# SIMPL Systems: On a Public Key Variant of Physical Unclonable Functions

Ulrich Rührmair

June 1, 2009

## 1 Abstract

This paper theoretically discusses a novel security tool termed *SIMPL system*, which can be regarded as a public key version of physical unclonable functions (PUFs). Like the latter, a SIMPL system $S$ is physically unique and non-reproducible, and implements an individual function $F_S$. In opposition to a PUF, however, a SIMPL system $S$ possesses a publicly known numerical description $D(S)$, which allows its digital simulation and prediction. At the same time, it is required that any digital simulation of a SIMPL system $S$ must work at a detectably lower speed than its real-time behavior.

In other words, the holder of a SIMPL system $S$ can evaluate a publicly known, publicly computable function $F_S$ faster than anyone else. This feature, so we argue in this paper, allows a number of improved practicality and security features. Once implemented successfully, SIMPL systems would have specific advantages over PUFs, certificates of authenticity, physically obfuscated keys, and also over standard mathematical cryptotechniques.

## 2 Introduction

Physical Unclonable Functions (PUFs) are a powerful new cryptographic primitive which has been discussed recently in a number of publications, e.g. [1, 2, 3, 4, 5, 6]. However, one potential downside of many PUF-based protocols is that they require a previously shared piece of information (usually some challenge-response-pairs) that is typically 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 suggest an alternative security tool called a *SIMPL system*, which could be understood as a public key version of standard PUFs. The acronym SIMPL stands for "SIMulation Possible, but Laborious", and 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 $\mathsf{Sim}$, 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. One of the special features of SIMPL systems is, however, that this "private key" is an irreproducible physical structure, which contains no secret information at all.

As we will argue in this paper, once implemented successfully, SIMPL systems would possess some notable security and practicality advantages: In opposition to standard binary keys, for example, SIMPL systems are naturally immune against invasive or side channel attacks, and also insensitive against viruses, Trojan horses or other malware. Furthermore, their security does not depend on the classical unproven crypto-assumptions (factoring or discrete log), but on independent assumptions. Compared to PUFs, they allow protocols without individual set-up phases and trusted central authorities, and also new types of applications.

The rest of this manuscript is organized as follows: In section 3 we give a full formal definition of SIMPL systems. Section 4 provides two formal SIMPL-based protocols for entity identification and message authentication. Subsequently, section 5 discusses the application of SIMPL systems to several concrete problems, illustrating their upsides by virtue of concrete examples. In section 6 we briefly discuss a few concrete implementation strategies for SIMPL systems, and conclude the paper in section 7.

## 3 Definition of SIMPL Systems

**Definition 3.1** ($(t_C, t_{Ph}, \epsilon)$-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 $(t_C, t_{Ph}, \epsilon)$-*SIMPL SYSTEM *if there is a numerical string $D(S)$, called the description of $S$, and a generic computer algorithm* $\mathsf{Sim}$ *such that the following conditions are met:*

1. *For all challenges $C_i \in \mathbf{C}$, the algorithm* $\mathsf{Sim}$ *on input*

$$\big(C_i,\, D(S)\big)$$

*outputs $R_i$ in feasible time.*

2. *Any cryptographic adversary Eve, who is bound to practically feasible probabilistic Turing computations and practically feasible physical actions, will succeed in the following* **security experiment** *with a probability of at most $\epsilon$:*

*(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 is given physical access to the system $S$ at adaptively chosen time points, and for adaptively chosen time periods. The only restriction is that her access times must add up to a total of at most $t_{Ph}$. After the access times have ended, she does not have physical access to $S$ anymore.*

*(c) Subsequently, 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 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 $2 \cdot t_{max}$.*

*Please note that the said probability of $\epsilon$ 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 definition are in order.

**Security Model.** Let us start by briefly discussing the security model of the definition. In practive, an adversary Eve can gather information about $S$ in essentially two ways. Firstly *computationally*, by obtaining and analyzing challenge-response-pairs $(C_i, R_i)$ and by analyzing the algorithm Sim and the description $D(S)$. Thereby the CRPs may either stem from eavesdropping some protocols, or they may be computed by the adversary himself by use of the algorithm Sim and the description $D(S)$, which are both public. These possibilities are reflected in item *2a* of the definition. Secondly, Eve may *physically* measure arbitrary features of the system $S$ at some point. This possibility is covered in item *2b*. The model reflects real-world situations, for example if $S$ was used in mobile systems for identification purposes. Then, Sim and $D(S)$ would be public, but at the same time an adversary would be strongly limited in his unrestricted physical access time. Typically, $t_{Ph}$ will be much shorter in such situations than the period $t_C$.

