Masking and Dual-rail Logic Don't Add Up

Patrick Schaumont[1] and Kris Tiri[2]

[1] ECE Department, Virginia Tech, Blacksburg VA 24061, USA
schaum@vt.edu
[2] Digital Enterprise Group, Intel Corporation, Hillsboro OR 97124, USA
kris.tiri@intel.com

Abstract. Masked logic styles use a random mask bit to de-correlate the power consumption of the circuit from the state of the algorithm. The effect of the random mask bit is that the circuit switches between two complementary states with a different power profile. Earlier work has shown that the mask-bit value can be estimated from the power consumption profile, and that masked logic remains susceptible to classic power attacks after only a simple filtering operation. In this contribution we will show that this conclusion also holds for masked pre-charged logic styles and for all practical implementations of masked dual-rail logic styles. Up to now, it was believed that masking and dual-rail can be combined to provide a routing-insensitive logic style. We will show that this assumption is not correct. We demonstrate that the routing imbalances can be used to detect the value of the mask bit. Simulations as well as analysis of design data from an AES chip support this conclusion.

1 Introduction

In recent years, several different circuit styles have been proposed to prevent side-channel attacks based on differential power analysis (DPA) [1]. These circuit styles attempt to remove the correlation between the power consumption and the signal values at selected internal circuit nodes. The circuit-level and logic-level techniques that have been proposed to remove this correlation fall into two major categories: masking techniques, which randomize power consumption, and dual-rail circuits, which flatten the power consumption.

In this paper, we show that a circuit contains inherent information leaks determined by the circuit structure at the module level. Indeed, each design has a specific power consumption characteristic determined by the ensemble of gates that make up that circuit. This characteristic can be quantified with a probability density function, which can be exploited for side-channel attacks. Earlier research has shown that single-bit masking can be broken by filtering of the masked probability density function [2]. We will demonstrate that this approach is applicable to all recently proposed secure masked logic design styles: those that are based on masking and pre-charged logic [3], and those that use a combination of masking and dual-rail techniques [4][5]. We will quantify the conditions under which these side-channel leaks become visible in the power

probability-density-function using simulation as well as using analysis of the layout-data of an actual chip.

Our attack is different from the ones that are usually considered for secure logic styles. For example, glitches have shown to be a source of side-channel leakage [6], because the presence of a glitch depends on the specific input data pattern on the circuit. Further, the arrival time of signals at gate inputs can cause small data-dependent variations on the switching time of gates [7]. This variation shows up in the power-consumption pattern and can be exploited in power analysis attacks. Third, the loading imbalance of dual-rail circuits causes small variations in power consumption [8], which then become a source of side-channel leaks as well. All these leaks are caused by electrical effects and thus are technology-dependent. In contrast, we will look at the circuit from a system perspective. The basis of our attack is not an electrical effect, but the probability density function (pdf) of the power consumption. We investigate the effect of circuit-level techniques (masking, dual-rail) on the pdf and conclude that in practical cases we can undo the effect of masking and dual-rail by filtering operations on the probability density function. The filtered pdf then becomes subject to standard differential power analysis attacks.

The paper is structured as follows. In section 2, we will review briefly the important properties of major secure logic styles. In section 3, we introduce a sample test-circuit, containing an S-box of 970 logic gates. Using a cycle-based model of the circuit in masked pre-charged logic (RSL), we derive the power-pdf using toggle-counting on the cycle-based simulation of the model. We show how a differential power analysis attack (DPA) can be performed. The same circuit then is modeled in masked dual-rail pre-charged logic (MDPL). Perfectly-matched dual-rail masked logic would result in a constant toggle-count, with a power pdf that contains a single impulse. However, we will analyze the effect of small mismatches in loading between the dual-rail nets. We will show that the mismatch in loading re-enables pdf analysis, and consequently a DPA attack. This conclusion contradicts claims of earlier research [4]. In section 4, we explore the consequences of our attack on a placed-and-routed dual-rail masked circuit. The circuit is a complete AES core containing 16K dual-rail gates. This illustrates that the attack mechanism also works on practical and large designs. We follow up with the conclusions in section 5.

