DIFFERENTIAL FAULT ANALYSIS ATTACK RESISTANT ARCHITECTURES FOR THE ADVANCED ENCRYPTION STANDARD*

Mark Karpovsky, Konrad J. Kulikowski, Alexander Taubin
Reliable Computing Laboratory,Department of Electrical and Computer Engineering,Boston University,8 Saint Mary's Street, Boston, MA 02215 {markkar, konkul, taubin}@bu.edu

Abstract: We present two architectures for protecting a hardware implementation of AES against side-channel attacks known as Differential Fault Analysis attacks. The first architecture, which is efficient for faults of higher multiplicity, partitions the design into linear (XOR gates only) and nonlinear blocks and uses different protection schemes for these blocks. We protect the linear blocks with linear codes and the nonlinear with a complimentary nonlinear operation resulting in robust protection. The second architecture uses systematic nonlinear (cubic) robust error detecting codes and provides for high fault detection for faults of low and high multiplicities but has higher hardware overhead.

Key words: Advanced Encryption Standard; Differential Fault Analysis

1. INTRODUCTION

Cryptographic algorithms are designed so that by observing only the inputs and outputs of the algorithm it is computationally infeasible to break the cipher, or equivalently determine the secret key used in encryption and decryption. Thus, the algorithm itself does not leak enough useful information during its operation to compromise its security. However, when a physical implementation of the algorithm is considered, additional information like power consumption, behavior as a result of internal faults, and timing of the circuit implementing the algorithm can provide enough

* This work was supported by the Academy of Finland, Project No 44876 (Finish Center of Excellence Program (2000-2005))

information to compromise the security of the system. Attacks based on the use of this implementation specific information are known as Side Channel Attacks (SCA) [1,2].

In this paper we focus on the SCA's known as Differential Fault Analysis (DFA) attacks against the Advanced Encryption Standard (AES) [3]. DFA attacks are based on deriving information about the secret key by examining the differences between ciphers resulting from correct operations and faulty operations. DFA attacks have been applied to AES in [4-8]. Several methods for protection of AES have been developed. However, the current methods do not provide an adequate solution since they either require duplication in time or space [9], or they are not effective for all fault attacks [10,11].

We propose two methods for the protection of one round of AES. The first is a hybrid method which partitions AES into linear (XOR only) and nonlinear blocks and uses different protection techniques for the two different types of circuits. We protect the nonlinear blocks by performing nonlinear complimentary operation with respect to the function of the original block. The linear block is protected with linear codes. Using this hybrid partitioning method allowed us save on redundant hardware.

The second method uses systematic nonlinear robust codes. For the robust codes used in the design the probability of error detection depends not only on the error pattern (as in the case for linear codes) but also on the data itself. If all the data vectors and error patterns are equiprobable, then the probability of injecting an undetectable error if the device is protected by our robust codes is 2^{-2r} versus 2^{-r} if the device is protected by any linear code with the same r (r is a number of redundant bits which are added for data protection).

The error detection procedures of both designs can be used to detect a DFA attack and disable the card preventing further analysis.

2. DFA ATTACK FAULT MODELS

We refer to a *fault* as a physical malfunction of a part of a circuit. An *error* is a manifestation of fault at the output of the device. An error is the difference (componentwise XOR) of the correct and distorted outputs of the device.

In this paper, we consider protection against a probabilistic attack. This attack does not necessitate chip depackaging nor expensive probing equipment and is therefore one of the more accessible attacks. In this model the attacker subjects the device to abnormal conditions which will generate faults in the circuit (radiation, high temperature, etc). We consider that

under these conditions the locations of faults is uniformly distributed throughout the circuit and that the probability that a fault will occur in any wire is characterized by the wire distortion rate *p* which is a characteristic of the attack performed. Thus the number of actual faults injected into a circuit is dependent on the size *N* of the circuit and the expected number of faults (multiplicity of faults) is pN where *N* is the number of gates in the circuit.

We present two architectures for the protection of a round of AES from probabilistic attacks. The first method, based on partitioning, is an efficient and effective method under an assumption that probabilistic attacks have a high wire distortion rate and therefore result in the injection of many faults at a time. For the cases where no assumptions can be made about the wire distortion rates, we propose an architecture based on robust codes which is effective for all fault multiplicities, but has a higher hardware overhead than the first.