**Immunity against Full Read-Out.** It follows from Definition 3.1 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 a relatively short timeframe. 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), or successive read-outs can only be carried out relatively slowly, or both holds.

**Immunity Against Cloning.** Please note further that Definition 3.1 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 an experiment on the original system $S$.

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 $S$.

Please note that 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, which takes most of the computational load, 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 $S$.

**Level of Formality and Practical Feasibility vs. Infeasibility.** Any formal definition of a cryptographic or security scheme must draw a distinction between tasks that are practically feasible and practically non-feasible. In mathematically based cryptography, this distinction is commonly made by refrering to the concepts of polynomial and superpolynomial time. These asymptotic notions, however, can meaningfully only be applied to an infinite function or to a familiy of finite functions. But real physical systems, such as PUFs or SIMPL systems, implement finite functions. If asymptotic concepts are applied in this situation, this leads to formal contradictions and problems. They can have the consequence that no function and no physical system can formally meet the definitions. Furthermore, the usual polynomial vs. exponential bounds may not fit the context very well. It can be shown that any physical system contains at most a polynomial amount of information (in bits) in its size (see [8] for a detailed treatment of the whole issue).

This means that we have two options for a formal definition of SIMPL systems: (i) We introduce a new computational model for our definition, which must be general enough to include arbitrary physical actions that Eve performs on S. Furthermore, we should add an asymptotic treatment that is not based on the usual polynomial/exponential distinction to the definition. Both steps make an already involved definition yet more complicated, without adding strongly to our purpose. (ii) We try to specify Eve's task and the general security model as carefully and precisely as possible, but without a formal computational model and without an asymptotic treatment. Whenever the computational resources play a role, we find shelter in the term "practically feasible".

We opted for the second possibility, since it leads – from our perspective – to a workable, intuitive and pragmatic definition. It is also free of direct contradictions [8]. Please note that the standard way of equating polynomial time with feasible computations has been subject to discussion, too [21]. Furthermore, the standard formalization of cryptography (polynomial Turing machine computations = feasible computations) may be incomplete and exhibits a gap, provided that the extended Church-Turing thesis is invalid (for the extended Church Turing thesis, see [24]). For example, since quantum computers could factor efficiently [26], but factoring may at the same time be infeasible on a Turing machine, we could eventually (in some decades) be faced with the following situation: We can formally prove that RSA is secure in our current, Turing machine based theoretical security model, while it may be fully insecure in practive (if quantum computers have been built by then).

We believe that this indicates that the value of using exactly polynomial time Turing computations as a model for practical feasibility is, to some extent, limited. While we are still aware of the beauty and elegancy this approach offers, we also feel no problem if we skip this paradigm when there are good reasons to do so.

**Time Gap between Eve and the SIMPL System.** The definition 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, as already said before, these asymptotic notions cannot be applied directly to the finite function which a SIMPL system implemenents without rising contradictions [8].

In the application protocols which we suggest (identificaton and on-the-fly message authentication), some detectable time difference at the time of the protocol execution suffices. No long-term security properties similar to the confidentiality of encryption are required; an ad-hoc distinction between fakers and honest identifiers or authentic messages suffices.

Note also that 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. This brings us into a region of absolute delay values where we can maintain security even in the face of unwanted side effects, such as network and transmission delays. In the ideal case, the speed up would certainly be a larger constant or super-linear factor, but it is not clear whether high degree polynomial speed gaps between the physical SIMPL system and a Turing machine, together with the uniqueness requirement for SIMPL systems, would be possible at all.

