Microelectronics Reliability 126 (2021) 114349

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

Impact of radiation-induced soft error on embedded cryptography algorithms

Vitor Bandeira [a], Jack Sampford [b], Rafael Garibotti [c], Matheus Garay Trindade [d], Rodrigo Possamai Bastos [d], Ricardo Reis [a], Luciano Ost [e,*]

[a] Instituto de Informática, PGMicro, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brazil
[b] Phixos, United Kingdom
[c] School of Technology, Pontifical Catholic University of Rio Grande do Sul, Brazil
[d] Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France * Institute of Engineering Univ. Grenoble Alpes, France
[e] Wolfson School, Loughborough University, Loughborough, United Kingdom

ARTICLE INFO

Keywords:
Radiation test
Reliability
FPGA
XTEA
AES
Cryptography
Radiation-induced soft errors

ABSTRACT

With the advance of autonomous systems, security is becoming the most crucial feature in different domains, highlighting the need for protection against potential attacks. Mitigation of these types of attacks can be achieved using embedded cryptography algorithms, which differ in performance, area, and reliability. This paper compares hardware implementations of the eXtended Tiny Encryption Algorithm (XTEA) and the Advanced Encryption Standard (AES) algorithms. Results show that the XTEA implementation gives the best relative performance (e.g., throughput, power), area, and soft error reliability trade-offs.

1. Introduction

Cryptography algorithms play a key role in daily practices (e.g., e-payment, data exchange) in several sectors of society, including financial, healthcare and government institutions. The implementation of cryptography algorithms in low-level hardware designs presents a unique set of constraints (e.g., hardware and computation resources) and additional performance metrics (e.g., power consumption) to be optimised when compared to software solutions. With these constraints in mind, different lightweight yet robust optimisation techniques have been thoroughly investigated in both ASIC [1,2] and FPGA [3,4].

Although versatile and cost-efficient, cryptography implementations on SRAM-based FPGAs are highly susceptible to radiation-induced soft errors, and the investigation of reliable solutions is of significant concern [5]. In this regard, different mitigation techniques and implementation schemes have been used to reduce the impact of soft errors on cryptography solutions implemented on FPGAs [6]. For instance, Bertoni et al. [5] use redundancy techniques in conjunction with error-detecting codes to detect single-bit faults. Banu et al. [7] describe an AES fault-tolerant model based on Hamming's error correction code. Similarly, Wu et al. [8] proposed a low-cost concurrent error detection for the AES using parity checking. While the above authors rely on either simulation or emulation approaches, the works in [9,10,11] conducted laser-induced fault experiments in dedicated cryptography hardware and software AES implementations.

AES implementation on hardware can require a significant area on a device, which is a premium resource within an embedded system. In this regard, alternative lightweight encryption algorithms, such as the eXtended Tiny Encryption Algorithm (XTEA), are increasingly gaining ground. Implementations of XTEA on FPGAs and ASICs have been compared to AES solutions [12], considering area and performance efficiency.

Differently from the above works, this paper *contributes* by assessing the effects of radiation-induced soft errors on the operation of AES and XTEA implementations on FPGA. Gathered results have been obtained through neutron radiation tests conducted with a neutron generator, considering all three AES standardised forms: AES-128, AES-192 and AES-256, and a 128-bit data block size XTEA implementation. The proposed implementation supports performing two sets of XTEA encryption or decryption calculations in parallel using the same 128-bit key and the same set of subkeys.

The rest of this paper is organised as follows. Section 2 describes the

* Corresponding author.
E-mail address: l.ost@lboro.ac.uk (L. Ost).

https://doi.org/10.1016/j.microrel.2021.114349
Received 21 May 2021; Received in revised form 2 August 2021; Accepted 17 August 2021
Available online 2 September 2021

implementation of two cryptography algorithms successfully validated on FPGA. Next, Section 3 presents the radiation test flow and set-up used to assess the soft error reliability of developed cryptography algorithms. In Section 4, area, performance, power consumption and soft error reliability results are discussed and evaluated. Finally, Section 5 points out conclusions.

2. Cryptography algorithm selection

This work considers two cryptography algorithm solutions: the Advanced Encryption Standard (AES) and the eXtended Tiny Encryption Algorithm (XTEA).