3. PROTECTION OF ONE ROUND OF AES BY HYBRID PARTITIONING

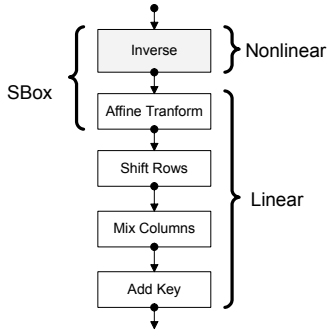


Fig. 1. Transformations involved in one typical round of encryption of AES

Encryption in AES-128 (AES with a 128-bit key) involves performing 10 rounds of transformations on a block of 128 bits with the last tenth round having one less transformation and with the first round being preceded by a round key addition. (The complete AES specification can be found in [3]) In each of the nine typical rounds there are four transformations: SBox, Shift Rows, Mix Columns, and Add Round Key. The last round differs from the rest in that it does not contain the Mix Columns transformation. The SBox transformation actually involves two operations: inversion in $GF(2^8)$ followed by an affine transform which involves a matrix multiplication *M* over $GF(2)$, followed by addition of a constant vector τ. With the

exception of inversion, all other transformations and operations are linear (Fig. 1). That is, they can all be implemented using XOR gates only.

When considering only one round, the 128-bit data path can be divided into four identical independent 32-bit sections. Furthermore, in each of the four partitions the nonlinear inversion is performed on 8-bit data block. Thus, the nonlinear section is composed of 16 disjoint blocks and the linear portion composed of four identical disjoint blocks (Fig. 2).

Based on this partitioning, we designed redundant protection hardware for each of the two types of blocks in the design. The details of each block's method of protection are discussed in the next section.

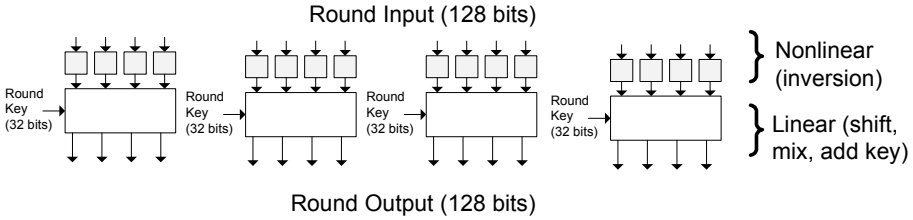


Fig 2. The nonlinear portion of one round can be separated into 16 identical independent blocks. The linear portion can be separated into 4 identical independent blocks.

3.1 Protection of Nonlinear Blocks

The nonlinear block performs inversion in $GF(2^8)$. Since zero does not have an inverse it is defined that the result of the inverse operation on zero is zero.

Our proposed fault detection circuitry for inverters is based on multiplication in $GF(2^8)$ of input and output vectors to verify the condition

$$X * X^{-1} = \left\{ \begin{matrix} 00000001 = I_8, if\ X \neq 0 \\ 00000000, if\ X = 0 \end{matrix} \right\}.$$

Instead of computing the whole eight bit product, we compute only the least $r=2$ bits of the product.

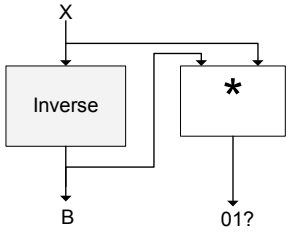


Fig 3. Architecture for protection of nonlinear block. The redundant portion performs partial multiplication in $GF(2^8)$

3.2 Analysis of Error Detecting Probabilities for Nonlinear Blocks

The number of bits, *r,* (r<9) in the signature (the number of bits resulting from partial multiplication), directly translates into the error detection capability of the protection scheme.

The probability that an error in the inverter will be missed is equal to the probability that two uniformly distributed random 8-bit vectors multiplied together will produce the expected *r*- bit constant I_r.

This protection scheme also has the advantage of being *robust* with respect to the $(8+r)$-bit output of the protected inverter (The probability of missing an error in the inverter depends not just only on the error itself but also on the input X). An error $e=(e_{B,}e_R)$ of $(8+r)$-bits, where e_B is an error at the output of the inverter and e_R is an error at the output of the redundant portion, is missed iff

$$[X^{-1}\oplus e_B]*[X]=I_r\oplus e_R$$

where $e_R\in GF(2^r), e_B\in GF(2^8)$ and $\oplus$ is bitwise XOR, or iff

$X*e_B=e_R$ where $X*e_B$ denotes *r* least significant bits of the product between X and e_I in $GF(2^8)$.

Thus, with the exception of an input X of all zeros, all error patterns e are detectable with probability of $1-2^{-r}$ for any given input X. Also, since error detection is dependant on the data X, the probability that an error will be missed after *m* random inputs is 2^{-rm}.

In one round of encryption of AES there are *T*=16 disjoint inverters, each with its own independent error detection. While for a single inverter the probability of missing an error is constant for all fault multiplicities that is not the case when multiple inverters are considered together. The probability that a fault will not be detected if it affects *t* inverters is q^t where q is the probability of missing a fault in one inverter.

