DFARPA: Differential Fault Attack Resistant Physical Design Automation

Mustafa Khairallah*, Rajat Sadhukhan†, Radhamanjari Samanta†, Jakub Breier*, Shivam Bhasin*,
Rajat Subhra Chakraborty†, Anupam Chattopadhyay*and Debdeep Mukhopadhyay†
*Nanyang Technological University, Singapore
mustafam001@e.ntu.edu.sg,{jbreier,sbhasin,anupam}@ntu.edu.sg
†Department of Computer Science and Engineering, IIT Kharagpur, India
{rschakraborty,debdeep}@cse.iitkgp.ernet.in, rajat.sadhukhan@iitkgp.ac.in, radhamanjari@gmail.com

***Abstract*—Differential Fault Analysis (DFA), aided by sophisticated mathematical analysis techniques for ciphers and precise fault injection methodologies, has become a potent threat to cryptographic implementations. In this paper, we propose, to the best of the our knowledge, the first "DFA-aware" physical design automation methodology, that effectively mitigates the threat posed by DFA. We first develop a novel floorplan heuristic, which resists the simultaneous corruption of cipher states necessary for successful fault attack, by exploiting the fact that most fault injections are localized in practice. Our technique results in the computational complexity of the fault attack to shoot up to exhaustive search levels, making them practically infeasible. In the second part of the work, we develop a routing mechanism, which tackles more precise and costly fault injection techniques, like laser and electromagnetic guns. We propose a routing technique by integrating a specially designed ring oscillator based sensor circuit around the potential fault attack targets without incurring any performance overhead. We demonstrate the effectiveness of our technique by applying it on state of the art ciphers.**

I. Introduction

Passive and active side channel attacks are growing threats against secure cryptographic hardware design. Though several mitigation techniques to counter such threats exist at the algorithm level, it is imperative to improve our existing physical design automation tools to generate hardened designs against such threats. There are several such design automation methodologies proposed for passive side channels [8]. There are also few reported works on classical fault tolerance placement and floorplanning techniques, but they are agnostic of the DFA security issues [7]. To the best of our knowledge, there are no prior works that propose physical design automation techniques nullifying general differential fault attack assumptions.

A. Our Contributions

- First, we summarize the fault attack assumptions and demonstrate how those can be nullified using generic floorplanning and routing techniques.
- Second, we identify and study trade-offs between security and performance overhead.
- Third, we validate our techniques by benchmarking on AES, and Plantlet, a representative block cipher, and stream cipher, respectively.

This paper is organized as follows. Sections II and III describe the two test-cases, DFA-aware floorplan on *AES* and DFA-aware routing on *Plantlet*. Section IV describes experimental results on the above two test cases. Finally, Section V concludes this work.

II. DFA-Aware Floorplan

A. The Floorplan Flow

The idea of the proposed DFA aware floorplan (Fig. 1) stems from the facts that 1) the exploitable fault space for ciphers is limited to a subspace of the universal fault space [2], and 2) for most fault injection techniques the effect of the fault is localized to a region, meaning the probability of a region inside a radius from the point of fault injection is high [1] [5].

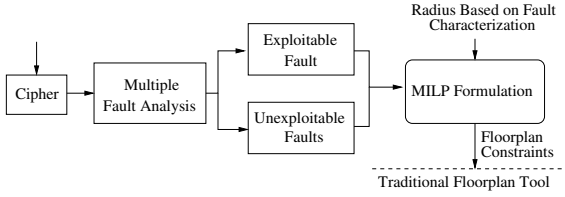


Fig. 1. DFA Aware Floorplanning Flow

In the following subsections, we explain the two steps of the methodology: 1) To perform a multiple fault analysis, and 2) To generate floorplan constraints for DFA-aware floorplan. We illustrate the process through an example on the Advanced Encryption Standard (AES), as that is the de-facto standard block cipher world-wide.

