



**Center for Embedded Computer Systems
University of California, Irvine**

Low Overhead DPA Countermeasure using ExCCel (Exploration of Complementary Cells)

Kazuyuki Tanimura and Nikil Dutt

Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-2625, USA

{ktanimur},{dutt}@uci.edu

CECS Technical Report <10-04>
March 19, 2010

Low Overhead DPA Countermeasure using ExCCel (Exploration of Complementary Cells)

Kazuyuki Tanimura and Nikil Dutt

Technical Report CECS-10-04

March 19, 2010

Center for Embedded Computer Systems, University of California, Irvine
Irvine, California 92697-3425, Email: {ktanimur},{dutt}@uci.edu

Abstract—Differential Power Analysis (DPA) side-channel attacks pose serious threats for embedded system security. WDDL was proposed as a countermeasure that can be incorporated into a conventional ASIC design flow using standard cells. However, our spice simulations show that DPA attacks on WDDL still leak secret keys to adversaries despite the doubled area and energy overheads due to the use of complementary cells. These overheads could be crucial problems for smart cards that require low power and small area. This paper proposes ExCCel, a simulated annealing based method that automatically generates and explores combinations of complementary cells for reducing the power-consumption dependency and overheads using standard cells. Our experimental results on the AES S-Box circuit with our explored complementary cells requires 6.1% and 2.1% additional area and energy while WDDL requires 100.3% and 93.4%, respectively. Moreover, ExCCel achieves higher DPA attack resistivity compared to WDDL in many cases. ExCCel's low area and energy overheads and better resistivity makes it a promising countermeasure for smart cards and mobile devices.

I. INTRODUCTION

Side-channel attacks on embedded systems are becoming a serious threat that compromises system security. In particular, Differential Power Analysis (DPA) attack proposed by Kocher et al. [1] is one of the most powerful side-channel attacks. A number of countermeasures have been proposed [2]–[20] to achieve DPA attack resistivity with varying efficacy in DPA attack resistivity and incurring different overheads. Wave Dynamic Differential Logic (WDDL) [7]–[11] was proposed especially as a countermeasure that can take standard cells and be incorporated to conventional ASIC design flow. However, our spice simulations show that DPA attacks on WDDL still leak secret key values to adversaries and furthermore incur significant overheads by doubling the area and energy consumed, since WDDL pairs a complementary cell with every cell in the original circuit. Standard cells are essentially weak against DPA attacks, and many inefficient complementary cells are paired in traditional WDDL. Also, no previous work has examined the selective insertion of complementary cells that simultaneously improves attack resistivity while lowering the area and energy overheads. Indeed the area and energy overheads could be crucial bottlenecks for implementation on smart cards and mobile devices that critically require low power and area.

In this paper, we propose ExCCel, a simulated annealing based method that automatically generates and explores combinations of complementary cells for reducing the power-consumption dependency and overheads so as to improve the DPA attack resistivity using standard cells. Experimental results on an AES S-Box circuit exemplar demonstrate that our complementary cell exploration yields higher DPA attack resistivity in many instances with significantly lower area (6.1% vs. 100.3%) and energy (2.1% vs. 93.4%) overheads compared with traditional WDDL. Thus ExCCel is a promising candidate for registering DPA attacks on smart cards and mobile devices.

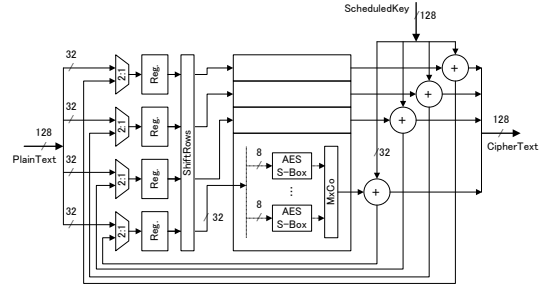


Fig. 1. AES Data Path

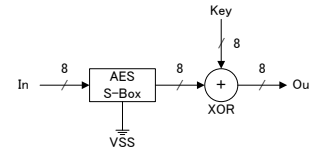


Fig. 2. DPA Attack Model on AES S-Box

II. DPA ATTACK MODEL ON AES S-BOX EXEMPLAR

We now illustrate the basic DPA attack procedure on an exemplar cryptographic module, the AES S-Box¹. Fig. 1 shows an AES data path [21]. PlainText is a 128-bit vector of arbitrary plain texts. CipherText is a 128-bit vector of cipher texts denoted as C . ScheduledKey is a 128-bit vector of scheduled key K that consists of 16 8-bit partial secret keys. A 128-bit AES uses 16 S-Boxes that dominate the power consumption of an entire AES circuit (approximately 75% [22]). Thus, adversaries are interested in collecting the power information of S-Boxes. In order to evaluate the DPA attack resistivity of the AES data path, an individual S-Box has to be evaluated. Fig. 2 shows a model of DPA attacks on a single AES S-Box [18], [23], [24]. In Fig. 2, Key represents the actual 8-bit partial secret key that adversaries attempt to reveal, and VSS is the point where electric current is measured.

In the DPA procedure, a selection function D is used, and Eq. (1) is the selection function [23] for the model depicted in Fig. 2; where $c \in C$ is an 8-bit cipher text, $K_{est.}$ is a estimated secret key, $k_{est.} \in K_{est.}$ is an 8-bit partial estimated secret key, and $Sbox^{-1}$ is the inverse AES S-Box function.

$$D(C, K_{est.}) = Sbox^{-1}(c \oplus k_{est.}) \quad (1)$$

Moreover, c is equivalent to Out of Fig. 2. Also, D returns the same value as In of Fig. 2 iff $k_{est.} = Key$.

The DPA procedure is conducted follows. Adversaries collect a number of cipher texts c and observe the corresponding VSS current of an AES S-Box. Note even if the VSS current includes noise from other circuits (e.g., MixCo, other S-Boxes

¹Although the AES S-Box is used as an exemplar, our approach is equally applicable to a wide range of other cryptographic hardware circuits.

in Fig. 1), DPA can statistically filter it out. These VSS current values are grouped into either G_0 or G_1 depending on the returned 8-bit values of D in Eq. (1). Let A_0 and A_1 be the average VSS current values of G_0 and G_1 , respectively. Since 8-bit $k_{est.}$ can be any integer between 0 and 255, there are $2^8 = 256$ possible sets of A_0 and A_1 . Amongst all possible $k_{est.}$ values, the one that makes a maximum differential power (denoted as DP) between A_0 and A_1 is assumed to be equal to the correct secret Key . Ideal countermeasures generate no differential power ($DP \approx 0$) so that adversaries are unable to reason $k_{est.} = Key$.

