Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MULTI-LEVEL AES DESIGN SECURITY: FROM SYSTEMC-TLM TO FPGA

[1]HASSEN MESTIRI, [1,2]YOUNES LAHBIB, [1]MOHSEN MACHHOUT, [1]RACHED TOURKI

[1]Electronics and Micro-Electronics Laboratory (E. µ. E. L), Faculty of Sciences of Monastir, University of Monastir, Tunisia

[2]Higher School of Technology and Computer Science (ESTI), University of Carthage, Tunisia

E-mail: hassen.mestiri@fsm.rnu.tn, younes.lahbib@esti.rnu.tn

ABSTRACT

Advanced Encryption Standard (AES) cryptographic system are widely used in embedded systems to secure secret information. One of the most powerful cryptanalysis techniques against the cryptographic systems is the fault injection attacks. The complexity of cryptographic systems is increasing which requires fast security attacks simulation against fault injection attacks. The multi-level Electronic System Level approach is one promising candidate that allows models to reach higher simulation speed. It is known that the SystemC Transaction Level Modeling (TLM) package simulates models 1000 times higher than classical Register Transfer Level (RTL) simulators.

In this paper, we present a secure reconfigurable AES design against the fault injection attacks at the Electronic System Level. Simulation results demonstrate that the simulation time is dependent of the fault detection schemes types. Moreover, The SystemC design is refined to RTL level. It is translated from SystemC description to a VHDL equivalent and implemented on Xilinx Virtex-5 FPGA. Experimental synthesis results show that the secure reconfigurable AES design is very robust against fault injection attacks and the fault detection scheme information redundancy allows a trade-off between the hardware overhead and the security against fault injection attacks.

Keywords: *Security, Hardware, SystemC Design, Fault Detection Schemes, Electronic System Level.*

1. INTRODUCTION

Currently, the complexity of cryptographic design is increasing than the design and verification capability of developers. To counter these problems, SystemC [1,2] is considered as the favorite hardware description language for modeling, hardware/software co-design and the verification at system level, and newly RTL synthesis. A lot of research has been done on AES [3-5] hardware implementation using VHDL. However, until now, only few works have been presented using the SystemC as an Electronic System Level Language candidate.

In [6], the authors present a fully automated runtime verification framework for the Assertion-Based Verification (ABV) of SystemC models. The Linear Temporal Logic (LTL) properties are checked during simulation by an automaton-based approach. They demonstrate the usefulness of their approach on a System on Chip (SoC) platform through a realistic AES encryption/decryption process.

In [7], the authors propose a generic SystemC implementation approach that allows the specification and validation of fully parameterisable transaction level properties. They demonstrate their approach on a realistic multi-level SoC platform and use the 128-bit key ten-round AES flavour as AES IP core.

In [8], the authors present a complete design flow of an AES/Data Encryption Standard (DES) cryptoprocessor, from the System Level Specification in SystemC to the final implementation on a prototype board. They use the cycle-accurate style, defined in the SystemC Verification Standard, along all the design flow

M. Askar et al. presented in [9] a SystemC implementation of a programmable cryptoprocessor for DES; Triple DES (TDES) and AES algorithms containing both encryption and decryption processes. They implement the algorithms, for all data and key lengths, in a single architecture instead of using separate architectures for each of the algorithm.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The analysis of previous works shows that the focus is AES implementations as an Electronic System Level model, but not AES model security. Without fault detection schemes [10-15] against fault-based attacks [16,17], the cryptographic systems can be broken. The fault injection attacks are a powerful technique which enables to break unprotected cryptographic schemes very efficiently. The idea is to inject one or several faults during the execution of an implementation and to use the faulty output to obtain information on the secret key stored in the secure component.

Simulations of AES models with fault attacks are time consuming, that's why Electronic System Level acceleration is mandatory in our case. We propose multi-level AES implementation with fault injection attacks, from SystemC-TLM [18] to RTL-VHDL. The TLM AES model is used to validate fault detection schemes. The RTL model, on the other hand, is the entry to FPGA synthesis. The same model is then emulated on FPGA with the presence of faults, and a secured reconfigurable AES design against fault injection attacks is taped-out. Three types of fault detection schemes are used to protect our design: hardware redundancy [10], temporal redundancy [11] and information redundancy [12].

