School of Electrical and
Electronic Engineering

Physically Unclonable Functions for Networked Device Authentication

Literature Review

Michael Walker (130623935)

EEE8097 Individual Project

Wireless Embedded Systems (MSc)

ABSTRACT

Physically Unclonable Functions are a relatively recent area of active research which can be seen as analogous to a fingerprint for hardware that potentially offer a way to provide low-cost, automated, trustworthy authentication of embedded systems. As the infrastructure of networked devices grows in our homes and workplaces so too will the need to protect ourselves from 'Hardware Trojans' and other threats. In reviewing the state-of-the-art of 'PUF's and their potential incorporation into modern ciphers as cryptographic primitives using fuzzy extractors the general significance and worth of an investigation into the implementation of practical Ethernet message authentication codes using Static RAM PUFs and Fuzzy Extractors as the core components is to be shown.

1. INTRODUCTION

The modern world now relies upon small, embedded and increasingly networked electronic devices to facilitate a multitude of industrial, automotive and commercial business tasks, and to provide convenience and entertainment in the domestic setting. Much of this pervasive assemblage of digital systems is both cryptographically unsecured and physically accessible to potential intruders.

As the quantity, connectivity and complexity of embedded devices continues to increase, we enter an era of ubiquitous, personalized and context-aware computing devices. The size and scope of the information stored, processed and conveyed by these systems is set to cause vast digital security risks[1] across all society unless inexpensive, low-power cryptographic primitives and protocols which are resilient to both conventional cryptanalysis as well as side-channel and physical attacks can be introduced into the embedded market[2].

A recently proposed foundational concept to build these security systems upon is that of Physically Unclonable Functions (henceforth called 'PUFs'). That is, a function that is '*an expression of an inherent and unclonable instance-specific feature of a physical object'[3]*. More tersely put, they can be considered the physical equivalents of an algorithmic one-way function[4] or as '*Hardware Biometrics*'[5].

The features of PUFs make them excellent cryptographic primitives to provide the secret-key material for any cryptography based security protocol, be it symmetric or public-key based. Utilization of PUFs in a *challenge-response* framework provides a light-weight solution to key distribution[6] and avoids key storage in non-volatile memory where it would be prone to side-channel attacks[7].

However as with human biometric techniques there is inherent noise that must be dealt with[8]. Conversely, contemporary cryptographic paradigms all require highly robust noise-free primitives for their operation. Fuzzy-Extraction[9] techniques have been proposed as a solution to bridge the gap and marry PUFs to Cryptography.

With strong cryptography comes trust. By using PUFs in Message Authentication Codes (MAC) as noted in [10] within existing networking protocols, that trust can be relied upon to create the secure embedded infrastructure that the rapidly approaching '*internet-of-things*'[11] will need.

In this paper we intend to critically review the present state-of-the-art of PUF technology and its applications to network authentication through fuzzy-data extraction techniques, and to propose practical realization and investigation of these features through a hardware implementation.

2. PHYSICALLY UNCLONABLE FUNCTIONS

PUFs were introduced as a concept in the seminal paper of the field by Pappu et al. in 2001[12] (although other work anticipated the idea; see [13] & [14]), in which they used the 'mesoscopic physics of coherent transport through a disordered medium' as a PUF; by which they meant the scattering pattern of laser light when shined through a rough block of glass.

This scattering, unique to each instance and irreproducible by any another can be measured, interpreted and stored as digital data and, if sufficient entropy is available[15], used to distinctively identify that particular instance against all others without fear of imposters, replicas or duplicates.

The key properties of good PUFs were summed up well in [16], A much compressed and distilled overview is presented as follows:

Constructability – Feasible to instantiate a *random* instance.

Physical Unclonability –Infeasible to instantiate a *specific* instance.

Evaluability – For any random valid *challenge* it evaluates a binary *response* value

Reproducibility – Probability of an evaluation of the same *challenge* producing the same *response* on the *same PUF* is *high*. (*Intra-distance* in small)

Uniqueness – Probability of an evaluation of the same *challenge* producing the same *response* on a *different PUF* is *low*. (*Inter-distance* is large)

Identifiability – Exhibits both Reproducibility and Uniqueness. (There is a High Probability that the *Intra–distance* is less than the *Inter–distance*).

Further 'Nice-to-Haves' are Unpredictability, Mathematical Unclonability, True Unclonability, 'One-wayness' and Tamper Evidence.

