

ENHANCED LBIST AES PROCESS FOR SECURED COMMUNICATIONS

1. N.MADHAVA RAO, 2. P.GURUBRAHMAM

1. PG Scholar, Dept of ECE, ABR College of Engineering and Technology, Kanigiri, Prakasam District

2. Assistant Professor, Dept of ECE, ABR College of Engineering and Technology, Kanigiri, Prakasam District

ABSTRACT:

The main motive of this project is to design a crypto device with low complexity and high security by using "ADVANCED AES" Algorithm along with BIST technique. The selective application of technological and related procedural safeguards is an important responsibility of every Federal organization in providing adequate security to its electronic data systems and coming to BIST concept there are two main functions that must be performed on-chip in order to implement built-in self-test (BIST): test pattern generation and output response analysis. The most common BIST schemes are based on pseudorandom test pattern generation using linear feedback shift registers (LFSR'S) and output response compaction using signature analyzers. To accomplish high security for a system we are using the crypto devices technique in our project. Further Scan chain reordering algorithm is implemented for crypto operations. Scan chain reordering provides, less power consumptions while doing operations.

KEYWORDS: CRYPTOGRAPHY, BIST, LFSR, AES, TPG, SA

INTRODUCTION:

Most of the user now a day's using wireless communication for fast sending and receiving the mails in less time and in less cost. When this way of communication is going on, the unauthorized people who have the intension to know about our conversation will hack the information within that frequency. After hacking the information the hacker can know about what we are discussing. This leads to leakage of information. Nowadays, secure circuits are commonly used for applications such as e-banking, pay tv, cell phone... Because they hold personal data and must process secure operations, security requirements such as source/sink

authentication, data integrity, confidentiality, or tamper resistance are maintained by means of several dedicated components. Confidentiality is ensured through cryptographic mechanisms generally implemented on co-processors. These mechanisms encode/decode plaintexts/cipher texts with the help of secret keys that must be preserved from compromise. Testing a secure circuit requires a specific attention since any undetected malfunction may induce a vulnerability and any extra test mechanism may induce new security vulnerabilities. For

instance, generation of deterministic test patterns and design for testability such as scan design provide very high fault coverage. This mechanism minimizes the probability to deliver a supposedly secure system, but actually faulty chip, which could fail to protect the secret data. However, the scan path itself may compromise the security of the system since it provides facilities for controlling or observing sensitive data (scan based attacks have been demonstrated in [1] and [2]). Specific secure scan design methodologies such as the ones detailed in [3] and [4] can prevent abusive usage of the scan path but requires extra area and design effort. Conversely, the built-in self test (BIST) approach does not require visible scan chains. When the test mode is started, scan chains are fed from on-chip test resources and scanned-out test responses are compacted into a signature. The only test output is this compacted signature or the comparison result of this signature with a pre-computed "gold" one. The BIST strategy is considered as a good alternative if it provides acceptable fault coverage and low area overhead (apart from its recurrent cost, extra area for BIST implementation may in turn be subject to faults and, consequently, must be kept as low as possible). Re-using a cryptographic core

("crypto core") as test pattern generator (TPG) or signature analyser (SA) for other cores in the system prevents the insertion of any other dedicated hardware. However efficiency in terms of pattern generation and response compaction must be evaluated. There are two main functions that must be performed on-chip in order to implement built-in self-test (BIST): test pattern generation and output response analysis. The most common BIST schemes are based on pseudorandom test pattern generation using linear feedback shift registers (LFSR) and output response compaction using signature analyzers.

