JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 10, ISSUE 04, 2023

Securing Emerging Non-Volatile Main Memory with Fast and Energy-Efficient AES In-Memory Implementation

K. Naga Dasaradha[1], M Sai Shivani[2], Nalla Supriya[2], Manthapuri Pallavi[2], Madanu Bala Sangeetha[2]

[1,2]Department of Electronics and Communication Engineering

[1,2]Malla Reddy Engineering College for Women (A), Maisammaguda, Medchal, Telangana.

Abstract

Power analysis attacks (PAAs), a class of side channel attacks based on power consumption measurements, are a major concern in the protection of secret data stored in cryptographic devices. In this paper, we introduce the secure double rate registers (SDRRs) as a register transfer level (RTL) countermeasure to increase the security of cryptographic devices against PAAs. We exploit the SDRR in a conventional advanced encryption standard (AES)-128 architecture, improving the immunity of the cryptographic hardware to the state-of-the-art PAAs. In the AES-128 exploiting SDRR, the combinational path evaluates random data throughout the entire clock cycle, and the interleaved processing of random and real data ensures the protection of both combinational and sequential logics. Our technique does not require the duplication of the combinational path to process the random data, thus limiting area overhead, unlike previous RTL countermeasures. The proposed approach is validated by means of PAAs based on real measurements on a field-programmable gate array implementation and on a 65-nm CMOS prototype chip. The protected implementation shows a strongly reduced correlation coefficient for the correct key, and more than three orders of magnitude increase in the measurements to disclosure with respect to the unprotected AES-128.

Keywords: Power analysis attack, secure double rate registers, register transfer level.

1. INTRODUCTION

1.1 Introduction To AES

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. It was ratified as a federal standard in May 2002. AES is the most recent of the four current algorithms approved for federal us in the United States. One should not compare AES with RSA, another standard algorithm, as RSA is a different category of algorithm. Bulk encryption of information itself is seldom performed with RSA.RSA is used to transfer other encryption keys for use by AES for example, and for digital signatures. AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. The only secret necessary to keep for security is the key. AES may configure to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. Each additional bit in the key effectively doubles the strength of the algorithm, when defined as the time necessary for an attacker to stage a brute force attack, i.e. an exhaustive search of all possible key combinations in order to find the right one

1.2 Some background on AES

In 1997 the US National Institute of Standards and Technology put out a call for candidates for a replacement for the ageing Data Encryption Standard, DES. 15 candidates were accepted for further consideration, and after a fully public process and three open international conferences, the number of candidates was reduced to five. In February 2001, the final candidate was announced and comments were solicited. 21 organizations and individuals submitted comments. None had any reservations about the suggested algorithm. AES is founded on solid and well-published mathematical ground, and appears to resist all known attacks well. There's a strong indication that in fact no back-door or known weakness exists since it has been published for a long time, has been the subject of intense scrutiny by researchers all over the world, and such enormous amounts of economic value and information is already successfully protected by AES. There are no unknown factors in its design, and it was developed by Belgian researchers in Belgium therefore voiding the conspiracy theories sometimes voiced concerning an encryption standard developed by a United States government agency. A strong encryption algorithm need only meet only single main criteria:

- There must be no way to find the unencrypted clear text if the key is unknown, except brute force, i.e. to try all possible keys until the right one is found.

A secondary criterion must also be met:

- The number of possible keys must be so large that it is computationally infeasible to actually stage a successful brute force attack in short enough a time.

The older standard, DES or Data Encryption Standard, meets the first criterion, but no longer the secondary one – computer speeds have caught up with it, or soon will. AES meets both criteria in all of its variants: AES-128, AES-192 and AES-256.

1.3 Encryption must be done properly

AES may, as all algorithms, be used in different ways to perform encryption. Different methods are suitable for different situations. It is vital that the correct method is applied in the correct manner for each and every situation, or the result may well be insecure even if AES as such is secure. It is very easy to implement a system using AES as its encryption algorithm, but much more skill and experience is required to do it in the right way for a given situation. No more than a hammer and a saw will make anyone a good carpenter, will AES make a system secure by itself. To describe exactly how to apply AES for varying purposes is very much out of scope for this short introduction.