The organization of this paper is as follows. Section 2 describes the background knowledge. Section 3 presents the SystemC AES design. The proposed secure reconfigurable AES design is presented in section 4. In section 5, the simulation and synthesis results are discussed and compared. Section 6 concludes the paper.

2. BACKGROUNDS

In this section the necessary background information and definitions to follow this paper is reviewed.

2.1 Advanced Encryption Standard

The AES is a symmetric block cipher that process data blocks using cipher keys with lengths of 128, 192 and 256 bits [3]. Each data block consists of 4×4 array of bytes called the state. The AES is a round-based encryption algorithm. The number of rounds is 10, 12, or 14, when the key length is 128, 192 or 256 bits, respectively. In the encryption of the AES algorithm, each round, except the final round, performs four transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey, while the final round does not have the MixColumns transformation. The key used in each round, called the round key, is generated from the initial key by a separate key scheduling module.

- SubBytes: a non-linear substitution step where each byte is replaced with another according to a lookup table.
- ShiftRows: a transposition step where each row of the state is shifted cyclically a certain number of steps.
- MixColumns: a mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation.
- AddRoundKey: each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

2.2 Fault Detection Schemes Against Fault Attacks

In the literature, a large number of fault detection schemes proposed against fault attacks are based on some sort of redundancy: hardware redundancy, temporal redundancy and information redundancy [10-15].

2.2.1 Hardware Redundancy

In hardware redundancy, two copies of the hardware are used concurrently to perform the same computation on the same data. After each computation, the results are compared and every difference is reported as a fault. The advantage of this technique is that it can detect both transient and permanent faults. However, it requires at least 100% hardware overhead.

2.2.2 Temporal Redundancy

In temporal redundancy, the same hardware is used to repeat the same process twice using the same input data. This technique uses minimum hardware overhead. Yet, it entails approximately 100% time overhead.

2.2.3 Information Redundancy

According to [19], the Information redundancy can be classified into three categories: parity-1, parity-16 and nonlinear robust codes.

The parity-1 uses only single bit parity for the entire 128-bit output and the fault is detected by comparing the predicted parity with the calculated parity at the end of each round.

In Parity-16, one parity bit is generated for each byte of the input and the fault is detected in the same way in Parity-1. While gaining higher fault coverage, the area overhead of Parity-16 is more than Parity-1.