AES is a symmetric cypher with a fixed block size of 128 bits and three key size settings: 128, 192, and 256 bits. AES performs four different transformations: Substitute-Bytes, ShiftRows, MixColumns and AddRoundKey. Although the same steps are used for both encryption and decryption, the order in which these steps are performed differs. In addition, the key size defines the number of rounds in which the AES is performed. AES also represents a large, relatively complex algorithm considering implementation on FPGA hardware due to its nature as a substitution-permutation type network, rather than the Feistel network design of older encryption algorithms. In this case, the plain text data is split into blocks and each block is substituted with a new value. Then, these blocks are recombined and modified using a key to calculate the output of the encryption round, with this process being highly parallelisable.

XTEA is a symmetric block cypher consisting of exclusive-or, addition, and shift operations. It is defined as a block size of 64 bits and a key length of 128 bits. To provide a 128-bit implementation, we calculate in parallel two sets of 64-bit blocks. Due to its simplicity, it can be used in resource-constrained environments. Furthermore, XTEA is considered a Feistel cypher, i.e., encryption and decryption operate similarly and both iterate a round function a fixed number of times. The developed XTEA solution relies on the use of combinatorial sections of logic to reduce computation time. Combinatorial logic, in this case, refers to logic that is not tied to a clock signal, in theory, meaning that it operates instantly as soon as the inputs to it are updated.

3. Radiation test methodology

This section covers all relevant steps used to assess the impact of the radiation-induced soft error on embedded cryptography algorithms, including the description of the radiation test flow, which is essential to guarantee the reproducibility of the radiation tests by other researchers; the presentation of the neutron radiation facility and the FPGA-based device under test (DUT) used in this work; and the description of the adopted fault classification.

3.1. Radiation test flow

Fig. 1 shows the seven-step radiation test flow used in this work. Initially, the FPGA board is reset (*Board Reset*) and this step is repeated every time a new radiation campaign begins. Then, the *FPGA Setup* is the moment when the bitstream is loaded into the FPGA. Note that all cryptography algorithms have been previously synthesised. Next, *Cryptography Algorithm Execution* starts the embedded algorithm, resulting in the generation of the Golden Reference. The *Radiation Setup* configures the number of campaigns and rounds, where each campaign has *N* rounds of execution of the cryptography algorithm. The purpose is to leave the FPGA board exposed to radiation for sufficient time to analyse both single and cumulative faults. From there, radiation campaigns can begin at the *Radiation Exposure* step.

At the end of each campaign, a log file with the results is generated in the *Data Acquisition* step. The resulting file is then evaluated to determine whether the amount of campaigns is sufficient for a given configuration. If not, a new campaign is then initiated and the process returns to the *Board Reset* step. Note the importance of dividing tests into campaigns and execution rounds, because if a bit-flip crashes the FPGA, it will only affect the current campaign, i.e., this campaign will have fewer execution rounds. On the other hand, when a new campaign starts, the previous errors are cleared. Finally, after all campaigns, a *Data post-processing* phase is carried out to extract the soft error reliability from each cryptography algorithm.

3.2. Radiation test set-up

The radiation tests were carried out at the GENEPI2 neutron source, at the LPSC [13]. GENEPI2 is a 14-MeV neutron generator with a maximum flux that exceeds the natural flux of 14-MeV neutrons at 40,000 ft. by a factor of 10^{10}. Fig. 2 shows the device under test facing

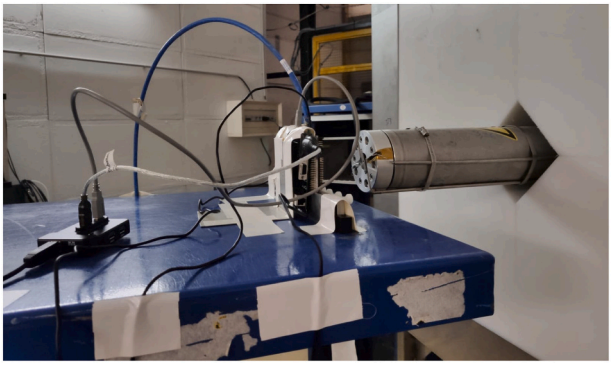

Fig. 2. Experimental setup with the irradiated FPGA facing the laser beam at the GENEPI2 facility.

