking PUF-Based Pattern Matching Key Generators via Helper Data Manipulation*. IACR Cryptology ePrint Archive, Report 2013/566.
4. Jeroen Delvaux, Ingrid Verbauwhede: *Key-recovery Attacks on Various RO PUF Constructions via Helper Data Manipulation*. IACR Cryptology ePrint Archive, Report 2013/610.
5. Jeroen Delvaux, Ingrid Verbauwhede: *Fault Injection Modeling Attacks on 65nm Arbiter and RO Sum PUFs via Environmental Changes.* IACR Cryptology ePrint Archive, Report 2013/619.
6. Srinivas Devadas: *Physical unclonable functions and secure processors*. Invited talk, Workshop on Cryptographic Hardware and Embedded Systems (CHES 2009), September 2009.
7. Blaise Gassend, Dwaine Clarke, Marten van Dijk, Srinivas Devadas: *Silicon physical random functions*. ACM Conference on Computer and Communications Security 2002: 148-160
8. Clemens Helfmeier, Dmitry Nedospasov, Christian Boit, Jean-Pierre Seifert: *Cloning Physically Unclonable Functions*. HOST 2013.
9. Daihyun Lim: *Extracting Secret Keys from Integrated Circuits*. MSc Thesis, MIT, 2004.
10. M. Majzoobi, F. Koushanfar and S. Devadas: *FPGA PUF using programmable delay lines*. IEEE Workshop Information Forensics and Security (WIFS), 2010.
11. Mehrdad Majzoobi, Farinaz Koushanfar, Miodrag Potkonjak: *Lightweight Secure PUFs.* ICCAD 2008: 607-673.
12. Mehrdad Majzoobi, Farinaz Koushanfar, Miodrag Potkonjak: *Testing techniques for hardware security.* In Proceedings of the International Test Conference (ITC), pages 1-10, 2008.
13. M. Majzoobi, F. Koushanfar and M. Potkonjak: *Techniques for Design and Implementation of Secure Reconfigurable PUFs*. ACM Trans. Reconfigurable Technology and Systems, vol. 2, no.1, 2009.
14. M. Majzoobi, E. Dyer, A. Elnably, F. Koushanfar: *Rapid FPGA Characterization using Clock Synthesis and Signal Sparsity", International Test Conference (ITC)*. pp. 1-10, 2010.
15. M. Majzoobi, F. Koushanfar: *Time-Bounded Authentication of FPGAs*. IEEE Transactions on Information Forensics and Security (TIFS), vol. 6, issue 3, pp. 1123-1135, 2011.
16. M. Rostami, M. Majzoobi, F. Koushanfar, D. Wallach, S. Devadas: *Robust and Reverse-Engineering Resilient PUF Authentication and Key-Exchange by Substring Matching*. IEEE Transactions on Emerging Topics in Computing, 2014.
17. Dominik Merli, Dieter Schuster, Frederic Stumpf und Georg Sigl: *Side-Channel Analysis of PUFs and Fuzzy Extractors*. TRUST 2011.
18. Dominik Merli, Dieter Schuster, Frederic Stumpf, Georg Sigl: *Semi-invasive EM attack on FPGA RO PUFs and countermeasures*. ACM Workshop on Embedded Systems Security (WESS'11), 2011.

19. Dominik Merli, Johann Heyszl, B. Heinz, Dieter Schuster, Frederic Stumpf, Georg Sigl: *Localized electromagnetic analysis of RO PUFs*. HOST 2013.
20. Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, Christian Boit: *Invasive PUF Analysis*. Fault Diagnonsis and Tolerance in Cryptography (FDTC'13), 2013.
21. Ravikanth Pappu: *Physical One-Way Functions*. PhD Thesis, Massachusetts Institute of Technology, 2001.
22. Ravikanth Pappu, Ben Recht, Jason Taylor, Neil Gershenfeld: *Physical One-Way Functions*, Science, vol. 297, pp. 2026-2030, 20 September 2002.
23. M. Riedmiller, H. Braun: *A direct adaptive method for faster backpropagation learning: The RPROP algorithm*. IEEE international conference on neural networks, pp. 586–591, 1993.
24. Ulrich Rührmair, Srinivas Devadas, Farinaz Koushanfar: *Security based on Physical Unclonability and Disorder*. In M. Tehranipoor and C. Wang (Editors): "Introduction to Hardware Security and Trust". Springer, 2011.
25. Ulrich Rührmair, Marten van Dijk: *Practical Security Analysis of PUF-based Two-Player Protocols.* CHES 2012.
26. Ulrich Rührmair, Marten van Dijk: *PUFs in Security Protocols: Attack Models and Security Evaluations*. IEEE Symposium on Security and Privacy (Oakland'13), 2013.
27. U. Rührmair, D.E. Holcomb: *PUFs at a glance*. DATE 2014, pp. 1-6, 2014.
28. Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, Jürgen Schmidhuber: *Modeling Attacks on Physical Unclonable Functions*. ACM Conference on Computer and Communications Security, 2010.
29. Ulrich Rührmair, Jan Sölter, Frank Sehnke: *On the Foundations of Physical Unclonable Functions.* Cryptology e-Print Archive, June 2009.
30. Ulrich Rührmair, Jan Sölter, Frank Sehnke, Xiaolin Xu, Ahmed Mahmoud, Vera Stoyanova, Gideon Dror, Jürgen Schmidhuber, Wayne Burleson, Srinivas Devadas: *PUF Modeling Attacks on Simulated and Silicon Data*. IEEE Transactions on Information Forensics and Security (IEEE T-IFS), 2013.
31. G. Edward Suh, Srinivas Devadas: *Physical Unclonable Functions for Device Authentication and Secret Key Generation.* DAC 2007: 9-14

Chapter 5

Applications of High-Capacity Crossbar Memories in Cryptography

As we have seen over the last two chapters, modeling attacks are a vicious threat to electrical Strong PUFs — even more so if combined with side channels. One reason for their effectiveness is that the input-output behavior of most electrical Strong PUFs is fully determined by a relatively small number of internal parameters, much fewer parameters than their number of CRPs. This leads to information-theoretic dependencies between the CRPs. Knowledge of a small fraction of CRPs may therefore allow the derivation of these internal parameters and a subsequent prediction of the entire CRP space. One well-known example are Arbiter PUFs, whose input-output behavior is fully determined by the delays accumulated in their subcomponents or stages. Knowing these delays allows prediction of the Arbiter PUF outputs, as we have seen over the last chapters.

This suggests that one countermeasure against modeling was to design PUFs in such a way that

(i) the PUF still possesses a very large number of CRPs (at least in absolute terms), while

(ii) *all* its CRPs are information-theoretically independent.

For straightforward reasons, such PUFs could be called *"super-high information content PUFs"* or *"SHIC PUFs"*. Once realized in practice, SHIC PUFs would have the advantage that they could implement all known Strong PUF protocols in an information-theoretically secure way, for example the known schemes for oblivious transfer, key exchange, or bit commitment [112, 16, 123, 125].

Let us discuss one instructive (but impractical) example to familiarize ourselves with SHIC PUFs, namely disordered solid-state surfaces. It is long known that the latter commonly have around 10^{15} surface atoms per square centimeter [151]. For the

sake of the arugument, suppose that such a surface has been prepared by an irregular, random manufacturing process. For example, let us assume that the surface has been oxidized incompletely, in such a way that every single atom is oxidized (or not oxidized) independently with probability $1/2$. Every atom then consequently carries an information content (or physical entropy) of one bit. This would result in an entropy of the surface of 10^{15} bits per cm^2.

What could be the challenges and responses of our hypothetical *"surface-PUF"*? It appears natural to let the challenge consist of the location (i.e., of the "coordinates") of a surface atom. By latest microscopic techniques, for example atomic force microscopes (AFMs), the oxidization state of each atom at a certain coordinate could be read out. The resulting bit (i.e., oxidized or non-oxidized) could be employed as the PUF-response. This would theoretically lead to 10^{15} information-theoretically independent CRPs.

Unfortunately, the above, surface-based example is not practical for a number of reasons. It cannot be embedded or integrated into microelectronic systems, and is not stable against wear-and-tear. This chapter is therefore concerned with the question whether *integrated* versions of SHIC PUFs could exist. Along our investigations, it turns out that there is a third requirement (besides the above two) which is crucial to fulfill:

(iii) The read-out speed of the SHIC PUF is inherently limited by its design, and is relatively small in absolute terms (such as 1000 CRPs/sec). It cannot be accelerated by the adversary in any way, for example by reading out several CRPs in parallel.

Interestingly, our above surface example does not meet property (iii), since the adversary could potentially read out different regions of the surface at the same time. Something similar would hold for very large SRAM arrays, which could be read out both quickly and in parallel by the adversary. This illustrates the challenge behind successful SHIC PUF design.

Still, this chapter shows how to construct integrated, electrical SHIC PUFs that meet the above properties (i) to (iii). Our approach consists of using a large, nanoscale, high-density crossbar memory with high entropy and intrinsically slow read-out speeds. Due to their particularly simple architecture of parallel wires, crossbar structures lend themselves easily to nanoscale miniaturization, leading to very small "footprints" of each PUF-response. Furthermore, the crossbars can be designed in such a way that only one PUF response can be read out at a time, and the read-out speed cannot be accelerated by the adversary; otherwise, the crossbar wires get overloaded and burn. Another notable technical aspect is that our large crossbar structures require diodes with very high rectification rates of around 10^7 in order to stabilize the read-out process and avoid parasitic currents. We demonstrate for the first time that such diodes can be produced by the ALILE process.

SHIC PUFs appear interesting from a wider, philosophical perspective: They explore the limits of PUFs by asking how much entropy can be contained in, and reliably read out from, PUFs. This view connects PUFs to the nanosciences and also to deeper questions on the the limits of small scale storage technology. In this sense, Cross-

bar PUFs could be seen as the first forerunners of a trend towards nanosecurity and nanotechnology within the PUF area, which is recently beginning to shape [109].

The publication used in this chapter is

- U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, G. Csaba: *Applications of High-Capacity Crossbar Memories in Cryptography*. IEEE Transactions on Nanotechnology, Vol. 10(3), pp. 489-498, 2011.

Applications of High-Capacity Crossbar Memories in Cryptography

Ulrich Rührmair, Christian Jaeger, Matthias Bator, Martin Stutzmann, Paolo Lugli, *Senior Member, IEEE*, and György Csaba

***Abstract*—This paper proposes a new approach for the construction of highly secure physical unclonable functions (PUFs). Instead of using systems with medium information content and high readout rates, we suggest to maximize the information content of the PUF while strongly reducing its readout frequency. We show that special, passive crossbar arrays with a very large random information content and inherently limited readout speed are suited to implement our approach. They can conceal sensitive information over long time periods and can be made secure against invasive physical attacks. To support our feasibility study, circuit-level simulations and experimental data are presented. Our design allows the first PUFs that are secure against computationally unrestricted adversaries, and which remain so in the face of weeks or even years of uninterrupted adversarial access. We term the new design principle a *"SHIC PUF,"* where the acronym SHIC stands for super high information content.**

***Index Terms*—Crossbar memories, nonvolatile memories, physical cryptography, physical unclonable function (PUF).**

I. Introduction

PHYSICAL unclonable functions (PUFs) are emerging as a new, powerful approach to cryptography and security applications [1]–[7]. Ideally, the security of a PUF should stem from the physical irreproducibility (or uniqueness) and the internal complexity of micro- or nanoscale physical systems. It should be based on the hard technological limitations and the formidable costs related to characterizing and remanufacturing physical objects with nanoscale precision. Contrary to the largest part of mathematical cryptography, its security should not depend on the computational power of the adversary, and should ideally not be vulnerable against the development of more efficient breaking algorithms or increasingly powerful computers.

Historically, the first PUFs were optical systems [1], which exhibited complex internal behavior and high structural information content, but required sensitive and expensive readout machinery. One much discussed recent possibility is to build on-chip PUFs from integrated electrical circuits [2], [3]. For example, the individual, subnanosecond delays between units of an readout machinery can carry a signature that is unique to each circuit. Nevertheless, it turns out that many of the currently suggested PUF circuits can be machine-learned [4], [5] in order to model their behavior. This allows the construction of an imitation device that behaves indistinguishably from the original circuit and which breaks its security.

It can be argued that most of the so far proposed architectures suffer from a common problem: the quantity of structural information that is effectively extracted from the object is too low to fully rule out machine learning or other algorithmic attacks. For the currently known circuit implementations, the "useful" amount of information is on the order of a few parameters (some real numbers with a limited precision) per circuit block, with the number of blocks being on the order of several hundred. For the entire circuit, the relevant information content is, therefore, presumably less than 1 kB. Optical PUFs [1] hold more structural information, but not dramatically: the number of scattering particles used in [1] is on the order of 10^5, even if the speckle pattern is very sensitive to their precise location. Due to these facts, the security of current PUF implementations is in principle susceptible to algorithmic attacks just as mathematical cryptography, and eventually rests on unproven computational complexity assumptions too.

In this paper, we propose a different approach to physical cryptography: we suggest circuits that maximize the effectively extractable structural information content of a physical system while drastically reducing the readout speed. We term this new concept a *"SHIC PUF"* (pronounce as "chique PUF"), where SHIC stands for super high information content. It allows the design of PUFs that remain secure over very long time periods, and which are naturally immune against any algorithmic attacks, including any machine-learning techniques. Their security can even withstand attackers with unlimited computational power. At the same time, the reduced readout speed does not restrict their usability in many relevant applications such as key exchange or credit cards.

Manuscript received July 9, 2009; revised February 8, 2010; accepted April 4, 2010. Date of publication May 3, 2010; date of current version May 11, 2011. This work was supported by the International Graduate School of Science and Engineering, the Institute for Advanced Study, Technische Universität München, and the VERSATILE Project. The review of this paper was arranged by Associate Editor D. Hammerstrom.

U. Rührmair is with the Computer Science Department, Technische Universität München, Munich D-80333, Germany (e-mail: ruehrmai@in.tum.de).

C. Jaeger and M. Stutzmann are with the Walter Schottky Institute, Technische Universität München, Munich D-80333, Germany (e-mail: christian.jaeger@wsi.tum.de; stutz@wsi.tum.de).

M. Bator was with the Walter Schottky Institute, Technische Universität München, Munich D-80333, Germany. He is now with the Paul Scherrer Institute, 5232 Villigen PSI, Switzerland (e-mail: Matthias.Bator@wsi.tum.de).

P. Lugli is with the Institute for Nanoelectronics, Technische Universität München, Munich D-80333, Germany (e-mail: lugli@tum.de).

G. Csaba was with the Institute for Nanoelectronics, Technische Universität München, Munich D-80333, Germany. He is now with the University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: gcsaba@nd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNANO.2010.2049367

Since memory circuits are naturally optimized to densely hold a large amount of information, it is straightforward to think of a specially designed memory as a promising way to implement SHIC PUFs. Indeed, we show that suitably designed solid-state memories can serve very well for that purposes.

The paper is organized as follows: the reader is provided with background on PUFs in Section II. The special requirements for a memory that can be used as a SHIC PUF are given in Section III. We argue in Section IV that passive crossbar memories represent the solution with highest security and lowest cost for a 2-D IC technology. In Section V, we show that with realistic device characteristics the readout from the crossbar will work reliably, and Section VI demonstrates that the crossbar can be made as slow as desired. Section VII outlines some specific material systems and realization possibilities. In Section VIII, we discuss preliminary experimental results. Section IX discusses a few implementation variants and the eventual limits of our approach. We summarize our study in Section X.

II. State of the Art on PUFs

A PUF is a physical system that maps challenges C_i to responses R_{C_i}, and which meets the following security feature: even if an adversary Eve (E) has unrestricted access to the PUF for a limited time period t, and even if Eve is provided with a large number of challenge-response pairs (C_i, R_{C_i}) of the system, it must still be impossible for her to fully characterize, learn, or understand the behavior of the system. After access to the PUF has been withdrawn, Eve should have a relatively low chance of predicting the correct response R_{C_i} to a randomly chosen, earlier unknown challenge C_i. Eve's actions during access are not restricted to determining as many standard challenge-response pairs (C_i, R_{C_i}) as possible, but she can perform arbitrary physical measurements on the system. This concept has at times also been referred to as a Strong PUF [8]; we use both expressions synonymously in this manuscript.

A simple and illustrative PUF-based cryptographic protocol that can be used for the identification of hardware systems or other entities is the following: in a presetting phase, a central authority CA measures some randomly selected (C_i, R_{C_i}) pairs of the PUF, and stores them in a secret list. Subsequently, the PUF is embedded in a hardware system S, or on a personal security token, and is released to the field. If the system S later wants to identify itself, the CA chooses some earlier measured challenges C_i at random, and sends them to S. If S answers with the correct responses R_{C_i}, then the CA can be certain that she is indeed communicating with S.

Typically, only a very small subset of the possible (C_i, R_{C_i}) pairs is used for the secret list of the CA and in the communication of the CA and the device. In contrast, the adversary E must know all (or almost all) possible (C_i, R_{C_i}) pairs in order to falsely claim ownership of the PUF. This is due to the fact that E has no way of knowing, which are the C_i challenges that the CA uses for testing.

The advantage of the described protocol is that it avoids the storage of digital keys in hardware systems, where they often be readout easily by invasive, side channel, or virus attacks. It also obviates the execution of computationally intensive asymmetric cryptoschemes in mobile device, since the security of the aforementioned scheme is built solely on the high challenge-response complexity of the PUF.

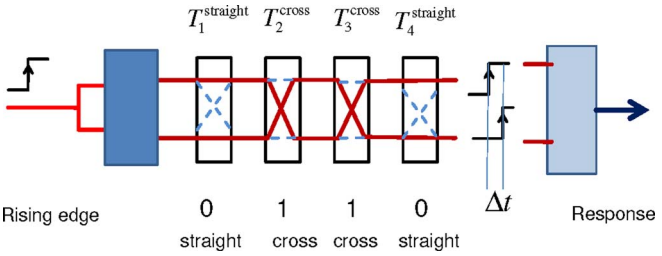

Fig. 1. Illustration of the arbiter PUF.

The first concrete implementation of a PUF was proposed in [1] and consists of an optical token with a very large number of randomly distributed light scatterers. A laser beam directed toward the token creates an interference pattern on a subsequent screen. One usually regards the angle and point incidence of the laser beam as the challenge C_i of this PUF, while the interference pattern (or a suitably chosen image transformation of it) is interpreted as the response R_{C_i}. The described PUF offers quite high internal complexity and security, but it is also quite impractical, with its readout apparatus being external, large, expensive, and sensitive to perturbations.

One important integrated, electrical example is the so-called arbiter PUF illustrated in Fig. 1. It consists of a sequence of k multiplexers, which are conditioned by a sequence of external bits $X[1], \ldots, X[k]$ [2], [3]. The incoming signal is split into two signals, which race against each other on two paths that are determined by the values of the $X[i]$. At the end of the structure, an "arbiter" consisting of a latch determines whether the top or bottom path arrived first, and correspondingly outputs a zero or a one. The arbiter PUF thus maps a k-bit input challenge $C_i = X[1], \ldots, X[k]$ to a 1-bit response R_{C_i}. Unfortunately, the effectively extracted information content is relatively low, and amounts just to a few delay values per multiplexer stage. This results in security problems; the arbiter PUF [4], [5] and also all subsequently improved versions (XOR arbiter, feed-forward arbiter) have been broken successfully by machine-learning algorithms [6].

Applications that have been proposed for PUFs are secure credit cards, access cards and passports, unforgeable labels for valuable goods, identification of entities in insecure networks, secure key exchange, and tamper-sensitive hardware. PUF-like structures called physically obfuscated keys have been used in the context of tamper-sensitive hardware and IP protection.

It is an open research problem to find integrated electrical PUFs that can reach the complexity of known optical implementations [1]. Furthermore, one would ideally like to develop PUFs whose security is strictly independent of the computational power of an attacker. Current design strategies in which several system subcomponents interact and produce readout values at high frequencies, may not suffice in order to meet these goals. To illustrate our point, consider a hypothetical

PUF-circuit with the following properties: it generates 1-bit responses R_{C_i}, allows a readout rate of 10 MHz, and has 10-Mb of relevant random structural features. Within seconds, a list of challenge-response pairs containing, in principle, all relevant information about this PUF can be extracted from it. From this point onward, its security is only upheld by the unproven computational hypothesis that the structural information cannot be extracted efficiently from the gathered data, and cannot be used for modeling and predicting the PUF subsequently. Similar considerations apply to the described optical PUF [1], which contains less than 10^5 scatterers and allows quite high readout bitrates.

Furthermore, the number of PUF components that interact with each other cannot be increased indefinitely, since this may result in stability issues, fading signals, and simplified effective behavior of the system due to averaging effects. Electrical PUFs suffer the most from this restriction.

The concept of an electrical PUF is somewhat related to the idea artificial fingerprint devices [9] (AFDs). In one proposed AFD, for example, a unique signature of the circuit is generated from polysilicon thin-film transistor characteristics and used instead of digitally stored keys [10]. However, this signature can be straightforwardly extracted from invasive or semiinvasive electrical measurements [11], and the behavior of the random structure can be easily imitated digitally by a lookup table. AFDs cannot provide fundamentally higher security than protected digital keys due to the small amount of easily accessible structural information contained in them.

III. ROM Memories as SHIC PUFs

Our approach to circumvent the known problems in the design of secure PUFs was to readout a very large amount of independent structural information while drastically reducing the readout speed—we termed this concept a SHIC PUF in Section I. Since memory circuits are already optimized in terms of information density and readout stability, it is suggestive to borrow concepts from this area in order to implement our idea. The C_i challenge becomes a memory address, and the R_{C_i} response is the information stored therein.

In order to be used as a SHIC PUF, a fixed content memory circuit (ROM), containing N bits of information, should satisfy the following requirements.

1) The readout speed is limited by the design of the circuit to k b/s for a small value of k.
2) The time T_{full} required for complete circuit characterization ($T_{\text{full}} = N/k$) exceeds the application lifetime of the circuits or the maximal access time of an adversary (depending on the application, it should be on the order of several days, weeks, or even years).

Further requirements, which are not essential, but can significantly improve security, are as follows.

3) The N-bit content of the memory is physically random. It is caused by irregularities in the manufacturing process, which even the manufacturer cannot fully control.
4) The readout speed is limited directly and intrinsically by the construction of the memory cells, bit and word lines, and not by an artificially slow-access module. It cannot be sped up by an invasive attacker who cuts off the module and uses different, faster circuitry to access the memory.

Reasonable values for the memory size are $N = 10^{10}$ bits and $k = 100$ b/s, resulting in $T_{\text{full}} = 10^8$ s or approximately three years. We will use these parameters as *"design target"* in the rest of this paper.

Since the targeted 10-Gb information content is well within the reach of today's semiconductor memories, there are numerous realization possibilities, provided that we only aim to meet requirements 1) and 2). Any sufficiently large memory with two access modules will do: the memory is first written with random bits, using a fast-access module. Then, this module is burnt or cut off, and only the second, slow module remains for readout. The advantage of such implementation is that novel technology is not required to realize the SHIC PUF.

If we add requirement 3) to our list, there are still plenty of options. For example, it is known that a nonwritable SRAM cell on power-up latches into a state that is decided by the tiny asymmetry between its two inverters (transistors) [12]. An array of such SRAM cells could carry the required large and physically random information content. It could also be possible to modify flash-based or phase-change memory designs to operate as a SHIC PUF, by exploiting the randomness of transistor characteristics (or the state or the information-carrying phase-change layer). Since the structural information is not modified during the lifetime of the device, no writing circuitry is required for this memory.

If we aim for maximal security, however, and assume that it may be feasible for Eve to tamper with the peripheral circuits of the memory block, then the memory must also meet requirement 4). This seems difficult to satisfy for a memory built from conventional microelectronic technologies, since even slow semiconductor memories are operating in the megahertz regime. In addition, our sought technology should maximize the information content per chip area. The footprint of a single bit should be small (ideally $A < 100\text{ nm} \times 100\text{ nm}$) so that the $N = 10^{10}$-bit memory would fit in a few centimeter-square area.

IV. SHIC PUFs Built from Crossbar Architectures

Cross-point architectures are the simplest functional nanodevices, possessing a very regular geometry and using only two-terminal passive devices. They hold a great promise in nanoelectronics, where fabrication challenges prohibit making a more complex, arbitrarily interconnected circuit. Circuit architectures built from memristive crossbars [13] are being actively researched today [14]. Crossbar architectures used as SHIC PUFs could, hence, achieve the highest device density, which is feasible by a certain technology node, making the PUFs small, highly secure against tampering, and potentially cheap.

The sketch of a crossbar array is shown in Fig. 2. A particular bit at the intersection of the horizontal and vertical lines is addressed by activating the corresponding bit and word lines and measuring the current flowing through the crossing. Usually, each junction is a multilayered structure showing nonlinear characteristics. We assume that only the storage array is

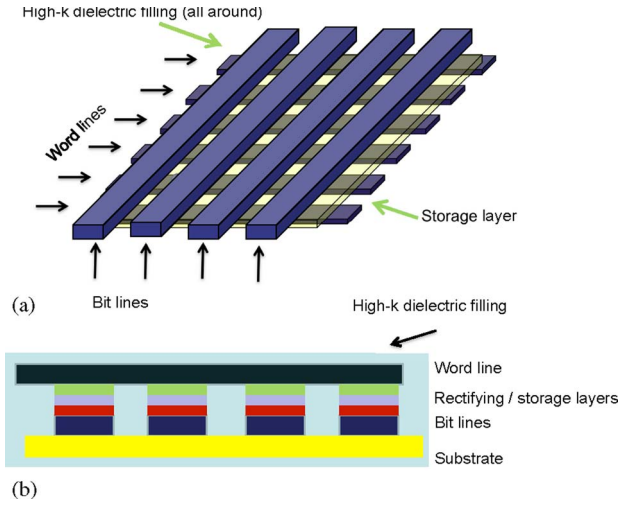

Fig. 2. Schematic illustration of a crossbar memory. (a) From a perspective view. (b) From the side view.

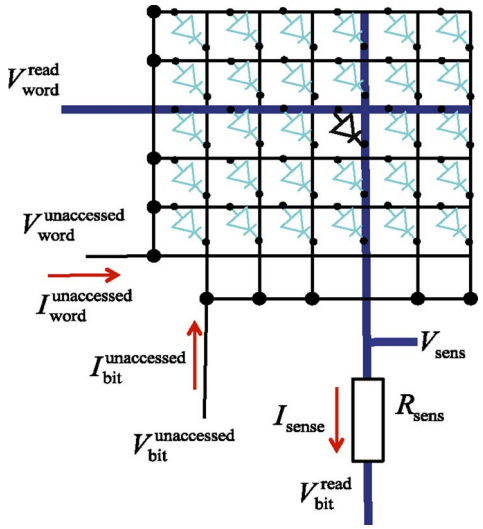

Fig. 3. Biasing scheme of a crossbar array.

implemented by crossbar technology and the readout apparatus is a silicon-based circuit [15].

A crossbar used as SHIC PUF is different from a standard crossbar memory in the following aspects.

1) The SHIC PUF-crossbar does not need to be writable. Through all its lifetime, it carries a hard-wired information content, defined by the storage layer, which is unique and random for each instance of the fabricated memory. The storage layer is an inhomogeneously conducting material, with resistance changing on the size scale of the $2F$ pitch size of the crossbar. (F is the lithography resolution.)
2) The space between the bit and word lines may be filled with a high-k (high dielectric constant) material, which creates large interwire and junction capacitances.
3) The entire memory is built as one monolithic block, where the number of bit and word lines being around $n = \sqrt{N} \approx 10^5$. This prevents the attacker from accelerating the readout by reading multiple memory banks in parallel.

The most unusual character of a PUF-crossbar is the 3), as mentioned earlier: large memory circuits are usually realized from multiple banks in order to reduce access time and improve noise margin and yield. We demonstrate in the next section that reliable readout in such large banks is nevertheless possible.

V. Accessing Information in Large Crossbar Circuits

The circuit schematics of a biased crossbar circuit are illustrated in Fig. 3. We assume that the accessed word and bit lines are biased on $V_{\text{bit}}^{\text{read}}$ and $V_{\text{word}}^{\text{read}}$ voltage, respectively. The unaccessed wires are on a fixed $V_{\text{bit}}^{\text{unaccessed}}$ and $V_{\text{word}}^{\text{unaccessed}}$ voltage.

For simplicity, all junctions of the aforementioned circuit are drawn by the same diode symbol, but their I–V characteristics are obviously different, carrying the random structural information. If only one bit of information is extracted per junction, we can refer to the diode as being in the "ON" or "OFF" state. There is a sense resistor (R_{sens}) connected to the accessed word line,

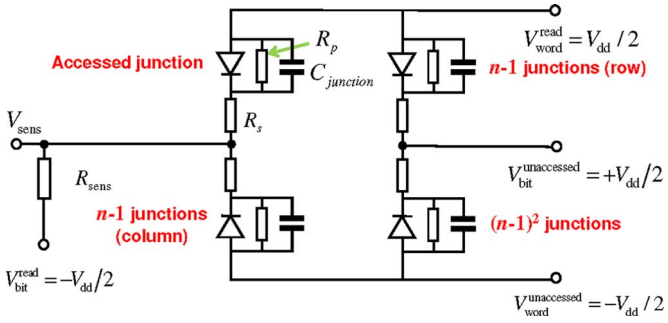

Fig. 4. Equivalent circuit of a crossbar array. Each diode symbol (with the corresponding R_s and R_p serial and parallel resistances) represents an "average" junction, i.e., all the junctions connected in parallel to the same two nodes.

which can have a low or a higher value, depending on whether current or voltage measurement is done by the sense amplifier.

Assuming that the series resistances of the wires are negligible (or they can be approximated by a lumped resistance), one can construct a simple equivalent circuit model of the array, which is shown in Fig. 4. Here an equivalent ("average") lumped circuit element substitutes the junctions connecting to the accessed word/bit line and the rest of the array. Junctions connected to the accessed bit line can directly interfere by the readout process, while others just add to the net current inflow (and dissipation) of the structure.

To interrogate the selected bit in the crossbar array, we apply the bias scheme of Fig. 4. Most of the unaccessed junctions ($(n-1)(n-1)$ of them) are reverse-biased ($V_{\text{word}}^{\text{unaccessed}} = -V^{dd}/2$, $V_{\text{bit}}^{\text{unaccessed}} = V^{dd}/2$), minimizing the magnitude of parasitic currents. The interrogated junction is the only forward-biased in the array ($V_{\text{word}}^{\text{read}} = V^{dd}/2$, $V_{\text{bit}}^{\text{read}} = -V^{dd}/2$), unaccessed junctions connecting to the accessed bit and word lines get zero bias.

As an illustration, Fig. 5(a) shows two typical diode I–V) curves, with a high and low series resistance, representing the binary information carried by the junction. Fig. 5(b) shows the

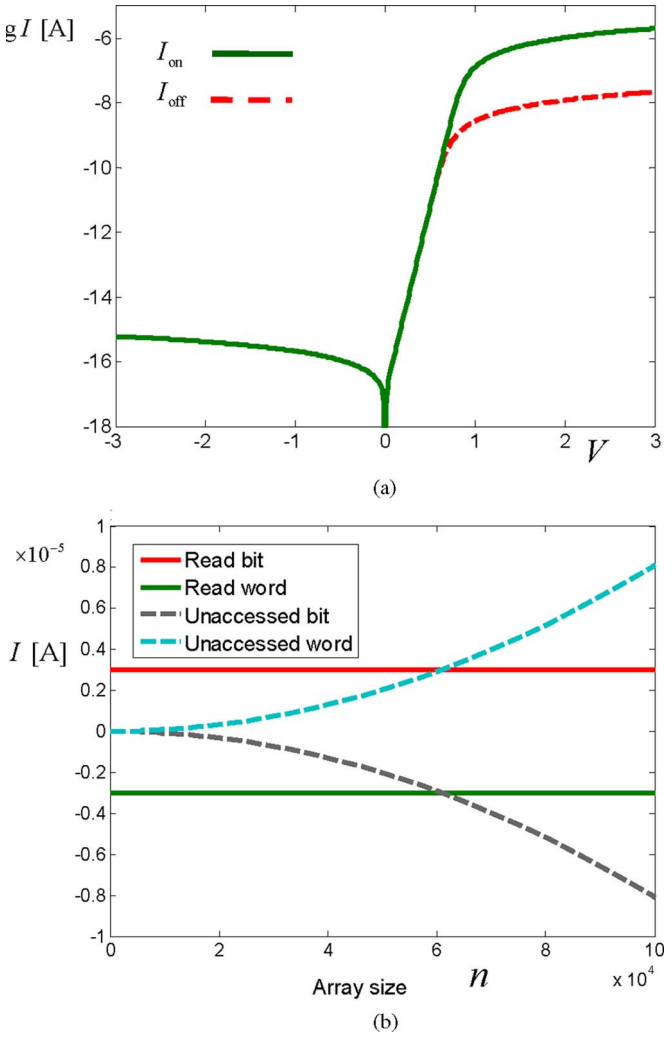

Fig. 5. (a) *I–V* characteristics of a diode-backed junction, using $I_s = 10^{-17}$ A, $R_p = 5 \times 10^{15}\ \Omega$, an ideality factor of 1.5 and serial resistances of $R_s = 1$ MΩ and $R_s = 100$ MΩ for the ON and OFF state of the junction, respectively. (b) Current inflow at different array sizes, as simulated by SPICE, at $V^{dd} = 2.0\,V$.

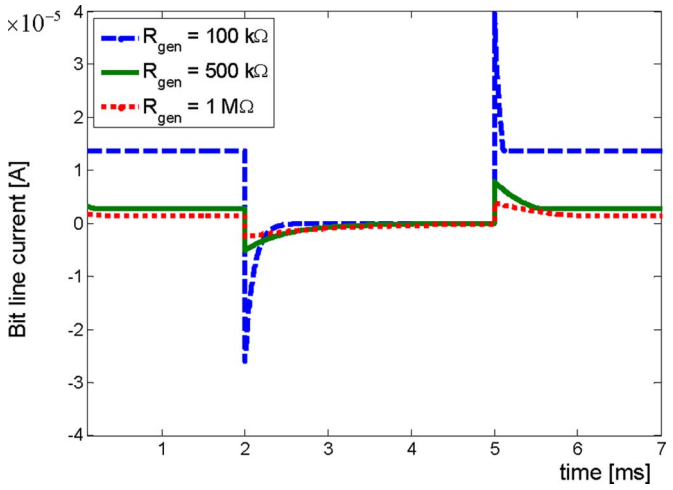

Fig. 6. Complete read cycle of the crossbar memory. The curves show the time dependence of $I_{\rm sens}$ current and the bias voltages. At $t = 2$ ms, the accessed junction is unbiased (reverse-biased) and at $t = 5$ ms, biased again.

sense current and the parasitic currents as a function of array size. For about $n \approx 6 \cdot 10^4$ array sizes, the net current flowing through the reverse-biased (unaccessed) junctions begins to exceed the "useful" current flowing through the accessed bit and word line. This causes unnecessary power dissipation, but the parasitic current on a single bit/word line [on average (I/n)] still remains small. Noise margin is high, as most of the parasitic paths avoid the accessed bit and word lines. Taking into account the bit and word line resistances would reduce the noise margin, but the calculations show that the diode-backed crossbar memory is scalable to very large array sizes, at least in the region of $n > 10^5$, $N > 10^{10}$, as required by the specified design target.

We did not investigate the construction of the memory-decoding circuit. For lithographic crossbars ($F > 100$ nm), the decoder must be straightforward to build, even if decoding a $10^5 \times 10^5$ size memory block would be quite unusual and impractical in conventional memory designs. For nonlithographic crossbars, several constraints arise at the construction of the "nano–micro" link. The reader is referred to the literature [16]–[18] for the solutions currently being researched.

VI. Slow ReadOut Speed and Security Against Invasive Attacks

Time-dependent behavior of the crossbar can be modeled by running a transient simulation on the circuit of Fig. 4. It is required to consider the serial (Thevenin equivalent) resistance of the voltage generators that drive the accessed/unaccessed bit and word lines—these $R_{\rm gen}$ generator resistances are not shown in Fig. 4.

Assuming that the series resistance of the bit/word wires is negligible (or can be approximated by a lumped resistance) and that the wires survive any current density, it is the generator resistance $R_{\rm gen}$ and the net capacitance of the word line C_Σ that determines the $\tau = R_{\rm gen} C_\Sigma$ time constant of the circuit.

Fig. 6 shows the simulation of a complete read cycle, using a junction capacitance of $C_{\rm junction} \approx 10^{-14}$ F and assuming generator resistances of $R_{\rm gen} = 100\,{\rm k}\Omega$, $R_{\rm gen} = 500\,{\rm k}\Omega$, and $R_{\rm gen} = 1\,{\rm M}\Omega$. At the beginning ($0 < t < 2$ ms), the crossbar is readout using the bias scheme of Fig. 4. At $t = 2$ ms, the generators abruptly unbias the accessed junction, i.e., both $V^{\rm read}_{\rm word}$ and both $V^{\rm read}_{\rm bit}$ switch polarity. At $t = 5$ ms, the polarity of this wires switch again; therefore, the interrogated junction is forward-biased again.

The simulations of Fig. 6 show that, for the parameters we choose, at least a few milliseconds must elapse between the subsequent readouts for the sense current/voltage to stabilize. The resulting readout speed of around 100 b/s corresponds well to the specifications in Section III.

Smaller values of $R_{\rm gen}$ lead to faster readout cycles and, at the same time, a higher capacitive peak current during the charge up of the bit or word lines. The wire can be overloaded and destroyed by this. $R_{\rm gen}$ has to be chosen in such a way that the driven word line is not destroyed; consequently, the speed of a large crossbar memory is limited by the finite current-carrying

ability of the nanoscale wires. Faster readout attempts will inevitably result in rapid destruction of these wires, rendering the structure unusable/unreadable.

The $C_{\rm junction} \approx 10^{-14}$ value we used in the calculations is about an order of magnitude higher than the geometric capacitance could alone provide. An elementary capacitance calculation ($C = \epsilon_r \epsilon_0 (A/d)$) gives $C = 10^{-15}$-F capacitance for an $A = 100\,{\rm nm} \times 100\,{\rm nm}$ junction (using $d = 1$ nm and $\epsilon_r = 10$). Additional parasitic capacitances, interwire capacitances, and equivalent capacitances associated with carrier mobilities can increase the junction capacitance to the desired value. Another possibility to increase capacitances is filling the gaps of the structure with high-k materials, with a dielectric constant being in the $\epsilon_r \approx 100$–1000 range. Such very high-k materials (ferroelectrics) were widely investigated and characterized for ferroelectric memory applications and their technology is compatible with standard silicon processing [19], though they considerably complicate the fabrication process. A lower $C_{\rm junction}$ value may also suffice, but the $R_{\rm gen}$ should be increased to maintain the same τ time constant. An excessively high $R_{\rm gen}$ decreases the noise margin of the memory.

