# An Attack on PUF-based Session Key Exchange, and a Hardware-based Countermeasure: Erasable PUFs

Ulrich Rührmair [†] [⋆], Christian Jaeger [◇], and Michael Algasinger [◇]

[†] Computer Science Department
[◇] Walter Schottky Institut
Technische Universität München
85748 Garching
ruehrmai@in.tum.de,
{christian.jaeger,michael.algasinger}@wsi.tum.de
http://www.pcp.in.tum.de

**Abstract.** We observe a security issue in protocols for session key exchange that are based on Strong Physical Unclonable Functions (PUFs). The problem is illustrated by cryptanalyzing a recent scheme of Tuyls and Skoric [1], which has been proposed for use in a bank card scenario. Under realistic assumptions, for example that the adversary Eve can eavesdrop the communication between the players and gains physical access to the PUF twice, she can derive previous session keys in this scheme. The observed problem seems to require the introduction of a new PUF variant, so-called *"Erasable PUFs"*. Having defined this new primitive, we execute some first steps towards its practical implementation, and argue that Erasable PUFs could be implemented securely via ALILE-based crossbar structures.

## 1 Introduction

**Motivation and Background.** Electronic devices have pervaded our everyday life, making them a well-accessible target for adversaries. Classical cryptography offers several measures against the resulting security and privacy problems, but they all rest on the concept of a secret binary key: They presuppose that the electronic devices can contain a piece of information that is, and remains, unknown to an adversary. However, this requirement can be 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.

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 $R_i$. Contrary to standard digital systems, a PUF's responses shall depend on the nanoscale structural disorder present in it. It is assumed that this disorder cannot be cloned or reproduced exactly, not even by the PUF's original manufacturer, and that it is unique to each PUF. This means that any PUF $S$ implements

[⋆] Corresponding author.

an individual function $F_S$ mapping challenges $C_i$ to responses $R_i$. The tuples $(C_i, R_i)$ are thereby often called the *challenge-response pairs (CRPs)* of the PUF.

Due to its complex internal structure, a PUF can avoid some of the shortcomings associated with digital keys. 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 include schemes for identification [2], key exchange [1], or digital rights management purposes [3] [4]. Another advantage of (Strong) PUFs is that they can lead to protocols whose security does not depend on the usual, unproven number theoretic assumptions (such as the factoring or discrete logarithm problem), but rests on independent hypotheses.

**Strong PUFs and Weak PUFs.** Two important subtypes of PUFs, which must explicitly be distinguished in this paper, are *Strong PUFs* [1] and *Weak PUFs* [2]. This distinction has been made first in [3], and has been elaborated on further in [5] [6] [7].

*Strong PUFs* are PUFs with a very large number of possible challenges. The adversarial ability to apply challenges to them and to read out their responses from them is usually not restricted. Their central security features are: (i) It must be impossible to physically clone a Strong PUF, i.e. to fabricate a second system which has the same challenge-response-behavior as the original PUF. This restriction must hold even for the original manufacturer of the PUF. (ii) Due to the very large number of possible challenges and the PUF's finite read-out rate, a complete measurement of all challenge-response pairs (CRPs) within a limited time frame (such as several days or even weeks) must be impossible. (iii) It must be difficult to numerically predict the response $R_i$ of a Strong PUF to a randomly selected challenge $C_i$, even if many other challenge-response pairs are known.

A complete formal specification of Strong PUFs is laborious and besides the scope of this paper, but can be found in [6]. Examples of candidates for Strong PUFs are complex optical scatterers [2] or special, delay-based integrated circuits [8] [9] [10] (albeit several of the latter have been broken up to a certain size in recent machine learning attacks [5]). Also analog circuits have been proposed recently [11].

*Weak PUFs* may have very few challenges — in the extreme case just one, fixed challenge. Their response(s) $R_i$ 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

[1] Strong PUFs have also been referred to as Physical Random Functions [8] [9], or (almost equivalently) as Physical One-Way Functions [2] in the literature.

[2] Weak PUFs have also been referred to as Physically Obfuscated Keys (POKs) [12]. Note that the predicate "Weak" is not meant to state that these PUFs are "bad" in any sense, we just follow the terminology introduced in [3].

EEPROM. Typical examples of Weak PUFs are the SRAM PUF [3], Butterfly PUF [4] and Coating PUF [13].

*Applications of Strong PUFs.* We are mostly concerned with Strong PUFs and variants thereof in this paper, whence we focus on them from now on. The archetypical application of Strong PUFs is the identification of entities over insecure networks. It has already been suggested in the first PUF publication [2] by the example of a bank card scenario, and works along the following lines. Each customer's bank card contains an individual Strong PUF. Before issuing the card, the bank measures several of the PUF's CRPs, and stores them secretly on its server. When the customer inserts his card into a terminal, the terminal contacts the bank. The bank chooses at random several challenges $C_i$ from its secret CRP list, and sends them to the terminal. The terminal obtains the corresponding responses $R_i$ from the PUF, and returns them to the bank. If they match the values in the CRP list, the bank considers the card as genuine. The scheme has the upsides of circumventing the need for secret keys or secret information on the vulnerable bank cards, and of avoiding the usual, unproven complexity theoretic assumptions of classical identification protocols.

