World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 221-227, 2014
ISSN 1818-4952
© IDOSI Publications, 2014
DOI: 10.5829/idosi.wasj.2014.29.csea.2271

Crypto-Devices Algorithms Test Techniques and Fault Detection

R. Udayakumar, V. Khanaa and K.P. Thooyamani

School of Computing Science, Bharath University - 73, India

Abstract: This paper describes a generic built-in self-test strategy for devices implementing symmetric encryption algorithms. Taking advantage of the inner iterative structures of crypto-cores, test facilities are easily set-up for circular self-test of the crypto-cores, built-in pseudorandom test generation and response analysis for other cores in the host device. Main advantages of the proposed test implementation are an architecture with no visible scan chain, 100% fault coverage on crypto-cores with negligible area Overhead, availability of pseudorandom test sources and very low aliasing response compaction for other cores.

Key words: Security · Digital circuit testing · Self-testing

INTRODUCTION

At the core of an electronic device offering digital security services is the cryptographic coprocessor that executes the cryptographic function. Such “crypto-cores” provide security services such as confidentiality, integrity and authentication.

Because weak “crypto-algorithms,” poor design of the device or hardware physical failures can render the product insecure and place highly sensitive information or infrastructure at risk, validation of crypto-algorithm and test of the hardware implementing such functions are essential.

U.S. National Institute for Standards and Technology (NIST) organized contest for selecting encryption functions. The Data Encryption Standard (DES) [1] was adopted as national standard in 1976 and the Advanced Encryption Standard (AES) [2] has been selected in October 2000. Since the hardware implementation of DES is not expensive, it is still currently used in many applications, frequently in the form of Triple DES for security improvement [3]. The validation of such algorithms as cryptographic functions has been widely discussed as part of the standardization process and is not further discussed here. This paper aims at providing efficient test solutions for possible physical failures on the electronic device implementing the cryptographic algorithm.

One approach for providing test solutions at different stages of an IC life cycle consists in including built-in self-test (BIST) resources into the circuit under test (CUT). Classically, storage elements are organized into scan chains and additional hardware is used for feeding the scan chains with pseudorandom test data and sinking the test responses before analysis of the compressed signature [4]. So, BIST does *not* provide full controllability and observability of the internal storage elements from the IC interface, namely the scan-in/scan-out pins. This major difference with the external testing strategy makes the scan based attacks not exploitable [5], [6] and thus avoiding the implementation of related countermeasures (e.g.,[6]–[8]). However, BIST must be implemented at low cost and its efficiency must be demonstrated in terms of fault coverage (FC) and test length.

Random pattern testability of crypto-cores has been discussed in [9]. Authors show how random data and possible errors can be easily propagated through typical operations involved in symmetric block encryption algorithms. The paper focus on data paths of nonstandard algorithms (e.g., 3-WAY cipher [10]) and the test solution lies on a classical centralized BIST architecture where extra test resources are inserted in the design for test pattern generation (TPG) and output response analysis (ORA) functions.

Authors in [11] proposed a self-test procedure for a 128-bit key AES core. The inner cyclic behaviour is exploited to test the hardware of the round while the key generation module is tested using patterns from the round output. Faults on the control part are not considered.

Corresponding Author: R. Udayakumar, School of Computing Science, Bharath University - 73, India.

Authors claim 0.76% of area overhead for self-test mode implementation and a test length of 12 ciphering cycles. In this paper, we propose a BIST solution for crypto-devices implementing standard symmetric block cipher algorithms DES and AES. This is an extension of the work presented in [12] that focuses on a specific AES architecture only. Efficient circular self-test schemes as well as TPG and ORA functions are easily implemented on AES and DES crypto-cores due to the iterative process involved in their algorithms. Efficiency of test modes (SELF_TEST, TPG, ORA) in terms of randomness, aliasing and cost of implementation are discussed and supported by experimental results.

The paper is organized as follows. Section II presents the proposed modification to the original secure device that allows the implementation of the Self Test, the TPG and the ORA modes. Section III analyzes testability and randomness properties of block cipher algorithms, while Sections IV and V present experimental results about Self Test and TPG, respectively. Section VI discusses the properties of the device when used as ORA. Eventually, Section VII concludes the paper.

Implementation of the Test Modes: Fig. 1 (except shaded areas) presents a typical implementation of the AES or DES crypto-algorithm. It is mainly composed of a *Key Generation* module and a *Round* module. In mission mode, after an initial operation (XOR between Key and Plaintext for AES and permutation of the plaintext for DES), the plaintext block is looped around the *Round* module several times (10 for AES, 16 for DES) before the final cipher is loaded into the R-out register, possibly after a final operation like the final permutation in DES. The widths of the data are 128 for AES and 64 for DES.