If more than a single bit of information is stored in the junctions (analogously to multibit storage in modern memories), then the required measurement precision will further slow down circuit operation. Another mode of operation is to compare the resistances of two randomly selected junctions. This may also provide compensation against power supply fluctuations and certain aging effects.

The adversarial attacker could try to manipulate or entirely replace the readout circuitry of the memory in order get quick access to its content. This can be done by using a smaller value for $R_{\rm gen}$ (i.e., decreasing the time constant) and/or reading out multiple bits in parallel.

Both of these approaches are prevented, however, if the wires have been set to have only a limited $i^{\rm in}_{\rm max}$ current-carrying capability. Decreasing the value of $R_{\rm gen}$ m-fold will result in m times larger peak currents and destroys the wire. Reading multiple (say h) bit values simultaneously loads the corresponding bit/word line with $h \times i_{\rm static}$ current and exceeds the value of $i^{\rm in}_{\rm max}$ already for small h. As our simulation results show, the $i^{\rm in}_{\rm max}$ current limit could still be a few times larger than the $i_{\rm static}$ steady-state current flow, meaning that a regime exists, where the crossbar would still be reliably readable, but at the same time secure.

A predetermined breaking point could be defined on the nanowires to control the maximum allowable current densities. The cryptographic application can tolerate, if a number bit or word lines are damaged during fabrication as the bad C_i, R_{C_i} pairs could be ignored in the cryptographic protocol.

If the crossbar is fabricated by state-of-the-art lithographic technology or with sublithographic resolution, tampering with the internal structure of the crossbar array seems to be technologically impossible, even for adversaries with practically unlimited financial resources. This prevents attacks in which the adversary would split the crossbar into several subblocks and reads them out in parallel or fabricate contacts to access inner nodes. The dense and regular structure of a crossbar and the transistorless construction of the storage block most likely prevents attacks that are conceivable for circuits made with conventional microelectronic technologies such as cryptographic processors [11].

VII. Realization Possibilities for the Random Information Content

One crucial component of the crossbar memory is the information-carrying layer. Ideally, its irregular structural features should result in a truly random information content of the memory. There are several suggestive random physical processes, which form the sought type of nanostructures.

1) One possibility involves a phase change material, which is illuminated by a random image (such as a series of unaligned speckle patterns), resulting in an inhomogeneously conducting media [20]. This method is not manufacturer resistant [5], meaning that a fraudulent manufacturer could generate more than one memory of the described type with the same information content.
2) Alternatively, a very thin oxide layer with a nonuniform thickness can provide a tunneling current, which is different from junction to junction.
3) Also crystallization processes exhibit an inherent randomness: the exact location of nucleation sites depend on atomic-scale defects or roughness of material surfaces. One example would be amorphous silicon, crystallized with a laser beam again in combination with a speckle pattern. For phosphorous-doped amorphous Si:H (a-Si:H), a resistivity change between crystallized and noncrystallized areas of 100 and 10^6 can be obtained for a P-concentration of $10^{-5}\%$ and 2% in the initial layer, respectively. These crystallization processes can reach resolutions below 100 nm due to the small heat diffusion length in the silicon [21]. A polycrystalline material can be doped as well, resulting in an inhomogenously doped semiconductor.

One particular advantage of these processes is that they can be made compatible with modern semiconductor manufacturing technology.

VIII. Generating the Information by Random Crystallization Processes

To illustrate the feasibility of a randomly conducting layer, we have performed experiments on a medium prepared with a random recrystallization process. Such crystallization processes are particularly attractive for our goal, since the nucleation site cannot be calculated or predicted, and the nucleation process is governed by atomic-scale inhomogeneities of the starting material.

We chose the aluminum-induced layer exchange (ALILE) process [22]–[24], which is used to crystallize amorphous silicon layers. In a typical ALILE process (which is illustrated in Fig. 7), an Al/amorphous Si (a-Si) layer stack, separated by a thin oxide film [see Fig. 7(a)], is annealed at temperatures below the eutectic temperature of the Al–Si system. Annealing of the sample leads to diffusion of Si atoms into the Al layer.

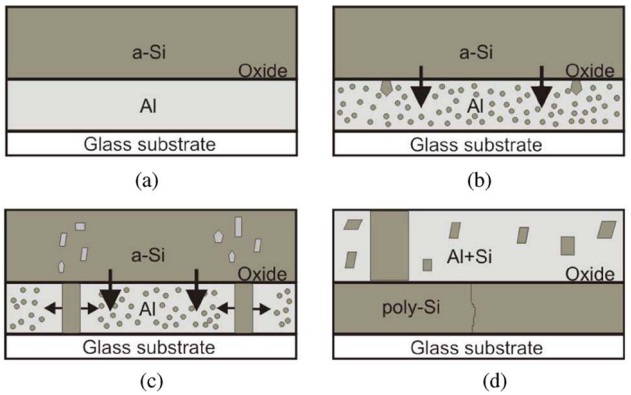

Fig. 7. Steps of the ALILE process. (a) Al/amorphous Si layer stack on glass substrate as starting configuration. (b) and (c) During the annealing, Si nuclei form in the Al and grow in size. (d) Finally, a closed polycrystalline layer has formed replacing the Al.

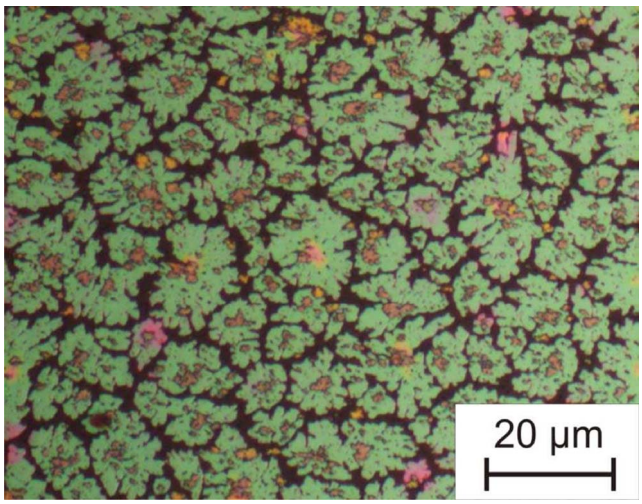

Fig. 8. Top-view optical microscopy image of the resulting ALILE layer.

Crystallite formation occurs, where local supersaturation of the Al with Si is achieved [see Fig. 7(b)]. In addition to that, irregularities and defects, e.g., grain boundaries of the Al, can serve as crystallization sites. As nuclei have appeared, they grow until they reach the substrate. From this point, crystallites grow laterally [see Fig. 7(c)] until a closed poly-Si layer has formed [see Fig. 7(d)]. After the process is completed, the silicon and aluminum layers exchanged their respective positions and the a-Si has been crystallized.

By adjusting the initial Al/Si layer thickness ratio, an incomplete poly-Si layer composed of not fully interconnected grains can be achieved. An example illustrating the randomness of such an incomplete ALILE layer is shown in Fig. 8, where the greenish area represents the crystallized silicon grains and the black surrounding the glass substrate. In this case, the Al remaining on top of the poly-Si layer has been removed by wet chemical etching. The ratio between the area covered with crystallites and the bare glass can be adjusted by the Al/Si thickness ratio. The size of the crystallites is determined by the annealing temperature and the initial layer thickness. Since the crystallized

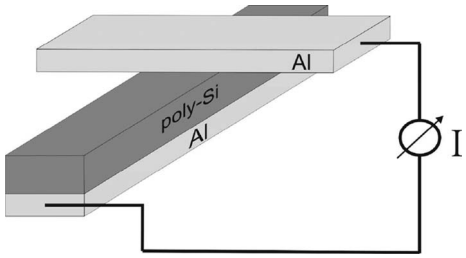

Fig. 9. Schematic illustration of a crossbar junction built by the ALILE process.

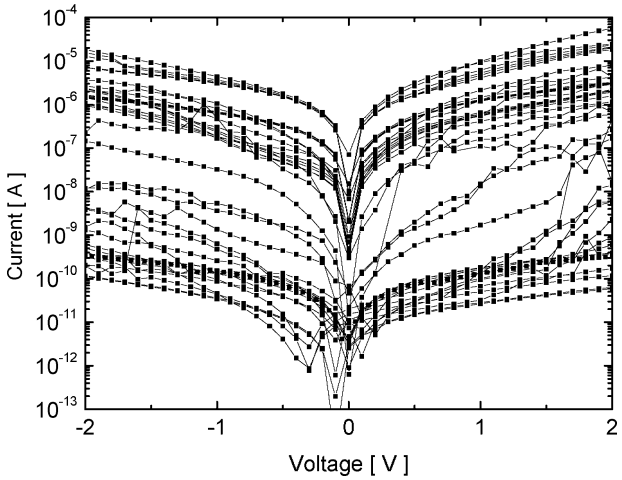

Fig. 10. Arbitrarily chosen measurements on ALILE poly-Si/Al junctions. Random resistive values are present due to the inhomogeneous structure of the poly-Si film.

silicon grains are Al-doped after the layer exchange process, this method results in conducting (Si-grains) and nonconducting (bare glass) regions in a truly random arrangement. Fully crystallized layers also reflect the randomness of their formation process and show inhomogeneous conductivity.

We have fabricated small-size crossbar structures having a crystallized layer as the information carrier. In order to obtain an Al back contact, we used the reverse configuration of the ALILE process [25] to fabricate Al/poly-Si wires with a width of 1–4 μm (see Fig. 9) on quartz glass substrates. Hydrogen passivation is used to reduce the hole carrier concentration in the poly-Si. The details of the hydrogenation process can be found elsewhere [23]. Then Al wires of the same size were evaporated with a mask, aligned perpendicular to the Al/poly-Si wires as sketched in Fig. 9.

Fig. 10 shows some measured *I–V* curves on these junctions. It is clear that the *I–V*curve shows sufficient randomness. We are currently working on scaling the feature sizes down to the 100-nm regime; this is required to obtain our *design target* (see Section III) within a reasonably small (cm^2 size) active chip surface.

The presented layer stack already shows a weakly rectifying behavior, due to the Schottky-type contact between Al and poly-Si layers. This nonlinearity is yet insufficient to make the

crossbar addressable and a separate diode layer is necessary to realize the selector elements.

There are several technologies in the recent literature giving solutions for the fabrication of the diode layer or the entire crossbar. Crossbar memories can be made from standard semiconductor material systems (silicon, poly-Si) and oxide-based switching layers [26], [27]. Molecular switching elements are researched to achieve true nanometer-scale storage [27], [28]. Crossbar memories are proposed to be built from semiconductors that enable low-temperature processing, higher integration densities (such as ZnO) [29], and back end of the line fabrication. This later possibility is especially promising: a low-cost crossbar layer placed on top of a silicon IC can also serve as a coating PUF [30], physically protecting the underlying circuitry. It would also be possible to use high-capacitance, high-resistance amorphous semiconductors (such as amorphous silicon suboxides [31]).

IX. Implementation Variants and Limits

There are several possible device variants for the crossbar PUF, depending on the application area and the desired level of security. The design target outlined in Section III results in a structure that may withstand about three years of *continuous* adversarial access until full characterization ($T = N/k = (10^{10}\,\mathrm{bits})/(100\,\mathrm{b/s}) = 10^8\,\mathrm{s} \approx 3$ years).

If the memory is realized as nonlithographic crossbar (with feature size in the 10-nm range), and the $k = 100$ b/s readout rate can be maintained, then a centimeter-square-size block stays secure for several decades. For lithographic crossbars, a few years should be achievable.

The total adversarial access time and the security lifetime of a product must be distinguished; during the few years lifetime of a credit card, for example, the maximal, hypothetical adversarial access time will never go beyond a few days (card is stolen and brought back unnoticed), and will typically be significantly lower. So a cheap, few millimeter-square-area lithographic crossbar can already provide a practically sufficient level of security. The ALILE technology described in Section VIII can be readily used for a number of applications, including credit cards, passports, and key exchange. These applications realistically require only a T of several days.

Small crossbar blocks will have lower RC constants than large memory banks. High-k materials still can help to keep capacitances high and access rates low. Our group is currently investigating a variant of the ALILE technology, which allows to further slow down the readout rates by introducing a large number of slow traps at the crossbar junctions.

If the access rate of single memory block is extremely low (k is only a few bits per second), then the crossbar can be partitioned into smaller, parallel accessed blocks without compromising security. For sublithographic crossbars, a number of smaller memory blocks may also provide ultimately high security, since invasive attacks (such as microprobing) become practically impossible at this scale.

If multilevel storage is applied, the tradeoff between size, security, and resolution can be improved yet further.

X. Conclusion

This paper proposed a new design paradigm for the construction of secure PUFs. While the standard approach is to employ many interacting components and high readout speeds, we suggest to use as many single, densely packed, independent subunits as possible while drastically reducing the readout frequency. This new principle allows the construction of the first PUFs which are secure even against computationally unbounded adversaries, and in the face of weeks or years of uninterrupted adversarial access. The slower readout speed seems no severe disadvantage in typical appliances such as key exchange, credit cards, or hardware tamper detection.

We suggested crossbar arrays as a preferable way to implement SHIC PUFs. Crossbar arrays lead to electrical SHIC PUFs that can be integrated conveniently on a chip. They reach ultimate information densities and are potentially cost effective, since they have a regular geometry and use only two-terminal passive devices. Due to their simple layout, they can be produced at the limit of current nanofabrication, which gives them high security against invasive attacks and increases their security lifetime. We further showed that it is possible to enforce the slow readout speed required for SHIC PUFs as an intrinsic property of the crossbar's wiring and cell architectures, and not only by an intentionally slow-access module, which might potentially be circumvented or cut off.

We have backed our new design proposal by a discussion of several concrete implementations, circuit simulation data, and an experimental feasibility study. Our research suggests that it should be possible to build a USB-stick-type device with dimensions of a few millimeter $\times$ 1 cm $\times$ 1.5 cm, which is secure for ideally up to tens of years, and which can be used for user identification, hardware identification, key exchange, and other security appliances. Other implementation variants tailored for specific settings can be made yet smaller and cheaper and integrated conveniently in existing microelectronic systems. For further information about ongoing research, referred to [32].

Acknowledgment

The authors would like to thank I. Szentmiklosi, M. Pra, T. Jun, M. Scholz, and X. Ju for valuable discussions and raising excellent questions. This study was conducted in the course of the Physical Cryptography Project of the Technische Universität München.

References

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical One-Way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002.

[2] D. B. Gassend, D. Clarke, M. van Dijky, and S. Devadas, "Silicon physical random functions," presented at the CSC, Washington, DC, Nov. 18–22 2002.

[3] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 10, pp. 1200–1205, Oct. 2005.

[4] D. Lim, "Extracting secret keys from integrated circuits," M.Sc. dissertation, MIT, Cambridge, MA, 2004.

[5] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," *Concurrency Comput., Pract. Exp.*, vol. 16, no. 11, pp. 1077–1098, 2004.

[6] U. Rührmair, J. Sölter, and F. Sehnke. (2009). On the foundations of physical unclonable functions, *Cryptology ePrint Archive: Rep. 2009/277* [Online]. Available: http://eprint.iacr.org/2009/277
[7] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," Cryptology ePrint Archive, Rep. 2010/251, 2010. Available: http://eprint.iacr.org/
[8] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Proc. 9th Int. Workshop CHES 2007 (LCNS)*, vol. 4727, New York: Springer-Verlag.
[9] S. Maeda, H. Kuriyama, T. Ipposhi, S. Maegawa, Y. Inoue, M. Inuishi, N. Kotani, and T. Nishimura, "An artificial fingerprint device (AFD): A study of identification number applications utilizing characteristics variation of polycrystalline silicon TFTs," *IEEE Trans. Electron Devices*, vol. 50, no. 6, pp. 1451–1458, Jun. 2003.
[10] S. Maeda, H. Kuriyama, T. Ipposhi, S. Maegawa, and M. Inuishi, "An artificial fingerprint device (AFD) module using poly-Si thin film transistors with logic LSI compatible process for built-in security," in *Proc. IEDM Tech. Dig. Int. Electron Devices Meeting*, 2001, pp. 34.5-1–34.5-4.
[11] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic processors-A survey," *Proc. IEEE*, vol. 94, no. 2, pp. 357–369, Feb. 2006.
[12] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA," in *Proc. IEEE Int. Workshop Hardware-Oriented Secur. Trust (HOST 2008)*, Jun., pp. 67–70.
[13] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.
[14] G. Snider, "Computing with hysteretic resistor crossbars," *Appl. Phys. A*, vol. 80, pp. 1165–1172, 2005.
[15] G. F. Cerofolini, G. Arena, C. M. Camalleri, C. Galati, S. Reina, L. Renna, and D. Mascolo, "A hybrid approach to nanoelectronics," *Nanotechnology*, vol. 16, pp. 1040–1047, 2005.
[16] A. DeHon, S. C. Goldstein, P. J. Kuekes, and P. Lincoln, "Nonphotolithographic nanoscale memory density prospects," *IEEE Trans. Nanotechnol.*, vol. 4, no. 2, pp. 215–228, Mar. 2005.
[17] R. Beckman, E. Johnston-Halperin, Y. Luo, J. E. Green, and J. R. Heath, "Bridging dimensions: Demultiplexing ultrahigh-density nanowire circuits," *Science*, vol. 310, pp. 465–228, Oct. 21, 2005.
[18] D. B. Strukov and K. K. Likharev, "Defect-tolerant architectures for nanoelectronic crossbar memories," *J. Nanosci. Nanotechnol.*, vol. 7, pp. 151–167, 2007.
[19] S. K. Dey and R. Zuleeg, "Processing and parameters of sol-gel PZT thin-films for GaAs memory applications," *Ferroeloectrics*, vol. 112, pp. 309–319, 1990.
[20] P. Nangle, "Programming method for non-volatile memory," U.S. Patent 7106622, Sep. 12, 2006.
[21] S. D. Brotherton, "Polycrystalline silicon thin film transistors," *Semicond. Sci. Technol.*, vol. 10, pp. 721–738, 1995.
[22] T. Antesberger, C. Jaeger, M. Scholz, and M. Stutzmann, "Structural and electronic properties of ultrathin polycrystalline Si layers on glass prepared by aluminum-induced layer exchange," *Appl. Phys. Lett.*, vol. 91, pp. 201909-1–201909-3, 2007. DOI: 10.1063/1.2803072.
[23] C. Jaeger, T. Antesberger, and M. Stutzmann, "Hydrogen passivation of ultra-thin low-temperature polycrystalline silicon films for electronic applications," *J. Non-Cryst. Solids*, vol. 354, no. 19–25, pp. 2314–2318, May 2008. DOI: 10.1016/j.jnoncrysol.2007.09.040.
[24] O. Nast, T. Puzzer, L. M. Koschier, A. B. Sproul, and S. R. Wenham, "Aluminum-induced crystallization of amorphous silicon on glass substrates above and below the eutectic temperature," *Appl. Phys. Lett.*, vol. 73, pp. 3214–3216, 1998.
[25] J. H. Kim and J. Y. Lee, "Al-induced crystallization of an amorphous Si thin film in a polycrystalline Al/ native SiO_2/amorphous Si structure," *Jpn. J. Appl. Phys.*, vol. 35, pp. 2052–2056, 1996.
[26] C. de Graaf, P. H. Woerlee, C. M. Hart, H. Lifka, P. W. H. de Vreede, P. J. M. Janssen, F. J. Sluijs, and G. M. Paulzen, "A novel high-density low-cost diode programmable read only memory," in *Proc. Int. Electron Devices Meeting*, Dec. 8–11, 1996, pp. 189–192.
[27] M. Johnson, A. Al-Shamma, D. Bosch, M. Crowley, M. Farmwald, L. Fasoli, A. Ilkbahar, B. Kleveland, T. Lee, T.-Y. Liu, Q. Nguyen, R. Scheuerlein, K. So, and T. Thorp, "512-Mb PROM with a three-dimensional array of diode/antifuse memory cells," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1920–1928, Nov. 2003.
[28] G. Csaba and P. Lugli, "Read-out design rules for molecular cross bar architectures," *IEEE Trans. Nanotechnol.*, vol. 8, no. 3, pp. 369–374, May 2009.
[29] M. Pra, G. Csaba, C. Erlen, and P. Lugli, "Simulation of ZnO diodes for application in non-volatile crossbar memories," *J. Comput. Electron.*, vol. 7, no. 3, pp. 146–150, Sep. 2008.
[30] P. Tuyls, G.-J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *Proc. CHES 2006*, pp. 369–383.
[31] R. Janssen, A. Janotta, D. Dimova-Malinovska, and M. Stutzmann, "Optical and electrical properties of doped amorphous silicon suboxides," *Phys. Rev. B*, vol. 60, pp. 13561–13572, 1999.
[32] C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, and M. Stutzmann, "Random pn-junctions for physical cryptography," *Appl. Phys. Lett.*, vol. 96, 172103, 2010. DOI: 10.1063/1.3396186.

Ulrich Rührmair studied mathematics at Ludwig-Maximilians-Universität München, Munich, Germany, and received the M.Sc. degree from the University of Oxford, Oxford, U.K.

He is the Founder and is currently the Head of the Physical Cryptography Project, Technische Universität München, Munich. His research interests include complexity theory at large and cryptography and security, in particular, alternative approaches such as quantum cryptography, DNA-based cryptography, and physical cryptography.

Christian Jaeger was born in Würzburg, Germany, in 1979. He received the Diploma degree in physics from Technische Universität München, Munich, Germany, in 2006, where he is currently working toward the Ph.D. degree at Walter Schottky Institute.

He was also involved with the Universitat de Barcelona, Barcelona, Spain and at the Research Center for Photovoltaics, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan. His research interests include the crystallization of amorphous silicon for large-area electronics and cryptographic applications.

Matthias Bator was born in Grudziadz, Poland. He received the Diploma degree in physics from Technical University Munich, Munich, Germany, in 2008. He is currently working toward the Ph.D. degree at Paul Scherrer Institute, Switzerland.

His current research focuses on the fabrication of multiferroic, in particular, magenetoelectric thin films by pulsed laser deposition and their characterization using X-ray and neutron diffraction techniques.

Martin Stutzmann was born in Frankenberg, Germany, on November 19, 1956. He received the Physics Diploma (*summa cum laude*) degree from the University of Marburg, Marburg, Germany, in 1980, the Licence de Physique degree from Université Paris VII, Paris, France, in 1982, with a fellowship from the German National Academic Foundation, and the Ph.D. (*summa cum laude*) degree, in 1983 under the supervision of J. Stuke. He received the Habilitation degree in experimental physics, in 1990.

During 1975–1976, he was in military service. From 1982 to 1983, he was a Research Scholar at Stanford University, where he was involved in the research on paramagnetic states in hydrogenated amorphous silicon and germanium. From 1983 to 1985, he was a Postdoctoral Researcher at Xerox Palo Alto Research Center, Palo Alto, CA, in the group of R. A. Street, where he was involved in the research on the light-induced metastability and substitutional doping of amorphous silicon. During 1985–1993, he was a Research Staff Member in the group of M. Cardona at the Max Planck Institute for Solid State Research, Stuttgart, Germany, where he was involved in the research on hydrogen in semiconductors, porous silicon, silicon sheet polymers, and spin-dependent recombination. Since 1993, he has been the Director of the Walter Schottky Institute, Technical University Munich, Munich, Germany, where he is the Chair of Experimental Semiconductor Physics II. His current research interests include semiconductor heterostructures and devices for bioelectronics, sensors, spintronics, and photovoltaics. Since 1995, he has been the Editor-in-Chief of the *Physica Status Solidi.*

Dr. Stutzmann received the Walter Schottky Prize of the German Physical Society in 1988. In 2006, he was Elected Fellow of the American Physical Society. He has been the Chair or Co-Chair of more than ten international scientific meetings. He has been involved with numerous scientific committees, e.g., for national and international conferences, the German Science Foundation, and the Alexander-von-Humboldt Foundation.

Paolo Lugli (SM'07) received the Graduate degree in physics from the University of Modena, Modena and Reggio Emilia, Italy, in 1979, and the M.Sc. and Ph.D. degrees in electrical engineering from Colorado State University, Fort Collins, in 1982 and 1985, respectively.

In 1985, he joined the Physics Department, University of Modena, as a Research Associate. From 1988 to 1993, he was an Associate Professor of solid-state physics at the Engineering Faculty, 2nd University of Rome, Tor Vergata, where he became a Full Professor of optoelectronics in 1993. In 2003, he joined the Technical University of Munich, Munich, Germany, where he is currently the Head of the Institute for Nanoelectronics. He has authored more than 250 scientific papers and or coauthored the books "The Monte Carlo Modeling for Semiconductor Device Simulations" (Springer, 1989) and "High Speed Optical Communications" (Kluwer Academic, 1999). His research interests include the modeling, fabrication, and characterization of organic devices for electronics and optoelectronics applications, the design of organic circuits, the numerical simulation of microwave semiconductor devices, and the theoretical study of transport processes in nanostructures.

Dr. Lugli was the General Chairman of the IEEE International Conference on Nanotechnology, Munich.

György Csaba was born in Budapest, Hungary, in 1974. He received the M.S. degree from the Technical University of Budapest, Budapest, Hungary, in 1998, and the Ph.D. degree from the University of Notre Dame, Notre Dame, IN, in 2003.

During 2004–2010, he was a Research Assistant at the Technical University of Munich, Munich, Germany. In 2010, he joined the faculty of the University of Notre Dame. His research current interests include in modeling of nanoscale systems (especially magnetic devices) and exploring their applications for nonconventional circuit architectures.

Chapter 6

Security Applications of Diodes with Unique Current-Voltage Characteristics

Among other things, the last chapter illustrated the pivotal role of the the aluminum-induced layer exchange (ALILE) process for Crossbar PUFs: Firstly, it enables the high rectification rates in the diodes that keep parasitic currents during read-out low. Secondly, it generates the required large entropy and disorder within the crossbars.

In this chapter, we show that the ALILE process can be employed for yet more, enabling disorder-based primitives beyond SHIC PUFs. In greater detail, we firstly show that it can be used for so-called *"certificates of authenticity"* or *"COAs"*. These are unforgeable, offline verifiable product labels, which consists of an unclonable physical structure (i.e., the diodes) plus a crytographic digital signature. Secondly, we illustrate that ALILE-diodes are suited as so-called *"Weak PUFs"* or *"physically obfuscated keys (POKs)"*. These are disorder-based key storage elements that can replace classical non-volatile memory (NVM). Diodes are particularly useful in both applications, since they are simple and very small circuit elements. Towards its end, the chapter also discusses the employment of ALILE-diodes in the context of SHIC PUFs once more in condensed form. While this creates a partial overlap with the last chapter, the high-level treatment of this chapter makes a good one-glance summary on this topic.

Two additional aspects are noteworthy, both of which arose after the original publication of this chapter article in 2010 [130]. Firstly, while the chapter suggests the use of ALILE-diodes as Weak PUFs only in connection with error correction mechanisms and external helper data, it is well conceivable that ALILE-diodes could be used as key storing elements *without* such helper data. The reason is that the observed diode-to-diode variations are much larger than the temperature fluctuations of each single diode, as has been confirmed recently in follow-up work at the Walter Schottky Institut of the TU München [108]. Every diode whose I-V-curve falls into a threshold region could be burnt and disabled by internal circuitry. The other diodes would then provide a stable key storage that functions *without* external helper data. Please note that this diode

selection and destruction could be accomplished easily by internal circuitry. Laborious external measurement or external tuning is not necessary. Such tuning has been suggested in the context of SRAM PUFs [55], for example, but brings about two disadvantages: Firstly, it is time and cost intensive, introducing extra steps in the production process. Secondly, it can also be used by adversaries, enabling cloning attacks on the tunable PUFs, for example SRAM PUFs [54]. The feature that ALILE diodes avoid external tuning and helper data is, to the knowledge of the author, new and unshared by existing Weak PUF designs.

Let us address a second noteworthy aspect. The ALILE process is no standard CMOS process, which represents a clear downside of the our approach. However, it is possible to integrate two different manufacturing process in one system on a chip (SoC). Similar techniques have been suggested in the context of ARM security solutions and INTEL authenticated flash, for example [146]. This has been brought very recently to our attention by R. Kumar and W. Burleson [70]. Even though further studies on this topic are necessary, this may improve the practicality of ALILE diodes for commercial applications.

Overall, the aim of the chapter could be seen as fundamental research that paves the way for new approaches in the area. It shows how crystallization processes can be exploited to achieve PUFs with ultra-high variance.

The publication which makes this chapter is

- U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, M. Stutzmann: *Security Applications of Diodes with Unique Current-Voltage Characteristics*. 14th International Conference on Financial Cryptography and Data Security (FC), 2010. Lecture Notes in Computer Science, Volume 6052, pp. 328-335, Springer, 2010.

A related piece of work at a high-ranking venue, which is explicitly not used in this thesis, is

- C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, M. Stutzmann: *Random p-n-junctions for physical cryptography*. Applied Physics Letters, Vol. 96(172103), 2010.

Security Applications of Diodes with Unique Current-Voltage Characteristics

(Short Paper)

Ulrich Rührmair[1,⋆], Christian Jaeger[2], Christian Hilgers[1], Michael Algasinger[1], György Csaba[3], and Martin Stutzmann[1]

[1] Computer Science Department
[2] Walter Schottky Institute
[3] Institute for Nanoelectronics
TU München, Germany
ruehrmai@in.tum.de, christian.hilgers@mytum.de, csaba@tum.de,
{christian.jaeger,michael.algasinger,stutz}@wsi.tum.de
http://www.pcp.in.tum.de

Abstract. Diodes are among the most simple and inexpensive electric components. In this paper, we investigate how random diodes with irregular $I(U)$ curves can be employed for crypto and security purposes. We show that such diodes can be used to build Strong Physical Unclonable Functions (PUFs), Certificates of Authenticity (COAs), and Physically Obfuscated Keys (POKs), making them a broadly usable security tool. We detail how such diodes can be produced by an efficient and inexpensive method known as ALILE process. Furthermore, we present measurement data from real systems and discuss prototypical implementations. This includes the generation of helper data as well as efficient signature generation by elliptic curves and 2D barcode generation for the application of the diodes as COAs.

Keywords: Physical Cryptography, Physical Unclonable Functions, Certificates of Authenticity, Random Diodes, ALILE Crystallization, SHIC PUFs.

1 Introduction

The use of physical systems with an irregular, at least partly random finestructure recently has gained strong attention in the security and crypto community. In lack of an established, common term, one might call the related field *physical cryptography*, distinguishing it from quantum cryptography or DNA-based approaches. As has been shown in a number of publications starting as early as in the 1980s [1], such disordered physical systems can lead to security applications with enhanced cost efficiency and/or security. Classes of systems that are useful in the area include Strong Physical Unclonable Functions (PUFs) [2] [3] [4], Certificates of Authenticity (COAs) [5] [6], or Physically Obfuscated Keys (POKs) [7] (also called Weak PUFs in [4]).

⋆ Corresponding author.

R. Sion (Ed.): FC 2010, LNCS 6052, pp. 328–335, 2010.

In this paper, we are concerned with the security applications of diodes with irregular $I(U)$ curves. Such diodes have been prepared in our group by a special, crystallization-based fabrication method known as ALILE process [8] [9]. As we are going to show, they can be employed as building blocks for all three named systems, i.e. both for Strong PUFs, COAs and POKs. Furthermore, they are cheap, take very small chip area, and have a good temperature stability. Therefore, so we argue, they have the potential to become a useful and broadly applicable tool in *physical cryptography*.

The paper is organized as follows. In section 2, we explain the ALILE fabrication process for our diodes. Section 3 describes the use of the diodes as COAs or unforgeable labels, and Section 4 discusses their employment as POKs. In Section 5, we illustrate how our diodes can help us to realize a special type of Strong PUF with high information content, which is naturally immune against machine learning attacks. Section 6 concludes the paper.

2 Sample Preparation

For the preparation of the random diodes we use the aluminum-induced layer exchange (ALILE) process [8] [9], which is known to result in polycrystalline films with p-type conduction [10]. This process is used to crystallize amorphous silicon (a-Si) layers exploiting the catalytic effect of aluminum. Here, an Al/oxide/a-Si layer stack is annealed at temperatures below the eutectic temperature of the Al-Si system. Annealing of the sample leads to diffusion of the Si atoms into the Al layer. Crystallite formation occurs where local supersaturation of the Al with Si is achieved. In addition to that, atomic-scale irregularities and defects, e.g. grain boundaries in the Al, can serve as crystallization sites. Thus, the actual crystallization sites can neither be predicted nor controlled, in particular not by the manufacturer of the structure. The same holds for the irregular crystallite growth.

To illustrate the natural randomness of the process, Fig. 1 a depicts the first step of crystallization recorded by an optical microscope, showing the random distribution of the initial crystallization sites. Fig. 1 b illustrates the random crystallite development in later states of the process. In the ALILE-based fabrication of our random diodes, we chose n-type crystalline silicon wafers as the substrate (see Fig. 1 c) [11].

Medium rectification rates of the diodes are observed for diodes prepared on highly doped wafers (e.g. $\rho = 0.003 - 0.007\ \Omega cm$). Such diodes, which exhibit random

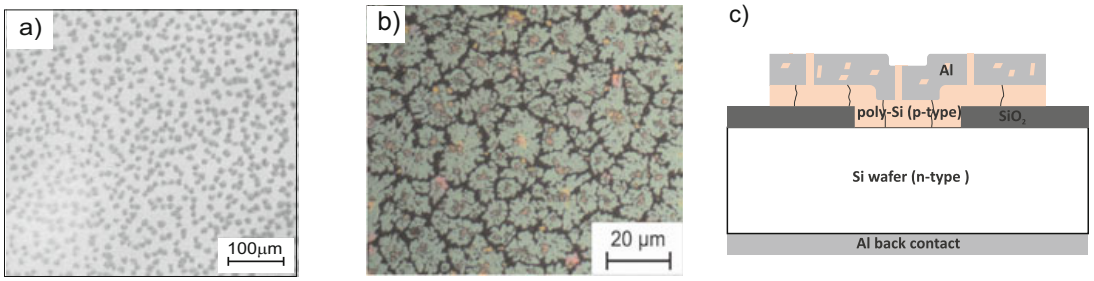

Fig. 1. (a) First crystallites (dark spots) appearing in the Al-matrix during the ALILE process. (b) Irregular growth of the crystallites. (c) Schematic sketch of the diodes' structure.

$I(U)$ characteristics over the whole current-voltage range (see Fig. 2 a), are ideally suited for applications such as electrical COAs (Sec. 3) or POKs (Sec. 4). A very high rectification ratio of the diodes (up to 2×10^7) is obtained for using low doped wafers (e.g. $\rho = 1 - 10\ \Omega cm$); see Fig. 3 a. This high rectification allows the application of the diodes in large crossbars structures with high information density, i.e. as Strong PUFs (see Sec. 5). Further details of the fabrication of ALILE layers and the diode fabrication can be found in [10] [11].

3 Electrically Readable Certificates of Authenticity

The use of a disordered physical structure as unforgeable label in connection with an accompanying digital signature has first been proposed in [1], and was termed Certificate of Authenticity (COA) in [5] [6]. COAs require a unique structure that generates a non-imitable *analog* measurement signal, which must be measured by an *external* measurement device. (Note that in opposition to that, most PUFs generate a digital output and have an integrated measurement device.)

Due to the complex and varying $I(U)$ curves, ALILE-diodes can be employed for said task. They can form cheap COAs whose electrical read-out allows very inexpensive readers.

Prototypical Implementation. To test how many different diodes can be distinguished reliably and repeatedly, we collected measurement data of 16 different individual diodes on one chip (Figure 2 a). For 10 out of 16 diodes we repeated every measurement 5 times, and determined the average $I(U)$ curves by taking the arithmetic mean. We also calculated the maximum deviation and the average deviation from the average $I(U)$ curve. In Figure 2 b, the deviation is given in per cent of the respective average value. We observe a decreasing deviations for higher positive voltages, whereas the deviation is slightly lower in the forward direction of the diodes (negative voltages).

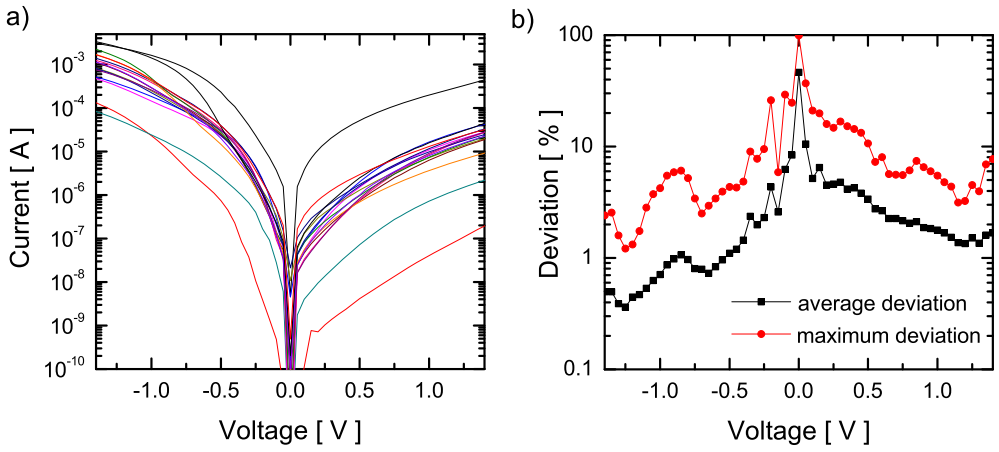

Fig. 2. a) Characteristic $I(U)$-curves of various diodes. b) Average and maximum deviation of the current values upon multiple measurements.

As straightforward helper data for reliable diode identification, the average curves were tabbed at the fixed voltages -1.3 V, -0.65 V, 0.65 V and 1.3 V. An obvious condition for reliable identification is that the average current values at each supporting point must at least allow a deviation as large as the maximum deviation shown in Figure 2 b. The maximum deviation values for our supporting points are as follows: 1.60 at -1.3 V; 2.95 at -0.65 V; 5.67 at 0.65 V; 3.96 at 1.3 V.