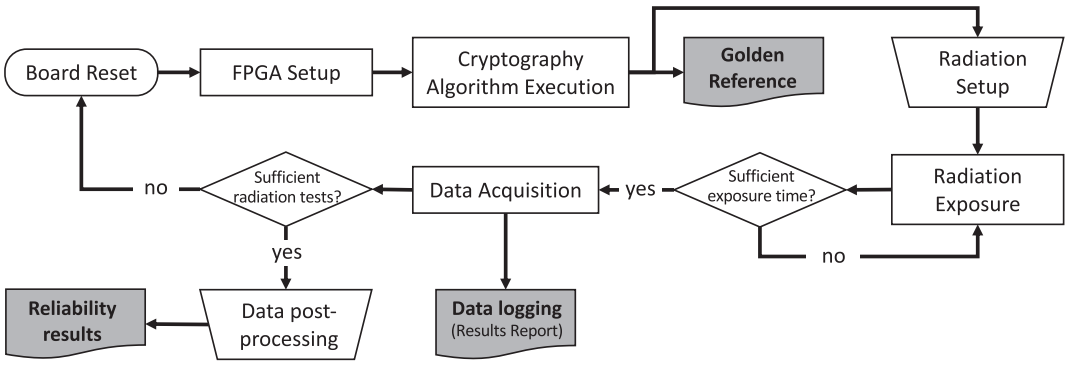

Fig. 1. Radiation test flow: covering from the FPGA initialisation to the reliability assessment of embedded cryptography algorithms.

the laser beam. The DUT was irradiated for 15 h and 10 min, yielding a total neutron fluence of 3.79 * 10^{11} n * cm^{-2} and an average flux of 6.910^6 n * cm^{-2}/s. Note that we did not use the maximum flux of the accelerator.

The device under test is the Arm MPS2+ FPGA (Altera Cyclone V — 5CEBA7F31C8N) prototyping board [14], which contains 300 K logic elements, 8 MB of SRAM and 16 MB of PSRAM. Each experiment performed in the radiation test flow consists of setting the key length of the cryptography algorithm, selecting between encryption or decryption, sending the key, as well as writing and reading back 16 bytes using the board's serial port. The purpose of generating a standard process for comparing cryptography algorithms is to assist in the reproducibility of future radiation tests.

3.3. Fault classification and metrics

To properly assess the impact of the radiation-induced soft error on the developed cryptography algorithms, the outcome of each experiment saved in the Results Report (Fig. 1) is compared with its Golden Reference in the *Data post-processing* step and classified according to one of the following classes: (a) Correct: the output matches the expected result; (b) Critical: when the received data has the same size (i.e., exactly 16 bytes read), but at least one bit of difference is identified; (c) Tolerable: when the output presents either less or more packets than expected or other modifications that would not affect the encryption/decryption operation.

The difference between *Critical* and *Tolerable* errors considers if the error is silent (i.e., the application ends without an error signal) or detectable (i.e., an error signal or unexpected behaviour). Silent errors are considered critical in this work as they might be propagated. In contrast, detectable errors can be tolerated as there is the possibility to rerun the algorithm to obtain the correct result. A similar fault classification is used to assess embedded algorithms under radiation-induced soft errors [15].

Further, while *Critical Errors* occur when the output differs from the Golden Reference, *Cumulative Errors* refer to the sum of all errors observed during the same campaign. For example, if the experiment's expected output is AA and the following is observed: (i) during the first 15 experiments the output is correct; (ii) starting on the 16th experiment the output becomes AB; (iii) finally, on the 25th experiment the output becomes CC. In this example, we consider that in (ii) and (iii) account for 2 *Cumulative Errors*.