It is also worth noting that the question of the general Turing-simulatability of physical systems and of the efficiency of such simulations has a long record. For example, the Extended Church-Turing Thesis conjectures that any physical system is in its computational power polynomially equivalent to a Turing machine [24]. In particular, it has been argued that no physical systems can solve NP-complete problems efficiently in practice [25]. On the other hand, it is known that quantum computers might violate the Extended Church-Turing Thesis, since they can solve the factoring problem in polynomial time, a task that classical Turing machines may not be capable of [26]. If we leave efficieny considerations aside, already Feynman believed that any physical system can in principle be simulated by a Turing machine [10]. Actually, most of these results seem to support the basic feasibility of the concept of a SIMPL system, and also the decision to not base our definition on the distinction between polynomial and exponential resources.

# 4 Protocols

We will now provide two exemplary protocols that can be realized by SIMPL systems.

## 4.1 Identification

We assume that Alice, who holds an individual SIMPL system $S$, has put $D(S)$, $\mathsf{Sim}$, $t_{max}$ and a description of $\mathbf{C}$ in a public register (we will not discuss PKI-related problems such as [9] here). Now, she can prove her identity to an arbitrary second party Bob as follows:

**Protocol 4.1:** Identification of Entities by SIMPL Systems

1. Bob obtains the information $D(S)$, $\mathsf{Sim}$, $t_{max}$, 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 $\mathsf{Sim}$ if for all $i = 1, \ldots, k$,

$$V_i \;=\; R_i.$$

   If this is the case, Bob believes Alice's identity, otherwise not.

**Security Heuristic.** 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 do not exceed $t_{Ph}$. In that case, Eve's probability to succeed in the protocol without possessing $S$ are less or equal to $\epsilon^k$.

**Practicality.** Bob can improve his computational efficiency by verifying the correctness of the responses $R_i$ only for a randomly chosen 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 3).

A number of concrete appliances can be derived from the above identification protocol, which will be discussed in section 5.

### 4.2 Authentication of Messages

Alice can also employ an individual SIMPL system $S$ being in her possession to authenticate messages to Bob. Again, we suppose that Alice has put $D(S)$, $\mathsf{Sim}$, $t_{max}$ and a description of $\mathbf{C}$ in a public register. Now, she can authenticate a message $N$ to Bob as follows:

**Protocol 4.2:** Authentication of Messages by SIMPL Systems

1. Bob obtains the information $D(S)$, $\mathsf{Sim}$, $t_{max}$ 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 uses $S$ to determine the corresponding values $R_1, \ldots, R_k$. She derives $l$ keys $K_1, \ldots, K_l$ from these values, for example by applying a suitably chosen, public hash function.

4. Alice uses a standard Message Authentication Code $MAC$ with the keys $K_1, \ldots, K_l$, and sends the values
$$N,\ MAC_{K_1}(N),\ \ldots,\ MAC_{K_l}(N)$$
to Bob.

5. Bob receives values $N', V_1, \ldots, V_l$. He measures the time that passed between sending the challenges $C_1, \ldots, C_k$ in step 2 and receiving the values $N', V_1, \ldots, V_l$. If it is above the threshold $2 \cdot t_{max}$, then he aborts the protocol.

6. Bob computes the values $R_1, \ldots, R_k$ by simulation via $\mathsf{Sim}$, and derives the keys $K'_1 \ldots, K'_l$ by application of the same hash function as in step 3.

7. Bob checks if for all $i = 1, \ldots, l$,
$$MAC_{K'_i}(N') = V_i.$$

If this is the case, he regards the message $N' = N$ as properly authenticated, otherwise not.

**Security Heuristic.** Again, $k$ and $l$ are the security parameters. The security of the protocol obviously depends on the security of the employed hash function and MAC, and otherwise follows from the fact that Eve cannot determine the responses $R_1, \ldots, R_k$ and the MAC-Keys $K_1, \ldots, K_l$ as quickly as Alice. The latter will hold as long as the lifetime of the device does not exceed the $t_C$, and as long as Eve's uninterrupted, unnoticed physical access does not accumulate to a time period longer than $t_{Ph}$. Under these provisions, Eve's probability to succeed decreases exponentially in $l$ and $k$.

If information theoretically secure hash functions and MACs are used, the security will not depend on any computational assumptions other than the security of the SIMPL system. Please note that MACs can be implemented very efficiently [11], meaning that their computation time does not strongly affect the protocol's security.

**Practicality.** Feedback loops may compensate for network delays, and verification of a randomly chosen subset of all MACs can improve Bob's computational efficiency.