1.4 Strong keys

Encryption with AES is based on a secret key with 128, 192 or 256 bits. But if the key is easy to guess it doesn't matter if AES is secure, so it is as critically vital to use good and strong keys as it is to apply AES properly. Creating1 good and strong keys is a surprisingly difficult problem and requires careful design when done with a computer. The challenge is that computers are notoriously deterministic, but what is required of a good and strong key is the opposite – unpredictability and randomness. Keys derived into a fixed length suitable for the encryption algorithm from passwords or pass phrases typed by a human will seldom correspond to 128 bits much less 256. To even approach 128--bit equivalence in a pass phrase, at least 10 typical passwords of the kind frequently used in day-to-day work are needed. Weak keys can be somewhat strengthened by special techniques by adding computationally intensive steps which increase the amount of computation necessary to break it. The risks of incorrect usage, implementation and weak keys are in no way unique for AES; these are shared by all encryption algorithms. Provided that the implementation is correct, the security provided reduces to a relatively simple question about how many bits the chosen key, password or pass phrase

really corresponds to. Unfortunately this estimate is somewhat difficult to calculate, when the key is not generated by a true random generator.

1.5 Security is relative

Security is not an absolute; it's a relation between time and cost. Any question about the security of encryption should be posed in terms of how long time, and how high cost will it take an attacker to find a key? Currently, there are speculations that military intelligence services possibly have the technical and economic means to attack keys equivalent to about 90 bits, although no civilian researcher has actually seen or reported of such a capability. Actual and demonstrated systems today, within the bounds of a commercial budget of about 1 million dollars can handle key lengths of about 70 bits. An aggressive estimate on the rate of technological progress is to assume that technology will double the speed of computing devices every year at an unchanged cost. If correct, 128-bit keys would be in theory be in range of a military budget within 30-40 years. An illustration of the current status for AES is given by the following example, where we assume an attacker with the capability to build or purchase a system that tries keys at the rate of one billion keys per second. This is at least 1 000 times faster than the fasted personal computer in 2004. Under this assumption, the attacker will need about 10 000 000 000 000 000 000 000 years to try all possible keys for the weakest version, AES-128. The key length should thus be chosen after deciding for how long security is required, and what the cost must be to brute force a secret key. In some military circumstances a few hours or days security is sufficient – after that the war or the mission is completed and the information uninteresting and without value. In other cases a lifetime may not be long enough

2. LITERATURE SURVEY

The literature survey focuses its attention towards AES, particularly to utilize under low power consumption, high security, better performance and improved efficiency. The implementation feasibility in VLSI environment is also studied and analyzed in depth.

2.1 Fault Analysis in AES-CBC Algorithm Using Hamming Code for Space Applications

National institute of standard and technology (2001) presented computer security. Two FIPS publications already prove the modes of operation for two particular block cipher algorithms [60]. Four of these modes are equivalent to the ECB, CBC, CFB, and OFB modes with the Triple DES algorithm (TDEA) as the underlying block cipher. For any given key, the block cipher algorithm of the mode consists of two function that are inverses of each other

Francois-Xavier Standaert, Gael Rouvroy, Jean-Jacques Quisquater, and JeanDidier Legat presented (2004) discussed about the Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware [31]. It addressed various approaches for efficient FPGA implementations of the Advanced Encryption Standard algorithm. In implementation of block ciphers, several strategies can produce effective designs. Inherent constraints of FPGAs were taken into account in order to define an efficient methodology. Inside these architectures, the authors proposed algorithmic optimizations for the substitution box, and also efficient combinations between the diffusion layer and the key addition.

Farhadian.A and Aref.M.R (2009) presented efficient method for simplifying and approximating the s-boxes based on power functions [29]. In this paper cipher algorithms, power functions over finite fields and special inversion functions have an important role in the S-box design structure. A new systematic efficient method is introduced to crypt analyze such S-boxes. This method is very simple and does not need any heuristic attempt and can be considered as a quick criterion to find some simple approximations. Using this new method, approximations can be obtained for advanced encryption

standard (AES) like S-boxes, such as AES, Camellia, and Shark and so on. Finally as an application of this method, a simple linear approximation for AES S-box is presented.

Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh (2011) presented a Compact Rijndael Hardware Architecture with S-Box Optimization [2]. Encryption and decryption data paths are combined and all arithmetic components are reused. An extremely small size of 5.4 K gates is obtained for a 128-bit key Rijndael circuit using a 0.11-µm CMOS standard. In order to minimize the hard-ware size, the order of the arithmetic functions were changed, and the encryption-decryption data paths are efficiently combined with respect to cell library. Logic optimization techniques such as factoring were applied to the arithmetic components, and gate counts were reduced.