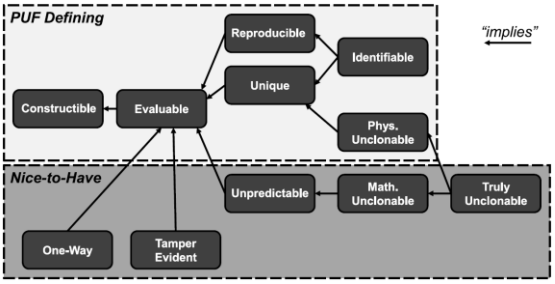

Figure 1: Relations of the properties of PUFs[16]

2.1. Silicon PUFs

A problem with optical and other non-electronic PUFs such as Coating PUFs[13], is that they cannot be implemented on ICs where cost benefits can best be realized by immediately integrating them into the digital circuitry of cryptographic IP Cores. Silicon (or Electronic) PUFs can be considered a separate class and rely on variations in electrical properties of circuit components caused by tiny deviations in the photolithographic processes that made them. Introduced a year after Pappu in the form of the Ring-Oscillator PUF[17], similar PUFs based on delay differences between the outputs of similar circuits include the Arbiter PUF[18, 19] from 2004 and the more recent Glitch PUF[20].

2.2. Bi-Stable PUFs

Silicon PUFs can be based on the bi-stability of silicon components rather than propagation delays. Perhaps the most promising is the SRAM PUF introduced in another significant paper in the field by Guajardo et al. in 2007[10]. It is also worth mentioning other later types such as; Latch[21], Flip-flop[22], Butterfly[23] and Buskeeper[24].

The later types were introduced to overcome a perceived shortcoming of SRAM, whereby data extraction must occur shortly after a power-up condition. Instead, the others can be polled continuously, and Butterfly PUFs have the further advantage of being synthesizable on FPGA.

However, all have the negative feature of requiring greater wafer area to ensure similar levels of entropy[4] (figure 3) and SRAM has been shown to be more resilient to thermal noise and aging effects[25] (figures 4 & 5). So it would seem likely that SRAM PUFs will become the dominant form in commercial ASIC design due to their intrinsically lower costs and familiar implementation, and as such are the focus of this paper. For more depth on this section see Maes[8].

2.3. SRAM PUFs

Static Random Memory (SRAM) is a common form of volatile memory already used extensively in many, if not most, electronic devices. In the conventional CMOS implementation a SRAM cell consists of 6 transistors arranged as in figure 2, at its core are two cross-coupled inverters made of two transistors each. Upon power-up the SRAM cell must eventually deflect from a theoretical metastable point to a stable '1' or '0' output after some settling time. This is due to the unavoidable device mismatch between the inverters[26, 27].

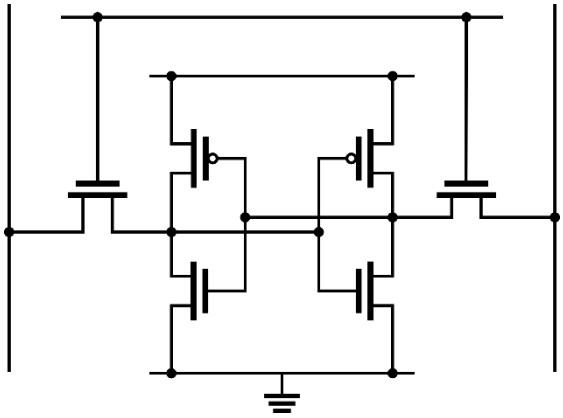

Figure 2: Circuit diagram for a standard six transistor CMOS implementation of a SRAM cell

PUF type	Area (mm^2)	Cells/ mm^2	Minimum #bits/mm^2		
			k=1, ℓ=1	k=60, ℓ=1	k=60, ℓ=40
SRAM	0.213	$\approx$ 1.2M	0.75M	0.95M	1.3M
DFF	0.392	$\approx$ 84k	28k	32k	37k
Latch	0.272	$\approx$ 0.12M	22k	25k	29k
Busk.	0.076	$\approx$ 0.22M	91k	0.11M	0.14M

Figure 3: Area Advantage of SRAM[28]

PUF Type	Before Ageing		After Ageing ($\sim$ 4.5 years)		After Ageing Separate P.D.	
	min	max	min	max	min	max
SRAM	5.0%	5.5%	7.0%	8.0%	5.5%	5.5%
Bus-keeper	3.5%	5.0%	5.5%	7.0%	3.5%	5.0%
Latch	2.0%	3.0%	5.0%	6.0%	2.5%	3.5%
DFF (#0,2,3)	2.5%	4.0%	4.5%	6.0%	3.5%	4.0%
DFF (#1)	3.5%	7.0%	4.0%	12.0%	n.a.	n.a.
Arbiter	2.5%	3.5%	3.0%	4.5%	n.a.	n.a.
Ring Osc.	0.9%	2.3%	3.4%	4.8%	n.a.	n.a.