# 5 Applications

We will now analyze the potential and advantages of SIMPL systems in a number of concrete applications of the above identification protocol. We will comparative analyses with standard binary techniques, PUFs, certficates of authentity (COAs) [12] and physically obfuscated keys (POKs) [3].

## 5.1 Identification of Computer Systems

One straightforward application of Protocol 4.1 is the identification of computer systems or other hardware. The computer system/hardware carries its individual SIMPL system $S$, and uses protocol 4.1 in order to identify itself.

**Comparative Analysis.** Compared to PUFs, identification based on SIMPLs works without joint set-up phases or online connections to central authorities. Furthermore, in opposition to physically obfuscated keys (POKs) [3] or standard private binary keys, it functions without secret binary information at all. Such information can be transfered by malware, or obtained via invasive, power consumption or emanation analysis, even if it exist in binary form only a short time in the system like POKs [13, 14, 15]. Furthermore, SIMPLs avoid the classical unproven number-theoretic assumptions (facoring, DL), but rest on other, independent assumptions.

## 5.2 Unforgeable Labels

The worldwide economic damage caused by faked branded products is on the order of several hundred billion dollars per annum [12, 17]. Truly unforgeable and inexpensive product labels have therefore been investigated intensively. One promising approach is to use disordered

and non-reproducible physical structures as unforgeable labels in one way or the other [1, 12, 18, 19, 20]. For example, *unique objects* (structures which generate a truly unique and non-imitable *analog* measurement signal) can be used in connection with digital signatures to form highly secure labels or 'certificates of authenticity' (COAs) [12], which can be verified offline.

An alternative is the use of PUFs as labels. They allow for an integrated read-out apparatus, digitized measurement signal, and potentially long-distance read-out. However, no offline label verification with standard PUFs is possible (albeit, to some extent, with POKs, at the price of a very high computational load in the tag). Standard techniques like RFID tags with secret keys face key read-out by invasive or side channel attacks.

Ideally, one would like to establish labels with the following properties: They contain no secret information at all, ideally also not in the form of a POK. They can bee be read out *digitally* and *over long distances*. They could be verified offline, without a central institution/database. SIMPL systems are the only structures known to the author that can realize such types of labels.

A SIMPL-label consists of a the following components:

(i) The SIMPL System $S$.

(ii) The description $D(S)$ and some product related info $I$.

(iii) The digital signature $Sig_{SK}(D(S), I)$, created by the secret signing key $SK$ of the label issuer, which is stored on the labeled item.

In the verification process, the testing apparatus obtains $D(S)$ from the label, verifies the digital signature via use of a publicly known key $PK$, and executes Protocol 4.1 in order to check the presence of the SIMPL system $S$. Only descriptions of $PK, \mathbf{C}$, $t_{max}$ and $\mathsf{Sim}$ needs to be hardwired into the apparatus, meaning that the testing apparatuses do not need to contain secret information.

### 5.3 Copy Protected Digital Content and DRM

SIMPL systems can also be applied to the management of digital rights and to the generation of copy protected content. The approach is similar to section 5.2: A *legitimate representation* of digital content consists of the following components: (i) The SIMPL system $S$. (ii) The digital content $Con$, plus some additional information $I$ where required. (iii) The digital signature $Sig_{SK}(D(S), Con, I)$.

A control device that wants to check whether a given content is in *legitimate form* before playing it, must hold $\mathsf{Sim}$, $t_{max}$, $\mathbf{C}$ and $PK$. It verifies whether the signature is valid, and whether the SIMPL system $S$ is present (by running Protocol 4.1). If these two conditions are fulfilled, it plays the content, otherwise not.

The advantage of this approach is that it merely requires digital communication, and that it therefore works in fully digital environments. For example, the role of the control device can be played by a trusted platform module or a CPU; the content $Con$, information $I$, description $D(S)$, signature $Sig_{SK}(D(S), Con, I)$ and the SIMPL system $S$ can be stored

somewhere else in the computer system. Alternatively, the SIMPL system $S$ plus its description $D(S)$ may be contained on an external, personalized plug-in device of a user (like a USB-stick). The content will only play while the device is plugged in.