Complementary to the fault classification, this work adopts the quantitative metric *mean work to failure (MWTF)* [16] to facilitate the analysis of soft errors. The MWTF shows the average amount of work that an application can perform until reaching a failure (i.e., higher values are better). This works uses the cryptography algorithms' runtime (i.e., the latency shown in Table 1) and the most critical vulnerability (i.e., critical errors) to calculate the MWTF, as shown in Eq. (1).

$$MWTF = \frac{1}{(execution\ time \times AVF_{Critical})} \quad (1)$$

The Architecture Vulnerability Factor (AVF) is used to measure the probability of a fault result in an error (i.e., SDC or Crash) [17]. This work uses the critical-based AVF as it includes the SDCs that actually lead to errors that might be propagated throughout the system. To calculate the $AVF_{Critical}$, the critical errors are divided by the number of experiments shown in Table 2.

4. Results

4.1. Performance, power and area evaluation

This Section provides the performance analysis of both RTL cryptography algorithm implementations in terms of FPGA area utilisation, dynamic power consumption, and data throughput. The FPGA utilisation comprises the number of registers, adaptive logic modules (ALMs) and block RAM (BRAM) bits required for each cryptography implementation. The throughput is calculated using the obtained values of latency in clock cycles and maximum achievable clock frequency, giving the amount of data that can be encrypted/decrypted per second. In turn, the dynamic power utilisation is measured according to the number of registers/ALMs used and their respective toggle rates. Table 1 shows that the XTEA implementation improves throughput by 48.6% and reduces ALM utilisation, register utilisation, and dynamic power consumption by 87.2%, 85.4%, and 86.5%, respectively.

The device utilisation of AES on the Cyclone V 5CEBA9F31C8 FPGA represents <1% of the available registers and 2.52% of the available ALUs. While the AES solution uses a negligible amount (0.01%) of the available BRAM bits, the developed XTEA implementation required any BRAM bits. The 2048 BRAM bits required by this implementation contain the S-box constants, the 256 8-bit constants used for the substitution process of the algorithm, which can be stored in BRAM rather than registers since they are constant values which are accessed using the value to be substituted as the address. These results indicate a poor efficiency concerning ALM utilisation compared to register utilisation, showing a near 1:1 ratio between the two. This is likely due to ALMs being used solely for the 8-input LUT element rather than the four register elements, since AES calculations involve many mathematical operations and intermediate stages, which would be implemented in combinatorial logic.

The XTEA performance results obtained show that the use of a lightweight encryption algorithm can provide key benefits over algorithms such as AES, when considering integration within an embedded design. When these implementations are designed with a focus on minimising area, and controlling for factors such as block size and duplex functionality, the use of XTEA over AES provides significant savings in terms of device area utilisation and power consumption. These savings are particularly significant when considering integration within resource-constraint devices.

4.2. Soft error assessment of AES and XTEA

This Section assesses the soft error impact on the AES and XTEA cryptography algorithms. Table 2 details the radiation exposure tests containing data from 13 campaigns and more than 90 thousand experiments (i.e., execution round for each cryptography algorithm).

During the radiation exposure, we ran campaigns with different numbers of experiments to gather more substantial data to assess the impact of cumulative errors on the cryptography algorithms. Figs. 3 and 4 show a timeline for the experiments detailed in Table 2.

Fig. 3 shows the comparison between the different AES versions,

Table 1
Comparison between AES and XTEA implementations.

Algorithm	Encryption/decryption latency (cycles)	Throughput @25 MHz (Mbit/s)	Register utilisation	ALM utilisation	Dynamic power consumption (mW)
AES-128	104	30.77	2959	2863	14.19
AES-192	124	25.81	2959	2863	14.19
AES-256	144	22.22	2959	2863	14.19
Duplex XTEA	70	45.71	878	769	3.66
Simplex XTEA	70	45.71	433	367	1.91

Table 2
Experimental details and soft error assessment analysis.