The data gathered by us shows that even at a (hypothetical) variance of up to 27% all the 16 diodes could still be distinguished reliably. At the same time, the diodes only showed deviation values of up to 6% in our experiments. This confirms the possibility for realiable identification.

Along the same lines, we executed a first estimation of the overall number of diodes that can be distinguished with the four supporting points at said voltage levels. We assumed a maximal, practically occurring measurement variance of 10% in our calculation, and obtained roughly 160 distinguishable diodes within the broad band, and around 200 distinguishable diodes in the whole current range. To further increase the complexity of the unique, analog COA-signal, experiments are on the way in our group to investigate the frequency spectra arising from networks of 10 to 100 random diodes. Such periodic networks of non-linear components can exhibit rich, complex spectra [12].

To collect additional support for the applicability of random diodes as unforgeable labels, we carried out a prototypical COA implementation on the basis of 2D barcodes. Parameters of interest here are the resulting barcode sizes and longterm security.

We started by selecting a suitable 2D-barcode, choosing the widely used Data Matrix Code, and implemented it by use of the libdmtx library [13]. Due to the limited storage capacity of barcodes, shorter signatures than RSA are preferred in the generation of COAs; our implementation is based on the bilinear pairing based scheme by Zhang, Safavi-Naini und Susilo (ZSS) [14], which allows signatures of only 160 bits. For the implementation we chose the PBC library [15] with the elliptic curve type F. We assumed that a single waver with 20 diodes is applied as unique object, and that the following information must be stored on the product: Manufacturer ID, product related information (16+48 bit); helper data (20 x 14 bit); digital signature (160 bit). Using a barcode module width 0.25 mm, this leads to a barcode of size 0.81 cm^2. We succesfully generated such a barcode with data from our real measurement data and for exemplary product related data.

Our diode-based approach to COAs therefore leads to inexpensive labels with barcode sizes of less than 1 cm^2. It allows one of the first electrical COAs with high security and complex analog output; previous COAs were mainly based on optical structures or radiowave scatterers. According to the estimate given in [16], the employed 160-bit elliptic curve signature will be secure until 2019. Signature security until the year 2050 is possible, again on the basis of elliptic curves, with key bitlength around 206 [16] and barcode sizes of still around 1 cm^2.

4 Physically Obfuscated Keys from Random Diodes

Random physical structures can also be used as a non-volatile storage for secret binary keys. Due to their disordered and/or tamper sensitive nature, they may be harder to

extract invasively than binary keys stored in EEPROM, for example. This concept has been termed a Physically Obfuscated Key (POK) [7], a Weak PUF [4] or also an obfuscating PUF [3]. Applications of POKs naturally include any cryptographic protocols based on secret binary keys, including hardware identification schemes of all sort. They are particularly well suited to store keys safely in small, inexpensive mobile systems, where effective key protection is otherwise difficult to achieve. As we are going to show, random ALILE-diodes can also be used as cheap, stable POKs with remarkably high information density.

Reliable Key Extraction. In the application of ALILE-diodes as POKs, our focus lies on the highly robust extraction of a string (the later key) from the $I(U)$ curves in Fig. 2a). In opposition to COAs, our helper data furthermore should not reveal any information about the binary key which it helps to extract from the POK (see also [17]), since the key must remain secret. We applied ideas taken from Linnartz et. al [18], where the y-axis of the verification measurement is split in equal sections, and the measured data points are shifted towards the arithmetic mean of these sections (i.e. away from the section borders in order to avoid bit flips) by the helper data.

Our data base were the $I(U)$-curves of the 16 diodes that we already used in section 3. Once more, we set the four supporting points at -1.3 V, -0.65 V, 0.65 V and 1.3 V. Our aim is to extract one bit from the current value at each of the four supporting points, four bits in total per $I(U)$-curve. Inspired by [18], we proceeded as follows: Firstly, we calculated at each supporting point k ($k = 1, \ldots, 4$) the median c_k of the current values of all diodes at this supporting point. Secondly, for each supporting point k, we divided the current-axis into 8 sections. Each section i ($i = 1, \ldots, 8$) its determined by its lower border b_i^k and upper border b_{i+1}^k, where $b_i^k = ((p+1)/(1-p))^{i-4} \cdot c_k$ for $i = 1, \ldots, 8$. In other words, the sections are of equal length on a logarithmic scale, and center around $b_4^k = c_k$. We choose $p = 0.5$ to compensate measurement errors of up to +/-50%. We further denote the arithmetic mean of the section i (with the borders b_i^k and b_{i+1}^k) as $m_{i,i+1}^k$. As is supported by our measurement data, we assume that the measurement points are distributed approximately uniformly over all sections. Under these circumstances, the helper data leaks few/none information about the extracted bit; see also [18].

During the enrollment phase of the POK at the manufacturer, we generate for every measurement s_k at the supporting point k helper data h_k in the following way:

$$h_k = \frac{m_{i,i+1}^k}{s_k} \quad \text{for the unique} \quad i \in \{1, \ldots, 8\} \quad \text{that satisfies} \quad b_i^k \leq s_k < b_{i+1}^k \quad (1)$$

During the verification the extracted bit $x(k)$ can be computed with a verification measurement v_k at supporting point k:

$$x(k) = \begin{cases} 0 & \text{if} \quad b_{2i}^k \leq h_k v_k < b_{2i+1}^k \\ 1 & \text{if} \quad b_{2i+1}^k \leq h_k v_k < b_{2i+2}^k \end{cases} \quad (2)$$

With $p = 0.5$ we could obtain 11 different bit strings out of the 16 diodes, while the helper data leaks less information about the bit strings. This means that at least 3 bits

per diode can be extracted in a stable manner and at an error compensation rate of 50% measurement deviation. Our results suggest the usability of one of the simplest and smallest electrical components – namely diodes – as POKs.

5 Machine Learning Resistant Strong PUFs via Crossbar Structures

A Strong PUF is a physical system S which meets the following requirements: (i) S can be excited with external stimuli or challenges C_i, upon which it reacts with corresponding responses R_{C_i}. (ii) It is infeasible, even for the original manufacturer of S, to produce a second system S' which has the same challenge-response-behavior as S. (iii) It is difficult for an adversary to correctly predict an unknown response R_C to a randomly chosen challenge C numerically, without conducting an actual measurement on S. This security feature shall hold even if many other challenge-response pairs (C_i, R_{C_i}) are known to the adversary, or if he had previous physical access to S for a limited period, during which he could conduct any physical measurement on S. In theory, these properties can be met due to the high disorder/information content and/or the complex internal model of S.

Applications of Strong PUFs include identification and key establishment between central authorities and mobile decentral systems [2] [19]. Their complex challenge-response behavior is sufficient to guarantee security in such applications. No execution of costly asymmetric schemes in the mobile systems is necessary.

Current candidates for electrical Strong PUFs contain only a relatively small (max. several hundreds) of *interacting* components. Thus, relatively few (again some hundred) internal parameters completely determine their behavior. This is one of the main reasons why basically all of them have been attacked successfully by machine learning techniques [3] [20]. An alternative design route to Strong PUFs, that has been suggested by our group in [21], is to employ as many (up to billions), densely packed random subunits as possible, which are read out *individually* and *independently* of each other. Our principle is comparable to a read-only memory with maximal size, random information content, and intrinsically limited read-out rate. We showed in [21] that large, monolithic, memory-like crossbar structures (Fig. 3b) based on random diodes are very well suited to realize this approach. Due to their simple and regular geometry, they can reach optimal information densities (up to 10^{10} to 10^{11} bits per cm^2). The crossbars can be designed in such a way that (i) parallel read-out of different memory units (i.e. diodes) is impossible; (ii) faster read-out than a preset limit leads to overloading and immediate destruction of the wiring, rendering the remaining structure unreadable. Note that the slow read-out rate is not enforced by an artificially slow access module or the like, but by the inductive and resistive capacitances of the structure itself [22].

The resulting Crossbar PUFs are provably immune against machine learning attacks: Their security merely depends on the access time of the adversary, and on the ratio of the already read-out bits vs. the number of overall bits stored in the structure. Modeling attacks subsequent to the read-out are fruitless, since all components are independent of each other. The exact security properties of Crossbar PUFs thereby depend on the employed circuit technology. With a 30nm technology, for example, Crossbar PUFs

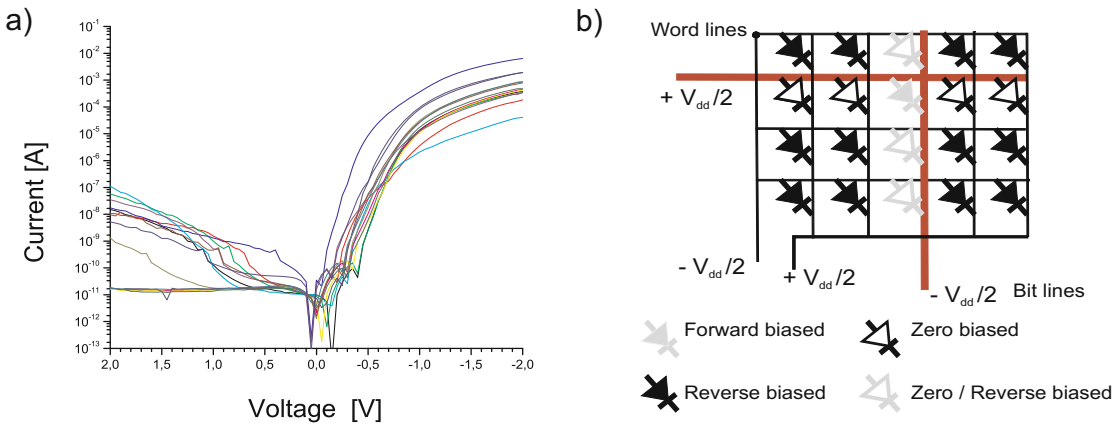

Fig. 3. a) $I(U)$ curves of diodes with high rectification rates; b) schematics of a crossbar structure

of size 1 cm^2 could achieve security of up to 3 years of continuous, uninterrupted adversarial access, while enabling read-out rates of 10^3 bits per second [21]. They could easily be implemented in plug-in devices and on chipcards. Note that all current Arbiter PUFs and variants that run at a 1 MHz CRP frequency become susceptible to modeling attacks after less than a second of uninterrupted adversarial read-out [3] [20].

One prerequisite left open in [21] was whether random diodes with a rectification ratio of at least 10^5 could be produced by inexpensive techniques. Such high rectification rates are necessary to realize stable read-out and to limit parasitic current paths in the monolithic, large crossbar [21] [22] . We have now been able to fabricate diodes with even higher rectification by use of the ALILE process (Fig. 3 a). They indeed enable the first electrical PUFs that remain secure in the face of adversarial access of up to years and against machine learning attacks, further illustrating the security potential of random diodes. We suggest the term SHIC PUFs (pronounce as *"chique PUFs"*) for this new type of PUF, where the acronym SHIC stands for Super High Information Content.

6 Summary

We have argued on the basis of real measurement data and prototypical implementations that random, irregular diodes can be applied for the construction of COAs, POKs and Strong PUFs at the same time. They have the advantage of being one of the smallest and simplest electrical components, and that they can be produced by inexpensive methods. This gives them a strong potential for physical cryptography applications.

Acknowledgements

The presented work was conducted within the Physical Cryptography Project at the TU München. We acknowledge financial support by the International Graduate School of Science and Engineering (IGSSE) and the Institute for Advanced Study (IAS) at the TU München. We thank Michael Scholz and Matthias Bator for useful discussions.

References

1. Bauder, D.W.: An Anti-Counterfeiting Concept for Currency Systems. Research report PTK-11990. Sandia National Labs, Albuquerque, NM (1983)
2. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical One-Way Functions. Science 297, 2026–2030 (2002)
3. Rührmair, U., Sölter, J., Sehnke, F.: On the Foundations of Physical Unclonable Functions, http://eprint.iacr.org
4. Tuyls, P., Schrijen, G.J., Skoric, B., van Geloven, J., Verhaegh, N., Wolters, R.: Read-Proof Hardware from Protective Coatings. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 369–383. Springer, Heidelberg (2006)
5. Vijaywargi, D., Lewis, D., Kirovski, D.: Optical DNA. In: Financial Cryptography 2009, pp. 222–229 (2009)
6. DeJean, G., Kirovski, D.: RF-DNA: Radio-Frequency Certificates of Authenticity. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 346–363. Springer, Heidelberg (2007)
7. Gassend, B.: Physical Random Functions, MSc Thesis, MIT (2003)
8. Nast, O., Wenham, S.R.: Elucidation of the layer exchange mechanism in the formation of polycrystalline silicon by aluminum-induced crystallization. Journal of Applied Physics 88, 124–132 (2000)
9. Nast, O., Hartmann, A.J.: Influence of interface and Al structure on layer exchange during aluminum-induced crystallization of amorphous silicon. Journal of Applied Physics 88, 716–724 (2000)
10. Antesberger, T., Jaeger, C., Scholz, M., Stutzmann, M.: Structural and electronic properties of ultrathin polycrystalline Si layers on glass prepared by aluminum-induced layer exchange. Appl. Phys. Lett. 91, 201909 (2007)
11. Jaeger, C., Algasinger, M., Rührmair, U., Csaba, G., Stutzmann, M.: Random pn-junctions for physical cryptography. Appl. Phys. Lett. 96, 172103 (2010)
12. Berkemeier, J., Dirksmeyer, T., Klempt, G., Purwins, H.-G.: Pattern Formation on a Nonlinear Periodic Electrical Network. In: Zeitschrift für Physik B Condensed Matter. Springer, Heidelberg (1986)
13. Laughton, M.: (2009), http://www.libdmtx.org/documentation.php
14. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer, Heidelberg (2004)
15. Lynn, B., et al. (2009), http://crypto.stanford.edu/pbc/
16. Lenstra, A.K.: Selecting cryptographic key sizes. Journal of Cryptology (2001)
17. Guajardo, J., Kumar, S., Schrijen, G., Tuyls, P.: FPGA Intrinsic PUFs and Their Use for IP Protection. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 63–80. Springer, Heidelberg (2007)
18. Linnartz, J.P., Tuyls, P.: New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 393–402. Springer, Heidelberg (2003)
19. Suh, G.E., Devadas, S.: Physical Unclonable Functions for Device Authentication and Secret Key Generation. In: DAC 2007, pp. 9–14 (2007)
20. Rührmair, U., Sehnke, F., Soelter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling Attacks on Physical Unclonable Functions, http://eprint.iacr.org
21. Rührmair, U., Jaeger, C., Bator, M., Stutzmann, M., Lugli, P., Csaba, G.: Applications of High-Capacity Crossbar Memories in Cryptography. Accepted at IEEE Transactions on Nanotechnology (to appear)
22. Csaba, G., Lugli, P.: Read-out design rules for molecular cross bar architectures. IEEE Transactions on Nanotechnology 8(3), 369–374 (2009)

Part III

Disorder-Based Hardware Beyond PUFs: SIMPL Systems

Chapter 7

SIMPL Systems, Or: Can we Design Cryptographic Hardware without Secret Key Information?

In this third part of the thesis, we deal with so-called *"SIMPL systems"*, or just *"SIMPLs"*. As mentioned earlier, the acronym stands for the term SIMulation Possible, but Laborious. SIMPLs could be seen as a public key version of PUFs: Similar to PUFs, each SIMPL system is physically unique and thus difficult to clone for the adversary, and exhibits some form of challenge-response behavior. Contrary to PUFs, however, each SIMPL possesses a public numeric description of its internal disorder, which allows everyone to simulate its challenge-response pairs (CRPs). Any such numeric simulation, and also any other, hardware-based emulation, is supposed to be notably slower than the real-time behavior of the SIMPL system, however. Taken together, these features allow a number of public-key like cryptographic protocols and applications that extend the applicability of pure PUFs. They include hardware identification protocols, but also more advanced schemes like message authentication, bit commitment, or coin flipping.

One specific highlight of SIMPLs is that they can be implemented without containing any security-critical information in the sense of Section 1.5 — or, in the language of this Chapter 7 and its title, without containing any secret key information. Recall from our treatment in Section 1.5 that this feature is not shared by the most popular PUF implementations, most notably SRAM PUFs and Arbiter PUF variants, and that currently no electrical PUF implementations with this feature are known at all. This makes SIMPL systems a particularly interesting tool. On the other hand, the supposed time difference between any adversarial emulation, the honest users of the system, and the real-time behavior of the physical SIMPL systems is an assumption that is non-trivial to fulfill, and which classical PUFs do not need to meet.

We take some time in this third part to successively unfold the concept of SIMPLs. In this Chapter 7, we give an overview of the basic ideas, implementations, and protocols of the concept. The upcoming chapters then treat certain aspects of SIMPLs in greater detail, namely their implementation (Chapter 8) and their use in cryptographic protocols (Chapter 9). The chapters have some overlap when it comes to the specification of SIMPLs and their basic use in identification and message authentication protocols. We encourage the readers to skip the respective parts in the later chapters. Still, each of them makes its own contributions to the bigger picture, and delivers its part in exploring uses and implementations of SIMPLs.

The publication used in this Chapter 7 is:

- U. Rührmair: *SIMPL Systems, Or: Can We Design Cryptographic Hardware without Secret Key Information?* 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), 2011. Lecture Notes in Computer Science, Vol. 6543, pp. 26-45, Springer, 2011.

An earlier, related piece of work, which is explicitly not used in this thesis, is

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

SIMPL Systems, or: Can We Design Cryptographic Hardware without Secret Key Information?

Ulrich Rührmair

Computer Science Department
Technische Universität München
Boltzmannstr. 3
85748 Garching
ruehrmai@in.tum.de
http://www.pcp.in.tum.de

Abstract. This paper discusses a new cryptographic primitive termed *SIMPL system*. Roughly speaking, a SIMPL system is a special type of Physical Unclonable Function (PUF) which possesses a binary description that allows its (slow) public simulation and prediction. Besides this public key like functionality, SIMPL systems have another advantage: No secret information is, or needs to be, contained in SIMPL systems in order to enable cryptographic protocols — neither in the form of a standard binary key, nor as secret information hidden in random, analog features, as it is the case for PUFs. The cryptographic security of SIMPLs instead rests on (i) a physical assumption on their unclonability, and (ii) a computational assumption regarding the complexity of simulating their output. This novel property makes SIMPL systems potentially immune against many known hardware and software attacks, including malware, side channel, invasive, or modeling attacks.

1 Introduction

Background and Motivation. Electronic communication and security devices are pervasive in our life. Just to name two examples, around five billion mobile phones are currently in use worldwide [1] [2], and the world market of smart cards has an estimated volume of over three billion pieces per year [3] [4]. Their widespread use makes such devices both a well-accessible and a worthwhile target for adversaries. Many security attacks are thereby not directed against the employed cryptographic primitives themselves, some of which have proven attack-resilient over surprisingly long time spans. Rather, they often apply physical or software attacks in order to extract the employed secret keys. Such key-extracting strategies are not just a theoretical concern, but have been demonstrated several times in widespread, commercial systems [5] [6] [7]. This drives the search for new mechanisms that protect — or better still: avoid! — secret keys in vulnerable hardware.

Physical Unclonable Functions (PUFs). The security primitive of a Physical Unclonable Function (PUF) [8] [9] [10] [11] was introduced, at least in part, in order to address some of the above problems. A PUF is a (partly) disordered physical system S that can be challenged with so-called external stimuli or challenges C_i, upon which it reacts with

I. Černá et al. (Eds.): SOFSEM 2011, LNCS 6543, pp. 26–45, 2011.

corresponding responses termed R_{C_i}. Contrary to standard digital systems, a PUF's responses shall depend on the nanoscale structural disorder present in the PUF. This disorder cannot be cloned or reproduced exactly, not even by its original manufacturer, and is unique to each PUF. Assuming the stability of the PUFs responses, any PUF S hence implements an individual function F_S that maps challenges C_i to responses R_{C_i} of the PUF. Due to its complex and disordered structure, a PUF can avoid some of the shortcomings associated with digital keys. For example, it is usually harder to read out, predict, or derive its responses than to obtain the values of digital keys stored in non-volatile memory. This fact has been exploited for various PUF-based security protocols [8] [9] [15] [22].

One prominent example are PUF-based identification schemes [8] [9] [10]. They are usually run between a central authority (CA) and a hardware carrying a (unique) PUF S. One assumes that the CA had earlier access to S, and could establish a large, secret list of challenge-response-pairs (CRPs) of S. Whenever the hardware wants to identify itself to the CA at some later point in time, the CA selects some CRPs at random from this list, and sends the challenges contained in these CRPs to the hardware. The hardware applies these challenges to S, and sends the obtained responses to the CA. If these responses match the pre-recorded responses in the CRP-list, the CA believes the identity of the hardware. Note that each CRP can only be used once, whence the CRP-list uses up over time, and needs to be large.

Private Key like Functionality of PUFs. The described protocol has several well-known advantages [8] [9]. However, one potential downside is that it presumes a previously shared piece of secret numerical information (i.e., the CRP-list). This information needs to be established in a secure set-up phase between the CA and the hardware, and must constantly be kept secret. In this particular structural aspect, PUFs are resemblant of classical private key systems.

Secret Information in PUFs. Another noteworthy point is that PUFs in general do not obviate the presence of secret information within cryptographic hardware. The secret information is no longer stored in digital form in two-level systems, such as digital secret keys stored in non-volatile memory cells. But still there is some sort of secret information present in most PUFs, whose disclosure breaks the security of the system. Let us name two examples to illustrate our point: In the case of SRAM PUFs the information that needs to be kept secret is the state of the SRAM cells after power up, or the tiny manufacturing variations of the SRAM cells that determine their state after power up [25]. Once this information is known to an adversary, he can numerically derive the same key as the cryptographic hardware embedding the SRAM PUF. In the case of Arbiter PUFs, the secret information are the internal runtime delays in the circuit stages [11]. If this information is known, the adversary can numerically simulate the behavior of the PUF output by an additive, linear model, again breaking its security [26].

In other words, the architectures of most current PUFs "hide" or "obfuscate" secret, security-relevant information very well in analog characteristics of integrated circuits. But at the same time, they do not avoid the need for secret information in hardware systems in principle; they just store it in a different form.

Our Contributions. This paper introduces a novel security primitive called a *SIMPL system*, whereby the acronym SIMPL stands for "SIMulation Possible, but Laborious". These systems have two interesting conceptual advantages: First, they are a PUF-like security tool, but possess some type of public key functionality. This improves their practical applicability. Second, they obviate the need for secret information in cryptographic systems, trading it for two other assumptions: (i) their physical unclonability, and (ii) the assumed computational overhead of numerically simulating their output (in comparison with their faster real-time behavior). We show that SIMPLs can realize basic communication protocols such as identification and message authentication, and briefly describe the application of these protocols in some concrete settings. We also discuss implementations of SIMPL systems, thereby surveying existing approaches, and propose a new optical implementation strategy. Proof-of-concept data on this optical implementation, which arose from other, recent research activities in our group [40], is presented in the appendix.

Related Work. The current paper is an extended version of [16]. Since the publication of [16], several papers of our group have dealt with the implementation of SIMPLs by integrated circuits [17] [18] [19] [20]. We emphasize that around the same time as [16], a comparable concept has been described independently in [21] under the name of a Public PUF (PPUF).

While stressing that both pieces of work are very interesting, let us briefly address a few differences between [21] and our studies. One difference is that we focus on SIMPL systems/Public PUFs for which the *relative* speed difference between the real hardware and the simulation is comparably low, for example only a small constant factor. Such systems seem to have milder complexity requirements and less stability issues. We argue that by applying feedback loops, not the relative, but the absolute time difference between such systems and any emulation can still be amplified to a sufficient absolute value. Once this absolute value is large enough, it enables secure identification and message authentication protocols, and could compensate network or other delays. Another reason for concentrating on systems with small speed gaps lies in the fact that the verification step in identification and message authentication must be carried out relatively efficiently (see Protocols 2 and 3).

Second, we center upon applications where the main advantage of SIMPL systems — that they can build security systems without secret key information — is most relevant. Two typical examples are the named identification and message authentication schemes. Should a shared secret key between two parties be required in a SIMPL-based communication infrastructure (for example in order to achieve confidentiality), SIMPL-based message authentication can be used together with the Diffie-Hellman protocol to exchange a session key. But this key ideally will not be stored permanently in the system.

To the contrary, [21] discuss a PPUF-scenario where a one-time, permanent secret key is exchanged in a computationally relatively intensive scheme. This scheme appears too time consuming for multiple session key exchange. Their setting hence puts key exchange on different security assumptions than classical protocols (like Diffie-Hellman), which is a strong achievement on its own. But they do not attempt to generally avoid the long-term presence of secret information in cryptographic hardware, as we aspire with SIMPL systems.

Finally, a very interesting and recommendable, but later source is [24], where time-bounded authentication for FPGAs is discussed.

Organization of this Paper. The rest of this manuscript is organized as follows: In Section 2, we give a semi-formal specification of SIMPL systems, and discuss their properties. Section 3 provides two formal SIMPL-based protocols for entity identification and message authentication. In Section 4 we discuss implementation candidates for SIMPL systems, and conclude the paper in Section 5.

2 SIMPL Systems and Their Properties

2.1 Informal Description of SIMPL Systems

We start by informally listing the properties of a SIMPL system. A physical system S is called a *SIMPL system* (or just a *SIMPL*) if the following holds:

1. S is a (partly) disordered physical system. It can be stimulated with challenges C_i, upon which it reacts with corresponding responses R_{C_i}. The responses are a function of the specific disorder present in S, and of the applied challenge C_i. The responses are assumed to be sufficiently stable to regard the behavior of S as a function F_S that maps challenges C_i to responses R_{C_i}.
2. Given a challenge C_i, it is possible to numerically simulate the corresponding response R_{C_i} of S with high accuracy. The simulation is carried out via an individual, public description $D(S)$ of S, and a public simulation algorithm Sim.
3. Any feasible algorithm, or any physical emulation, that predicts the responses of S correctly (i.e., which computes F_S), is noticeably slower than the real-time behavior of S.
4. It is difficult to physically clone S, i.e. to produce a second system S' which generates the same responses on almost all possible challenges with comparable speed. This must hold even if the internal characteristics and disorder of S, the description $D(S)$, and many CRPs of S are known.

Put in one sentence, the holder of a secure SIMPL system S is able to evaluate a publicly known, publicly computable individual function F_S *faster* than anyone else.

2.2 Semi-formal Specification of SIMPL Systems

The above properties can also be coined into a semi-formal specification of SIMPL systems. The style of the specification follows the specifications and definitions that have been presented in [22]. It specifies the security of SIMPL systems as a "game" with the adversary, thereby introducing a relatively precise adversarial model.

Specification 1 ($(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL SYSTEMS.). *Let S be a physical system mapping challenges C_i to responses R_{C_i}, with $\mathbf{C}$ denoting the finite set of all possible challenges. Let $c > 1$ be a constant, and let furthermore t_{max} be the maximum time (over all challenges $C_i \in \mathbf{C}$) which it takes until the system S has generated the response R_{C_i} to the challenge C_i. S is called a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-*SIMPL SYSTEM *if there is a string $D(S)$, called the description of S, and a computer algorithm* Sim *such that the following conditions are met:*

1. *For all challenges $C_i \in \mathbf{C}$, the algorithm* Sim *on input $\big(C_i,\ D(S)\big)$ outputs R_{C_i} in feasible time.*
2. *Any cryptographic adversary Eve will succeed in the following* **security experiment** *with a probability of at most ϵ:*
   (a) *Eve is given the numerical description $D(S)$ and the code of the algorithm* Sim *for a time period of length t_C.*
   (b) *Within the above time period t_C, Eve can q times adaptively query an oracle O for arbitrary responses R_{C_i} of S .*
   (c) *Within the above time period t_C, Eve is furthermore given physical access to the system S at adaptively chosen time points, and for time periods of adaptively chosen lengths. The only restriction is that her access times must add up to a total of at most t_{Ph}.*
   (d) *After the time period t_C has expired, Eve is presented with a challenge C_{i_0} that was chosen uniformly at random from the set $\mathbf{C}$, and is asked to output a value V_{Eve}.*

   We say that Eve succeeds in the described experiment if the following conditions are met:

   (i) $V_{Eve} = R_{C_{i_0}}$.
   (ii) The time that Eve needed to output V_{Eve} after she was presented with C_{i_0} was at most $c \cdot t_{max}$.

   Please note that the said probability of ϵ is taken over the uniformly random choice of $C_{i_0} \in \mathbf{C}$, and the random choices or actions that Eve might take in steps 2a, 2c and 2d.

Discussion. Let us briefly discuss the security model underlying Specification 1. In practical applications of SIMPL systems, Eve can gather information about S in three ways: (i) She analyzes the algorithm Sim and the description $D(S)$, which are both public. (ii) She collects as many challenge-response-pairs (C_i, R_{C_i}) of S as possible from external sources, for example protocol eavesdropping. (iii) Eve *physically* measures the system S. She may determine CRPs by such measurements, but also other, more general characteristics of the system.

These three types of attacks must be covered in our security model, and they are: Possibility (i) is covered in item *2a* of Spec. 1, (ii) is reflected in item *2b*, and (iii) is implicit in item *2c*. Since the physical access time and the time in which Eve can prepare her attack by previous computations differ strongly in most application scenarios, it makes sense to distinguish between t_{Ph} and t_C in Spec. 1.

We also chose the value c, which describes the time gap between Eve and the SIMPL system, to be a flexible system parameter. This keeps the definition general and allows its application to different types of SIMPLs. In many practical applications, even small values (e.g. around 2) may suffice for c. See also the discussion in Section 2.3, paragraphs on *constant vs. super-polynomial time gap* and *feedback loops*.

2.3 Properties of SIMPL Systems

Let us discuss a few properties of SIMPL systems implied by Specification 1.

Immunity against ϵ-fraction Read-out and Simulation. Spec. 1 implies that for any SIMPL system it must be impossible to measure the values R_{C_i} for more than an ϵ-fraction of all parameters $C_i \in \mathbf{C}$ within time t_{Ph}. Otherwise, Eve could create a lookup table for an ϵ-fraction of all possible values R_{C_i} during step 2c. This could enable her to succeed in the described experiment with probability greater than ϵ. Therefore, for any SIMPL system the set of possible measurement parameters $\mathbf{C}$ must be very large.

For the same reasons, it must be impossible for Eve to determine more than an ϵ-fraction of all CRPs within time t_C by exhaustive simulation on the basis of Sim and $D(S)$. This again implies that $\mathbf{C}$ must be very large (for example exponential in some system parameter), and/or that the simulation must be time consuming.

Immunity against Cloning. Spec. 1 also implies that previous physical access for time t_{Ph} and computations of time t_C do not allow Eve to build a *physical clone* S' of the system S, for whose responses R'_{C_i} it holds that

$$R_{C_i} = R'_{C_i} \quad \text{for more than an } \epsilon\text{-fraction of all } C_i \in \mathbf{C},$$

and for which the evaluation of the R'_{C_i} works within time $c \cdot t_{max}$. Spec. 1 both rules out the possibility to build an exact physical reproduction of S, or the feasibility to fabricate a *functional* clone, i.e., a physical system of a possibly very different structure or lengthscale than S, which still generates its response R'_{C_i} within time $c \cdot t_{max}$.

Constant vs. Super-polynomial Time Gap. Spec. 1 stipulates that the time gap between Eve and the real SIMPL system S must be at least a constant factor $c > 1$. This seems surprising: Being used to the formalism of complexity-based classical cryptography, one might expect the stipulation of an exponential gap. But it is unclear whether SIMPLs with an exponential time margin between Eve and the SIMPL exist at all. The only known, realistic computational systems which might outperform Turing architectures by a super-polynomial factor are quantum computers [35]. But standard quantum computers possess no immunity against physical cloning, since they could be mass-fabricated with the same functionality. They are hence unsuited as SIMPL systems. A further setback in the search for SIMPLs with an exponential security margin is that it has been frequently hypothesized within the computational complexity community that there are no realistic hardware systems that solve NP-complete problems efficiently in practice, i.e. by using polynomial resources. Two recent sources in this context are [33] [34].

Still, meaningful applications for SIMPL systems may not require exponential speed gaps. In the appliances we suggest in this paper (namely identification and on-the-fly message authentication), a constant, detectable time difference suffices. An exponential time gap between the SIMPL system and any simulation machine may even be undesirable there, since it could lead to time consuming verification steps in the Protocols 2 and 3.

Feedback Loops. In order to enable large absolute time margins, the absolute (but not the relative!) time difference between the original SIMPL system and any fraudster can be amplified via feedback loops. In a nutshell, such feedback-loops can be set up as follows: Presented with a challenge C_1, the SIMPL systems successively determines a

sequence of k challenge-responses-pairs $(C_1, R_{C_1}), (C_2, R_{C_2}), \ldots, (C_k, R_{C_k})$, where later challenges C_n are determined by earlier results R_{C_m}, with $k \geq n > m \geq 1$. The tuple (C_1, R_{C_k}) can then be regarded as the overall challenge-response pair determined by the SIMPL. The application of such feed-back loops can help us to compensate network and transmission delays.

Let us make a concrete example in order to illustrate our point. Suppose that we possess a SIMPL system S which produces its responses in t_{max} of 10 nanoseconds (ns), and which possesses a speed advantage of $c = 2$ over all simulations. That means that any adversary cannot produce the response to a randomly chosen challenge within 20 ns. This tiny difference would not be detectable in many practical settings, for example in large networks with natural delays. Nevertheless, the application of repeated feedback loops can amplify not the relative, but the absolute time margin, such as to 1 millisecond (ms) vs. 2 ms, or also 1 sec vs. 2 sec.

SIMPLs with Multi-bit Output. In some applications, it is found convenient if a SIMPL system produces not just one bit as response, but a multi-bit output. Some implementations of SIMPLs have this property naturally (for example the optical implementation of section 4.3). Otherwise, feedback loops can allow us to create multi-bit outputs from SIMPL systems with 1-bit outputs: One simply considers a concatenation (or some other function, for example a hash function) of the last n responses $R_{C_{k-n+1}}, \ldots, R_{C_k}$ in the feedback loop. This concatenation (or function) can be taken as the overall output of the SIMPL.

Another option to create "large" SIMPL systems with k-bit outputs from "small" SIMPL systems with 1-bit outputs is to employ k such SIMPL systems in parallel, and to directly concatenate their responses to produce a k-bit overall output. This method has been suggested already in the context of PUFs in [13].

Error Correction. Please note that in the Spec. 1, in the above discussion, and also in the upcoming protocols in Section 3, we assume that the responses of the SIMPL system are stable. In practice, error correction and helper data must, and can, be applied to achieve this goal; see, for example, [9] [37] [38] [39].

3 Protocols and Applications

We will now describe two exemplary protocols that can be realized by SIMPL systems, and discuss some application scenarios.

3.1 Identification of Entities

We assume that Alice holds an individual $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system S, and has made the corresponding data $D(S)$, Sim, the value $c \cdot t_{max}$, and a description of $\mathbf{C}$ public. Now, she can prove her identity to an arbitrary second party Bob, who knows $D(S)$, Sim, $c \cdot t_{max}$ and $\mathbf{C}$, as follows (with k being the security parameter of the protocol):

Protocol 2: IDENTIFICATION OF ENTITIES BY SIMPL SYSTEMS

1. Bob chooses k challenges $C_1, \ldots, C_k$ uniformly at random from $\mathbf{C}$.
2. **For** $i = 1, \ldots, k$ **do**:
   (a) Bob sends the value C_i to Alice.
   (b) Alice determines the corresponding response R_{C_i} by an experiment on her SIMPL system S, and sends this value to Bob.
   (c) Bob receives an answer from Alice, which we denote by V_i. If Alice's answer did not arrive within time $c \cdot t_{max}$, then Bob sets $V_i = \perp$ and continues the for-loop.
3. Bob computes the value $R_{C_i}^{Sim} = \mathsf{Sim}(C_i, D(S))$ for all $i = 1, \ldots, k$, and verifies if $R_{C_i}^{Sim} = V_i \neq \perp$. If this is the case, Bob believes Alice's identity, otherwise not.

Discussion. In a nutshell, the security of the protocol follows from the fact that an adversary is unable to determine the values R_{C_i} for randomly chosen C_i comparably quickly as Alice, provided that: (i) The lifetime of the system S (and the period since $D(S)$ was made public) does not exceed t_C, and (ii) the adversary's accumulated physical access times do not exceed t_{Ph} (see Spec. 1). In that case, the adversary's probability to succeed in the protocol without possessing S decrease exponential in k.

Bob can improve his computational efficiency by verifying the correctness of the responses R_{C_i} only for a randomly chosen subset of all responses. If necessary, possible network and transmission delays can be compensated for by amplifying the absolute time gap between Eve and S through feedback loops (see Section 2.3).

If the SIMPL system has multi-bit output, then a value of $k = 1$, i.e. a protocol with one round, may suffice. In these cases, the parameter ϵ of the multi-output SIMPL system will in itself be exponentially small in some system parameter (for example in the size of the sensor array in the optical SIMPLs discussed in Section 4).

3.2 Authentication of Messages

Alice can also employ an individual $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system S in her possession to authenticate messages to Bob. Again, we suppose that the values $D(S)$, Sim, $c \cdot t_{max}$, and a description of $\mathbf{C}$ are public.

Protocol 3: AUTHENTICATION OF A MESSAGE N BY SIMPL SYSTEMS

1. Alice sends the message N, which shall be authenticated, to Bob.
2. Bob chooses $k \cdot l$ challenges $C_1^1, \ldots, C_k^1, C_1^2, \ldots, C_k^2, \ldots, C_1^l, \ldots, C_k^l$ uniformly at random from $\mathbf{C}$.
3. **For** $i = 1, \ldots, l$ **do**:
   (a) Bob sends the values $C_1^i, \ldots, C_k^i$ to Alice.
   (b) Alice determines the corresponding responses $R_{C_1^i}, \ldots, R_{C_k^i}$ by experiments on her SIMPL system S.
   (c) Alice derives a MAC-key K_i from $R_{C_1^i}, \ldots, R_{C_k^i}$ by a publicly known procedure, for example by applying a publicly known hash function to these values. She sends $MAC_{K_i}(N)$ to Bob.

(d) Let us denote the answer Bob receives from Alice by V_i. If V_i did not arrive in time $c \cdot t_{max} + t_{MAC}$, where t_{MAC} is the time to derive K_i and compute $MAC_{K_i}(N)$, then Bob sets $V_i = \perp$ and continues the for-loop.