Assuming that the distribution of faults is uniform, the probability that a fault of multiplicity *l* will affect *t* out of *T* inverters can be determined as:

$$P_T(t,l)=\frac{N_T(t,l)}{2^l} \quad \text{where} \quad N_T(t,l)=\binom{T}{t}[t^l-\sum_{j=1}^{t-1}N_t(j,l)] \ .$$

Thus, for AES and its *T*=16 inverters the probability of missing a fault of multiplicity *l* in the whole nonlinear portion of encryption of one round is

$$Q_T(l)=\sum_{i=1}^{\min(T,l)} q^i(P_T(i,l))$$

The detection probabilities for the sixteen inverters of AES with two bit signatures (r=2) were simulated using C++. A two input gate level C++ model of the circuit was built with the ability to induce faults at the output of each gate. The model was simulated for different multiplicities of faults. Two types of simulations were performed. The first considered each of the inverters to have an independent error signal. That is, it was assumed that the circuitry which checked the 32 bits of the total error signature for 16 invertors was fault free. Another simulation was performed with each of the error outputs of the nonlinear block being combined together to produce only two error signals for the whole nonlinear portion. In this simulation, the error signals which were expected to have a value of one (the least significant bit) were ANDed together while the other bit, which was expected to be zero, was ORed together. In this simulation this circuitry was **not** assumed to be fault free. In both of the simulation types, a XOR type of fault was induced (the output of the faulty gate was flipped from its correct value).

As Fig.4a shows, the computed and the experimental miss rates for the case of independent errors in the inverters are quite similar but are not exactly equal. Their difference can be accounted to the approximation in the calculated value of q, the miss rate for one inverter. In the calculation it was assumed to be constant for all multiplicities l. This approximation was not completely correct. As Fig.4b shows there are variations in this probability for small fault multiplicities, but a constant value of $2^{-r} = 0.25$ was used in the calculations.

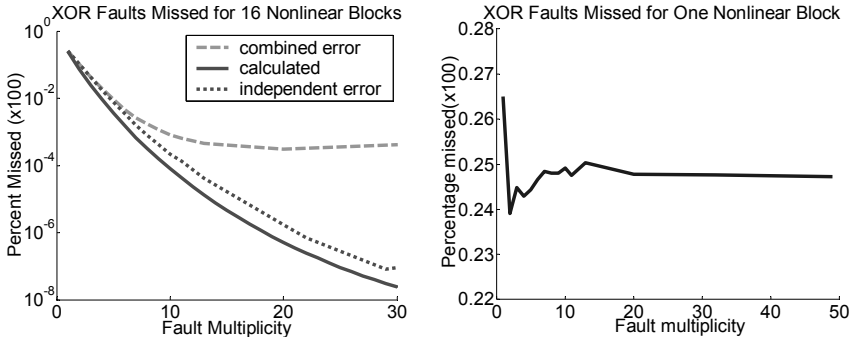


Fig 4. a. Theoretical and experimental miss rates for 16 inverters with independent error signals (dotted line) and with combined signals (dashed line) for r=2 bits in inverter's signatures for 1 input text. b. Experimental miss rate for one nonlinear block as a function of multiplicity of faults.

The simulation results in which the error outputs were combined together are significantly worse (dashed line). For fault multiplicities of ten and higher the miss rate reached a constant of about 0.1 % for one input text.

3.3 Protection of Linear Blocks

Each on of the four linear blocks has 64 bits of input (32 bits from the nonlinear portion and 32 bit of round key) and a 32 bits of output. Due to its large number of inputs and outputs and a relatively small gate count, a linear code proved to be the most cost efficient in terms of its hardware overhead to error detection ratio. The linear block performs three transformations: affine transform, mix columns, and add RoundKey.

The outputs Y can be written in terms of the inputs *B* in the following way:

$$
\begin{aligned}
Y1 &= 02 \bullet (M(B1) \oplus \tau) \oplus 03 \bullet (M(B2) \oplus \tau) \oplus \\
&\quad M(B3) \oplus \tau \oplus M(B4) \oplus \tau \oplus RK1, \\
Y2 &= M(B1) \oplus \tau \oplus 02 \bullet (M(B2) \oplus \tau) \oplus \\
&\quad 03 \bullet (M(B3) \oplus \tau) \oplus M(B4) \oplus \tau \oplus RK2, \\
Y3 &= M(B1) \oplus \tau \oplus M(B2) \oplus \tau \oplus \\
&\quad 02 \bullet (M(B3) \oplus \tau) \oplus 03 \bullet (M(B4) \oplus \tau) \oplus RK3, \\
Y4 &= 03 \bullet (M(B1) \oplus \tau) \oplus M(B2) \oplus \tau \oplus \\
&\quad M(B3) \oplus \tau \oplus 02 \bullet (M(B4) \oplus \tau) \oplus RK4,
\end{aligned}
$$

where $\bullet$ is multiplication in $GF(2^8)$, M is the binary *(8x8)* matrix , $M(Bi)$ is multiplication over GF(2), τ is a constant as defined in AES and RKi are round keys[6].