The AES ciphers a block of 256 bits plaintext into a 256 bits cipher text with the help of a 256, 192 or 256- bits secret key K . The 256-bits plaintext is organized into a 4×4 matrix of 16 bytes. After a first XOR operation between K and the plaintext, the algorithm consists in several rounds: 10, 12 or 14 rounds according to the key length 128, 192 or 256 bits. Every round except the last one is composed of four operations: Subbytes is a substitution of text bytes with the help of substitution tables called Sboxes, ShiftRows consists in circular shifts on the matrix lines, MixColumns is a multiplication by a known matrix in the Galois field, and AddRoundKey is a XOR operation between the partially ciphered text and the round key RK_i ; RK_i being derived from the initial secret key K . The last round does not execute the MixColumns operation. Without loss of generality, we assume hereafter 256-bits key and thus 10 rounds. Figure 1 presents the base iterative implementation of the AES algorithm. It is mainly composed of a Key Generation module and a Round module. After 10 iterations of the Round module, the controller set the Encryption signal that loads the cipher text into the output register $R2$.

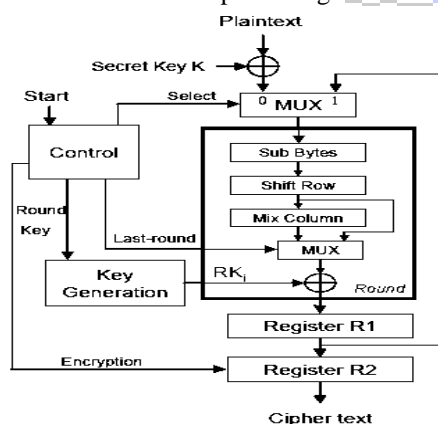


Figure 1: AES base iterative implementation

In this paper we investigate 1/ the self-testability of the AES crypto core, and 2/ its use as TPG or SA.

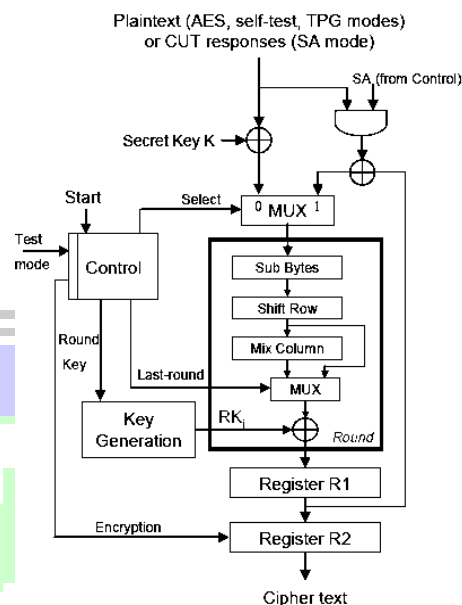


Figure 2: AES TPG/SA implementation

The three new behavioural modes (TPG/SA/SELF_TEST) entail the addition of extra control, and new operations in the data path (AND, XOR) of the base implementation. Figure 2 depicts the introduced slight changes. During the first round of the mission mode (encryption), self-test or TPG modes, the select signal is set to 0. The self-test of the core is further discussed in section 3. In TPG mode, the select signal allows to load the seed of the generator. Next, the select signal is set to 1 while the SA is set to 0. Test patterns are issued from $R2$ at every clock cycle with the help of the Encryption signal that enables the $R2$ load operation after every round. Evaluation of the so-generated test vectors is presented in the section 4. For test response compaction (SA mode), select and SA signals are set to 1. An XOR operation is performed between one response of the Core Under test (CUT) and the result of the previous round. The final signature obtain after compaction of all the test responses is loaded into the $R2$ register. Diagnostic facilities can be implemented using the Encryption signal for enabling the analysis of intermediate signatures. The Key Generation module is also slightly modified in such a way that during self-test, TPG and SA modes, the 10th round key is used as the primary key for the next

PROPOSED SYSTEM (AAES)	EXISTING SYSTEM(AES)
1. RCON is fixed.	1. RCON is not fixed.
2. MIXED MULTIPLICATION is also fixed.	2. MIXED MULTIPLICATION is also not fixed.
3. It can handle Data up to: 64 bits & SKey up to: 56 bit.	3. It can handle Data up to: 256,192,256 bits & Key up to: 256,192,256.
4. Not much secure, since all blocks are not dependent.	4. More secure, since all blocks are dependent.
5. KEY is same for all blocks.	5. KEY is not same for all blocks.
6. KEY,USER are dependent	6. KEY,USER are independent
7. It Feistel network	7. Substitution permutation network

round keys generation. Usually, the original secret K is used as primary key at the beginning of every encryption, or, in other words, every ten rounds. This behaviour is maintained for mission mode. Table 1 reports the areas of original and modified AES crypto cores in terms of cell number. Both architectures have been described in VHDL