Ashkan Masoomi and Roozbeh Hamzehiyan (2012) presented a new approach for detecting and correcting errors in satellite communications based on Hamming Error Correcting Code [6]. A novel model to detect and correct Single Event Upsets in on-board implementations of the AES algorithm was based on hamming error correcting code. Single Even Upset (SEU) faults occur in the on-board during encryption due to radiation. Some of the AES modes like ECB, CBC, OFB, CFB and CTR performances have been analyzed. From that, CTR mode has been recommended as the optimum choice for satellite applications.

Ramesh Babu, George Abraham and Kiransinh Borasia (2012) presented a Review on Securing Distributed Systems Using Symmetric Key Cryptography [66]. It was used to evaluate the importance of Symmetric Key Cryptography for Security in Distributed Systems. Two symmetric key cryptographic algorithms DES and AES were commonly used. These two algorithms were evaluated on the parameters such as key size, block size, number of iterations. From the literatures reviewed of various implementations and analysis of both the algorithms, it can be concluded that AES algorithm has over-shadowed the DES algorithm in many areas.

Karri, R., Wu, K., Mishra, P., and Kim, Y. (2002) Concurrent Error Detection Schemes for Fault-Based Side-Channel Cryptanalysis of Symmetric Block Ciphers [45]. They presented algorithm level, round level, and operation level CED (Concurrent Error Detection) architectures for symmetric block ciphers. The algorithm was independent and can be applied to almost any symmetric block cipher. The proposed scheme introduced moderate area overhead and interconnects complexity to achieve permanent as well as transient fault tolerance. This approach assumes that the key RAM, comparator, or both encryption and decryption modules simultaneously are not under attack or faulty.

Ross Anderson, Eli Biham, Lars Knudsen (1999) presented a Proposal for the Advanced Encryption Standard [71]. Its design is highly conservative, yet still allows a very efficient implementation. With a 128-bit block size and a 256-bit key, it is as fast as DES on the market leading Intel Pentium/MMX platforms yet we believe it to be more secure than three-key triple-DES. The linear transformations were just bit 27 permutations, which were applied as rotations of the 32-bitwords in the bit slice implementation. The authors also considered replacing the XOR operations by seemingly more complex operations, such as additions. Finally cognate algorithms with the same structure as Serpent but with block sizes of 64, 256 and 512 bits.

A.J.Elbirt, W.Yip, B.Chetwynd, C.Paar (2000) presented an FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists [27]. Reprogrammable devices such as Field Programmable Gate Arrays (FPGAs) are highly attractive options for hardware implementations of encryption algorithms as they provide cryptographic algorithm agility, physical security, and potentially much higher performance than software solutions. The implementations of each algorithm will be compared in an effort to determine the most suitable candidate for hardware implementation within commercially available FPGAs. A design methodology was established which

in turn led to the architectural requirements for a target FPGA. The best speed-optimized implementations were identified for each AES finalist in both non-feedback and feedback modes.

Pawel Chodowiec, Kris Gaj, Peter Bellows and Brian Schott (2001) presented Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board [64]. Full implementations of the new Advanced Encryption Standard, Rijndael, and the older American federal standard, Triple DES, were developed and experimentally tested using the SLAAC-1V FPGA accelerator board, based on Xilinx Virtex 1000 devices. Demonstration of a capability to enhance our circuit to handle the encryption and decryption throughputs of over 1 Gbit/s regardless of the chosen algorithm. For Rijndael in the basic iterative architecture, the results for encryption and decryption are different, with decryption slower than encryption by about 13% in experimental testing. The IPSec-compliant encryption/decryption units of the new Advanced Encryption Standard - Rijndael and 28 the older encryption standard Triple DES have been developed and tested experimentally.