4. For $i = 1, \ldots, k$ and $j = 1, \ldots, l$, Bob computes the values $R_{C_i^j}^{Sim} = \mathsf{Sim}(C_i^j, D(S))$ by simulation via Sim. He derives the keys $K_1^{Sim} \ldots, K_k^{Sim}$ by application of the same procedure (e.g. the same publicly known hash function) as Alice in step 3c.
5. For all $i = 1, \ldots, k$, Bob checks if it holds that $MAC_{K_i^{Sim}}(N) = V_i \neq \perp$. If this is the case, he regards the message N as properly authenticated, otherwise not.

Discussion. In a nutshell, the security of the protocol follows from the fact that an adversary cannot determine the responses $R_{C_i^j}$ and the MAC-Keys $K_1, \ldots, K_l$ as quickly as Alice. As earlier, verification of a randomly chosen subset of all MACs can improve Bob's computational efficiency in step 5. Depending on the exact circumstances, a few erroneous V_i may be tolerated in step 5, too.

We assume without loss of generality that the MAC can be computed quickly (including the derivation of the MAC keys $K_1, \ldots, K_l$), i.e., within time t_{MAC}, and that t_{MAC} is small compared to t_{max}. Again, this condition could be realized by amplification through feedback loops if necessary (see Section 2.3). Furthermore, it is known that MACs can be implemented very efficiently [27]. If information-theoretically secure hash functions and MACs are used, the security of the protocol will not depend on any assumptions other than the security of the SIMPL system.

If the SIMPL system has a multi-bit output, then values of $k = 1$, i.e., sending just one challenge in each round, or of $l = 1$, i.e., employing just one round of communication, may suffice. Such a multi-bit output can arise either naturally, for example through the choice of the SIMPL system itself (as noted earlier, the optical SIMPL system presented in Section 4.3 has this property). Or it can be enforced by feedback loops, or by using several independent SIMPL systems in parallel (see Section 2.3, page 32). In fact, such measures even are strictly necessary to uphold the protocol's security if the constant c has got a very low value.

3.3 Application Scenarios

Secure Communication Infrastructures. Within the given space restrictions, we will now discuss the application of SIMPL systems to secure communication in networks, illustrating their potential in such a setting. Consider a situation where k parties $P_1, \ldots, P_k$ and a trusted authority TA participate in a communication network. Assume that each party P_i carries its own SIMPL S_i in its hardware, and that a certificate C_i has been issued for each party by the TA. The certificate includes the identity and the rights of Party P_i, and has the form

$$C_i = \big(Id_i, Rights_i, D(S_i), Sig_{TA}(Id_i, Rights_i, D(S_i))\big).$$

Under these provisions, the parties can mutually identify themselves by Protocol 2, they can establish authenticated channels with each other by Protocol 3, and they can exchange session keys via the Diffie-Hellman protocol [32] over these authenticated channels. The whole architecture works without permanent secret keys, or without any other secret information that is stored permanently in the hardware of the parties $P_1, \ldots, P_k$.

It also seems well applicable to cloud computing: All personal data could be stored centrally. Session keys could be exchanged by the Diffie-Hellman protocol over channels authenticated by the SIMPL systems. These keys can be used to download the personal data in encrypted form from the central storage. The keys can be new in each session, no permanent secret keys in the mobile hardware are be necessary.

The above approaches can further be combined with tamper-sensitive SIMPL systems. These SIMPLs may cover hardware which has a functionality $Func_i$ as long as it is non-manipulated. Each certificate C_i could then also include the functionality of the hardware, i.e., it could be of the form

$$C_i = \big(Id_i, Rights_i, Func_i, D(S_i), Sig_{TA}(Id_i, Rights_i, Func_i, D(S_i))\big).$$

By running the identification protocol (Prot. 2), party P_i can prove to party P_j that the SIMPL system S_i is non-tampered, and that the hardware hence has the claimed functionality $Func_i$. Please note that the optical SIMPL systems we propose in this paper is naturally tamper sensitive; the tamper sensitivity of such optical scattering structures has already been shown in detail in [8].

Two other Applications. Let us, in all brevity, point to two other applications of SIMPL systems. They are described in more detail in [16].

A first application is the generation of unforgeable labels for products or security tokens. SIMPL systems can create labels which do not contain any secret information, which can be verified offline, and which only require remote, digital communication between the label and a testing device. These properties are not met by other known labeling techniques: RFID-tags with secret keys obviously contain secret information; PUF-based labels contain secret information in the case of Weak PUFs, and require an online database in the case of Strong PUFs [8]; and current Certificates of Authenticity (COAs) [28] [30] require analog near-field measurements in the verification step.

Another application area of SIMPLs lies in the context of the digital rights management problem. SIMPLs can create unclonable representations of digital content [16]. Similar to the unforgeable labels, these unclonable representations of digitial content do not contain any secret information. They can be verified for their validity offline and by mere digital communication between a tester and the device carrying the unclonable representation. Again, in combination these features are not met by any comparable technique known to the author. In [29] [30] [31], for example, the random features of the data carrier must be determined in the near-field by analog measurements.

4 Implementation of SIMPL Systems

Let us now turn to the practical implementation of SIMPL systems. We will give an overview of existing ideas and challenges, and propose one new, optical concept.

4.1 Challenges

It turns out that there are some strong challenges in the realization of SIMPL systems. The three non-trivial requirements that need to be balanced are complexity, stability,

and simulatability: On the one hand, the output of a SIMPL system must be sufficiently complex to require a long computation/simulation time. On the other hand, it must be simple enough to allow simulation at all, and to enable the determination of $D(S)$ by measurement or numeric analysis techniques. A final requirement is that the simulation can be carried out *relatively* efficiently by everyone (this is necessary to complete the verification steps in the identification and message authentication protocols quickly); while, at the same time, even a very well equipped attacker, who can potentially attempt to parallelize the simulation on many powerful machines, cannot simulate as fast as the real-time behavior of the SIMPL system. In the sequel, we will discuss a few implementations that try to meet these seemingly conflicting requirements.

4.2 Electrical SIMPL Systems

Since the first publication of [16], a sequence of papers of our group has dealt with the implementation of SIMPL systems by electrical, integrated circuits [17] [18] [19] [20]. We tried to exploit two known speed bottlenecks of modern CPUs: Their problems in dealing simultaneously with very large amounts of data, and the complexity of simulating analog, parallel phenomena. Let us briefly summarize these approaches, quoting from said papers.

"Skew" SRAM Memories. A first suggestion made in [17] [18] [19] [20] is to employ large arrays of SRAM cells with a special architecture named "skew design". In this design, the read- and write behavior of the cells is dependent on the applied operational voltage. The simulation of a skew SRAM memory in a feedback loop of a very large number of successive read- and write events then seems somewhat laborious to simulate on a standard architecture. The hypothesis put forward in [17] [18] [19] [20] is that this creates a small, constant simulation overhead. Two essential assumptions in this concept are: (i) No parallelization is possible, since the successive read- and write events in the feedback loop are made dependent on the previous read results. And (ii), since no parallelization is possible, the limiting factor for an adversary is his clock frequency, which is quite strongly limited by current technology.

As described in the listed references, the idea shows strong promise to succeed against any adversaries with a limited financial budget, and in particular against any FPGA-based attacks. Future work will need to show how large the exact simulation margin is, and whether it is indeed sufficient to defeat an adversary with large resources, who is capable of fabricating ASICs. Due to its relatively easy realizability and good security level, the idea could have a strong potential for the consumer market.

Two-dimensional Analog Computing Arrays. A second suggestion of [17] [18] [19] [20] consists of using analog, two-dimensional computing arrays. The authors suggest the use of so-called cellular non-linear networks (CNNs) which are designed to imitate non-linear optical systems. Due to their analog and inherently parallel nature (many cells exchange information at the same time), it is suggested that CNNs are time consuming to simulate on a digital, sequential architecture.

This idea has its assets on the security side: Since it is based on manufacturing mismatches in CNN fabrication that currently seem unavoidable, it shows promise of defeating even attackers with very strong financial resources, and of being manufacturer

resistant in the sense of [23]. It requires the use of analog circuits, though, which might potentially be unsuited for low-cost applications.

Other Approaches. Independently, the work of other groups has lead to different structures that could be used as SIMPLs. The implementation of PPUFs presented in [21] could potentially be downscaled to become a SIMPL system, even though it would have to be carefully investigated how resilient such small-scale instances are against parallelization attacks. Another very interesting, FPGA-based candidate for SIMPLs is implicit in the work of [24].

4.3 Integrated Optical SIMPLs

Also optical structures can be used as SIMPL systems. The rationale behind employing optics is as follows: First, optical systems can potentially achieve faster component interaction than electronic systems; this promises to create the desired speed advantage over any electronic simulator. The phenomenon of optical interference has no electronic analog at room temperature [41], and can create a computational overhead for electronic simulators. Second, the material degradation of optical systems is low, and their temperature stability is known to be high [41] [42]. Even very complex and randomly structured optical systems, whose internal complexity creates the desired speed gaps, can produce outputs that are stable against aging and environmental conditions.

The concrete optical SIMPL we suggest is depicted schematically in Figure 1. It comprises of an immobile laser diode array with k phase-locked diodes $D_1, \ldots, D_k$ [43], which is used to excite a disordered, random scattering medium. The diodes can be switched on and off independently, leading to 2^k challenges C_i. These can be written as $C_i = (b_1, \ldots, b_k)$, where each $b_i \in \{0, 1\}$ indicates whether diode D_i is switched on or off. (Note that the diode array must indeed be phase locked in order to allow interference of the different diode signals.) At the right hand side of the system, an array of l light sensors $S_1, \ldots, S_l$, e.g. photodiodes, measures the resulting light intensities locally. A response R_{C_i} consist of the intensities $I_1, \ldots, I_l$ in the l sensors. Instead of phase-locked diode arrays, also a single laser source with a subsequently placed, inexpensive light modulator (as contained in any commercially available beamer) can be employed.

Under the provision that a *linear* scattering medium is used in such integrated optical SIMPLs, the following analysis holds[44]. Every diode D_i with $b_i = 1$ creates a

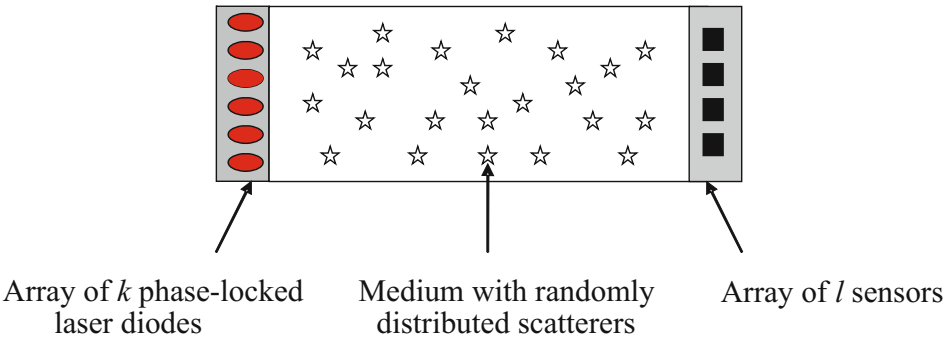

Fig. 1. An integrated optical SIMPL system

lightwave, which is scattered in the medium and arrives at the sensor S_j with amplitude E_{ij} and phase shift θ_{ij}. The intensity I_j at the sensor S_j is then given by [42]

$$I_j = \left|E_j\right|^2 = \left|\sum_i b_i \, E_{ij} \, \cos\, \theta_{ij}\right|^2. \tag{1}$$

For the said linear scattering medium, the amplitude E_{ij} and phase shift θ_{ij} are independent of whether the other diodes are switched on or off. One can hence collect many CRPs

$$(C_m, R_{C_m}) = ((b_1, \ldots, b_k), (I_1, \ldots, I_l)),$$

and derive the values E_{ij} and θ_{ij} from knowledge of these many (C_m, R_{C_m}). One suited approach are machine learning techniques, for example a standard machine learning regression.

Once the parameters E_{ij} and θ_{ij} are known, the simulation of a response $R_{C_m} = (I_1, \ldots, I_l)$ to a given challenge $C_m = (b_1, \ldots, b_k)$ can be executed by simple calculation following Eqn. 1. The time margin to the real system will be small, but likely detectable: The real system creates its output and the complex interference in nanoseconds, while the calculation of Eqn. 1 requires around $k \cdot l$ multiplications and $k \cdot l$ additions. Some of these computations can be parallelized, and the values $E_{ij} \cdot \cos\theta_{ij}$ can be precomputed. Still, even for a moderate size of the two-dimensional diode and sensor arrays of around $100 \times 100 = 10^4$ each, the number of additions is on the order of 10^8. This seems to create exactly the constant, notable time gap that we require in SIMPLs.

A first proof-of-concept for this integrated optical approach, which is not optimized in terms of speed, but shows the feasibility of the output simulation/prediction on the basis of real data, is given in the appendix.

4.4 Further Implementation Strategies

Let us discuss a few further implementation strategies for SIMPLs.

Employing PUFs with Reduced Complexity. One generic strategy for the realization of SIMPL systems, which has been suggested already in [16], is the following: Employ a PUF or a PUF-like structure; and reduce its inner complexity until it can be characterized by measurements and simulated, or until it can successfully be machine learned. If the level of complexity is still sufficient, then this simulation will be more time consuming than the real-time behavior of the system. In fact, the suggestions of the previous subsections used this strategy already, since both CNNs and integrated optical structures have already been suggested as PUFs in earlier work [36] [12]. But also any other PUFs could be used in this strategy, for example Pappu's original optical PUF with a reduced number of scatterers [8], as suggested in [16].

Simulation vs. Verification. Another interesting idea is to exploit the well-known asymmetry between actively computing a solution for a certain problem and verifying the correctness of a proposed solution (as also implicit in the infamous P vs. NP question) [16]. Exploiting this asymmetry could lead to protocols of the following kind: A SIMPL

system provides the verifier in an identification/authentication protocols with some extra information that allows the verifier to *verify* its answers fast. To illustrate our point, imagine an analog, two-dimensional, cellular computing array whose behavior is governed by partial differential equations (PDEs), such as the CNN described in section 4.2. Then, verifying the validity of a given final state of such a PDE-driven system (i.e. verifying that this state is indeed a solution of the PDEs driving the system) could be much more time efficient than computing this solution from scratch. Furthermore, the verifier could not only be given external outputs of such a two-dimensional array (e.g. values in boundary cells), but also internal submeasurements (e.g. values in inner cells) that help him to verify the output quickly.

The simulation vs. verification strategy can help to relieve the seeming conflict between the requirement for fast simulation on the side of the verifier (who may not be well equipped on the hardware side) and the necessary time margin to an attacker (who may be very well equipped on the hardware side), which we already addressed in Section 4.1.

5 Summary, Discussion, and Future Work

Summary. This paper introduced a security concept termed "SIMPL Systems". We started by a explaining the basic idea and by giving a semi-formal specification of SIMPL systems. We subsequently discussed some basic properties that follow from this specification. We then presented two protocols that can be realized by SIMPL systems, namely identification and message authentication. These protocols exploit the fact that the holder of a SIMPL system is the only person who can determine the response of the SIMPL to a randomly chosen challenge within a certain time frame. We argued that the can be used to set up special, secure communication infrastructures which obviate the long-term storage of any form of secret keys in hardware. We listed other applications of SIMPL systems, for example as unforgeable labels and in the digital rights management problem.

We next discussed the practical implementation of SIMPL systems. We gave an overview of existing, electrical candidates, and then suggested a new optical implementation based on light scattering. We gave a proof-of-concept for this optical SIMPL by using data from a first prototype, which had been set-up by our group in a different context [40]. This data shows the general feasibility of predicting such systems, but was not yet optimized in terms of speed. We also presented generic and/or future implementation strategies for SIMPLs, for example the use of PUFs with reduced complexity, or exploiting the asymmetry between actively computing and merely verifying a solution to a given problem (as implicit in the well-known P vs. NP question).

Discussion. Let us conclude this work by a detailed comparative analysis. As said earlier, there are some similarities between classical private/public key cryptoschemes and SIMPL systems: The numeric description $D(S)$ is some analog to a public key, while the physical system S itself constitutes some functional equivalent to a private key. This provides SIMPLs with some public-key like functionality and with the resulting practicality advantages.

At the same time, there is one important difference to classical public-key systems: This new type of "private key" S is no secret numeric information, but a randomly structured, hard-to-clone *physical system*. It has the interesting feature of not containing any form of secret information. Neither in an explicit digital form like a digital key in classical hardware. Nor in a hidden, analog form such as internal PUF parameters (for example the mentioned delay values in the Arbiter PUFs, or the parameters determining SRAM behavior). All internal characteristics of a SIMPL, including its precise internal configuration, can be publicly known without compromising the security of the derived cryptographic protocols.

The security of SIMPL systems is not free of assumptions, though. Instead of presupposing the secrecy of some sort of information, it rests on the following two hypotheses: (i) on the computational assumption that no other, well-controllable, configurable, or even programmable hardware can generate the complex responses of a SIMPL with the same speed, and (ii) on the physical assumption that it is practically infeasible for Eve to exactly clone or rebuild the SIMPL system, even though she knows its internal structure and properties.[1]

It is long accepted that computational assumptions play a standard role in classical cryptography, and they are also a part of the security assumptions for SIMPL systems; but SIMPLs show that one can trade the need for secret information in the hardware against assumptions on the physical unclonability of the SIMPL system. This can surprisingly obviate the familiar requirement that cryptographic hardware must contain secret key information of some sort.

Future Work and Prospects. Future work on SIMPLs will likely concentrate on new protocols beyond identification and message authentication, and on formal security proofs for such protocols. But perhaps the greater challenge lies on the hardware side: Even though there are several promising candidates (see Section 4), the issue of finding a highly secure, practical, and cheap implementation of SIMPL systems appears not to be fully settled yet. If such an implementation is found, or if the existing implementation candidates are shown to possess all necessary properties, this could change the way we exercise cryptography today.

References

1. http://www.cbsnews.com/stories/2010/02/15/business/main6209772.shtml
2. http://www.bbc.co.uk/news/10569081
3. http://www.eurosmart.com/images/doc/Eurosmart-in-the-press/2006/cardtechnologytoday_dec2006.pdf
4. http://www.gsaietsemiconductorforum.com/2010/delegate/documents/GASSELGSALondon20100518presented.pdf (Slide 23)

[1] The reader can verify the plausibility of the latter unclonability property by considering the optical implementation of section 4.3: Even if the positions of all scattering centers and the other irregularities in the plastic matrix were known in full detail, it would still be infeasible to rebuild the whole system with perfect precision.

5. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Manzuri Shalmani, M.T.: On the power of power analysis in the real world: A complete break of the KEELOQ code hopping scheme. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 203–220. Springer, Heidelberg (2008)
6. Kasper, T., Silbermann, M., Paar, C.: All you can eat or breaking a real-world contactless payment system. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 343–350. Springer, Heidelberg (2010)
7. Anderson, R.J.: Security Engineering: A Guide to Building Dependable Distributed Systems, 2nd edn. Wiley, Chichester (2008) ISBN: 978-0-470-06852-6
8. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical One-Way Functions. Science 297, 2026–2030 (2002)
9. Pappu, R.: Physical One-Way Functions, PhD Thesis, MIT
10. Gassend, B., Clarke, D.E., van Dijk, M., Devadas, S.: Silicon physical random functions. In: ACM Conference on Computer and Communications Security 2002, pp. 148–160 (2002)
11. Gassend, B., Lim, D., Clarke, D., Dijk, M.v., Devadas, S.: Identification and authentication of integrated circuits. Concurrency and Computation: Practice & Experience 16(11), 1077–1098 (2004)
12. Tuyls, P., Skoric, B.: Strong Authentication with Physical Unclonable Functions. In: Petkovic, M., Jonker, W. (eds.) Security, Privacy and Trust in Modern Data Management, Springer, Heidelberg (2007)
13. Edward Suh, G., Devadas, S.: Physical Unclonable Functions for Device Authentication and Secret Key Generation. In: DAC 2007, pp. 9–14 (2007)
14. Gassend, B., Dijk, M.v., Clarke, D.E., Torlak, E., Devadas, S., Tuyls, P.: Controlled physical random functions and applications. ACM Trans. Inf. Syst. Secur. 10(4) (2008)
15. Rührmair, U.: Oblivious Transfer based on Physical Unclonable Functions (Extended Abstract). In: Acquisti, A., Smith, S.W., Sadeghi, A.-R. (eds.) TRUST 2010. LNCS, vol. 6101, pp. 430–440. Springer, Heidelberg (2010)
16. Rührmair, U.: SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions. Cryptology ePrint Archive, Report 2009/255 (2009)
17. Rührmair, U., Chen, Q., Lugli, P., Schlichtmann, U., Stutzmann, M., Csaba, G.: Towards Electrical, Integrated Implementations of SIMPL Systems. Cryptology ePrint Archive, Report 2009/278 (2009)
18. Chen, Q., Csaba, G., Ju, X., Natarajan, S.B., Lugli, P., Stutzmann, M., Schlichtmann, U., Rührmair, U.: Analog Circuits for Physical Cryptography. In: 12th International Symposium on Integrated Circuits (ISIC 2009), Singapore, December 14-16 (2009)
19. Rührmair, U., Chen, Q., Stutzmann, M., Lugli, P., Schlichtmann, U., Csaba, G.: Towards electrical, integrated implementations of SIMPL systems. In: Samarati, P., Tunstall, M., Posegga, J., Markantonakis, K., Sauveron, D. (eds.) WISTP 2010. LNCS, vol. 6033, pp. 277–292. Springer, Heidelberg (2010)
20. Chen, Q., Csaba, G., Lugli, P., Schlichtmann, U., Stutzmann, M., Rührmair, U.: Circuit-based Approaches to SIMPL Systems. Accepted by Journal of Circuits, Systems and Computers (2010) (to appear)
21. Beckmann, N., Potkonjak, M.: Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 206–220. Springer, Heidelberg (2009)
22. Rührmair, U., Busch, H., Katzenbeisser, S.: Strong PUFs: Models, Constructions and Security Proofs. In: Sadeghi, A.-R., Naccache, D. (eds.) Towards Hardware Intrinsic Security: Foundation and Practice, Springer, Heidelberg (2010) (to appear)
23. Gassend, B.: Physical Random Functions, MSc Thesis, MIT (2003)

24. Majzoobi, M., Elnably, A., Koushanfar, F.: FPGA Time-Bounded Unclonable Authentication. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 1–16. Springer, Heidelberg (2010)
25. Guajardo, J., Kumar, S.S., Schrijen, G.-J., Tuyls, P.: FPGA Intrinsic PFs and Their Ue For IP Potection. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 63–80. Springer, Heidelberg (2007)
26. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling Attacks on Physical Unclonable Functions. In: 17th ACM Conference on Computer and Communications Security (2010); Previous versions available from Cryptology ePrint Archive, Report 251/2010
27. Halevi, S., Krawczyk, H.: MMH: Software message authentication in the gbit/Second rates. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 172–189. Springer, Heidelberg (1997)
28. DeJean, G., Kirovski, D.: RF-DNA: Radio-Frequency Certificates of Authenticity. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 346–363. Springer, Heidelberg (2007)
29. Kariakin, Y.: Authentication of Articles. Patent writing, WO/1997/024699 (1995), http://www.wipo.int/pctdb/en/wo.jsp?wo=1997024699
30. Vijaywargi, D., Lewis, D., Kirovski, D.: Optical DNA. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 222–229. Springer, Heidelberg (2009)
31. Hammouri, G., Dana, A., Sunar, B.: CDs Have Fingerprints Too. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 348–362. Springer, Heidelberg (2009)
32. Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Transactions on Information Theory IT-22, 644–654 (1976)
33. Yao, A.C.-C.: Classical physics and the Church-Turing Thesis. Journal of the ACM 50(1), 100–105 (2003)
34. Aaronson, S.: NP-complete Problems and Physical Reality. In: Electronic Colloquium on Computational Complexity (ECCC), 026 (2005)
35. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5), 1484–1509 (1997)
36. Csaba, G., Ju, X., Ma, Z., Chen, Q., Porod, W., Schmidhuber, J., Schlichtmann, U., Lugli, P., Rührmair, U.: Application of Mismatched Cellular Nonlinear Networks for Physical Cryptography. In: IEEE CNNA (2010)
37. Lim, D.: Extracting Secret Keys from Integrated Circuits. M.Sc. Thesis, MIT (2004)
38. Suh, G.E., O'Donnell, C.W., Sachdev, I., Devadas, S.: Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions. In: Proc. 32nd ISCA, New York (2005)
39. Yu, M.-D.(Mandel) Devadas, S.: Secure and Robust Error Correction for Physical Unclonable Functions. IEEE Design & Test of Computers 27(1), 48–65 (2010)
40. Rührmair, U., Knobling, R., Weiershäuser, A., Urban, S., Finley, J.: Secure Integrated Optical Physical Unclonable Functions (2010) (in preparation)
41. Lipson, S.G.: Optical Physics, 3rd edn. Cambridge University Press, Cambridge (1995) ISBN 0-5214-3631-1
42. Demtröder, W.: Experimentalphysik 2: Elektrizität und Optik. Springer, Heidelberg (2004) ISBN-10: 3540202102
43. Zhou, D., Mawst, L.J.: Two-dimensional phase-locked antiguided vertical-cavity surface-emitting laser arrays. Applied Physics Letters (2000)
44. Sölter, J.: Personal Communication (2010)

Chapter 8

Towards Electrical, Integrated Implementations of SIMPL Systems

The last Chapter 7, among other things, gave an abstract overview of several possible implementations of SIMPL systems. In this chapter, we dive deeper into the details, discussing two techniques in more depth: Firstly, SRAM-based SIMPLs that utilize a special "skew" design of the individual SRAM cells; secondly, SIMPLs that employ analog circuits, so-called cellular non-linear networks (CNNs).

The central question behind hardware implementations of SIMPLs is: How can they establish the speed or time difference between any emulation and an individual SIMPL system? The two above implementations find their own answers to this problem. They exploit two well-known bottlenecks of standard CPU architectures: SRAM SIMPLs make use of the limited clock frequencies by which current architectures can operate, and suggest a non-parallelizable function to be computed by the SIMPL hardware. CNN-based SIMPLs, on the other hand, utilize the fact that current CPUs cannot carry out analog computations on a massive amount of data in parallel.

While the question of practical and efficient SIMPLs or PPUFs [8] currently remains at least partly open, we believe that the approaches detailed in this chapter represent a promising step towards this goal, and provide inspiration for follow-up works.

The publication that is used in this chapter is:

- U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems*. Fourth Workshop in Information Security Theory and Practice (WISTP), 2010. Lecture Notes in Computer Science, Volume 6033, pp. 277-292, Springer, 2010.

The following preprint is related work, but is explicitly not used in this thesis:

- U. Rührmair, Q. Chen, P. Lugli, M. Stutzmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems*. Cryptology ePrint Archive, Report 2009/278, 2009.

Towards Electrical, Integrated Implementations of SIMPL Systems

Ulrich Rührmair[1], Qingqing Chen[2,3], Martin Stutzmann[4], Paolo Lugli[3], Ulf Schlichtmann[2], and György Csaba[3]

[1] Computer Science Department
[2] Institute for Electronic Design Automation
[3] Institute for Nanoelectronics
[4] Walter Schottky Institute
TU München, Germany
ruehrmai@in.tum.de, stutz@wsi.de,
{qingqing.chen, lugli, ulf.schlichtmann, csaba}@tum.de
http://www.pcp.in.tum.de

Abstract. This paper discusses strategies for the electrical, integrated implementation of a novel security tool termed *SIMPL system*, which was introduced in [1]. SIMPL systems are a public key version of Physical Unclonable Functions (PUFs). Like a PUF, each SIMPL system S is physically unique and non-reproducible, and implements an individual function F_S. In opposition to a PUF, every SIMPL system S possesses a publicly known numerical description $D(S)$, which allows its digital simulation and prediction. However, any such simulation must work at a detectably lower speed than the real-time behavior of S. As argued in [1], SIMPL systems have practicality and security advantages over PUFs, Certificates of Authenticity (COAs), Physically Obfuscated Keys (POKs), and also over standard mathematical cryptotechniques. This manuscript focuses on electrical, integrated realizations of SIMPL systems, and proposes two potential candidates: SIMPL systems derived from special SRAM-architectures (so-called "skew designs" of SRAM cells), and implementations based on analog computing arrays called Cellular Non-Linear Networks (CNNs).

Keywords: Physical Cryptography, Physical Unclonable Functions, SIMPL Systems, Public Key Systems.

1 Introduction

Physical Unclonable Functions (PUFs) are a relatively young, emerging cryptographic primitive [2] [3] [4] [5] [6] [7]. However, one potential downside of PUF-based protocols is that they usually require a previously shared piece of information (typically some challenge-response-pairs) that was established in a joint set-up phase between the communicants. Alternatively, an online connection to a trusted authority at the time of the protocol execution must be employed. In this particular structural aspect, PUFs are resemblant of classical private key systems.

In this paper, we are concerned with an alternative security tool called *SIMPL systems*, which is a public key version of standard PUFs. SIMPL systems have been introduced in [1]. The acronym SIMPL stands for "SIMulation Possible, but Laborious", and

P. Samarati et al. (Eds.): WISTP 2010, LNCS 6033, pp. 277–292, 2010.

hints at the critical security feature of these structures. A physical system S is called a *SIMPL system* if the following holds:

1. It is possible for everyone to numerically simulate and, thus, to predict the physical behaviour of S with very high accuracy. The basis of the simulation is an individual description $D(S)$ of S, and a generic simulation algorithm Sim, which are both publicly known.
2. Any sufficiently accurate numerical simulation — as well as any arbitrary physical emulation of S — is slower than the real-time behavior of S. Determining the system's behavior by an actual measurement on the original system S works detectably quicker than any other approach.
3. It is difficult to physically reproduce or clone S.

Put together in one sentence, the holder of a SIMPL system S can compute a publicly known, publicly computable individual function F_S faster than anyone else. Applying the familiar public key terminology to this situation, one could state that the numeric description $D(S)$ essentially serves as a public key, while the physical system S constitutes an equivalent to a private key. This "private key", however, is a physically irreproducible structure, which contains no secret information at all. This leads to several significant security advantages, which have been discussed in [1].

One critical question is certainly how SIMPL systems can be implemented in practice. We suggest two variants based on integrated electrical circuits in this publication: Firstly, special SRAM-memories based on a newly developed "skew" design, which leads to fuzzy memory cell behavior at quickly varied operational voltages. Secondly, we propose analog circuits known as Cellular Non-Linear Networks (CNN), whose cells evolve over time in an analog, highly parallel fashion. This can help them to outperform classical architectures on certain computational tasks, as is required from SIMPL systems.

Organization of the Paper. The rest of this manuscript is organized as follows: In section 2 we cite and discuss the formal specifications of SIMPL systems of [1]. Section 3 provides one example protocol and briefly discusses applications and advantages of SIMPL systems. In section 4 we treat the implementation of SIMPL systems by Cellular Non-Linear Networks. Section 5 introduces SIMPL systems based on special SRAM architectures. We conclude the paper in section 6.

2 SIMPL Systems

The following specification of SIMPL systems has been provided in [1].

Specification 1 (**(t_C, t_{Ph}, ϵ)-SIMPL Systems).** *Let S be a physical system mapping challenges C_i to responses R_i, with $\mathbf{C}$ denoting the finite set of all possible challenges. Let furthermore t_{max} be the maximum time (over all challenges $C_i \in \mathbf{C}$) which it takes until the system has generated the corresponding response R_i. S is called a* (t_C, t_{Ph}, ϵ)-SIMPL SYSTEM *if there is a numerical string $D(S)$, called the description of S, and a generic computer algorithm* Sim *such that the following conditions are met:*

1. *For all challenges $C_i \in \mathbf{C}$, the algorithm* Sim *on input*

$$\big(C_i,\, D(S)\big)$$

outputs R_i in feasible time.
2. *Any cryptographic adversary Eve will succeed in the following* **security experiment** *with a probability of at most ϵ:*
   (a) For a time period of length t_C, Eve is given the numerical description $D(S)$ and the code of the algorithm Sim.
   (b) Within this time period t_C, Eve is furthermore granted adaptive physical access to the system S at adaptively chosen time points. However, her overall access times must add up to a total of at most t_{Ph}.
   (c) After the time period t_C has expired, Eve can still access $D(S)$ and Sim*, but has no physical access to S any more. She is presented with a challenge C_{i_0} that was chosen uniformly at random from the set $\mathbf{C}$, and must output a value V_{Eve}.*

   We say that Eve succeeded in the above experiment if the following conditions are met:
   (i) $V_{Eve} = R_{i_0}$.
   (ii) The time that Eve needed to output V_{Eve} after she was presented with C_{i_0} is at most $2 \cdot t_{max}$.

   The said probability of ϵ is taken over the uniformly random choice of $C_{i_0} \in \mathbf{C}$, and the random choices or actions that Eve might take in steps 2a, 2b and 2c.

Some remarks on the specification are in order.

Security Model. Let us start by briefly discussing the security model of the specification. In practice, an adversary Eve can gather information about S in essentially two ways. Firstly *computationally*, by analyzing challenge-response-pairs (C_i, R_i) and by analyzing the algorithm Sim and the description $D(S)$. The CRPs may either stem from eavesdropping on protocols, or they may be computed by the adversary hinself via the algorithm Sim and the description $D(S)$. These possibilities are reflected in item *2a* of the specification. Secondly, Eve may *physically* measure arbitrary features of the system S at some point. For example, she might try to obtain some physical characteristics or internal parameters of the system which are not easily deducable from knowing many CRPs, but which could speed up her simulation. This possibility is covered in item *2b*. The model tries to reflect real-world situations, for example if S was used in mobile systems for identification purposes.

Immunity against Full Read-Out. It follows from Specification 1 that for any SIMPL system S, it must be impossible to measure the values R_i for *all* possible parameters $C_i \in \mathbf{C}$ within the timeframe t_{Ph}. Otherwise, Eve could create an exhaustive lookup-table for all possible values R_i during step 2b, which would enable her to succeed in the described experiment. Hence, for any SIMPL system either the set of possible measurement parameters $\mathbf{C}$ must be very large (for example exponential in some system parameter) and/or successive read-outs can only be carried out relatively slowly.

Immunity Against Cloning. Please note further that Specification 1 implies that previous physical access and a number of known Challenge-Response-Pairs of S must not enable Eve to do one of the following:

1. Build an *exact physical clone* S' of the system S, for which
$$R_i = R_i' \quad \text{for (almost) all } C_i \in \mathbf{C},$$
and for which the evaluation of the R_i' works comparably quickly as by an experiment on S.
2. Build a *functional physical clone* S' of S, which may be a physical system of a possibly very different structure or different lengthscales than S, that enables Eve to determine the values R_i for (almost) all $C_i \in \mathbf{C}$ correctly and comparably quickly as by experiment on S.
3. Build a *digital clone*, which is a computer algorithm Alg that numerically computes the values
$$Alg(C_i) \; = \; R_i$$
for (almost) all $C_i \in \mathbf{C}$ comparably quickly as by an experiment on S.

The inability for *digital cloning* implies a number of non-trivial requirements: Firstly, it logically includes the immunity against full read-out that we discussed earlier. Secondly, it implies that the behaviour of S cannot be learned by a machine learning algorithm that has a very rapid prediction phase, which works on a comparable timescales as the real-time behavior of S. Thirdly, and most generally, it implies that the simulation of S on the basis of $D(S)$ cannot be split into a possibly laborious precomputation phase independent of a concrete challenge, and a specific computation phase that very rapidly determines R_i once C_i is given.

In the sequel, we will sometimes refer to the immunity of S against cloning also as the unreproducibility or the uniqueness of S.

Feedback Loops and Security Margin. Specification 1 stipulates that the time gap between Eve and the real SIMPL system must be at least a factor of 2. This seems surprising: One might expect a polynomial vs. exponential distinction here. However, such asymptotic notions cannot be applied directly to the finite function which a SIMPL system implements without rising contradictions [9]. Furthermore, it is not clear whether a unique, non-reproducible hardware system with a truly exponential speed up exists at all: Quantum computers or quantum hardware are clonable, and other *practical* physical system with an exponential speed up over classical Turing machines currently are not known [19] [20] [21].

Nevertheless, in the application protocols which we suggest (identification and on-the-fly message authentication), a *detectable* time difference at the time of the protocol execution suffices. No security properties similar to the long-term confidentiality of encryption are required, that would make a polynomial vs. exponential time gap necessary.

Furthermore, the absolute (but not the relative!) time difference between the original system and Eve can be amplified via feedback loops. There, the SIMPL systems successively determines a sequence of challenge-responses-pairs $(C_{i_1}, R_{i_1}), (C_{i_2}, R_{i_2}), \ldots,$

(C_{i_k}, R_{i_k}), in which later challenges C_{i_m} are determined by earlier results R_{i_l}, with $m > l$. In this context, (C_{i_1}, R_{i_k}) can be regarded as the overall challenge-response pair determined by the structure, and the set $\mathbf{C}$ and t_{max} can be adjusted accordingly. Such feedback loops shift us into a region of absolute delay values (e.g. seconds) where we can maintain security even in the face of unwanted side effects, such as network and transmission delays.

Different Adversarial Scenarios. The specification leaves to some extent open which specific resources Eve may employ during her attack. There are several meaningful scenarios, leading to different security notions.

1. CONSUMER SECURITY: Eve is assumed to be a private person, possibly very educated in cryptographic and security matters, but with a budget not exceeding one million dollars.
2. TECHNOLOGICAL SECURITY: We assume that Eve is allowed to use basically unlimited financial resources, and faces no restrictions other than those induced by current technology.

When we say that a SIMPL system is secure in one of the above scenarios, we mean that it remains secure in the sense of Specification 1 if Eve is allowed the described resources. Which type of security we seek strongly depends on the intended application. A SIMPL system that is not technologically secure, but consumer secure might still find very fruitful applications in the consumer market. One should have this fact in mind, and not aim for technological security only when designing SIMPL systems.

3 Protocols and Applications

We will now quote one exemplary protocol that can be realized by SIMPL systems in order to illustrate their working principle [1]. A few applications and the advantages of SIMPL systems are briefly discussed, too.