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

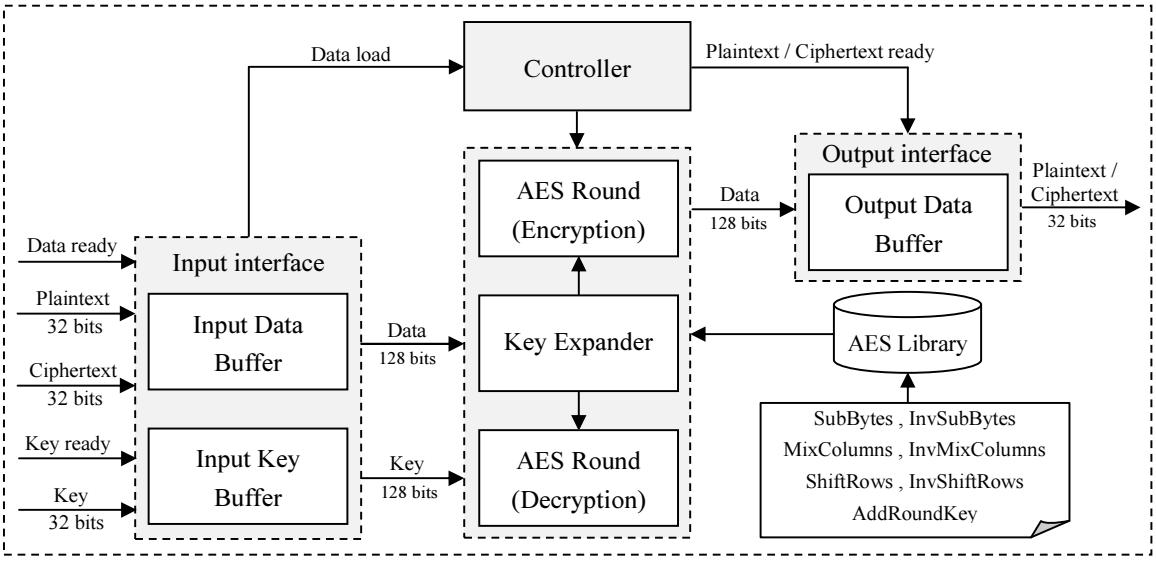

Figure 1: Block Diagram of AES 32-Bit

The nonlinear robust codes are based on the addition of two cubic networks. They allow to produce r-bit signatures to detect faults. This method offers good fault coverage. Yet, its hardware overhead is comparable to the hardware redundancy.

3. SYSTEMC ADVANCED ENCRYPTION STANDARD DESIGN

5.1 SystemC AES Design

Two techniques for implementing AES algorithm are presented in the literature: the pipeline and the iterative architecture.

To get a faster implementation in terms of throughput, all rounds of the algorithms can be implemented sequentially using pipeline structure. The pipeline architecture considers all AES rounds as separate components. The major disadvantage of implementing the pipeline architecture is requires large area and high power consumption [20]. In contrast, the iterative architecture requires an extremely small area and low power consumption. Since our implementation is designed to the embedded systems and especially the smart cards. Hence, we chose the iterative architecture.

Figure 1 shows the block diagram of our AES 32-bit. It takes four 32-bit for the input data and four 32-bit for the cipher key. Then it performs the encryption/decryption process and the output data it as four 32-bit.

The proposed AES encryption/decryption design considers the basic iterative architecture where a single round component is used. The same component is used for the 10 rounds in the AES algorithm. The iterative architecture of AES is composed of seven Modules:

- Input Data Buffer and Input Key Buffer are input blocks for the input data and the initial key. The input data and key are 32-bit length but the outputs are 128-bit wide, so Input Data Buffer and Input Key Buffer have to buffer the data and key during the loading process.
- Controller is a state controller that drives the synchronization between the other modules.
- AES Round Encryption performs the encryption of the input data.
- AES Round Decryption performs the decryption of the input data.
- Key Expander is used to generate the round keys from the cipher key.
- Output Data Buffer takes the outputs with 128-bit length and converts it into four 32-bit words.

5.2 AES Design Validation

The AES verification environment is presented in Figure 2. The stimulus generation module generates random inputs to the two different designs: the AES SystemC design and the AES TLM reference design. We performed the simulation by considering 4000000 random values of plaintext and cipher key for each input.

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

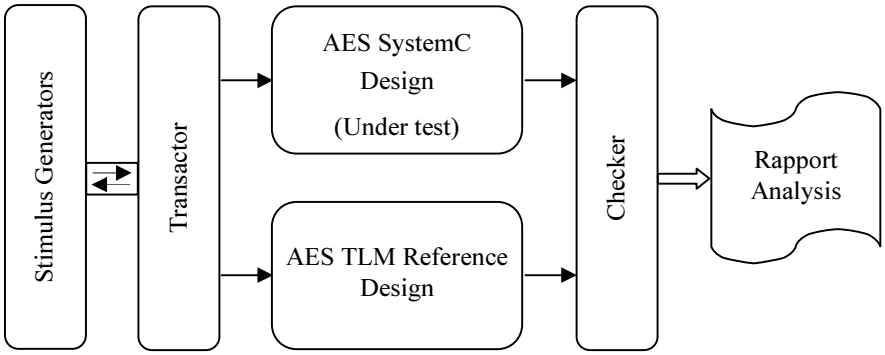

Figure 2: TLM Verification Environment

The outputs of both designs are passed to the checker that compares them. If a difference between them is found, an error is reported and the simulation ends. The verification was executed with several different seeds during long periods of time. By comparing the simulation results, we can see that every time the two outputs are equal. This implies that our AES SystemC Design works correctly.

4. THE PROPOSED SECURE RECONFIGURABLE AES DESIGN

This section is devoted to the description of the proposed secure reconfigurable AES design. In this design, we used SystemC as system level modeling language for hardware design. Using SystemC, we design a secure AES design against the fault injection attacks.

