Analyzing and Optimizing the Dummy Rounds Scheme

Stanislav Jeřábek, Jan Schmidt
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
Email: {jerabst1, schmidt}@fit.cvut.cz

***Abstract*—The dummy rounds protection scheme, intended to offer resistance against Side Channel Attacks to Feistel and SP ciphers, has been introduced in earlier work. Its experimental evaluation revealed weaknesses, most notably in the first and last round. In this contribution, we show that the situation can be greatly improved by controlling the transition probabilities in the state space of the algorithm. We derived necessary and sufficient conditions for the round execution probabilities to be uniform and hence the minimum possible. The optimum trajectories over the state space are regular and easy to implement.**

***Index Terms*—Hiding, Markov Chain, FPGA, DPA, hiding in time, dummy rounds.**

1. Introduction

In [1], we proposed a scheme to make hardware implementations of Feistel Networks [2] and Substitution-Permutation Networks [3] more resistant against Side Channel Attacks (SCA) such as Differential Power Analysis (DPA) [4] [5]. The technique used, namely *Dummy Rounds* appeared before in software implementation [6] and in specialized cryptoprocessors [7]. It also has resemblances with some other software countermeasures, such as Dummy Cycles [8], Random Order Execution [9], or Shuffling [10].

The Dummy Rounds scheme employs the fact that the cipher networks consists of similar *rounds*. It further assumes that the implementing hardware can execute $M > 1$ rounds in a clock cycle.

In each clock cycle, all the M rounds are cascaded. The controller chooses a random number $\mu, m \leq \mu \leq M$, where the minimum m is another architectural constant. The result of the first μ rounds is used as the result of that clock cycle. These rounds are the *active* rounds. The results from the rest of the rounds (the *redundant* rounds) are discarded, see Figure 1.

The randomness of the execution is supposed to hide the real computation from an attacker. To prevent redundant rounds from leaking data, they process random data rather than the real data from preceding rounds.

The choice of μ is limited in certain states of the algorithm. It may need to execute all active rounds in a given number of clock cycles, or there can be lack of unexecuted active rounds with respect to the minimum m.

Experimental evaluation on the PRESENT cipher [11] in [1] did not give approving results. After refining the accuracy measuring process, much better and almost satisfactory results were obtained. The biggest weakness was in the first clock cycle.

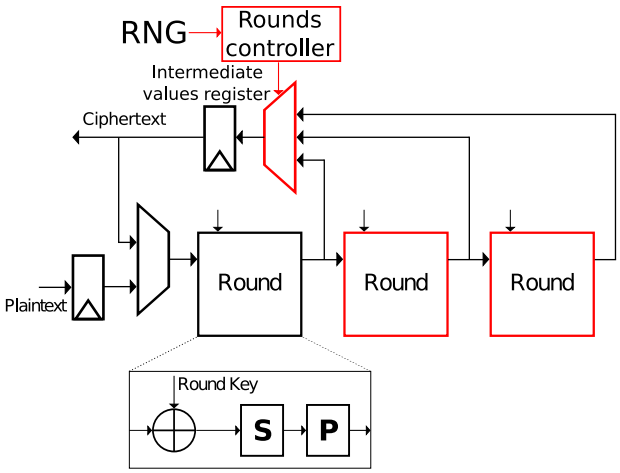

Figure 1. Dummy cycles countermeasure scheme [1], simplified.

Assume a cipher with C rounds, implemented in N clock cycles. Further, let r_n be the number of rounds accepted up to the step n, $n \leq N$. Then, the state space of the algorithm is delimited by the following inequalities:

$$r_n \leq Mn \tag{1}$$

$$r_n \geq mn \tag{2}$$

$$r_n + m(N - n) \leq C \tag{3}$$

$$r_n + M(N - n) \geq C \tag{4}$$

An example of the state space resulting from $m = 1$, $M = 3$, $C = 32$, $N = 16$ is in Figure 2. This is the state space of the tested PRESENT implementation. We can see why clock cycle 1 is a problem. Due to $m = 1$, the first clock cycle *must* execute the first round as active, and the last clock cycle *must* execute the 32nd round as active. In the case of PRESENT, those are the rounds that leak most information [12].