Figure 4: Ageing Resilience Advantage of SRAM[25]

PUF Type	Meas. Data (instances x nr. of bits)	FHD (noise) -40°C		+25°C		+85°C	
		min	max	min	max	min	max
SRAM	4 x 65536	7.0%	8.0%	5.0%	6.0%	6.5%	8.0%
Bus-keeper	2 x 8192	8.0%	11.0%	3.0%	4.5%	15.5%	20.5%
Latch	2 x 8192	15.0%	28.0%	2.5%	3.5%	8.0%	18.0%
DFF (#0,2,3)	3 x 8192	10.0%	17.0%	3.0%	4.0%	16.0%	21.0%
DFF (#1)	1 x 8192	10.0%	33.0%	3.0%	10.0%	12.0%	24.0%
Arbiter	1 x 8192	3.0%	4.5%	2.5%	4.0%	2.5%	4.5%
Ring Osc.	1 x 3840	1.6%	3.9%	0.6%	2.8%	1.4%	3.9%

Figure 5: Thermal Noise Resilience Advantage of SRAM[25]

3. DERIVING SECURITY FROM PUFS

Essential to the Unclonability of PUFs as crypto-primitives is the *inherent* nature of how it identifies the hardware device it is incorporated within, rather than being an *assigned* identity from without. This means that, like a biometric, a PUF needs to be enrolled before it can be used. Another peculiarity of the inherent identity it shares with biometrics is its 'fuzzy' nature[29], i.e. it produces *noisy* data.

This implies that, when evaluating the responses to challenges, authentic identity must be surmised from the response passing some test of closeness to the expected value (hamming distance equal to less than a set threshold), rather than an exact match. However, it would seem evident that some form of cryptographic protection is also required to prevent replay and man-in-the-middle attacks, this can be provided by encryption yet this raises a new problem.

Traditional Ciphers rely upon the exact, precise nature of their keys and introducing noise makes them unworkable. Introduced with the concept of 'fuzzy commitment'[30] and advanced with the later model of 'Fuzzy Extractors' is an approach for allowing noisy data to be incorporated into keys using, most crucially, error correction codes.

3.1. Fuzzy Extractors

Fuzzy Extractors were first presented in the literature by Dodis et al. in 2004[31]. In outline they consist of two parts; a 'Secure Sketch' that mitigates the noise problem, and 'Randomness extractor' which ensures robustness of the ciphertext through the concept of 'Privacy Amplification' which was adapted from quantum cryptography by Bennett et al. in 1995[32].

Much of the most cited literature available on fuzzy extraction is focused on its mathematical underpinnings [33-36], while little is available to explain the specifics of practical implementation clearly. Kang et al.[9] provided a clean and easily understood implementation that will be used and explained here as a reference and it also provides a good basis for further work, However the accumulated work in the field by Dodis, Tuyls[37] and Škorićs[38] is central.

During the Enrolment phase (generation procedure) The PUF is challenged for a response. The initial response is combined with random data and passed through a cryptographic hash function leading to an encrypted response, this is the *Randomness Extractor*. At the same time the original unencrypted response is combined with the output of a high redundancy error correcting encoder, this is the '*Secure Sketch*' and along with the first random string is outputted as 'helper data'. The idea is that the encrypted response and helper data cannot be used to discover the PUF plaintext response.

During Verification (reproduction procedure) an error correcting decoder and identical components to the enrolment circuitry are used in a different arrangement called the '*Secure Sketch Recovery Procedure'* that after reapplying the Randomness extractor component reproduces the same plaintext PUF response if given the same helper data and crucially, if the PUF used is the same.

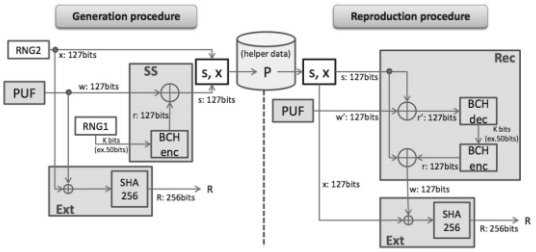

Figure 6: Data flow of the reference Fuzzy Extractor implementation[9]

3.1.1. Implementing the Secure Sketch

The reference model from Kang uses BCH error correction, as does Arakala[39] in the much cited 'Pin Sketch' implementation. However alternative error correction codes such as the Hamming, Hadamard can also be used. The more complex Reed-Solomon is more suitable for burst errors which less applicable to PUFs[40]. Combinations of more than one code are also possible such as in Hao[41]. Investigation into the performance of different ECCs in this context is quite likely to be a fruitful research area.

