A SYSTEM FOR MULTI-LEVEL AUTHENTICATION WITH PHYSICALLY UNCLONABLE FUNCTIONS

BY

S. T. CHODEN KONIGSMARK

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Advisers:

Professor Martin D. F. Wong
Associate Professor Deming Chen

ABSTRACT

We analyze deficiencies in existing Physically Unclonable Function (PUF) systems and protocols, and propose a new system of PUFs (SoP) that is numerically secure under extended attacker privileges and attack scenarios. Our proposed system uses a multi-level authentication scheme and employs different designs of PUF to achieve high security with low computational complexity and small footprint.

By employing role-specific PUF designs, SoP reduces the area over existing PUF-based authentication solutions by more than 68%. The key principles are: (i) reduce assumptions required to guarantee numerical security to a minimum set of practical assumptions; (ii) combine different PUF types to optimize security while minimizing resource requirements; (iii) provide multiple layers of authentication as a force-multiplier for the trusted party.

This multi-level protocol resolves security deficiencies with regard to man-in-the-middle attacks and challenge-response-pair (CRP) storage issues in conventional PUF protocols. Furthermore, SoP allows recognition and sealing of security breaches. A mathematical formulation of the attack complexity and statistical evaluation based on simulated PUF data show the strength of this new protocol.

ACKNOWLEDGEMENTS

First of all, I thank my wife and family, who have been very supportive of my graduate studies and were always there for me. I thank my advisors, Professor Martin Wong and Professor Deming Chen, for their encouragement and for their guidance. I thank them for believing in me, and for setting high expectations so that I continuously improved myself. Our discussions and brainstorming sessions have been crucial to my research progress. I thank my mentor and friend Leslie Hwang for providing her help to jump-start my graduate studies. She has been helpful in all aspects, and I have deep gratitude for her. I would also like to thank my other research group members for a great collaborative environment.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

In the age of ubiquitous computing, our society relies on computing devices at every level, and with this comes the increasing need for computer security. While we are aware of the damages that could be caused by misuse of credit-cards or RFID-tags, the emergence of ubiquitous computing has brought new issues and further increases the importance of secure authentication. With wearable technology and personal medical devices, an adversary that circumvents authentication protocols can directly impact the physical well-being of users. Moreover, such authentication failures in wireless sensor networks for border control and defense purposes can have even worse consequences. A unifying characteristic of the new device generation is their need for security with minimal resources and power consumption, and thus typically only allowing minimum resource allocation for security components.

Software and network security have gained increased attention and are widely perceived to provide the necessary means for secure communication, authentication and data storage. However, without the appropriate hardware mechanisms, no system can be secure. For example, side-channel information leakage was used to successfully attack AES implementations [1], and secret keys can be extracted from volatile memory long after the device is off [2].

Conservative security and authentication protocols are based on secure storage and usage of secret keys [3]. In the past, this meant that non-volatile memory (NVM) and fuses were used to provide a secret key [4]. However, NVM is prone to invasive physical and non-invasive imaging attacks [5]. Additionally, classic cryptographic algorithms are too complex and power intensive for many applications [5].

As a lightweight, silicon intrinsic solution to hardware security, Physically Unclonable Functions (PUFs) [2] were proposed. PUFs use device intrinsic variations in otherwise equally designed circuit elements as secret. Thus, PUFs do not exhibit the same weaknesses towards non-invasive imaging and invasive attacks. However, PUFs only provide the basic building block for security and have to be embedded into a system that can participate in an authentication protocol. Many existing protocols require expensive components that are unsuitable for low-resource applications, and furthermore do not take full advantage of the flexibility of the variety of PUF designs that past research has yielded.

In the era of ubiquitous computing, authentication systems have to be resource- and power-efficient while providing enough flexibility to target a variety of applications. In the following, we describe key specifications of our PUF based low-cost authentication system:

- Very light usage of active security components without costly hash functions on the chip or error correcting schemes.
- Low false-negative rate to reduce the time and energy spent in authentication iterations.

To address the demand for lightweight security components, the unique contributions of this paper include:

- A SoP that employs different PUF types with multiple challenge and response levels to maximally exploit the advantages of distinct PUF designs while minimizing resource allocation.
- Identification of weaknesses in existing PUF based authentication schemes and a novel authentication protocol based on the SoP that increases reliability and security.
- Usage of secret hashing at the trusted party to exponentially increase the set of challenge-response pairs that a malicious party faces without implementing these expensive functions on the PUF circuit.

- A system and protocol for authentication that reduces the area as measured in gate-equivalent units by more than 68%.

The remainder of this thesis is structured as follows. In Chapter 2, we review the background of PUF and PUF-based authentication. The main part of this thesis begins with Chapter 3, where issues in current PUF protocols are identified. In Chapter 4, we propose a new protocol and give a detailed security analysis in Chapter 5. An experimental evaluation that shows feasibility of the protocol is provided in Chapter 6. This thesis concludes with a summary in Chapter 7.

CHAPTER 2
BACKGROUND AND PREVIOUS WORK

PUFs are characterized by their challenge (input) and response (output) behavior. Typically, a PUF consists of a number of equally designed components that have slightly different physical properties due to manufacturing variations. The challenge to a PUF is used to select which PUF components are compared for their physical properties and the response is a bit or bit-string representing the outcome of pairwise comparison of the selected elements. In the following, we will first establish quality metrics for PUF, which will then be used to provide an overview of existing PUF designs. Due to their high volatility, the reliability of PUF designs can be a restriction for several applications, motivating additional circuitry for error correction. This chapter concludes with an overview of prior work on PUF based authentication.