A second, central application of Strong PUFs, which also has already been suggested in [2] (page 2029), and which has been worked out in greater detail in [1], is the distribution of a secret key between different parties, for example the terminal and the bank. We are mainly concerned with this second application in this paper.

**Our Contributions.** Our first contribution is to observe a problem in the repeated PUF-based session key exchange. We illustrate this problem by the example of a recent protocol by Tuyls and Skoric [1], which has originally been suggested for use in a bank card scenario. We show how to cryptanalyze this protocol under the presumptions that an adversary can eavesdrop the communication between the terminal and the bank, that he has got access to the PUF more than once, and that no secret digital information can be stored on the card. These presumptions seem very natural, even more so in the original application scenario of bank cards or credit cards (see section 2). The problem which our attack exploits is that the CRP-information used to derive a key remains present in the PUF after the completion of the key exchange protocol.

Second, we reason that the described problem cannot be solved via protocol or software measures, and also not on the basis of current PUF architectures. Resolution seems to require the introduction of a new PUF variant, so-called Erasable PUFs. They are a special type of Strong PUF, with the additional feature that the value of single responses can be erased or altered without affecting the value of all other responses. We specify this new primitive, and show how it can be used to fix the above security issues.

Third, we suggest one possible implementation strategy for Erasable PUFs: Large, monolithic crossbar arrays of diodes with random current-voltage characteristics. It has already been demonstrated in earlier work that such crossbar arrays can act as secure Strong PUFs [14] [15] [16]. We now show that the information stored in the diodes of the crossbar can be erased individually: By applying dedicated voltage pulses to selected crossbar wires, the current-voltage curve of any single diode can be altered individually, and without affecting the other diodes in the array. We present measurement

data from single ALILE-diodes fabricated in our group that supports the feasibility of the described approach.

**Related Work.** There is no related work concerning the cryptanalysis of the Strong PUF-based session key exchange protocol by Tuyls and Skoric. In general, the cryptanalysis of PUF-based protocols appears to be a relatively recent field. Previous PUF attacks mainly focused on breaking the security properties of PUFs themselves (for example by modeling Strong PUFs via machine learning techniques [5]), but not on analyzing PUF protocols.

With respect to Erasable PUFs, there is obviously a large body of work on Strong PUFs and Weak PUFs, but none of them explicitly considered the property of erasing individual CRPs without affecting other CRPs. The category of PUFs which comes closest to Erasable PUFs are Reconfigurable PUFs (r-PUFs) [17], but the previously proposed optical, scattering-based implementation of r-PUFs has the property that inevitably *all* CRPs are altered by the reconfiguration operation. No erasure or alteration on a single CRP level is enabled. See also section 4 for a further discussion.

**Organization of the Paper.** In Section 2, we illustrate a security problem occurring in PUF-based key establishment protocols. Section 4 discusses the implementation of Erasable PUFs via crossbar structures. Section 4 describes a few obstacles in the practical realization of Erasable PUFs. Section 5 gives some background on the recent concept of a Crossbar PUF. Section 6 describes how information can be erased from Crossbar PUFs, implementing Erasable PUFs. We conclude the paper in Section 7.

## 2 A Problem with PUF-based Session Key Establishment

### 2.1 The Protocol of Tuyls and Skoric

A specific Strong PUF-based protocol for combined identification and session key establishment has been suggested recently in [1]. It is illustrated in Fig. 1. The protocol is run between a Bank on the one hand and an Automated Teller Machine (ATM) plus a security token carrying the Strong PUF on the other hand. It presumes that all involved parties have knowledge of a public one-way hash function $h$, and of a publicly known error correction scheme, which is used to derive secrets $S$ from a given noisy PUF-response $R$ and helper data $W$.

The protocol presupposes a set-up phase, in which the bank has got direct access to the Strong PUF. The bank first of all establishes a (large) secret list of the form $\{C_i, W_i, S_i'\}$. Thereby the $C_i$ are randomly chosen challenges, $W_i$ denotes helper data that is generated by the bank from the corresponding (noisy) responses $R_i$ of the PUF, and $S_i'$ refers to secret information that is derived from the noisy response by use of the helper data. Furthermore, the bank chooses a secret numerical value $x$ at random, and writes $h(x)$ onto the card. After that, the card is released to the field.

Each subsequent execution of the protocol is run between the bank and the ATM/PUF. At the beginning of the protocol, the token stores the number $n$ of previous protocol executions, the value $m = h^n(x)$, and an identification number of the Strong PUF, denoted as $ID_{PUF}$.

The Bank initially holds a list of the form $\{C_i, W_i, S_i'\}$ that is stored together with $ID_{PUF}$ in the Bank's database. The value $n'$ says how often the Bank has been engaged in a session key exchange protocol with the PUF, and $m' = h^{n'}(x)$. The rest of the protocol is described in Fig. 1, which is essentially taken from [1]. At the end of the protocol, the Bank and the ATM/PUF have established a joint session key $K$.