B. Multiple Fault Analysis

Multiple Fault Analysis attempts to explore the fault space which is exploitable to a fault analysis. Formally, let us define a block cipher as $E_k : \mathcal{P} \rightarrow \mathcal{C}$, where $p \in P$ is the plaintext and $c \in C$ is the ciphertext, while $k \in \mathcal{K}$ is the secret key. The internal of the cipher is usually denoted by a state matrix, B^r, where r denotes the round number. Thus, a block cipher can be formalized as a composition of M rounds, where $B^0 = p, B^r = F_r(B^{r-1}, k^{r-1}), c = B^M$, $1 \leq r < M$ and the r^{th} round is denoted as F_r. The master key is the key k^0, while all other round keys are derived by applying a key-scheduling algorithm. The state of the cipher,

B^r, is usually composed of t components of w-bits each. Thus, $B^r = (B_1, B_2, \cdots, B_t)^r$. The round function, F_r, is typically made of two types of transformations: 1) linear diffusion layers: $D_r(B_{i_1}, B_{i_2}, \cdots, B_{i_l}) = a_1 B_{i,1} \oplus a_2 B_{i,2} \cdots a_l B_{i,l}$, and 2) non-linear layer: $S_r(B_{j,n})$, usually called the Substitution Box (S-Box) and applied on each component independently. The total fault space $\mathcal{F}$ is of cardinality 2^{tw}. We assume a fault value $f \in \mathcal{F}$, and decide whether it is exploitable.

In the analysis, we assume that in a chosen round, r, the induced fault, $f^r = (f_1, \cdots, f_t)^r$ diffuses in the cipher and corrupts all the components of the state. We subsequently track the propagation of the fault and consider the effect of the penultimate diffusion layer, D_{M-1}, which acts on the state components $(B_{i_1}, B_{i_2}, \cdots, B_{i_l})^{M-1}$, and the faulty state $(B'_{i_1}, B'_{i_2}, \cdots, B'_{i_l})^{M-1}$. We compute the differential value of the component and denote it as: $a_1\Delta(B_{i_1}) \oplus a_2\Delta(B_{i_2}) \cdots a_l\Delta(B_{i_l})$, where $\Delta(B_{i_l}) = B_{i_l} \oplus B'_{i_l}$ is the differential value due to the penultimate round diffusion layer.

Finally the penultimate round key addition is done via a linear operation, which is followed by the last round S-Box. The attacker then checks for the equations:

$$\begin{aligned} \Delta(B^M_{1_1}) &= S^{-1}(C_{1_1} \oplus K^M_{1_1}) \oplus S^{-1}(C'_{i_l} \oplus K^M_{1_1}) \\ &\cdots \quad \cdots \\ \Delta(B^M_{1_l}) &= S^{-1}(C_{1_l} \oplus K^M_{i_l}) \oplus S^{-1}(C'_{i_l} \oplus K^M_{i_l}) \end{aligned}$$

The fault f^r is defined to be **exploitable** if the solution to the above system of equations leads to a key-space of the key components, $K_{i_1}, \cdots, K_{i_l}$ less than 2^{wl}. We illustrate this process with a case study on AES.

C. Exploitable Fault Space of AES

AES-128 is a block cipher with 10-rounds where each state can be represented as a two-dimensional matrix; $B_{ij}(0 \le i \le 3, 0 \le j \le 3)$ represents a byte in the state matrix.

The regions correspond to diagonals, where$diagonal(D_i)$ [6] of a state matrix is defined to be a set of 4-byte elements such that: $D_i(0 \le i \le 3) = (B_{j,(i+j) \ modulo \ 4} : 0 \le j \le 3)$.

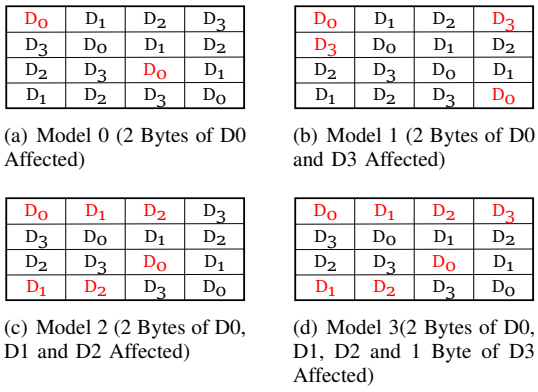
TABLE I
BINARY VALUES OF x_{ij} AND y_{ij}

x_{ij}	y_{ij}	**Physical Interpretation**	**constraint**
0	0	i is to the left of j	$x_i + b_i \le x_j$
1	0	i is to the right of j	$x_i - b_j \ge x_j$
0	1	i is at bottom of j	$y_i + h_i \le y_j$
1	1	i is at top of j	$y_i - h_j \ge y_j$

We classify the faults in the four diagonals as defined before, depending on which we have 4-diagonal fault models as shown below.