2.1 Quality Metrics

The two main metrics of quality for PUF designs are uniqueness and reliability. We use inter-chip and intra-chip distances [6] for uniqueness and reliability to establish design criteria and implementation guidelines for SoP. Inter-chip Hamming distance HD_{inter} is a metric for the difference between manufactured PUF instances and thus shows the usability of static variations because of the manufacturing process. Ideally, the average difference between all physical instances would be close to 50%, as a ratio higher or lower than that represents a bias in the response. It is computed as the average Hamming distance between different PUF instantiations for the same set of challenges, as shown in Equation (1). Here, m represents the number of PUF instances and C is the common challenge used during uniqueness evaluation.

$$HD_{inter} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{k=i+1}^{m} HD(R_i, R_k), \quad \text{with } R_i = PUF_i(C), R_k = PUF_k(C) \tag{1}$$

Intra-chip Hamming distance HD_{intra} represents the variability of the responses to the same challenge on the same PUF instance under environment variations. It is a metric for the reliability of the PUF and lower values indicate higher reliability under dynamic variations, e.g. temperature fluctuations. Equation (2) shows the intra-chip Hamming distance for N challenges, which are issued K times. The nominal response of the PUF is $R(C_i)$, while the response under environment variations is $R'(C_i)$.

$$HD_{intra} = \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{k=1}^{K} HD\big(R(C_i), R'(C_i)\big) \tag{2}$$

$$HD_{intra} = \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{k=1}^{K} HD\big(R(C_i), R'(C_i)\big)$$

When averaged over the bit-length, we refer to the fractional Hamming distances FHD_{intra} and FHD_{inter}.

2.2 PUF Designs

In this section, we will introduce the concepts of several different PUF designs that are relevant to this work. The quality characteristics of these designs are summarized in Table 1.

SRAM-PUF leverages device-specific process variations to generate a secret from the SRAM startup state. It is a low-cost PUF that is suitable for ID or key generation [7], [8].

In the Arbiter PUF [4], shown in Figure 1, an output bit is generated from a delay comparison of two equally designed paths. The challenge is used to select a specific path by controlling multiplexers that select between wire segments of equal parameters. Rührmair et al. presented model-building techniques that successfully

predicted the behavior of an Arbiter PUF with small training data [9]. A more secure iteration is achieved by combining N Arbiter PUFs by XOR connection of the individual response bits to the N-XOR-Arbiter PUF, as shown in Figure 2. These XOR-Arbiter PUFs increase the difficulty of creating a model for the PUF, but the XOR combination also linearly reduces the reliability of the PUF [10]. It was shown that a 6-XOR Arbiter PUF is very expensive to model [9].

Table 1 Comparison of PUF Designs.

Design Comparison	**PUF Designs**	
	$\boldsymbol{FHD_{inter}}$	$\boldsymbol{FHD_{intra}}$
ClockPUF [11]	0.503	0.057
SRAM [7]	0.4997	< 0.12
Arbiter PUF	0.51	0.05
4-XOR Arbiter PUF [10]	0.51	0.19
Ring-Oscillator [4]	0.4614	0.0048

Ring-Oscillator PUFs (RO-PUFs) [4] consist of a number of ring-oscillators that are designed to be equal. Based on the challenge, two multiplexers select one ring-oscillator each and their frequency is compared using a counter for each of the oscillators. This comparison determines the output bit of the RO-PUF. RO-PUF has been shown to achieve high reliability, but is comparably power-intensive and slow as many cycles are needed to distinguish the respective oscillator frequencies. Figure 3 shows a Ring Oscillator and how a pairwise comparison of Ring Oscillator frequencies builds the RO-PUF.

Yao et al. introduced a PUF design based on the clock network of the chip that inherits the clock network's stability but introduces enough variations to show randomness [11]. They choose specific sinks in the clock network and branch the clock signal from these sinks in a return path that gathers process variations to a network of multiplexers and an Arbiter. The Arbiter can then be used to compare the delay of different paths and generate a response bit.

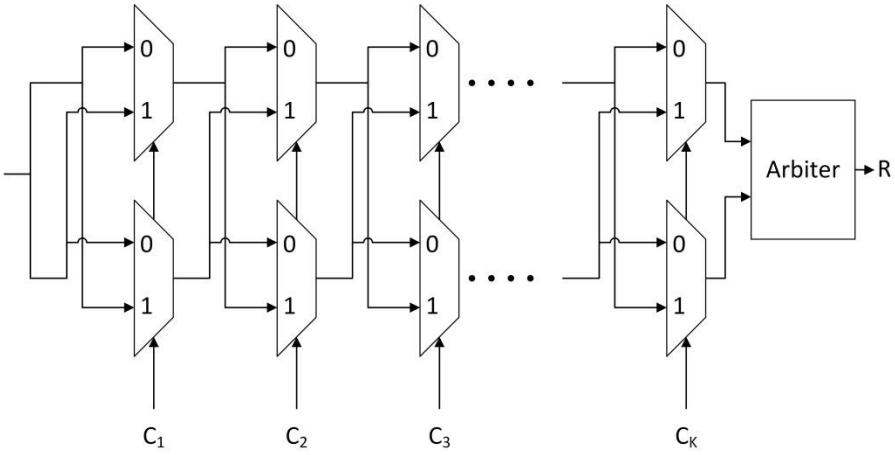

Figure 1 The Arbiter PUF determines the response R by evaluating the delay difference of two equally designed but physically different paths. The challenge to the PUF determines the path that the signals take.

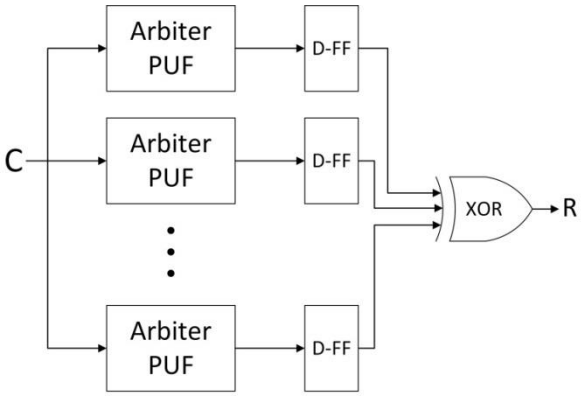

Figure 2 The XOR-Arbiter PUF consists of N parallel Arbiter PUFs. The response is generated through XOR-connection of all Arbiter PUF responses.