2 Masked, Dual-rail and Pre-charged Logic

This section presents a brief review of the major secure logic styles. All of the logic styles apply pre-charging, and a selected combination of dual-rail and masking. We discuss, in sequence, Random Switching Logic (masked pre-charged logic), Wave Dynamic Differential Logic (dual-rail pre-charged logic), and Masked Differential Pre-charge Logic (masked dual-rail pre-charged logic).

2.1 Random Switching Logic (RSL)

In masking techniques, the computation performed by a logic gate is hidden by masking the actual data with a (pseudo-)random mask bit. The mask bit can be extracted afterwards to reveal the resulting data. A systematic implementation of masking, called RSL, was proposed by Suzuki [3]. The RSL nor and nand gates are defined as follows.

$$
\begin{aligned}
nor_{rsl}: \quad & z = \overline{\overline{e} + x.y + (x+y).\overline{r}} \\
nand_{rsl}: \quad & z = \overline{\overline{e} + x.y + (x+y).r} \\
& with\ x = a \oplus r,\ y = b \oplus r,\ z = q \oplus r
\end{aligned}
$$

These equations illustrate the transformation of the unmasked inputs a and b into an unmasked output q. Signal r is the mask bit, which switches the RSL gate between two complementary forms. Signal e is the enable bit, which serves to eliminate glitches on the RSL gate.

In an RSL circuit, only the primary input signals are masked, and the mask is removed again at the primary outputs. Internal signal nodes remain in masked form while traveling from one gate to the next. All gates are connected to the mask bit as well as to the enable bit. Assume for a moment that e is 1, then the mask bit switches the gate between dual configurations as shown below.

Evaluate with r = 1	Evaluate with r = 0
$nor_{rsl}\|_{r=1} = \overline{x.y}$	$nor_{rsl}\|_{r=0} = \overline{x+y}$
$nand_{rsl}\|_{r=1} = \overline{x+y}$	$nand_{rsl}\|_{r=0} = \overline{x.y}$

The signal e is only 1 after all inputs (x, y, r) have arrived, and serves to eliminate glitches on the RSL gate. The signal e also has the effect of a pre-charge signal. When e is low, the outputs of all RSL gates are zero. When e is high, the outputs of all RSL gates will evaluate the input signals. Therefore, there are only two transitions in an RSL gate that disclose information which is correlated to the input data: $0 \rightarrow 0$ and $0 \rightarrow 1$. Another way of formulating this is to say that the dynamic power consumption of RSL directly reflects the number of '1' data signals in the circuit. This is important since, as we will show later, the behavior of a circuit can be characterized by an average number of '1' data signals. Since the two dual configurations, in which the circuit can be depending on the value of the mask bit, each have a characteristic average number of '1' data signals, the value of the mask can be extracted by estimating this average.

Note that the authors of [2] incorrectly assumed that the enable signal only ensures that the logic is glitch-free. They did not discuss the pre-charge effect of the enable signal. As a result the effect of the random mask bit is not as visually noticeable in the power transient as they put forward.

2.2 Wave Dynamic Differential Logic (WDDL)

WDDL is a dual-rail pre-charged logic style, with a logic-1 and a logic-0 represented by a complementary differential pair as $(1,0)$ and $(0,1)$ respectively [9]. WDDL inserts a zero-spacer $(0,0)$ at the beginning of each clock cycle. As a result, there will be exactly one $0 \rightarrow 1$ transition per differential pair and per clock cycle, regardless of the logic value carried by the pair. WDDL is implemented with complementary logic. Using the same notation as for RSL, we have the following relations.