Copy-protected digital content of the described sort can also be distributed securely in an online fashion: The user sends the description $D(S_{User})$ to the server, which returns $Con$, $I$ and $Sig_{SK}(D(S_{User}), Con, I)$. This allows new, secure distribution channels.

**Comparative Analysis.** Approaches based on unique physical structures (i.e. physical systems that create a unique and non-imitable analog measurement signal) up to date required the *direct* and *analog* measurement of said unique signals, for example the optical signal generated by the unique irregularities of a CD (see, for example, [22]). This direct, analog measurement cannot be made by a TPM or the CPU itself. That means that they can be spoofed by manipulated third system components, which execute the analog measurements for them, but communicate false results. This is in contrast to the fully digital execution of Protocol 4.1, where the TPM or CPU merely needs its own clock to measure the response time of the SIMPL system.

### 5.4 Copy Protected Software

The above scheme of *legitimate representations of content* has one disadvantage: It allows the playing of pirated content on "old" playing devices or "old" hardware systems, which do not check whether inserted content has the legitimate form. If the protected content is a piece of computer software, the picture changes: Since the software is active code, it can enforce said check by itself. It will *never* run unless the digital signature and the SIMPL system are present, even if a security check is not routinely enforced. To cope with the fact that old hardware will not contain SIMPL systems, the software can be issued together with a USB stick carrying the SIMPL system (see section 5.3).

Software $Soft$ protected by this means can also be distributed online (see again the last section): The user sends $D(S_{User})$ to the company, and gets $Soft$, $I$ and $Sig_{SK}(D(S_{User})$, $Soft, I)$ in return.

**Comparative Analysis.** The advantages are similar to section 5.3. In addition, software protection based on SIMPL systems cannot be fooled by a sandbox simulation of a fraudster who knows the binary keys/chip identities that were used to bind software to a particular piece of hardware. This limits the effect of a single extracted key which was made public.

### 5.5 Tamper Evident Hardware

If a hardware system is covered by a SIMPL system (similar to a coating PUF [4]), it is possible to verify the integrity of the hull by remote, digital communication. The hardware system stores $D(S_{Hull})$, $I$, $Sig_{SK}(D(S_{Hull}), I)$, where $SK$ is the secret key of the trusted manufacturer of the hardware system, and $I$ is some optional information about the hardware system. In order to verify the non-tamperedness, the validity of the signature is tested, and Protocol 4.1 is carried out between the hardware system and a verifier.

**Comparative Analysis.** The security is neither based on secret binary keys nor on POKs, with the upsides as above.

# 6 Implementation of SIMPL Systems

The main focus of this manuscript was to suggest the theoretical concept of a SIMPL system and to illustrate its application potential. Its central aim is not to discuss possible pratical implementations of such systems. Nevertheless, we will briefly mention some promising strategies to that end.

## 6.1 SIMPL Systems from PUFs with Reduced Complexity

One generic strategy for the construction of SIMPL systems is to reduce the inner complexity of physical systems that have been suggested as PUFs. Sufficient reduction may eventually lead to systems that can be simulated or predicted numerically. If the system is still complex enough, simulation or prediction will nevertheless be slower than the system's real-time behavior.

In order to make this approach applicable, we need a strategy to obtain the general simuation algorithm $\mathsf{Sim}$ and the specific description $D(S)$. Two suggestive approaches are machine learning or other numeric system analysis of the system on the one hand, or direct physical probing of the system on the other hand.

If machine learning is applied, the following steps would need to be executed:

1. Generate a physical system that has been (or could be) suggested as a Physical Unclonable Function, but reduce its complexity to a point where it can be machine learned.
2. Determine many challenge response pairs of the system.
3. Use the CRPs as input for a ML algorithm until it has been trained successfully (or as input to some other numeric system analysis).
4. Use a binary code of the trained algorithm as the description $D(S)$ (a general simulation algorithm $\mathsf{Sim}$ is unneccesary in this case).

Likewise, one might use physical probing in step 2 of the procedure in order to determine $D(S)$. The role of $\mathsf{Sim}$ would in this case be played by a physical simulator matching the employed system, for example an simulator based on Maxwell's equations.

## 6.2 Optical Systems

Let us now discuss the concrete application of the above strategy to optical PUFs.