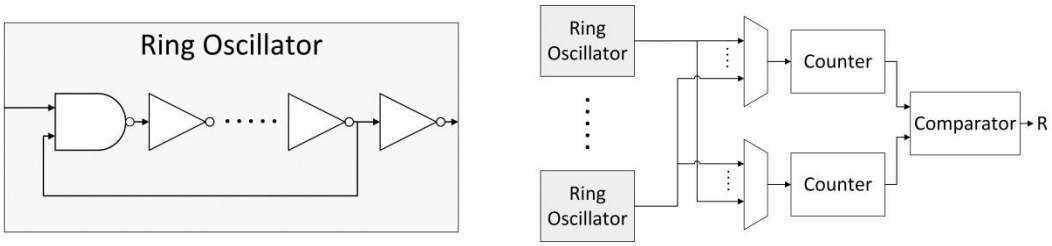

Figure 3 A Ring Oscillator consists of an inverter chain with an odd number of inverting stages (left). The challenge to the RO-PUF determines the choice of Ring Oscillator pair, and the output is the result of a frequency comparison.

2.3 Helper Circuits

Most PUF designs only generate one output bit, as two circuit elements are compared. An expensive way of extracting multiple output bits is to implement as many PUFs as the desired number of output bits. For lightweight applications, a pseudo-random number generator (PRNG) can also be used as a challenge expander, e.g. a Linear Shift Feedback Register (LFSR) [5],[12]. Based on an initial challenge as the seed, the PRNG will sequentially create new challenges and therefore multiple output bits.

As PUF elements are designed to be equal, their response is volatile and dynamic variations such as temperature or supply voltage variations can lead to noise in the PUF response. To eliminate the noise, PUFs can be used with error correcting codes (ECC) and fuzzy extractors [13]. A Ring-Oscillator PUF design employing index-based syndrome encoding (IBS) was shown to have very high reliability characteristics [13]. Further improvements over this exist, that allow a tradeoff between design complexity and error correction [14]. However, these error correcting techniques are very expensive in area and energy compared to the cost of the actual PUF, suggesting that they are less suitable for lightweight applications.

2.4 Background on PUF-Based Authentication

We use the terminology of [5] and refer to the *Prover* as a device with direct access to a PUF and the *Verifier* as a device used by the trusted party to discern whether a *Prover* contains a real PUF or is a counterfeit. An authentication protocol determines the interaction between *Prover* and *Verifier*.

To successfully authenticate the *Prover*, the *Verifier* requires previous knowledge on the PUF that is accessed by it. In the PUF challenge-response-pair (CRP) scenario, the *Verifier* typically stores information about the expected CRP behavior of the PUF [4],[5]. Two main approaches have been proposed:

Create a database of CRPs at the *Verifier* side by initially issuing a possibly limited amount of challenges to the PUF and recording the responses [4]. This is

typically done in an enrollment phase directly after manufacturing and assumes that model-building-attacks against the PUF are impossible due to high complexity [4].

Create a model of the PUF with machine learning techniques to be able to accurately estimate the PUF response to arbitrary challenges [5]. This approach has to include measures that disallow model-building for a malicious third party, e.g. by disabling direct PUF access [5].

Both of these approaches have disadvantages that can ultimately lead to infeasibility and security issues in conventional authentication protocols.

2.5 Authentication Protocols

A simple authentication protocol based on issuing random challenges with known responses is presented in [4]. The trusted party can validate the PUF responses against the known responses. To handle man-in-the-middle attacks, it is proposed to only use each challenge once.

Reverse fuzzy extractor [12] is a lightweight authentication scheme that attempts to move computationally complex or resource intensive components off the PUF-circuit to the authentication granting authority. It is based on reversing error correction schemes employed to increase PUF reliability.

Another recent approach is Public PUF (PPUF) [15], where detailed physical characteristics of each PUF instance are public, allowing anyone to simulate PUF behavior. A PUF is then verified by not only providing the correct response to a challenge, but doing so in a much shorter time than possible with simulations. As the true response can be simulated before to issuing the challenge, no previous CRP storage is required. As PPUF has high latency and power consumption, extensions with the same principle were developed [16]. Due to the computational requirements and detailed device-specific measurements, PPUF is most suitable for small-scale applications.

Slender PUF protocol is another lightweight protocol [5] that can be used for identification and authentication. The protocol has two main ideas: (i) only

substrings of responses are provided, and (ii) the challenges to the PUF are jointly created by both the *Prover* and the *Verifier*. The *Verifier* is assumed to have an ideal model of the PUF, so that a response for any possible challenge can be generated. The substring received from the *Prover* is then used to check whether it is indeed a substring of the real PUF response. The second idea is that neither party is allowed to solely generate the challenge; thus the challenge comes from a pseudo random number generator (PRNG), and the seed to this PRNG is determined by randomly generated numbers (Nonces) from both parties.

CHAPTER 3
PUF SECURITY ISSUES

In this chapter, we aim to provide a clear framework under which PUF designs should be evaluated by addressing possible security weaknesses regarding previously published PUF designs and authentication protocols. In Section 3.1, we discuss the impossible or impractical assumption of a known model for PUF designs. Then, we discuss consistency issues and memory requirements of common PUF assumptions. Finally, we investigate the lack of design diversity in current protocols, which will be the core of our proposed protocol in Chapter 4.