AES SELF-TEST

This paragraph presents both theoretical results on the required test length for AES self-test and fault simulation results. As discussed in [7], pseudo-random testing is an efficient technique for crypto cores. High fault coverage can be achieved with short pseudo-random test sequences because traditional cryptographic operations (XOR, substitution, modulo...) are easily tested with random data. Moreover, the

inherent properties of these operations allow the propagation of random data through the circuit. Because the AES core is mainly made up of Sboxes (83% of the AES area for implementing the SubByte operation), we first focused on the testability of these components. The minimum deterministic test set for 100% fault coverage is 203 patterns long for one 8-input bits Sbox. An in-house fault simulator and a heuristic have been used for building up the fault dictionary and defining this minimal test set. From this number, and in order to be conservative, we compute the minimal-length random sequence that would include these 203 patterns with a given confidence [8]: where $P[X \leq n]$ is the confidence level, k is the number of targeted patterns, p is the probability that every random pattern occurs (here $p=1/28$) and n is the number of random patterns that have to be sampled. In our case, 203 patterns must be obtained after n random patterns with a confidence level of 99%, i.e.: From this equation it comes that the minimal random sequence length is $n = 2534$ patterns. This result stands for the whole set of Sboxes in the AES core since they are tested in parallel. The same experiment have been performed for various implementations of the Sboxes and thus for different minimal deterministic test sets. In any case, the theoretical minimal length of the random sequence for including the targeted deterministic patterns ranges from 2400 to 2600 patterns. The upper bound is thus set to 2600 random patterns to test the Sboxes whatever their implementation. Concerning the other round operations: ShiftRow function

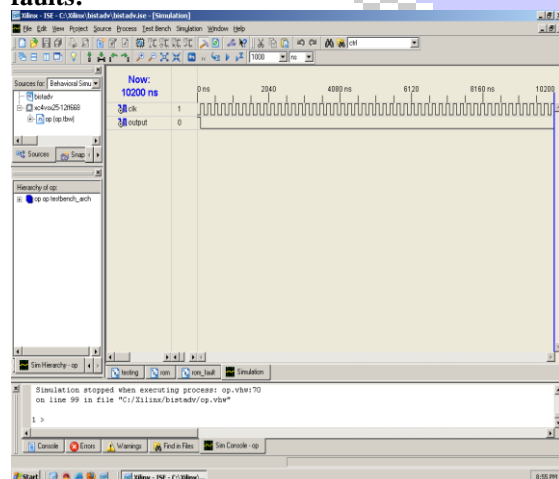
requires only wires for its implementation and is tested when every bit of this interconnection structure has been set to both "0" and "1" (under the assumption of stuck-at fault model). This should be easily achieved with the patterns issued from the Sboxes (bijective operations fed with 2600 random patterns). MixColumn and AddRoundKey operations are mainly xor trees and should be very easily tested too using random patterns issued from the Sboxes. In order to confirm this hypothesis, we have performed a fault simulation on the proposed AES core sets in self-test mode. The test response (or signature) is only observed after simulation of the whole sequence, not at every round. The self-test structure is initialised with a randomly chosen plaintext and a secret key. This experiment has shown that all the faults have been tested after 2100

round cycles (to be compared with the 2534 random patterns theoretically required for 100% fault coverage on the first experimented Sboxes implementations). This experiment has been repeated with different plaintexts and secret keys as starting points: 2100 to 2500