- DM0 (Fig. 2(a)): Fault induced in 1 of the diagonals.
- DM1 (Fig. 2(b)): Fault induced in at most 2 diagonals.
- DM2 (Fig. 2(c)): Fault induced in at most 3 diagonals.
- DM3 (Fig. 2(d)): Fault induced in at most all 4 diagonals.

Fault injected at the input of the 8^{th} round makes the fault propagate to the entire state at the output of the 9^{th} round as we observe the penultimate round diffusion layerwhich operates on 4 bytes of the 9^{th} round-state. Using the MixColumns matrix values as per the AES specification, $\Delta(B^{10}_{i,j}) = 2\Delta(B^9_{i,j}) + 3\Delta(B^9_{i+1,j+1}) + \Delta(B^9_{i+2,j+2}) + \Delta(B^9_{i+3,j+3})$. Subsequently, we write further 4 equations considering the effect of the last round ShiftRows. For example, we write $\Delta(B^{10}_{i,j}) = S^{-1}(c_{i,j} \oplus K^{10}_{i,j}) \oplus S^{-1}(c'_{i,j} \oplus K^{10}_{i,j})$. Indeed



(a) Model 0 (2 Bytes of D0 Affected)

(b) Model 1 (2 Bytes of D0 and D3 Affected)

(c) Model 2 (2 Bytes of D0, D1 and D2 Affected)

(d) Model 3(2 Bytes of D0, D1, D2 and 1 Byte of D3 Affected)

Fig. 2. Diagonal Fault Models

depending on the underlying fault model $DM0$, $DM1$, $DM2$, or $DM3$ in which the induced fault f^8 belongs, we have 3 zero values, 2 zero values, 1 zero value, or no zero value of the differentials $\Delta(B^9_{i,j})$, $\Delta(B^9_{i+1,j+1})$, $\Delta(B^9_{i+2,j+2})$, $\Delta(B^9_{i+3,j+3})$. This leads to different reduced key spaces, i.e. $2^8, 2^{16}, 2^{24}$, or 2^{32}. This shows that while all the first three fault models are exploitable, the last fault model $DM3$ is unexploitable.

D. Constraint Generation for MILP

Let the unexploitable fault space of the state matrix of a cipher be denoted by the fault patterns, $F^U = \{F^U_{i_1}, \cdots, F^U_{i_p}\}$, where each F^U corresponds to a group of registers, $F^U_{i_1} = \{B_{j_1}, \cdots, B_{j_q}\}$. The objective of the floorplan tool would be to generate constraints so that the attacker always gets the unexploitable faults, which implies that the registers, $B_{j_1}, \cdots, B_{j_q}$ are placed inside the radius R. Likewise, we have an exploitable fault space denoted as $F^E = \{F^E_{i_1}, \cdots, F^E_{i_s}\}$, where each fault pattern is denoted as say, $F^E_{i_1} = \{B_{j_1}, \cdots, B_{j_t}\}$. The floorplan tool now sets constraints so that any pair of these registers is placed at least a distance of R apart.

1) MILP Formulation: The objective function for the MILP aims to minimize the area $(width \times height)$ of the chip. Let there be n blocks or macros $(M_i : (0 \le i < n))$ each having a breadth $(b_i : (0 \le i < n))$, a height $(h_i : (0 \le i < n))$ and a coordinate of the lower left corner, denoted by (x_i, y_i). Let $H = \sum_{i=0}^{n-1} h_i$ and $B = \sum_{i=0}^{n-1} b_i$ respectively. The block dimension constraints can be represented by following two inequalities:

$x_i + b_i \le B$ and $y_i + h_i \le H$

Non-overlapping constraints for two blocks, i and j, use the two binary variables x_{ij}, y_{ij} to represent the relative positions of them with respect to the other as given in Table I. Following four constraints denote the block non-overlapping constraints:

$x_i + b_i \leq x_j + B \times (x_{ij} + y_{ij}),\ 1 \leq i < j < n$
$y_i + h_i \leq y_j + H \times (1 + x_{ij} - y_{ij}),\ 1 \leq i < j < n$
$x_i - b_j \geq x_j - B \times (1 - x_{ij} + y_{ij}),\ 1 \leq i < j < n$
$y_i - h_j \geq y_j - H \times (2 - x_{ij} - y_{ij}),\ 1 \leq i < j < n$

2) MILP Formulation for AES: As described before, we have obtained the exploitable and unexploitable fault spaces. The floorplan algorithm should be constrained to ensure that the inter-distances of the registers corresponding to exploitable faults should be at least R distance apart, whereas, those corresponding to the unexploitable faults should be within a distance of R.