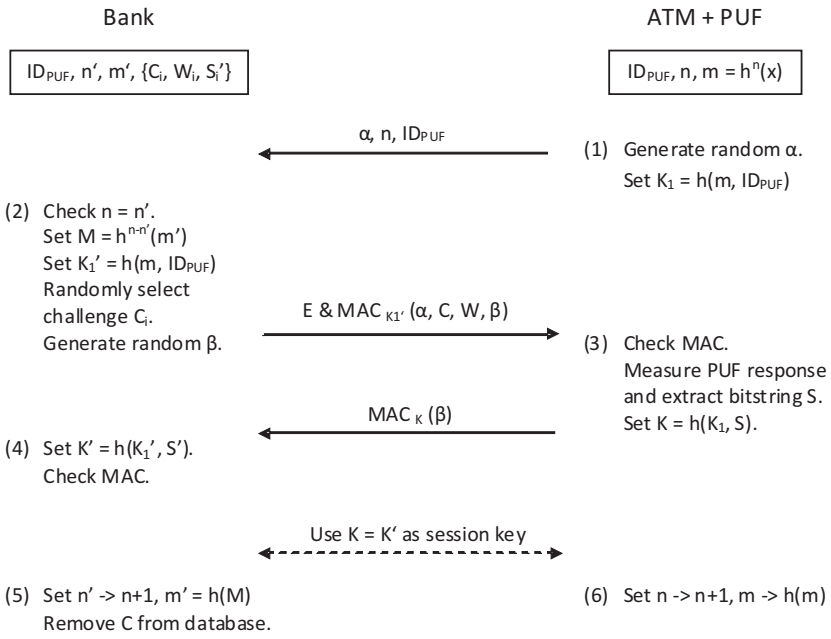


**Fig. 1.** A protocol for combined identification and session key exchange based on Strong PUFs, which has been suggested by Tuyls and Skoric in [1].

### 2.2 Problems Arising from Repeated Access to the PUF

We will now present an attack on the repeated use of the above protocol, which allows Eve to derive previous session keys.

The attack makes the following assumptions: (A) Eve can eavesdrop the communication between the bank and the ATM/PUF. (B) No secret digital keys or other secret digital values can be stored safely on the security token. (C) Eve gains access to the security token at least twice, and can measure selected CRPs from the Strong PUF on the token. All of these assumptions are relatively well motivated: If a secure channel would be at hand, which cannot be eavesdropped by Eve, then no complicated session key exchange protocol is necessary. The secret keys could simply be exchanged by sending them over this channel. Likewise, if we were to assume that secret keys could be

stored safely on the token, then the use of PUFs is unnecessary: The token could execute all necessary communication securely via classical, secret key based cryptography. Finally, assumption (C) is straightforward: For example in a bank card scenario, where an adversary might operate with faked terminals/readers that are under his control, and where the card is inserted multiple times into these terminals/readers. Again, if we do not allow an adversary to obtain physical access to the card, then the use of PUFs is unnecessary in the first place.

Eve's attack works in three successive phases executed at times $T_1, T_2$ and $T_3$.[3] In the first phase at time $T_1$, we presume that Eve has got access to the token according to assumption (C). By assumption (B), she can read the current values of $n$ and $m$ at time $T_1$ from the token, denoted by $n(T_1)$ and $m(T_1)$.

In the second attack phase at time $T_2$, we assume that Eve eavesdrops a session key establishment protocol between the bank and the ATM/PUF. This is possible according to assumption (A). From the first message sent in the protocol, which we denote by $\alpha(T_2), n(T_2), ID_{PUF}$, Eve learns the current counter value $n(T_2)$. Since Eve already knows $n(T_1)$ and $m(T_1)$ from phase 1, she can deduce the current state $m(T_2) = h^{n(T_2)}(x) = h^{n(T_2)-n(T_1)}(m(T_1))$. This allows her to derive the value of the preliminary key $K_1$ at time $T_2$ by setting $K_1(T_2) = h(m(T_2), ID_{PUF})$. Now, when the bank sends the protocol message $E\&MAC_{K_1'(T_2)}(\alpha(T_2), C(T_2), W(T_2), \beta(T_2))$, Eve can remove the encryption, because she knows $K_1(T_2) = K_1'(T_2)$. She learns $C(T_2)$ and the helper data $W(T_2)$. This closes Eve's contribution in the second attack phase. In the further course of the protocol (and without Eve's involvement), the ATM/PUF measures the PUF and extracts a secret bitstring $S(T_2)$ from its responses. Finally, the ATM/PUF sets the session key to be $K(T_2) = h(K_1(T_2), S(T_2))$.

In the third attack phase at time $T_3$, we assume that Eve has got access to the security token and the Strong PUF, and that she can measure CRPs of the Strong PUF. This is in accordance with assumption (C). Eve uses this ability to measure the PUF's responses $R(T_2)$ that correspond to the challenge(s) $C(T_2)$. Note that the Strong PUF's responses are time invariant and are not actively altered by any protocol participant. Hence Eve can determine $R(T_2)$, even though the time has progressed to $T_3$ at this point. Eve also knows $W(T_2)$, whence she can derive $S(T_2)$ from the responses $R(T_2)$. This enables her to compute $K(T_2) = h(K_1(T_2), S(T_2))$, since she knows $K_1(T_2)$ already. In other words, Eve obtains the session key $K(T_2)$ that was derived and used at time $T_2$, breaking the protocol's security.