3.1 Known Model Assumption

Protocols for authentication based on PUFs typically require either a known model of the PUF [5], which is examined in this section, or a large set of CRPs [4], discussed in Section 3.3. Protocols such as Slender PUF Protocol assume that the trusted party can compute a true model for the PUF response, but an adversary cannot. This is justified by suggesting that the *Prover* circuit initially contains sensing capabilities that provide direct access to the PUF. After model generation, the trusted party will externally disable the direct access and thus hinder the malicious party from creating a model. As model-building attacks might be possible when the full PUF response is accessible to the outside [5],[12], the ability to disable the sensing connection is key to this assumption.

However, we recognize severe obstacles in this assumption: Physical modification of the circuit to disable the sensing capabilities after manufacturing is not scalable and may negatively influence the circuit or PUF behavior. However, logical disabling of the sensing capability cannot be considered to be secure, as an adversary could acquire the relevant knowledge to re-enable it. We identify this as a major deficiency in existing PUF-based security protocols and propose to treat

the trusted and malicious party equally. Thus, we require that modeling capabilities of the trusted party lead to the same for the adversary and present an authentication scheme that does not rely on one-sided knowledge of a true PUF model.

3.2 Security Enabled by Conventional Hardware

PUF based systems typically also contain non-PUF circuits and components. To achieve clear security principles, we introduce the concept of security enabling and functionality enabling components:

- Security enabling component: This component is required to provide security of the system. However, the system functionality does not depend on it and modifications of these components do not limit the functionality.
- Functionality enabling component: This is a key component of the system, and changes lead to malfunctioning or unexpected behavior.

Components such as a *True Random Number Generator* or a *Pseudo Random Number Generator* cannot be assumed to work as designed unless implemented with PUF, as a malicious party must be considered to be capable of invasive attacks. Employing such circuitry in a security enabling function will therefore pose a security threat. Since these conventional circuits may fail, suggested protocols such as [5] become attackable and cannot be used to guarantee security. Functionality enabling components do not pose a risk, as their modifications will render the system useless. For example, modifying the challenge-expander described in Chapter 2.3 is possible, but will modify the whole challenge-response behavior of the PUF and thus make it unusable.

3.3 Exponential Memory Requirements

When protocols do not make a known model assumption, they typically require that PUF responses can only be used once, as in [4]. This is legitimate and is a great hindrance to man-in-the-middle attacks, but also exposes the protocol to denial-of-service (DoS) attacks: A malicious third party can query the *Verifier* until

the stored CRP set is exhausted. This could be mitigated by storing a huge amount of CRPs at the *Verifier* side, but this ultimately causes a data storage problem. To withstand a DoS attack, the *Verifier* would be required to store large numbers of long CRP strings and additional synchronization bits.

3.4 Design Uniformity

Another major shortcoming that we observe in current protocols is the usage of only one type of PUF design [4][5]. This is an issue, as the wide variety of PUF designs that recent research has produced is not fully exploited. We emphasize that there exist many designs that have various advantages, such as high resistivity against model building attacks, high reliability, and low power consumption.

To deal with these various issues discussed in this chapter, we will propose a novel SoP that exploits the advantages of different PUF designs and is used in conjunction with our proposed multi-level authentication protocol in Chapter 4.

CHAPTER 4
SYSTEM OF PUFS

In this chapter, we will explain the system of PUFs (SoP) and SoP based authentication protocol to address the security issues identified in Chapter 3. We first present the multilevel authentication protocol and explain the interaction of different PUF designs to achieve hardware based authentication with reduced resource cost. Then, the SoP specific features of breach recognition and recovery are explained. Finally, we discuss the role specific design choices of the PUFs involved in SoP and provide reference selections.

4.1 Multilevel Authentication

In Chapter 3, we observed that research in PUFs has yielded a variety of different PUF designs with unique characteristics and strengths. We embed a SoP with role-specific PUF designs into a multilevel authentication protocol to fully leverage the advantages of each PUF design.

To provide authentication in multiple levels, the proposed system consists of three different PUFs, as shown in Figure 4.

- Hidden PUF is a reliable PUF with challenge-length l_H that does not expose the PUF responses to the outside and thus is *hidden*. It limits the exposure of Guard PUF and Secure PUF inputs to the outside.
- Guard PUF provides the first level of authentication with a challenge-length of l_G. Although of reasonable complexity, we assume this PUF to be modeled by the trusted party and thus also any attacking party. It not only acts as a guarding stage before the Secure PUF, but also indirectly propagates errors of the Hidden PUF to the Verifier and thus reduces the critical level-2 false-negative rate.

- Secure PUF is the secure backbone of SoP and is impossible to model within reasonable time and computational complexity. This PUF has a challenge-length of l_S.

The specifics of the authentication protocol are explained in Figure 5. Initially, the *Verifier* chooses a challenge C_{l1} of bit-length l_G, equal to the challenge-length of the Guard PUF. This challenge is chosen randomly and can thus fulfill the role of a Nonce to thwart replay attacks. Then, the challenge is issued to the *Prover*. Here, the Hidden PUF will generate the response R_H, which is internally connected to the Guard PUF and Secure PUF inputs. From R_H and C_{l1}, the level-1 response R_G is generated. The output of the Hidden PUF is directly connected to the Guard PUF and the Secure PUF; it is not available to the outside. After accepting the level-1 response through a Hamming-distance check against the model M_{l1}, the Verifier will send the level-2 challenge C_{l2} of length $l_{Hash} = l_S - l_H$ as a secret hash function of the initial challenge and the level-1 response. The *Prover* will generate the level-2 response from the Hidden PUF response R_H and the level-2 challenge through the Secure PUF. Finally, the *Verifier* will verify the level-2 response R_S by calculating the Hamming-distance against the true response stored in the database $DB(C_{l1})$.

