FPGA-based Strong PUF with Increased Uniqueness and Entropy Properties

Chongyan Gu, Neil Hanley, and Máire O'Neill
CSIT, ECIT, Queen's University Belfast, Belfast, U.K., BT3 9DT
Email: {c.gu, n.hanley}@qub.ac.uk, m.oneill@ecit.qub.ac.uk

***Abstract*—Physical unclonable functions (PUFs), are a type of physical security primitive which enable identification and authentication of hardware devices, such as field programmable gate arrays (FPGAs) and application specific integrated circuits (ASICs). Arbiter PUFs were the first proposed Strong PUF and are also widely studied. However, these designs often suffer from poor uniqueness and reliability characteristics leaving them vulnerable to modeling attacks, as well as being difficult to implement on FPGAs due to the physical layout restrictions. Some more recent designs based around non-linear voltage transfer characteristics, or non-linear currents improve the resistance against modeling attacks. However they can only be implemented on ASICs due to their voltage/current requirements. To address this problem, we propose a new PUF circuit that offers a significantly higher theoretical entropy than the traditional Arbiter PUF construction, and which is specifically designed for FPGAs. The proposed work is verified on a low-cost Nexys4 board which contains a Xilinx Artix-7 FPGA fabricated at 28nm. The experimental results give a uniqueness of $20\,\%$, considerably higher than the reported $9\,\%$ of a traditional Arbiter PUF design, and an expected reliability of $\approx 96\,\%$ over an environmental temperature range of $0^o\,C$ to $75^o\,C$, with a reliability of $\approx 92\,\%$ with $\pm 10\,\%$ variation in supply voltage.**

I. Introduction

With the increasing emergence of mobile electronic devices over the last two decades, the Internet of things (IoT) has become a reality and its influence on our day to day activities is set to further increase with a projected 50 Billion connected devices by the year 2020 [1]. These smart devices and sensors will be found in our homes, our cars, our workplaces *etc.*. However this poses serious security and privacy issues as highlighted by the recent IoT based distributed denial of service (DDoS) attacks [2]. While conventional cryptographic approaches involving complex computations might not always be suitable for IoT endpoints (such as sensors) due to energy and area overheads, there will be a large class of intermediate powered devices that secret keys need to be stored on and secured against attackers. This could potentially open up new attack vectors for criminal hackers to exploit as they will often have physical access to these IoT devices. This has led to a high demand for cryptographic mechanisms to protect user privacy and data security.

Physical unclonable functions (PUFs) are a security primitive which utilise the inherent variations that occur during manufacturing in order to generate a unique intrinsic identifier for a device. Such a primitive has a number of desirable properties from a security aspect, such as the ability to provide low-cost identification of an integrated circuit (IC) (Weak PUF) or to provide a variability aware circuit that returns a device specific response to an input challenge (Strong PUF). This gives some advantages over current state-of-the-art alternatives for a number of applications, *e.g.* secure key storage for embedded IoT applications. In general, no special manufacturing process is required to integrate a PUF to a design lowering the overall cost of the secure IC, and everything can be kept on-chip allowing it to be used as a hardware root of trust for all security or identity related operations on the device. This subsequently enables a multitude of higher level secure cryptographic protocols such as the aforementioned secure key storage and/or device authentication.

Arbiter PUFs, based on the timing and delay characteristics of silicon circuits, were the first proposed Strong PUF architecture [3] and are widely studied. It consists of two parallel identical n-stage multiplexer (MUX) chains with the outputs fed into an arbiter to determine which signal arrived first in order to form 1-bit of an n-bit PUF. The design and implementation of digital Strong PUF circuits is challenging, particularly for FPGA as the routing paths are restricted by the existing fabricated circuit. Due to their flexibility, lower time to market, and increasing density, FPGAs are increasingly used for many applications. Since the circuits depend upon process variations, even small changes in environmental conditions, such as voltage or temperature, or an unbalanced design that introduces a bias in favour of one path over another, will affect their performance. The original Arbiter PUF [3], [4] designs suffer from poor uniqueness and repeatability properties, however subsequent works [5] improved both results. FPGA implementation is still a non-trivial issue however with the authors in [6], [7] using an additional tuning circuit in order to balance the delay lines on their FPGA based Arbiter PUF.