In this procedure, adversaries have to attempt 256 possibilities of $k_{est.}$ exhaustively for each S-Box. However, since a 128-bit AES uses 16 S-Boxes, the number of the attempts becomes a maximum of $2^8 \times 16$ that is much smaller than attempting 2^{128} possible values of 128-bit K .

The DP is calculated as follows.

$$0 \leq t < \text{clock period} \quad (2)$$

$$A_0(t) = \frac{1}{|G_0|} \sum_{D(C, K_{est.}) \in G_0} p_c(t) \quad (3)$$

$$A_1(t) = \frac{1}{|G_1|} \sum_{D(C, K_{est.}) \in G_1} p_c(t) \quad (4)$$

$$|G_0| + |G_1| = |C| \quad (5)$$

$$DP(D(C, K_{est.}), P) = \arg \max_t |A_0(t) - A_1(t)| \quad (6)$$

t is the time of sampling VSS current, and sampling rate should be faster than twice the clock speed so that multiple points of DP curves can be compared in Eq. (6). $p_c(t)$ is observed VSS current corresponding to cipher text c at time t , and P is the set of $p_c(t)$. G_0 and G_1 are the groups comprised of $p_c(t)$. The method of grouping $p_c(t)$ is arbitrary, but one of the common methods is referring to the i th bit of the value that D returns. For instance, $p_c(t)$ is grouped into

$$\left. \begin{array}{l} G_0 \text{ when } D[i] = 0 \\ G_1 \text{ when } D[i] = 1 \end{array} \right\} \quad (7)$$

Another method is referring to Hamming weight of the 8-bit value that D returns [25]. For instance, $p_c(t)$ is grouped into

$$\left. \begin{array}{l} G_0 \text{ when } 0 \leq \text{Hamming weight of } D < 4 \\ G_1 \text{ when } 4 < \text{Hamming weight of } D \leq 8 \end{array} \right\} \quad (8)$$

$|C|$ is the number of cipher text c . $|G_0|$ and $|G_1|$ are the number of the group members. DP is the function that returns a differential power value.

III. RELATED WORK

A. Countermeasures

Against DPA attacks, several countermeasures have been proposed [2]–[20]. Ambrose et al. [2] proposed a randomization technique of operation execution timing. Masking countermeasures [3]–[6], [18]–[20], [26] conceal intermediate variables with random or complementary values. These countermeasures suffer from performance degradation.

Another approach for mitigating DPA attack threats is minimization of power imbalance in a circuit. Sense Amplifier Based Logic (SABL) [13]–[15] and other transistor level countermeasures [12], [16], [17] can minimize the imbalance; it is, however, impractical to design a full-custom chip every time.

WDDL [7]–[11] and other dual-rail pre-charge logic styles [18], [19] were proposed as cell level countermeasures that can take standard cells and be incorporated into conventional ASIC design flow. WDDL compensates switching-probability-dependent power consumption by putting complementary cells next to the original circuit. For instance, AND cells are paired with OR cells. Thus, every WDDL cell is comprised of 2

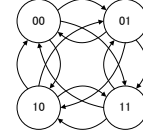


Fig. 3. Example of An Eulerian Graph When Input Vectors Are 2 Bits

standard cells resulting in a doubling of the area and power, making it unsuitable for area- and power-constrained mobile applications. Although standard cell libraries are often used in ASIC design flow, cryptographic modules composed of standard cells are weak against DPA attacks due to the power-consumption dependency in the standard cells on input values.

B. Current Observation Methods and Accuracy

Accuracy of the current observation is vital in evaluating DPA attack resistivity. The most accurate method performs actual measurements [5], [7], [20], [27], [28]; however, manufacturing prohibitively expensive test chips is necessary. Spice simulation [2], [29] is less accurate but still good enough to evaluate transistor level countermeasures. Note that Spice simulations do not take into account electrical noise in the real environment, which allows evaluations of countermeasures under the ideal condition for adversaries. Hamming weight/distance models [3], [26], [30]–[35] assume that power consumptions are proportional to either Hamming weight or Hamming distance. These methods are fast but not accurate enough to evaluate transistor nor cell level countermeasures.

IV. PRELIMINARY EVALUATION OF WDDL

We now discuss the limitations of WDDL based on Spice simulations.

A. Input Vectors

The power consumption of a CMOS circuit is dominated by dynamic power, and DPA attacks exploit it. The dynamic power applicable to DPA includes short-circuit power and subVt leakage [36], [37], but both of them have dependency on transistor switches; $\{0 \rightarrow 1\}$ and $\{1 \rightarrow 0\}$. In order to conduct a complete DPA attack resistivity evaluation, the dynamic power of all input vector bit transitions has to be observed.

A sequence of input vectors are fed into the I_n of Fig. 2. In practice, a smaller sequence of input vectors is preferable due to Spice simulation run time. The smallest sequence of input vectors that comprehend all combinations of input vector bit transitions is found in an Eulerian path. Fig. 3 shows an example of an Eulerian graph when input vectors are 2 bits. Each node has directed edges to the rest the nodes. The nodes are states of the input vectors, and they transit from one state to another following the edges. The Eulerian path that includes every edge on the graph is the minimum sequence of input vector transitions, and the path is easily found by a depth first search. The smallest number of n -bit input vector sequence is the number of nodes in an Eulerian path, which is

$$2^n \times (2^n - 1) + 1 \quad (9)$$

The result of Eq. (9) should be the same as that of Eq. (5) since the numbers of input and output vectors are the same. A sample sequence of input vectors for Fig. 3 is

$$00 \rightarrow 01 \rightarrow 00 \rightarrow 10 \rightarrow 00 \rightarrow 11 \rightarrow 01 \rightarrow 10 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 11 \rightarrow 00. \quad (10)$$

B. Preliminary Experiments

In order to evaluate the impact of WDDL, we conducted the following preliminary experiments. Fig. 4 (a) shows a XOR cell that has two inputs: x and y . Fig. 4 (b) shows a WDDL XOR that has four inputs: x , \bar{x} , y , and \bar{y} . \bar{x} and \bar{y} are negations of inputs x and y , respectively. The complementary cell of Fig. 4 (b) is XNOR. Both Fig. 4 (a) and (b) have VSS,

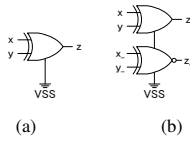


Fig. 4. (a) XOR Cell (XOR2X2) (b) WDDL XOR (XOR2X2 and XNOR2X2)

TABLE I
PRELIMINARY EXPERIMENTAL SETUP

Target Designs	XOR Cell & WDDL XOR [9]
Current Measurement Tool	Synopsys NanoSim C-2009.06
Plotting Tool	Synopsys CosmosScope C-2009.06-SP1
Standard Cell Library	SAED_EDK 90nm

and both the primary and complementary cell of Fig. 4 (b) share the same V_{SS} .