### 6.2.1 SIMPLs derived from Pappu's optical PUF

The optical PUF of [1, 2] seems to be suitable for our task. Its complexity can be reduced almost continuously by taking smaller and smaller numbers of scatterers. As a concrete figure, we suggest to use around $10^5$ scatterers in a suitable matrix. The scatterers should be as equal in size and as perfectly spherical in shape as possible. The material between the scatterers should be perfectly homogeneous in its refractive index. These two constraints seem to be achievable by current nanofabrication techniques: Any changes in the systems

structure that are significantly smaller than the wavelength of the employed laser light (which is several hundred nanometer!) will not affect the interference outcome.

Under these provisions, a numerical simulation of the described structure does not need to consider the pairwise interplay of *all* volume subunits of size $\lambda^3$ of the matrix, as suggested in [1, 2]. Instead, it suffices to consider the interplay of the $10^5$ scattering centers.

This implies that Sim must only include the material constants, such as refractive index of the matrix and of the scatterers, the scatterer size and shape, etc. The individual description $D(S)$ must merely include the position of the scattering centers. These can be determined sufficiently accurately by established optical imaging techniques (also in 3D), for example NMR.

Using the estimate given in [1] as a basis, the simulation of such a structure should take roughly $10^5 \times 10^5 \times 10^2 = 10^{12}$ steps. The real scattering structure, on the other hand, would generate the resulting pattern in picoseconds. Recording of the speckle pattern does not necessarily have to be carried out by slow CCD cameras, but by a smaller and extremely quick arrays of photosensitive diodes. Overall, this makes such a reduced complexity PUF a good candidate for a SIMPL system.

Please note that the description $D(S)$ does not necessarily need to be determined by physical imaging. An alternative that makes the prediction more robust against possible deviations from an idealized system (such as deviations from the assumed perfect scatterer shape, or from the perfectly homogenous matrix material), is to employ machine learning methods. Certain well-known techniques, such as Support Vector Machines, are specialized to learn large, linear systems; they are known to be robust against noise both in the input data as well as in the model [27, 28]. The trained ML algorithm would then serve as $D(S)$.

### 6.2.2 Integrated Optical PUFs

Also an integrated PUF with reduced complexity can be used. Its working principle is depicted schematically in Figure 1: An immobile laser diode array with $k$ phase-locked diodes [30] is used to excite a disordered scattering medium. (Alternatively, one light source together with light modulators may be employed.) The diodes can be switched on and off independently, leading to $2^k$ challenges $C_i = (b_1, \ldots, b_k)$, where each $b_k$ indicates whether diode $k$ is switched on or off. The diode array must be phase locked. At the right hand side, an array of $l$ photodetectors measures the resulting light intensities pointwise. The responses $R_i$ consist of the pointwise intensitites in the photodetectors. Related structures have been proposed in [6, 7].

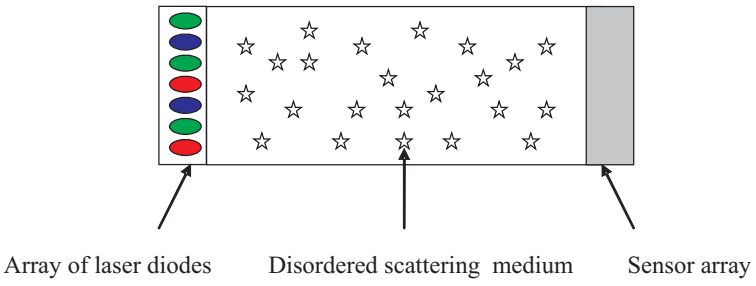


Figure 1: An integrated optical PUF/SIMPL system

Integrated optical PUFs have the following special feature: If the employed scattering medium is linear, then the output $R_i$ for a complex challenge vector $C_i$ can be inferred from a few known responses by the superposition principle of optics [29]. This works in the following fashion: Suppose that some person knew the intensities, and also the relative phases, of the light vectors in the $l$ photodetectors for all $k$ challenges from the set $E = \{C_i = (b_1, \ldots, b_k) \mid b_i \in \{0,1\} \text{ and } \sum_{j=1,\ldots,k} b_j = 1\}$. $E$ contains all the challenges in which only one diode is switched on, and all others are switched off. Then, the person can deduce the output $R_i$ for a more complex challenge vector $C_i$ as a direct function of the light intensities and relative phases that resulted from the challenges in the set $E$. She can do so by applying the superposition principle.