Attribute	AES-256		AES-192		AES-128		XTEA		Total
Campaigns	3		2		3		5		13
Mode	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption	
Experiments	2998	2758	3920	3583	11,917	4587	36,377	24,026	90,166
Critical + Tolerable = Cumulative Errors	3 + 1 = 4	6 + 1 = 7	2 + 0 = 2	4 + 1 = 5	6 + 3 = 9	4 + 2 = 6	10 + 6 = 16	4 + 7 = 11	39 + 21 = 60
Normalised MWTF	2.17	1.00	4.95	2.26	5.98	3.45	16.28	26.88	–

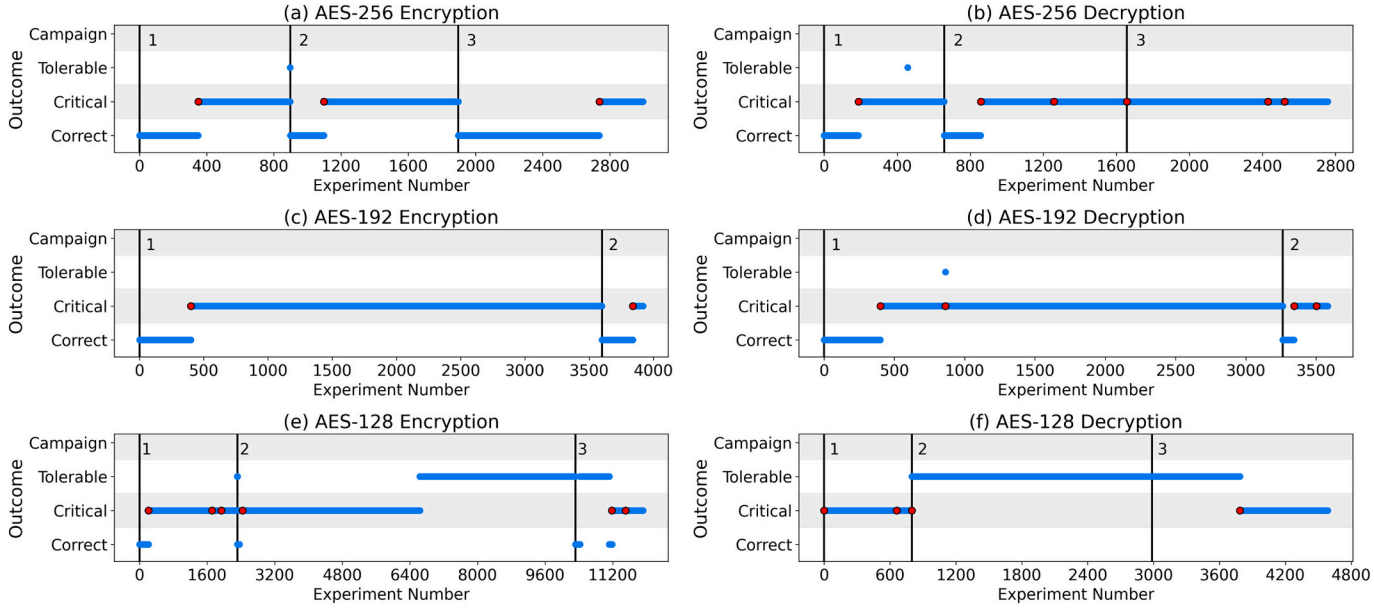

Fig. 3. Comparison between AES key sizes. Black lines delimit campaigns and red circles a new observable critical error. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