Modelling attacks employing machine learning (ML) methods have been reported to successfully break the security of a wide range of Arbiter PUF architectures by building a software model of the variability using the CRP [8]. The Arbiter PUF responses can be attacked individually, building up a linear additive delay model for each bit. Although the XOR Arbiter PUF [9], feed-forward Arbiter PUF [4] and lightweight Arbiter PUF [10], increase the resistance of such modelling attacks, they can also be broken given enough CRPs [8]. A non-linear PUF circuit, based on Voltage Transfer Characteristics (VTC), was proposed specifically to be resistant to such attacks [11], as well as a similar circuit based on current mirrors [12]. However, these methods were simulated for ASICs and they are not suitable for FPGA implementation. Arunkumar *et al.* [13] suggest properties that designers should aim to meet when designing ML resistant Strong PUF designs. However a practical implementation of these proposals is still an open

978-1-4673-6853-7/17/$31.00 ©2017 IEEE

problem.

To address some of the issues outlined above, in this paper we propose a new robust FPGA-based Strong PUF design. More specifically, the research contributions of this paper are as follows:

- We propose an improved Strong Arbiter PUF design, which is composed of two groups of flip-flops and MUXs. On each stage different flip-flops are selected as the delay element by the challenge. The response is $[1, 0]$ depending on the outcome of the race condition created between the two delay paths.
- A comparison of the theoretical entropy provided by the original Arbiter PUF and the proposed Strong PUF designs is provided, with the proposed design having $(2 \cdot \log_2(m))$ times higher entropy (where m is the number of delay elements at each stage), trading off a slight increase in the required area due to an efficient utilisation of resources.
- We give empirical experimental results showing that the proposed design has an expected uniqueness of $\approx 20\,\%$, compared to $\approx 9\,\%$ for the original design on the same platform.

The rest of this paper is organized as follows; Section II introduces the Strong PUF architecture. The FPGA implementation of the proposed design is outlined in Section III. Experimental results are given in Section IV, and conclusions are drawn in Section V.

II. FPGA-based Strong PUF Design

A. Architecture Model

The output of a traditional delay based Arbiter PUF depends on a sum of n delay elements from two delay paths chosen by a CRP set of bit length n. The proposed Strong PUF design trades off area for complexity by selecting 1 of m of delay elements at each stage, with the challenge reversed between paths to ensure that the same pairs of delay elements aren't selected each time. The design of a 1-bit response generation cell is shown in Fig. 1 and comprises an array of $2 \times (m \times n)$ delay elements, $\triangle T^U_{n,m}$ and $\triangle T^L_{n,m}$, and where $k = \log_2(m)$. T^U and T^L respectively represent the upper and lower delay paths. An arbiter at the end of two delay paths is used to determine which delay path is faster, and then outputs a response of 1 or 0. The delay elements of each stage, $\triangle T^U_{n,m}$ and $\triangle T^L_{n,m}$, are chosen by challenges C of length $n \cdot k$. To further increase the security, the XOR and lightweight Arbiter PUF extensions could be employed in the proposed design, however this is outside the of scope of this paper.

B. Entropy Analysis

Unpredictability requires that an adversary cannot efficiently predict the response of a PUF to an unknown challenge from previously observed CRPs. Shannon Entropy is used to assess the unpredictability of PUF output for each stage (Eq. 1).

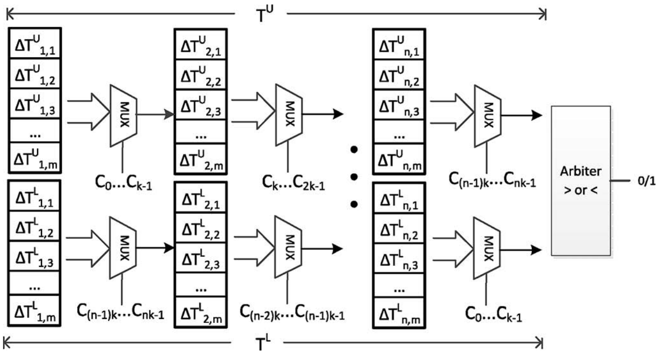


Fig. 1: Proposed Strong PUF architecture design.

$$
\begin{aligned}
\mathrm{H}(\Phi_s) &= -\sum_{i=1}^{m^2} p_i \cdot \log_2(p_i) \\
&= -\sum_{i=1}^{m^2} \frac{1}{m^2} \cdot \log_2\left(\frac{1}{m^2}\right) \\
&= \log_2\left(m^2\right) = 2 \cdot \log_2(m)
\end{aligned} \tag{1}
$$

Here Φ_s represents a given PUF instance of our proposed design, and p_i is the probability of a given pair of delays being selected by the challenge, which is assumed to be uniformly random. Given m delays in the upper and lower paths of each stage, this gives m^2 permutations in total, allowing us to derive the entropy increase of each stage as shown in Fig. 2 with $p_i = \frac{1}{m^2}$.