3.1.2. Privacy Amplification

The reference model uses the SHA-2 Cryptographic Hash; there are again, many possibilities for experimentation with different hash families. The recently introduced concept of sponge hashing, as used in the new SHA-3 standard[42] could be of interest as it's reference VHDL implementation supposedly outperforms SHA-2[43] and does not suffer the length-extension weakness of SHA-2 (and SHA-1) which could be important if the output is used for Message Authentication Codes[44].

Also required is True Random Number Generators (TRNGs). The leading paper on the topic by Tsoi discusses using FPGA synthesizable oscillator phase noise[45]. Many Methods for FPGA synthesis of TRNGs are explained well in [46], Whereas [47] explains another method using metastability.

3.2. Ethernet Incorporation with MACs

Authenticating a device only once is not necessarily enough to ensure security. Periodic challenge-responses via the network would facilitate greater trust and reliability. This leads to some problematic constraints as PUF data is necessarily finite yet the time period over which challenges could be made is theoretically unbounded.

Dodis proposes '*Extractor-MACs*' for this purpose[33] which require additional constraints – applicable on the cryptographic hashing function especially - which are difficult to critique due to complexity and unfamiliarity, further investigation will be needed. Still even with advanced techniques such as above key reuse will inevitably have to play a part

This is a relatively unexplored area in the literature and it is hoped some further progress can be made in the field through this project.

4. PROJECT PROPOSAL

Through an implementation of a working PUF-based authentication system much is to be learnt about the practicalities of a still quite novel security technique and its application to network security in particular. It would be of benefit to discover the difficulties encountered in such an approach and many of the design decisions taken during the process will have guiding implications for further research into the area.

4.1. FPGA Implementation

It is proposed to use FPGA synthesis techniques to create a hardware device that performs the necessary steps. These components in broad terms being:

- SRAM Interface
- Random Number Generation
- Error Correction
- Cryptographic Hashing
- Ethernet Interfacing

While project specific code will be necessary in parts, a large repository of VHDL (or Verilog) open-source licensed source code is already available for a number of these components at OpenCores.org and academic work on the efficient FPGA implementation of FEC codes such as Reed-Solomon[48], cryptographic hashes such as SHA-2[49], random number generation[45] and LFSRs[50] has been carried out and can be incorporated into the design. Integration of pre-existing IP will minimize the original work required in developing the system to core areas of research interest.

4.2. Hardware, SRAM Memory & Ethernet

FPGA implementation will require an actual FPGA chip. A readily accessible source for these which are familiar to this author are the Altera DE-1 Development Board developed by Terasic[51], which offers an adequately powerful Cyclone II FPGA chip in a convenient package.

While the board incorporates an SRAM chip, the necessary power control is not afforded to the programmable FPGA, neither is an Ethernet port available. Thus these two components of the system must be provided by a separate board that interfaces to the DE-1 via its 2 40-way GPIO pins, a prototype board has already been fabricated at this time.

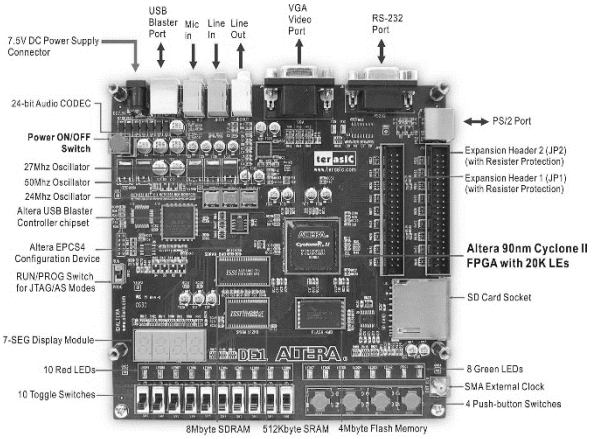

Figure 7: The DE-1 Development board proposed for use in the project

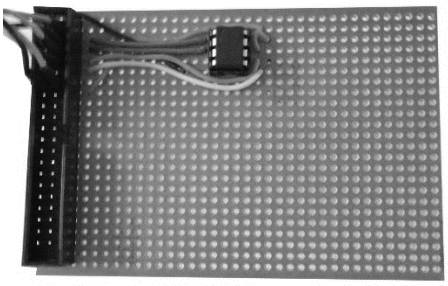

Figure 8: Prototype board with Microchip 23LCV512 SRAM ready for testing

Any Standard Ethernet port component will suffice, however SRAM chips are implemented in a variety of different ways that mean so chips are unworkable. Some chips have been tested in [52] and the author has unsuccessfully tried a Microchip 23LCV512[53].