3. PROPOSED METHOD

A. AES-128 Fundamentals

The AES-128 algorithm is a block cypher working on 128-bit wide data and key words [29]. Encryption consists in iterated operations, known as “rounds.” Each round is composed of suboperations working on a single byte, and known as “layers.” The AES-128 is composed of 11 rounds, which are in turn composed of four layers (except round 0 and round 10): SubstituteBytes, ShiftRows, MixColumns, and AddRoundKey. Usually, the target function of PAAs is the output of the AddRoundKey layer of round 0 or the output of the SubBytes layer of round 1; if round 0 key is hardwired in the device, an attack is successful if the attacker can recover this secret key. Thus, the AddRoundKey and the SubstitutionBytes phases are the most critical from a security perspective.

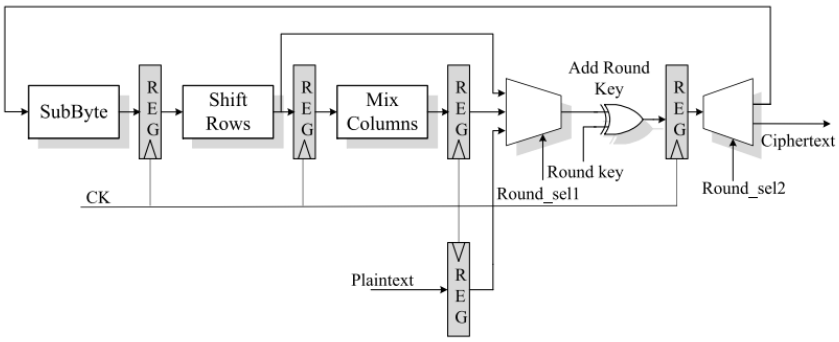

Fig. 1: Architecture block diagram of the reference AES-128 encryption unit (AES-0) (control path and key scheduler are not shown).

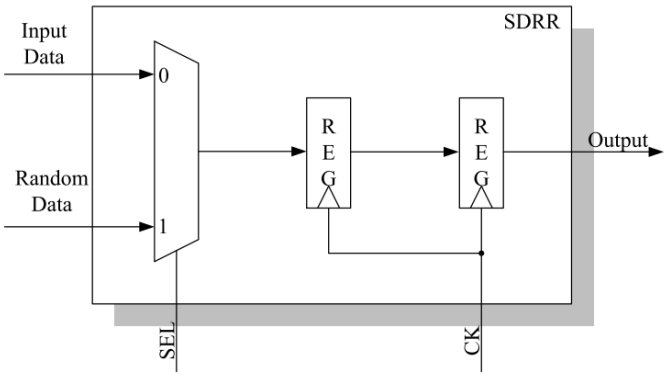

Fig. 2: Block scheme of the proposed SDRR.

B. Architecture of the Reference AES Encryption Unit

The architecture of the reference AES encryption unit (AES-0) is shown in Fig. 1 [30]. The data path of the core has an iterative structure, with an inner-round pipeline composed of four separate blocks, each performing one layer of the AES round. Each of these four blocks is implemented by means of a combinational network and a pipeline register to store the datum at the output of each block. The architecture includes also a finite-state machine and additional control logic blocks, not shown. Four clock cycles are required to process a round, and the entire encoding process of a 128-bit plaintext block (11 rounds) takes 44 clock cycles. This architecture has been chosen because the resource usage (area) for the combinational logic is close to the resource usage (area) for sequential logic, resulting in balanced power consumption (and information leakage) between combinational logic and registers.

C. Proposed Secure Double Rate Register and the Protected AES-128 Architecture

Double-data-rate computation has been previously adopted to counteract fault analysis in [31], but at the best of our knowledge, this is the first work to exploit double-data-rate computation as a countermeasure against PAAs. The block scheme of the proposed SDRR is depicted in Fig. 2. The SDRR is composed of two cascaded registers and an input multiplexer which allows selecting the input data of the first register. The flip-flops in the SDRR are clocked by the CK signal (whose frequency is doubled with respect to the reference architecture clock). The clock signal of the reference architecture (SEL signal) is used to select between real and random data. By using the SDRR in place of conventional registers, when the real input datum is stored in one of the two registers of the SDRR, a random datum is stored in the other one and vice versa. The proposed AES architecture is shown in Fig. 3. It exploits a pipelined precharging technique to force the logic circuit to a random state Sr. In Fig. 3, the SEL signal can be thought as the clock signal of the reference AES architecture: SDRRs store data on both rising and following edges of the SEL signal (according to the CK signal in Fig. 3). For this purpose, conventional registers are replaced by SDRRs. In each stage of the pipeline, the SDRR feeds the combinational logic alternating the correct and the random data. In this way, the correct data and random data are processed and stored simultaneously, exploiting the diffusion property of the cryptographic algorithm. To better explain this point, we refer to the power consumption model in [14].