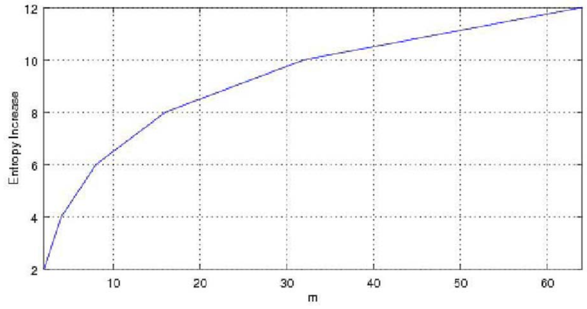


Fig. 2: Entropy increase per stage.

C. An Example of Circuit Design Using Flip Flops

For an FPGA implementation, flip-flops can be used to implement the delay elements for the proposed design [14][15]. The design of a 1-bit response module with $m = 2$ is schematically shown in Fig. 3, and comprises of a group of multiplexers (MUXs) and flip flops. To generate a single bit *R*, $2 \times n$ MUX gates and $4 \times n$ flip flops are cascaded in a row. The first stage of the proposed Strong PUF circuit is reset by *CLEAR* and then activated by a rising edge of the *START* signal, which is connected to the clock port of each flip flop. A MUX is used to select which delay element goes into the next stage depending on the challenge C_i. The output of the delay element feeds into the clock of the delay elements in the next stage. At the end of two delay paths, upper T^U and lower T^L, the signals, Q^U and lower Q^L, are output to an Arbiter, which consists here of cross-coupled NAND gates, in order to determine which delay path is faster. To generate an n-bit response, the above *1*-bit design is repeated n times.

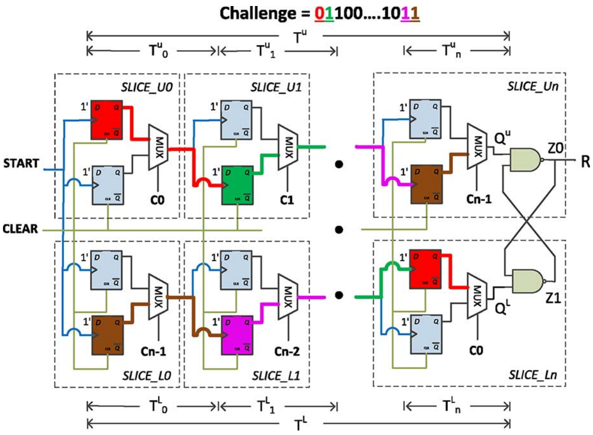
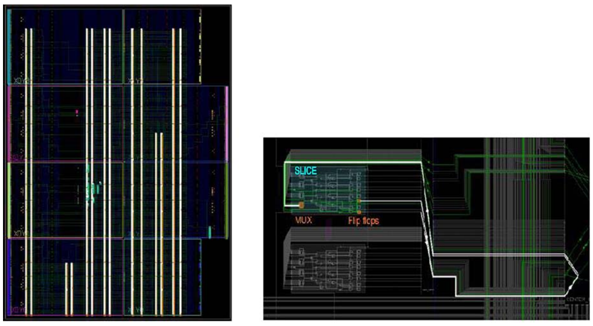


Fig. 3: Proposed design with $m = 2$.

III. FPGA Implementation

As previously mentioned, Arbiter PUF FPGA implementations can suffer from a significantly biased uniqueness properties due to the difficulties in balancing the routing paths. A recent implementation of a 64-bit Arbiter PUF on a Digilent Nexys4 evaluation board with a Xilinx Artix-7 FPGA [7] required 129 slices to generate a single-bit response. For our proposed design using four delay elements at each stage, 128 slices are required on the same Nexys4 evaluation board in order to generate a single bit. The extra complexity comes at no extra cost in slices due to increased utilisation of the slice components. An image of the FPGA floor plan generated by the Xilinx Vivado 2016.2 tool of the proposed design is shown in Fig. 4a. In each delay unit, the 2 flip-flops and 1 MUX are implemented in a single slice as shown in Fig. 4b, which also shows the fixed routing path.