TABLE I shows the experimental setup. We deploy cells from the Synopsys 90nm generic library (SAED_EDK 90nm). The target hardware design files are described in Verilog gate-level netlists. NanoSim measures the V_{SS} current of each design, where the time resolution is 10ps. The V_{DD} is 1.2v.

Fig. 5 shows an experimental result of Fig. 4 (a). The top two charts are the voltages of x and y of Fig. 4 (a). These inputs follow the sequence of Eq. (10) at 2GHz. The bottom chart represents the V_{SS} current of the XOR cell in Fig. 4 (a). Fig. 5 shows that the V_{SS} current flows only when inputs are switching, which is the CMOS feature. Furthermore, the inputs, x and y , are asymmetric in terms of the V_{SS} current. For example, comparing the cases x and y are switching $\{x:0 \rightarrow 1, y:1 \rightarrow 0\}$ and $\{x:1 \rightarrow 0, y:0 \rightarrow 1\}$; the V_{SS} currents are not the same. Indeed, the V_{SS} current is uniquely dependent on input values. The difference of the V_{SS} current is small for a single cell, but an AES S-Box contains several hundred cells in general. An adversary can exploit accumulation of these differences to gather useful information for an attack.

Fig. 6 shows an experimental result of the WDDL XOR in Fig. 4 (b). The top four charts are the voltages of x , x_{-} , y , and y_{-} of Fig. 4 (b). These inputs follow the sequence of Eq. (10) at 2GHz as well. The bottom chart represents the V_{SS} current of Fig. 4 (b). As in Fig. 5, the inputs of the WDDL XOR are asymmetric in terms of the V_{SS} current in Fig. 6. Accordingly, the V_{SS} current of the WDDL XOR is still uniquely dependent on input values².

These simple experiments show that WDDL is in fact not efficient to mitigate the dependency on the input values despite the doubled area overhead. WDDL works well only on the assumption that the inputs of cells are symmetric in terms of the V_{SS} current. In order to achieve such symmetry, specially designed complementary cells are required, which

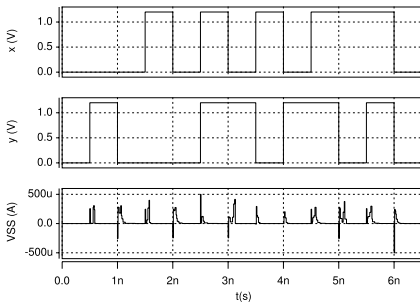


Fig. 5. V_{SS} Current of a XOR Cell

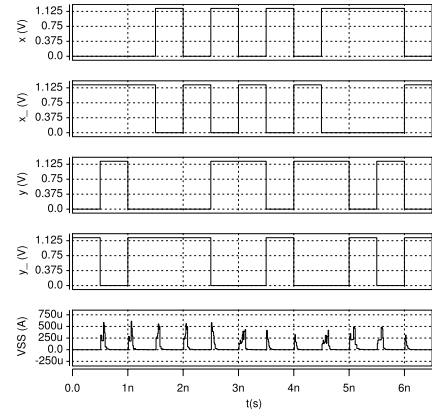


Fig. 6. V_{SS} Current of a WDDL XOR

are not available in most standard cell libraries. We overcome this problem by exploring combinations of complementary cells that reduce the power dependency and overheads while simultaneously improving DPA resistivity using standard cells.

V. COMPLEMENTARY CELLS EXPLORATION

Our exploration approach for reducing overheads and improving DPA attack resistivity uses simulated annealing to selectively add complementary cells instead of pairing a complementary cell with every cell in the original circuit. In fact, SAED_EDK 90nm has 147 logic cells³, and the number of possible combinations is 147^j , where j is the number of the complementary cells. Hence, it is computationally intractable to explore every combination, and a heuristic approach is mandatory to tackle this problem.

A. Exploration Method

1) Problem Formulation for Finding Complementary Cells That Improve The DPA Attack Resistivity:

- Objective: Minimize DP curve peaks (defined in Eq. (11))
- Input: AES S-Box (Verilog gate-level netlist)
- Output: Complementary cells (Verilog gate-level netlist)
- Constraints: Use standard cells
Use negations of original input wires

Our proposed exploration method uses simulated annealing and adds some randomly selected cells in parallel with an original circuit (AES S-Box), as shown in Fig. 7. At each iteration step of simulated annealing, one randomly selected cell from the standard cell library is added to “Comp. Cells”(Fig. 7), which represents the complementary cells. Every time a cell is added, the V_{SS} current of the combined design (original circuit and added complementary cells) is measured using NanoSim. After that, the evaluation function (defined later) evaluates the DP curves of the combined design, and the decisions whether or not to accept the design are made depending on the temperatures. Until the program is terminated, it attempts and evaluates as many different combinations of complementary cells as possible, and the best design is saved. Note that we do not modify the original circuit since it is already synthesized.

As Fig. 7 shows, complementary cells use only In_{-} (negations of In) for the following reasons. Firstly, D flip-flop generally has two outputs: Q and QN . QN (negation of Q) is connected to In_{-} , and QN is not used in an AES data path (Fig. 1). Secondly, since complementary cells are isolated from any cells in the original circuit, there is no delay overhead.

²We used XOR cells, but we observed similar results for other cells.

³Flip-flops and analog cells are not included in this number

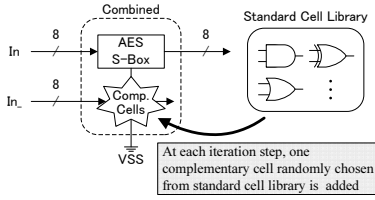


Fig. 7. Proposed Exploration Method

Furthermore, the output wires of “Comp. Cells” in Fig. 7 are floating so that the output values are not propagated into the next clock cycle.

