Attacking RO-PUFs with Enhanced Challenge-Response Pairs

Nils Wisiol, Marian Margraf

Attacking RO-PUFs with Enhanced Challenge-Response Pairs

Nils Wisiol* and Marian Margraf

Freie Universität Berlin, Germany
{nils.wisiol,marian.margraf}@fu-berlin.de

Abstract. This paper studies the security of Ring Oscillator Physically Unclonable Function (PUF) with Enhanced Challenge-Response Pairs as proposed by Delavar et al. We present an attack that can predict all PUF responses after querying the PUF with $n+2$ attacker-chosen queries. This result renders the proposed RO-PUF with Enhanced Challenge-Response Pairs inapt for most typical PUF use cases, including but not limited to all cases where an attacker has query access.

1 Introduction

Recently, Physically Unclonable Functions (PUFs) have received increasing interest as cheap and secure cryptographic key storage and as cryptographic primitives for advanced protocols [2,4]. Security for PUFs has been formally defined [1], and modeling attacks on PUFs with "many" challenge-response pairs (so-called *strong PUFs*, [10]) have been empirically studied [11,6]. Various implementations for PUFs have been proposed [12,9]; among popular implementations are Arbiter PUFs [5] and Ring Oscillator PUFs [13], although both suffer from significant weaknesses and drawbacks [11].

Classic ring oscillator PUFs with n ring oscillators possess only $O(n^2)$ challenge-response pairs, which limits their use cases. Most recently, Delavar et al. [3] have published a PUF construction based on ring oscillators that accepts an *exponential* number of challenges.

We briefly review their proposed scheme in Sec. 2, introducing a slightly different notation. In Sec. 3 we will present an efficient and computationally easy attack that breaks the security of the proposed PUF design. Finally, in Sec. 4 we will discuss reasons and consequences.

2 Construction

In this section, we will briefly review the construction proposed by Delavar et al [3]. We choose a different, but equivalent notation; we omit details of the scheme where they are not needed for our attack.

A Ring Oscillator PUF with Enhanced Challenge-Response Pairs (Enh-RO-PUF) consists of an array of n ring oscillators (RO). Each ring oscillator possesses

a frequency that is characteristic for this particular ring instance and is due to manufacturing imperfections.

Consider a given Enh-RO-PUF instance. We denote the frequency of the n ring oscillators by $f_i \in \mathbb{R}$, for $1 \leq i \leq n$. Let τ be a (for this instance) fixed integer $1 \leq \tau \leq \frac{n}{2}$. Furthermore, we choose an instance-specific $(n-1)$-bit random seed vector S.

Delavar et al. provide algorithms to compute τ and S from physical properties of the PUF instance. For our analysis, it is unimportant how these values are computed; we only stress that τ and S are constants for each Enh-RO-PUF instance.

In order to achieve an exponential number of challenges, the Enh-RO-PUF accepts any subset of the n ring oscillators as a challenge C, e.g. for $n = 256$, we could have the challenge set $C = \{42, 123, 200\}$.

Based on the challenge $C = \{c_1, ..., c_k\}$, the Enh-RO-PUF computes the result of shifting the seed S by $\sum_{j=1}^{k} c_j$ bits. We denote this as a function ρ that maps challenges to $(n-1)$-bit vectors,

$$\rho(C) = \rho(\{c_1, ..., c_k\}) = \text{shift}_{\sum c_j} S,$$

where shift_l is the l-bit-shift operator and $\sum c_j$ denotes the sum of the indices of all selected ring oscillators. At this point we stress that shift is an $(n-1)$-periodic operation, i.e. shifting by l bit leads to the same result as shifting by $l + (n-1)$ bit. Thus, for any challenge C, we have $\rho(C) = \rho(C \cup \{n-1\})$. Also note that shift can easily be inverted.

For any given ring oscillator index i, the *comparison vector* is an $n-1$ bit vector that compares the frequency f_i to all other frequencies in the array. More formally, we define the comparison vector $\varphi(i)$ to be $\varphi(i) = (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)^T$ where $a_j = 1$ if $f_i > f_j$ and $a_i = 0$ otherwise.

The Enh-RO-PUF response to a challenge $C = \{c_1, ..., c_k\}$ is an $(n-1)$-bit vector given by

$$\text{res}(C) = \text{res}(\{c_1, ..., c_k\}) = \varphi(c_1) \oplus \cdots \oplus \varphi(c_k) \oplus \rho(C).$$

As already suggested by the set notation, the response is independent of the order of the c_j by construction.

3 Attack

In the proposed PUF scheme, pseudorandom numbers $\rho(C)$ are added for protecting the scheme against information leakage. The key weakness is the predictability of the difference of certain random numbers. We can hence obtain information that was supposed to be kept secret.

For the Enh-RO-PUF, we have $\rho(C_1) \oplus \rho(C_2) = 0$ if $\sum c_i^{(1)} = \sum c_i^{(2)}$ for two challenges $C_1 = \{c_1^{(1)}, ..., c_k^{(1)}\}$, $C_2 = \{c_1^{(2)}, ..., c_l^{(2)}\}$. Using the information about

$\rho(C_1) \oplus \rho(C_2)$, we can recover the comparison vector $\varphi(n-1)$ with two queries $C_1 = \{1\}$, $C_2 = \{1, n-1\}$ to the PUF:

$$\begin{aligned} \mathrm{res}(C_1) &= \varphi(1) \oplus \rho(C_1), \\ \mathrm{res}(C_2) &= \varphi(1) \oplus \varphi(n-1) \oplus \rho(C_2), \\ \mathrm{res}(C_1) \oplus \mathrm{res}(C_2) &= \varphi(n-1) \oplus (\rho(C_1) \oplus \rho(C_2)) = \varphi(n-1). \end{aligned}$$

Knowing the comparison vector $\varphi(n-1)$, we can compute the seed S of the given Enh-RO-PUF instance by querying $C_3 = \{n-1\}$, as $\mathrm{res}(\{n-1\}) = \varphi(n-1) \oplus \mathrm{shift}_{n-1}(S)$.