5. CONCLUSIONS

PUFs hold a great deal of promise as a fulcrum around which the security of our embedded ecology can be raised. SRAM PUFs appear to offer the best compromise in the characteristics required for the archetypal PUF as the technology filters into mainstream application and Fuzzy Extraction seems the likely model for dealing with the inherent noisy output that is unavoidable, while the best subcomponents for error correction and hashing have yet to be discovered. The use of PUFs in the generation of Message Authentication Codes is still in its infancy and presents an interesting avenue of research.

With this in mind the project as proposed offers a rudimentary and earnest start to the exploration of this exciting field and should provide interesting insights into both the application of PUFs in general and their adaptability and suitability to the task of networked authentication via Message Authentication Codes.

6. REFERENCES

[1] F. Stajano, and R. Anderson, "The Resurrecting Duckling: security issues for ubiquitous computing," *Computer,* vol. 35, no. 4, pp. 22-26, 2002.

[2] M. Rostami, J. B. Wendt, M. Potkonjak, and F. Koushanfar, "Quo vadis, PUF?."

[3] R. Maes, "Physically Unclonable Functions: Constructions, Properties and Applications," p. 5: Springer Berlin Heidelberg, 2013.

[4] R. van den Berg, "Entropy analysis of Physical Unclonable Functions," MSc. thesis, Eindhoven University of Technology, 2012.

[5] Y. Meng-Day, R. Sowell, A. Singh, D. M'Raihi, and S. Devadas, "Performance metrics and empirical results of a PUF cryptographic key generation ASIC." pp. 108-115.

[6] J. Guajardo, S. S. Kumar, and P. Tuyls, "Key Distribution for Wireless Sensor Networks and Physical Unclonable Functions," *Printed handout of Secure Component and System Identification—SECSI*, pp. 17-18, 2008.

[7] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning physically unclonable functions." pp. 1-6.

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

[9] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura, "Performance Analysis for PUF Data Using Fuzzy Extractor," *Ubiquitous Information Technologies and Applications*, pp. 277-284: Springer, 2014.

[10] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," *Cryptographic Hardware and Embedded Systems - CHES 2007*, Lecture Notes in Computer Science P. Paillier and I. Verbauwhede, eds., pp. 63-80: Springer Berlin Heidelberg, 2007.

[11] R. H. Weber, and R. Weber, *Internet of Things*: Springer, 2010.

[12] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science,* vol. 297, no. 5589, pp. 2026-2030, 2002.

[13] R. Posch, "Protecting devices by active coating," *Journal of Universal Computer Science,* vol. 4, no. 7, pp. 652-668, 1998.

[14] K. Lofstrom, W. R. Daasch, and D. Taylor, "IC identification circuit using device mismatch." pp. 372-373.

[15] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal,* vol. 27, no. 1, pp. 379-423, 623-656, July, October, 1948, 1948.

[16] R. Maes, "Physically Unclonable Functions: Constructions, Properties and Applications," pp. 51-64: Springer Berlin Heidelberg, 2013.

[17] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions." pp. 148-160.

[18] 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 for identification and authentication applications." pp. 176-179.

[19] Z. C. Jouini, J. L. Danger, and L. Bossuet, "Performance evaluation of physically unclonable function by delay statistics." pp. 482-485.

[20] J. H. Anderson, "A PUF design for secure FPGA-based embedded systems." pp. 1-6.

[21] Y. Su, J. Holleman, and B. P. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations," *Solid-State Circuits, IEEE Journal of,* vol. 43, no. 1, pp. 69-77, 2008.

[22] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices."

[23] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls, "The butterfly PUF protecting IP on every FPGA." pp. 67-70.

[24] P. Simons, E. van der Sluis, and V. van der Leest, "Buskeeper PUFs, a promising alternative to D flip-flop PUFs." pp. 7-12.

[25] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. Van der Sluis, and V. van der Leest, "Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS." pp. 486-489.