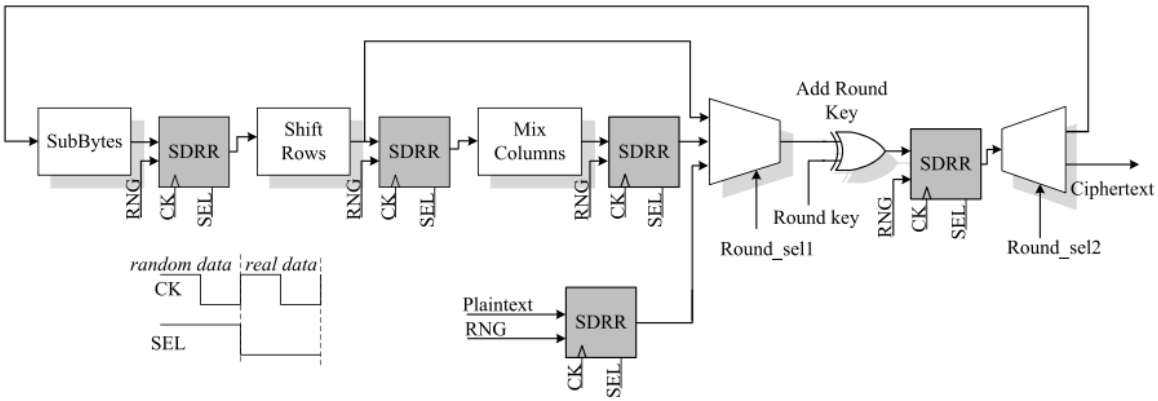

Fig. 3: Block diagram of the proposed counter measured AES architecture.

Avital et al. [20] have shown that the conventional RPL [14] offers a reduction of the signal-to-noise ratio (SNR) for the attacker, but this reduction depends strongly on the duration of the precharge phase. If the attacker gains knowledge of the duration of the precharge phase, he/she can retrieve the secret key by analyzing the correlation in a narrow time window around the switch from the precharge phase to the evaluation phase. The usage of the proposed SDRR blocks avoids this critical problem allowing the combinational path to evaluate the random data throughout the entire clock cycle, and not only a portion of it. The adoption of the interleaved processing of the random data along with real data ensures the protection of both the combinational and sequential logics and offers a twofold improvement. 1) It removes memory effects on the combinational path of a stage of the pipeline. 2) The presence of two registers in each SDRR unit ensures that the total consumption due to data storage is randomized by means of the simultaneous storage of the random data alongside with the real data. The proposed technique does not require the duplication of the data path for the random datum, and this limits the area and power consumption overhead. Furthermore, the proposed architecture does not suffer from critical problems related to the imbalance of the fan-out, because the real and random data share the same data path.