The grayed areas in Fig. 1 depict the modifications to support the three new modes: SELF_TEST, TPG, ORA.

In SELF_TEST, TPG and mission modes, the SA signal is set to 0, so that the gray XOR operation is transparent. During the first round, the Select signal is set to 0 to load either the plaintext or the seed for the generator. This signal is set to 1 during next rounds.

In ORA mode, the input of the circuit is the output of the CUT. SA and Select are set to 1, so that an XOR operation is performed between one response of the CUT and the result of the previous round in the crypto-core. The final signature obtained after compaction of all the test responses is loaded into R-out. In TPG mode, R-out is loaded every clock cycle.

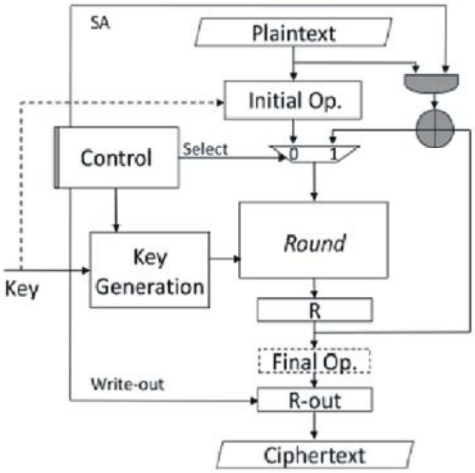


Fig. 1:

During the SELF_TEST mode, an initial message M1 is encrypted into M2=Round(M1) and the process is repeated *n* times (Mi+1=Round(Mi), i □{1...,n}). Finally, the output data Mn+1 is stored into R-out for comparison with the expected golden value. Diagnostic features can be implemented in SELF_TEST and ORA modes using the write-out signal to enable the analysis of intermediate signatures.

Concerning round key generation, either the keys are precomputed and stored in the circuit or the key generation module calculates the sequence of keys. For the latter case, AES is modified in such away that during SELF-TEST, TPG and SA modes, the tenth round key is used as the primary key for the next round key generation. In this way, during self-test, the key generation module receives as many different stimuli as rounds. For DES, this is not necessary because the key generation module does not contain any logic. The round keys are simply formed of subsets of bits of the initial key.

For security reasons, we assume a shadow register for storing the key used during the test mode. Test keys are assumed to be equals for all the circuits since fault simulation is performed only once. Afterwards, crypto-cores can be differentiated using different secret keys for their mission modes.

TPG, ORA and SELF_TEST modes have been implemented in DES and AES circuits. Circuits have been described in VHDL and synthesized with Synopsys Design Compiler.

- Using a 350-nm CMOS library provided by AMS.

- The area overhead is about 6% (271 cells) for the DES and 3% (507 cells) for AES. For comparison, 128-bits register built-in logic block observer (BILBO) [13], which supports TPG and ORA functions, requires 859 cells. Our solution is a good alternative for TPG when the secure device is already present into the design.

Testability and Randomness: Security provided by block cipher algorithms such as DES and AES relies on two main properties named *Diffusion* and *confusion* [14], [15]. *Confusion* refers to making the relationship between the key and the cipher text as complex and involved as possible. *Diffusion* refers to the property that redundancy in the statistics of the plaintext is dissipated in the statistics of the cipher text. For diffusion to occur, a change in a single bit of the plaintext should result in changing the value of many ciphertext bits. These properties are supported by the Feistel network [16] for the DES and by the substitution–permutation network for the AES. AES and DES also have two common characteristics. First, they are iterative algorithms. DES is composed of 16 rounds while AES is made of 10 rounds. All rounds are (quasi) identical, i.e., the result of a round is used as the input of the next round. Second, since encryption/decryption are bijective operations for a given key, each round is a bijective operation too (on a set of 2^{64} elements for DES on a set of 2^{128} elements for AES).

The diffusion property is a very interesting feature with regard to the test of their hardware implementation. It implies that every input bit of a round influences many output bits, i.e., every input line of a round is in the logic cone of many output bits. In other words, an error caused by a fault in the body of the round is very likely to propagate to the output. Thus, the circuit is very observable. Moreover, since rounds are bijective, the input logic cone of every output contains many inputs. In other words, each fault is highly controllable. Therefore, these circuits are highly testable by nature whatever the implementations. This point motivates the implementation of a SELF_TEST mode.