The equations are presented when two blocks at (x_i, y_i), and (x_j, y_j) are to be placed at a distance more than R apart. Therefore, i, j are chosen from the same diagonal, DM_k, $0 \leq k < 4$. The objective of MILP would be to optimize the parameter height of the chip, denoted by h. Let $F_s^U \in F^U$ and $F_s^E \in F^E$, where $F_s^U = \{B_{m_1}, \cdots, B_{m_q}\}$ and $F_s^E = \{B_{n_1}, \cdots, B_{n_q}\}$. The equation system takes R as an input depending on the fault clustering property, and forms the following equations:

$minimize : h$
$subject\ to :$
$x_i + b_i + R \leq B,\ n_1 \leq i \leq n_q,\ i \in F_s^E$
$y_i + h_i + R \leq h,\ n_1 \leq i \leq n_q,\ i \in F_s^E$

$$\left.\begin{array}{l} x_i + b_i + R \leq x_j + B(x_{ij} + y_{ij}) \\ y_i + h_i + R \leq y_j + H(1 + x_{ij} - y_{ij}) \\ x_i - b_j + R \geq x_j - B(1 - x_{ij} + y_{ij}) \\ y_i - h_j + R \geq y_j - H(2 - x_{ij} - y_{ij}) \end{array}\right\} \begin{array}{l} n_1 \leq i < j \leq n_q \\ :\ i, j \in F_s^E \end{array}$$

$x_i, y_i \geq 0,\ n_1 \leq i \leq n_q$

While for rest of the blocks belonging to different diagonal, following equations have to be applied:

$x_i + b_i \leq B,\ m_1 \leq i \leq m_q,\ i \in F_s^U, i \notin F_s^E$
$y_i + h_i \leq h,\ m_1 \leq i \leq m_q,\ i \in F_s^U, i \notin F_s^E$

$$\left.\begin{array}{l} x_i + b_i \leq x_j + B(x_{ij} + y_{ij}) \\ y_i + h_i \leq y_j + H(1 + x_{ij} - y_{ij}) \\ x_i - b_j \geq x_j - B(1 - x_{ij} + y_{ij}) \\ y_i - h_j \geq y_j - H(2 - x_{ij} - y_{ij}) \end{array}\right\} \begin{array}{l} m_1 \leq i < j \leq m_q \\ :\ i, j \in F_s^U, i \notin F_s^E \end{array}$$

$x_i, y_i \geq 0,\ m_1 \leq i < m_q$

Likewise, following are the constraints when blocks have to be within radius R such that they belong to different diagonals:

$$\left.\begin{array}{l} x_i + b_i + R > x_j + B(x_{ij} + y_{ij}) \\ y_i + h_i + R > y_j + H(1 + x_{ij} - y_{ij}) \\ x_i - b_j + R < x_j - B(1 - x_{ij} + y_{ij}) \\ y_i - h_j + R < y_j - H(2 - x_{ij} - y_{ij}) \end{array}\right\} \begin{array}{l} m_1 \leq i < j \leq m_q \\ :\ i, j \in F_s^U, \\ i \in F_s^E \end{array}$$

Finally, the binary variables x_{ij} and y_{ij} which represent relative position of blocks satisfy the following set of equations:

$x_{ij}, y_{ij} \in \{0, 1\},\ n_1 \leq i < j \leq n_q$
$x_{ij}, y_{ij} \in \{0, 1\},\ m_1 \leq i < j \leq m_q$

III. DFA-Aware Routing

DFA aware floorplan increases the fault injection effort. However, sophisticated injection methods like laser and EM can still inject single-bit errors, which for example in case of AES, can be sufficient to retrieve the secret key. To counter such cases, reactive countermeasures which detect injection attempts are deployed. Such countermeasures work as physical sensors to detect high-energy injections like laser and EM [3].