4. SIMULATION RESULTS

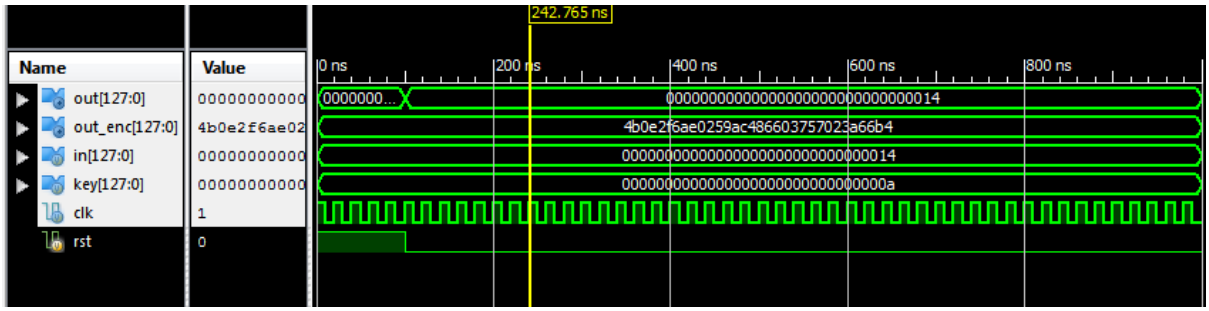

Fig. 4: Simulation output.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	8648	17600	49%
Number of fully used LUT-FF pairs	0	8648	0%
Number of bonded IOBs	514	100	514%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig. 5: Design summary.

```
LUT6:I0->O            1    0.043   0.350  e1/r1/r5/m3/m6/Mxor_b<3>_xo<0>1 (e
LUT5:I4->O            4    0.043   0.630  e1/r1/r5/Mxor_roundout_83_xo<0>1 (
LUT6:I0->O            5    0.043   0.626  e1/r1/r6/m1/s6/s1/Mxor_out_temp<3>
LUT6:I1->O            1    0.043   0.350  e1/r1/r6/m3/m4/Mxor_b<3>_xo<0>1 (e
LUT6:I5->O            4    0.043   0.630  e1/r1/r6/Mxor_roundout_99_xo<0>1 (
LUT6:I0->O            5    0.043   0.636  e1/r1/r7/m1/s4/s1/Mxor_out_temp<3>
LUT6:I0->O            1    0.043   0.350  e1/r1/r7/m3/m6/Mxor_b<3>_xo<0>1 (e
LUT5:I4->O            4    0.043   0.630  e1/r1/r7/Mxor_roundout_83_xo<0>1 (
LUT6:I0->O            5    0.043   0.626  e1/r1/r8/m1/s6/s1/Mxor_out_temp<3>
LUT6:I1->O            1    0.043   0.350  e1/r1/r8/m3/m4/Mxor_b<3>_xo<0>1 (e
LUT6:I5->O            4    0.043   0.630  e1/r1/r8/Mxor_roundout_99_xo<0>1 (
LUT6:I0->O            5    0.043   0.636  e1/r1/r9/m1/s4/s1/Mxor_out_temp<3>
LUT6:I0->O            1    0.043   0.350  e1/r1/r9/m3/m5/Mxor_b<3>_xo<0>1 (e
LUT3:I2->O            7    0.043   0.556  e1/r1/r9/Mxor_roundout_91_xo<0>1 (
LUT6:I2->O            5    0.043   0.362  e1/r1/r10/Mxor_roundout_91_xo<0>1
OBUF:I->O                  0.000          out_enc_91_OBUF (out_enc<91>)
----------------------------------------
Total                     17.309ns (1.290ns logic, 16.019ns route)
                                   (7.5% logic, 92.5% route)
```

5. CONCLUSION

In this paper, we have introduced the SDRR as an RTL countermeasure to increase the security of cryptographic implementations to PAAs. The proposed SDRR technique has been exploited to protect an AES-128 cryptographic core. The proposed approach allows the combinational path to process the random data throughout the clock cycle and the sequential logic to store the real and random data simultaneously, without duplicating the combinational path for the random data. We have validated the proposed RTL countermeasure by means of PAAs carried out with real measurements on both an FPGA implementation and a 65-nm CMOS prototype chip. The protected implementations showed a strongly reduced correlation coefficient of the correct key and more than three orders of magnitude increase in the measurements to disclosure with respect to the unprotected AES-128. The MTD increased by more than three orders of magnitude also in the case of technology-oriented PAAs. To provide further validation of the proposed approach, we referred also to information-theoretic security metrics. The validation results based on SNR, MI, and Welsh's t-test data demonstrated the effectiveness of the proposed technique in counteracting PAAs using dynamic power. The area overhead resulted to be 33%, whereas power consumption, despite a nearly threefold increase, is reasonable, given that the data path has twice the registers and twice the clock frequency compared with the reference architecture. The higher power consumption overhead with respect to other RTL countermeasures is justified by the higher level of protection achieved by the proposed architecture, which guarantees the protection of the full AES-128 core and not only limited parts of it, as in other works.

REFERENCES

[1] Madakam, Somayya, R. Ramaswamy, and Siddharth Tripathi. "Internet of Things (IoT): A literature review." Journal of Computer and Communications 3, no. 05 (2015): p.164.

[2] [2] Wang, Yong, Garhan Attebury, and Byrav Ramamurthy. "A survey of security issues in wireless sensor networks." IEEE Communications Surveys Tutorial (2006).

[3] Veeramallu, B., S. Sahitya, and Ch LavanyaSusanna. Veeramallu, B., S. Sahitya, and Ch LavanyaSusanna. "Confidentiality in Wireless sensor Networks." International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.