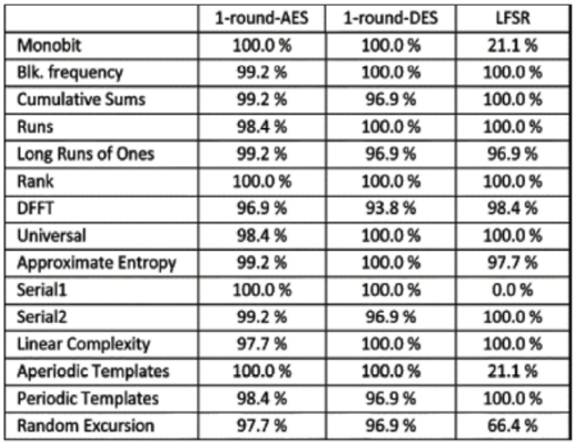
By nature, diffusion and confusion decorrelate the output values from the input ones. It has been demonstrated that by feeding back the output to the input, the generated output sequence has random properties [17], [18]. This property can be used for building a pseudorandom test source within a secure device implementing a DES or an AES core. Note that the test sequence generated from a crypto-core output can be composed of either results of successive encryptions, or results of successive rounds since these algorithms are iterative. In the first case, one pattern is generated after each encryption, i.e., 10 cycles for AES and 16 cycles for DES. The randomness of the first solution has been investigated in [12] for AES. In the second case, a pattern is generated after every round, i.e., after every clock cycle for parallel implementations. We will refer to *1-round-AES* and *1-round-DES* in the sequel of the paper to denote this scheme. For test purposes, this solution is highly preferable because it reduces the overall test time. However, the randomness of the sequence obtained according to this scenario must be investigated.

Several empirical metrics can be used for evaluating the randomness of a sequence. We chose the NIST test battery [19] that is composed of 15 statistical tests. For instance, the first test, called *Frequency Test*, determines whether the numbers of ones and zeros in the bit stream are balanced. We first compared 1.5 Mb streams issued from the rightmost bit of, respectively, 1-round-AES, 1-round-DES and an LFSR with internal XORs (primitive polynomial $p(x)=x^{128}+x^{29}+x^{27}+x^{2}+1$). These bit streams can be used for instance to feed a single scan chain in a CUT. LFSR seed, initial plaintexts and key for 1-round-AES and 1-round-DES have been randomly chosen. Bit streams issued from 1-round-AES and 1-round-DES pass all tests while the LFSR passes only 11 tests out of the 15 (the poor result for the LFSR is explained by the fact that the length of the bit stream is far shorter than the whole cycle length, i.e., 1.5 Mb versus 2^{128}-1 bits).

For word-based pattern generation (e.g., multiple scan chains architectures), we first checked the randomness of the streams issued from each output bit of the 1-round-AES, 1-round-DES and LFSR output, considering each bit independently. Table I shows, for each NIST test, the percentage of bit streams that pass the test.

Concerning the 1-round-DES, since the 32 right bits of a DES round are copied to the 32 left bits before execution of the next round, the right bit streams and the corresponding left bit streams are identical but shifted in time by one cycle. In order to reduce this strong dependency among bits (and in particular from a test usage perspective), either the 32 right bits or alternatively 32 left bits should be used to test a CUT. The 64 bit streams can be used if the CUT is a combinational circuit since there is no problem of correlation between the test

TABLE I
PROPORTION OF THE BIT STREAMS PASSING EVERY TEST



	1-round-AES	1-round-DES	LFSR
Monobit	100.0 %	100.0 %	21.1 %
Blk. frequency	99.2 %	100.0 %	100.0 %
Cumulative Sums	99.2 %	96.9 %	100.0 %
Runs	98.4 %	100.0 %	100.0 %
Long Runs of Ones	99.2 %	96.9 %	96.9 %
Rank	100.0 %	100.0 %	100.0 %
DFFT	96.9 %	93.8 %	98.4 %
Universal	98.4 %	100.0 %	100.0 %
Approximate Entropy	99.2 %	100.0 %	97.7 %
Serial1	100.0 %	100.0 %	0.0 %
Serial2	99.2 %	96.9 %	100.0 %
Linear Complexity	97.7 %	100.0 %	100.0 %
Aperiodic Templates	100.0 %	100.0 %	21.1 %
Periodic Templates	98.4 %	96.9 %	100.0 %
Random Excursion	97.7 %	96.9 %	66.4 %

data in the scan chains in this case. Statistical results of Table 1 (for 1-round-DES) are collected only from the 32 right output bits.