The states of the algorithm, together with transition probabilities, form a Markov chain. Using the state probabilities, we can calculate the probability that the round r was executed as active in a given state. In general, these probabilities vary with clock cycle number for any given round number. The clock cycle with the maximum round execution probability offers the best point for an attack on

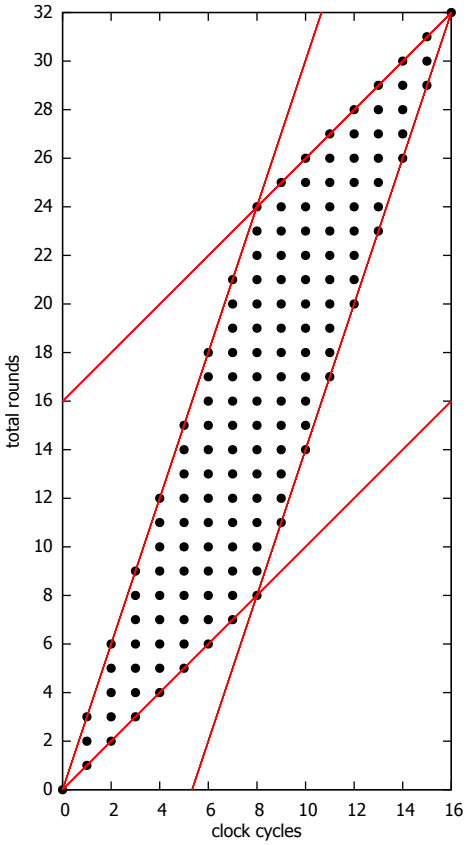

Figure 2. A state space for $m = 1$, $M = 3$, $C = 32$, $N = 16$. The lines represent Equations 1 to 4.

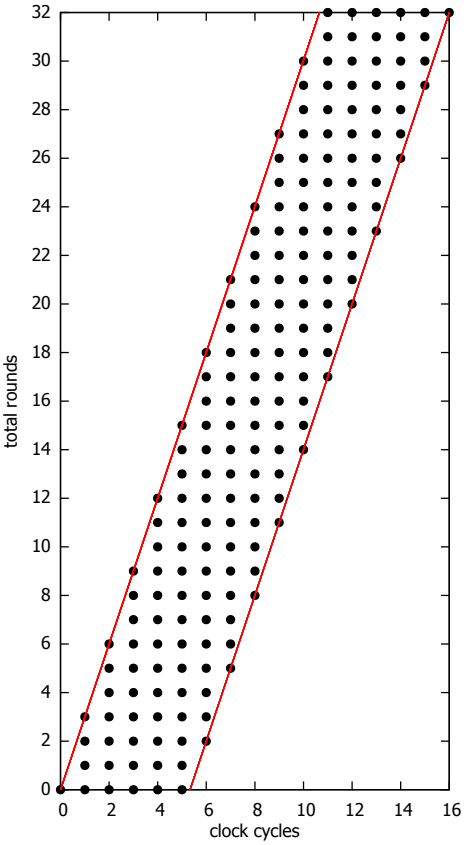

Figure 3. A state space for $m = 0$, $M = 3$, $C = 32$, $N = 16$. The lines represent Equations 1 and 3.

the given round. The gist of this contribution is to *design* the transition probabilities so that the probability of round execution remains the minimum possible over the entire computation.

First, we discuss the case $m > 0$ in Section 2. In Section 3, we develop a finer model than the state-level Markov chain presented above. In that model, we are able to derive state probabilities and round execution probabilities as described in Section 4. Using the results, we design the optimum transition probabilities in Section 5. The measure of protection can be tied directly to architectural parameters, namely the work effort investment, as discussed in Section 6.

2. Architectural parameters

The problem with the first and last rounds follows directly from the fact that $m > 0$. There is no freedom and no randomness in the first and last clock cycle. Therefore, we have to fix $m = 0$ in all cases. The modified state space, already presented in [1], is in Figure 3.

As a remedy to the leak in clock cycle 0, the original proposal suggest to randomly postpone the beginning of the computation. This is precisely what can happen with $m = 0$: there can be a random number of redundant rounds at the beginning, and then some active rounds can occur. Therefore, any scheme with $m = 0$ fulfills this request as a special case.

3. A slot-level model and round control

In the above mentioned Markov chain, the transitions have a regular structure. Let $S_{n,r}$ be the state that has executed rounds $1 \dots r$ in the clock cycle n. From this state, transitions to states $S_{n+1,r+m}, \dots S_{n+1,r+M}$ are possible.

In the original proposal, M rounds are executed serially, and a random output is chosen. We have to suppose that the attacker can distinguish the execution of a particular round. Then, instead of N clock cycles, we model $K = MN+1$ *slots*. Then, a yet simpler (but larger) model can be constructed.

Let $S_{k,r}$ be the state that has executed rounds $1 \dots r$ in the slot k. From this state, only two transitions are possible. Either, the next round will be taken as active, which leads to the state $S_{k+1,r+1}$. Or, the round is redundant, which transits to the state $S_{k+1,r}$. An example is in Figure 4.

This model is more general than the round control in the original proposal. That controller takes $m \dots M$ active rounds, and the rest is discarded, so that only thick lines in Figure 4 can be followed. Practically at no hardware cost, we can obtain finer control, more random operation and simpler analysis.