### 2.3 Consequences for CRP Refreshment and Identification

It has been suggested in [1] that a session key $K$ established via the protocol of Fig. 1 could be used to achieve CRP refreshment between the ATM and the Bank. To that end, the ATM would, in regular intervals, execute the following steps: (i) Measure new data of the form $\{C_i(T_j), W_i(T_j), S_i'(T_j)\}$ (where $T_j$ can be an arbitrary point in time).

[3] In the description of our attack, we will need to consider the value of various protocol parameters, such as $n$, $m$, or $K_1$, at different points in time. To avoid confusion, we use the notation $n(T)$, $m(T)$, $K_1(T)$ (or similar expressions) to denote the values of $n, m$ or $K_1$ at time $T$.

(ii) Exchange a session key $K(T_j)$ via the protocol of Fig. 1. (iii) Send the encrypted message $E\&MAC_{K(T_j)}\{C_i(T_j), W_i(T_j), S'_i(T_j)\}$ to the Bank. (iv) The Bank decrypts this message, and adds $\{C_i(T_j), W_i(T_j), S'_i(T_j)\}$ to its CRP list. This process is termed CRP refreshment. This method allows shorter CRP lists and saves storage requirements on the bank.

But in the attack scenario described in section 2.2, i.e. under the provisions (A) to (C), Eve can break this scheme. First, she can apply the attack described in section 2.2 to obtain $K(T_j)$. She can then decrypt the message $E\&MAC_{K(T_j)}\{C_i(T_j), W_i(T_j), S'_i(T_j)\}$, and hence learns the values $\{C_i(T_j), W_i(T_j), S'_i(T_j)\}$ that were intended for CRP refreshment. This enables her to impersonate the PUF in subsequent identification protocols that are built on these CRP values. For example, it allows her to build faked bank cards.

### 2.4 Generality and Difficulty of the Problem

The problem we observed in the previous sections does not only apply to the protocol of Fig. 1. It could be argued that any PUF-based protocol for key establishment between a central authority and decentral principals (terminals, hardware, etc.) involves, explicitly or implicitly, the basic procedure that is shown in Fig. 2.

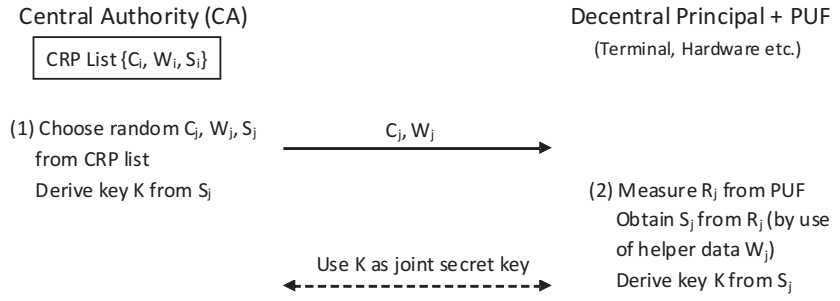


**Fig. 2.** The "raw", basic building block for PUF-based key exchange. In practice, it can and will usually be accompanied by other measures, such as message authentication or authentication of the physically transferred PUF.

Any protocol of this form is prone to the type of attack described in section 2.2. Considering the protocol of Fig. 2 sheds light on the heart of the problem: Eve can break the protocol by firstly eavesdropping the $C_j, W_j$. Subsequent one-time access to the PUF allows her to measure the corresponding $R_j$ and to derive the corresponding $S_j$. This enables her to obtain $K$. We will not give a full formal proof of this statement, but believe that adapted variants of this simple attack can be mounted on any Strong PUF-based session key exchange. One example for such an adapted attack on a much more complicated protocol was given in Sec. 2.2. The key issue in all cases seems that the response information used for the derivation of $K$ is still extractable from the Strong PUF at later points in time.

It would be hence necessary to "erase" the responses $R_j$ from the Strong PUF after they have been used for key derivation. Note that in such an "erasure" operation, all other responses $R_i$ (with $i \neq j$) must remain unchanged: If they were altered, the list $\{C_i, W_i, S_i\}$ stored at the central authority would no longer be valid. It could neither be used for further key establishment protocols of the above type, nor for the typical PUF-based identification schemes (see Sec. 1).

## 3 Erasable PUFs

We will now make some first steps work towards a hardware-based solution of the above security problem, introducing a new variant of Strong PUFs: So-called Erasable PUFs. For reasons of clarity and unambiguity, we slightly deviate from the established notation for PUFs in the following specification, and denote the response of a PUF $S$ to a challenge $C$ by $R_C^S$.