(a) Proposed 64-bit Strong PUF design

(b) Each delay element

Fig. 4: Placement and routing of proposed Strong PUF design using Vivado

IV. Experimental Results

Experimental results for a 64-bit variant of the Strong PUF design with $m = 2$ were acquired from 10 low-cost Digilent Nexys4 boards. The temperature test was undertaken by using a thermoelectric plate to adjust the temperature of a single FPGA from 0^oC to 75^oC, while the voltage test was carried out over a range of 1.0 V (the core voltage) ±10%.

A. Uniqueness

A PUF design is expected to produce a different response when implemented on different devices when supplied with the same challenge. Uniqueness measures inter-chip variation by evaluating how will design can differentiate d different devices. It is calculated using the inter-chip Hamming Distance (HD) as shown in Eq. 2. R_i and R_j represent the n-bit responses generated from two chips i and j using the same challenge C.

$$Uniqueness = \frac{2}{d(d-1)} \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \frac{HD(R_i, R_j)}{n} \times 100 \quad (2)$$

Ideally, when a PUF circuit is implemented on different devices it should produce an average inter-chip HD close to 50% when supplied with the same challenge, implying that half the response bits are different between the two devices even though the same challenge has been used. Eq. 2 is an average of all possible pair-wise average HDs among d devices, and expresses an estimate of the inter-chip variation in terms of PUF responses for the same challenge. The uniqueness result of the proposed Strong PUF design is ≈ 20 % as shown in Table I. The research in [7], using the same FPGA device to implement the traditional Arbiter PUF, recorded a uniqueness of ≈ 9.42 %. A reported uniqueness result of the traditional Arbiter PUF on an ASIC is 23 % [5]. Compared to the results from FPGA, the proposed PUF design has a similar uniqueness result to that from an ASIC, demonstrating a significant improvement in its ability to distinguish between different devices.

B. Reliability

Ideally, a given PUF design, implemented on any device should be able to perfectly reproduce its output whenever it is queried with a challenge. However, environmental fluctuations in temperature and power supply voltage, as well as the natural properties of metastability cause noisy responses. Therefore, the reliability of a response is defined as the percentage of noisy response bits, and which quantifies the error in the PUF response. For a device d_i, reliability is established as a single value by finding the average intra-chip HD, HD_{intra}, of s $n-$bit response samples, $R_i^{'}$, taken at different supply voltages and temperatures compared to a baseline $n-$bit reference response, R_i, taken at nominal operating conditions. The intra-chip HD is defined as follows:

$$HD_{INTRA} = \frac{1}{s} \sum_{t=1}^{s} \frac{HD(R_i, R_{i,t}^{'})}{n} \times 100 \qquad (3)$$

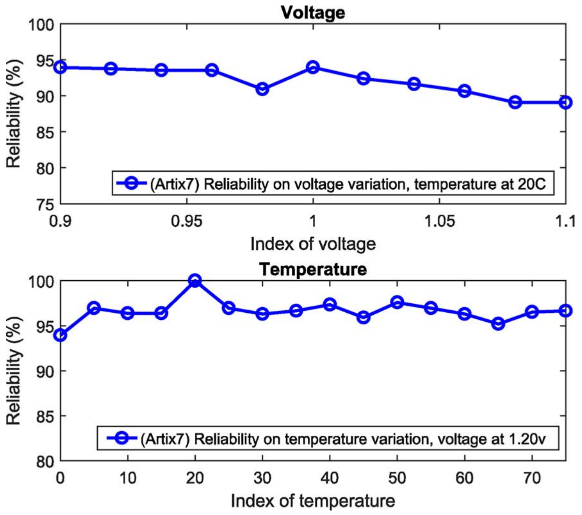


Fig. 5: The reliability result

where $R(i,t)'$ is the $t-$th sample of R_i'. The reliability is equal to $100 - HD_{INTRA}$.

The ideal value for reliability is 100%. Fig. 5 shows the reliability results of the proposed PUF design are 96.64% over an environmental temperature range of 0^oC to 75^oC, and 92.04% over a range of $\pm 10\%$ variation in the supply voltage.