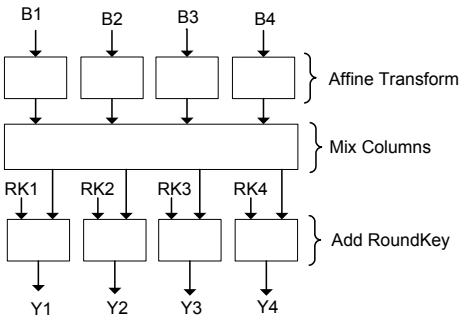


Fig 5. Transformations performed in one linear block.

The design of the linear code for the block was based on the observation that an implementation of the sum $Y1 \oplus Y2 \oplus Y3 \oplus Y4$ is much simpler than of the original block. Indeed:

$$S = Y1 \oplus Y2 \oplus Y3 \oplus Y4$$
$$= M(B1 \oplus B2 \oplus B3 \oplus B4) \oplus RK1 \oplus RK2 \oplus RK3 \oplus RK4 .$$

This function S is computed by the linear predictor and used as an eight-bit redundant signature for the original linear block (see Fig.6). Under fault-free operation, the output of the linear predictor should be equal to the sum of the output of the original linear block. The Error Detecting Network (EDN) sums the output (block P in Fig 6) and compares it to the expected value.

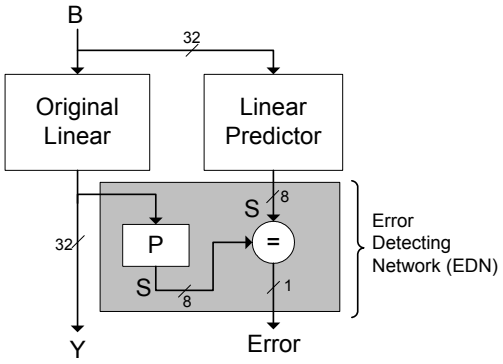
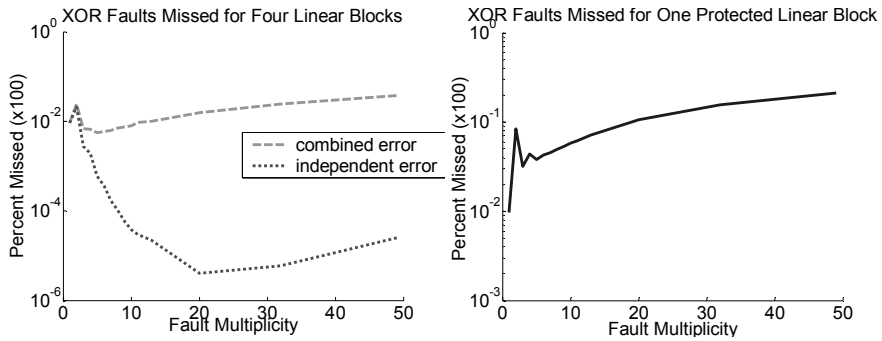


Fig 6. Architecture for protection of linear block.

3.4 Analysis of Error Detection Probabilities for Linear Blocks

A gate-level model of the linear block was built and simulated in C++. Like in the simulations for nonlinear blocks, faults were injected randomly (with equal probabilities of a fault at outputs of the gate) into the circuit with random and uniformly distributed multiplicities in range from 1 to 50. The results of these simulations are presented in Fig.7.

Similarly to the simulations performed on the nonlinear blocks, simulations for fault detection probabilities for one linear block (Fig.7b) and four linear blocks (Fig.7a) with independent and combined error signals were performed. The fault miss rate for one linear block resulted in a miss rate of $5 \pm 6\%$ for one text input. In the case of 4 linear blocks the design where the error signals from each linear block were not fault-free (Fig.7a, dashed line) were significantly worse than when the errors signals were independent for each block (Fig.7a, dotted line)



(a) (b)

Fig 7. a. Experimental miss rates for 4 linear blocks with independent error signals (dotted line) and with combined signals (dashed line) for one input text. b. Experimental miss rate for one linear block as a function of multiplicity of faults.