2) *Evaluation Function*: The proposed exploration method attempts to minimize

$$\sum_{i=0,4} DP_i + DP_{hw} - DP_{ave}. \quad (11)$$

Eq. (11) is a sum of DP curve peaks; where DP_i is Eq. (6) with the grouping in Eq. (7), and DP_{hw} is Eq. (6) with the grouping in (8). In Eq. (11), only two i th bits are selected for DP_i since a evaluation function with more than two did not cause better results in our experiments. In addition, DP_{ave} is an average differential power when $Key \neq k_{est}$. DP_{ave} is subtracted in Eq. (11) since the smaller difference between $(DP_i + DP_{hw})$ and DP_{ave} , makes DPA more difficult.

In each iteration step of the exploration procedure, VSS current is measured and evaluated by Eq. (11) using 65281 input vectors according to Eq. (9). We do not need more than this number of VSS current traces because Spice simulations do not take into account electrical noise and return the same result for the same input vectors.

3) *Temperatures*: In our simulated annealing formulation, two temperatures are used, and both of them are defined as

$$temperature(s) = \frac{1}{(1 + e^{-0.8(7-s)})} \quad (12)$$

Eq. (12) is a Sigmoid function, where s represents iteration steps. Fig. 8 shows that the temperature is cooled gradually when s is small and drops dramatically at some point. Moreover, it is easy to modify the inflection point and inclination of the curve by substituting the coefficients. In Eq. (12), the temperature becomes 0.5 (acceptance rate is 50%) when $s = 7$.

4) Pseudo Code of ExCCel:

```

BEGIN ExCCel
1: mdl = Original Circuit {INITIAL DESIGN}
2: evaluation_best = mdl.evaluate {INITIAL EVALUATION}
3: s_outer = 0
4: loop {OUTER}
5:   mdls = next_mdls = [mdl, mdl, mdl, mdl]
6:   loop {INNER}
7:     s_inner = 0
8:     while rand < temperature(s_inner) do
9:       tmp_mdl = mdls.select_worst.add_rand_cell
10:      if next_mdls.select_worst.evaluate > tmp_mdl.evaluate then
11:        next_mdls.push(tmp_mdl).discard_worst
12:        s_inner += 1
13:      else
14:        s_inner += 1
15:      end if
16:    end while
17:    localbest_mdl = mdls.pop_worst {POP THE LOCAL BEST DESIGN}
18:    if mdls.empty? then
19:      mdls = next_mdls
20:      if next_mdls.select_best == localbest_mdl then
21:        break
22:      end if
23:    end if
24:  end loop {INNER}
25:  if evaluation_best > localbest_mdl.evaluate then

```

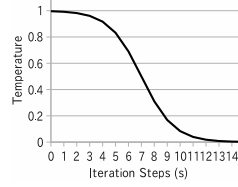


Fig. 8. Sigmoid Function (Eq. (12))

```

26:   evaluation_best = localbest_mdl.evaluate
27:   localbest_mdl.save {THE GLOBAL BEST DESIGN IS UPDATED}
28:   s_outer += 1
29:   else if rand < temperature(s_outer) then
30:     s_outer += 1
31:   else
32:     break
33:   end if
34: end loop {OUTER}
END ExCCel

```

In the above pseudo code, Lines 1 and 2 copy the original circuit to `mdl` and evaluate it with the `evaluate` function in Eq. (11), and this design becomes the initial point of the exploration. Our exploration method uses two loops: inner and outer. s_{inner} and s_{outer} (initialized at Lines 7 and 3) counts the step of the inner and outer loop, respectively. The outer loop (Lines 4–34) compares the local best designs generated by the inner loop (Lines 6–24) and select the global best design among them based on the evaluation (Eq. (11)). In order to generate the local best designs, four copies of the `mdl` are stored in `mdls` (Line 5), one randomly selected cell from standard cell library is added to each of the `mdls` (Line 9), and the added `mdl` is stored into `next_mdls` (Line 11) if the added `mdl` is better than one of the four `mdls` in `next_mdls`. If the better `mdl` is (or is not) found, s_{inner} is incremented (or decremented) (Lines 12 and 14), which is used for the inner temperature (Eq. (12)). The temperature decides whether or not to continue the process of adding cells (Line 8). Subsequently, the worst design of `mdls` are popped (Line 17). When `mdls` becomes empty, the `next_mdls` is copied to the `mdls` (Line 19) so that another cell will be added to the `mdls` in the next inner loop iteration. When the evaluations of the `mdls` stop improving, the inner loop is exited (Lines 20–22). The rest of the outer loop compares the local best designs that that are lastly popped in the inner loop (Line 17). If a local best design is better than the preceding global best design, it is replaced by the new local best design (Lines 25–27). If the better global design is (or is not) found, s_{outer} is incremented (or decremented) (Lines 28 and 30), which is used for the outer temperature (Eq. (12)). The temperature decides whether to exit the loop when the global best design evaluation stops improving (Lines 29–33). Since the proposed exploration adds cells one by one, the number of complementary cells becomes small, approximately 1~20 cells.

B. Composite Field AES S-Box

Amongst the variety of AES S-Box implementations [21], we chose the composite field AES S-Box proposed by Satoh et al. [38] for our experiments. The composite field AES S-Box is the smallest design as far as we know, and thus allows us to save on Spice simulation run times. Of course, we believe the same method can be also applied to other types of S-Box implementations as well. TABLE II shows the standard cells used for synthesizing the composite field AES S-Box (without countermeasures). In TABLE II, the number preceding “X” represents the number of inputs, and the number after “X” represents drive strength (e.g., XOR2X1 represents an XOR gate with 2 inputs and 1 drive strength).

C. Circuit Partitioning for Exploration-Time Reduction

In our exploration method, each iteration step requires a Spice simulation result. Since the exploration conducts thousands of iteration steps, it is impractical to simulate the whole circuit each time. In order to terminate the exploration in a manageable amount of time, we applied the divide-and-conquer approach. Since original circuit and complementary cells do not share the same input wires, the complementary cells will not affect the behavior of the original circuit, and the VSS current of the original circuit and the complementary cells

can be measured individually. The V_{SS} current of the original circuit needs to be measured only once at the beginning of the exploration because the proposed method changes only the combinations of the complementary cells. Hence, the part that has to be re-measured in each iteration step is only the complementary cells. The area of the complementary cells tends to be smaller than the original circuit so that the Spice simulation run time at each iteration step becomes shorter. Indeed, the one-time NanoSim simulation run time of the AES S-Box (TABLE II), which uses 65281 8-bit input vectors, is reduced from 4 hours to 2~3 minutes by applying the partitioning. Thus, the proposed exploration method is still feasible to run, and the whole procedure with the partitioning terminates within 3 days approximately. Note that the partitioning is used only in the proposed exploration procedure but not for the DPA attack results shown in the next section.