$$
\begin{aligned}
&nor_{wddl}: && (z, z_c) = (x_c.y_c, x + y) \\
&nand_{wddl}: && (z, z_c) = (x_c + y_c, x.y) \\
&\text{when } e = 1 \rightarrow && x = a,\ y = b, \\
& && x_c = \overline{a},\ y_c = \overline{b},\ q = z \\
&\text{when } e = 0 \rightarrow && x = 0,\ y = 0, \\
& && x_c = 0,\ y_c = 0
\end{aligned}
$$

The differential input pairs are (x, x_c) and (y, y_c), and they are generated from the inputs when the enable signal e is 1. A $(0,0)$ spacer is inserted when the enable signal e is 0. Note that the pre-charge signal e is not present on individual gates, but merely controls zero-spacer insertion on the primary inputs of the circuit. WDDL gates are implemented with positive logic only; when an inversion is needed the differential wire pairs are switched. The zero-spacers therefore propagate as a wave through the circuit. There are two possible transitions in a WDDL gate that disclose information related to the data input signal: $(0,0) \rightarrow (0,1)$ and $(0,0) \rightarrow (1,0)$. Consequently, if a differential wire pair maintains symmetry, power consumption will remain constant. If, on the other hand, there are small loading imbalances between the two wires making up a pair, there will be a residual information leak.

2.3 Masked Dual-Rail Pre-charge Logic (MDPL)

MDPL combines the ideas of RSL and WDDL into a masked, dual-rail pre-charged logic style [4]. Like WDDL, MDPL represents a logic-1 with a differential pair $(1,0)$ and a logic-0 with a differential pair $(0,1)$. In addition, a zero-spacer $(0,0)$ is used to pre-charge all differential pairs once per clock cycle. The zero-spacer is inserted on the primary inputs of the circuit under control of the pre-charge signal e. Like RSL, MDPL also uses a mask bit to switch the circuit between two complementary forms. MDPL enables a compact logic formulation using majority-gates, which are gates that implement the majority-count function (MAJ) of their inputs.

$$
\begin{array}{ll}
nor_{mdpl}: & (z, z_c) = (MAJ(x_c, y_c, r), MAJ(x, y, r_c)) \\
nand_{mdpl}: & (z, z_c) = (MAJ(x_c, y_c, r_c), MAJ(x, y, r)) \\
 & with\ MAJ(a, b, c) = a.b + a.c + b.c
\end{array}
$$

$$
\begin{array}{ll}
\text{when } e = 1 \rightarrow & x = a \oplus b,\ y = b \oplus r, \\
 & x_c = \overline{a} \oplus r,\ y_c = \overline{b} \oplus r, \\
 & q = z \oplus r, r_c = \overline{r} \\
\text{when } e = 0 \rightarrow & x = 0,\ y = 0,\ r = 0,\ x_c = 0,\ y_c = 0
\end{array}
$$

Assume for a moment that e is 1, then the mask bit switches the gate between dual configurations as shown below.

Evaluate when r=1		Evaluate when r=0	
$nor_{mdpl}\|_{r=1}$	$= (x_c + y_c, x.y)$	$nor_{mdpl}\|_{r=0}$	$= (x_c.y_c, x + y)$
	$= (a + b, \overline{a}.\overline{b})$		$= (\overline{a}.\overline{b}, a + b)$
	$= (a + b, \overline{a + b})$		$= (\overline{a + b}, a + b)$
$nand_{mdpl}\|_{r=1}$	$= (x_c.y_c, x + y)$	$nand_{mdpl}\|_{r=0}$	$= (x_c + y_c, x.y)$
	$= (a.b, \overline{a} + \overline{b})$		$= (\overline{a} + \overline{b}, a.b)$
	$= (a.b, \overline{a.b})$		$= (\overline{a.b}, a.b)$

The dual rail and precharge behavior of the MDPL gate fulfills the duty of the enable signal in RSL. It ensures that there is a single data dependent transition per clock cycle. There are only two possible data-dependent transitions in an MDPL circuit: either $(0, 0) \rightarrow (0, 1)$, or $(0, 0) \rightarrow (1, 0)$. Moreover, the specific transition that a pair will take depends on the logic value as well as on the random mask pair (r, r_c). For this reason, MDPL is believed to have no loading symmetry requirements between the wires of a differential pair. We will demonstrate that this is not correct.