In the case of multiple scan chains, another important issue is the randomness of the so-formed *vectors* $V_j=(v_{0j},....v_{nj})$ where v_{ij} is the j^th value of the i^th bit stream. In fact, it could happen that every bit stream $\{v_{i1},v_{i2},....,v_{ik}\}$ is random while the sequence of vectors is not. For instance, if the bit streams are random but equal, the sequence of vectors is only composed of all-0s vectors and all-1s vectors. The NIST suite allows checking the randomness of *bit streams* only. Thus, for checking the randomness of the vectors, we built up a "bit stream" obtained by concatenating the successive vectors. The so-obtained 1-round-AES and 1-round-DES bit streams pass all the NISTS tests, while the LFSR bit stream passes only 2 tests over 15.

As a conclusion of these experiments, the randomness properties of the bit streams issued from AES and DES crypto-cores are as good as or even better than LFSR bit streams. Thus, the 1-round-AES and 1-round DES can be considered as a candidate for pseudorandom pattern generation.

Self-Test Mode: This section provides results in terms of stuck-at FC obtained on the crypto-cores using the SELF_TEST mode. First, a theoretical approach allows precalculating the number of encryptions needed to reach 100% FC with a given confidence level. The theoretical number of required encryptions for 100% FC is then confirmed by fault simulation.

We considered the following aspects.

- The stream generated by a crypto algorithm when the input is fed by its output can be considered as random. This property has been discussed in Section III.
- Clearly, the presence of a fault modifies the round output and thus the next test pattern (circular scheme). This is taken into account in our experiments in which we injected every single stuck-at fault, fault simulated the circuit and only observed the value of the final signature produced by the crypto-core (DES or AES).
- One of the operations of the crypto-algorithm is a substitution function that is implemented by S-boxes. S-boxes represent the largest part of the crypto-cores. Their inputs are independently fed by a subpart of the round inputs.We can therefore assume that they are fed by a random source, receiving a pattern every clock cycle.
- An S-box needs *k* deterministic patterns to be fully tested and it receives one random pattern every clock cycle.
- Other parts of the round module (mainly wires and XOR operations) receive one pattern every clock cycle as well. Since the other parts have lower complexity than the S-boxes, it can be expected that they will be fully tested by the time the S-boxes received a sufficient number of test patterns. The theoretical number of required clock cycles (or patterns) can be precomputed as the theoretical number of clock cycles for testing the S-boxes. Equation (1) gives the length (*n*) of the minimal-length random sequence that includes the *k* targeted test patterns with a given confidence level *P* [20], with p being the probability of each pattern to appear

$$P[X \le n] = 1 - \sum_{i=1}^{k} 1^{i+1} \binom{k}{i} (1-ip)^n.$$

Conversely to the so-called circular BIST approach where dedicated source and sink test resources (respectively TPG and ORA functions) are build from existing FFs from the CUT and extra logic (e.g., LFSR based pseudo-random generation and signature analysis), the proposed SELF_TEST scheme only requires a slight modification of the cryptocore's controllers. Circular BIST improvements as proposed in [21] for breaking correlations in the test patterns, limit cycles and jump to states that detect random-pattern-resistant faults do not improve the proposed approach due to inherent random-pattern testability of the crypto-cores and their propensity to generate sufficiently long cycles.

DES Self-Test: We considered a confidence level of P=99% for a sequence of k=64 patterns. This sequence represents an upper bound since it corresponds to the exhaustive test set for any six-input S-box. Each pattern has the same probability to appear in the pseudorandom scheme, i.e.,p=1/64. From (1), it comes that the minimal-length random sequence is n=540patterns. According to the implementation of the S-boxes and the actual number of required test patterns *k* (k=64), the length of the test procedure can vary from 440 clock cycles (28 encryptions) to 540 cycles (34 encryptions).

Experimental results confirmed our hypothesis. We fault simulated the DES with several keys and initial input messages. After 25 encryptions (i.e., 400 clock cycles), the whole circuit (round module and control module) has always been fully tested.