VI. EXPERIMENTAL RESULTS

A. Experimental Setup

TABLE III summarizes the experimental setup for evaluating the DPA attack resistivity of S-Boxes. The composite field AES S-Box is used for the 3 target designs: “Normal” without countermeasures (the same design used for TABLE II), “WDDL” pairing each cell with a complementary cells, and “ExCCel” using the explored complementary cells. All of them are described in Verilog gate-level netlists. NanoSim measures the V_{SS} current, where the time resolution is 10ps. Since the input vectors arrive in every 5ns, the number of points for t is $5ns/10ps=500$ (see Eqs. (2) and (6)). Default settings are used for other NanoSim accuracy parameters. The model of DPA attacks on AES S-Boxes is depicted in Fig. 2, and Key is fixed at **AE**.

B. Comparison in DPA Attack Resistivity

Figs. 9, 10, and 11 show the results of DPA attacks on Normal, WDDL, and ExCCel S-Boxes, respectively. The horizontal axes represent the 8-bit estimated secret keys ($k_{est.}$), and the vertical axes are the differential power (DP) in Eq. (6) with the grouping in Eqs. (7) or (8). In Fig. 9 with Eq. (8), there is a sharp peak when $Key = k_{est.}$. In contrast, in Fig. 10 with Eq. (7), the 5th peak of $k_{est.}$ is equal to Key . The rank

TABLE II
STANDARD CELLS USED FOR SYNTHESIZING A COMPOSITE FIELD AES S-BOX (WITHOUT COUNTERMEASURES)

Cell	Count
AND2X2	1
AOINX1	20
ISOLANDX1	4
NAND2X0	26
NOR2X0	3
OA21X1	2
OA22X1	1
XNOR2X1	26
XNOR3X1	12
XOR2X1	35
XOR3X1	13
Total	143

TABLE III
EXPERIMENTAL SETUP

Target Designs	Composite Field AES S-Boxes [38] (Normal, WDDL [9], ExCCel)
Synthesis Tool	Synopsys Design Compiler C-2009.06 (Delay Constrain is set at 5ns)
Current Measurement Tool	Synopsys NanoSim C-2009.06
Standard Cell Library	SAED_EDK 90nm
Program Language	Ruby 1.9.2dev [39] + NArray-0.5.9p7 [40]
OS	CentOS release 5.4
CPU	Dual Core Opteron 2.4 GHz x2
Memory	2GB

TABLE IV
RANKS OF CORRECT Key AMONGST ALL $k_{est.}$

	Eq. (7)								Eq. (8)
	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	
Normal	2	1	1	1	1	1	1	1	1
WDDL	1	1	102	7	1	2	1	2	5
ExCCel	98	2	2	76	45	1	1	1	29

TABLE V
COMPLEMENTARY CELLS USED FOR (a) WDDL AND (b) EXCCEL

(a)		(b)	
Cell	Count	Cell	Count
OR2X2	1	INVX8	1
OAINX1	20	XOR3X1	1
ISOLORX1	4	AND3X4	1
NOR2X0	26	AO21X2	1
NAND2X0	3	XNOR2X2	1
AO21X1	2	OAI21X2	1
AO22X1	1	OAI22X2	1
XNAND2X1	26		
XNAND3X1	12		
XNOR2X1	35		
XNOR3X1	13		
Total	143	Total	7

of the correct Key peak gives the adversaries an order of $k_{est.}$ they should attempt. Therefore, the larger number of the rank, the higher DPA attack resistivity. In Fig. 11 with Eq. (8), the 29th peak of $k_{est.}$ is equal to Key , which demonstrates that ExCCel has even higher DPA attack resistivity than WDDL for this grouping.

TABLE IV summarizes only the ranks of the correct Key amongst all $k_{est.}$. As TABLE IV shows, ExCCel has higher DPA resistivity than Normal and WDDL in many cases. In some cases ($i=2,5,7$ with Eq. (7)), WDDL has higher resistivity in TABLE IV; however, note that WDDL incurs significant overheads by doubling the area and energy consumed, which are discussed in the next subsection.

C. Comparison in Overheads

1) *Area Overhead*: TABLE V shows the complementary cells used for WDDL and ExCCel. Also, Fig. 12 (a) is the comparison in area, and it shows that ExCCel requires 6.1% additional area while WDDL requires 100.3%.

2) *Energy Overhead*: Fig. 12 (b) is the comparison in average energy consumption, and it shows that ExCCel requires 2.1% additional energy while WDDL requires 93.4%.

In summary, our experimental results demonstrate that ExCCel is able to generate significantly more efficient DPA-attack-resistant hardware in area and energy overheads compared with WDDL. Note that the delay overhead of ExCCel is zero since the ExCCel does not modify the original circuit.

VII. CONCLUSIONS

This paper proposed ExCCel, a simulated annealing based method that automatically generates and explores combinations of complementary cells to improve DPA attack resistivity with small area and energy overheads. Using the AES S-Box circuit as an exemplar, we observed that ExCCel yielded DPA-attack-resistant hardware with 6.1% and 2.1% additional area and energy while WDDL incurs 100.3% and 93.4%, respectively. We believe ExCCel is a promising approach for achieving efficient DPA attack resistivity in area- and power-constrained applications (e.g., smart cards). ExCCel also gives circuit designers the ability to explore and evaluate tradeoffs between security and hardware overheads for different applications. There are several directions for future work, including; intelligent selection of cells during simulated annealing; physical realization of ExCCel on FPGA platforms for prototyping; and techniques to speed up the exploration procedure.