How can one obtain the relative phases? Either, one adds some extra measurement device to the scatterer and determines them directly. Or, one applies a hidden variable model and machine learning techniques [27] to a large number of CRPs.

This has got two consequences. On the one hand, Eve can apply the above procedure to infer from many CRPs a prediction model. On the other hand, also Alice can apply this procedure in order to obtain and publish $D(S)$. Overall, the good news prevail, since Eve's prediction will likely not work on the same timescales as the real optical system, especially if a large number of laser diodes is employed. This makes integrated optical systems good and perhaps even more practical candidates than standard optical PUFs.

**Non-Linear Materials.** Also non-linear scattering media, for example quantum dots, may be used in connection with the described integrated systems. This disables output prediction by superposition and the application of simple and straightforward machine learning techniques. It realizes higher effective information densities in the scattering medium, and could eventually lead to a SIMPL system $S$ with a higher time gap between Eve and the real owner of $S$.

In particular, strong non-linearities will make it necessary to again inspect up to all volume units of size $\lambda^3$ in the simulation and prediction process, as analysed in [1]. This means that for an integrated system of size $100\mu m \times 1mm \times 1mm$ and a wavelength of $100nm$, overall up to $10^3 \times 10^4 \times 10^4$ basic simulation steps are necessary. In other words, the time gap for non-linear integrated systems will be quite significant.

### 6.3 Outlook: Speeding Up Bob's Task

We conclude by an outlook on future practicality optimization. In the Protocols 4.1 and 4.2, Bob checks Alice's answers for correctness by simulation of the SIMPL system. This necessarily imposes some computational load on him. As mentioned in section 4, the load can be diminished by choosing a random subset of Alice's answers for verification, but this might still be unsatisfactory in certain situations.

Another strategy is to utilize the computational asymmetry between computing a solution and verifying it for correctness. This asymmetry has been long known in complexity theory, and is fundamental to the well-known distinction between the complexity classes P and NP. One may consider exploiting this asymmetry to speed up Bob's task.

For example, Alice could add some extra informations $E_i$ that act similar to an NP-certificate in steps 3 and 4 of Protocols 4.1 and 4.2. They may allow Bob to verify the

correctnes of the response $R_i$ (Prot. 4.1) or of the MACs (Prot. 4.2) more quickly. The $E_i$ may consist of intermediate measurements steps or results which arose as a sideproduct in the measurement of the $R_i$. In particular, $S$ may be designed in such a way that it generates such values intendedly.

If the physical behavior of $S$ is governed by differential equations, for example, Bob's verification may be sped up by inserting certain intermediate values measured by Alice directly into the diffential equations, and checking them for correctness (instead of doing a full simulation from scratch). If $S$ was a complex network, where the response $R_i$ would usually only be measured at the boundaries, Alice might provide measurement values of the inner nodes in order to allow quicker verification. In dependency of the concrete system $S$, many other possibilities are conceivable.

# 7 Conclusions

This paper introduced a novel security concept termed "SIMPL Systems", which can be regarded as a public key version of physical unclonable functions. 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. As discussed in detail, this can lead to critical security and practicality advances in a number of applications.

After abstract and introduction, we provided a formal definition of SIMPL systems, which was based on the concept of a *security experiment*. It avoided asymptotic concepts like polynomial time or negligible probability, trying to strike some balance between formality, pragmatism and formal soundness. Then, we gave two example protocols based on SIMPL systems, namely entity identification and message authentication. We gave security heuristics as to why these protocols are secure, provided that the employed structures are true SIMPL systems.

In the next section, we described concrete security applications of SIMPL systems, which included the identification of hardware, digital rights management and unforgeable labeling. We argued that the use of SIMPL systems in these settings leads to some notable advantages over standard cryptotechniques, and also over alternative approaches including PUFs, POKs, and COAs/unique objects. After that, we briefly discussed possible implementation strategies for the practical realization of SIMPL systems.

The presented material seems to indicate the potential and practical value of SIMPL systems, and will hopefully lead the fundament for further investigations on the topic. Future work could focus on the concrete practical realization of these structures, especially on promising IC-based candidates, and on full formal security proofs and models.