Our proposed SystemC design was designed to achieve the following objectives:

- Secure the transmitted data against the fault injection attacks
- Use the appropriate fault detection scheme according to the constraints required.

Figure 3 depicts a general view of our proposed secure reconfigurable AES design.

As seen in Figure 3, the proposed reconfigurable design of AES 32-bit consists of five major modules: Input interface, Output interface, Controller unit, SystemC AES, Fault Detection Schemes (FDS).

The Input and Output interfaces are used to connect the input and output data with controller unit and the secure SystemC AES design. The Controller unit is a state controller that drives the synchronization between the other modules.

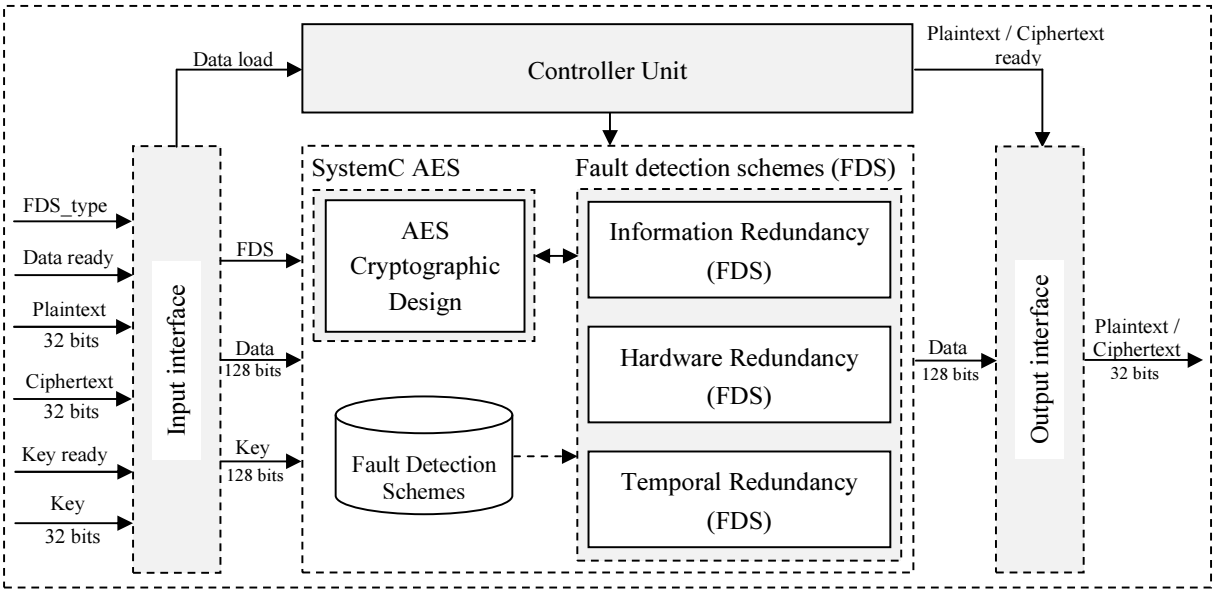

Figure 3: Secure Reconfigurable AES 32-Bit Design

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

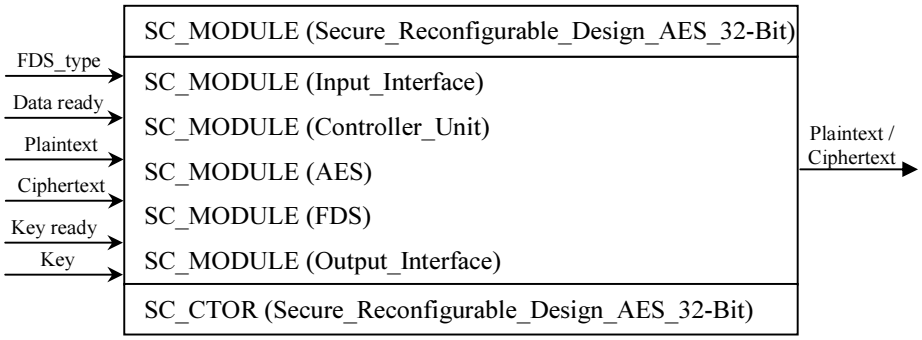

Figure 4: UML Class Diagram and Packages of the Secure Reconfigurable AES Design

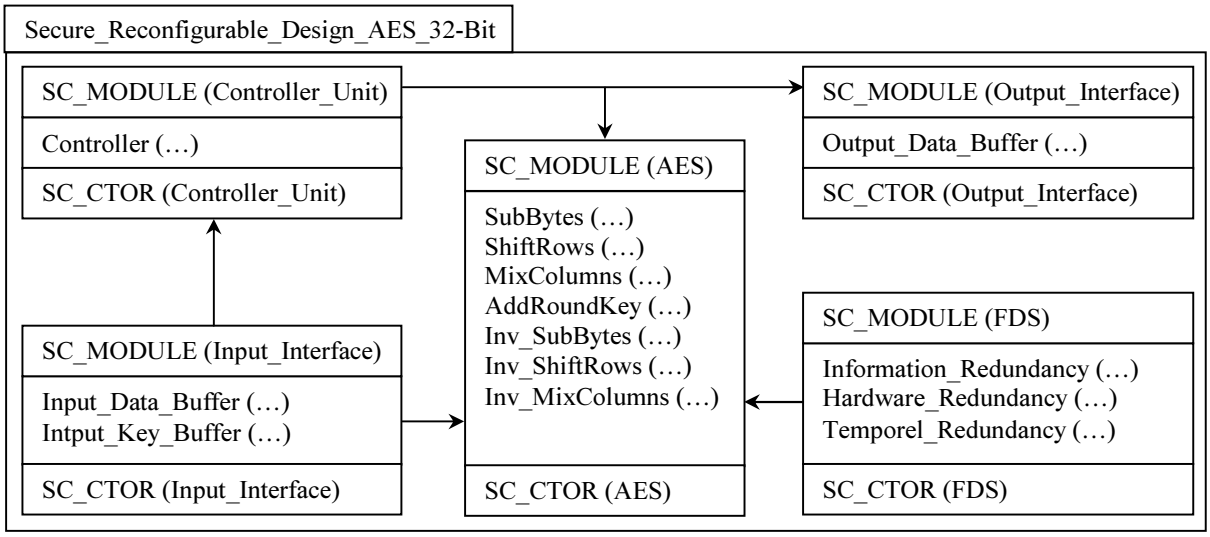

Figure 5: UML Package Diagram of the Secure Reconfigurable AES Design

The FDS module is the main module that interfaces with AES module to protect the transmitted data. Three types of fault detection schemes are used: hardware, temporal and information redundancy. The selection of appropriate schemes is performed via *FDS_type* signal.

The UML class diagram of the secure reconfigurable SystemC AES design and its package are illustrated in Figure 4 and Figure 5 respectively.