**Specification 1** (ERASABLE PUFS)**.** *A physical system $S$ is called an* ERASABLE PUF *if it is a Strong PUF with the following additional properties:*

- *There is a special, physical erasure operation $\mathbf{ER}(\cdot)$. It takes as input a challenge $C_0$ of $S$. It turns $S$ into a system $S'$ with the following properties:*
  - *$S'$ has got the same set of possible challenges as $S$.*
  - *For all challenges $C \neq C_0$, it holds that $R_C^{S'} = R_C^S$ .*
  - *Given $S'$ and $C_0$, it is impossible to determine $R_{C_0}^S$ with a probability that is substantially better than random guessing.*

Note that Specification 1 is not meant to be a full-fledged formal definition, but shall mainly express the properties of Erasable PUFs in a compact, semi-formal manner. Its style follows [6].

Given the discussion of the previous sections, it is now relatively straightforward to fix the security issues of the protocols of Fig. 1 and 2.

1. PROTOCOL OF FIG. 1: Use an Erasable PUF in the protocol, and add the erasure operation $\mathbf{ER}(C)$ at the end of step (3).
2. PROTOCOL OF FIG. 2: Use an Erasable PUF in the protocol, and add the erasure operations $\mathbf{ER}(C_j)$ to the end of step (2).

These steps disable the attacks that have been presented in the previous sections: When Eve has got access to the PUF at a later point in time, she can no more determine the PUF responses used for previous key derivation, as the responses have been erased from the system.

## 4 Obstacles in the Implementation of Erasable PUFs

The implementation of Erasable PUFs on the basis of established PUF architectures turns out to be intricate; we will summarize the occurring difficulties in this section. One reason for the appearing problems is that Erasable PUFs must combine the following properties:

(i) They must be Strong PUFs, i.e. they must have very many possible challenges, and must be able to withstand full read-out for long time periods, i.e. weeks or months.
(ii) They must allow the erasure or alteration of single responses, without affecting other responses.

These properties rule out Weak PUFs and their current implementation candidates [3] [4] [13] from the start, since they simply do not fulfill condition (i) above, i.e. they are no Strong PUFs.

An alternative approach would be to modify Strong PUF architectures in order to obtain Erasable PUFs. The erasure operation could, for example, alter some internal components of a Strong PUF. But unfortunately, all popular candidates for Strong PUFs [2] [8] [9] [10] [11] create their responses in a complex interplay of many or even all internal components. Altering one single component will not only change a single response, but will affect many other responses, too. Their responses cannot be altered individually, i.e. with single CRP granularity.

Another, straightforward idea would be to attach an access control module to a Strong PUF. The module could store a list of "forbidden" challenges and prevent the application of these challenges to the Strong PUF. But this approach is costly in practice: It requires non-volatile memory, which must store potentially large amounts of challenges. Furthermore, it cannot reach ultimate security levels: The control module might be circumvented or cut off by a well-equipped attacker, and the content of the memory (i.e. the forbidden challenges) might be manipulated.

The existing concept that presumably comes closest to Erasable PUF are Reconfigurable PUFs (r-PUFs), which were introduced in [17]. By definition, each r-PUF possesses a reconfiguration operation, in which all CRPs of the r-PUF can be changed. However, the currently suggested optical implementation of r-PUFs has the property that all responses are altered by the reconfiguration operation, disabling it as an Erasable PUF. For electrical implementations of r-PUF based on phase-change materials, which are only briefly mentioned asides in [17], it is yet unclear whether they would be Strong PUFs at all, i.e. whether they could be designed to withstand full read-out in short time.

Eventually, there is one recent Strong PUF candidate that seems appropriate to implement Erasable PUFs: So-called Crossbar-based PUFs. They have originally been introduced in [14] [15] [16], and will be treated in the next section.

## 5 Strong PUFs based on Crossbar Structures

Recent work [14] [15] [16] investigated the realization of a special type of Strong PUF (so-called "SHIC PUFs" [4]). These are Strong PUFs with the additional following properties:

(i) The PUF possesses maximal information content and density, with all CRPs being mutually (i.e. pairwise) information-theoretically independent.
(ii) The PUF can only be read out at slow rates.

[4] SHIC abbreviates the term "Super-High Information Content", and is pronounced as *"chique"*.

The motivation behind investigating this type of Strong PUFs was to protect PUFs against any modeling attacks. Such attacks use known CRPs in order to extrapolate the PUF's behavior on new CRPs, and constitute a serious challenge for the security of Strong PUFs [5]. SHIC PUFs are automatically invulnerable against such modeling attempts, since all of their CRPs are information-theoretically independent: Knowing a subset of CRPs hence does not allow conclusions about other CRPs.

Concrete target parameters for the construction of SHIC PUFs discussed in [14] [15] [16] were an information content of up to $10^{10}$ bits and read-out speeds of $10^2$ to $10^3$ bits per second. As argued in [15], such relatively slow read-out speeds are no problem in many typical applications of Strong PUFs, such as bank card identification, key exchange, or also oblivious transfer [18]. On the upside, the combination of slow read out and high information content can potentially immunize the PUF against full read-out for up to month or years of uninterrupted, unnoticed adversarial access [15]. For comparison, several known Strong PUF architectures with a MHz read-out rate can be modeled (and hence broken) via a number of CRPs that can be read out in a few seconds [5].