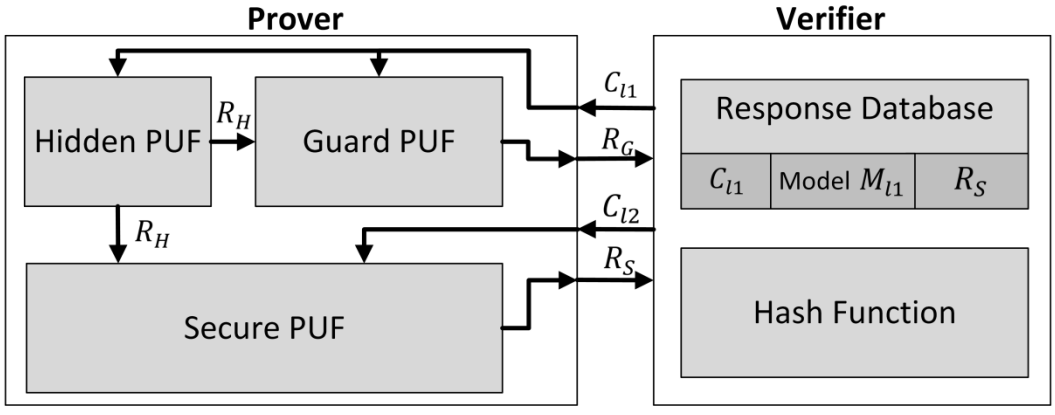

Figure 4 Components of the lightweight System of PUF (SoP).

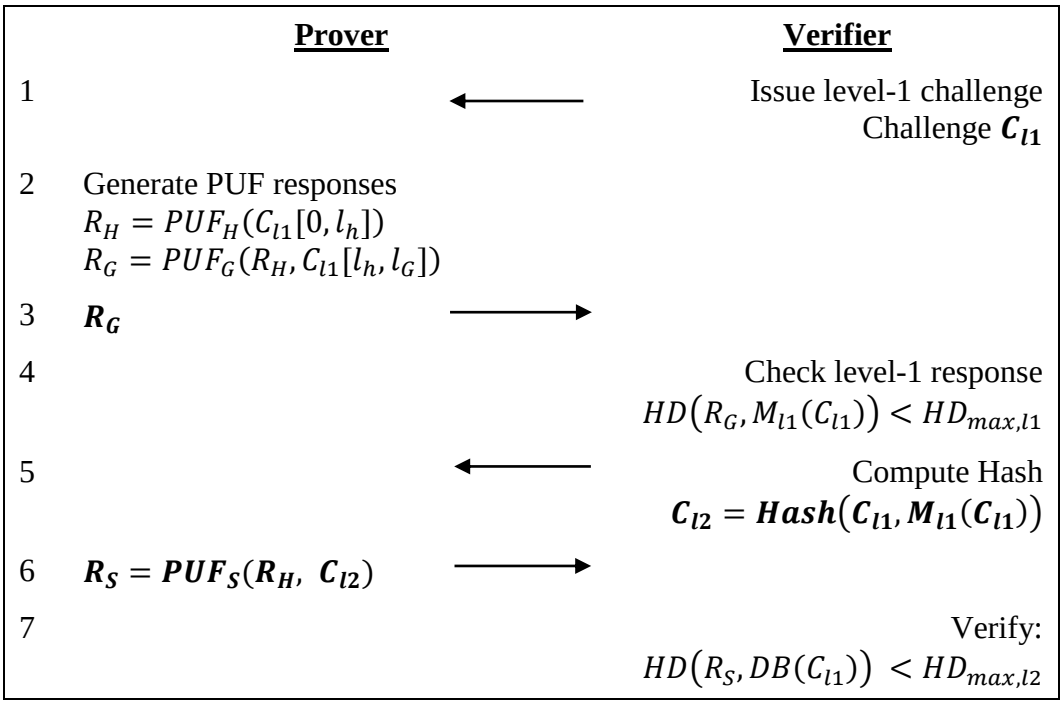

Figure 5 Proposed authentication protocol between the *Prover* and the *Verifier*. The bolded items are content of the messages, which are denoted by arrows.

4.2 Breach Recognition and Recovery

As explained in Section 4.1, the Guard PUF could be modeled by an adversary. Note that although this is possible, it will come with a considerable resource cost, as the PUF is not fully exposed to the outside.

In this multilevel scheme, a security breach can be recognized: When a false *Prover* P' repeatedly replies with the correct level-1 response $R'_G = P'(C_{l1})$ but with incorrect level-2 responses, it is apparent that the adversary must have been able to build a model for the PUF and that, therefore, the first-level of SoP is compromised. Due to the multilevel nature of SoP, it is still possible to recognize the true PUF: Until all stored level-2 challenges are exhausted, the true *Prover* may successfully authenticate, at which point the Breach recovery may be initiated.

One of the major advantages of the SoP is that it may be used with different PUF designs. For higher cost applications with a larger resource budget, the Guard PUF can be designed to be reconfigurable. This can effectively reset the level-1

PUF behavior. As the correct *Prover* can be authenticated even after a level-1 breach, the effort of the adversary can thus be reset.

4.3 Design Choices

In combination with the wide range of existing PUF designs, our proposed SoP and multilevel authentication protocol enables a variety of design choices adequate for different application scenarios. In ultra-low cost applications such as wireless-sensor networks, secure authentication is an important requirement, but the resources per sensor-node are minimal. Thus, a minimum implementation of SoP is possible that does not implement breach recovery and thus only requires simple PUFs – in case of a breach, the breached sensor-node can be considered to be dead. Additionally, instead of using multiple PUFs in parallel to increase the number of output bits, a single PUF can be created with challenge expanders. This drastically reduces area and power consumption, but reduces the output entropy.