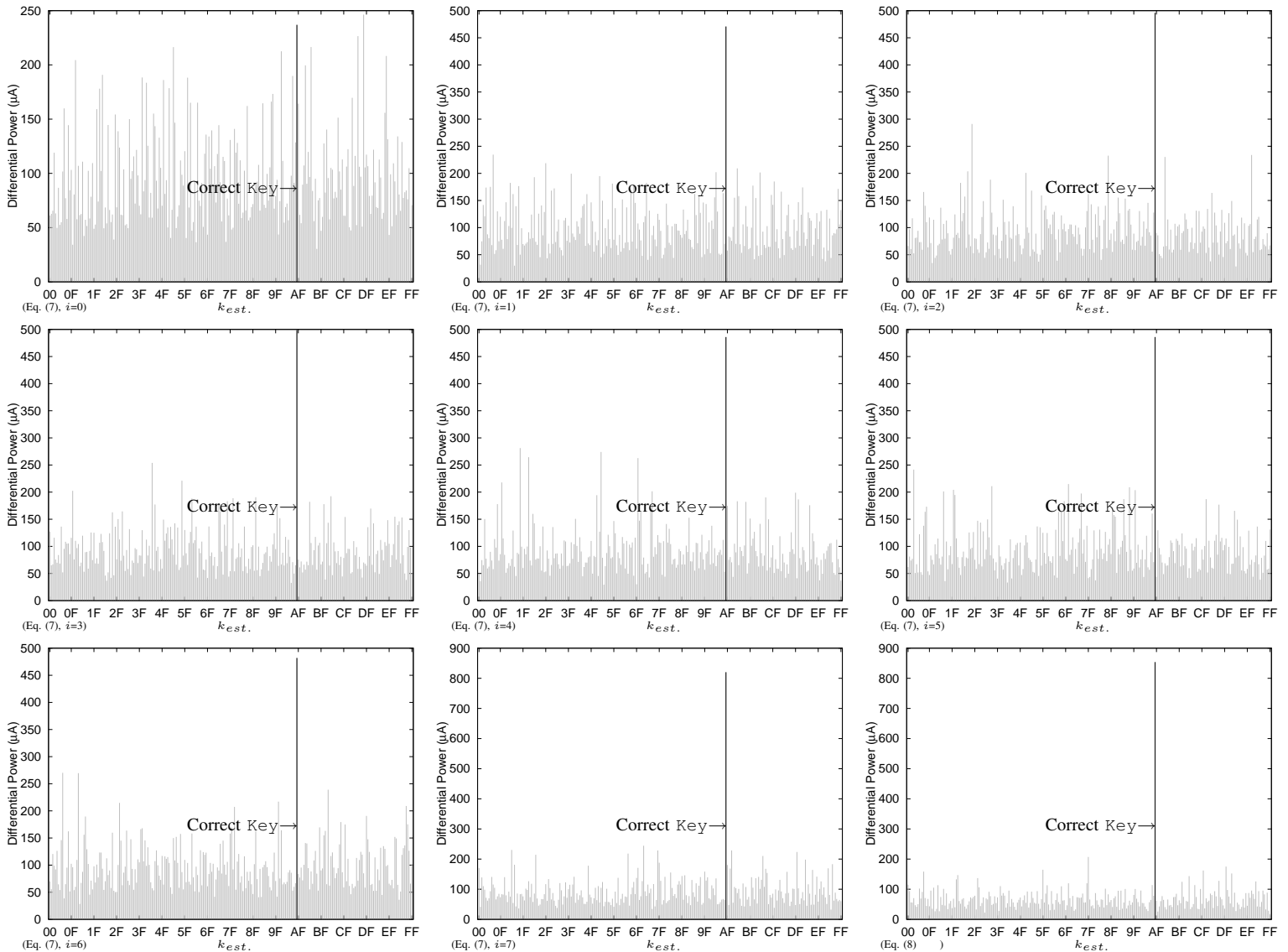


Fig. 9. Result of DPA Attack on Normal S-Box ($\text{Key}=\text{AE}$)

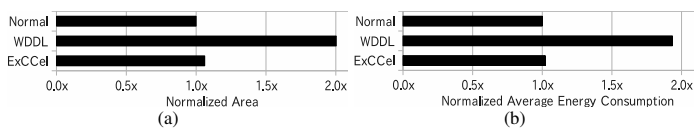


Fig. 12. Comparison in (a) Area and (b) Average Energy Consumption (Normalized by Normal S-Box)

ACKNOWLEDGMENT

The authors thank Dr. Morioka, NEC Lab. for his courtesy in providing the composite field AES S-Box Verilog code. This work is supported by the Yoshida Scholarship Foundation.

REFERENCES

- [1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. CRYPTO*, 1999, pp. 388–397.
- [2] J. Ambrose, R. Ragel, and S. Parameswaran, "Rijid: Random code injection to mask power analysis based side channel attacks," in *Proc. DAC*, Jun. 2007, pp. 489–492.
- [3] J. D. Golic and C. Tymen, "Multiplicative masking and power analysis of AES," in *Proc. CHES*, 2003, pp. 198–212.
- [4] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proc. CHES*, 2001, pp. 309–318.
- [5] C. Gebotys, "A table masking countermeasure for low-energy secure embedded systems," *IEEE Trans. VLSI Syst.*, vol. 14, no. 7, pp. 740–753, Jul. 2006.
- [6] J.-S. Coron and L. Goubin, "On boolean and arithmetic masking against differential power analysis," in *Proc. CHES*, 2000, pp. 231–237.
- [7] K. Tiri, D. Hwang, A. Hodjat, B. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "A side-channel leakage free coprocessor IC in 0.18µm CMOS for embedded AES-based cryptographic and biometric processing," in *Proc. DAC*, Jun. 2005, pp. 222–227.
- [8] —, "Prototype ic with wddl and differential routing - DPA resistance assessment," in *Proc. CHES*, 2005, pp. 354–365.
- [9] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proc. DATE*, vol. 1, Feb. 2004, pp. 246–251 Vol.1.
- [10] —, "A digital design flow for secure integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 7, pp. 1197–1208, Jul. 2006.
- [11] —, "A vlsi design flow for secure side-channel attack resistant ICs,"