In the UML class diagram and its package, the main module *Secure_Reconfigurable_Design_AES_32-bit* is composed of five secondary modules. The *FDS* module, which based on three types of the fault detection schemes, is implemented to verify the correct operation of the functional designs even under the presence of fault injection attacks.

5. SIMULATION AND SYNTHESIS RESULTS

In this section, we describe the simulation and implementation results which were carried out to evaluate the performances of the fault detection schemes for the encryption module.

First, we analyze the impact of the fault detection schemes on the simulation time. Next, the reconfigurable SystemC AES design is refined to RTL and the FPGA performances of the fault detection schemes discussed and compared.

5.1 Simulation Time

To analyze the impact of the fault detection schemes on simulation time, we injected 1000 random faults into the secure reconfigurable AES design and we measured the mean simulation time for the encryption process.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Simulation Time Analysis of the SystemC AES Design Using Fault Detection Schemes

SystemC AES Design		kTime (s)	uTime (s)	rTime (s)
Original AES Design		0.042	1.596	1.644
AES Design protected by FDS	Hardware Redundancy [10]	0.047 *(11.90%)*	1.649 *(3.32%)*	1.721 *(4.68%)*
	Temporal Redundancy [11]	0.074 *(76.19%)*	3.152 *(97.49%)*	3.246 *(97.44%)*
	Information Redundancy [12]	0.051 *(21.43%)*	2.304 *(44.36%)*	2.372 *(44.28%)*

Before presenting and discussing the results, it is important to note that the error margin of the measurement tool we employed (Linux command time) is not significant and does not affect the results.