In addition to the degree of freedom previously described, each of the three PUFs has to be chosen with great attention to its specific criteria, as shown in Table 2. The high reliability of the Hidden PUF is one of the main requirements, as any bit-error in R_H will automatically falsify the Guard PUF and Secure PUF responses due to the avalanche criterion, which specifies that a single bit flip in the input will lead to half of the output bits flipping on average. Therefore, a high error probability in the Hidden PUF will require an increased error threshold for both levels.

The high resilience to modeling attacks of the Secure PUF is a criterion fundamental to the security of the proposed authentication scheme. Since the only role of the Secure PUF is that of defending the Prover against modeling attacks, the reliability is of only secondary concern. Additionally, the input of the Secure PUF cannot be arbitrarily selected from the outside, considerably increasing the difficulty of modeling.

These clear distinctions among the roles of the PUFs significantly simplify the design task and reduces the cost, as highly secure designs typically are

unreliable and have to be corrected with expensive and complex fuzzy extractors as described in Chapter 2.

Table 2 PUF Design Criteria and Example Implementation.

PUF	**Criteria**	**Examples**
Low cost Hidden PUF	High reliability	RO PUF[4], Error corrected PUF[13]
High cost Hidden PUF	High reliability Reconfigurable	Recyclable PUF [17]
Guard PUF	Low cost	Arbiter PUF [4]
Secure PUF	High security	XOR-Arbiter PUF[9], lightweight-PUF [18], interleaved Arbiter PUF [10]

CHAPTER 5
SECURITY CONSIDERATIONS

In this chapter, we will provide a security evaluation of SoP and the authentication protocol proposed in Chapter 4. It was previously stated that the first level of SoP may be modeled by an adversary without limiting the security of SoP. Accordingly, we will show that security of the authentication protocol is not contingent upon secrecy of the first-level challenge-response model.

5.1 Attack Scenarios

In the following, we will discuss several possible scenarios, which include those that would lead to successful masquerading of a malicious party as the true *Prover* in previous protocols. We will show that in practice, our proposed protocol is not vulnerable against man-in-the-middle attacks and is resilient to denial-of-service attacks that would disable other protocols with limited stored CRP sets.

5.1.1 Simple Attacker

The attacker has to correctly guess the level-1 response with l_G bits and the level-2 response with l_S bits. The probability for a randomly guessed false positive P_{RGFP} is:

$$P_{RGFP} = \left(\sum_{i=0}^{HD_{max,l1}-1} \binom{l_G}{i} 0.5^{l_G} \right) \left(\sum_{i=0}^{HD_{max,l2}-1} \binom{l_S}{i} 0.5^{l_S} \right)$$

For the lightweight system evaluated in Chapter 6, this probability evaluates to $P_{RGFP} = 1.8 * 10^{-8}$.

5.1.2 Strong Knowledge Attack

We consider the attack-scenario that the malicious party was in physical possession of the true *Prover*. It is possible that the malicious party generated a

model for the Guard PUF, as both input and output are directly exposed [9]. When trying to authenticate itself to a true *Verifier*, it will therefore correctly respond to the initial *level-1 challenges* with *level-1 responses* and will receive the corresponding *level-2 challenges*. However, it is numerically impossible that the attacker was able to generate a valid model for the full PUF system including the Secure PUF, as outlined in Chapter 4.1. Nonetheless, it is possible that the attacker gathered a large amount of CRPs including the *level-2 challenges* and *level-2 responses*. Consider the attacker to have obtained k_{att} CRPs. He only has to store *level-1 challenges* of length l_G, *level-2 challenge* and *level-2 response,* which both have length $l_S - l_H$, for each CRP. Considering the pure data, this requires the attacker to store $D_{attacker} = (l_G + 2l_S - l_H)\frac{bit}{CRP}$, whereas the trusted party stores only $D_{trusted} = (l_G + l_S)\frac{bit}{CRP}$. As the *level-2 challenge* length is much larger than the *level-1 challenge* length, the trusted party uses only half as much memory as the malicious party. Furthermore, the malicious party has to store an exponential number of CRPs to achieve realistic authentication probabilities. The probability for a successful knowledge attack P_{KA} is:

$$P_{KA} = \frac{\mathrm{k_{att}}}{2^{l_S}} + \left(1 - \frac{\mathrm{k_{att}}}{2^{l_S}}\right) P_{RGFP} .$$

In our scenario, this means that an attacker with a model for the level-1 behavior will require $k_{att} = 2 * 10^{17}$ stored level-2 responses to achieve a false-positive rate of 1%. This will require storage of $D_{attacker} = 2 * 10^{17} * 144b \sim 3.4 * 10^{9} GB$.

This demonstrates the efficiency of our multilevel authentication and shows that SoP acts as a force-multiplier that supports the trusted party in authentication by drastically reducing memory requirements.

5.2 Denial-of-Service

One security issue observed in conventional authentication protocols was that of denial-of-service (DoS): If the protocol handles man-in-the-middle attacks by using each challenge only once, a malicious party can disrupt the protocol

without any specific knowledge of the PUF by repeatedly initiating authentication until the pool of stored CRPs is fully drained.

This is not an issue with our proposed SoP, as only the level-2 CRPs are stored in a database and thus exhaustible. Since the level-2 stage is only reached when the level-1 challenge has been correctly answered, a malicious party cannot initiate a DoS attack without an extensive modeling attack against the level-1 CRP behavior.

CHAPTER 6
EXPERIMENTAL EVALUATION

For this experimental evaluation, we simulated the lightweight variant of SoP as described in Chapter 4.3. The specific configuration of our evaluated SoP consists of synthetic implementations for three selected PUF designs according to the criteria in Table 2 and the specific design characteristics Table 1:

- Hidden PUF: 16-Bit input RO-PUF.
  RO-PUF is selected due to the high reliability.
- Guard PUF: 32-Bit Arbiter PUF.
  As an intermediate PUF, guard PUF is designed to be inexpensive, and thus the Arbiter PUF was selected.
- Secure PUF: 64-Bit 4-XOR Arbiter PUF.
  As the security of SoP depends on the robustness against modeling attacks of the Secure PUF, we select an expensive implementation with reduced reliability.