The sensor is composed of a watchdog ring oscillator (WRO) and a phase detection (PD) circuit. High energy injections impact signal propagation delay, which disturbs the phase of WRO. This phase change can detected by the PD. Authors in [3] report a higher detection rate for digital PD as compared to Phase Locked Loop (PLL) based PD circuit. This sensor stays independent of the underlying circuit and thus can be used in a plug-and-play configuration. A high-level design of the sensor is depicted in Fig. 3. We assume a ball-grid array package, motivating front side injection. Routing the WRO on top-metal layer facilitates detection.

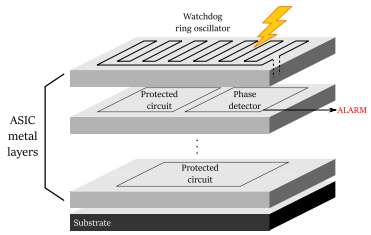


Fig. 3. Ring oscillator sensor deployed on top of the protected circuit.

Design Flow. The design of a custom WRO, which is the key component in the proposed countermeasure, in an automated digital design flow, is a tricky task, as the tools would optimize WRO with multiple inverters. Consequently, a slightly modified flow is adopted. The front-end flow consists of the following steps:

1) A gate level netlist of WRO is written.
2) The number of inverters in the WRO is defined by the required frequency. However, the number of corners in the physical design and consequently, the sensitivity depends on the number of gates/inverters in the design. Therefore, the designer can add more buffers to have more anchor points.

After synthesis, to merge WRO and detector with the cipher, the conventional back-end design flow is invoked with the addition of the following steps:

1) WRO gates are placed manually as per the desired structure.
2) Physical constraints ensure these gates are not optimized during the automatic placement phase.
3) After the rest of the design (the cipher and the PD circuit) is placed, the WRO wires are routed through the top metal layer.

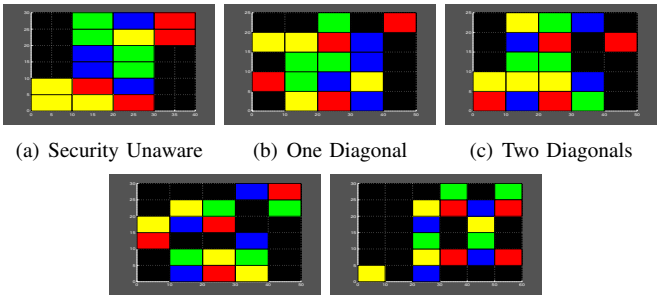
Restricting the top metal layer for WRO results in lower routing capacity for the same core area, which can possibly lead to a routing congestion. Therefore, area overhead estimates are made from protected and unprotected layouts of the Plantlet stream cipher.

IV. Experimental Studies

A. DFA-aware Floorplan

We have compared the area impact on the floorplan of AES blocks by imposing security aware and normal MILP

formulation. Our main concern will be on the position of 16 blocks of AES, where each block consists of shift register and S-box unit. It is assumed that the block is already available as macro in library; equal in height and width. For our experiment, we have assumed every block has height (h_i) 5 $units$ and width(b_i) 10 $units$, the radius R is taken to be 10 (max $\{h_i, b_i\}$). To make the chip dimension *almost*



(a) Security Unaware (b) One Diagonal (c) Two Diagonals

(d) Three Diagonals (e) Four Diagonals

Fig. 4. Security-aware Floorplanning ($H = 30$ $units$ and $L = 40$ $units$, iteration 1)

near to square shape, we have fixed $H = 30$ $units$ and $B = 40$ $units$ in MILP constraints file and executed for all four types of diagonal fault model scenarios along with DFA-unaware scenario.