Table 1 present a comparison between the original and the secure AES design in terms of simulation time (kernel time (kTime), user time (uTime) and real time (rTime)).

The hardware-based scheme proposed in [10] slightly affect the simulation time of the original design, it requires only 3.32% simulation user time overhead. Contrast to the hardware-based scheme, the temporal-based scheme proposed in [11] causes a 97.49% simulation time overhead. The information redundancy scheme [12] requires more simulation time overhead than the hardware-based scheme presented in [10].

Our conclusion is that the simulation time is dependent of the fault detection schemes types. Yet, many parameters (frequency, area, throughput, …) should be evaluated to determine the impact of the fault detection schemes on the AES performances. To this end, the TLM reconfigurable AES design is refined to RTL level. It is translated from SystemC description to a VHDL equivalent and then synthesized into FPGA.

5.2 FPGA Implementation

In this section we present the simulation results which were carried out to evaluate the performances of fault detection schemes in the RTL level. Each fault detection scheme has been applied to the AES encryption/decryption. The original and protected AES designs have been described using VHDL, simulated by ModelSim 6.6 and synthesized with Xilinx ISE 10.1.03. The FPGA target is XC5VLX50T from Xilinx Virtex-5 family.

As seen in Table 2, the fault coverage (FC), the number of occupied slices, the frequency (in megahertz), the throughput (in megabits per second), the area overhead and the frequency degradation for the original and protected AES designs, in encryption/decryption process, are presented.

The fault coverage presents the percentage of fault detection capabilities against the faults injection attacks. The original AES encryption/decryption occupies 1200 slices for 206.85 MHz Frequency. Adding fault detection scheme to the AES design implies increasing the amount of resources, of computations, or both. As shown in Table 2, the fault detection scheme [10], based on hardware redundancy, reaches 100% fault coverage. It provides high level of security to secure the AES against fault injection attacks.

Table 2: FPGA Implementation of the AES Design Using Fault Detection Schemes

AES Design		Encryption/Decryption				
		FC (%)	Area (Slice) (overhead)	Freq.(MHz) (degradation)	Throu. Encry (Mbps)	Throu. Decry (Mbps)
Original AES Design		-	1200	206.85	2206.45	1103.27
AES Design protected by FDS	Hardware Redundancy [10]	100%	2076 *(73%)*	206.60 *(0.12%)*	2203.73	1101.87
	Temporal Redundancy [11]	100%	1357 *(13.08%)*	187.86 *(9.18%)*	1001.92	500.96
	Information Redundancy [12]	99.998%	1400 *(16.67%)*	193.22 *(6.59%)*	2061.01	1030.50

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The hardware-based scheme proposed in [10] slightly affects the throughput of the original design, but it has the highest cost in terms of area. It requires approximately 73% area overhead. The temporal-based scheme proposed in [11] gives 100% fault coverage and slightly affect the frequency. However, it requires twice more clock cycles and causes 54.6% throughput degradation. As shown in Table 2, the information redundancy scheme decreases the throughput of the original design, but they require less area overhead than the hardware-based scheme presented in [10]. Therefore the information-based redundancy allows a trade-off between the performances and the security of the AES design.

6. CONCLUSION

In this paper, we present a secured reconfigurable AES design against the fault injection attacks at the Electronic System Level. After an overview of fault detection schemes against fault injection attacks, we proposed a SystemC AES design and secured reconfigurable AES design. Simulation results show that the simulation time is dependent of the fault detection schemes types. Moreover, The SystemC design is refined to RTL level. It is translated from SystemC description to a VHDL equivalent and implemented on Xilinx Virtex-5 FPGA. FPGA implementation results show that the information-based redundancy allows a trade-off between the security and the performances of the AES design.

REFRENCES:

[1] IEEE Standard SystemC Language Reference Manual, IEEE Standard 1666, 2005.

[2] Open SystemC Initiative (OSCI), homepage, http://www.systemc.org/.