3.5 Complete Round of Encryption

One typical round of encryption was constructed from the protected linear and nonlinear blocks. Fig.8 shows one forth of one round of the encryptor. The complete round is composed of four identical blocks arranged in parallel. The error signals of the nonlinear portion are chained together to output only a 2 bit signature for all of the 16 inverters. Likewise, the error signal from the four linear blocks is chained together to produce 1 error signal for the whole linear portion of the round. Thus there are 3 error outputs for the whole round. Under fault free operation the nonlinear error outputs should have a value of 01 (excluding an input of all zeros in the input) and the linear error output should have a value of 0.

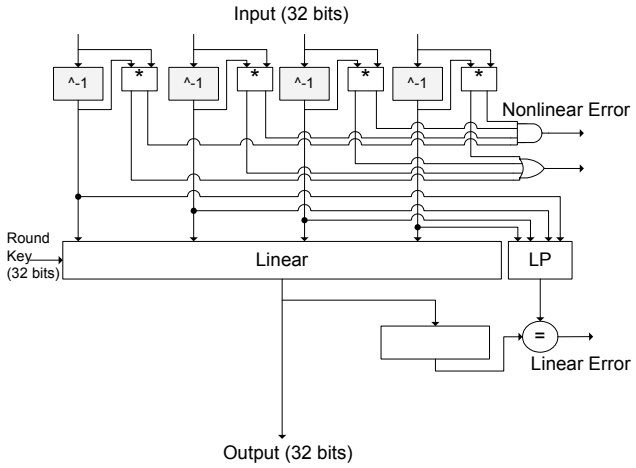


Fig 8. One forth of a typical round of encryption.

The complete protected round has a total hardware overhead (in terms of 2-input gates) of 35%. Table 1 summarizes the overheads for each type of block.

A C++ model of one complete protected round of encryption was built. For random and uniformly distributed texts and round keys, faults of different multiplicities were injected into the circuit. The results for stuck-at-one, stuck-at-zero and XOR fault simulation are presented in Fig.9.

Table 1. Sizes of Components of One Complete Round of Encryption in Terms Two-Input Gates

Component	Gate Count for Original AES	Gate Count for the Redundant Portions	Overhead
Linear Portion (4 blocks)	896	460	51.3%
Nonlinear Portion (16 blocks)	2800	800	28.5%
Error Chaining	0	33	-
Total	3696	1293	35%

When fault detection is considered for one input text, as in Fig.9, the experimental miss rate for stuck-at-one (sa1) and stuck-at-zero (sa0) faults was higher than that for XOR faults. For the unidirectional faults (only sa1 or only sa0), not every injected fault will manifest itself in the circuit. On average, only about 50% of the injected unidirectional faults will manifest themselves at an output of a gate. For XOR faults, since the fault involved flipping a value, 100% of the faults are manifested. Thus, the miss rate curves of the unidirectional faults presented in Fig.9 should be shifted to more precisely reflect the fault multiplicity.

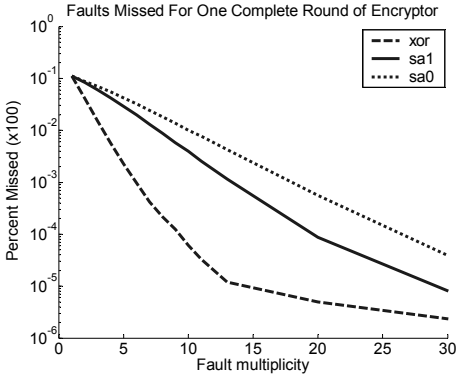


Fig 9. Fault Miss Rates for one complete round of encryption for one input text.

As mentioned in Section 3.2, the design of the nonlinear block resulted in robust protection with respect to its output. Since the nonlinear blocks account for a large portion of the total hardware, it was expected that the whole round will exhibit partial robust behavior. That is, as long as fault affects the nonlinear blocks, it is expected that the detection of that fault is dependent on the input text. Thus, fault miss rate should decrease when multiple random text inputs are considered for the same fault. Simulation results for multiple text inputs for unidirectional and XOR faults are presented in Fig.10.

The simulation results in Fig.10 show considerable improvement for unidirectional faults when multiple text inputs are considered. XOR faults showed limited improvement. The manifestation of stuck-at-faults is different depending on the data, resulting in a different error distribution for each input. That is not the case for XOR faults. XOR fault error manifestation is much less dependant on the data.