[26] B. Cheng, S. Roy, and A. Asenov, "The impact of random doping effects on CMOS SRAM cell." pp. 219-222.

[27] "SRAM Circuit Design and Operation," *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies*, Frontiers In Electronic Testing, pp. 13-38: Springer Netherlands, 2008.

[28] R. Van Den Berg, B. Škorić, and V. Van Der Leest, "Bias-based modeling and entropy analysis of PUFs." pp. 13-20.

[29] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*: Springer Berlin Heidelberg, 2013.

[30] A. Juels, and M. Wattenberg, "A fuzzy commitment scheme." pp. 28-36.

[31] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong

keys from biometrics and other noisy data." pp. 523-540.

[32] C. H. Bennett, G. Brassard, C. Crepeau, and U. M. Maurer, "Generalized privacy amplification," *Information Theory, IEEE Transactions on,* vol. 41, no. 6, pp. 1915-1923, 1995.

[33] Y. Dodis, J. Katz, L. Reyzin, and A. Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," *Advances in Cryptology-CRYPTO 2006*, pp. 232-250: Springer, 2006.

[34] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs, "Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors," *Advances in Cryptology – EUROCRYPT 2008*, Lecture Notes in Computer Science N. Smart, ed., pp. 471-488: Springer Berlin Heidelberg, 2008.

[35] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis, "Fuzzy extractors for continuous distributions," in Proceedings of the 2nd ACM symposium on Information, computer and communications security, Singapore, 2007, pp. 353-355.

[36] X. Boyen, "Reusable cryptographic fuzzy extractors," in Proceedings of the 11th ACM conference on Computer and communications security, Washington DC, USA, 2004, pp. 82-91.

[37] P. Tuyls, B. Škoriç, and T. Kevenaar, *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*: Springer, 2007.

[38] B. Škorić, P. Tuyls, and W. Ophey, "Robust key extraction from physical uncloneable functions," *Applied Cryptography and Network Security*, pp. 99-135, 2005.

[39] A. Arakala, J. Jeffers, and K. J. Horadam, "Fuzzy Extractors for Minutiae-Based Fingerprint Authentication," *Advances in Biometrics*, Lecture Notes in Computer Science S.-W. Lee and S. Li, eds., pp. 760-769: Springer Berlin Heidelberg, 2007.

[40] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient Helper Data Key Extractor on FPGAs," *Cryptographic Hardware and Embedded Systems – CHES 2008*, Lecture Notes in Computer Science E. Oswald and P. Rohatgi, eds., pp. 181-197: Springer Berlin Heidelberg, 2008.

[41] F. Hao, R. Anderson, and J. Daugman, "Combining cryptography with biometrics effectively," *University of Cambridge Computer Laboratory, Tech. Rep*, 2005.

[42] N. I. o. S. a. Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," 2014.

[43] E. Homsirikamol, M. Rogawski, and K. Gaj, "Comparing hardware performance of round 3 SHA-3 candidates using multiple hardware architectures in Xilinx and Altera FPGAs." pp. 19-20.

[44] G. D. Bertoni, Joan; Peeters, Michaël; Van Assche, Gilles. "The Keccak sponge function family," http://keccak.noekeon.org/.

[45] K. H. Tsoi, K. H. Leung, and P. H. W. Leong, "Compact FPGA-based true and pseudo random number generators." pp. 51-61.

[46] M. Dichtl, and J. D. Golić, "High-speed true random number generation with logic gates only," *Cryptographic Hardware and Embedded Systems-CHES 2007*, pp. 45-62: Springer, 2007.

[47] M. Majzoobi, F. Koushanfar, and S. Devadas, "FPGA-based true random number generation using circuit metastability with adaptive feedback control," *Cryptographic Hardware and Embedded Systems–CHES 2011*, pp. 17-32: Springer, 2011.

[48] A. Al Azad, M. Huq, and I. R. Rokon, "Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog."

[49] N. Sklavos, and O. Koufopavlou, "Implementation of the SHA-2 hash family standard using FPGAs," *The Journal of Supercomputing,* vol. 31, no. 3, pp. 227-248, 2005.

[50] P. Alfke, "Efficient Shift Registers, LFSR Counters, and Long PseudoRandom Sequence Generators," XAPP052, Xilinx, 1996.

[51] Altera, "DE1 Development and Education Board User Manual ".

[52] C. Böhm, and M. Hofer, "Physical Unclonable Functions in Theory and Practice," p. 12: Springer, 2012.

[53] "Microchip 23LCV512 Datasheet," Microchip, ed., 2012.