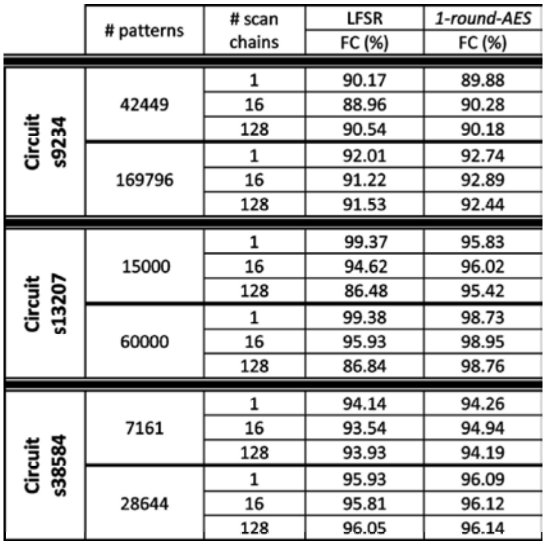
Concerning the key generation module, it mainly consists of wiring. If the primary key is not hard-wired, a solution to test this module is to use the Boolean inverse value of the key during the last encryption. In this way, all the stuck-at faults in this module are tested.

AES Self-Test: From (1), it comes that the minimal random sequence length should include n=2593random patterns for exhaustively testing the AES S-boxes (k=256) with a confidence level of 99%. For various implementations of the AES S-boxes and thus for different deterministic test sets, we calculated the length of the random sequence for including the *k* targeted deterministic patterns. The minimal length is ranging from 2400 to 2593 patterns depending on the implementation.

This hypothesis has been confirmed during the experiments since 100% FC has been achieved on the whole circuit (round module, key generation module, controller) after 210 encryptions (i.e., 2100 rounds). This experiment has been repeated with different plaintexts and secret keys as starting points and we obtained test sequences ranging from 2100 to 2500 patterns.

Test Pattern Generation Mode: It has been shown in Section III that 1-round-AES and 1-round-DES deliver sequences with good random properties. Thus, both can be used for random testing of other modules in the system. In this section, we investigate how good the sequences are compared to standard LFSR generated sequences for random testing. For that we fault simulated some circuits with the so-obtained sequences. Since 1-round-DES and 1-round-AES sequences present similar random properties, we did the experiments with 1-round-AES sequences only.

Table 2: Fault coverage



	# patterns	# scan chains	LFSR	*1-round-AES*
			FC (%)	FC (%)
Circuit s9234	42449	1	90.17	89.88
		16	88.96	90.28
		128	90.54	90.18
	169796	1	92.01	92.74
		16	91.22	92.89
		128	91.53	92.44
Circuit s13207	15000	1	99.37	95.83
		16	94.62	96.02
		128	86.48	95.42
	60000	1	99.38	98.73
		16	95.93	98.95
		128	86.84	98.76
Circuit s38584	7161	1	94.14	94.26
		16	93.54	94.94
		128	93.93	94.19
	28644	1	95.93	96.09
		16	95.81	96.12
		128	96.05	96.14

Fault simulations have been performed on ISCAS'89 benchmark circuits s9234, s13207 and s38584 using the Synopsys Tetramax test suite. They contain, respectively, 228, 669 and 1451 flip-flops. Each circuit has been analyzed using three scan chains configurations: single san chain, 16 scan chains and 128 scan chains.

Table 2 reports FC obtained on these circuits. The first column gives the length of the test sequence. The second one shows the three scan chain configurations. For the single scan chain, the scan path is fed by the rightmost output bit of the TPG. The 16 scan chains of the second configuration are fed by 16 randomly chosen output bits of the TPG. In the last configuration, the whole set of 128 output bits is used.

It can be seen that similar FC is obtained from LFSR and 1-round- AES. To go into details, among the 18 simulations presented here, the highest FC is achieved only three times out of 18 by the LFSR TPG. As expected, the LFSR gives good results on single bit streams but it is less successful on multiple bits generation. This is explained by the correlations that exist between the different streams issued from the LFSR. To overcome this problem, an extra XOR network can be implemented between the TPG and the CUT (i.e., phase shifter) at the cost of additional hardware.

Output Response Analyzer Mode: The role of the ORA is to compact into a single word the sequence of test responses coming from the CUT. The so-obtained signature is compared to the expected one.