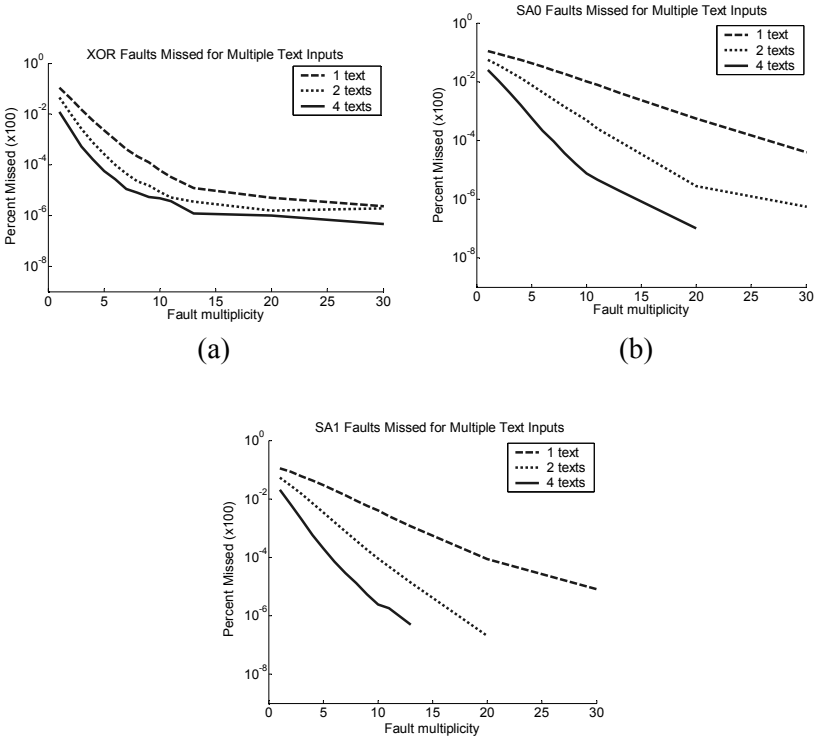


Fig 10. Simulated miss rates for one round of encryptor for multiple texts with a. XOR faults b. Stuck-at-zero faults c. Stuck-at-one faults

The simulation results show that the miss rate improves as fault multiplicity increases. With 4 random text inputs the miss rate drops to 0.0001% for stuck at one faults with a fault multiplicity of 14 (see Fig.10c). The detection for small fault multiplicities is considerably worse since they will affect a small number of blocks. When only one block is affected the detection is only as good the detection in one block.

The design and simulations were only performed for one block of encryption.

4. PROTECTION OF AES BY NONLINEAR SYSTEMATIC ROBUST CODES

4.1 General Robust Architecture

Robust codes [12] can be used to extend the error coverage of linear prediction schemes for AES. Only two extra cubic networks computing $y(x) = x^3$ in $GF(2^8)$ are needed, one in the extended device, and one in the Error Detection Network. The architecture of one round AES encryption with robust protection is presented in Fig.11.

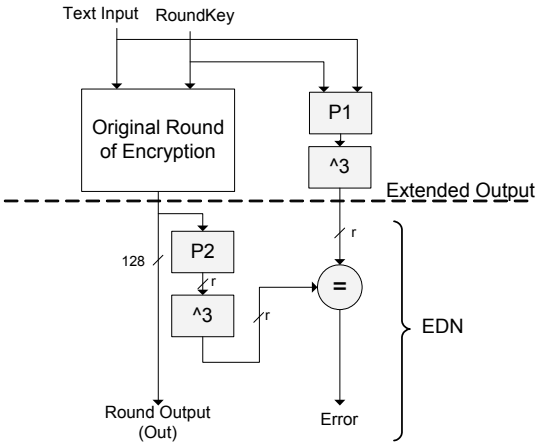


Fig 11. General architecture which uses systematic nonlinear robust protection.

In the architecture in Fig.11 a single linear predictor (block P1) is required for the encryptor. (Note that in this context a linear predictor is such that it generates a signature, which is a linear combination of the outputs of the round. It does not mean that the predictor contains only linear elements.)

The *r*-bit signature of the linear predictor, is cubed in $GF(2^r)$ to produce an *r*-bit output signature (block ^3 in Fig.11), which is nonlinear with respect to the output of the round.

For the robust architecture we have designed a linear predictor which can be used to generate a *r*=32-bit signature. The predictor P1 is designed in a similar fashion to the linear predictor for the linear block presented in the previous section. For four bytes of the output, the predictor predicts one byte, $L'(j)$ $(j = 0,1,2,3)$. For encryption this simplifies to eliminating the mix columns transformation in the predictor P1.