Test pattern generation

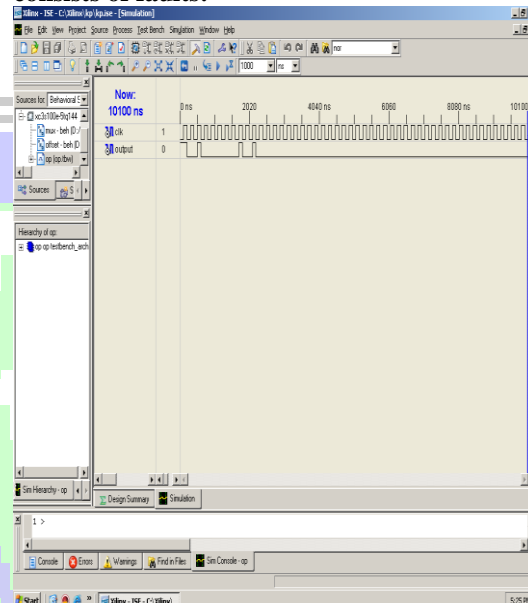
This section questions the randomness of the patterns issued from the proposed AES-based TPG, called "1-AES-round" TPG in the following. Inherent property of the basic operations involved during AES encryption and statistical analysis of the data issued from the base crypto algorithm demonstrate that the whole AES encryption process is a very good 256-bits random number generator [9]. It can be used for instance as random number generator for stream ciphering operations implemented in the same system. The randomness properties of the 256 generated 1-bit streams and their usage as test sequences have been studied in [10]. However in this study, the whole AES mission mode was used to provide a new random word at every encryption cycle (i.e. every 10 rounds). The main drawback of this approach is that a test pattern is generated every ten clock cycles. Here, in the proposed TPG mode implementation, one pattern is issued at every round, or in other words at every clock cycle (all 2256 This constraint allows a fair comparison with the usual LFSR-based TPGs that generate a new pseudo-random data at every clock cycle.

SIMULATION RESULTS: Cryptography which has no of faults:



patterns have been required for 100% fault coverage. We did the same experiments with two other AES logic implementations and obtained similar results. From a practical point of view, 2600 round cycles in self-test mode should be sufficient to test the whole structure with a confidence level of 9

Simulation report for a cryptography which consists of faults:



Conclusion

In the context of secure circuits, BIST approaches appear as good alternatives since they do not rely on visible scan chains. However they require extra hardware extra hardware for implementing test pattern generation, signature analysis and corresponding control logic. In this project, a solution is presented that consists using an AES-based cryptographic core commonly embedded in secure systems. Three additional modes are added to the current mission of the AES crypto core, one for self-test, one for pseudo-random test pattern generation and one for signature analysis. Efficiency of these three modes has been demonstrated with scan chain reordering algorithm. Extra cost in terms of area is very low even compared to the implementation of a BILBO register. Because only one AES core may be originally embedded in the system, it will be interesting to study concurrent test pattern generation and response compression. Furthermore, since secure systems requires very high quality

testing strategies, it may be necessary to apply deterministic patterns to some systems cores due to their resistance to pseudo-random test sequences. Techniques such as TPG reseeding should be investigated in this case.

REFERENCES:

- [1] C.H. Gebotys, R.J. Gebotys. A Framework for Security on NoC Technologies. Proc of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'03),2003.
- [2] K. Okeya, T. Takagi. A More Flexible Countermeasure against Side Channel Attacks Using Window Method. CHES'2003,LNCS 2779,pp.397-410,2003.
- [3] J.S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. CHES'99,LNCS 1717, pp.292-302,1999. 27
- [4] M. Joye,C. Tymen. Protections against differential analysis for elliptic curve cryptography: An algebraic approach. CHES'2001,LNCS 2162,pp.377-390, 2001.
- [5] P.Y. Liardet, N.P Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. CHES 2001, LNCS 2162,pp.391-401,2001.
- [6] E. Brier, M. Joye. Weierstrass Elliptic Curves and Side-Channel Attacks. PKC 2002, LNCS 2274,pp.335-345,2002.
- [7] M. Joye, J.J. Quisquater. Hessian elliptic curves and side-channel attacks. CHES 2001, LNCS 2162, pp.402-410,2001.
- [8] Okeya, K., Sakurai, K., Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack. INDOCRYPT 2000, LNCS1977,pp.178-190,2000.

Bibliography

.