3.1 Identification by SIMPL Systems

We assume that Alice, who holds an individual SIMPL system S, has put $D(S)$, Sim, t_{max} and a description of $\mathbf{C}$ in a public register. Now, she can prove her identity to an arbitrary second party Bob as follows [1]:

Protocol 2 (Identification of Entities by SIMPL Systems)

1. Bob obtains the information $D(S)$, Sim, t_{max}, and $\mathbf{C}$ associated with Alice from the public register.
2. Bob sends a number of randomly chosen challenges $C_1, \ldots, C_k \in \mathbf{C}$ to Alice.
3. Alice determines the corresponding responses $R_1, \ldots, R_k$ by experiment on her SIMPL system S, and returns them immediately to Bob.
4. Bob receives values $V_1, \ldots, V_k$, and measures Alice's response time (i.e. the time between the two events of sending $C_1, \ldots, C_k$ and receiving $V_1, \ldots, V_k$). If this time is above the threshold $2 \cdot t_{max}$, he aborts the protocol.

5. Bob checks through simulation by the algorithm Sim if for all $i = 1, \ldots, k$,

$$V_i \; = \; R_i.$$

If this is the case, Bob believes Alice's identity, otherwise not.

Security. As usual, k is the security parameter of the protocol. In a nutshell, the protocol works because Eve is unable to determine the values R_i for randomly chosen C_i comparably quickly as Alice, provided that: (i) The lifetime of the system S (and the period since $D(S)$ was made public) does not exceed t_C, and (ii) Eve's accumulated physical access times to S do not exceed t_{Ph}. In that case, Eve's probability to succeed in the protocol without possessing S are less or equal to ϵ^k.

Practicality. Bob can improve his computational efficiency by verifying the correctness of the responses R_i merely for a randomly chosen, smaller subset of $\{1, \ldots, k\}$. If necessary, possible network and transmission delays can be compensated for in advance by amplifying the absolute time gap between Eve and S through feedback loops (see discussion in section 2). Also the asymmetry between checking a solution and computing a solution may be exploited in future protocols (see section 6.3 of [1]).

3.2 Applications and Advantages of SIMPL Systems

Straightforward applications of the above identification protocol include [1]:

(i) Identification of hardware and computer systems.
(ii) Secure labeling of valuable items, such as branded products, pharmaceuticals, passports, bank notes, credit cards, and the like.
(iii) Unclonable (copy protected) representations of digital content and software, digital rights management.
(iv) Tamper sensitive hardware environments.

The upside of using SIMPL systems in these situations over standard mathematical cryptotechniques or alternative approaches such as Certificates of Authenticity [11] or PUFs has been discussed in detail in [1]. It includes: (i) SIMPL systems do neither contain nor constitute any sort of secret binary information. This makes them naturally immune against any side channel, invasive or malware attack. (ii) They allow protocols that are independent of the standard, unproven number theoretic assumptions (factoring, discrete log). (iii) They have strong practicality advantages over COAs and PUFs, due to their public key nature. (iv) They allow new DRM techniques, or unforgeable labels that can be read out digitally over long distances, and which can be verified offline at the same time [1].

These assets make them a worthwhile target for future investigations. In particular, it would be important to find electrical, integrated implementations — an issue which was left open in [1].

4 SIMPL Systems from Cellular Non-linear Networks

4.1 Introduction and General Idea

A first *electrical and on-chip* candidate for SIMPL systems are Cellular Non-Linear Networks (CNNs) [22]. If successfully implemented, they would result in a *technologically secure* SIMPL system (see page 281).

CNNs are analog computing arrays with a regular, periodic, cellular structure. The cells are characterized by a dynamical state variable, and their time evolution depends on their own internal state and on the inputs from their neighbouring cells. On an abstract level, their behavior is given and determined by so-called templates, which in the simpliest case are real-valued matrices. On a circuit level, it is given by the transistor architecture of a cell, which implements the behavior specified by the templates.

More specifically, each cell is characterized by a dynamical state variable x, which obeys the following, ordinary differential equation (ODE):

$$\dot{x}_{ij} = -x_{ij} + \sum_{k,l} \mathbf{A}_{i,j,k,l} y_{kl} + \sum_{k,l} \mathbf{B}_{i,j,k,l} u_{kl} + z_{ij}$$

i.e. the time derivative of the state variable (for the cell with i, j indices) depends on the y output of the neighboring cells (denoted by the k, l indices) via a the $\mathbf{A}$ cloning templates. Each cell has a bias (z) and inputs, which are coupled by the $\mathbf{B}$ template to the equation.

As a mathematical model, CNNs are very general; for example, cellular automata [13] can be interpreted as a special CNN which operates on discrete variables in discrete time (and where rules replace the ODE-based description). CNNs are also known to be Turing-complete [14]. CNNs often have multiple layers, and these layers are also coupled to each other via $\mathbf{B}$ templates.

Due to their analog and highly parallel architecture, CNNs have a remarkable computing power and efficiency. Already in 2004, a state-of-the-art programmable, commercially available CNN in a 0.35-μm standard CMOS technology exhibited peak computing figures of 330 GOPS [23] (or 3.6 GOPS/mm^2 and 82.5 GOPS/W in terms of area and power consumption). These numbers are yet excelled by non-programmable CNNs, which we propose for use as SIMPL systems. In specialized tasks, it is known that CNNs can outperform digital computers by a factor of up to 1,000 [24] [25]. CNNs are the largest analog circuits, with the CNN referred to above [23] containing 3.75 million transistors.

A further important property of CNNs is that their functionality is especially sensitive to the inevitable variations in the fabrication process, unless special countermeasures are taken. This can make the function F_S computed by a CNN S truly unique. At the same time, since CNNs are integrated electrical systems, dedicated on-chip measurement circuitry can determine the fabrication mismatches, and deliver a sufficiently detailed description $D(S)$ to simulate F_S. Such types of self-measuring cells are already today in standard use for calibration purposes [26]. Furthermore, it is known that there is a stable regime where the fabrication mismatches determine the CNN behavior, and where they override circuit noise and temperature variations [27] [28]. Altogether, said properties make CNNs quite interesting candidates for SIMPL systems.

4.2 Implementation

We propose two concrete candidates for CNN-based SIMPL systems. Firstly, CNNs employed for specialized tasks (see above), for example image processing tasks, where they are known to outperform classical architectures by factors of 10 – 1,000 [24] [25] [30].

Another attractive option, which we discuss in greater detail, is a template and circuit-design that has been recently devised in our group [29]. It is inspired by the high internal complexity of optical PUFs [2], in whose time evolution many internal scattering components interact in parallel, leading to a high computational complexity and to laborious simulatability.

Our template has the remarkable property that it effectively transfers optical behavior onto a CNN (i.e. onto an electrical integrated circuit), which then behaves quasi-optical, that is, similar to an optical system. In particular, the electrical current flowing through a certain reference point in each CNN-cell is equivalent to the local light intensity in an optical interference reference system.

The upcoming figures provide the templates and cell architecture of this 3-layer CNN, as well as simulation results that confirm the quasi-optical behavior. Figure 1 shows the templates and the interaction structure of the proposed 3-layer CNN. Figure 2 illustrates the circuit-level design. Figure 3 provides simulation data which shows the quasi-optical interference patterns in the linear (left) and non-linear/mismatched case (right). Figure 4 illustrates that local changes in the structure propagate globally. This further illustrates the quasi-optical nature and the high computational complexity of the structure: Its evolution involves many interacting subunits in parallel.

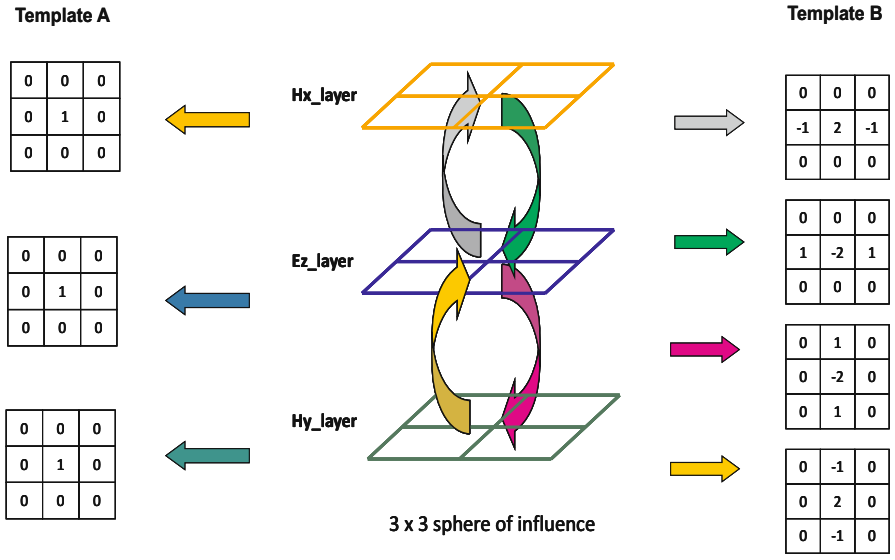

Fig. 1. Templates and interaction structure of our 3-layer CNN-SIMPL system

The described CNN-design seems particularly suited as SIMPL system because its quasi-optical behavior fosters pairwise interaction between the cells throughout the

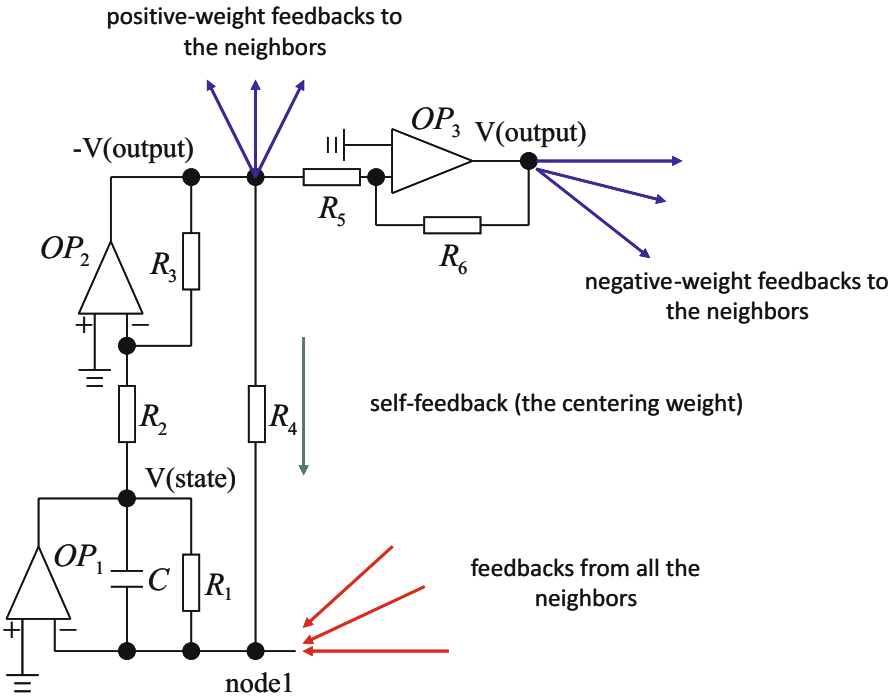

Fig. 2. Circuit level design of our proposed CNN-SIMPL system

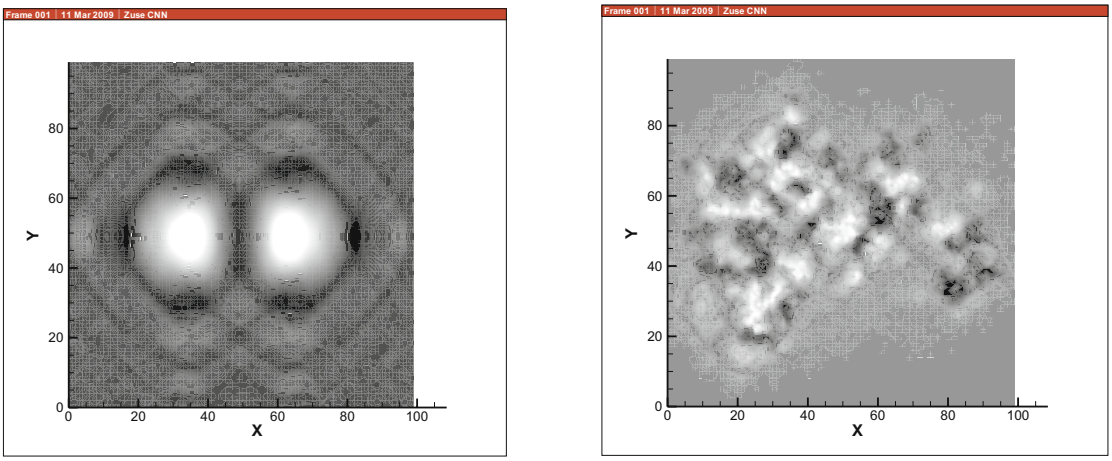

Fig. 3. Simulated behavior of the CNN-SIMPL system. The brightness levels illustrate the currents at a fixed reference point in each cell within a 100 × 100 cell structure. Left: Linear case, without fabrication mismatches, and with two excitation sources. Right: Non-linear case, resulting from fabrication mismatches, again two excitation sources. The left picture nicely shows the quasi-optical interference behavior. The non-linear case obviously provides a much more complex and richer regime, which is preferable for our purposes.

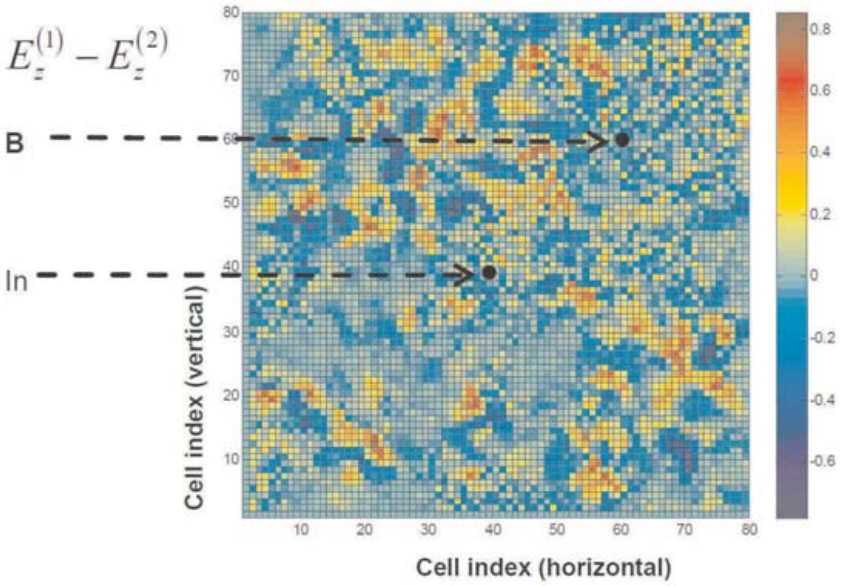

Fig. 4. A difference map that illustrates the global sensitivity of our CNN design to a local change in the structure. We changed only a single template at a particular position (denoted by B in the figure), which was even located far away from the input exciting the structure (marked as In). This altered the global behavior of the circuit detectably. The figure shows the difference of the values E_z^1 and E_z^2 obtained by two simulations, one for the original value of the templates, the other for one template value in position B altered.

structure. This leads to a particularly strong, inherent parallelism, which will be costly to simulate on digital architectures. Furthermore, as we could show in simulations, the behavior of the quasi-optical SIMPL automatically shifts into a non-linear, highly complex regime through the occurring manufacturing mismatches, which can be exploited even better for our purposes. In opposition to three-dimensional optical PUFs, its description $D(S)$ can be determined by in-built on chip measurement circuitry.

Another very important characteristics of our circuit that its behavior is sensitive, but not chaotic. Chaotic circuits are well known [15] and several CNN templates are known to realize chaos [16] [17] [18]. The time trajectories of a chaotic system are irreproducible in a real physical environment and are hence unsuited as a SIMPL system.

Security Aspects. A 100×100 cell CNN with our architecture leads to the following specific numbers: It requires a description $D(S)$ containing about $10^4 \cdot 19$ template values, which is about $100\ kB$ of information. In order to simulate the real-time evolution which the CNN undergoes in a few microsecond time frame, 10^4 coupled differential equations need to be solved (i.e., one for each cell). We estimate that this gives us a speed advantage of 10 – 100 to comparable digital computing machines. Please note also that CNNs are very small and energy efficient, allowing their integration into small devices, while classical architectures with comparable computing power will often be distinguishable already by their size on mobile devices such as smart cards or security tokens.

5 SIMPL Systems from Special SRAM Memories

5.1 Introduction and General Idea

One practical and stable, but only consumer secure SIMPL candidate will be presented in this section. It is based on a special design of SRAM memories, which we call *"skew design"*. Its basic idea is to design the SRAM-cells such that they exhibit varying behavior in different operational voltage regions. Some of the cells (cells of type 1) will function properly over the whole operational voltage range. Others, of type 2, will possess stable read operations, but exhibit (intended) write failures whenever the operational voltage VDD is below a certain threshold. This means that in these VDD regions, the content of the cell is not changed or affected by write procedures. Below the threshold, however, the write operation in cells of type 2 functions properly. Finally, there are cells of type 3, which contain a fixed bit value (0 or 1). It has been hardwired into them already in their fabrication, and their content cannot be changed by any write operation at all, regardless of the applied operational voltage.

Now, imagine an SRAM-memory M where cells of the described three types are randomly distributed or mixed. We call such a memory a "skew memory". Imagine further that on the basis of M, we build a larger hardware system S, which repeats the following feedback loop l times at maximal operational speed.

Feedback loop, iteration i:

1. Write bitvalues $b_1^i, \ldots, b_k^i$ into the addresses $WR_1^i, \ldots, WR_k^i$ of M.
2. Read out the bit values $B_1^i, \ldots, B_m^i$ from the addresses $READ_1^i, \ldots, READ_m^i$.
3. Switch to operational voltage $VDD(i)$.
4. Determine the parameters necessary for the next iteration, namely $b_1^{i+1}, \ldots, b_k^{i+1}$, $WR_1^{i+1}, \ldots, WR_k^{i+1}$, $READ_1^{i+1}, \ldots, READ_m^{i+1}$, $VDD(i+1)$, as a pseudo-random function of the values $B_1^i, \ldots, B_m^i$ obtained in step 2.

S is depicted schematically in Fig. 5. In order to associate a global input and a global output with S, we may say that the values $b_1^0, \ldots, b_k^0$, $WR_1^0, \ldots, WR_k^0$, $READ_1^0, \ldots, READ_m^0$, $VDD(0)$ that are necessary to start the loop, constitute its global input. After the last of the l iterations, the values $B_1^l, \ldots, B_m^l$ can serve as the global output of S. Alternatively, one may define the global output to be a function (e.g. a hash function) of the values $B_1^{l-q+1}, \ldots, B_m^{l-q+1}$, $B_1^{l-q+2}, \ldots, B_m^{l-q+2}, \ldots, B_1^l, \ldots, B_m^l$ that occured in the last q iterations of the loop. In this sense, we can interpret the behavior of S as a function F_S mapping global inputs to outputs.
Then, F_S has the following properties:

(i) F_S can be individualized by changing the design of the memory M. To that aim, for example memory cells of type 3 (fixed bitvalues) can be distributed randomly over the memory in a final fabrication step.
(ii) If the distribution of the cells of type 1, 2 and 3 is known, the function F_S can be simulated digitally.
(iii) The simulation of F_S on a standard architecture will be slower than the real-time computation of F_S by S. Also configurable hardware or ASICs that are not based

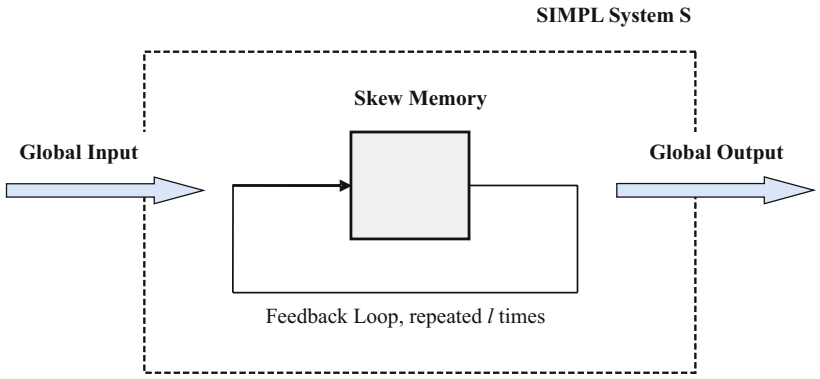

Fig. 5. Schematic illustration of the input–output behavior of S and of the function F_S

on a skew design will have a speed disadvantage. In both cases, the speed gap will only be a constant factor, however.

(iv) If the special skew design of SRAM cells is legally protected, then an adversary needs his own chip foundry to produce a hardware system that implements F_S comparably quickly, since ordering ASICs with a skew design will be legally prohibited.

The above properties qualify S as a consumer secure SIMPL system. We will discuss the practical implementation over the next section.

5.2 Implementation

A concrete skew design developed in our group [12] is illustrated in Fig. 1a), with width and length specified beside each transistor. The functionality of the design based on TSMC 0.18 μm technology has been succssfully verified with Spectre [31] simulations. The corresponding results are illustrated in Figure 7. In our case, $VDD_{min} = 1.4\,V$, $VDD_{max} = 1.7\,V$, and $VDD_{funcmin} = 1.58\,V$.

The memories, which will all share the same layout, can be individualized towards the end of manufacturing by fixing the content of some individually chosen cells to certain values. This means that the resulting structure will not be *manufacturer resistant* in

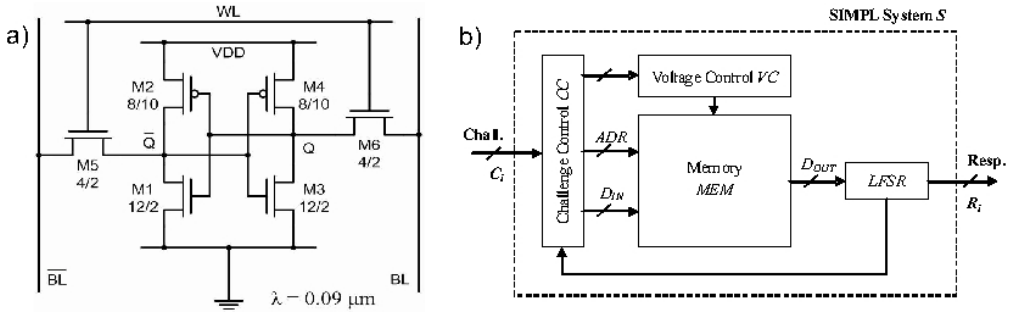

Fig. 6. (a) The SRAM cell layout. (b) The basic operation cycle of the SRAM-SIMPL system.

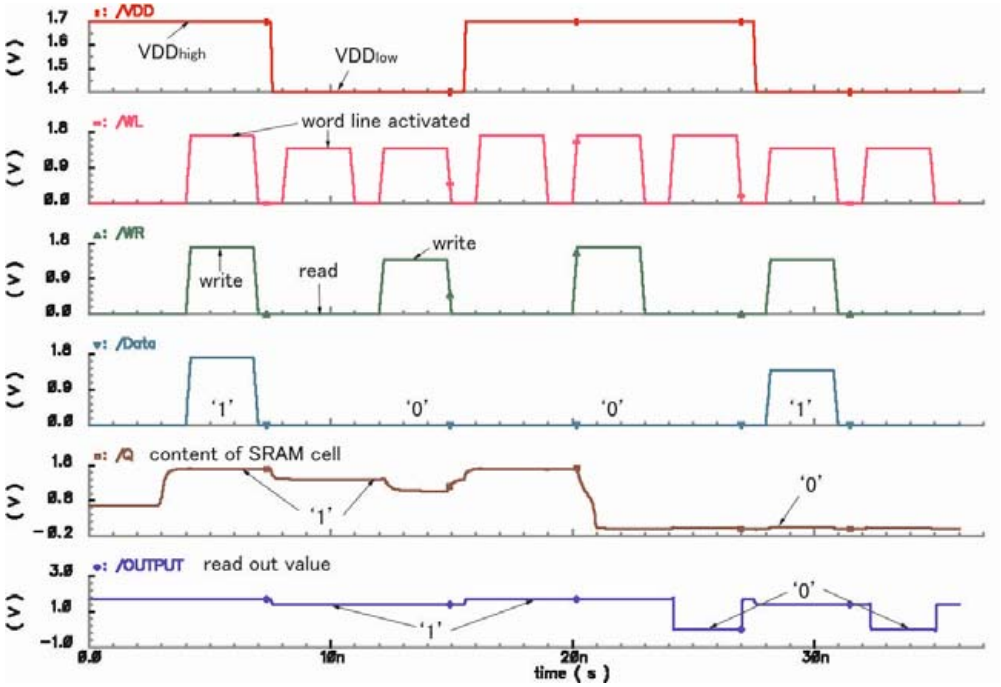

Fig. 7. Spectre Simulations confirm the desired behavior: Write failures occur at certain voltages, meaning that the content of the SRAM cell remains unchanged in the WRITE operation. At the same time, the READ operation functions properly at all voltages.

the sense of [4], but will at least require a fraudster to possess its own chip foundry. The common SRAM-cell arragement will be contained in the general simulation algorithm Sim, and the individual description $D(S)$ consists of the cells that have been fixed to certain values. Please note that the described individualization can be carried out on the basis of a pseudorandom number sequence, which means that a *short, few-hundred bit long random seed* s suffices as $D(S)$.

The basic implementation of the feedback-loop is sketched in Fig. 1b). The implementation of the pseudo-random generator is carried out by an LSFR, since a LSFR works very quickly. Computationally more laborious PRNGs could perhaps be implemented more quickly by a fraudster in his hardware. He would thereby regain some of his speed disadvantage. Please note that we do not require a PRNG with cryptographic security in this application, but merely a PRNG with a long periodicity, such that as many memory cells as possible are at least once written to or read from in the feedback loop.

The *relative* speed advantage of the real system can be further amplified by activating and writing into multiple word lines during one write cycle. Due to the skew design, the same value written in several lines will not necessarily result in the same cell content. Based on the simulation data we obtained, we estimate that the relative speed advantage of a SIMPL SRAM memory will be a factor on the order of 10, even compared to dedicated, configurable hardware such as FPGAs. At the same time, since all operations on the SRAM-memory are fully digital and well-defined, the content of the memory can be precisely simulated and predicted.

Compared to optical SIMPLs [1] and CNN-SIMPLs, the great advantage of the SRAM-variant is its practicality and stability. It can be implemented relatively cheaply, integrated in existing systems, and requires only very short descriptions $D(S)$. This comes at the cost of losing their technological security, and exchanging it against consumer security (see page 281). Nevertheless, this seems acceptable in many applications.

Security Aspects. Let us discuss a few security relevant aspects. A fraudster who wants to imitate the skew SIMPL systems without a skew architecture has a number of basic possibilities.

First of all, he may try to implement the feedback loop in full logic, that is, without any memory cells at all. His hope may be that pure logic operations work faster than memory read and write steps, and that he can so outperform (or at least match) the speed of the original SIMPL. However, if the memory is sufficiently large, then the construction of such a pure logic will be prohibited by size and complexity constraints.

This means that the faker needs to employ some sort of memory in his attempts. SRAM memories are, in general, the fastest currently available technology, meaning that the faker should use SRAM cells, too. If he cannot rely on skew cells, however, he cannot obtain the result of the WRITE operation in a skew cell (which is a function of the WRITE value, the actual operational voltage and the type of the cell) within one WRITE step.

The faker rather needs to compute the resulting value "by hand" before he writes it into a classical cell. To that end, he needs to look up the type of the cell before writing the value. That costs him one extra read operation before he executes the write procedure. Furthermore, computing the resulting write value "by hand" also costs time.

Overall, a faker without a skew memory requires one read operation, some computation and one write operation in order to emulate what happens within one write step of the skew memory. This provides a speed advantage of a factor around 2, as desired.

As said earlier, our group currently investigates designs where the SIMPL memory allows to write the same bit block into more than one word line simultaneously. The values that arrive in the multiple lines eventually differ due to the individual skew design of the cells. This could rise the speed advantage to a constant factor on the order of 10.

6 Conclusions

SIMPL Systems are a novel security concept, which can be regarded as a public key version of Physical Unclonable Functions [1]. Structurally, they function like a private/public key cryptosystem, with the notable difference that the equivalent to the private key is a physically hard-to-reproduce structure, which does not contain any secret information at all. This leads to critical security and practicality advances. In this paper, we reviewed the basic concepts presented in [1], but mainly focused on promising IC-based implementations of SIMPL systems.

Our first idea was to employ large, analog computing arrays as SIMPL systems. They evolve in parallel and by exchanging analog signals between their subunits, creating a significant computational power and complexity. At the same time, the arrays can be

designed to strongly depend on fabrication mismatches, making the function which they implement individual and unique. We suggested to use cellular, non-linear networks with special templates, since they are the largest currently known analog circuits with up to millions of transistors. We proposed one concrete design on the template and circuit level, and evaluated its functionality in several simulations. One important asset of CNN-based SIMPL systems was that they can eventually lead to technologically secure SIMPL systems.

Our second idea was to use special ASICs as SIMPL systems, whose circuit design implements one specific digital function more efficiently than a standard architecture. We suggested special SRAM designs, where the dimensions of the SRAM-cells are varied in such a fashion that their functionality depends on the applied operational voltage. This creates a small, constant computational overhead in the simulation of the cells, especially in the case where many subsequent read and write operations are applied at maximal speed and at quickly varied operational voltages in a feedback loop. The feedback loop also allows us to extend the relative, small computational overhead to larger absolute (but not relative!) time margins.

Future work will focus on implementing these structures in silicon, and on the analysis of their concrete time margins over cryptographic adversaries.

Acknowledgements

This work was conducted in the course of the Physical Cryptography Project at the TU München. Support by the Institute for Advanced Study and the International Graduate School of Science and Engineering of the TU München is gratefully acknowlegded. We thank Peter Vogl, Tamas Roska, and Wolfgang Porod for helpful discussions.

References

1. Rührmair, U.: SIMPL Systems: On a Public-Key Variant of Physical Unclonable Functions. Available from IACR Preprint Archive. Report 2009/255, `http://eprint.iacr.org`
2. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical One-Way Functions. Science 297, 2026–2030 (2002)
3. Pappu, R.: Physical One-Way Functions, PhD Thesis, MIT
4. Gassend, B.: Physical Random Functions, MSc Thesis, MIT (2003)
5. Tuyls, P., Schrijen, G.-J., Škorić, B., van Geloven, J., Verhaegh, N., Wolters, R.: Read-Proof Hardware from Protective Coatings. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 369–383. Springer, Heidelberg (2006)
6. Edward Suh, G., Devadas, S.: Physical Unclonable Functions for Device Authentication and Secret Key Generation. In: DAC 2007, pp. 9–14 (2007)
7. Tuyls, P., Skoric, B.: Strong Authentication with PUFs. In: Petkovic, M., Jonker, W. (eds.) Security, Privacy and Trust in Modern Data Management. Springer, Heidelberg (2007)
8. Tuyls, P., Skoric, B., Kevenaar, T. (eds.): Security with Noisy Data. Springer, Heidelberg (2007)
9. Rührmair, U., Sölter, J., Sehnke, F.: On the Foundations of Physical Unclonable Functions (2009) (submitted), `http://eprint.iacr.org/`
10. Feynman, R.P.: Simulating Physics with Computers. International Journal of Theoretical Physics 21(6&7), 467–488 (1982)

11. DeJean, G., Kirovski, D.: RF-DNA: Radio-Frequency Certificates of Authenticity. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 346–363. Springer, Heidelberg (2007)
12. Srinivas, B.N.: SRAM for use in Physical Cryptography. MSc Thesis, Department for Electrical Engineering and Information Technology, TU München (2009)
13. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601–644 (1983)
14. Roska, T., Chua, L.O.: The CNN universal machine: An analogic array computer. Circuits and Systems II: IEEE Transactions on Analog and Digital Signal Processing 40(3), 163–173 (1993)
15. Kennedy, M.P.: Three steps to chaos. II: A Chua's circuit primer. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 40(10), 657–674 (1993)
16. Zou, F., Nossek, J.A.: A chaotic attractor with cellular neural networks. IEEE Transaction on Circuits and Systems 38, 811–812 (1991)
17. Ogorzalek, M.J., Galias, Z., Dqbrowski, A.M., Dqbrowski, W.R.: Chaotic Waves and Spatio-Temporal Patterns in Large Arrays of Doubly-Coupled Chua' s Circuits. IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 42(10) (October 1995)
18. Gomez-Gesteira, M., de Castro, M., Perez-Villar, V., Chua, L.O.: Experimental Chua's Circuit Arrays As an Autowave Simulator. IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 46(4) (April 1999)
19. Yao, A.C.-C.: Classical physics and the Church-Turing Thesis. Journal of the ACM 50(1), 100–105 (2003)
20. Scott Aaronson: NP-complete Problems and Physical Reality. Electronic Colloquium on Computational Complexity (ECCC), 026 (2005)
21. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5), 1484–1509 (1997)
22. Chua, L.O., Roska, T.: Cellular Neural Networks and Visual Computing: Foundations and Applications. Cambridge University Press, Cambridge (2005)
23. Rodriguez-Vazquez, A., Linan-Cembrano, G., Carranza, L., Roca-Moreno, E., Carmona-Galan, R., Jimenez-Garrido, F., Dominguez-Castro, R., Meana, S.: ACE16k: The Third Generation of Mixed Signal SIMD-CNN ACE Chips Toward VSoCs. IEEE Trans. on Circuits and Systems – I 51(5), 851–863 (2004)
24. Chua, L.O., Roska, T., Kozek, T., Zarandy, A.: CNN Universal Chips crank up the computing power. IEEE Circuits and Devices Magazine 12(4), 18–28 (1996)
25. Cellular Wave Computers for Nano-Tera-Scale Technology – beyond spatial-temporal logic in million processor devices. Electronics Letters 43(8) (April 12, 2007)
26. Roska, T.: Private communication
27. Xavier de Souza, S., Yalcin, M., Suykens, J., Vandewalle, J.: Toward CNN Chip-Specific Robustness. IEEE Trans. on Circuits and Systems – I 51(5), 892–902 (2004)
28. Hillier, D., Xavier de Souza, S., Suykens, J., Vandewalle, J.: CNNOPT Learning CNN Dynamics and Chip-Specific Robustness. In: International Workshop on Cellular Neural Networks and Their Applications (2006)
29. Csaba, G., Ju, X., Chen, Q., Porod, W., Schmidhuber, J., Lugli, P., Rührmair, U.: On-Chip Electric Waves: An Analog Circuit Approach to Physical Uncloneable Functions. Report No. 2009/246 (2009), `http://eprint.iacr.org/`
30. Roska, T.: Cellular Wave Computers for Brain-Like Spatial-Temporal Sensory Computing. IEEE Circuits and Systems Magazine 5(2), 5–19 (2005)
31. Virtuoso Spectre Circuit Simulator, Cadence Design Systems, `http://www.cadence.com`

Chapter 9

SIMPL Systems as a Keyless Cryptographic and Security Primitive

In this chapter, we try to explore the limits of the cryptographic applicability of SIMPL systems. We do not deal with concrete implementations any more, since they were detailed in the last chapters. Rather, we assume that such realizations already have been found, and subsequently ask: What could they be used for, and what is the realistic potential of SIMPLs in fundamental cryptographic protocols?

Along these lines, we first deal with SIMPL protocols for identification and message authentication as in previous chapters. Subsequently, we go one step beyond. We suggest and investigate for the first time SIMPL-based coin tossing, bit commitment, and zero-knowledge protocols. We also evaluate the potential of SIMPLs and PPUFs [8] for cryptographic key exchange, reaching a relatively negative conclusion for the practical prospects of such protocols. The conclusion is partly in opposition to other recent works on this topic, which suggest key exchange as an important application of PPUFs [8].

This chapter consists of the following publication:

- U. Rührmair: *SIMPL Systems as a Keyless Cryptographic and Security Primitive*. Cryptography and Security 2012. Lecture Notes in Computer Science, Volume 6805, pp. 329-354, Springer, 2012.

The chapter concludes this thesis.

SIMPL Systems as a Keyless Cryptographic and Security Primitive

Ulrich Rührmair

Department for Electrical Engineering and Information Technology, TU München
Fraunhofer Institute for Secure Information Technology
Munich, Germany
ruehrmair@in.tum.de
http://www.pcp.in.tum.de

Abstract. We discuss a recent cryptographic primitive termed *SIMPL system*, where the acronym stands for *SIM*ulation *P*ossible, but *L*aborious. Like Physical Unclonable Functions (PUFs), SIMPL systems are disordered, unclonable physical systems with many possible inputs and a complex input-output behavior. Contrary to PUFs, however, each SIMPL system comes with a publicly known, individual numeric description that allows its slow simulation and output prediction. While everyone can determine a SIMPL system's output slowly by simulation, only its actual holder can determine the output fast by physical measurement. This added functionality allows new public key like protocols and applications.

But SIMPLs have a second, perhaps more striking advantage: No secret information is, or needs to be, contained in SIMPL systems in order to enable cryptographic security. Neither in the form of a standard digital key, nor as secret information hidden in the random, analog features of some hardware, as it is the case for PUFs. The security of SIMPL systems instead rests on (i) an assumption regarding their physical unclonability, and (ii) a computational assumption on the complexity of simulating their output. This provides SIMPL systems with a natural immunity against any key extraction attacks, including malware, side channel, invasive, and modeling attempts.

In this manuscript, we give a comprehensive discussion of SIMPLs as a cryptographic and security primitive. Special emphasis is placed on the different cryptographic protocols that are enabled by this new tool.

Keywords: SIMPL Systems, Public Key Cryptography, Physical Unclonable Functions, Hardware Security.

1 Introduction

Background and Motivation. Electronic communication and security devices are pervasive in our life. Just to name two examples, around five billion mobile phones are currently in use worldwide [1,2], and the world market of smart cards has an estimated volume of over three billion pieces per year [3,4]. Their widespread use makes such devices both a well-accessible and a worthwhile target for adversaries. Many security attacks thereby are not targeted against the employed cryptographic primitives themselves, some of which have proven attack-resilient over surprisingly long time spans.

D. Naccache (Ed.): Quisquater Festschrift, LNCS 6805, pp. 329–354, 2012.

Instead, they try to extract the employed secret keys by physical or software methods. Such key-extracting strategies are not just a theoretical concern, but have been demonstrated several times in widespread, commercial systems [5,6,7]. This drives the quest for new mechanisms that protect — or better still: avoid! — the presence of secret keys in vulnerable hardware system.