The model is still a Markov chain, thanks to $m = 0$. Without this restriction, it would lose the Markov property.

We made the following formal step to simplify the expressions describing the model (esp. their indices). With $m = 0$, the accessible part of the state space is always

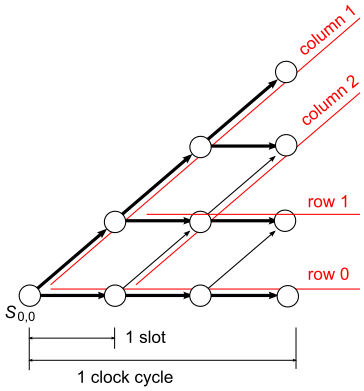

Figure 4. A part of a slot-level model with $m = 0$, $M = 3$

a rhomboid. Therefore, we use an alternate coordinate system of rows and columns along the sides of the rhomboid (Figure 4). As in the previous model, the row r corresponds to the completed sequence $1 \dots r$.

The width W of the rhomboid is

$$W = MN - C - 1 \tag{5}$$

Whereas the architectural parameter M expresses the overhead in hardware, W captures the overall relative overhead in work effort, and, thus, in energy consumption. The unprotected computation has, of course, $M = 1$ and $W = 1$.

4. Probabilities analysis

In the above described model, let

- $s_{c,r}$ be the probability of the state $S_{c,r}$ in column c and row r,
- $p_{c,r}$ be the probability, that the next round will be active in the state $S_{c,r}$,
- $\rho_{c,r}$ be the probability, that the model will arrive at $S_{c,r}$ by executing the round r.

Then, the correctness of the computation requires that

$$p(W, r) = 1, r = 0 \dots C - 1 \tag{6}$$

and

$$p(c, C) = 0, c = 1 \dots W \tag{7}$$

The initial state has probability 1, that is,

$$s_{1,0} = 1 \tag{8}$$

For chosen probabilities $p_{c,r}, c = 1 \dots W - 1, r = 0 \dots C - 1$, state and round execution probabilities can be calculated as

$$s_{c,0} = s_{c-1,0}(1 - p_{c-1}), c = 1 \dots W \tag{9}$$

$$s_{c,r} = s_{c,r-1}p_{c,r-1} + s_{c-1,0}(1 - p_{c-1}), c = 1 \dots W, r = 1 \dots C \tag{10}$$

$$\rho_{c,r} = s_{c,r-1}p_{c,r-1}, c = 1 \dots W, r = 1 \dots C \tag{11}$$

The calculation proceeds from bottom row up, and within a row, from left to right. Refer also to Figures 5 and 6.

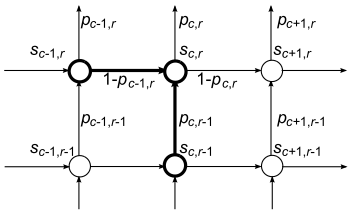

Figure 5. State probability derivation in a slot-level model

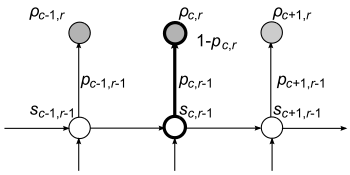

Figure 6. Round execution probability in a slot-level model

5. Probability design

Let us consider an attack to round r. The weakest point for this attack is the slot $c+r$ (in the original coordinates) where the round probability $\rho_{c,r}$ is maximum. Yet, the round r must be executed at some time, that is

$$\sum_{c=1}^{W} \rho_{c,r} = 1, r = 1 \dots C \tag{12}$$

For optimum protection, we thus require

$$\rho_{c,r} \stackrel{!}{=} 1/W, c = 1 \dots W, r = 1 \dots C \tag{13}$$

Using Equation 11, we obtain

$$p_{c,r} \stackrel{!}{=} 1/W s_{c,r-1}, c = 1 \dots W, r = 1 \dots C \tag{14}$$

Combining Equations 9, 10 and 14, we again define a calculation that proceeds bottom-up and left-to-right and gives the optimum transition probabilities together with state probabilities as a by-product. Notice that there is no choice in the process, that is, all the transition probabilities follow from the requirement in Equation 13 and there is only one solution.

There are explicit formulas for the transition probabilities. It can be proven (by a rather tedious proof) that the above recurrent computation gives

$$p_{c,0} = \frac{1}{W - c + 1}, c = 1 \dots W \tag{15}$$

$$p_{c,r} = 1, c = 1 \dots W, r = 1 \dots C - 1 \tag{16}$$

This means that the optimum protection executes a number of redundant rounds first, given by transition probabilities in Equation 15. Then, it executes all rounds as active, and finally executes redundant rounds to the required number of slots. Refer also to Figure 7.