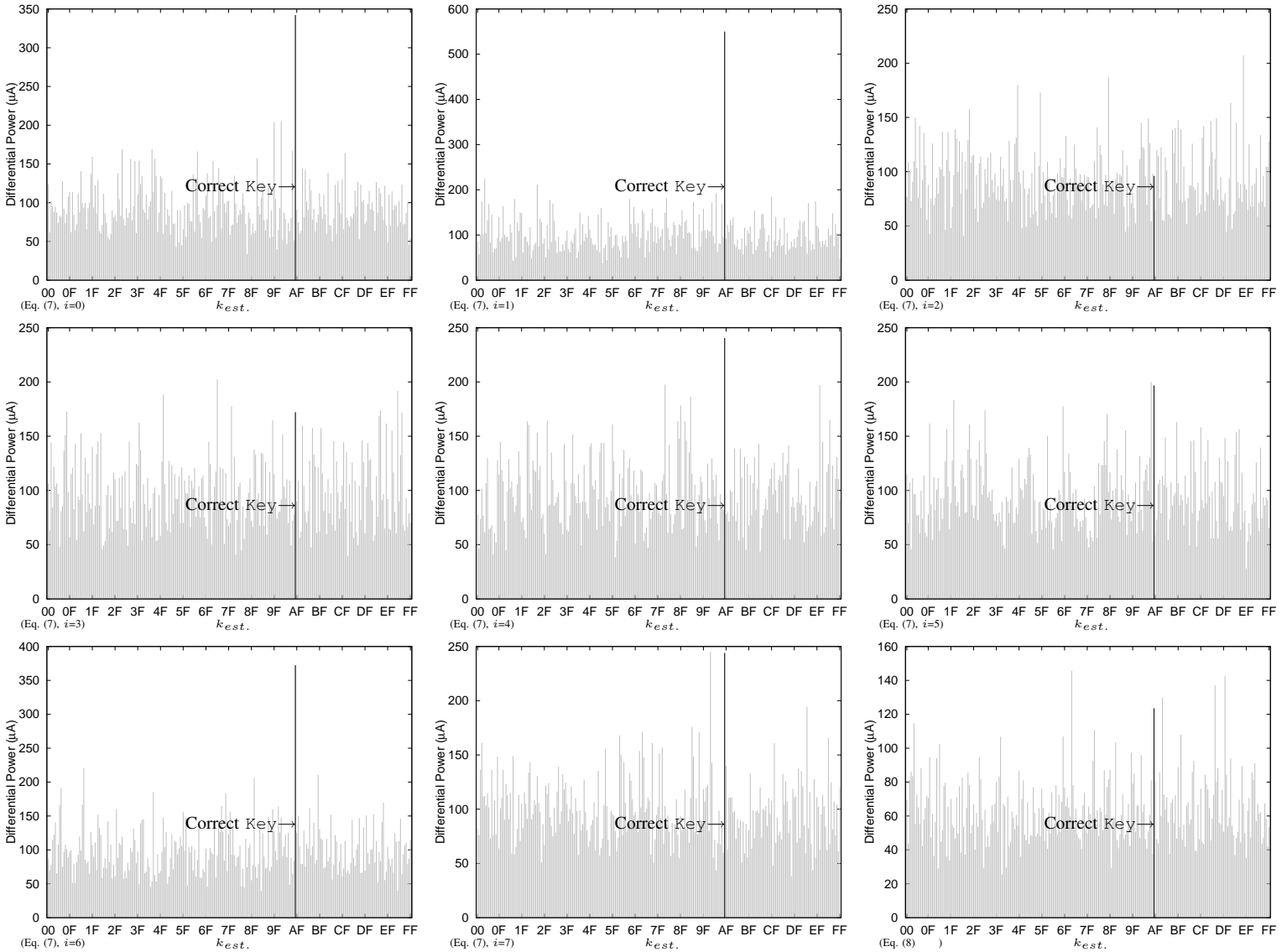


Fig. 10. Result of DPA Attack on WDDL S-Box (Key=AE)

in *Proc. DATE*, vol. 3, Mar. 2005, pp. 58–63.

- [12] —, “Design method for constant power consumption of differential logic circuits,” in *Proc. DATE*, 2005, pp. 628–633.
- [13] —, “Securing encryption algorithms against DPA at the logic level: Next generation smart card technology,” in *Proc. CHES*, 2003, pp. 125–136.
- [14] —, “Charge recycling sense amplifier based logic: securing low power security ICs against DPA,” in *Proc. ESSCIRC*, Sept. 2004, pp. 179–182.
- [15] K. Tiri, M. Akmal, and I. Verbauwhede, “A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards,” in *Proc. ESSCIRC*, Sept. 2002, pp. 403–406.
- [16] M. Khatir, A. Moradi, A. Ejlali, M. T. M. Shalmani, and M. Salmasizadeh, “A secure and low-energy logic style using charge recovery approach,” in *Proc. ISLPED*, 2008, pp. 259–264.
- [17] S. Guilley, P. Hoogvorst, Y. Mathieu, R. Pacalet, and J. Provost, “CMOS structures suitable for secured hardware,” in *Proc. DATE*, 2004, pp. 1414–1415.
- [18] P. Schaumont and K. Tiri, “Masking and dual-rail logic don’t add up,” in *Proc. CHES*, 2007, pp. 95–106.
- [19] T. Popp and S. Mangard, “Masked dual-rail pre-charge logic: DPA-resistance without routing constraints,” in *Proc. CHES*, 2005, pp. 172–186.
- [20] M. Saeki, D. Suzuki, K. Shimizu, and A. Satoh, “A design methodology for a DPA-resistant cryptographic LSI with RSL techniques,” in *Proc. CHES*, 2009, pp. 189–204.
- [21] T. Sugawara, N. Homma, T. Aoki, and A. Satoh, “Differential power analysis of AES ASIC implementations with various S-box circuits,” in *Proc. ECCTD*, Aug. 2009, pp. 395–398.
- [22] S. Morioka and A. Satoh, “An optimized S-Box circuit architecture for low power AES design,” in *Proc. CHES*, 2002, pp. 172–186.
- [23] P. Yu and P. Schaumont, “Secure FPGA circuits using controlled placement and routing,” in *Proc. CODES+ISSS*, 2007, pp. 45–50.
- [24] A. K. Zadeh and C. H. Gebotys, “Side channel aware leakage management in nanoscale cryptosystem-on-chip (coc),” in *Proc. ISQED*, 2009, pp. 230–235.
- [25] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Investigations of power analysis attacks on smartcards,” in *Proc. USENIX Workshop on Smartcard Technology*, 1999, p. 17.
- [26] F.-X. Standaert, E. Peeters, and J.-J. Quisquater, “On the masking countermeasure and higher-order power analysis attacks,” in *Proc. ITCC*, vol. 1, Apr. 2005, pp. 562–567.
- [27] S. Guilley, L. Sauvage, P. Hoogvorst, R. Pacalet, G. Bertoni, and S. Chaudhuri, “Security evaluation of WDDL and SecLib countermea-