It has been shown in [14] [15] [16] that SHIC PUFs can be realized by large, monolithic crossbar architectures. At each crosspoint of the crossbar, a diode with a random current-voltage characteristic is present. The necessary random variation in the diodes is generated by a random crystallization technique known as ALILE process. We will review the necessary basics of this approach in this section; much further detail can be found in [14] [15] [16].

*ALILE Crystallization.* In order to construct a Strong PUF with the above properties, one first requires a solid-state fabrication process that generates a maximal amount of entropy in the PUF. The authors of [14] [15] [16] turned to crystallization processes to this end, since the crystallization step amplifies minuscule variations in the starting conditions (such as atomic-scale roughness) to larger, stable variations in the system (for example the shape, size and position of the crystallites). Among many possible crystallization processes, they eventually selected the so-called aluminum-induced layer exchange (ALILE) process [20] [21], since it is a simple crystallization process that involves few production steps and inexpensive starting materials. It results in polycrystalline films with p-type conduction [22], and creates a highly disordered and random structure comprising of crystallized silicon grains (Si) and aluminum (Al). Fig. 3 a depicts the top view onto a crystallized system, illustrating the occurring randomness. By changing the process parameters, the size and density of the grains can be tuned as desired.

*Diodes and Crossbar Read-Out.* In order to read out the information contained in the system, a circuit architecture known as crossbars can be employed. It consists of two sets of parallel wires, one of them applied on the top, the other one at the bottom of the structure. Both sets are arranged orthogonally to each other. The basic schematics are illustrated in Fig. 3 b. Due to the p-n-type cross section of the entire system (the film of p-type conduction is generated on an n-type wafer to this end), each virtual crossing of the crossbar acts like a p-n-diode, with rectification rates of up to $10^7$ [16]. Its $I(V)$ curve can be read out by applying a voltage at two chosen crossbar wires (bit and word

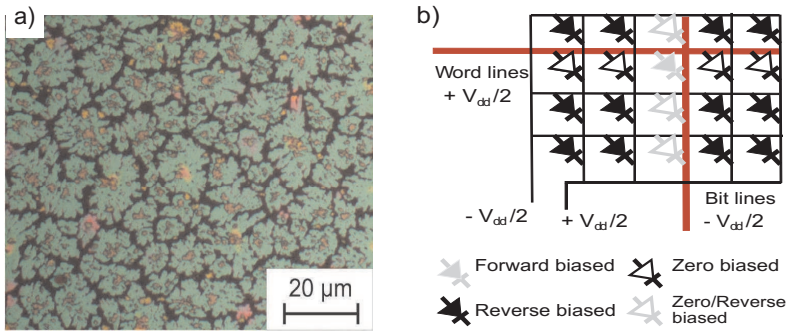


**Fig. 3.** (a) Randomly shaped and located Si crystallites (top view, showing the extension in $x$-$y$-directions). (b) Schematic illustration of the crossbar architecture and the diodes at the crossings. Also read-out process, i.e. the selection of a bit line and a word line in order to address and read out a single diode, is illustrated.

lines, in analogy to a memory), as illustrated in Fig. 3 b [15]. Due to the random nature of the ALILE crystallization process, the diodes show current-voltage curves which are very irregular and individual in shape. The individual curves differ in their currents by up to four decimal orders of magnitude, but are still stable against aging and multiple measurement [14] [16]. As shown in [14], at least three bits of information can be extracted reliably from each crossing.

*Information Content and Inherently Slow Read-Out Speed.* Using crossbar architectures has two advantages. First, they can reach ultimate information densities due to their very simple architecture of parallel wires. The information density and content targeted in [15] were $10^{10}$ bits per cm$^2$. Secondly, they can be designed with an inherently limited read-out speed. To achieve this, the Crossbar PUF is built in one large, monolithic block, not from separate blocks as modern semiconductor memories, and is made from wires that have only finite current-carrying capacity. Simulations conducted in [15] showed that in such large monolithic blocks, several milliseconds must elapse before the sense current/voltage stabilizes. This results in read-out speeds of around 100 bits/sec. Any faster read-out attempts would overload and destroy the wires, leaving the remaining structure unusable [15].

## 6 Erasing Information from Crossbar Structures

We now investigate if – and how – information can be erased from Crossbar PUFs. Since the information is contained in the diodes' current-voltage characteristics, any erasure operation must target the diodes, changing their $I(V)$-curves irreversibly. We could not build a device with $10^{10}$ crossings within the scope of this paper, but argue on the basis of measurement curves obtained from stand-alone fabricated in our group. The fact that the behavior of these single diodes scales very well to the operation of large, monolithic blocks of diodes has been proven in all detail in earlier work [15].

The “erasure operation” works as follows. A specific diode in the crossbar array is chosen by selecting the corresponding bit and word lines of the crossbar structure, similar to the read-out procedure for the crossbars. Then a short voltage pulse of 4 V to 5 V is applied in reverse direction to the diode. This induces a breakdown in the ALILE diode, which destroys the individual information present in the $I(V)$ curve, and makes all curves after erasure “standardized” and very similar in shape.