6.1 Area Cost Comparison

To provide an estimate of the area overhead incurred by authentication protocols based on PUF, we performed a consistent gate-level evaluation. For this comparison, we evaluate the main components, as the control logic introduces negligible overhead.

In Table 3, the cost of our proposed lightweight SoP is given as 1767 gate-equivalent units (GEs). Due to the entropy loss inflicted by the Syndrome generator, the Reverse Fuzzy Extractor requires longer responses and additionally employs a hash function, leading to a total cost of 5458 GEs as shown in Table 4. Despite cost saving measures, such as a lightweight hash function and challenge expanders instead of multiple parallel PUF instances, Reverse Fuzzy Extractor does not take

advantage of the unique strength of different PUF designs and instead relies on conventional circuitry. Therefore, our lightweight SoP reduces the gate count by 68%.

Note that the Reverse Fuzzy Extractor requires a 6-XOR Arbiter PUF implementation, as the input-output behavior is fully exposed and could therefore be modeled when a less complex PUF design is chosen [9]. In comparison, SoP does not fully expose the challenges of the secure PUF, warranting for a less complex implementation.

Table 3 Gate Equivalent Cost of Lightweight System of PUFs.

Component	**Explanation**	**Gate-equivalent units**
Hidden PUF	16-Bit RO-PUF	145
Guard PUF	32-Bit Arbiter PUF	130
Secure PUF	64-Bit 4-XOR PUF	1032
Challenge Expanders	16-Bit + 32-Bit + 64-Bit LFSRs	460
Total		*1767*

Table 4 Gate Equivalent Cost of Reverse Fuzzy Extractor.

Component	**Explanation**	**Gate-equivalent units**
PUF	64-Bit 6-XOR PUF	1544
Challenge Expander	255-Bit LFSR	1024
Syndrome Generator	234-Bit LFSR	940
SPONGE Hash	256-Bit light-weight Hash	1950
Total		*5458*

6.2 Hamming Distances

The move towards SoP that contains PUF sequences requires special consideration of the individual design characteristics. As explained in Section 4.3, we chose the Hidden PUF as an implementation of a RO-PUF, as it is the most reliable design available from the comparison in Table 1. The Hamming distances of each PUF component are shown in Figure 6, and the propagated error from the Hidden PUF can be seen in the Guard PUF and Secure PUF around $HD = 16$ and $HD = 32$ respectively. This shows that even a minor error in the Hidden PUF leads to a large error with $FHD_{intra} = 0.5$ for Guard PUF and Secure PUF. Figure 6 also shows why we selected $HD_{max,l1} = 5$ and $HD_{max,l2} = 21$ as parameters for the protocol: The real responses of the Guard PUF have a Hamming Distance of 5 or less to the ideal response. Similarly, the correct responses of the Secure PUF have a Hamming Distance of 21 or less.

6.3 Authentication

For authentication, the false-positive and false-negative rates are an important quality metric, as they represent the amount of authentication attempts that were falsely accepted or rejected, respectively. In our experiments, $P_{FN,l1} = 7.8\%$ of the level-1 responses had an error that exceeded the tolerance of $HD_{max,l1}$ and were thus falsely rejected. The cause of this lies in the strict avalanche criterion, and the series connection between the Hidden PUF and the Guard PUF. Thus, the protocol behaves as intended and rejects bit-errors in the Hidden PUF already at the first level. With an adequately chosen tolerance $HD_{max,l2}$ at the second level and the Hidden PUF errors already filtered at the first level, only $P_{FN,l2} = 0.257\%$ of the level-2 responses were incorrectly rejected.