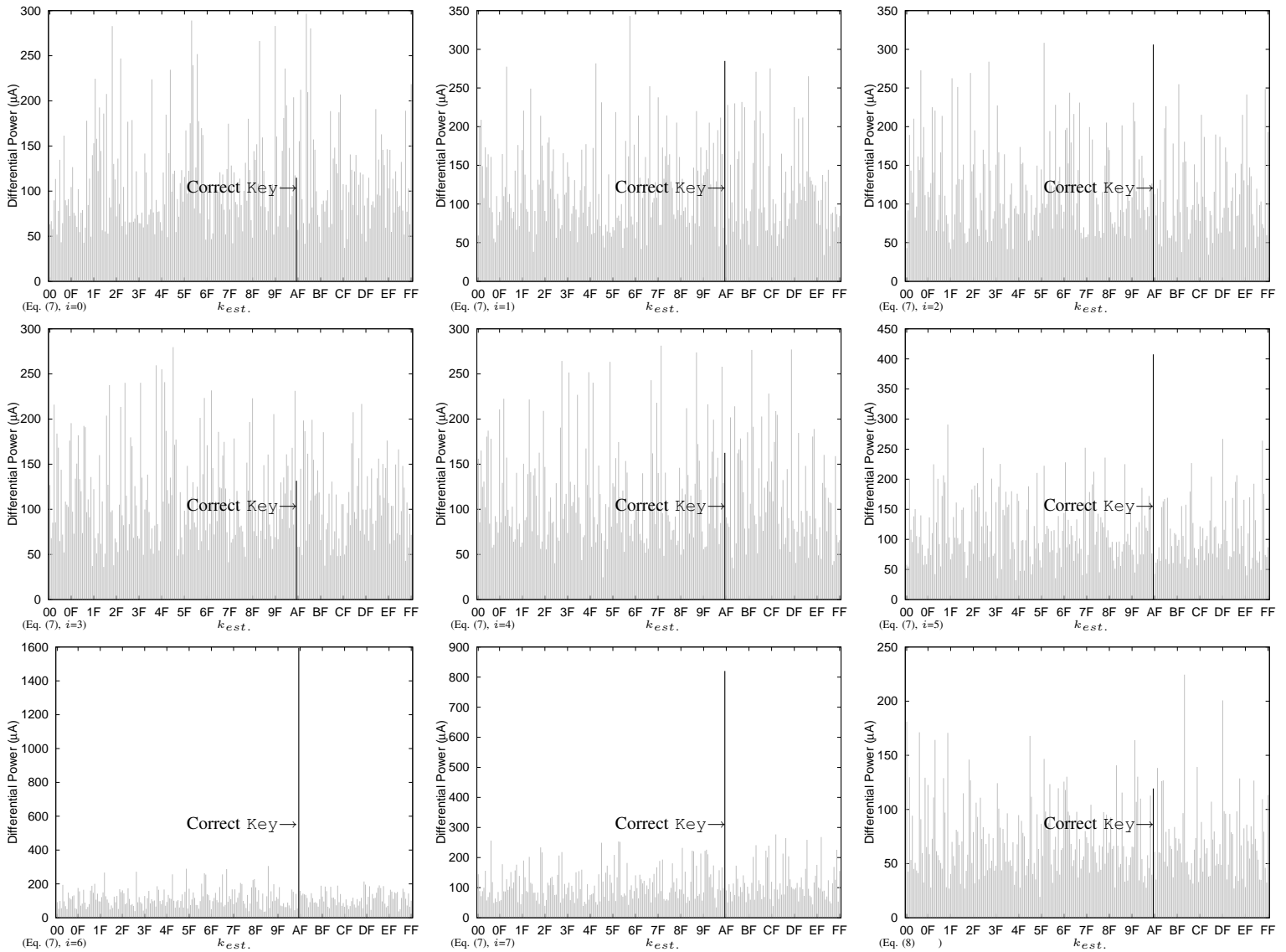


Fig. 11. Result of DPA Attack on ExCCel S-Box (Key=AE)

asures against power attacks,” *IEEE Trans. Comput.*, vol. 57, no. 11, pp. 1482–1497, Nov. 2008.

- [28] D. Suzuki and M. Saeki, “Security evaluation of DPA countermeasures using dual-rail pre-charge logic style,” in *Proc. CHES*, 2006, pp. 255–269.
- [29] F. Regazzoni, S. Badel, T. Eisenbarth, J. Grobschadl, A. Poschmann, Z. Toprak, M. Macchetti, L. Pozzi, C. Paar, Y. Leblebici, and P. Jenne, “A simulation-based methodology for evaluating the DPA-resistance of cryptographic functional units with application to CMOS and MCML technologies,” in *Proc. SAMOS*, Jul. 2007, pp. 209–214.
- [30] A. Sasaki and K. Abe, “Algorithm-level evaluation of DPA resistance to cryptosystems,” *Electrical Engineering in Japan*, vol. 165, no. 3, pp. 37–45, 2008.
- [31] K. Tiri and I. Verbauwhede, “Simulation models for side-channel information leaks,” in *Proc. DAC 2005*, Jun. 2005, pp. 228–233.
- [32] C. Clavier, J.-S. Coron, and N. Dabbous, “Differential power analysis in the presence of hardware countermeasures,” in *Proc. CHES*, 2000, pp. 252–263.
- [33] B. Eric, C. Christophe, and O. Francis, “Correlation power analysis with a leakage model,” in *Proc. CHES*, 2004, pp. 16–29.
- [34] S. Werner, L. Kerstin, and P. Christof, “A stochastic model for differential side channel cryptanalysis,” in *Proc. CHES*, 2005, pp. 30–46.
- [35] T. S. Messerges, “Using second-order power analysis to attack DPA resistant software,” in *Proc. CHES*, 2000, pp. 238–251.
- [36] J. Giorgetti, G. Scotti, A. Simonetti, and A. Trifiletti, “Analysis of data dependence of leakage current in CMOS cryptographic hardware,” in *Proc. GLVLSI*, 2007, pp. 78–83.
- [37] L. Lin and W. Burleson, “Leakage-based differential power analysis (LDPA) on sub-90nm CMOS cryptosystems,” in *Proc. ISCAS*, 2008, pp. 252–255.
- [38] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A compact rijndael hardware architecture with S-Box optimization,” in *Proc. ASIACRYPT*, 2001, pp. 239–254.
- [39] Ruby programming language. [Online]. Available: <http://www.ruby-lang.org/>
- [40] Numerical ruby narray. [Online]. Available: <http://narray.rubyforge.org/>