Physical Unclonable Functions (PUFs). The security primitive of a Physical Unclonable Function (PUF) [8,9,10,11] was introduced, at least in part, in order to address some of the above problems. A PUF is a (partly) disordered physical system S that can be challenged with so-called external stimuli or challenges C_i, upon which it reacts with corresponding responses termed R_{C_i}. Contrary to standard digital systems, a PUF's responses shall depend on the nanoscale structural disorder present in the PUF. This disorder cannot be cloned or reproduced exactly, not even by its original manufacturer, and is unique to each PUF. Assuming the stability of the PUF's responses, any PUF S hence implements an individual function F_S that maps challenges C_i to responses R_{C_i}. Due to its complex and disordered structure, a PUF can avoid some of the shortcomings associated with digital keys. For example, it is usually harder to read out, predict, or derive its responses than to obtain the values of digital keys that are stored in non-volatile memory. This fact has been exploited for various PUF-based security protocols [8,9,15,28].

One prominent example are PUF-based identification schemes [8,9,10]. They are usually run between a central authority (CA) and a hardware carrying a (unique) PUF S. One assumes that the CA had earlier access to S, and could establish a large, secret list of challenge-response-pairs (CRPs) of S. Whenever the hardware wants to identify itself to the CA at some later point in time, the CA selects some CRPs at random from this list, and sends the challenges contained in these CRPs to the hardware. The hardware applies these challenges to S, and sends the obtained responses to the CA. If these responses match the pre-recorded responses in the CRP-list, the CA believes the identity of the hardware.

Private Key like Functionality of PUFs. The described protocol has several well-known advantages [8,9]. However, one potential downside is that it presumes a previously shared piece of secret numerical information (i.e., the CRP-list). This information needs to be established in a secure set-up phase between the CA and the hardware, and must constantly be kept secret. Furthermore, the CRP-list uses up over time, since no single CRP should be used more than once in the identification process, and hence must be large. In these aspects, PUFs are resemblant of classical private key systems.

Secret Information in PUFs. Another noteworthy point is that PUFs in general do not obviate the presence of secret information within cryptographic hardware. The secret information is no longer stored in digital form in two-level systems, such as digital secret keys stored in non-volatile memory cells. But there is still some sort of secret information present in most PUFs, whose disclosure breaks the security of the system. Let us name two examples: In the case of SRAM PUFs the information that needs to be kept secret is the state of the SRAM cells after power up, or the tiny manufacturing variations of the SRAM cells that determine their state after power up [30]. Once this information is known to an adversary, he can numerically derive the same key as the

cryptographic hardware embedding the SRAM PUF, and break the system. In the case of Arbiter PUFs, the secret information are the internal runtime delays in the circuit stages [11]. If this information is known, the adversary can numerically simulate the behavior of the PUF output by an additive, linear model, again breaking its security [31].

In other words, the architectures of most current PUFs "hide" or "obfuscate" secret, security-relevant information very well in analog characteristics of integrated circuits. But at the same time, they do not avoid the need for secret information in hardware systems in principle; they just store it in a different form.

Our Contributions. Our contribution in this paper is a discussion and comprehensive overview of SIMPL systems as a new security primitive. We present the first formal specification of SIMPL systems, and show that they can implement a multitude of communication protocols, including identification, message authentication, coin flipping, bit commitment, and zero-knowledge proofs. We analyze scenarios in which these protocols can be applied, including secure communication in networks, item tagging and digital rights management. Furthermore, we survey existing hardware implementation candidates. Some emphasis is placed on the broad cryptographic potential of SIMPLs, and on their ability to construct security hardware without secret key information.

Related Work. The current paper is an extended version of [16] and [20]. Since [16], several follow-up papers of our group have focused on the implementation of SIMPLs by electrical circuits [17,18,19,21] and optical structures [20]. We emphasize that around the same time as [16], a comparable concept has been described completely independently in [24] under the name of a Public PUF (PPUF), and has been applied for key exchange purposes. It builds on a ideas and hardware architectures discussed already in [25]. Another closely related, but later idea is the concept of time-bounded authentication (TBA) [26], which has been suggested for identification schemes on FPGAs.

Organization of this Paper. The rest of this manuscript is organized as follows: In Section 2, we give a semi-formal specification of SIMPL systems, and discuss their properties. Sections 3 to 5 discuss protocols that can be realized on the basis of SIMPL systems and PPUFs, starting with identification and message authentication (Sec. 3), two-player protocols (Sec. 4), and key exchange (Sec. 5). Section 6 treats applications of SIMPL systems, and Section 7 surveys the existing implementation candidates. We conclude the paper in Section 8.

2 Specification and Properties of SIMPL Systems

2.1 Informal Description

We start this section by an informal description of the notion of a SIMPL system[1]. A physical system S is called a *SIMPL system* (or just a *SIMPL*) if the following holds:

[1] As mentioned in the abstract, the acronym SIMPL stands for SIMulation Possible, but Laborious.

1) S is a partly disordered physical system. It can be stimulated with challenges C_i, upon which it reacts with corresponding responses R_{C_i}. The responses are a function of the specific disorder present in S and of the applied challenge C_i.
2) The responses are assumed to be sufficiently stable to regard the behavior of S as a function F_S that maps challenges C_i to responses R_{C_i}. The pairs of the form (C_i, R_{C_i}) are often called the challenge-response pairs or CRPs of S.
3) It is possible (at least for the original manufacturer of S) to derive an individual numeric description $D(S)$ of S and an algorithm Sim. By use of $D(S)$ and Sim, everyone can simulate the correct responses R_{C_i} of S to any challenges C_i, or can at least verify a purported response R_{C_i} to a challenge C_i for correctness.
4) Any numeric simulation and any physical emulation that can predict the responses of S is noticeably slower than the real-time behavior of S. This must hold for simulation via Sim and $D(S)$, but must also apply to any adversarial algorithms and physical emulators. It must be upheld if the adversary has knowledge of $D(S)$, Sim, of all internal characteristics and disorder of S, and had earlier access to S.
5) It is difficult to physically clone S, i.e., to produce a "copy" S' which generates the same responses as S with comparable speed. Again, this must hold even for an adversary who knows $D(S)$, Sim, the internal characteristics and disorder of S, and who had earlier access to S.

Under these circumstances, a SIMPL system S computes the publicly known, publicly computable function F_S *faster* than anything or anyone else. In particular, the holder of S can determine the function value $F_S(C_i) = R_{C_i}$ for a randomly chosen challenge C_i faster than any adversary. This feature lies at the heart of all SIMPL-based security protocols.

Interestingly, the concept of a SIMPL is related to some well-known work of Feynman, who investigated the Turing-simulatability of physical systems in [32]. He conjectured that (i) all physical systems can, in principle, be simulated by Turing machines, but that (ii) such simulation cannot always be carried out in real time and will create a computational overhead [32]. SIMPL systems can be seen as a special application of these ideas in cryptography and security, combining them with the recent concept of physical unclonability.

2.2 Semi-formal Security Specification

The above properties can be coined into a semi-formal security specification of SIMPL systems. Its style follows the specifications presented in [27,28]. The specification describes the security of SIMPL systems as a "game" with the adversary, thereby introducing a relatively precise, parametric adversarial model.

Specification 1 ($(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL SYSTEMS.). *Let S be a physical system mapping challenges C_i to responses R_{C_i}, with $\mathbf{C}$ denoting the finite set of all possible challenges. Let $c > 1$ be a constant, and let furthermore t_{max} be the maximum time (over all challenges $C_i \in \mathbf{C}$) which it takes until the system S has generated the response R_{C_i} to the challenge C_i.*

S is called a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL SYSTEM *if there is a string $D(S)$, called the description of S, and a computer algorithm* Sim *such that the following conditions are met:*

1. *For all challenges $C_i \in \mathbf{C}$, the algorithm* Sim *on input $\big(C_i,\ D(S)\big)$ outputs R_{C_i} in feasible time.*
2. *For all binary strings X of length q, any cryptographic adversaries Eve will* SUCCEED *in the following* **security experiment** *with a probability of at most ϵ:*
   (a) Eve is given the string X, the numerical description $D(S)$ and the code of the algorithm Sim *for a time period of length t_C.*
   (b) Within the above time period t_C, Eve is furthermore given physical access to the system S at adaptively chosen time points, and for time periods of adaptively chosen lengths. The only restriction is that her access times must add up to a total of at most t_{Ph}.
   (c) After the time period t_C has expired, Eve is presented with a challenge C^* that was chosen uniformly at random from the set $\mathbf{C}$, and is asked to output a value V_{Eve}.

We thereby say that Eve SUCCEEDS *in the described experiment if the following conditions are met:*

(i) $V_{Eve} = R_{C^*}$.
(ii) The time that Eve needed to output V_{Eve} after she was presented with C^* was at most $c \cdot t_{max}$.

Said probability of ϵ is taken over the uniformly random choice of $C^* \in \mathbf{C}$, and the random choices or actions that Eve might take in steps 2a, 2b and 2c.

The Value of a Semi-Formal Specification. It is clear that Specification 1 is no consistent formal definition. Too many central aspects remain undefined from a strictly formal perspective (and the author is well aware of this). For example, it is not specified exactly how the adversary is formalized: Is he a classic probabilistic Turing machine (TM)? He should not be a classical TM, since he must be able to conduct physical actions on the SIMPL system while he has access to it. After all, a classical TM cannot execute such physical actions.

But how else could the adversary be formalized? Currently, there is no existing formal model that could capture all possible physical actions he might perform. In lack of such a model, a formal, consistent definition seems impossible.

Does that mean that we have to confine ourselves with the informal description of Section 2.1? This would be quite disadvantageous, since the description does not seem specific enough to capture the essence of SIMPL systems. The exact adversarial attack model is unclear, and there is no thorough specification what the "security" of a SIMPL system means. For example, it is not stated in which sense it shall be infeasible for an adversary to determine the responses of the SIMPL system as quickly as the original system.

The route that we propose in the above Specification 1 is, to some extent, a compromise. We intentionally leave some of the aspects of the definition imprecise; one example is the absence of an exact computational model that underlies the adversary's actions. Nevertheless, we believe that the specification helps to illustrate the exact nature of SIMPL systems more exactly, and allows us to specify a number of security parameters that are central to a SIMPL system's security.

Among other things, the specification can hence help to develop a common language and a communication interface between the developers of SIMPL-based protocols, and the hardware designers of the SIMPL systems themselves. A thorough and well-defined communication between these two groups is essential to securely apply SIMPLs in practice.

2.3 Properties of SIMPL Systems

Let us now discuss several features of SIMPL systems that follow from Specification 1.

Immunity against ϵ-fraction Read-out and Simulation. It must be practically impossible to measure the response values R_{C_i} of a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system for more than an ϵ-fraction of all possible challenges $C_i \in \mathbf{C}$ within time t_{Ph}. Otherwise, Eve could create a lookup table for an ϵ-fraction of all possible values R_{C_i} during step 2b. This would enable her to succeed in the security experiment of Specification 1 with probability greater than ϵ. This implies that the set of possible measurement parameters $\mathbf{C}$ must be very large, preferably exponential in some system parameter.

For the same reasons, it must be impossible for Eve to determine more than an ϵ-fraction of all CRPs within time t_C by exhaustive simulation on the basis of Sim and $D(S)$.

If S is a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system, then previous physical access for time t_{Ph} and computations of time t_C must not allow Eve to build a "clone" S' whose responses R'_{C_i} possess the following properties: (i) $R_{C_i} = R'_{C_i}$ for more than an ϵ-fraction of all $C_i \in \mathbf{C}$, and (ii) the generation of the R'_{C_i} works quickly, i.e., within time $c \cdot t_{max}$.

Immunity against Cloning. If S is a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system, then previous physical access for time t_{Ph} and computations of time t_C must not allow Eve to build a "clone" S' whose responses R'_{C_i} possess the following properties: (i) $R_{C_i} = R'_{C_i}$ for more than an ϵ-fraction of all $C_i \in \mathbf{C}$, and (ii) the generation of the R'_{C_i} works quickly, i.e., within time $c \cdot t_{max}$. In particular, the following three types of clones must be infeasible:

- *Physical clones*, i.e., exact physical reproductions of S that show the same challenge-response behavior on the same timescales.
- *Digital clones*, i.e., computer algorithms which numerically generate the same responses as S as fast as S.
- *Functional clones*, i.e., physical systems with a possible different structure or larger lengthscales that generate the same responses as fast as S.

The non-feasibility of functional clones is a particularly subtle requirement. It implies that there are no physical systems whose fabrication can be better controlled (for example because they operate on larger length scales), and which can emulate S in real-time. The related idea of simulating physical systems with (better controllable) other physical systems has again been discussed first by Feynman in [32].

No Secret Information in SIMPLs and the Role of the String X. The security of SIMPL systems should not depend on the secrecy of some sort of binary information contained in the SIMPL. Even if the adversary knows all details about the internal configuration of the SIMPL system, he shall be unable to break its security. Specification 1 formalizes this requirement by allowing the adversary to know any bitstring X of length q when trying to imitate the input–output behavior of the system. If, for example, one would try to construct a SIMPL by using a digital system with some secret key of length q, then the adversary could succeed in the experiment with probability one by using this key as the additional input X. No such digital, secret key based system can therefore serve as a SIMPL system in our sense.

Constant vs. Super-polynomial Time Gap. The time gap between Eve and the real SIMPL system S is required to be at least a constant factor $c > 1$ in Specification 1. This seems surprising, since one might expect the stipulation of an exponential gap here. The reasons for our choice are as follows. First, SIMPL systems with a small, constant speed advantage seem easier to realize in practice than systems with larger gaps, leaving alone systems with exponential margins. Secondly, it is unclear whether SIMPLs with an exponential time margin between Eve and the SIMPL exist at all. The only known realistic computational systems which might outperform Turing architectures by a super-polynomial factor are quantum computers [54]. But standard quantum computers possess no immunity against physical cloning. They could be mass-fabricated with the same functionality, and therefore appear unsuited as SIMPL systems. Third, it has been frequently hypothesized within the computational complexity community that there are no realistic hardware systems at all that solve NP-complete problems efficiently in practice. Two recent sources in this context are [52,53]. This further delimits the hope of SIMPL systems which possess an exponential security margin over Eve.

Fortunately, it turns out that many applications of SIMPL systems do not require exponential speed gaps. The protocols we suggest in this paper show that a constant, detectable time difference suffices in order to implement various cryptographic tasks (see Sections 3 to 5). An exponential time gap between the SIMPL system and any simulation machine is even undesirable in these protocols, since it would lead to too time consuming simulation steps for the honest protocol participants.

Feedback Loops. In order to create larger time margins, the absolute, but not the relative (!) time difference between the original SIMPL system and any fraudster can be amplified via feedback loops. Such feedback-loops can be constructed as follows: Presented with a challenge C_1, the SIMPL systems successively determines a sequence of k challenge-responses-pairs $(C_1, R_{C_1}), (C_2, R_{C_2}), \ldots, (C_k, R_{C_k})$, where later challenges C_n are determined by earlier results R_{C_m}, with $k \geq n > m \geq 1$. The tuple (C_1, R_{C_k}) is then regarded as the overall challenge-response pair of the SIMPL system; see [19] for further details. This strategy can amplify the absolute time margin between the SIMPL and the simulator and compensate network and transmission delays.

A concrete example will probably illustrate our point best. Let us assume that we possess a SIMPL system S which produces its responses in t_{max} of 10 nanoseconds (ns), and which possesses a speed advantage of $c = 2$ over all simulations. Any adversaries then cannot produce the response to a randomly chosen challenge within 20 ns. This tiny difference of 10 ns vs. 20 ns would not be detectable in many practical

settings, for example in networks with natural delays. Nevertheless, the application of repeated feedback loops can amplify not the relative, but the absolute time margin, to values such as 1 millisecond (ms) vs. 2 ms, or 1 sec vs. 2 sec. These values allow compensation of small transmission delays.

SIMPLs with Multi-bit Output. It can be convenient if a SIMPL system produces not just one bit as response, but a multi-bit output. Some implementations of SIMPLs have this property naturally (such as the optical implementation of section 7.3). Otherwise, feedback loops can allow us to create multi-bit outputs from SIMPL systems with 1-bit outputs: One simply considers a concatenation (or some other function, for example a hash function) of the last n responses $R_{C_{k-n+1}}, \ldots, R_{C_k}$ in the feedback loop as the overall output. Another option to create "large" SIMPL systems with k-bit outputs from "small" SIMPL systems with 1-bit outputs is to use k "small" SIMPLs in parallel, and to directly concatenate their responses [13].

A Digital Quasi-SIMPL (Which Does not Meet Specification 1). It may be useful for the readers to attempt to design digital, secret key based systems that have some of the properties of SIMPL systems. We call such systems quasi-SIMPLs. One possibility to construct a quasi-SIMPL is as follows: One takes a private key, public key pair (sk, pk) from a standard digital signature scheme, stores the secret key sk in a hardware system, and makes pk public. Upon receiving a challenge C, the hardware chooses a random number r of length k (with k being a public security parameter), and computes the hardware's response as $R_C = Sig_{sk}(C\|r)$ ($\|$ denoting concatenation). In order to verify that a certain response R_C is correct, one must test by exhaustive search if R_C is a correct signature of the string $C\|r$ for some bitstring r of length k. Choosing k of the correct length will create the desired speed gap.

If the key sk is stored safely in the hardware system, then — seen merely from the outside — it will behave similar as a SIMPL system, i.e., as a quasi-SIMPL. Nevertheless, we would like a true SIMPL system to be free of any secret key information; it would be desirable if Specification 1 ruled out quasi-SIMPLs. And indeed it does: setting the string $X = sk$ allows Eve to succeed in the security experiment of Specification 1 with probability 1. This again illustrates the usefulness of the specification, and stresses the important function of the string X within the specification.

Some early ideas related to quasi-SIMPLs, which are independent of our work, can be found in [33] and [34].

Error Correction. In Specification 1 and throughout the rest of the paper, we assumed for the simplicity of our treatment that the responses of a SIMPL system are stable. In practice, error correction must and can be applied to achieve this goal. We refer the reader to the comprehensive existing work on this topic [9,56,57,58,59], and ignore error correction aspects in the rest of the paper.

3 Identification and Message Authentication

We now proceed to several cryptographic protocols that can be implemented by SIMPL systems, starting with the identification of entities and the authentication of messages.

3.1 Identification of Entities

We assume that Alice holds an individual $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system S, and has made the corresponding data $D(S)$, Sim, the value $c \cdot t_{max}$, and a description of $\mathbf{C}$ public. Now, she can prove her identity to an arbitrary second party Bob as follows, with k being the security parameter of the protocol:

Protocol 2. IDENTIFICATION OF ENTITIES

1. Bob chooses k challenges $C_1, \ldots, C_k$ uniformly at random from $\mathbf{C}$.
2. **For** $i = 1, \ldots, k$ **do**:
   (a) Bob sends the value C_i to Alice.
   (b) Alice determines the corresponding response R_{C_i} by an experiment on her SIMPL system S, and sends this value to Bob.
   (c) Bob receives an answer from Alice, which we denote by V_i. If Alice's answer did not arrive within time $c \cdot t_{max}$, then Bob sets $V_i = \perp$ and continues the for-loop.
3. Bob computes the value $R_{C_i}^{Sim} = \mathsf{Sim}(C_i, D(S))$ for all $i = 1, \ldots, k$, and verifies if $R_{C_i}^{Sim} = V_i \neq \perp$. If this is the case, Bob believes Alice's identity, otherwise not.

In a nutshell, the security of the protocol follows from the fact that an adversary is unable to determine the values R_{C_i} for randomly chosen C_i comparably quickly as Alice. This holds as long as (i) the lifetime of the system S (and the period since $D(S)$ was made public) does not exceed t_C, and (ii) the adversary's accumulated physical access times do not exceed t_{Ph} (see Specification 1). In that case, the adversary's probability to succeed in the protocol without possessing S decrease exponential in k.

Bob can improve his computational efficiency by verifying the correctness of the responses R_{C_i} only for a randomly chosen subset of all responses. If necessary, possible network and transmission delays can be compensated for by amplifying the absolute time gap between Eve and S through feedback loops (see Section 2.3).

If the SIMPL system has multi-bit output (see Section 2.3), then a value of $k = 1$, i.e., a protocol with one round, may suffice. In these cases, the parameter ϵ of the multi-output SIMPL system will in itself be exponentially small in some system parameter (for example in the size of the sensor array in the optical SIMPLs discussed in Section 7.3).

3.2 Authentication of Messages

Alice can also employ an individual $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system S in her possession to authenticate messages to Bob. Again, we suppose that the values $D(S)$, Sim, $c \cdot t_{max}$, and a description of $\mathbf{C}$ are public.

Protocol 3. AUTHENTICATION OF A MESSAGE N

1. Alice sends the message N that shall be authenticated to Bob.
2. Bob chooses $k \cdot l$ challenges $C_1^1, \ldots, C_k^1, C_1^2, \ldots, C_k^2, \ldots, C_1^l, \ldots, C_k^l$ uniformly at random from $\mathbf{C}$.

3. **For** $i = 1, \ldots, l$ **do**:
   (a) Bob sends the values $C_1^i, \ldots, C_k^i$ to Alice.
   (b) Alice determines the corresponding responses $R_{C_1^i}, \ldots, R_{C_k^i}$ by experiments on her SIMPL system S.
   (c) Alice derives a MAC-key K_i from $R_{C_1^i}, \ldots, R_{C_k^i}$ by a publicly known procedure, for example by applying a publicly known hash function to these values. She sends $MAC_{K_i}(N)$ to Bob.
   (d) Let us denote the answer Bob receives from Alice by V_i. If V_i did not arrive in time $c \cdot t_{max} + t_{MAC}$, where t_{MAC} is the time to derive K_i and compute $MAC_{K_i}(N)$, then Bob sets $V_i = \perp$ and continues the for-loop.
4. For $i = 1, \ldots, k$ and $j = 1, \ldots, l$, Bob computes the values $R_{C_i^j}^{Sim} = \mathsf{Sim}(C_i^j, D(S))$ by simulation via Sim. He derives the keys $K_1^{Sim} \ldots, K_k^{Sim}$ by application of the same procedure (e.g. the same publicly known hash function) as Alice in step 3c.
5. For all $i = 1, \ldots, k$, Bob checks if it holds that $MAC_{K_i^{Sim}}(N) = V_i \neq \perp$. If this is the case, he regards the message N as properly authenticated, otherwise not.

The idea behind the protocol is that an adversary cannot determine the responses $R_{C_i^j}$ and the MAC-Keys $K_1, \ldots, K_l$ as quickly as Alice. As earlier, verification of a randomly chosen subset of all MACs can improve Bob's computational efficiency in step 5. Depending on the exact circumstances, a few erroneous V_i may be tolerated in step 5, too.

We assume without loss of generality in Protocol 3 that the MAC can be computed quickly (including the derivation of the MAC keys $K_1, \ldots, K_l$), i.e., within time t_{MAC}, and that t_{MAC} is small compared to $c \cdot t_{max}$. Again, this condition could be realized by amplification through feedback loops if necessary (see Section 2.3). It is known that MACs can be implemented very efficiently [38]. If information-theoretically secure hash functions and MACs are used, the security of the protocol will not depend on any assumptions other than the security of the SIMPL system.

If the SIMPL system has a multi-bit output, then values of $k = 1$, i.e., sending just one challenge in each round, or of $l = 1$, i.e., employing just one round of communication, may suffice. Such a multi-bit output can arise either naturally, for example through the choice of the SIMPL system itself (as noted earlier, the optical SIMPL system mentioned in Section 7.3 has this property). Or it can be enforced by feedback loops, or by using several independent SIMPL systems in parallel (see Sections 2.3 and 2.3). In fact, such measures even are strictly necessary to uphold the protocol's security if the constant c has got a very low value.

4 Two-Player Protocols

SIMPL systems also have a notable potential for two-player protocols. This extends their application potential, but had not been addressed in earlier publications. Three important protocols are covered in this section.

4.1 Coin Flipping

Coin flipping [35] is a long known two-player protocol which can serve well as a first simple touchstone for the potential of SIMPLs with respect to two-party schemes. Its basic setting is as follows: Two players Alice and Bob want to communicate over a binary channel in order to produce a random binary value B ("a fair coin") as output. The protocol must guarantee that the output cannot be biased or pre-determined by one of the players; see [35] and [48] for more details.

In our setting, we assume that Alice holds a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system with description $D(S)$, and that Bob knows $D(S)$, Sim, and $\mathbf{C}$. Without loss of generality, we assume that the responses of S have a length of one bit (otherwise, one can take the exclusive or of all single bits in the response string, or apply another suited function to the responses). Under these circumstances, a time-restricted coin flipping protocol based on SIMPL systems can be implemented as follows:

Protocol 4. COIN FLIPPING

1. Alice sends a randomly chosen challenge $C \in \mathbf{C}$ to Bob.
2. Bob immediately after receipt of C answers by sending a random bit r to Alice.
3. Alice verifies if she received r within time less than $c \cdot t_{max}$ after she sent C. If not, she aborts the protocol. Otherwise, she determines R_C by measurement on S, and sets the flipped coin to be $B = R_C \oplus r$.
4. Bob verifies if $C \in \mathbf{C}$, and aborts if this is not the case. He determines R_C by simulation, and sets the flipped coin to be $B = R_C \oplus r$.

The security of the protocol straightforwardly follows from the assumption that S is a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system: If Alice receives the value r within time $c \cdot t_{max}$, then Bob cannot know R_C before he sends away r. He hence cannot choose r as a function of R_C in order to bias the outcome of B. Protocol 4, for the first time, illustrates a potential for two-player protocols in SIMPLs which goes beyond the classical identification and message authentication applications.

4.2 Bit Commitment

Can more advanced two-party protocols be realized on the basis of SIMPL systems? One good candidate to investigate is bit commitment (BC) [47,48].

BC is a two-player protocol where one party acts as the sender, and a second party acts as the receiver. The sender holds a bit b at the beginning of the protocol, while the receiver holds the empty input. The protocol has two stages, a commit phase and a reveal phase. At the end of the commit phase, the sender and receiver must have interacted in such a way that the sender has bound or committed himself to the bitvalue b by the communication, but that the receiver does not know this value, and finds it infeasible to derive it from the communication. In the reveal phase, the sender "opens" his commitment and allows the receiver to learn b. After completion of the commit phase, it must be infeasible for the sender to change the commitment he made, and to run the reveal phase in such a way that the receiver learns a different bit $1 - b$. Further details and a formal definition can be found in [48]. Bit commitments are important components of

zero-knowledge proofs [49,50], and other, more general two-party cryptographic protocols [51]; see again [48] for further information.

The SIMPL-based BC scheme we suggest here employs interactive hashing (IH) [44] as a sub-protocol. IH is another useful two-player protocol, in which Alice's initial input is an m-bit string C, and Bob has no input. At the end of the protocol, Alice and Bob know two m-bit strings C_0 and C_1, with the properties that (i) $C_j = C$ for some bit $j \in \{0, 1\}$, but Bob does not know the value of j, and that (ii) the other string C_{1-j} is a random bitstring of length m, which neither Alice nor Bob can determine alone. Secure IH can be realized in an information theoretic fashion, i.e., independently of any computational or other unproven assumptions. For further details, see [44,45,46].

In the following Protocol 5, Alice acts as the sender and Bob as the receiver of the bit b. We assume that Bob holds a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system S, and that Alice knows $D(S)$, Sim and $\mathbf{C}$, and holds a bit b she wants to commit. The protocol splits in a commit phase and a reveal phase, and works as follows.

Protocol 5. BIT COMMITMENT

Commit Phase:

1. Alice chooses a random challenge C from $\mathbf{C}$, and determines R_C by simulation.
2. Alice and Bob start an interactive hashing protocol. Alice's input is C, and Bob's input is the empty string. Both get two strings C_0 and C_1 as output.
3. Alice determines the index i for which $C_i = C$, and sends Bob the value $i \oplus b$.

Reveal Phase:

4) Alice sends Bob the values i and R_{C_i} (which is equal to R_C if Alice behaves honestly, and hence known to her from step 1).
5) Bob checks if the time interval between the start of the IH protocol in step 2 and the reception of the values i and R_C in step 4.2 is smaller than $c \cdot t_{max}$. If this is the case, he verifies by measurement on S that the value R_{C_i} sent by Alice is correct. If this holds, too, he accepts the BC as valid, and reveals the committed bit by computing $(i \oplus b) \oplus i = b$.

Please note that the commit phase and the reveal phase of this scheme must be executed relatively closely after each other. In particular, Alice must not have time to compute the value $R_{C_{1-i}}$ in the time interval between completion of the interactive hashing protocol in step 2 and the reveal step 4. If she could compute $R_{C_{1-i}}$, she can open the commitment at will by sending either the values i and R_{C_i}, or the values $1 - i$ and $R_{C_{1-i}}$ in step 4.

This means that the so-called binding property of the above BC scheme (i.e., the fact that Alice cannot change the value anymore after the commit phase) is conditional upon the prompt execution of the reveal phase. On the other hand, the so-called hiding property of the scheme (i.e., the fact that Bob will not learn b unless the reveal phase is executed) is unconditional: No matter how much time passes, Bob cannot learn the bit b unless Alice gets engaged in the reveal phase.

This implies that if the protocol fails to be executed within said time limits (for example, because the network is down, or other delay occurs), it can be restarted arbitrary many times without endangering the confidentiality of Alice's bit b. The time restriction will therefore not constitute a severe disadvantage in many settings.

4.3 Zero-Knowledge Proofs

Zero-knowledge proofs (ZK proofs) [49,50] are a very powerful two-party scheme, in which one party acts as the so-called prover, the other as the so-called verifier. The setting is as follows: The prover is in possession of a solution W to a computationally hard problem Π (for example, a three-coloring of a certain, publicly known, hard graph G), and wants to prove to the verifier that he indeed knows such a solution W to Π — but without revealing W to the verifier. For further details, see [49,50,48]. Some application examples of ZK proofs are passwords schemes and authentication systems, as well as the enforcement of honest behavior in cryptographic protocols while maintaining the privacy of the users. Along these lines, they are an essential component in secure multi-party computations [36,48].

In the following, we give a ZK proof for the three-coloring of a graph that rests on the above SIMPL-based BC protocol. By a well-known reduction result [48] and the NP-completeness of the three-coloring problem, this implies that there are SIMPL-based ZK proofs for all languages in NP. Our proof again employs interactive hashing as a subprotocol; see Section 4.2. In our protocol, we assume that a finite graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ is public, and that Alice knows a three coloring $W : V \to \{00, 01, 11\}$ for this graph. Furthermore, we suppose that Bob holds a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system S, and that Alice knows $c \cdot t_{max}$, $D(S)$, Sim and $\mathbf{C}$. Finally, without loss of generality we assume that the output of S are one-bit values (otherwise, one can take for example the XOR of all output bits to obtain one-bit responses, or apply another suitable function to the output bits).

Protocol 6. ZK PROOF OF A THREE-COLORING W

1. Alice selects $2n$ challenges $C_1, \ldots, C_{2n}$ at random, and determines $R_{C_1}, \ldots, R_{C_{2n}}$ by simulation.
2. Alice selects a random permutation π over $\{00, 01, 11\}$, and forms the string $L = \pi(W(1)) \cdot \pi(W(2)) \cdots \pi(W(n))$.
3. Alice and Bob run $2n$ interactive hashing protocols. In the i-th protocol, Alice's input is C_i, and Alice's and Bob's output is C_i^0, C_i^1. We denote by $k_i \in \{0, 1\}$ the index for which $C_i = C_i^{k_i}$, and define K as $K = k_1 \cdot k_2 \cdots k_{2n}$.
4. Alice sends the string $X = X_1 \cdots X_{2n} = L \oplus K$ to Bob.
5. Bob at random chooses an edge $e = (l, m) \in E$ and sends e to Alice.
6. Alice sends the four values $T = k_{2l-1}, U = k_{2l}, V = k_{2m-1}, W = k_{2m}$ and the corresponding responses $R_{C^T_{2l-1}}, R_{C^U_{2l}}, R_{C^V_{2m-1}}, R_{C^W_{2m}}$ to Bob.
7. Bob verifies if: (i) The two vertices of the edge e are colored differently. He does so by checking whether $(X_{2l-1} \oplus k_{2l-1}) \cdot (X_{2l} \oplus k_{2l}) \neq (X_{2m-1} \oplus k_{2m-1}) \cdot (X_{2m} \oplus k_{2m})$. (ii) The purported responses $R_{C^T_{2l-1}}, R_{C^U_{2l}}, R_{C^V_{2m-1}}, R_{C^W_{2m}}$ are correct. He does so by measurement on S. (iii) The time that passed between step 3 and step

6 is at most $c \cdot t_{max}$. If (i) to (iii) hold, Bob accepts this run of the protocol as successful.

The protocol has an error rate of up to $1 - 1/|E|$. As usual, polynomially many independent runs can downscale this error rate to any desired value [48]. As noted earlier, it can be observed that if a single run of the protocol fails to be executed within the required time limits (for example, because the network is down), the confidentiality of Alice's three-coloring W is still maintained. This is guaranteed by the fact that the SIMPL-based bit commitment scheme of Protocol 5 is unconditionally hiding.

5 Key Exchange

Secure key exchange is another central cryptographic task in which SIMPL systems and Public PUFs can assist us. We treat this topic at the end of our protocol discussion for two reasons: First of all, we use material that was originally introduced by others (namely Protocol 7); and second, because one suggested scheme (Protocol 8) builds on the message authentication method of the earlier Section 3.2.

5.1 Key Exchange via PPUFs

As noted in Section 1, PPUFs [24] are an essentially equivalent concept to SIMPLs. One application suggested in [24] is a key exchange scheme. It requires a special type of SIMPL system, which we call a PPUF, giving honor and credit to [24].

Let S be a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-SIMPL system, and let the function F_S implemented by S fulfill the following additional properties:

(i) F_S is a one-to-one function.
(ii) F_S is a one-way function, i.e., it is hard to invert.
(iii) The time gap c between any simulation and the real-time behavior of S is very large (examples discussed later on require orders of $c > 10^5$ or similar magnitudes).

Under these circumstances, we call S a $(t_{max}, c, t_C, t_{Ph}, q, \epsilon)$-PPUF. Implementations of such systems have been suggested in [24].

On the basis of a PPUF, we can implement a key exchange scheme as described in Protocol 7. Before giving the protocol, we stress once more that the protocol has originally not been devised by us, but is an abstraction from the concrete setting of [24] (i.e., from the concrete PPUF implementation that is used there).

We assume that Alice holds the PPUF S and that Bob knows the corresponding sets and algorithms $D(S)$, Sim and $\mathbf{C}$.

Protocol 7. KEY EXCHANGE WITH PPUFS

1. Bob chooses at random a subset $\mathbf{U}$ of the set of all possible challenges $\mathbf{C}$, with the property that $\mathbf{U}$ can be characterized by a short string I_U.
2. Bob chooses k random challenge $C_1, \ldots, C_k$ from $\mathbf{U}$. He derives a key K from $C_1, \ldots, C_k$ by a publicly known procedure (e.g., a hash function), and determines $R_{C_1}, \ldots, R_{C_k}$ by simulation of S.

3. Bob sends $I_U, R_{C_1}, \ldots, R_{C_k}$ to Alice.
4. Alice uses the PPUF S for a simple exhaustive search in order to find $C_1, \ldots, C_k$: She applies all possible challenges $C' \in \mathbf{U}$ to the PPUF, and compares the response to $R_{C_1}, \ldots, R_{C_k}$. If it matches R_{C_i}, she has found C_i. She derives the same key K from the responses by using the same publicly known procedure as Bob.

Depending on the exact PPUF S that is in use, examples for suitable choices for the sets $\mathbf{U}$ could be the set of all challenges in $\mathbf{C}$ that start with a certain substring; sets of the form $\mathbf{U} = \{x_0, ..., x_0 + n\}$, where x_0 and n are natural numbers; or sets of the form $\mathbf{U} = \{H(x) \,|\, x \in \{x_0, ..., x_0 + n\}\}$, where x_0 and n are natural numbers, and H is a publicly known hash function. The latter choice for $\mathbf{U}$ has been employed in the original protocol of [24]. It possesses several advantages, such as distributing the challenges somewhat randomly within $\mathbf{C}$.

Discussion and Analysis. Note that S and F_S must really fulfill the properties (i) to (iii) stated in Section 5.1 in order to make the protocol work: If F_S was not one-to-one, then the determination of the C_i is ambiguous; Alice's and Bob's keys will not match. Secondly, if F_S was not one-way, then an adversary could eavesdrop the communication, learn $R_{C_1}, \ldots, R_{C_k}$, invert F_S in order to learn $C_1, \ldots, C_k$, and thus derive K. Finally, if feature (iii) is not fulfilled, an adversary Eve could *by numerical simulation* perform the same exhaustive search as Alice in order to identify the values $C_1, \ldots, C_k$ relatively efficiently (see also below). Properties (i) to (iii) therefore are necessary requirements. This is in opposition to earlier protocols, where the employed SIMPL system does not need to fulfill (i) to (iii), making their hardware implementation easier. For example, Protocols 2 to 6 could work with SIMPLs with small time gaps c.

We now analyze the security margin of the protocol in more detail (compare [24]). Let us assume that Bob can simulate the PPUF's response on any challenge in time t_{sim}. As follows from Specification 1, $c \cdot t_{max} \leq t_{sim}$. Furthermore, Specification 1 implies that Alice can execute her measurement on S in time t_{max}, and any adversary Eve requires at least time $c \cdot t_{max}$ in order to simulate the PPUF's response to a randomly chosen challenge.

It therefore holds for Alice's expected workload W_A and Bob's workload W_B in the above protocol that $W_A \approx t_{max} \cdot k/(k+1) \cdot |\mathbf{U}|$, and $W_B \approx t_{sim} \cdot k \geq c \cdot t_{max} \cdot k$. On the other hand, an adversary Eve who numerically simulates all responses $C \in \mathbf{U}$, and who can simulate one response in time $c \cdot t_{max}$, has an expected workload of $W_E \approx c \cdot t_{max} \cdot k/(k+1) \cdot |\mathbf{U}|$. Note that the factors $k/(k+1)$ come in due to standard probability theory as we consider expected workloads.

Thus, the relative advantage of Alice over an adversary who applies the above simple attack strategy of exhaustive search, is $W_E/W_A \approx c$, or

$$W_E \approx W_A \cdot c. \tag{1}$$

In other words, Eve's workload is only separated by the SIMPL system's constant c from the workload of Alice. In order to achieve a long term security of the key, this requires a very large c or substantial values for W_A. Let us consider a few examples: If we stipulate that W_E is required to be on the order of 100 years for security reasons, then $c = 10^5$ makes a workload of $W_A \approx 8.76$ hours necessary for Alice; $c = 10^7$

implies $W_A \approx 5.3$ min; and in order to achieve $W_A \approx 0.3$ sec, a time gap of $c = 10^{10}$ is required. It seems yet uncertain if such large time gaps can be achieved by practical and inexpensive hardware implementations of SIMPL systems; an alternative method that requires only smaller values for c is described in the upcoming Section 5.2.