6. Discussion

Section 5 proves that there is always an optimum solution which satisfies Equation 13. An attack to any round must collect more traces to achieve certain probability, that the desired round has been executed with a given

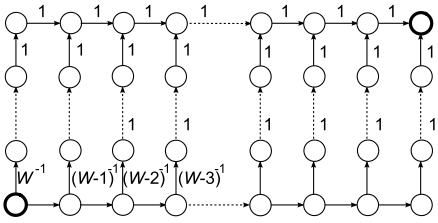

Figure 7. Optimum trajectories in a slot-level model

probability in the collected traces. For hit probability h, the relative increase q in the number of traces is

$$q = \frac{\log(1-h)}{\log(1-1/W)} \tag{17}$$

It can be seen that the amount of protection depends on work effort only. The function is, unfortunately, almost linear in the practical range of work effort, see Table 1. With an average work effort, around 40 times the number of traces are required to collect compared with the unprotected circuit.

W	q	W	q
2	7	11	49
3	12	12	53
4	17	13	58
5	21	1 4	63
6	26	15	67
7	30	16	72
8	35	17	76
9	40	18	81
10	44	19	86

TABLE 1. MULTIPLES OF REQUIRED TRACES q AS A FUNCTION OF WORK EFFORT W FOR $h = 0.99$

7. Conclusion

For the dummy rounds scheme, there is always an optimum set of transition probabilities which makes the round execution probabilities uniform for a particular round. This ensures maximum resistance against an SCA targeted to a particular round. A trajectory in the optimum set executes a random number of redundant rounds first, then all the active rounds, and then redundant rounds again. The transition probabilities in the first phase must follow a formula presented in the paper.

The protection scheme forces the attacker to collect a multiple of traces sufficient to attack the original circuit. The multiple is roughly proportional to the relative work effort invested into the protection.

The optimization presented here can, in principle, be applied to other similar schemes. For example, the random window technique in [7] desynchronizes collected power traces. The authors do not analyze how uniform the desynchronization is. The dummy operations schedule could be optimized to keep the probability that a particular trace has a particular time alignment as low as possible.

Acknowledgment

We thank again Prof. Dr.-Ing. Tim Güneysu for his suggestions, which, not surprisingly, also follow from the analysis in this contribution.

This work was partially funded by the CELSA project "DRASTIC: Dynamically Reconfigurable Architectures for Side-channel analysis protection of Cryptographic implementations" (CELSA/17/033), the grant GA16-05179S of the Czech Grant Agency, "Fault-Tolerant and Attack-Resistant Architectures Based on programmable Devices: Research of Interplay and Common Features" (2016-2018) and CTU project SGS17/213/OHK3/3T/18.

References

[1] S. Jeřábek, J. Schmidt, M. Novotný, and V. Miškovský, "Dummy rounds as a DPA countermeasure in hardware," in *2018 21st Euromicro Conference on Digital System Design (DSD)*, Aug 2018, pp. 523–528.

[2] H. Feistel, "Cryptography and computer privacy," *Scientific American*, vol. 228, no. 5, pp. 15–23, 1973.

[3] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct 1949.

[4] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[5] M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart, "Power analysis, what is now possible..." in *Advances in Cryptology — ASIACRYPT 2000*, T. Okamoto, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 489–502.

[6] B. Gierlichs, J.-M. Schmidt, and M. Tunstall, "Infective computation and dummy rounds: Fault protection for block ciphers without check-before-output," in *Progress in Cryptology – LATINCRYPT 2012*, A. Hevia and G. Neven, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 305–321.

[7] S. Pontié, P. Maistri, and R. Leveugle, "Dummy operations in scalar multiplication over elliptic curves: a tradeoff between security and performance," *Microprocessors and Microsystems*, vol. 47, no. Part A, pp. 23–36, 2016.

[8] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential power analysis in the presence of hardware countermeasures," in *Cryptographic Hardware and Embedded Systems — CHES 2000*, Ç. K. Koç and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 252–263.

[9] S. Tillich, C. Herbst, and S. Mangard, "Protecting AES software implementations on 32-bit processors against power analysis," in *Applied Cryptography and Network Security*, J. Katz and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 141–157.

[10] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F.-X. Standaert, "Shuffling against side-channel attacks: A comprehensive study with cautionary note," in *Advances in Cryptology – ASIACRYPT 2012*, X. Wang and K. Sako, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 740–757.

[11] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, P. Paillier and I. Verbauwhede, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466.

[12] J. Zhang, D. Gu, Z. Guo, and L. Zhang, "Differential power cryptanalysis attacks against PRESENT implementation," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 6, Aug 2010, pp. V6–61–V6–65.