The output of the linear predictor, $L'(j)$, is a 4-byte word which is linearly related to the output of one round of AES. The function of $L'(j)$ with respect to *Out(i,j)* can be written as:

$$L'(j) = \bigoplus_{i=0}^{3} Out(i, j) \text{ where } j \in \{0,1,2,3\}.$$

Thus, the following expressions are valid for AES:

$$\begin{aligned} L'(0) = {} & 01 \bullet Sub(In(0,0)) \oplus 03 \bullet Sub(In(1,0)) \oplus Sub(In(2,0)) \oplus Sub(In(3,0)) \oplus \\ & Sub(In(0,0)) \oplus 02 \bullet Sub(In(1,0)) \oplus 03 \bullet Sub(In(2,0)) \oplus Sub(In(3,0)) \oplus \\ & Sub(In(0,0)) \oplus Sub(In(1,0)) \oplus 02 \bullet Sub(In(2,0)) \oplus 03 \bullet Sub(In(3,0)) \oplus \\ & 03 \bullet Sub(In(0,0)) \oplus Sub(In(1,0)) \oplus Sub(In(2,0)) \oplus 02 \bullet Sub(In(3,0)) \\ & \oplus RK(0,0) \oplus RK(1,0) \oplus RK(2,0) \oplus RK(3,0) \end{aligned}$$

$$\begin{aligned} = {} & Sub(In(0,0)) \oplus Sub(In(1,0)) \oplus Sub(In(2,0)) \oplus Sub(In(3,0)) \\ & \oplus RK(0,0) \oplus RK(1,0) \oplus RK(2,0) \oplus RK(3,0), \end{aligned}$$

where $\bullet$ is multiplication in $GF(2^8)$, $In(i, j)$ is a text input byte to the round, $RK(i, j)$ is one byte round key, and $Sub(In(i, j))$ is the SubBytes transformation on the byte $In(i, j)$ as defined in the AES standard [6].

Since $Sub(In(i, j)) = M(In(i, j)^{-1}) \oplus \tau$,

We have $L'(0)$:

$$\begin{aligned} L'(0) = {} & M(In(0,0)^{-1} \oplus In(1,1)^{-1} \oplus In(2,2)^{-1} \oplus In(3,3)^{-1}) \\ & \oplus RK(0,0) \oplus RK(1,0) \oplus RK(2,0) \oplus RK(3,0). \end{aligned}$$

Extending the procedure to the rest of the bytes of encryption yields:

$$\begin{aligned} L'(1) = {} & M(In(0,1)^{-1} \oplus In(1,2)^{-1} \oplus In(2,3)^{-1} \oplus In(3,0)^{-1}) \\ & \oplus RK(0,1) \oplus RK(1,1) \oplus RK(2,1) \oplus RK(3,1), \end{aligned}$$

$$L'(2) = M(In(0,2)^{-1} \oplus In(1,3)^{-1} \oplus In(2,0)^{-1} \oplus In(3,1)^{-1}) \\ \oplus RK(0,2) \oplus RK(1,2) \oplus RK(2,2) \oplus RK(3,2),$$
$$L'(3) = M(In(0,3)^{-1} \oplus In(1,0)^{-1} \oplus In(2,1)^{-1} \oplus In(3,2)^{-1}) \\ \oplus RK(0,3) \oplus RK(1,3) \oplus RK(2,3) \oplus RK(3,3).$$

In the Error Detecting Network (EDN) the block P2 compresses the 128 bits into r=32 by bitwise XOR to match the output of the predictor P1. The output of the block P2 is also cubed in $GF(2^r)$. It is this cubed compressed output which is compared to the cubed output of the predictor P1. Under correct operation these two outputs should be equal. It was shown in [12] that the introduction of the nonlinear cubic operation resulted in the reduction of the fraction of undetectable errors at the extended output from 2^{-r} to 2^{-2r} without increasing the redundancy r of the original linear code.

4.2 Analysis for Error Detecting Probabilities for Robust Architecture

A gate-level model of this design was simulated using C++. Two types of simulations were performed. In the first faults were injected into all parts of the circuits. The other assumed that the error detecting network (block P2, the second cubic network and the comparator) were fault-free. The results of these gate level simulations are presented in Fig.12.

To explain the results of simulations of Fig.12 we note that the EDN in simulated architecture was unprotected. As a result, it had a significant impact on the overall fault detection for the proposed design. As Fig.12 shows, when faults were included in the EDN (dashed and dotted lines for stuck-at-zero and stuck-at-one respectively), the fault detection probabilities remained almost constant for higher fault multiplicities. In contrast, when the EDN was considered to be fault free, the fault miss rate quickly dropped as fault multiplicity decreased. The EDN accounts for about 25% of the complete protected round.