Finally, we note that protocol in practice requires an authenticated channel, which can either be realized by classical means, or by SIMPL/PPUF-based message authentication a la Protocol 3.

5.2 Authenticated Key Exchange by SIMPLs and Diffie-Hellman

An alternative approach to Protocol 7 is to combine the Diffie-Hellman key exchange protocol with the SIMPL-based message authentication scheme of Protocol 3. This presumes that Alice holds a $(t^A_{max}, c^A, t^A_C, t^A_{Ph}, q^A, \epsilon^A)$-SIMPL system S_A, Bob holds a $(t^B_{max}, c^B, t^B_C, t^B_{Ph}, q^B, \epsilon^B)$-SIMPL system S_B, and that both know the respective values $D(S_A), D(S_B)$, c^A, c^B, t^A_{max}, t^B_{max}, and the algorithm Sim. The protocol is straightforward, but we include it for reasons of completeness.

Protocol 8. AUTHENTICATED KEY EXCHANGE BY SIMPLS AND DH (SCHEMATIC)

1. Alice chooses a random exponent a. She sends the message g^a to Bob, authenticated by use of her SIMPL System S_A and Protocol 3.
2. Alice chooses a random exponent b. He sends the message g^b to Bob, authenticated by use of his SIMPL System S_B and Protocol 3.
3. Both form the exchanged key as $K = g^{ab}$.

One asset of Protocol 8 is that it inherits its long-term security and its authenticated channel from two different sources. It can be carried out efficiently (if SIMPLs with small c^A, c^B and t^A_{max}, t^B_{max} are used), and can hence be employed for the ad-hoc exchange of session keys in communication networks. These keys can be erased whenever needed, being in line with our overall goal of avoiding the long term-presence of secret keys in hardware.

The long-term confidentiality of the protocol, on the other hand, is derived from the well-established Diffie Hellman (DH) assumption. It establishes a large, asymptotically exponential security margin between the computational effort that must be invested by the honest parties to run the protocol and by the adversary to obtain the exchanged key.

Please note in this context that the DH function is a digital function that is optimized in terms of its security properties. It does not need to fulfill any other, possibly involved criteria. Contrary to that, the function implemented by the PPUF/SIMPL in Protocol 7 must be a non-invertible function, similar to the DH function. But in addition, it must depend on unclonable random analog features of the hardware, be stable against environmental conditions and aging, and must be vastly faster than any digital simulator. We feel that this agglomeration of features could potentially become problematic, and that the simulation gap of SIMPLs/PPUFs might be overstretched when it is used to establish the long-term security of a key or the long-term confidentiality of data.

In our opinion, Protocol 8 thus constitutes a viable, at times preferable alternative to Protocol 7,

6 Applications of SIMPL Systems

6.1 Secure Communication Infrastructures

Within the given space restrictions, we will now discuss the application of SIMPL systems to secure communication in networks, illustrating their potential in such a setting. Consider a situation where k parties $P_1, \ldots, P_k$ and a trusted authority TA participate in a communication network. Assume that each party P_i carries its own SIMPL S_i in its hardware, and that a certificate C_i has been issued for each party by the TA. The certificate includes the identity and the rights of Party P_i, and has the form

$$C_i = \big(Id_i, Rights_i, D(S_i), Sig_{TA}(Id_i, Rights_i, D(S_i))\big).$$

Under these provisions, the parties can mutually identify themselves by Protocol 2, they can establish authenticated channels with each other by Protocol 3. They can exchange session keys via the use of the Protocol 8 (or, alternatively, Protocol 7). The whole architecture works without permanent secret keys, or without any other secret information that is stored permanently in the hardware of the parties $P_1, \ldots, P_k$.

It also seems well applicable to cloud computing: All personal data could be stored centrally. Session keys could be exchanged by the Diffie-Hellman protocol over channels authenticated by the SIMPL systems (Protocol 8). These keys can be used to download the personal data in encrypted form from the central storage. The keys can be new in each session, no permanent secret keys in the mobile hardware are be necessary.

The above approaches can further be combined with tamper-sensitive SIMPL systems. These SIMPLs may cover hardware which has a functionality $Func_i$ as long as it is non-manipulated. Each certificate C_i could then also include the functionality of the hardware, i.e., it could be of the form

$$\begin{aligned} C_i = \big(&Id_i, Rights_i, Func_i, D(S_i), \\ &Sig_{TA}(Id_i, Rights_i, Func_i, D(S_i))\big). \end{aligned}$$

By running the identification protocol (Prot. 2), party P_i can prove that the SIMPL system S_i is non-tampered, and that the hardware hence has the claimed functionality $Func_i$. Please note that the optical SIMPL systems we propose in this paper is naturally tamper sensitive; the tamper sensitivity of such optical scattering structures has already been shown in detail in [8].

Finally, by using Protocols 4, 5 and 6, all parties can execute several typical two-party computations with each other, leading to various further cryptographic applications.

6.2 Two Other Applications: Unforgeable Labels and DRM

Let us in all brevity sketch to two other applications of SIMPL systems, which have been described in more detail in [16].

The first of these applications is the generation of unforgeable labels for products or security tokens. SIMPL systems can create labels which do not contain any secret information, which can be verified offline, and which only require remote, digital communication between the label and a testing device.

SIMPL systems can be applied in this context. A SIMPL-label consists of the following components: (i) The SIMPL System S; (ii) The description $D(S)$ and some product related info I; and (iii) the digital signature $Sig_{SK}(D(S), I)$, created by the secret signing key SK of the label issuer. Components (ii) and (iii) are digital information that can be stored on the labeled item of value, for example via a printed barcode or electronic means.

In the verification of a label, a testing apparatus obtains $D(S)$ from the label, verifies the digital signature via use of a publicly known verification key PK, and executes Protocol 2 in order to check the presence of the SIMPL system S. A description of $\mathbf{C}$, t_{max} and Sim need to be hardwired into the apparatus together with PK. If more than one label issuer is involved, the apparatus can store more than one public verification key, or standard signed key certificates can be employed.

Labels based on SIMPL system have interesting advantages: They can be read out digitally and remotely. Secondly, they can be verified be offline, i.e. without an online connection to a central institution/database. The labels do not contain any secret information at all, also not in the form of a PUF. Finally, also the testing apparatus that evaluates the validity of a label does not need to contain any form of secret information. The only secret key involved in the scheme remains centrally with the issuer of the label, where it can be well protected. In combination, these features distinguish SIMPL-based labels from other known approaches.

Note that the issuer of a SIMPL-based labels can create the required signature of component (iii) remotely, i.e., he does not need to be present at the production site where the label is generated and attached to the item of value. His secret signing key can be kept to him alone. This is particularly useful in situations where illegitimate overproduction at remote manufacturing sites must be encountered.

Another application area of SIMPLs lies in the context of the digital rights management problem (DRM). Similar to the above labels, SIMPLs can also create unclonable representations of digital content, including software [16]. These unclonable representations do not contain any secret information, and can be verified by a testing device that does not need to contain any secret keys either. The verification works offline and by mere digital communication between the testing device and the device carrying the unclonable representation. Again, in combination these features are not met by any comparable technique known to the author. In [40,41,42], for example, the random features of the data carrier must be determined in the near-field by analog measurements. The features must be communicated correctly by the analog measurement apparatus (e.g., the optical drive) to a central module (e.g., a TPM) that decides about the validity of the content, meaning that the measurement apparatus must be trusted.

7 Implementation of SIMPL Systems

We now turn to the practical implementation of SIMPL systems. Our aim is to give an overview of the particular challenges in the realization of SIMPLs and the existing implementation candidates, and to refer the reader to the existing literature for the details of the described approaches.

7.1 Challenges

There are some clear challenges in the realization of SIMPL systems. Three non-trivial requirements that must be balanced are complexity, stability, and simulatability: On the one hand, the output of a SIMPL system must be sufficiently complex to require a long computation/simulation time. On the other hand, it must be simple enough to allow simulation at all, and to enable the determination of $D(S)$ by measurement or numeric analysis techniques. A final requirement is that the simulation can be carried out *relatively* efficiently by everyone (this is necessary to complete the verification steps in the identification and message authentication protocols quickly); while, at the same time, even a very well equipped attacker, who can potentially attempt to parallelize the simulation on many powerful machines, cannot simulate as fast as the real-time behavior of the SIMPL system. In the sequel, we list several implementations that show potential to meet these demanding requirements.

7.2 Electrical SIMPL Systems

Since the first publication of [16], a sequence of papers of our group has dealt with the implementation of SIMPL systems by electrical, integrated circuits [17,18,19,21]. We tried to exploit two known speed bottlenecks of modern CPUs: Their problems in dealing simultaneously with very large amounts of data, and the complexity of simulating inherently analog and parallel phenomena. Let us briefly summarize these approaches from said papers.

"Skew" SRAM Memories. A first suggestion made in [17,18,19,21] is to employ large arrays of SRAM cells with a special architecture named "skew design". In this design, the write behavior of the cells is dependent on the applied operational voltage. If the operational voltage is below a certain threshold, all write operations malfunction. The simulation of many successive read- and write events of the skew SRAM memory under quickly varied operational voltages on a standard architecture then necessarily creates some computational overhead, since in the standard architecture the bit values that are effectively written into the cells must be pre-computed as a function of the operational voltages and the a priori unknown content of the target cell. The hypothesis put forward in [17,18,19,21] is that this creates a small, constant simulation overhead, in particular that it creates the necessity for additional read-operations. Two essential ingredients in this concept are: No parallelization is possible, since the successive read- and write events in the feedback loop are made dependent on the previous read results. And since no parallelization is possible, the limiting factor for an adversary is his clock frequency, which is quite strongly limited by current technology.

As argued in the listed references, the idea shows promise to succeed against any adversaries with a limited financial budget, and in particular to defeat any FPGA-based attacks. Future work will need to characterize how large the exact simulation margin is, and whether it is indeed sufficient to defeat an adversary with strong financial resources who is capable of fabricating ASICs. Due to its relatively easy realizability and good security level, the concept has a good potential for the consumer market.

Two-dimensional Analog Computing Arrays. A second suggestion of [17,18,19,21] consists of using analog, two-dimensional computing arrays. The authors suggest the

use of so-called cellular non-linear networks (CNNs) which are designed to imitate non-linear optical systems. Due to their analog and inherently parallel nature (many cells exchange information at the same time), CNNs are time consuming to simulate on a digital, sequential architecture. This claim is supported by the standard literature on CNNs, which describes that these analog architectures can outperform classical digital computers by factors of up to 1,000 in certain, specialized tasks like image recognition [22,23].

The use of CNNs has its assets on the security side: Since it is based on manufacturing mismatches in CNN fabrication that currently seem unavoidable, it could eventually defeat even attackers with very strong financial resources, and has the potential to create SIMPLs that cannot even be clobed by their own manufacturer (i.e., SIMPLs which are manufacturer resistant in the sense of [29]). On the downside, since CNNs are complex analog circuits, they might be less suited for low-cost applications.

Other Electrical Approaches. Independently, the work of other groups has lead to different electrical structures that could be used as SIMPLs. The implementation of PPUFs presented in [24] could potentially be downscaled to become a SIMPL system, even though it would have to be carefully investigated how resilient such small-scale instances are against parallelization attacks. Another very interesting, FPGA-based candidate for SIMPLs is implicit in the work of [26].

7.3 Integrated Optical SIMPLs

A second route that was followed in the implementation of SIMPL systems is the employment of optical structures [16,20]. The rationale behind this strategy is as follows: First, optical systems can potentially achieve faster component interaction than electronic systems; this promises to create the desired speed advantage over any electronic simulator. In particular, the phenomenon of optical interference has no electronic analog at room temperature [61], and can create a computational overheads. Second, the material degradation of optical systems is low, and their temperature stability is known to be high [61,62]. Even very complex and randomly structured optical systems, whose internal complexity creates the desired speed gaps, can produce outputs that are relatively stable against aging and environmental conditions.

A concrete optical SIMPL system was suggested in [20]. It comprises of an immobile laser diode array with k phase-locked diodes $D_1, \ldots, D_k$ [63], which is attached to a disordered, random optical scattering medium. The diodes can be switched on and off independently, leading to 2^k possible challenges or inputs C_i to the medium. These challenges can be written as $C_i = (b_1, \ldots, b_k)$, where each $b_i \in \{0, 1\}$ indicates whether diode D_i is switched on or off. Note that the diode array must indeed be phase locked in order to allow interference of the different diode signals. At the oppposite side of the medium, an array of l light sensors $S_1, \ldots, S_l$, e.g. photodiodes, measures the resulting wave front when leaving the scattering medium: It detects the local light intensities at each of the sensors. A response R_{C_i} thus consist of the intensities $I_1, \ldots, I_l$ in the l sensors. Instead of phase-locked diode arrays, also a single laser source with

a subsequently placed, inexpensive light modulator (as contained in any commercially available beamer) can be employed.

Under the provision that a *linear* scattering medium is used in such integrated optical SIMPLs, the input/output behavior of this SIPML can be machine learned and predicted. This was shown by a proof of concept implementation in [20]. As argued in the same publication, there is also a time margin between any numeric simulator and real implementations of the system that are optimized with respect to speed: While the real system can create its output pattern in nanoseconds, the simulation requires around $k \cdot l$ additions of precomputed values. For moderate sizes of the system of $k = l = 10^4$, this requires 10^8 precomputed values and 10^8 additions. This can create exactly the notable, constant speed gap between the real system and the simulator that is required in SIMPL systems.

7.4 Other Implementation Strategies

There are two further promising implementation strategies that could assist us in creating secure future generations of SIMPLs.

Employing PUFs with Reduced Complexity. One generic further strategy for the realization of SIMPL systems, which has been suggested already in [16], is the following: Employ a PUF or a PUF-like structure; and reduce its inner complexity until it can be characterized by measurements and simulated, or until it can successfully be machine learned. If the level of complexity is still sufficient, then this simulation will be more time consuming than the real-time behavior of the system. In fact, some suggestions of the previous subsections used this strategy already, since both CNNs and integrated optical structures have already been suggested as PUFs in earlier work [55,12].

Simulation vs. Verification. Another idea is to exploit the well-known asymmetry between actively computing a solution for a certain problem and verifying the correctness of a proposed solution (as also implicit in the infamous P vs. NP question) [16]. Exploiting this asymmetry could lead to protocols of the following kind: A SIMPL system provides the verifier in an identification/authentication protocols with some extra information that allows the verifier to *verify* its answers fast. To illustrate our point, imagine an analog, two-dimensional, cellular computing array whose behavior is governed by partial differential equations (PDEs), such as the CNN described in section 7.2. Then, verifying the correctness of a given final state of such a PDE-driven system (i.e. verifying that this state is indeed a solution of the PDEs driving the system) could be much more time efficient than computing this solution from scratch. Furthermore, the verifier could not only be given external outputs of such a two-dimensional array (e.g. values in boundary cells), but also internal sub-measurements (e.g. values in inner cells) that help him to verify the output quickly.

The simulation vs. verification strategy can help to relieve the tension between the requirement for fast simulation on the side of the verifier (who may not be well equipped on the hardware side) and the necessary time margin to any attackers (who may be very well equipped on the hardware side), which we already mentioned in Section 7.1.

8 Summary, Discussion, and Future Work

8.1 Summary

This paper introduced and discussed a security concept termed *SIMPL system*. We started out by explaining the basic idea behind this new concept, and developed a semi-formal specification of the exact security properties of SIMPL systems in Section 2. Some basic properties that follow from this specification were discussed in the same section, for example the impossibility for cloning a SIMPL system, or for reading out its entire CRP-space. Next, we presented several protocols that can be realized by SIMPL systems in Sections 3 to 5. They include identification, message authentication and key exchange schemes, as well as two-party protocols like coin-flipping, bit commitment, and zero-knowledge proofs of NP-complete languages. We argued that the time restrictions required for these protocols (i.e., the fact that some of them must be executed withint a certain time bound in order to guarantee their security) do not too strongly diminish their practical usability in many relevant settings. Our work reveals the substantial *cryptographic* potential of SIMPL systems, including their application to classical two-party problems, which was previously undiscovered.

Concrete application scenarios of SIMPLs were discussed in Section 6. We described communication infrastructures that work without permanent secret key information in the hardware, and where the hardware can remotely prove its functionality to other parties. Other applications we investigated were unforgeable product labels and digital rights management. In all of these scenarios, SIMPL systems allow us to design cryptographic hardware that does not contain any secret key information, that is, any information whose disclosure breaks the security of the system. This can lead to future generations of hardware that does not require costly protection mechanisms on the physical and software level – there simply is no secret key to protect in SIMPL based hardware. This could make future security hardware more lightweight, mobile and secure at the same time.

Finally, the implementation of SIMPL systems was addressed in Section 7. Due to the large body of existing work, we focused on surveying current implementation candidates, and provided the reader with references to the literature. We covered electrical implementations based on special SRAM memories, two-dimensional analog arrays known as cellular non-linear networks (CNNs), and addressed suggestions by other groups based on circuit glitches and FPGAs. We also pointed to a recent, promising, and integrated optical candidate.

8.2 Discussion and Analysis

Let us conclude this work by a detailed comparative analysis of SIMPL systems. As said earlier, there are some obvious similarities between classical private/public key cryptoschemes and SIMPL systems: The numeric description $D(S)$ is some analog to a public key, while the physical system S itself constitutes some equivalent to a private key. This provides SIMPLs with a public-key like functionality. It allows new protocols and leads to several practicality advantages, as discussed in previous sections.

Still, there is one important difference to classical, mathematical public-key systems: Our "private key" is no secret number, but a randomly structured, hard-to-clone *physical*

system, the SIMPL system S. It has the interesting feature of not containing any form of secret information: Neither in an explicit digital form like a digital key in classical hardware. Nor in a hidden, analog form such as internal PUF parameters (for example the mentioned delay values in Arbiter PUFs, or the parameters determining SRAM behavior in SRAM PUFs). All internal characteristics of a SIMPL, including its precise internal configuration, can be publicly known without compromising the security of the derived cryptographic protocols.

The security of SIMPL systems is not free of assumptions, though. Instead of presupposing the secrecy of some sort of information, it rests on the following two hypotheses: (i) on the computational assumption that no other, well-controllable, configurable, or even programmable hardware can generate the complex responses of a SIMPL with the same speed, and (ii) on the physical assumption that it is practically infeasible for Eve to exactly clone or rebuild the SIMPL system, even though she knows its internal structure and properties. [2]

It is long accepted that computational assumptions play a standard role in mathematical cryptography, and they are also a part of the security assumptions for SIMPL systems; but SIMPLs show that one can trade the need for secret information in the hardware against assumptions on the physical unclonability of the system. This can surprisingly obviate the familiar requirement that cryptographic hardware must contain secret key information of some sort. By the protocols presented in this paper, the communicants can nevertheless execute a very large number of cryptographic protocols and tasks, without employing long-term present secret key information.

8.3 Future Work and Prospects

Future work on SIMPLs will likely concentrate on developing new protocols for SIMPL systems, and on devising formal security proofs for these protocols. For example, it seems interesting if time-restricted, but still useful variants of secure multi-party computation could be implemented by SIMPLs, and how the security of such constructions could be proven. But perhaps the greater challenge lies on the hardware side: Even though there are several promising candidates (see Section 7), the issue of finding a highly secure, practical, and cheap implementation appears not to be fully settled yet. If such an implementation is found, or if the existing implementation candidates are shown to possess all necessary properties, this could potentially change the way we exercise cryptography and security today.

Acknowledgements. The author would like to thank Jürg Wullschleger for suggesting the presented coin flipping protocol, and Ulf Schlichtmann, Stefan Wolf, Jürg Wullschleger, Georg Sigl, Srinivas Devadas, Miodrag Potkonjak and Farinaz Koushanfar for very useful discussions on the general topic of SIMPLs/PPUFs, and David Naccache for a very helpful exchange on quasi-SIMPLs. This work was done within the physical cryptography project at the TU München.

[2] The reader can verify the plausibility of the latter unclonability property by considering the optical implementation of section 7.3: Even if the positions of all scattering centers and the other irregularities in the scattering medium were known in full detail, it would still be infeasible to rebuild the scattering medium with perfect precision.

References

1. http://www.cbsnews.com/stories/2010/02/15/business/main6209772.shtml
2. http://www.bbc.co.uk/news/10569081
3. http://www.eurosmart.com/images/doc/Eurosmart-in-the-press/2006/cardtechnologytoday_dec2006.pdf
4. http://www.gsaietsemiconductorforum.com/2010/delegate/documents/GASSELGSALondon20100518presented.pdf. (Slide 23)
5. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Manzuri Shalmani, M.T.: On the power of power analysis in the real world: A complete break of the KEELOQ code hopping scheme. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 203–220. Springer, Heidelberg (2008)
6. Kasper, T., Silbermann, M., Paar, C.: All you can eat or breaking a real-world contactless payment system. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 343–350. Springer, Heidelberg (2010)
7. Anderson, R.J.: Security Engineering: A Guide to Building Dependable Distributed Systems, 2nd edn. Wiley, Chichester (2008) ISBN: 978-0-470-06852-6
8. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical One-Way Functions. Science 297, 2026–2030 (2002)
9. Pappu, R.: Physical One-Way Functions, PhD Thesis, MIT
10. Gassend, B., Clarke, D.E., van Dijk, M., Devadas, S.: Silicon physical random functions. In: ACM Conference on Computer and Communications Security 2002, pp. 148–160 (2002)
11. Gassend, B., Lim, D., Clarke, D., van Dijk, M., Devadas, S.: Identification and authentication of integrated circuits. Concurrency and Computation: Practice & Experience 16(11), 1077–1098 (2004)
12. Tuyls, P., Skoric, B.: Strong Authentication with Physical Unclonable Functions. In: Petkovic, M., Jonker, W. (eds.) Security, Privacy and Trust in Modern Data Management. Springer, Heidelberg (2007)
13. Edward Suh, G., Devadas, S.: Physical Unclonable Functions for Device Authentication and Secret Key Generation. In: DAC 2007, pp. 9–14 (2007)
14. Gassend, B., van Dijk, M., Clarke, D.E., Torlak, E., Tuyls, P., Devadas, S.: Controlled physical random functions and applications. ACM Trans. Inf. Syst. Secur. 10(4) (2008)
15. Rührmair, U.: Oblivious Transfer Based on Physical Unclonable Functions. In: Acquisti, A., Smith, S.W., Sadeghi, A.-R. (eds.) TRUST 2010. LNCS, vol. 6101, pp. 430–440. Springer, Heidelberg (2010)
16. Rührmair, U.: SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions. Cryptology ePrint Archive, Report 2009/255 (2009)
17. Rührmair, U., Chen, Q., Lugli, P., Schlichtmann, U., Stutzmann, M., Csaba, G.: Towards Electrical, Integrated Implementations of SIMPL Systems. Cryptology ePrint Archive, Report 2009/278 (2009)
18. Chen, Q., Csaba, G., Ju, X., Natarajan, S.B., Lugli, P., Stutzmann, M., Schlichtmann, U., Rührmair, U.: Analog Circuits for Physical Cryptography. In: 12th International Symposium on Integrated Circuits (ISIC 2009), Singapore, December 14-16 (2009)
19. Rührmair, U., Chen, Q., Stutzmann, M., Lugli, P., Schlichtmann, U., Csaba, G.: Towards electrical, integrated implementations of SIMPL systems. In: Samarati, P., Tunstall, M., Posegga, J., Markantonakis, K., Sauveron, D. (eds.) WISTP 2010. LNCS, vol. 6033, pp. 277–292. Springer, Heidelberg (2010)
20. Rührmair, U.: SIMPL systems, or: Can we design cryptographic hardware without secret key information? In: Černá, I., Gyimóthy, T., Hromkovič, J., Jefferey, K., Králović, R., Vukolić, M., Wolf, S. (eds.) SOFSEM 2011. LNCS, vol. 6543, pp. 26–45. Springer, Heidelberg (2011)

21. Chen, Q., Csaba, G., Lugli, P., Schlichtmann, U., Stutzmann, M., Rührmair, U.: Circuit-based approaches to SIMPL systems. Journal of Circuits, Systems, and Computers, JCSC 20, 107–123 (2011), doi:10.1142/S0218126611007098
22. Chua, L.O., Roska, T., Kozek, T., Zarandy, A.: CNN Universal Chips crank up the computing power. IEEE Circuits and Devices Magazine 12(4), 18–28 (1996)
23. Roska, T.: Cellular Wave Computers for Nano-Tera-Scale Technology — beyond spatial-temporal logic in million processor devices. Electronics Letters 43(8) (April 12, 2007)
24. Beckmann, N., Potkonjak, M.: Hardware-based public-key cryptography with public physically unclonable functions. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 206–220. Springer, Heidelberg (2009)
25. Koushanfar, F., Potkonjak, M.: CAD-based Security, Cryptography, and Digital Rights Management. In: DAC 2007, pp. 268–269 (2007)
26. Majzoobi, M., Elnably, A., Koushanfar, F.: FPGA Time-Bounded Unclonable Authentication. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 1–16. Springer, Heidelberg (2010)
27. Rührmair, U., Sölter, J., Sehnke, F.: On the Foundations of Physical Unclonable Functions. IACR Cryptology E-print Archive, Report No. 227/2009 (2009)
28. Rührmair, U., Busch, H., Katzenbeisser, S.: Strong PUFs: Models, Constructions and Security Proofs. In: Sadeghi, A.-R., Naccache, D. (eds.) Towards Hardware Intrinsic Security: Foundation and Practice. Springer, Heidelberg (2010)
29. Gassend, B.: Physical Random Functions, MSc Thesis, MIT (2003)
30. Guajardo, J., Kumar, S.S., Schrijen, G.-J., Tuyls, P.: FPGA Intrinsic PUFs and Their Use for IP Protection. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 63–80. Springer, Heidelberg (2007)
31. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling Attacks on Physical Unclonable Functions. In: 17th ACM Conference on Computer and Communications Security (2010); Previous versions available from Cryptology ePrint Archive, Report 251/2010
32. Feynman, R.P.: Simulating physics with computers. International Journal of Theoretical Physics (1982)
33. Naccache, D., Raihi David, M.: Procede de Generation de Signature Numeriques de Messages. French Patent, Publication Number 2733378, National Registration Number 9504753 (1995)
34. Naccache, D.: Method for the Generation of Electronic Signatures, in particular for Smart Cards. US Patent Number 5,910,989 (1999)
35. Blum, M.: Coin flipping by telephone. In: Proc. IEEE Spring COMPCOM, pp. 133–137. IEEE, Los Alamitos (1982)
36. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of The Nineteenth Annual ACM Symposium on Theory of Computing (1987)
37. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In: 27th Annual Symposium on the Foundations of Computer Science, FOCS (1986)
38. Halevi, S., Krawczyk, H.: MMH: Software Message Authentication in the Gbit/Second Rates. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 172–189. Springer, Heidelberg (1997)
39. DeJean, G., Kirovski, D.: RF-DNA: Radio-Frequency Certificates of Authenticity. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 346–363. Springer, Heidelberg (2007)
40. Kariakin, Y.: Authentication of Articles. Patent Writing, WO/1997/024699 (1995), available from `http://www.wipo.int/pctdb/en/wo.jsp?wo=1997024699`

41. Vijaywargi, D., Lewis, D., Kirovski, D.: Optical DNA. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 222–229. Springer, Heidelberg (2009)
42. Hammouri, G., Dana, A., Sunar, B.: CDs Have Fingerprints Too. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 348–362. Springer, Heidelberg (2009)
43. Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Transactions on Information Theory IT-22, 644–654 (1976)
44. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP using any one-way function. Journal of Cryptology 11(2), 87–108 (1998)
45. Savvides, G.: Interactive Hashing and reductions between Oblivious Transfer variants. PhD thesis, McGill University, Montreal (2007)
46. Haitner, I., Reingold, O.: A new interactive hashing theorem. In: IEEE Conference on Computational Complexity (2007)
47. Blum, M.: Coin flipping by telephone. In: Gersho, A. (ed.) Advances in Cryptography, pp. 11–15. University of California, Santa Barbara (1982)
48. Goldreich, O.: The Foundations of Cryptography, vol. 1. Cambridge University Press, Cambridge (2001)
49. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the Association for Computing Machinery 38(3), 691–729 (1991)
50. Brassard, G., Chaum, D., Crepeau, C.: Minimum disclosure proofs of knowledge. JCSS 37, 156–189 (1988)
51. Kilian, J.: Founding cryptography on oblivious transfer. In: Proc. 20th ACM Symposium on Theory of Computing, pp. 20–31. ACM Press, Chicago (1988)
52. Yao, A.C.-C.: Classical physics and the Church-Turing Thesis. Journal of the ACM 50(1), 100–105 (2003)
53. Aaronson, S.: NP-complete Problems and Physical Reality. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 026 (2005)
54. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5), 1484–1509 (1997)
55. Csaba, G., Ju, X., Ma, Z., Chen, Q., Porod, W., Schmidhuber, J., Schlichtmann, U., Lugli, P., Rührmair, U.: Application of Mismatched Cellular Nonlinear Networks for Physical Cryptography. In: IEEE CNNA (2010)
56. Lim, D.: Extracting Secret Keys from Integrated Circuits. M.Sc. Thesis, MIT (2004)
57. Suh, G.E., O'Donnell, C.W., Sachdev, I., Devadas, S.: Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions. In: Proc. 32nd ISCA, New York (2005)
58. Yu, M. D.M., Devadas, S.: Secure and Robust Error Correction for Physical Unclonable Functions. IEEE Design & Test of Computers 27(1), 48–65 (2010)
59. Armknecht, F., Maes, R., Sadeghi, A.-R., Sunar, B., Tuyls, P.: Memory Leakage-Resilient Encryption Based on Physically Unclonable Functions. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 685–702. Springer, Heidelberg (2009)
60. Rührmair, U., Weiershäuser, A., Urban, S., Hilgers, C., Finley, J.: Secure Integrated Optical Physical Unclonable Functions (2010) (in preparation)
61. Lipson, S.G.: Optical Physics, 3rd edn. Cambridge University Press, Cambridge (1995) ISBN 0-5214-3631-1
62. Demtröder, W.: Experimentalphysik 2: Elektrizität und Optik. Springer, Heidelberg (2004) ISBN-10: 3540202102
63. Zhou, D., Mawst, L.J.: Two-dimensional phase-locked antiguided vertical-cavity surface-emitting laser arrays. Applied Physics Letters (2000)

Part IV

Appendix

Appendix A

Publications Employed as Chapters and Contributions of the Candidate

We list the publications that have directly been used as chapters of this cumulative thesis below. As required by the doctoral dissertation statutes of the TU München, we also detail the contributions of the candidate to each of these publications.

- CHAPTER 1 was solely written by the candidate. Parts of it will be published as an invited book chapter that will appear in 2014/15 as

  - U. Rührmair: *Disorder-based Security Hardware: An Overview*. In: Secure System Design and Trustable Computing, Chip Hong Chang and Miodrag Potkonjak (Ed.), Springer 2014/15 (to appear).

- CHAPTER 2 employs the publication

  - U. Rührmair, D.E. Holcomb: *PUFs at a Glance*. Design, Automation & Test in Europe (DATE), pp. 1-6, 2014.

  The candidate conceptualized, initiated and led the paper. He is responsible for its outline and the distinction between Weak and Strong PUFs. He wrote the majority of the paper, including Sections I, II.A, III, and the larger part of Section IV.

- CHAPTER 3 uses the publication

  - U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions*. ACM Conference on Computer and Communications Security (ACM CCS), pp. 237-249, 2010.

The candidate conceptualized, initiated and led this paper. He was responsible for the writing process and the paper's organization, and wrote the largest part of it. Among other things, he contributed the distinction between Weak, Strong and Controlled PUFs that is essential to the papers outline, selected the PUFs and the architectures to be machine learned, and conceptualized the machine learning experiments and the other analyses in the paper.

Two other, related publications, which are explicitly not used in this cumulative thesis, are:

- 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 Transactions on Information Forensics and Security, Vol. 8(11), pp. 1876-1891, 2013.
- U. Rührmair, J. Sölter: *PUF Modeling Attacks: An Introduction and Overview.* Design, Automation & Test in Europe (DATE), pp. 1-6, 2014.

- CHAPTER 4 employs the publication

  - U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, W. Burleson: *Efficient Power and Timing Side Channels for Physical Unclonable Functions*. 16th Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2014. Lecture Notes in Computer Science, Springer, 2014 (to appear).

  The candidate conceptualized, initiated and led this paper and conceptualized the conducted experiments. In particular, he contributed the central idea of combining machine learning and side channel attacks, and the concrete approach of applying side channels to learn information about the single outputs of Arbiter PUFs within an XOR-based Arbiter PUF structure. He was responsible for the writing process and the paper organization and wrote a large part of the paper.

  Two related publications, which are explicitly not used in this cumulative thesis, are

  - U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, W. Burleson: *Power and Timing Side Channels for PUFs and their Efficient Exploitation.* IACR Cryptology ePrint Archive, Report 2013/851, 2013.
  - A. Mahmoud, U. Rührmair, M. Majzoobi, F. Koushanfar: *Combined Modeling and Side Channel Attacks on Strong PUFs.* IACR Cryptology ePrint Archive, Report 2013/632, 2013.

- CHAPTER 5 employs the publication

  - U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, G. Csaba: *Applications of High-Capacity Crossbar Memories in Cryptography.* IEEE Transactions on Nanotechnology 10(3), pp. 489-498, 2011.

The candidate conceptualized and led this paper, organized its content, and directed and initiated the simulation and hardware experiments. In particular, the concept of a SHIC PUF (i.e., a PUF with a very high information content and intrinsically slow read-out speeds that offers information-theoretic security), which is at the heart of the paper, is due to him. He also contributed to the development of the ideas of using crossbars as SHIC PUFs, for example to the realization of the intrinsically slow read-out speed and other features, or set the security and design goals for the examined architectures.

- CHAPTER 6 utilizes the work

  – U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, M. Stutzmann: *Security Applications of Diodes with Unique Current-Voltage Characteristics.* 14th International Conference on Financial Cryptography and Data Security (FC), 2010. Lecture Notes in Computer Science, Volume 6052, pp. 328-335, Springer, 2010.

  The candidate conceptualized, initiated and led the paper and directed the hardware experiments and the necessary research. He wrote the largest part of the paper. The idea of using ALILE diodes as the three primitives discussed in the paper (Weak PUFs/POKs, COAs, and SHIC PUFs) is due to him, and also the explicit distinction of these primitives.

  A related piece of work at a leading venue, which is explicitly not used in this cumulative thesis, is

  – C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, M. Stutzmann: *Random p-n-junctions for physical cryptography.* Applied Physics Letters 96, 172103, 2010.

- CHAPTER 7 uses the paper

  – U. Rührmair: *SIMPL Systems, Or: Can We Design Cryptographic Hardware without Secret Key Information?* 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), 2011. Lecture Notes in Computer Science, Volume 6543, pp. 26-45, Springer, 2011.

  which is a single author paper of the candidate.

- CHAPTER 8 employs the work

  – U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems.* 4th Workshop in Information Security Theory and Practice (WISTP), 2010. Lecture Notes in Computer Science, Volume 6033, pp. 277 - 292, Springer, 2010.

The candidate conceptualized, initiated and led this paper, organized its content, and directed the simulation and hardware experiments. He invented the concept of a SIMPL system and contributed to the development of the ideas of using SRAM cells and analog circuits as SIMPL systems. The specification of SIMPL systems and the idea of using feedback loops in order to amplify the absolute time difference in SIMPL systems is due to him. He wrote the largest part of the paper.

- CHAPTER 9 utilizes the work

  – U. Rührmair: *SIMPL Systems as a Keyless Cryptographic and Security Primitive.* Cryptography and Security 2012. Lecture Notes in Computer Science, Vol. 6805, pp. 329-354, Springer, 2012.

  which is again a single author paper of the candidate.

According to Google scholar, the publications used directly as chapters together have been quoted over 220 times (status: June 1, 2014).

Appendix B

Complete Publication List

Not all publications of the candidate have been used in this thesis. A complete list is given below for completeness (status: June 1, 2014), ordered chronologically and by publication medium.

Journals:

1. C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, M. Stutzmann: *Random p-n-junctions for physical cryptography.* Applied Physics Letters 96, 172103, 2010. [59]

2. U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, G. Csaba: *Applications of High-Capacity Crossbar Memories in Cryptography.* IEEE Transactions on Nanotechnology 10(3), pp. 489-498, 2011. [129]

3. H. Langhuth, S. Frederic, M. Kaniber, J. Finley, U. Rührmair: *Strong Photoluminescence Enhancement from Colloidal Quantum Dot Near Silver Nano-Island Films.* Journal of Fluorescence 21(2), pp. 539-543, 2011. [72]

4. Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, M. Stutzmann, U. Rührmair: *Circuit-based Approaches to SIMPL Systems.* Journal of Circuits, Systems and Computers 20(1), pp. 107-123, 2011. [21]

5. P. Lugli, A. Mahmoud, M. Algasinger, M. Stutzmann, G. Csaba, U. Rührmair: *Physical Unclonable Functions based on Crossbar Arrays for Cryptographic Applications.* International Journal of Circuit Theory and Applications 41(6), pp. 619-633, 2013. [81]

6. U. Rührmair, M. van Dijk: *On the Practical Use of Physical Unclonable Functions in Oblivious Transfer and Bit Commitment Protocols.* Journal of Cryptographic Engineering 3(1), pp. 17-28, 2013. [125]

7. 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 Transactions on Information Forensics and Security 8(11), pp. 1876-1891, 2013. [138]

Conferences:

8. Q. Chen, G. Csaba, X. Ju, S.B. Natarajan, P. Lugli, M. Stutzmann, U. Schlichtmann, U. Rührmair: *Analog Circuits for Physical Cryptography.* 12th International Symposium on Integrated Circuits (ISIC), pp. 121-124, 2009. [18]
(This paper received the Best Paper Award.)