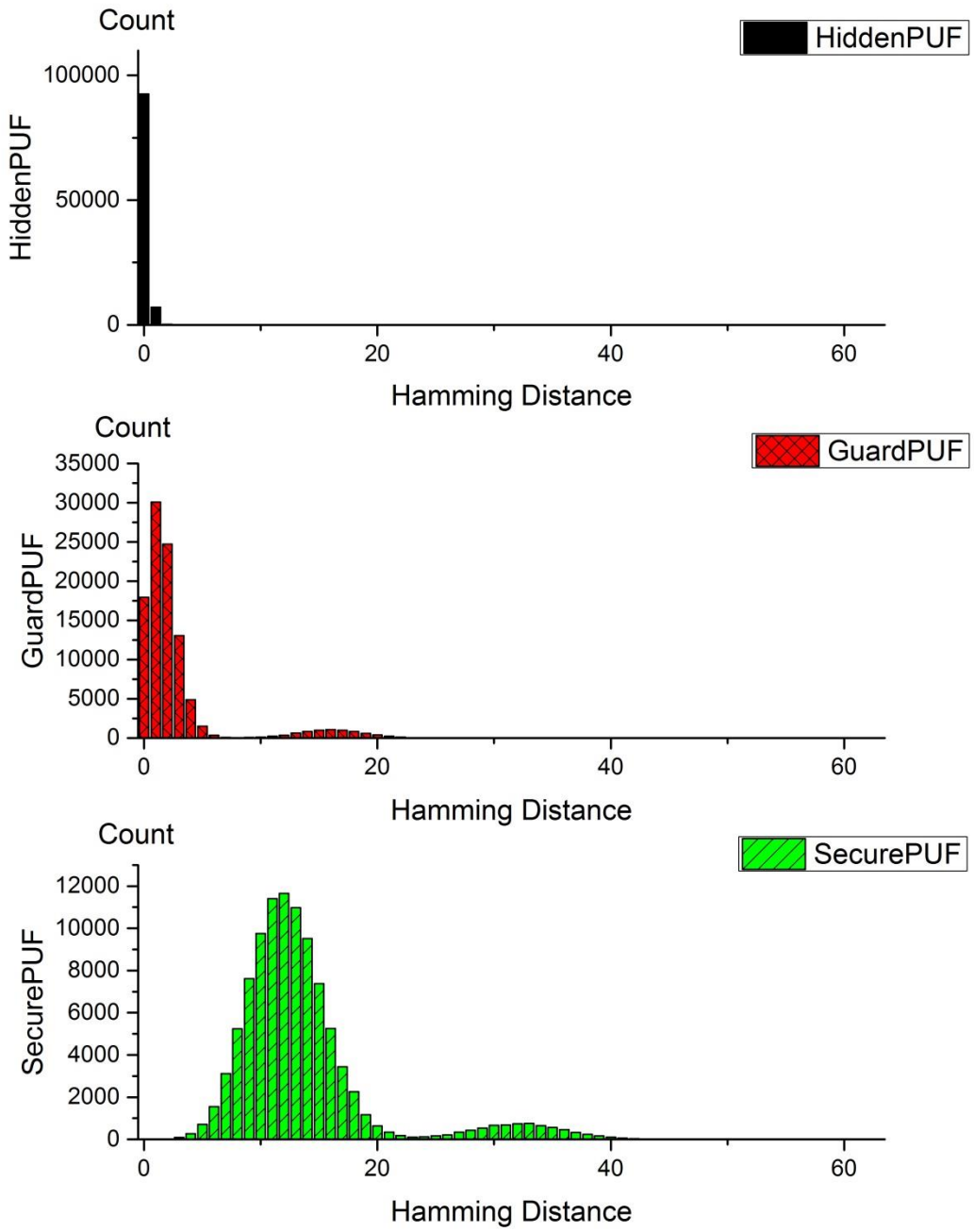

Figure 6 Hamming distances of each of the three PUF components. It can be observed that an error in the Hidden PUF (top) will propagate and lead to a large error in the Guard PUF (center) and the Secure PUF (bottom) due to the strict avalanche criterion.

CHAPTER 7
CONCLUSION

We presented a multi-level authentication protocol that takes advantage of a combination of different PUF-designs to minimize resource allocation. SoP does not require expensive error-correction, as high reliability designs are employed where required. By specifying the role of each employed PUF design, SoP not only simplifies the design, but also allows the selection of PUF designs according to their own unique strengths. In comparison to existing protocols, the need for latency and power intensive hash functions on the PUF circuit is replaced by a combination of strong PUFs and a secret hash at the verifier side, significantly reducing the area. With breach recognition and recovery, new security features are introduced and shown to improve robustness against attacks, while simultaneously enhancing reusability. A low-cost implementation of SoP was shown to reduce the area by 68% in a gate-level comparison. This low resource allocation and high flexibility allow SoP to provide a security solution tailored for ubiquitous computing devices.

REFERENCES

[1] S. B. Ors, G. Frank, E. Oswald, B. Preneel, and A. Graz, "Power-analysis attack on an ASIC AES implementation," *Inf. Technol. Coding Comput. 2004. Proceedings. ITCC 2004. Int. Conf.*, vol. 2, 2004.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002.

[3] A. Salomaa, *Public-Key Cryptography*, vol. 23. Springer-Verlag New York Incorporated, 1996.

[4] G. E. G. Suh and S. Devadas, "Physical Unclonable Functions for device authentication and secret key generation," in *44th ACM/IEEE Design Automation Conference*, 2007.

[5] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, "Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching," in *IEEE Symposium on Security and Privacy Workshops*, 2012.

[6] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," *Towar. Hardware-Intrinsic Secur.*, pp. 3–37, 2010.

[7] C. Bohm, M. Hofer, and W. Pribyl, "A microcontroller SRAM-PUF," *5th Int. Conf. Netw. Syst. Secur.*, Sep. 2011.

[8] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *Comput. IEEE Trans.*, vol. 58, no. 9, pp. 1198–1210, 2009.

[9] U. Rührmair, F. Sehnke, and J. Sölter, "Modeling attacks on physical unclonable functions," *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010.

[10] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Testing techniques for hardware security," *IEEE Int. Test Conf.*, Oct. 2008.

[11] Y. Yao, M. Kim, J. Li, I. Markov, and F. Koushanfar, "ClockPUF: Physical Unclonable Functions based on clock networks," in *Design, Automation & Test in Europe*, 2013.

[12] A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, K. U. Leuven, and E. Cosic, "Reverse fuzzy extractors : Enabling lightweight mutual authentication for PUF-enabled RFIDs," *Financ. Cryptogr. Data Secur.*, pp. 374–389, 2012.

[13] M.-D. (Mandel) Yu and S. Devadas, "Secure and Robust Error Correction for Physical Unclonable Functions," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 48–65, Jan. 2010.

[14] M. Hiller, D. Merli, F. Stumpf, and G. Sigl, "Complementary IBS: Application specific error correction for PUFs," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012, no. i.

[15] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding*, 2009.

[16] M. Potkonjak, S. Meguerdichian, A. Nahapetian, and S. Wei, "Differential public physically unclonable functions: architecture and applications," *48th ACM/EDAC/IEEE Des. Autom. Conf.*, 2011.

[17] S. Katzenbeisser, Ü. Kocabaş, V. Van Der Leest, A. Sadeghi, G. Schrijen, H. Schröder, and C. Wachsmann, "Recyclable PUFs : Logically Reconfigurable PUFs," *J. Cryptogr. Eng.*, pp. 177–186, 2011.

[18] M.-D. (Mandel) Yu, D. M. Raihi, R. Sowell, and S. Devadas, "Lightweight and secure PUF key storage using limits of machine learning," in *Cryptographic Hardware and Embedded Systems – CHES 2011*, 2011.