However, two different test response sequences may lead to the same signature if multiple erroneous responses (due to faults into the CUT) are generated (i.e., aliasing). The probability of aliasing measures the signature analyzer quality. In cryptographic terms, the signature can be seen as the hash of the input vector sequence and the aliasing probability as the collision probability. For the proposed structure, the aliasing probability after *n* test cycles is

$$P(aliasing) = \frac{1}{2^m} - \left(\frac{1}{2^m}\right)^n$$

where m is the data path bit width. The interested reader can refer to [12] for computation details. With the assumption that all possible errors are equally likely, the fault masking probability tends toward $1/2^m$ for large *n*. Note that this aliasing probability is equivalent to the one of LFSR-based analyzer (i.e., MISR) [22].

$$P(aliasing/MISR) = \frac{2^{n-1}-1}{2^{m+n-1}-1}.$$

CONCLUSION

BIST approaches are effective for secure circuits since they do not rely on visible scan chains, thus preventing scan-based attacks. In this paper, a generic BIST solution for devices including a symmetric cryptographic core has been presented. Conversely to standard BIST solution, this technique entails a negligible area overhead. We proposed a self-test procedure whose principle is to feed the core with its own output and let the device run for a certain number of encryptions and then to compare the output of the final encryption with a precomputed signature. In a very short test time, 100% of FC is achieved. Finally, we also addressed the use of the crypto-cores as TPGs and ORAs for the test of other cores in the circuit. Statistical analysis and experiments show that crypto-cores are good alternatives when already present into the design.

REFERENCES

1. National Bureau of Standards, 1977. U.S. Department of Commerce, "Data encryption standard, federal information processing standard (FIPS)," Publication, pp: 46.
2. Joan, D. and R. Vincent, The Design of Rinjael, AES- The Advanced Encryption Standard. New York: Springer-Verlag.
3. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Special Publication 800-67, Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, 2008.
4. Bardell, P.H. and W.H. McAnney, 1982. "Self testing of multichip logic modules," in Proc. Int. Test Conf., pp: 200-204.
5. Yang, B., K. Wu and R. Karri, 2004. "Scan-based side-channel attack on dedicated hardware implementations on data encryption standard," in Proc. Int. Test Conf., pp: 339-344.
6. Yang, B., K. Wu and R. Karri, 2006. "Secure scan: A design-for-test architecture for crypto chips," IEEE Trans. Computer.-Aided Design Integer. Circuits Syst., 25(10): 2287-2293.
7. Lee, J., M. Tehranipoor, C. Patel and J. Plusquellic, 2005. "Securing scan design using lock and key technique," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (DFT), pp: 51-62.
8. Hely, D., F. Bancel, M.L. Flottes and B. Rouzeyre, 2007. "Securing scan control in crypto chips," J. Electron. Test.: Theory Appl., 23(5): 457-464.
9. Schubert, A. and W. Anheier, 2000. "On random pattern testability of cryptographic VLSI cores," J. Electron. Test.: Theory Appl., 16(3): 185-192.
10. Daemen, J., 1995. "Cipher and Hash Function Design, Strategies based on linear and differential cryptanalysis," Ph.D. dissertation, Katholieke Universities Leuven, Leuven, Belgium.
11. Yang, B. and R. Karri, 2005. "Crypto BIST: A built in self test architecture for crypto chips," in Proc. 2nd Workshop Fault Diagnosis Tolerance Cryptography (FDTC), pp: 95-108.
12. Doulcier, M., M.L. Flottes and B. Rouzeyre, 2008. "AES-based BIST: Selftest, test pattern generation and signature analysis," in Proc. 4th IEEE Int. Symp. Electron. Des., Test Appl. (DELTA), pp: 314-321.
13 Konemann, J. Mucha and G. Zwiehoff, 1979. "Built-in logic block observation technique," in Proc. IEEE Int. Test Conf., pp: 37-41.
14. Shannon, C., 1948. "A mathematical theory of communication," Bell Syst. Tech. J., 27(4): 379-423.
15. Shannon, C., 1949. "Communication theory of secrecy systems," Bell Syst. Tech. J., 28(4): 656-715.
16. Feistel, H., 1973. "Cryptography and computer privacy," Sci. Amer. Mag., 228: 15-23.
17. Schneier, B., 1996. Applied Cryptography, 2nd ed. New York: Wiley.

18. Hellekalek, P. and S. Wegenkittl, 2003. “Empirical evidence concerning AES,” ACM Trans. Model. Compute. Simul., 13(4): 322-333.
19. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Special Publication 800-22, Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD.
20. Shioda, S., 2007. “Some upper and lower bounds on the coupon collector problem,” J. Compute. Appl. Math., 200(1): 154-167.
21. Touba, N.A., 2002. “Circular BIST with state skipping,” IEEE Trans. Very Large Scale Integer. (VLSI) Syst., 10(5): 668-672.
22. Bardell, P.H., W.H. McAnney and J. Savir, 1987. Built-in Test for VLSI: Pseudorandom Techniques. New York: Wiley.