9. M. Steinebach, S. Zmudsinski, S. Katzenbeisser, U. Rührmair: *Audio watermarking forensics: detecting malicious re-embedding.* IS&T/SPIE Electronic Imaging Conference – Media Forensics and Security XII, 2010. [169]

10. U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, M. Stutzmann: *Security Applications of Diodes with Unique Current-Voltage Characteristics.* 14th International Conference on Financial Cryptography and Data Security (FC), 2010. Lecture Notes in Computer Science, Volume 6052, pp. 328-335, Springer, 2010. [130]

11. G. Csaba, X. Ju, Z. Ma, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, U. Rührmair: *Application of Mismatched Cellular Nonlinear Networks for Physical Cryptography.* 12th IEEE International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), pp. 1-6, 2010. [30]

12. U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems.* 8th Workshop in Information Security Theory and Practice (WISTP), 2010. Lecture Notes in Computer Science, Volume 6033, pp. 277 - 292, Springer, 2010. [121]

13. U. Rührmair, S. Katzenbeisser, M. Steinebach, S. Zmudzinski: *Watermark-Based Authentication and Key Exchange in Teleconferencing Systems.* 11th Conference on Communications and Multimedia Security (CMS), 2010. Lecture Notes in Computer Science, Volume 6109, pp. 75 - 80, Springer, 2010. [131]

14. U. Rührmair: *Oblivious Transfer based on Physical Unclonable Functions (Extended Abstract).* 3rd International Conference on Trust and Trustworthy Computing (TRUST), 2010. Lecture Notes in Computer Science, Volume 6101, pp. 430 - 440, Springer, 2010. [112]

15. F. Sehnke, C. Osendorfer, J. Sölter, J. Schmidhuber, U. Rührmair: *Policy Gradients for Cryptanalysis.* 20th International Conference on Artificial Neural Networks (ICANN), 2010. Lecture Notes in Computer Science, Volume 6354, pp. 168-177, Springer, 2010. [160]

16. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions.* 17th ACM Conference on Computer and Communications Security (ACM CCS), pp. 237-249, 2010. [134]

17. U. Rührmair: *SIMPL Systems, Or: Can We Design Cryptographic Hardware without Secret Key Information?* 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), 2011. Lecture Notes in Computer Science, Volume 6543, pp. 26-45, Springer, 2011. [115]

18. U. Rührmair, C. Jaeger, M. Algasinger: *An Attack on PUF-based Session Key Exchange, and a Hardware-based Countermeasure: Erasable PUFs.* 15th International Conference on Financial Cryptography and Data Security (FC), 2011. Lecture Notes in Computer Science, Volume 7035, pp. 190-204, 2012. [128]

19. Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Rührmair: *The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions*. 4th IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 134-141, 2011. [19]

20. Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Rührmair: *Characterization of the Bistable Ring PUF.* Design, Automation & Test in Europe (DATE), pp. 1459-1462, 2012. [20]

21. U. Rührmair, M. van Dijk: *Practical Security Analysis of PUF-based Two-Player Protocols*. 14th Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2012. Lecture Notes in Computer Science, Volume 7428, pp. 251-267, Springer, 2012. [123]

22. U. Rührmair, M. van Dijk: *PUFs in Security Protocols: Attack Models and Security Evaluations.* 34th IEEE Symposium on Security and Privacy (Oakland), pp. 286-300, 2013. [124]

23. M. van Dijk, U. Rührmair: *Protocol Attacks on Advanced PUF Protocols and Countermeasures.* Design, Automation & Test in Europe (DATE), pp. 1-6, 2014. [40]

24. U. Rührmair, D.E. Holcomb: *PUFs at a Glance.* Design, Automation & Test in Europe (DATE), pp. 1-6, 2014. [127]

25. U. Rührmair, U. Schlichtmann, W. Burleson: *Special Session: How Secure are PUFs Really? On the Reach and Limits of Recent PUF Attacks.* Design, Automation & Test in Europe (DATE), pp. 1-6, 2014. [132]

26. U. Rührmair, J. Sölter: *PUF Modeling Attacks: An Introduction and Overview*. Design, Automation & Test in Europe (DATE), pp. 1-6, 2014. [135]

27. U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, W. Burleson: *Efficient Power and Timing Side Channels for Physical Unclonable Functions*. 16th Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2014. Lecture Notes in Computer Science, Springer, 2014 (to appear). [140]

Invited Book Chapters:

28. U. Rührmair, H. Busch, S. Katzenbeisser: *Strong PUFs: Models, Constructions and Security Proofs.* In: Towards Hardware Intrinsic Security, A.-R. Sadeghi, P. Tuyls (Ed.), pp. 79-96, Springer, 2010. [119]

29. U. Rührmair, S. Devadas, F. Koushanfar: *Security based on Physical Unclonability and Disorder.* In: Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang (Ed.), pp. 65-102, Springer, 2012. [122]

30. U. Rührmair: *SIMPL Systems as a Keyless Cryptographic and Security Primitive.* Cryptography and Security 2012. Lecture Notes in Computer Science, Vol. 6805, pp. 329-354, Springer, 2012. [116]

31. U. Rührmair: *Disorder-based Security Hardware: An Overview.* In: Secure System Design and Trustable Computing, Chip Hong Chang and Miodrag Potkonjak (Ed.), Springer 2014/15 (to appear). [118]

Preprints:

32. U. Rührmair: *SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions.* Cryptology ePrint Archive, Report 2009/255, 2009. [111]

33. U. Rührmair, J. Sölter, F. Sehnke: *On the Foundations of Physical Unclonable Functions.* Cryptology ePrint Archive, Report 2009/277, 2009. [136]

34. U. Rührmair, Q. Chen, P. Lugli, M. Stutzmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems*. Cryptology ePrint Archive, Report 2009/278, 2009. [120]

35. G. Csaba, X. Ju, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, U. Rührmair: *On-Chip Electric Waves: An Analog Circuit Approach to Physical Unclonable Functions*. IACR Cryptology ePrint Archive, Report 2009/246, 2009. [29]

36. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions.* IACR Cryptology ePrint Archive, Report 2010/251, 2010. [133]

37. U. Rührmair: *Physical Turing Machines and the Formalization of Physical Cryptography.* IACR Cryptology ePrint Archive, Report 2011/188, 2011. [113]

38. U. Rührmair: *SIMPL Systems as a Keyless Cryptographic and Security Primitive.* IACR Cryptology ePrint Archive, Report 2011/189, 2011. [114]

39. M. van Dijk, U. Rührmair: *Physical Unclonable Functions in Cryptographic Protocols: Security Proofs and Impossibility Results.* IACR Cryptology ePrint Archive, Report 2012/228, 2012. [39]

40. 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.* IACR Cryptology ePrint Archive, Report 2013/112, 2013. [137]

41. U. Rührmair, C. Hilgers, S. Urban, A. Weiershäuser, E. Dinter, B. Forster, C. Jirauschek: *Optical PUFs Reloaded.* IACR Cryptology ePrint Archive, Report 2013/215, 2013. [126]

42. A. Mahmoud, U. Rührmair, M. Majzoobi, F. Koushanfar: *Combined Modeling and Side Channel Attacks on Strong PUFs.* IACR Cryptology ePrint Archive, Report 2013/632, 2013. [82]

43. U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, W. Burleson: *Power and Timing Side Channels for PUFs and their Efficient Exploitation.* IACR Cryptology ePrint Archive, Report 2013/851, 2013. [139]

According to Google scholar, the above publications have been quoted over 600 times altogether (status: June 1, 2014).

Bibliography

[1] D. Adams: *The hitchhiker's guide to the galaxy*. Pan Books, 1979.

[2] R.J. Anderson: *Security Engineering: A Guide to Building Dependable Distributed Systems*. Second Edition. Wiley, 2008.

[3] F. Armknecht, R. Maes, A.-R. Sadeghi, B. Sunar, P. Tuyls: *Memory Leakage-Resilient Encryption Based on Physically Unclonable Functions*. ASIACRYPT 2009, pp. 685-702, 2009.

[4] F. Armknecht, R. Maes, Ahmad-Reza Sadeghi, F.-X. Standaert, C. Wachsmann: *A Formal Foundation for the Security Features of Physical Functions*. IEEE Symposium on Security and Privacy 2011, pp. 397-412, 2011.

[5] Y. Aumann, Y. Z. Ding, M. O. Rabin: *Everlasting security in the bounded storage model*. IEEE Transactions on Information Theory, Vol. 48(6), pp. 1668-1680, 2002.

[6] Y. Aumann, M. O. Rabin: *Information theoretically secure communication in the limited storage space model*. CRYPTO 1999, pp. 65-79, 1999.

[7] D.W. Bauder: *An anti-counterfeiting concept for currency systems.* Sandia National Labs, Albuquerque, NM, Tech. Rep. PTK-11990, 1983.

[8] N. Beckmann, M. Potkonjak: *Hardware-based public-key cryptography with public physically unclonable functions*. Information Hiding 2009, pp. 206-220, 2009.

[9] C.H. Bennett, G. Brassard: *Quantum cryptography: Public key distribution and coin tossing.* IEEE International Conference on Computers, Systems and Signal Processing, Vol. 175(150), p. 8, 1984.

[10] D. J. Bernstein, J. Buchmann, E. Dahmen (Ed.): *Post-Quantum Cryptography*. Springer, 2009. ISBN 978-3-540-88701-0.

[11] M. Bhargava, K. Mai: *A High Reliability PUF Using Hot Carrier Injection Based Response Reinforcement*. CHES 2013, pp. 90-106, 2013.

[12] C. Böhm, M. Hofer: *Physical Unclonable Functions in Theory and Practice*. ISBN 978-1-4614-5040-5. Springer, 2013.

[13] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, P. Tuyls: *Efficient Helper Data Key Extractor on FPGAs*. CHES 2008, pp. 181-197, 2008.

[14] J.D.R. Buchanan, R. Cowburn, A. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. Allwood, M. Bryan: *Fingerprinting documents and packaging*. Nature, Vol. 436(7050), p. 475, 2005.

[15] J. Buchmann, A. May, U. Vollmer: *Perspectives for cryptographic long-term security*. Communications of the ACM, Vol. 49(9), pp. 50-55, 2006.

[16] C. Bruzska, M. Fischlin, H. Schröder, S. Katzenbeisser: *Physical Unclonable Functions in the Universal Composition Framework.* CRYPTO 2011, pp. 51-70, 2011.

[17] C. Cachin, U. Maurer: *Unconditional Security Against Memory-Bounded Adversaries*. CRYPTO 1997, pp. 292-306, 1997.

[18] Q. Chen, G. Csaba, X. Ju, S.B. Natarajan, P. Lugli, M. Stutzmann, U. Schlichtmann, U. Rührmair: *Analog Circuits for Physical Cryptography.* 12th International Symposium on Integrated Circuits (ISIC), pp. 121-124, 2009.

[19] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Rührmair: *The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions*. 4th IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 134-141, 2011.

[20] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Rührmair: *Characterization of the Bistable Ring PUF.* 17th Design, Automation and Test in Europe (DATE), pp. 1459-1462, 2012.

[21] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, M. Stutzmann, U. Rührmair: *Circuit-based Approaches to SIMPL Systems.* Journal of Circuits, Systems and Computers 20(1), pp. 107-123, 2011.

[22] Y. Chen, M.K. Mihcak, D. Kirovski: *Certifying authenticity via fiber-infused paper*. SIGecom Exchanges, Vol. 5(3), pp. 29-37, 2005.

[23] M.C. Chu , L.L. Cheng, L.M. Cheng: *A novel magnetic card protection system*. European Convention on Security and Detection, pp. 207-211, 1995.

[24] W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J.A. Halderman, E.W. Felten: *Fingerprinting Blank Paper Using Commodity Scanners*. IEEE Symposium on Security and Privacy 2009, pp. 301-314, 2009.

[25] Clay Mathematical Institure Milllenium Prize on P vs NP. Downloaded from http://www.claymath.org/millennium/P_vs_NP, August 2012.

[26] C.T. Clelland, V. Risca, C. Bancroft. *Hiding messages in DNA microdots.* Nature, Vol. 399(6736), pp. 533-534, 1999.

[27] I.J. Cox, M. L. Miller, J.A. Bloon, J. Fridrich, T. Kalker: *Digital Watermarking and Steganography*. Morgan Kaufmann, 2008.

[28] C. Crepeau: *Efficient cryptographic protocols based on noisy channels*. EUROCRYPT 1997, pp. 306-317, 1997.

[29] G. Csaba, X. Ju, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, U. Rührmair: *On-Chip Electric Waves: An Analog Circuit Approach to Physical Unclonable Functions*. IACR Cryptology ePrint Archive, Report 2009/246, 2009.

[30] G. Csaba, X. Ju, Z. Ma, Q. Chen, W. Porod, J. Schmidhuber, U. Schlichtmann, P. Lugli, U. Rührmair: *Application of Mismatched Cellular Nonlinear Networks for Physical Cryptography*. IEEE CNNA 2010, pp. 1-6, 2010.

[31] W.E. Cobb, E.D. Laspe, R.O. Baldwin, M.A. Temple, Y.C. Kim: *Intrinsic Physical-Layer Authentication of Integrated Circuits*. IEEE Transactions on Information Forensics and Security, Vol. 7(1), pp. 14-24, 2012.

[32] I. Damgard, A. Scafuro: *Unconditionally Secure and Universally Composable Commitments from Physical Assumptions*. ASIACRYPT 2013, pp. 100-119, 2013.

[33] G. DeJean, D. Kirovski: *RF-DNA: Radio-Frequency Certificates of Authenticity*. CHES 2007, pp. 346-363, 2007.

[34] J. Delvaux, I. Verbauwhede: *Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise*. HOST 2013, pp. 137-142, 2013.

[35] J. Delvaux, I. Verbauwhede: *Fault Injection Modeling Attacks on 65nm Arbiter and RO Sum PUFs via Environmental Changes*. IACR Cryptology ePrint Archive, Report 2013/619, 2013.

[36] D. Deutsch: *Quantum theory, the Church-Turing principle and the universal quantum computer*. Proceedings of the Royal Society of London; Series A, Mathematical and Physical Sciences. Vol. 400(1818), pp. 97-117, 1985.

[37] D. Deutsch, A. Ekert, R. Luppachini: *Machines, Logic and Quantum Physics*, arXiv:math/9911150v1, 1999. Downloaded from http://arxiv.org/abs/math.LO/9911150, August 2012.

[38] M. van Dijk: *System and method of reliable forward secret key sharing with physical random functions.* US Patent No. 7,653,197, October 2004.

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

[40] M. van Dijk, U. Rührmair: *Protocol Attacks on Advanced PUF Protocols and Countermeasures.* 17th Design, Automation and Test in Europe (DATE), pp. 1-6, 2014.

[41] Y. Z. Ding: *Provable Everlasting Security in the Bounded Storage Model*. PhD Thesis, Harvard University, Cambridge (Massachusetts), USA, 2001.

[42] R. Feynman: *Simulating Physics with Computers.* International Journal of Theoretical Physics, Vol. 21, (6&7), pp. 467-488, 1982.

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

[44] B. Gassend, D.E. Clarke, M. van Dijk, S. Devadas: *Silicon physical random functions*. ACM CCS 2002, pp. 148-160, 2002.

[45] B. Gassend, D.E. Clarke, M. van Dijk, S. Devadas: *Controlled Physical Random Functions*. ACSAC 2002, pp. 149-160, 2002.

[46] B. Gassend, M. van Dijk, D.E. Clarke, E. Torlak, S. Devadas, P. Tuyls: *Controlled physical random functions and applications*. ACM Transactions on Information and System Security, Vol. 10(4), 2008.

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

[48] A. Gehani, T. LaBean, J. Reif: *DNA-based cryptography.* Aspects of Molecular Computing, pp. 167-188, Springer, 2004.

[49] R.N. Goldman: *Non-counterfeitable document system*. US-Patent 4,423,415. Publication date: 1983. Priority date: 1980.

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

[51] P. Gutmann, *Secure deletion of data from magnetic and solid-state memory*. USENIX Security Symposium, 1996.

[52] T. Haist, H.J. Tiziani: *Optical detection of random features for high security applications.*. Optics communications, Vol. 147.1, pp. 173-179, 1998.

[53] G. Hammouri, A. Dana, B. Sunar: *CDs Have Fingerprints Too*. CHES 2009, pp. 348-362, 2009.

[54] C. Helfmeier, C. Boit, D. Nedospasov, J.-P. Seifert: *Cloning Physically Unclonable Functions*. HOST 2013, pp. 1-6, 2013.

[55] M. Hofer, C. Böhm: *An Alternative to Error Correction for SRAM-Like PUFs*. CHES 2010, pp. 335-350, 2010.

[56] D.E. Holcomb: *PUFs at a Glance*. Talk, Hot Topic Session 12.2, Design, Automation & Test in Europe (DATE) 2014, 2014.

[57] D.E. Holcomb, W.P. Burleson, K. Fu: *Initial SRAM state as a fingerprint and source of true random numbers for RFID tags*. Conference on RFID Security, 2007.

[58] D.E. Holcomb, W.P. Burleson, K. Fu: *Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers*. IEEE Transactions on Computers, Vol. 58(9), pp. 1198-1210, 2009.

[59] C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, M. Stutzmann: *Random pn-junctions for physical cryptography*. Applied Physics Letters 96, 172103, 2010.

[60] Y. Kariakin: *Authentication of articles*. Patent writing, WO/1997/024699, 1995. Available from http://www.wipo.int/pctdb/en/wo.jsp?wo=1997024699

[61] S. Katzenbeisser, Ü. Kocabas, V. van der Leest, A.-R. Sadeghi, G.-J. Schrijen, H. Schröder, C. Wachsmann: *Recyclable PUFs: Logically Reconfigurable PUFs*. CHES 2011, pp. 374-389, 2011.

[62] S. Katzenbeisser, Ü. Koçabas, V. Rozic, A.-R. Sadeghi, I. Verbauwhede, C. Wachsmann: *PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon*. CHES 2012, pp. 283-301, 2012.

[63] A. Kerckhoffs: *La cryptographie militaire*. Journal des sciences militaires, Vol. IX, pp. 5-38, 1883.

[64] J. Kilian: *Founding cryptography on oblivious transfer*. STOC 1988, pp. 20-31, 1988.

[65] D. Kirovski: *Toward an automated verification of certificates of authenticity*. EC 2004, pp. 160-169, 2004.

[66] D. Kirovski: *Point Compression for Certificates of Authenticity*. Data Compression Conference 2004, p. 545, 2004.

[67] D. Kirovski: *Anti-counterfeiting: Mixing the Physical and the Digital World*. In: Towards Hardware-Intrinsic Security, A.-R. Sadeghi, D. Naccache (Eds.), pp. 223-233, Springer, 2010.

[68] D. Kirovski, *personal communication*, Dagstuhl 2008.

[69] A.R. Krishna, S. Narasimhan, X. Wang, S. Bhunia: *MECCA: A Robust Low-Overhead PUF Using Embedded Memory Array*. CHES 2011, pp. 407-420, 2011.

[70] R. Kumar, W. Burleson: *Personal communication*. 2014.

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

[72] H. Langhuth, S. Frederic, M. Kaniber, J. Finley, U. Rührmair: *Strong Photoluminescence Enhancement from Colloidal Quantum Dot Near Silver Nano-Island Films*. Journal of Fluorescence, Vol. 21(2), pp. 539-543, 2010.

[73] J.W. Lee, D. Lim, B. Gassend, G.E. Suh, M. van Dijk, S. Devadas: *A technique to build a secret key in integrated circuits with identification and authentication applications.* IEEE VLSI Circuits Symposium, pp. 176-179, 2004.

[74] V. van der Leest, B. Preneel, E. van der Sluis: *Soft Decision Error Correction for Compact Memory-Based PUFs Using a Single Enrollment*. CHES 2012, pp. 268-282, 2012.

[75] A. Leier, C. Richter, W. Banzhaf, H. Rauhe: *Cryptography with DNA binary strands*. BioSystems, Vol. 57(1), pp. 13-22, 2000.

[76] A.K. Lenstra, E.R. Verheul: *Selecting Cryptographic Key Sizes*. Journal of Cryptology, Vol. 14(4), pp. 255-293, 2001.

[77] D. Lim: *Extracting Secret Keys from Integrated Circuits*. MSc Thesis, MIT, 2004.

[78] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. van Dijk, S. Devadas: *Extracting secret keys from integrated circuits*. IEEE Transactions on VLSI Systems, Vol. 13(10), pp. 1200-1205, 2005.

[79] G. Lindstrom, G. Schullstrom: *Verifiable identification document*. US-Patent 3,636,318. Publication date: 1972. Priority date: 1968.

[80] K. Lofstrom, W.R. Daasch, D. Taylor: *IC identification circuit using device mismatch.* ISSCC 2000, pp. 372-373, 2000.

[81] P. Lugli, A. Mahmoud, M. Algasinger, M. Stutzmann, G. Csaba, U. Rührmair: *Physical Unclonable Functions based on Crossbar Arrays for Cryptographic Applications.* International Journal of Circuit Theory and Applications 41(6), pp. 619-633, 2013.

[82] A. Mahmoud, U. Rührmair, M. Majzoobi, F. Koushanfar: *Combined Modeling and Side Channel Attacks on Strong PUFs.* IACR Cryptology ePrint Archive, Report 2013/632, 2013.

[83] A. Maiti, I. Kim, P. Schaumont: *A Robust Physical Unclonable Function With Enhanced Challenge-Response Set*. IEEE Transactions on Information Forensics and Security, Vol 7(1), pp. 333-345, 2012.

[84] A. Maiti, P. Schaumont: *Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive*. Journal of Cryptology, Vol. 24(2), pp. 375-397, 2011.

[85] R. Maes: *An Accurate Probabilistic Reliability Model for Silicon PUFs*. CHES 2013, pp. 73-89, 2013.

[86] R. Maes, A. Van Herrewege, I. Verbauwhede: *PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator*. CHES 2012, pp. 302-319, 2012.

[87] R. Maes, P. Tuyls, I. Verbauwhede: *Low-Overhead Implementation of a Soft Decision Helper Data Algorithm for SRAM PUFs*. CHES2009, pp. 332-347, 2009.

[88] R. Maes, I. Verbauwhede: *Physically unclonable functions: A study on the state of the art and future research directions*. In: Towards Hardware-Intrinsic Security, A.-R. Sadeghi, D. Naccache (Eds.), pp. 3-37. Springer, 2010.

[89] M. Majzoobi, F. Koushanfar, M. Potkonjak: *Lightweight Secure PUFs.* IC-CAD 2008, pp. 607-673, 2008.

[90] M. Majzoobi, F. Koushanfar, M. Potkonjak: *Testing techniques for hardware security.* ITC 2008, pp. 1-10, 2008.

[91] U. Maurer: *Conditionally-perfect secrecy and a provably-secure randomized cipher*. Journal of Cryptology, Vol. 5(1), pp. 53-66, 1992.

[92] U. Maurer: *Cryptography 2000±10*. In Reinhard Wilhelm (Ed.): Informatics – 10 Years Back. 10 Years Ahead. Lecture Notes in Computer Science, Vol. 2000, pp. 63-85, Springer, 2001. ISBN 3-540-41635-8.

[93] D. Merli, D. Schuster, F. Stumpf, G. Sigl: *Side-Channel Analysis of PUFs and Fuzzy Extractors*. TRUST 2011, pp. 33-47, 2011.

[94] D. Merli, J. Heyszl, B. Heinz, D. Schuster, F. Stumpf, G. Sigl: *Localized electromagnetic analysis of RO PUFs*. HOST 2013, pp. 19-24, 2013.

[95] M.K. Mihcak: *Overview of Recent Content Authentication Research at MSR Crypto, Redmond*. Available from https://www.yumpu.com/en/document/view/10835269/m-kivanc-mihcak-uvigo-tv, or from http://tv.uvigo.es/uploads/material/Video/91/Kivanc_Mihcak.pdf.

[96] D. Nedospasov, J.-P. Seifert, C. Helfmeier, C. Boit: *Invasive PUF Analysis*. FDTC 2013, pp. 30-38, 2013.

[97] Y. Oren, A.-R. Sadeghi, C. Wachsmann: *On the Effectiveness of the Remanence Decay Side-Channel to Clone Memory-Based PUFs*. CHES 2013, pp. 107-125, 2013.

[98] R. Ostrovsky, A. Scafuro, I. Visconti, A. Wadia: *Universally Composable Secure Computation with (Malicious) Physically Uncloneable Functions*. EUROCRYPT 2013, pp. 702-718, 2013.

[99] E. Öztürk, G. Hammouri, B. Sunar: *Towards robust low cost authentication for pervasive devices*. IEEE PerCom 2008, pp. 170-178, 2008.

[100] R. Pappu: *Physical One-Way Functions.* PhD Thesis, MIT, 2001.

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

[102] C. Pomerance: *A Tale of Two Sieves*. Notices of the AMS, Vol. 43(12), pp. 1473-1485, 1996.

[103] M. Potkonjak: *Hardware based cryptography.* US-Patent 8379856 B2. Priority date: June 17, 2009.

[104] M. Potkonjak: *Personal communication*, 2011.

[105] R.L. van Renesse: *3DAS: a 3-dimensional-structure authentication system*. European Convention on Security and Detection, pp. 45-49, 1995.

[106] R.L. van Renesse: *Optical document security.* Artech House, third edition, 2005. ISBN-10: 1580532586

[107] R. Rivest: *Illegitimi non carborundum*. Invited keynote talk, CRYPTO 2011. Downloaded from http://www.rsa.com/rsalabs/presentations/Riv11b.slides.pdf, August 2012.

[108] J. Rombach: *Elektrische Charakterisierung zufälliger pn-Dioden für die Kryptographie*. Bachelor thesis, TU München, 2012.

[109] M. Rostami, J.B. Wendt, M. Potkonjak, F. Koushanfar: *Quo Vadis, PUF? Trends and Challenges of Emerging Physical-Disorder based Security*. Design, Automation & Test in Europe (DATE 2014), 2014.

[110] U. Rührmair: *On the Formal Foundations of Physical Unclonable Functions.* Security Hardware in Theory and Practice - A Marriage of Convenience. Event 08253, Schloss Dagstuhl, Germany, 2008.

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

[112] U. Rührmair: *Oblivious Transfer based on Physical Unclonable Functions (Extended Abstract)*. TRUST 2010, pp. 430 - 440, 2010.

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

[114] U. Rührmair: *SIMPL Systems as a Keyless Cryptographic and Security Primitive.* IACR Cryptology ePrint Archive, Report 2011/189, 2011.

[115] U. Rührmair: *SIMPL Systems, Or: Can We Design Cryptographic Hardware without Secret Key Information?* 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), 2011. Lecture Notes in Computer Science, Volume 6543, pp. 26-45, Springer, 2011.

[116] U. Rührmair: *SIMPL Systems as a Keyless Cryptographic and Security Primitive.* In: Cryptography and Security: From Theory to Applications — Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, D. Naccache (Ed). Lecture Notes in Computer Science, Vol. 6805, pp. 329-354, Springer, 2012.

[117] U. Rührmair: *Disorder-based Security Hardware.* PhD Thesis, Technical University of Munich, to be submitted, 2014.

[118] U. Rührmair: *Disorder-based Security Hardware: An Overview.* Invited book chapter, M. Potkonjak and C.-H. Chang (Ed.), Springer, 2014/15, to appear.

[119] U. Rührmair, H. Busch, S. Katzenbeisser: *Strong PUFs: Models, Constructions and Security Proofs.* To appear in A.-R. Sadeghi, P. Tuyls (Editors): Towards Hardware Intrinsic Security: Foundation and Practice. Springer, 2010.

[120] U. Rührmair, Q. Chen, P. Lugli, M. Stutzmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems*. Cryptology ePrint Archive, Report 2009/278, 2009.

[121] U. Rührmair, Q. Chen, M. Stutzmann, P. Lugli, U. Schlichtmann, G. Csaba: *Towards Electrical, Integrated Implementations of SIMPL Systems.* 8th Workshop in Information Security Theory and Practice (WISTP), 2010. Lecture Notes in Computer Science, Volume 6033, pp. 277 - 292, Springer, 2010.

[122] U. Rührmair, S. Devadas, F. Koushanfar: *Security based on Physical Unclonability and Disorder.* In: Introduction to Hardware Security and Trust, M. Tehranipoor, C. Wang, pp. 65-102. Springer New York, 2012.

[123] U. Rührmair, M. van Dijk: *Practical Security Analysis of PUF-Based Two-Player Protocols*. CHES 20120, pp. 251-267, CHES 2012.

[124] U. Rührmair, M. van Dijk: *PUFs in Security Protocols: Attack Models and Security Evaluations*. IEEE Symposium on Security and Privacy 2013, pp. 286-300, 2013.

[125] U. Rührmair, M. van Dijk: *On the Practical Use of Physical Unclonable Functions in Oblivious Transfer and Bit Commitment Protocols.* Journal of Cryptographic Engineering 3(1), pp. 17-28, 2013.

[126] U. Rührmair, C. Hilgers, S. Urban, A. Weiershäuser, E. Dinter, B. Forster, C. Jirauschek: *Optical PUFs Reloaded.* IACR Cryptology ePrint Archive, Report 2013/215, 2013.

[127] U. Rührmair, D.E. Holcomb: *PUFs at a Glance.* 17th Design, Automation and Test in Europe (DATE), pp. 1-6, 2014.

[128] U. Rührmair, C. Jaeger, M. Algasinger: *An Attack on PUF-Based Session Key Exchange and a Hardware-Based Countermeasure: Erasable PUFs*. Financial Cryptography 2011, pp. 190-204, 2012.

[129] U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, G. Csaba: *Applications of High-Capacity Crossbar Memories in Cryptography.* IEEE Transactions on Nanotechnology 10(3), pp. 489-498, 2011.

[130] U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, M. Stutzmann: *Security Applications of Diodes with Unique Current-Voltage Characteristics*. Financial Cryptography and Data Security (FC 2010), Lecture Notes in Computer Science, Vol. 6052, pp. 328-335, Springer Verlag, 2010.

[131] U. Rührmair, S. Katzenbeisser, M. Steinebach, S. Zmudzinski: *Watermark-Based Authentication and Key Exchange in Teleconferencing Systems.* 11th Conference on Communications and Multimedia Security (CMS), 2010. Lecture Notes in Computer Science, Volume 6109, pp. 75 - 80, Springer, 2010.

[132] U. Rührmair, U. Schlichtmann, W. Burleson: *Special Session: How Secure are PUFs Really? On the Reach and Limits of Recent PUF Attacks.* 17th Design, Automation and Test in Europe (DATE), pp. 1-6, 2014.

[133] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions.* IACR Cryptology ePrint Archive, Report 2010/251, 2010.

[134] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions*. ACM CCS, pp. 237-249, 2010.

[135] U. Rührmair, J. Sölter: *PUF Modeling Attacks: An Introduction and Overview*. 17th Design, Automation and Test in Europe (DATE), pp. 1-6, 2014.

[136] U. Rührmair, J. Sölter, F. Sehnke: *On the Foundations of Physical Unclonable Functions.* Cryptology e-Print Archive, June 2009.

[137] 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.* IACR Cryptology ePrint Archive, Report 2013/112, 2013.

[138] 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 Transactions on Information Forensics and Security, Vol. 8(11), pp. 1876-1891, 2013.

[139] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, W. Burleson: *Power and Timing Side Channels for PUFs and their Efficient Exploitation.* IACR Cryptology ePrint Archive, Report 2013/851, 2013.

[140] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, W. Burleson: *Efficient Power and Timing Side Channels for Physical Unclonable Functions*. 16th Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2014. Lecture Notes in Computer Science, Springer, 2014 (to appear).

[141] See http://www.informatik.uni-trier.de/ Ley/db/conf/ches/index.html

[142] See http://www.informatik.uni-trier.de/ LEY/db/conf/host/index.html

[143] See http://www.gi-de.com/en/trends_and_insights/banknote_circulation/life_of_a_banknote/life-of-a-banknote.jsp

[144] See http://rmaes.ulyssis.be/pufbib.php

[145] See http://www.answers.com/topic/certegy-inc-1

[146] See http://www.design-reuse.com/articles/16975/arm-security-solutions-and-intel-authenticated-flash-how-to-integrate-intel-authenticated-flash-with-arm-trustzone-for-maximum-system-protection.html

[147] See http://www.adnas.com/products/signaturedna

[148] See http://www.date-conference.com/conference/session/4.3

[149] See http://www.date-conference.com/conference/session/12.2

[150] See http://www.date-conference.com/category/session-types/tutorial

[151] See http://www.chemistryexplained.com/St-Te/Surface-Chemistry.html

[152] See www.nxp.com/documents/other/75017366.pdf

[153] See http://www.nxp.com/news/press-releases/2013/02/nxp-strengthens-smartmx2-security-chips-with-puf-anti-cloning-technology.html

[154] See http://investor.microsemi.com/releasedetail.cfm?ReleaseID=731250

[155] See http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2

[156] See http://en.wikipedia.org/wiki/Pollen#mediaviewer/File:Lilium_auratum_-_pollen.jpg

[157] See http://en.wikipedia.org/wiki/Filter_paper#mediaviewer/File:Filter_paper_840_3x3_copy.jpg

[158] See http://de.wikipedia.org/wiki/Compact_Disc#mediaviewer/Datei:REM_CD_GEPRESST.jpg

[159] See http://www.chipworks.com/components/com_wordpress/wp/wp-content/uploads/2013/08/A5-Processor-from-ipad-Mini-300x249.jpg. Image is by Chipworks Inc. (www.chipworks.com), oral and written permission for reproduction granted by D. James of Chipworks Inc. to the author on June 10/11, 2014.

[160] F. Sehnke, C. Osendorfer, J. Sölter, J. Schmidhuber, U. Rührmair: *Policy Gradients for Cryptanalysis.* 20th International Conference on Artificial Neural Networks (ICANN), 2010. Lecture Notes in Computer Science, Volume 6354, pp. 168-177, Springer, 2010. [160]

[161] A. Shamir, N. van Someren: *Playing "Hide and Seek" with Stored Keys*. Financial Cryptography 1999: 118-124.

[162] C. E. Shannon: *A Mathematical Theory of Communication*. Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, 1948. ISSN 0005-8580.

[163] C. E. Shannon: *Communication Theory of Secrecy Systems*. Bell System Technical Journal, Vol. 28(4), pp. 656-715, 1949.

[164] A. Sharma, L. Subramanian, E.A. Brewer: *PaperSpeckle: microscopic fingerprinting of paper.* ACM CCS 2011, pp. 99-110, 2011.

[165] P.W. Shor: *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM Journal on Computing, Vol. 26(5), pp. 1484-1509, 1997.

[166] G.J. Simmmons: *Identification of data, devices, documents and individuals*. Annual International Carnahan Conference on Security Technology, pp. 197-218, 1991.

[167] J.R. Smith, A.V. Sutherland: *Microstructure based indicia*. Second Workshop on Automatic Identification Advanced Technologies, pp. 79-83, 1999.

[168] S. Skorobogatov: *Low temperature data remanence in static RAM*. Technical Report UCAM-CL-TR-536, Computer Laboratory, University of Cambridge, 2002. Available from http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-536.pdf.

[169] M. Steinebach, S. Zmudsinski, S. Katzenbeisser, U. Rührmair: *Audio watermarking forensics: detecting malicious re-embedding.* IS&T/SPIE Electronic Imaging Conference – Media Forensics and Security XII, 2010.

[170] S. Stepney: *Journeys in non-classical computation*. In: T. Hoare, R. Milner (Eds.): Grand Challenges in Computing Research, pp. 29-32. Swindon, BCS, 2004.

[171] M. Stutzmann, G. Csaba, P. Lugli, J. Finley, C. Jirauschek, C. Jaeger, U. Rührmair: *Towards Electrical, Integrated Implementations of SIMPL Systems.* European Patent (EP) 2230794 A3. Priority date: March 16, 2009.

[172] G.E. Suh, D.E. Clarke, B. Gassend, M. van Dijk, S. Devadas: *AEGIS: architecture for tamper-evident and tamper-resistant processing*. ICS 2003, pp. 160-171, 2003.

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

[174] D. Suzuki, K. Shimizu: *The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes*. CHES 2010, pp. 366-382, 2010.

[175] C. Troncoso, D. De Cock, B. Preneel: *Improving secure long-term archival of digitally signed documents*. StorageSS 2008: 27-36

[176] P. Tuyls, G.J. Schrijen, B. Skoric, J. van Geloven, N. Verhaegh, R. Wolters: *Read-Proof Hardware from Protective Coatings.* CHES 2006, pp. 369-383, 2006.

[177] P. Tuyls, B. Skoric: *Strong Authentication with Physical Unclonable Functions.* In: Security, Privacy and Trust in Modern Data Management, M. Petkovic, W. Jonker (Eds.), pp. 133-148, Springer, 2007.

[178] P. Tuyls, B. Skoric, S. Stallinga, A.H.M. Akkermans, W. Ophey: *An information theoretic model for physical uncloneable functions*. IEEE International Symposium on Information Theory, p. 141, 2004.

[179] P. Tuyls, B. Skoric, S. Stallinga, A.H.M. Akkermans, W. Ophey: *Information-Theoretic Security Analysis of Physical Uncloneable Functions*. Financial Cryptography, pp. 141-155, 2005.

[180] A.W. Vaidya: *Keeping card data secure at low cost.* European Convention on Security and Detection, pp. 212-215, 1995.

[181] D. Vijaywargi, D. Lewis, D. Kirovski: *Optical DNA*. Financial Cryptography 2009, pp. 222-229, 2009.

[182] S. Wolf, J. Wullschleger: *Oblivious Transfer Is Symmetric*. EUROCRYPT 2006, pp. 222-232, 2006.

[183] Wikipedia's article on cryptography. Downloaded from http://en.wikipedia.org/wiki/Cryptography, August 2012.

[184] Wikipedia's article on data remanence. Downloaded from http://en.wikipedia.org/wiki/Data_remanence, August 2012.

[185] Wikipedia's article on a one-electron universe. Downloaded from http://en.wikipedia.org/wiki/One-electron_universe, April 2014.

[186] Wikipedia's article on RSA numbers. Downloaded from http://en.wikipedia.org/wiki/RSA_numbers, August 2012.

[187] Wikipedia's article on "the magic words are squeamish ossifrage". Downloaded from http://en.wikipedia.org/wiki/The_Magic_Words_are_Squeamish_Ossifrage, August 2012.

[188] Wikipedia's article on the VENONA project. Downloaded from http://en.wikipedia.org/wiki/Venona_project, August 2012.

[189] D. Yamamoto, K. Sakiyama, M. Iwamoto, K. Ohta, T. Ochiai, M. Takenaka, K. Itoh: *Uniqueness Enhancement of PUF Responses Based on the Locations of Random Outputting RS Latches*. CHES 2011, pp. 390-406, 2011.

[190] M.-D. Yu, D. M'Raihi, R. Sowell, S. Devadas: *Lightweight and Secure PUF Key Storage Using Limits of Machine Learning*. CHES 2011, pp. 358-373, 2011.