# Acknowledgements

The author would like to thank Christian Osendorfer, Christian Jirauschek, Martin Stutzmann, Ulf Schlichtmann, Gyorgy Csaba, Jürgen Schmidhuber, Tamas Roska, Peter Vogl, and Frank Sehnke for enjoyable and helpful discussions on the topic.

## References


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

[2] R. Pappu, *Physical One-Way Functions*, PhD Thesis, MIT.

[3] Blaise Gassend, *Physical Random Functions*, MSc Thesis, MIT, 2003.

[4] Pim Tuyls, Geert Jan Schrijen, Boris Skoric, Jan van Geloven, Nynke Verhaegh, Rob Wolters *Read-Proof Hardware from Protective Coatings.* CHES 2006: 369-383

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

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

[7] P. Tuyls, B. Skoric, T. Kevenaar (Eds.) *Security with Noisy Data.* Springer 2007.

[8] U. Rührmair, J. Sölter, F. Sehnke. *On the Foundations of Physical Unclonable Functions.* Submitted, 2009. Available from http://eprint.iacr.org/

[9] C. Ellison and B. Schneier: *Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure.* Computer Security Journal, v 16, n 1, 2000, pp. 1-7.

[10] Richard P. Feynman, *Simulating Physics with Computers.* International Journal of Theoretical Physics, Vol. 21, No. 6&7, pp. 467–488, 1982.

[11] S. Halevi, H. Krawczyk: *MMH: Software Message Authentication in the Gbit/Second Rates.* FSE 1997: 172-189

[12] Gerald DeJean, Darko Kirovski: *RF-DNA: Radio-Frequency Certificates of Authenticity.* CHES 2007: 346-363.

[13] P. C. Kocher, J. Jaffe, B. Jun: *Differential Power Analysis.* CRYPTO 1999: 388-397

[14] J.-J. Quisquater and D. Samyde: *Electromagnetic analysis (EMA): Measures and counter-measures for smart cards.* Springer LNCS Vol. 2140, pp. 200 – 210. Springer, 2001.

[15] Ross Anderson: *Security Engineering: A Guide to Building Dependable Distributed Systems.* John Wiley & Sons. Second Edition, 2008. ISBN 9780470068526, 0470068523.

[16] U. Feige, A. Fiat, A. Shamir: *Zero-Knowledge Proofs of Identity.* J. of Cryptology 1(2): 77-94 (1988)

[17] World Economic Forum, Davos, 2004.

[18] J. 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. 236, p. 475, 2005.

[19] Catherine Taylor Clelland, Viviana Risca, Carter Bancroft: *Hiding Messages in DNA Microdots.* Nature Vol 399, pp. 533 – 534, 1999.

[20] Simmons G J: *Identification of data, devices, documents and individuals.* Proc 25th Ann. Intern. Carnahan Conf. on Security Technology, pp 197 - 218, Taipei, Taiwan, ROC, IEEE (Oct. 1991).

[21] Christos H. Papadimitriou: *Computational Complexity.* Addison Wesley, 1994.

[22] Youry Kariakin: *Authentication of Articles.* Patent writing, WO/1997/024699, available from http://www.wipo.int/pctdb/en/wo.jsp?wo=1997024699, 1995.

[23] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich and T. Kalker: *Digital Watermarking and Steganography (Second Edition)*, Morgan Kaufmann, 2008.

[24] Andrew Chi-Chih Yao: *Classical physics and the Church-Turing Thesis.* Journal of the ACM 50(1), 100-105, 2003.

[25] Scott Aaronson: *NP-complete Problems and Physical Reality.* Electronic Colloquium on Computational Complexity (ECCC), 026, 2005.

[26] Peter W. Shor: *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* SIAM J. Comput. 26(5): 1484-1509 (1997)

[27] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer New York, 2006.

[28] L. Bottou, *Large-scale Kernel Machines.* MIT Press, 2007.

[29] Stepeh G. Lipson: *Optical Physics.* 3rd ed., Cambridge University Press, 1995. ISBN 0-5214-3631-1.

[30] D. Zhou, L.J. Mawst: *Two-dimensional phase-locked antiguided vertical-cavity surface-emitting laser arrays.* Applied Physics Letters, 2000.