The detection performance differed substantially for this approach than that of the first. In this design, even for low fault multiplicities the miss rate remained at a low 0.1%. This improvement came at a price. When a single round of encryption is protected using this approach the overhead of the protection exceeds 150% in terms of the gate count. This high overhead is a result of the high cost of the cubic networks. The overhead can be decreased when the complete AES is protected using this method, including decryption and key expansion. Protecting a larger design offsets the large cost of the cubic networks.

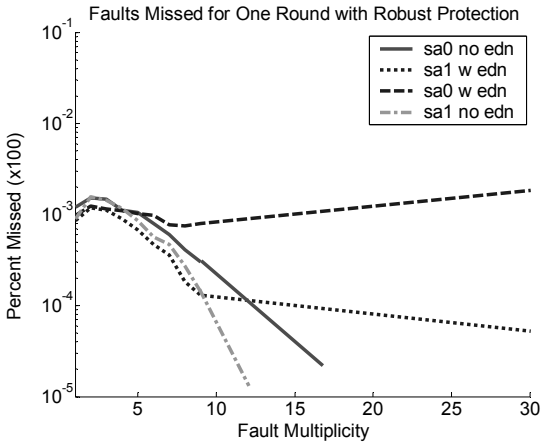


Fig12. Simulation results for one round of encryption with robust protection for one input text.

We note also that this architecture provides for robust detection of XOR faults such that the corresponding errors do not depend on input texts (these may be faults in the linear parts or in the output register of the round). For these faults the probability of missing a fault for one input text is $2^{-r+1} = 2^{-31}$ and probability of missing a fault for all texts is $2^{-2r} = 2^{-64}$.

5. CONCLUSIONS

We presented two methods for protecting the Advanced Encryption Standard against Differential Fault Attacks. The two methods had different overheads and different fault detection probabilities characteristics. We presented also gate-level simulation results for one round of encryption for both architectures.

The first method, which is useful for attacks with high wire distortion rate and based on a hybrid partitioning, had a low hardware overhead of 35%. This method was able to achieve a fault miss rate of 0.01% for one stuck-at-fault with multiplicity of 30 for one text input. For faults of small multiplicity the method's detection rate was substantially worse.

The second method, which is efficient for all wire distortion rates and is based on systematic robust codes, had a high hardware overhead of 150%. However, this method had a much lower miss rate for faults of small multiplicities. Even for faults of multiplicity of one, the miss rate was only

about 0.1%. This method is also very efficient for XOR faults resulting in errors which do not depend on input texts.

References

[1] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Crypto 96, Proceedings, Lecture Notes In Computer Science ol. 1109, N. Koblitzed., Springer-Verlag, 1996.

[2] P. Kocher, J. Jaffe, B. Jun, "Differential Power Analysis," Advances in Cryptology - Crypto 99 Proceedings, Lecture Notes In Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999. [2] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, *Side Channel Cryptanalysis of Product Ciphers*, ESORICS '98 Proceedings, 1998, pp. 97-110.

[3] FIPS PUB 197: *Advanced Encryption Standard,* http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[4]C.N. Chen and S.-M.Yen, *Differential Fault Analysis on AES Key Schedule and Some Countermeasures*, ACISP 2003, LNCS 2727, pp.118-129, 2003

[5] P. Dusart, G. Letourneux, O. Vivolo, *Differential Fault Analysis on AES,* Cryptology ePrint Archive, Report 2003/010. Available: http://eprint.iacr.org/2003/010.pdf

[6]C. Giraud. *DFA on AES*. Cryptology ePrint Archive, Report 2003/008. Available: http://eprint.iacr.org and http://citeseer.nj.nec.com/558158.html

[7] Johannes Blömer, Jean-Pierre Seifert: Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). Financial Cryptography 2003: pp. 162-181

[8] Jean-Jacques Quisquater, Gilles Piret, "A Differential Fault Attack Technique Against SPN Structures, with Application to the AES and KHAZAD", (CHES 2003), Volume 2779 of Lecture Notes in Computer Science, pages 77-88, Springer-Verlag, September 2003

[9] Ramesh Karri, Kaijie Wu, Piyush Mishra, Yongkook Kim, Concurrent Error Detection of Fault Based Side-Channel Cryptanalysis of 128-Bit Symmetric Block Ciphers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.21, No.12, pp. 1509-1517, 2002

[10] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri and V. Piuri, *Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard,* IEEE Transactions on Computers, VOL. 52, NO. 4, 2003

[11] Ramesh Karri, Grigori Kuznetsov, Michael Gössel: Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers. CHES 2003. pp.113-124

[12] M.G.Karpovsky and A. Taubin, "A New Class of Nonlinear Systematic Error Detecting Codes", *to be published in IEEE Info Theory, 2004*