This effect has been observed by us in *all measured diodes*; three illustrative examples for $I(V)$-curves before and after breakdown are shown in Fig. 4. While the large variations in the original curves range over four orders of magnitude, there is little individuality left after breakdown. The curves after breakdown also differ strikingly from the original curves. Considering the development of the relative positions of the curves over the full voltage range shows that not even the relative positioning of the curves is preserved.

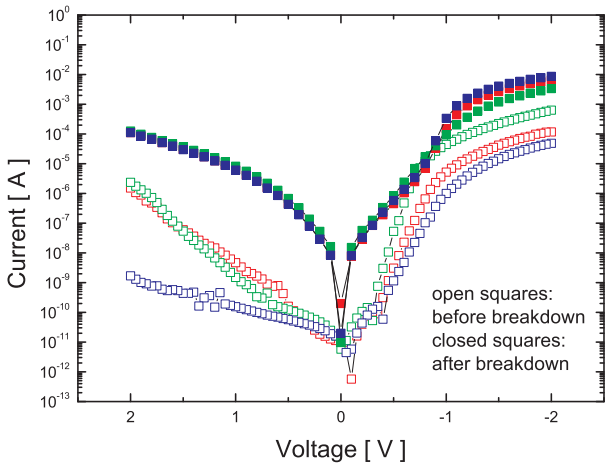


**Fig. 4.** The curves of three exemplary diodes (red, blue and green) before and after breakdown.

The fact that the new curves are uncorrelated to the old ones is a consequence of the physical effect behind the breakdown of the diodes. Our explanation of this mechanism is the presence of a thin natural oxide film between the p- and n-layers, effectively resulting in a p-i-n-structure. Such an additional i-layer would strongly reduce the tunneling current in reverse direction (as observed by us), which otherwise had to be expected to be high due to the large hole carrier concentration in the ALILE layers (up to $10^{19}$ $\mathrm{cm}^{-3}$) [16]. The assumption of an intermediate oxide layer is further supported by the fact that diodes which were exposed to hydrofluoric acid (HF) vapor prior to the deposition of the ALILE layers *did not show* comparable rectification rates; the HF vapor is known to remove Si-oxide, leading to a destruction of the possible p-i-n -structure [23]. The described voltage pulse in reverse direction then simply burns and removes this i-layer.

This physical mechanism behind the erasure supports the security of our construction, for the following reasons: First, the destruction of the thin, irregular oxide film cannot be reversed physically by Eve. Second, after the oxide layer has been removed, independent and secondary features of the structure dominate the $I(V)$ curve (whereby their effect on the randomness of the curve is by far not as strong as the original configuration, see Fig. 4). From knowing the new curves after breakdown, it is therefore impossible to conclude backwards on the shape of the original $I(V)$ curves before breakdown.

Finally, please note that the operational voltage for measurement of the diodes in the crossbar structure lies between -2V and +2V. The erasure operation hence is just a factor of around 2 away from the standard operation of the crossbar. This is compatible with the use of wires with finite current-carrying capacity, which was indispensable to enforce the slow read-out rate of the crossbar (see Section 5, page 11, and [15]).

## 7 Summary

We made the following contributions in this paper. First, we observed a security problem in a recently published session key exchange protocol by Tuyls and Skoric [1], which is based on Strong Physical Unclonable Functions (PUFs). We cryptanalyzed the protocol under the relatively mild presumptions that the adversary gains access to the PUF twice, that she can eavesdrop the communication between the involved parties, and that no secret information can be stored on the card. As discussed earlier, these presumptions are well-motivated, for example in the bank card scenario in which the protocol had been proposed originally. Our attack has severe consequences for the security of the proposed bank card application. The noted security problem seems to be general, applying to any comparable session key exchange based on Strong PUFs.

Second, we introduced a new PUF variant, so-called Erasable PUFs, in order to resolve the described security issue. These are special Strong PUFs, with the additional property that the information stored in single responses of theirs can be irreversibly erased without changing any other response values. As we argued, currently known PUF architectures are unsuited to this end: They either are no Strong PUFs in the first place. Or, they have many interplaying components, which prevents that a single response can be changed without affecting the other responses. The latter problem holds for all delay-based PUFs, but also for the current, optical implementations of Reconfigurable PUFs.

We therefore, thirdly, investigated new architectures for implementing Erasable PUFs. We suggested the use of crossbar structures with randomly crystallized ALILE-diodes. It was known from recent work [14] [15] [16] that such “Crossbar PUFs” can act as Strong PUFs with very high information content and densities and inherently slow read-out speed. We now discussed how the information stored in the ALILE-diodes of the crossbar can be erased individually. Our erasure process works by applying a relatively small threshold current to selected bit and word lines of the crossbar. This induces a “breakdown” in the diode, as it burns intermediate oxide layers. The process is irreversible, and transforms the individual $I(V)$ curve of any diode into an uncorrelated, new one. The threshold current is low enough to be compatible with the finite current carrying capacity of the crossbar wires and the read-out mechanism of the crossbar ar-

ray. We supported our proposal by measurements on single, stand alone ALILE-diodes fabricated in our group. It had been shown in extensive simulations in previous work [15] that the behavior of such diodes scales to large diode arrays.