	FPGA results		ASIC results [5]	
	U (%)	R (%)	U (%)	R (%)
Traditional APUF	9.42 [7]	-	23.00	95.20 (Rt) 96.30 (Rv)
Proposed PUF	20.00	96.60 (Rt) 92.04 (Rv)	-	-

TABLE I: The comparison of uniqueness and reliability results on FPGA and ASIC

Table I shows the comparison of uniqueness and reliability results between the traditional Arbiter PUF design and the proposed Strong PUF design on FPGA and ASIC. Rt and Rv represent the reliability results of the PUF design under temperature and voltage experiments. The proposed Strong PUF design exhibits a better uniqueness result on FPGA, equivalent to the result on ASIC, and achieves high reliability results on FPGA, which is notable since the proposed Strong PUF design enables comparable reliability results as the traditional Arbiter PUF on ASIC.

V. Conclusions

In this paper, a new, strong and robust FPGA-based PUF design is proposed, with the generation of the response dependent on creating a race condition between two identical delay paths. The circuit layout is controlled using scripts to ensure balanced routing when targeting a low-cost Xilinx Artix-7 FPGAs. The proposed PUF design has $2 \cdot log_2\left(m\right)$ times higher theoretical entropy than the traditional Arbiter PUF, and the experimental results show promising uniqueness and reliability properties. Future work to evaluate the design on a larger set of FPGA devices for increased statistical confidence in the results is ongoing, as well as an analysis of its resistance to modelling type attacks.

Acknowledgment

This work has been supported by the KeyHAS project, the R&D program of IITP/MSIP (Study on secure key hiding technology for IoT devices), and by the SPARKS project, funded by EU 7th Framework Programme (FP7/2007-2013, grant agreement no. 608224; www.project-sparks.eu).

References

[1] CISCO. Internet of things IoT. Accessed: 23-07-2015. [Online]. Available: http://www.cisco.com/web/solutions/trends/iot/portfolio.html

[2] KrebsonSecurity. Ddos on dyn impacts twitter, spotify, reddit. Accessed: 08-11-2016. [Online]. Available: https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/

[3] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM conference on Computer and Communications Security*, May 2002, pp. 148–160.

[4] G. Blaise, L. Daihyun, C. Dwaine, M. V. D., and S. D., "Identification and authentication of integrated circuits," *ACM Concurrency and Computation: Practice and Experience*, vol. 16, no. 11, pp. 1077–1098, 2004.

[5] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Proc. Symposium on VLSI Circuits*, June 2004, pp. 176–179.

[6] M. Majzoobi, A. Kharaya, F. Koushanfar, and S. Devadas, "Automated design, implementation, and evaluation of arbiter-based PUF on FPGA using programmable delay lines," *Cryptology ePrint Archive*, 2014.

[7] Y. Hori, H. Kang, T. Katashita, A. Satoh, S. Kawamura, and K. Kobara, "Evaluation of physical unclonable functions for 28-nm process field-programmable gate arrays," *Journal of Information Processing*, vol. 22, no. 2, pp. 344–356, 2014.

[8] U. Ruhrmair, J. Solter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[9] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Automation Conference (DAC'07)*, San Diego, CA, Jun. 2007, pp. 9–14.

[10] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure pufs," in *Proc.IEEE/ACM International Conference on Computer-Aided Desig (ICCAD'08).*, Nov 2008, pp. 670–673.

[11] V. Arunkumar and K. Sandip, "A novel modeling attack resistant PUF design based on non-linear voltage transfer characteristics," in *Proc.IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE'15)*, Grenoble, France, Mar. 2015, pp. 653–658.

[12] K. Raghavan and B. Wayne, "On design of a highly secure PUF based on non-linear current mirrors," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust, HOST'14*, Arlington, VA, USA, May 2014, pp. 38–43.

[13] V. Arunkumar, P. C. Vinay, P. B. Charles, and K. Sandip, "Machine learning resistant strong PUF: possible or a pipe dream?" in *Proc.IEEE International Symposium on Hardware Oriented Security and Trust, (HOST'16)*, McLean, VA, USA, May 2016, pp. 19–24.

[14] C. Gu, J. Murphy, and M. O'Neill, "A unique and robust single slice FPGA identification generator," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'14)*, Melbourne, Australia, Jun. 2014, pp. 1223–1226.

[15] C. Gu and M. O'Neill, "Ultra-compact and robust FPGA-based PUF identification generator," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'15)*, Lisbon, Portugal, May 2015.