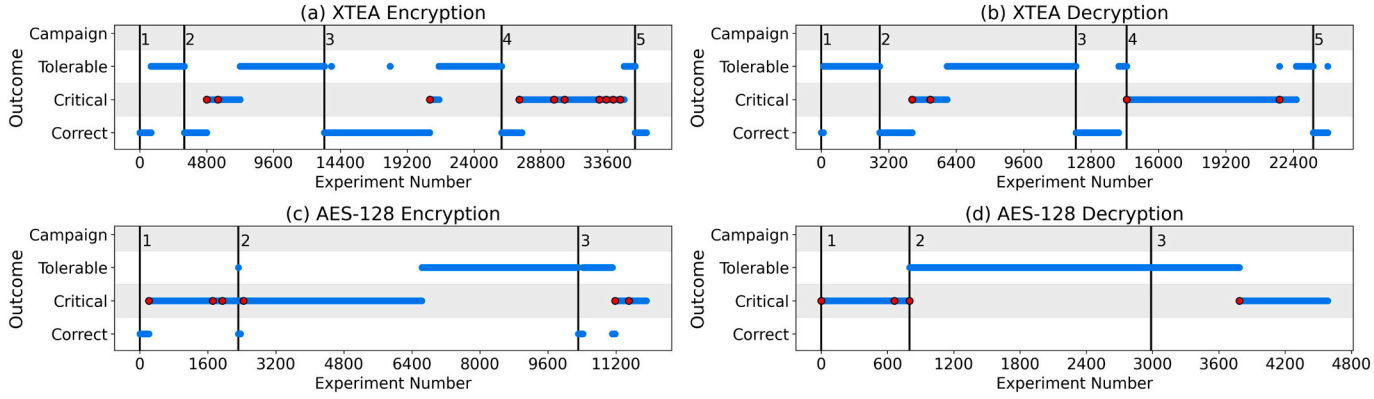

Fig. 4. AES-128 and XTEA comparison. Black lines delimit campaigns and red circles a new observable critical error. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

namely 128-, 192-, and 256-bit. AES algorithm implementations are particularly susceptible to errors [5]; thus, a single-bit flip in their early encryption rounds has a high probability of leading to an erroneous output, as clearly evidenced in Fig. 3 – with the exception of the AES-192 (Fig. 3(c)) that does not present tolerable errors. Notwithstanding the higher number of measurements that have been collected from 128-bit AES campaigns, the number of errors is similar to those collected from the other two AES solutions. Although a reasonable number of *Tolerable Errors* is found, the 128-bit AES solution still provides the best resilience to the occurrence of soft errors (w.r.t. 192- and 256-bit AES) as expressed by the MWTF and shown in the bottom row of Table 2.

Fig. 4 shows the comparison between AES-128 and XTEA. Results reveal that the higher the application throughput and the longer the exposure time, the higher the number of communication errors between the board and the host computer (Fig. 4 (a–b), and campaigns 2–3 of (c–d)). As expected, the longer exposure time also incurs an increased number of *Cumulative Errors*. For example, the campaign 4 of the XTEA encryption had 7 bit-flip modifications, which did affect the algorithm behaviour (*Critical Errors*). In turn, the 8th bit-flip lead to a *Tolerable Error*. Finally, the campaign 5 of the XTEA encryption is the only one that did not generate errors. This is believed to be due to the short campaign exposure time.

Another difference between the cryptography algorithms is their behaviour under the presence of flipped-bits. For example, the campaigns 3 (a), 3 (b), and 2 (d) in Fig. 3 had an error that put the algorithm in a loop, causing a repeated application output with the same wrong pattern with lengths of 4 to campaigns 3 (a) and 3 (b) and 14 to campaign 2 (d). These type of loop errors were not seen in XTEA. We believe that the complexity of AES algorithm control flow logic makes it more susceptible to entering a loop w.r.t. the XTEA.

4.3. Relative performance and reliability trade-off

The radar plot in Fig. 5 compares the relative performance, area and soft error reliability of AES-128 and XTEA where: *TR* is the throughput; *HW* stands for hardware-saving; *EN* means energy-saving; *AR* represents the algorithm reliability; and *WU* is the worldwide utilisation. Collected values are normalised between scores of 1 and 5.

The XTEA performance results obtained show that the use of a lightweight encryption algorithm can provide key benefits over algorithms such as AES, when considering integration within an embedded design. When these implementations are designed with a focus on minimising area, and controlling for factors such as block size and duplex functionality, the use of XTEA over AES provides significant savings in terms of device area utilisation and power consumption. In addition, its data throughput is twice that of AES-128. On the other hand, AES is found in and required by many real applications, being the most widely used and strongest symmetric-key block cypher worldwide. Regarding reliability, XTEA presented a better MWTF than AES-128 by 7.78× for decryption and 2.72× for encryption. This demonstrates why it is growing in popularity in secure resource-constrained devices.

5. Conclusion and perspectives

This research presented the performance, and soft error assessment analysis of XTEA and AES cryptography implementations developed on an SRAM-based FPGA. Results show that the XTEA implementation gives the best relative performance, area, and soft error reliability trade-offs. The lower radiation sensitivity of XTEA can be explained by the relation between its performance and resources utilisation, i.e., reduced vulnerability window and number of sensitive bits w.r.t. AES implementations. Future works include the soft error assessment of other cryptography algorithms, considering the use of mitigation techniques that would provide the lowest overhead to each algorithm.