TABLE II
FLOORPLAN: FAULT MODEL SCENARIOS

Fault Model	Height	Width	Area	Runtime(s)
One Diagonal	25	50	1250	0.318
Two Diagonals	25	50	1250	0.210
Three Diagonals	30	50	1500	0.311
Four Diagonals	30	60	1800	0.312
DFA-unaware	30	40	1200	0.206

After imposing security aware model restrictions to our floorplan problem depending on whether the fault is in one, two, three, or, four diagonals, the MILP solver gave the results as shown in Fig. 4. Note here as seen in case of (Fig. 4(b)) all red blocks belonging to same diagonal are far apart by R unit distance, satisfying the constraints on set $F_s^E = \{B_{00}, B_{11}, B_{22}, B_{33}\}$. It can also be noted from the Fig. 4(b) that one of the red blocks is surrounded by three other blocks (green, yellow, blue), belonging to other three diagonals within radius R unit, satisfying the constraints on set $F_s^U = \{B_{00}, B_{10}, B_{20}, B_{30}\}$. Continuing thus, in Fig. 4(e), while the diagonals are close, the blocks inside all the diagonals are apart. Without imposing any security, the floorplan looks as given in Fig. 4(a), where blocks inside diagonals are close. The height, width and area comparison of the overall block after floorplan is shown in Table II. It can be seen that there is an increase in maximum 44% area (DM3 model) when compared to normal floorplanning. Table II also captures the runtime for each case, from which it is clear that imposing extra constraints does not cost much overhead.

B. DFA-aware Routing

This study has been done on the Plantlet stream cipher [4]. It consists of two registers, one 61-bit LFSR (L_t) and one 40-bit NFSR (N_t). The modified design flow proposed in Section III has been applied to Plantlet, using Synopsys Digital Design Flow and TSMC 65nm Technology with 9 metal layers. The same cipher core has been implemented using both the conventional and modified design flows. In the protected implementation, layers 9 and 8 are reserved for the ring oscillator, layers 7 and 6 are reserved for the power grid and the lower 5 layers are used for routing the design. The implementation results are shown in Table III. The implementation results show that using only 31 inverters, the whole layout can be covered with any cell at most one cell away from the nearest net of the WRO.

TABLE III
POST-LAYOUT RESULTS: PLANTLET

Feature	Unprotected	Protected
Area (μm^2)	1293	1358 (5%)
Max. Path Delay (ns)	0.61	0.62 (1%)
Avg. Dynamic Power (μW)	259.26	551.5 (212%)
Utilization Factor	0.6	0.6 (0%)

The routing flow and the sensor design have almost no performance overhead and add only 5% to the overall layout area. However, the dynamic power analysis shows that a trade-off has to be made between security and power consumption. The power results in Table III are calculated for the maximum possible frequency (1.6 GHz). While many applications use clock frequencies as low as 100 KHz, consuming much less power, the switching frequency of the WRO depends on its design and cannot be controlled by the user. Nevertheless, the WRO can be enabled only when the device is operated in secure mode, and disabled otherwise.

V. CONCLUSION

A set of techniques for DFA-aware physical design automation is introduced in this paper. The fault model assumptions for both block and stream ciphers are first studied, and to nullify those, floorplanning and routing flow are suggested. For both the techniques, the performance overhead is minimal.

REFERENCES

[1] J. Cong and B. Xiao. Defect tolerance in nanodevice-based programmable interconnects: Utilization beyond avoidance. In *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, 2013.

[2] X. Guo, D. Mukhopadhyay, C. Jin, and R. Karri. Security analysis of concurrent error detection against differential fault analysis. *Journal of Cryptographic Engineering*, 2015.

[3] W. He, J. Breier, and S. Bhasin. Cheap and cheerful: A low-cost digital sensor for detecting laser fault injection attacks. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 27–46, 2016.

[4] V. Mikhalev, F. Armknecht, and C. Müller. On ciphers that continuously access the non-volatile key. *IACR Transactions on Symmetric Cryptology*, 2017.

[5] E. I. Muehldorf. Fault clustering: modeling and observation on experimental lsi chips. *IEEE Journal of Solid-State Circuits*.

[6] D. Saha, D. Mukhopadhyay, and D. RoyChowdhury. A diagonal fault attack on the advanced encryption standard. Cryptology ePrint Archive, Report 2009/581, 2009. http://eprint.iacr.org/2009/581.

[7] L. Sterpone and M. Violante. A new reliability-oriented place and route algorithm for sram-based fpgas. *IEEE Transactions on Computers*, 55(6):732–744, 2006.

[8] K. Tiri and I. Verbauwhede. A vlsi design flow for secure side-channel attack resistant ics. In *Proceedings of DATE*, pages 58–63, 2005.