Finally, given the random seed S, extracting the other comparison vectors $\varphi(i)$ for $1 \leq i \leq n$ takes one additional query $C = \{i\}$ each. As $\varphi(n-1)$ is already known, this takes a total of $n-1$ queries.

Summing up, we can reconstruct S and all $\varphi(i)$, $1 \leq i \leq n$, with $n+2$ chosen queries to the PUF. This renders the security features of this PUF scheme ineffective.

4 Discussion

Given a Enh-RO-PUF instance, using $n+2$ attacker-chosen queries, our attack is able to predict the entire challenge-response behavior. Hence, the Enh-RO-PUF is not a strong PUF with a large number of unpredictable responses, as originally claimed by the authors [3]. Given the original author's security model, where an attacker is able to apply challenges and read out responses without restriction, the design needs to be considered broken.

Throughout the design, the security rationale was that masking the response with an (to the attacker) unknown number S will stop any attack. To circumvent that protection, it is crucial to our attack that $\rho(C_1) \oplus \rho(C_2)$ can be computed by an attacker without prior knowledge of S or any $\varphi(i)$.

To mitigate the attack, we could choose $\rho(C) = h(\langle C, S \rangle)$ where h is a cryptographic hash function that obeys the avalanche criterion, and $\langle \cdot, \cdot \rangle$ is an appropriate encoding. By the avalanche criterion, $h(\langle C_1, S \rangle)$ has expected Hamming distance $\frac{1}{2}$ to $h(\langle C_2, S \rangle)$, which gives $\rho(C_1) \oplus \rho(C_2)$ an expected equal number of zeros and ones. By the assumptions on h, an attacker could not efficiently retrieve $\rho(C_1) \oplus \rho(C_2)$.

However, in this adapted PUF scheme, the values $\varphi(i)$ are not needed to achieve the desired PUF behavior. The hash values $\rho(C)$ already provide reliable, unique and random responses to given challenges C. In fact, the response function $\mathrm{res}'(C) = h(\langle C, S \rangle)$ utilizes S as a weak PUF and implements the Challenge Response Authentication Mechanism [8] with a simplified version of HMAC [7]. This widely used protocol provides evidence that authentication using the PUF is secure as long as S remains secret. However, due to the typically large implementation size of the hash function on FPGAs, this scheme fails its purpose.

This demonstrates that ρ needs to be a pseudorandom function that is hard to invert, properties that easy-to-implement choices for ρ will not be able to

provide. Hence, the design flaw of the Enh-RO-PUF can only be fixed with the introduction of another, essentially unrelated cryptographic primitive that is secure on its own.

Furthermore, we point out that if the l-th bit of $\varphi(i)$ equals 0, we have $f_i < f_l$ and thus the i-th bit of $\varphi(l)$ must be 1. These relations may be able to extend the attack surface beyond our attack.

Many use cases of Physically Unclonable Functions are fundamental to safety or security of applications. We emphasize that cryptographic primitives need a throughout study of their security before they can be considered secure, and we encourage further research in the area of cryptanalysis and Physically Unclonable Functions.

References

1. Armknecht, F., Maes, R., Sadeghi, A.R., Standaert, F.X., Wachsmann, C.: A formalization of the security features of physical functions. In: Security and Privacy (SP), 2011 IEEE Symposium on. pp. 397–412. IEEE (2011)
2. Bolotnyy, L., Robins, G.: Physically unclonable function-based security and privacy in RFID systems. In: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications. pp. 211–220. PERCOM '07, IEEE Computer Society, Washington, DC, USA (2007). https://doi.org/10.1109/PERCOM.2007.26
3. Delavar, M., Mirzakuchaki, S., Mohajeri, J.: A ring oscillator-based PUF with enhanced challenge-response pairs. Canadian Journal of Electrical and Computer Engineering **39**(2), 174–180 (2016)
4. Eichhorn, I., Koeberl, P., van der Leest, V.: Logically reconfigurable PUFs: Memory-based secure key storage. In: Proceedings of the Sixth ACM Workshop on Scalable Trusted Computing. pp. 59–64. STC '11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/2046582.2046594
5. Gassend, B., Clarke, D., Van Dijk, M., Devadas, S.: Silicon physical random functions. In: Proceedings of the 9th ACM conference on Computer and communications security. pp. 148–160. ACM (2002)
6. Hospodar, G., Maes, R., Verbauwhede, I.: Machine learning attacks on 65nm arbiter PUFs: Accurate modeling poses strict bounds on usability. In: Information Forensics and Security (WIFS), 2012 IEEE International Workshop on. pp. 37–42. IEEE (2012)
7. Krawczyk, H., Canetti, R., Bellare, M.: HMAC: Keyed-hashing for message authentication **RFC 2104** (1997)
8. Myers, J.G.: SMTP service extension for authentication **RFC 2195** (1999)
9. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical one-way functions. Science **297**(5589), 2026–2030 (2002)
10. Rührmair, U., Busch, H., Katzenbeisser, S.: Strong PUFs: models, constructions, and security proofs. In: Towards hardware-intrinsic security, pp. 79–96. Springer (2010)
11. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 237–249. ACM (2010)

12. Simons, P., van der Sluis, E., Van der Leest, V.: Buskeeper PUFs, a promising alternative to D flip-flop PUFs. In: Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on. pp. 7–12. IEEE (2012)
13. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Proceedings of the 44th annual Design Automation Conference. pp. 9–14. ACM (2007)