[4] Eisenbarth, Thomas, and Sandeep Kumar. "A survey of lightweightcryptography implementations." IEEE Design & Test of Computers 24.6 (2007).

[5] Banik, Subhadeep, Andrey Bogdanov, and Francesco Regazzoni. "Exploring energy efficiency of lightweight block ciphers." International Conference on Selected Areas in Cryptography. Springer, Cham, 2015.

[6] Bogdanov, Andrey, et al. "PRESENT: An ultra-lightweight block cipher." CHES. Vol. 4727. 2007.

[7] Borghoff, Julia, et al. "PRINCEa low-latency block cipher for pervasive computing applications." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2012.

[8] Beaulieu, Ray, et al. "The SIMON and SPECK lightweight block ciphers." Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE. IEEE, 2015.

[9] Suzaki, Tomoyasu, et al. "TWINE: A Lightweight Block Cipher for Multiple Platforms." Selected Areas in Cryptography. Vol. 7707. 2012.

[10] Li, Wei, et al. "Security analysis of the LED lightweight cipher in the internet of things." Jisuanji Xuebao(Chinese Journal of Computers) 35.3 (2012): p.434-445.

[11] Shibutani, Kyoji, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. "Piccolo: An ultra-lightweight blockcipher." In CHES, vol. 6917, pp. 342-357. 2011.

[12] Wu, Wenling, and Lei Zhang. "LBlock: a lightweight block cipher." In Applied Cryptography and Network Security, pp. 327-344. Springer Berlin/Heidelberg, 2011.

[13] Daemen, Joan and Rijmen, Vincent. "The design of Rijndael: AES-the advanced encryption standard.", Springer Science & Business Media, 2013.

[14] Descriptions of SHA-256, SHA-384, and SHA-512. http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384- 512.pdf.

[15] Al Hasib, Abdullah, and Abul Ahsan Md Mahmudul Haque. "A comparative study of the performance and security issues of AES and RSA cryptography." Third International Conference on Convergence and Hybrid Information Technology, 2008. Vol.2.

[16] Feldhofer, Martin, Johannes Wolkerstorfer, and Vincent Rijmen. "AES implementation on a grain of sand." IEE Proceedings-Information Security 152, no. 1 (2005): p.13-20.

[17] Moradi, Amir, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. "Pushing the limits: a very compact and a threshold implementation of AES." In Eurocrypt, vol. 6632, pp. 69-88. 2011.

[18] Hocquet, Cdric, Dina Kamel, Francesco Regazzoni, Jean-Didier Legat, Denis Flandre, David Bol, and Franois-Xavier Standaert. "Harvesting the potential of nano-CMOS for lightweight cryptography: an ultra-lowvoltage 65 nm AES coprocessor for passive RFID tags." Journal of Cryptographic Engineering 1, no. 1 (2011): p.79-86.

[19] Kerckhof, Stphanie, Franois Durvaux, Cdric Hocquet, David Bol, and Franois-Xavier Standaert. "Towards green cryptography: a comparison of lightweight ciphers from the energy viewpoint." Cryptographic Hardware and Embedded SystemsCHES 2012 (2012): p.390-407.

[20] Batina, Lejla, et al. "Dietary recommendations for lightweight block ciphers: power, energy and area analysis of recently developed architectures." International Workshop on Radio Frequency Identification: Security and Privacy Issues. Springer, Berlin, Heidelberg, 2013.

[21] Banik, Subhadeep, Andrey Bogdanov, and Francesco Regazzoni. "Exploring the energy consumption of lightweight blockciphers in FPGA." International Conference on ReConFigurable Computing and FPGAs (ReConFig), 2015 , pp.1-6.

[22] Kong, Jia Hao, Li-Minn Ang, and Kah Phooi Seng. "A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments." Journal of Network and Computer Applications 49 (2015): p.15-50.

[23] Wenceslao Jr, Felicisimo V., et al. "Modified AES Algorithm Using Multiple S-Boxes." The Second International Conference on Electrical, Electronics, Computer Engineering and their Applications (EECEA2015). 2015.

[24] Kawle, Pravin, et al. "Modified Advanced Encryption Standard." International Journal of Soft Computing and Engineering (IJSCE) 4 (2014).