[3] National Institute of Standards and Technology (NIST), Advanced Encryption Standard(AES), *Federal Information Processing Standards Publication 197 (FIPS 197)*, November 2001.

[4] H. Mestiri, M. Machhout, R. Tourki, "Performances of the AES design in 0.18µm CMOS technology", *IEEE, 7th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2012.

[5] S. Arrag, A. Hamdoun, A. Tragha, S.E. Khamlich, "Implementation of Stronger AES by Using Dynamic S-Box Dependent of Master Key", *Journal of Theoretical & Applied Information Technology*, Vol. 53, No. 2, 2013, pp. 196-204.

[6] M. Kallel, Y. Lahbib, A. Baganne, R. Tourki, "Runtime verification of transaction level SystemC models using an aspect-based approach", *International Journal of Computer Sciences and Engineering Systems*, Vol. 5, No. 1, 2011, pp. 45-55.

[7] M. Kallel, Y. Lahbib, A. Baganne, R. Tourki, "Monitoring transaction level SystemC models using ageneric and aspect-oriented framework", *International Journal of Computer Aided Engineering and Technology*, Vol. 4, No. 3, 2012, pp. 229-249.

[8] J. Castillo, P. Huerta, J. I. Martinez, "SystemC Design Flow for a DES/AES CryptoProcessor", *In WSEAS Transactions on Information Science and Applications*, Athens, 2004, pp. 193-198.

[9] M. Askar, T. Egemen, "Design and SystemC Implementation of a Crypto Processor for AES and DES Algorithms", *In: Information Security & Cryptology Conference with International Participation*, 2007, pp. 145–149.

[10] M. Joye, P. Manet, and J.B. Rigaud, "Strengthening Hardware AES Implementations Against Fault Attacks", *IET Information Security*, 2007, pp. 106-110.

[11] T.G. Malkin, F.-X. Standaert, and M. Yung, "A Comparative Cost/Security Analysis of Fault Attack Countermeasures", *In L. Breveglieri et al. (Eds.), Fault Diagnosis and Tolerance in Cryptography, Springer-Verlag Berlin Heidelberg*, LNCS. 4236, 2006, pp. 159-172

[12] H. Mestiri, N. Benhadjyoussef, M. Machhout, R. Tourki, "High performance and Reliable Fault Detection Scheme for the Advanced Encryption Standard", *International Review on Computers and Software (IRECOS)*, Vol. 8, No. 3, 2013, pp. 730-748.

[13] H. Mestiri, N. Benhadjyoussef, M. Machhout, R. Tourki, "A Robust fault detection scheme for the advanced encryption standard", *International Journal of Computer Network and Information Security (IJCNIS)*, Vol. 5, No. 6, 2013, pp. 49-55.

[14] J. Chu, M. Benaissa, "Error Detecting AES Using Polynomial Residue Number Systems", *Microprocessors and Microsystems,* Vol. 37, No. 2, 2013, pp. 228-234.

[15] H. Mestiri, N. Benhadjyoussef, M. Machhout, R. Tourki, "An FPGA implementation of the AES with fault detection countermeasure", *In: International Conference on Control, Decision and Information Technologies,* 2013, pp. 264-270.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16] S. S. Ali, D. Mukhopadhyay, "Differential fault analysis of AES-128 key schedule using a single multi-byte fault", *in: Smart Card Research and Advanced Applications, Springer,* LNCS. 7079, 2011, pp. 50-64.

[17] C. H. Kim, "Differential fault analysis against AES-192 and AES-256 with minimal faults", *in: Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2010, pp. 3-9.

[18] Transaction Level Modeling Library, Available: www.accellera.org.

[19] X. Guo, D. Mukhopadhyay, and R. Karri, "Provably Secure Concurrent Error Detection Against Differential Fault Analysis", *IACR Cryptology ePrint Archive,* Available from:eprint.iacr.org/2012/552.pdf, 2012.

[20] T. Subashri, R. Arunachalam , B. Gokul-Vinoth-Kumar, V. Vaidehi,: "Pipelining Architecture of AES Encryption and Key Generation with Search Based Memory", *International journal of VLSI design & Communication Systems*, Vol. 1, No. 4, 2010, pp. 48-60.