## Acknowledgements

This work was conducted in the course of the Physical Cryptography Project at the TU München, with support by the Institute for Advanced Study (IAS) and International Graduate School of Science and Engineering (IGSSE) at the TU München.

## References

1. P. Tuyls, B. Skoric: *Strong Authentication with Physical Unclonable Functions*. In: Security, Privacy and Trust in Modern Data Management, M. Petkovic, W. Jonker (Eds.), Springer, 2007.
2. R. Pappu, B. Recht, J. Taylor, N. Gershenfeld: *Physical One-Way Functions*, Science, vol. 297, pp. 2026-2030, 20 September 2002.
3. Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, Pim Tuyls: *FPGA Intrinsic PUFs and Their Use for IP Protection*. CHES 2007: 63-80
4. Sandeep S. Kumar, Jorge Guajardo, Roel Maes, Geert Jan Schrijen, Pim Tuyls: *The Butterfly PUF: Protecting IP on every FPGA.* HOST 2008: 67-70
5. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber: *Modeling Attacks on Physical Unclonable Functions.* Accepted at ACM Conference on Computer and Communications Security, 2010. Previous versions available from Cryptology ePrint Archive, Report 251/2010, 2010.
6. 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.
7. U. Rührmair, J. Sölter, F. Sehnke: *On the Foundations of Physical Unclonable Functions.* Cryptology e-Print Archive, June 2009.
8. B. Gassend, D. Lim, D. Clarke, M. v. Dijk, S. Devadas: *Identification and authentication of integrated circuits.* Concurrency and Computation: Practice & Experience, pp. 1077 - 1098, Volume 16, Issue 11, September 2004.
9. J.-W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. *A technique to build a secret key in integrated circuits with identification and authentication applications.* In Proceedings of the IEEE VLSI Circuits Symposium, June 2004.
10. M. Majzoobi, F. Koushanfar, M. Potkonjak: *Lightweight Secure PUFs.* IC-CAD 2008: 607-673.
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*. IEEE CNNA - 12th International Workshop on Cellular Nonlinear Networks and their Applications, 2010.
12. Blaise Gassend, *Physical Random Functions*, MSc Thesis, MIT, 2003.
13. Pim Tuyls, Geert Jan Schrijen, Boris Skoric, Jan van Geloven, Nynke Verhaegh, Rob Wolters *Read-Proof Hardware from Protective Coatings.* CHES 2006: 369-383
14. U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, and M. Stutzmann: *Security Applications of Diodes with Unique Current-Voltage Characteristics.* Financial Cryptography and Data Security, 2010.

15. U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, and G. Csaba: *Cryptographic Applications of High-Capacity Crossbar Memories*. IEEE Transactions on Nanotechnology, 99,1, 2010.
16. C. Jaeger, M. Algasinger, U. Rührmair, G. Csaba, M. Stutzmann: *Random pn-junctions for physical cryptography.* Applied Physics Letters, Vol. 96, 172103, 2010.
17. K. Kursawe, A.-R. Sadeghi, D. Schellekens, B. Skoric, P. Tuyls: *Reconfigurable Physical Unclonable Functions – Enabling Technology for Tamper-Resistant Storage.* HOST 2009: 22-29
18. U. Rührmair: *Oblivious Transfer based on Physical Unclonable Functions (Extended Abstract)*. TRUST Workshop on Secure Hardware, Berlin (Germany), June 22, 2010. Lecture Notes in Computer Science, Volume 6101, pp. 430 - 440. Springer, 2010.
19. G. E. Suh, S. Devadas: *Physical Unclonable Functions for Device Authentication and Secret Key Generation.* DAC 2007: 9-14
20. O. Nast and S.R. Wenham: *Elucidation of the layer exchange mechanism in the formation of polycrystalline silicon by aluminum-induced crystallization.* Journal of Applied Physics, Vol. 88, pp. 124-132, 2000.
21. O. Nast and A.J. Hartmann: *Influence of interface and Al structure on layer exchange during aluminum-induced crystallization of amorphous silicon.* Journal of Applied Physics, Vol. 88, pp. 716-724, 2000.
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. 2007, Vol. 91, Page 201909.
23. R. J. Carter and R. J. Nemanich: *HF vapour cleaning of oxide on c-Si*. Properties of Crystalline Silicon, EMIS Datareviews Series No 20, University of Virginia, USA, 1999.
24. G. Majni and G. Ottaviani: *Growth kinetics of (111)Si through an Al layer by solid phase epitaxy.* Journal of Crystal Growth 46, 119, 1979.
25. Daihyun Lim: *Extracting Secret Keys from Integrated Circuits*. MSc Thesis, MIT, 2004.
26. M. Majzoobi, F. Koushanfar, M. Potkonjak: *Testing Techniques for Hardware Security.* IEEE International Test Conference, 2008.