CRediT authorship contribution statement

Vitor Bandeira: Investigation, Validation, Data curation, Visualization, Methodology, Writing - Original Draft, Writing - Review & Editing.

Jack Sampford: Software, Investigation, Validation, Writing – Review and Editing.

Matheus G. Trindade: Investigation, Methodology, Writing - Original Draft.

Rafael Garibotti: Conceptualization, Methodology, Writing – Original Draft, Writing - Review & Editing, Funding acquisition.

Rodrigo P. Bastos: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Ricardo Reis: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Luciano Ost: Conceptualization, Methodology, Writing – Original Draft, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

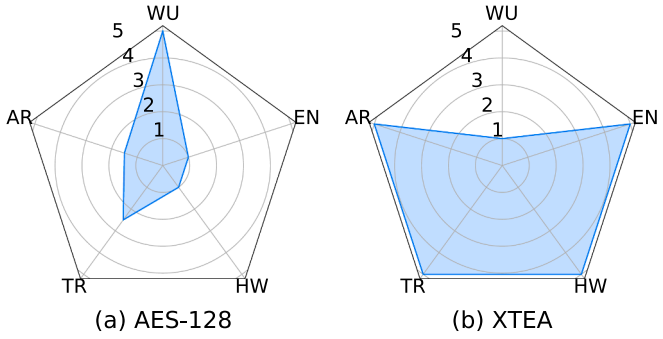

Fig. 5. Comparison summary of Algorithm Reliability (AR), Throughput (TR), Resource-saving (HW), Energy-saving (EN), and Worldwide utilisation (WU) between (*a*) AES-128 and (*b*) XTEA.

Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The radiation test cost was supported by MultiRad project funded by Région Auvergne-Rhône-Alpes's international ambition pack.

References

[1] M.A. Bahnasawi, et al., ASIC-oriented comparative review of hardware security algorithms for internet of things applications, ICM (2016) 285–288.
[2] J.W. Yu, et al., Benchmarking and optimizing AES for lightweight cryptography on ASICs, LCW (2019) 1–12.
[3] H. Zodpe, A. Sapkal, An efficient AES implementation using FPGA with enhanced security features, JKSUES 32 (2) (2020) 115–122.
[4] X. Zhang, Optimization and implementation of AES algorithm based on FPGA, in: IEEE ICCC, 2018, pp. 2704–2709.
[5] G. Bertoni, et al., Error analysis and detection procedures for a hardware implementation of the advanced encryption standard, IEEE TC 52 (4) (2003) 492–505.
[6] S. Jamuna, et al., Design and implementation of reliable encryption algorithms through soft error mitigation, IJCNIS 12 (2020) 41–50.
[7] R. Banu, T. Vladimirova, Fault-tolerant encryption for space applications, IEEE TAES 45 (1) (2009) 266–279.
[8] K. Wu, Low cost concurrent error detection for the advanced encryption standard, in: IEEE ITC, 2004, pp. 1242–1248.
[9] M. Agoyan, Single-bit DFA using multiple-byte laser fault injection, in: IEEE HST, 2010, pp. 113–119.
[10] J. Dutertre, Fault round modification analysis of the advanced encryption standard, in: IEEE HST, 2012, pp. 140–145.
[11] C. Roscian, Frontside laser fault injection on cryptosystems - application to the AES' last round, in: IEEE HOST, 2013, pp. 119–124.
[12] J.-P. Kaps, Chai-tea, cryptographic hardware implementations of xTEA, in: INDOCRYPT, 2008, pp. 363–375.
[13] F. Villa, Accelerator-based neutron irradiation of integrated circuits at GENEPI2 (France), in: IEEE REDW, 2014, pp. 1–5.
[14] ARM, Arm MPS2+ FPGA prototyping board. https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/mps2, 2021.
[15] M.G. Trindade, Assessment of machine learning algorithms for near-sensor computing under radiation soft errors, in: IEEE ICECS, 2020, pp. 1–4.
[16] G.A. Reis, et al., Software-controlled fault tolerance, ACM TACO (2005) 366–396.
[17] S.S. Mukherjee, A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor, in: IEEE/ACM MICRO, 2003, pp. 29–40.