3 An Attack using the Power Probability Density Function

In this section we demonstrate the weaknesses of each of the above mentioned logic styles using a DPA attack on a simple test circuit. The circuit in Figure 1 is a simplified encryption circuit consisting of an AES Sbox and a key addition. A test-bench drives the circuit with 8-bit pseudorandom data. The Sbox and key-addition are modeled at gate-level using 970 logic gates (99 not-gates, 388 nand-gates and 483 nor-gates). The DPA attack will attempt to reveal the key by using the output data and the simulated power consumption. The power consumption is simulated by toggle counting, with each gate output of the above-mentioned 970 gates contributing to the overall toggle count. The simulation is cycle-based, and makes abstraction of detailed delay effects as well as electrical variations between individual nets and gates. We use this idealized model to demonstrate that the side-channel does not rely on an electrical effect, but rather on the logic structure of the circuit.

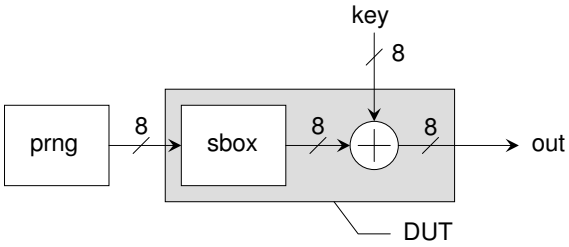

Fig. 1. The test circuit: AES Sbox and key addition

3.1 Random Switching Logic (RSL)

The first simulation implements the above circuit using RSL gates. The power pdf is calculated by monitoring the toggle count over the input signal space, and converting the toggle count to a histogram. Each power-consuming $0 \rightarrow 1$ transition contributes a unit weight to the overall toggle count. The resulting power pdf is plotted as a dashed line in Figure 2. For example, the dashed line indicates a value of about 0.034 in bin 485. This means that the probability that exactly 485 gates (of the 970) will carry a logic-1 during an evaluate-period (e=1) is about 0.034.

Figure 2 also illustrates two other distributions which are obtained as follows. We sorted all the power samples in two groups according to the value of the mask bit (zero or one), and created a partial pdf for each group. The partial pdf are drawn as bar-charts in two different shades. The light-shaded bars correspond to power samples with a zero mask bit. The dark-shaded bars are those samples which have a mask bit of one. Note that, while individual light bars and dark bars are drawn aside from each other, both charts use the same set of bins. Each bin contains a light bar and a dark bar, and the sum of both bars amounts to the level of the dashed line.

The pdf has several interesting properties. First, the distributions for mask r=1 and mask r=0 do not overlap nicely. This is expected as the mask bit puts the circuit in one of the two complementary forms. The two complementary forms do not perform the same calculations on the internal data signals and as a result do not have the same characteristic average number of '1' data signals. In fact, when we see a toggle count below 485, we can say with high probability that the mask bit is zero (r=0). When the toggle count is above 485, the mask bit is probably one (r=1). Detailed analysis reveals that the two sub-histograms are mirror-images of each other around toggle count 485.

To prepare an RSL power-trace for DPA, we can fold the power trace around the average value 485. The resulting power trace will closely approximate the unmasked power trace. This works because of the following reason. RSL allows only two possible transitions on a masked net: $0 \rightarrow 0$ and $0 \rightarrow 1$ transitions. The sum of these two types of transitions thus must equal the total number of masked nets (970). The mask bit has the effect of interchanging the $0 \rightarrow 0$

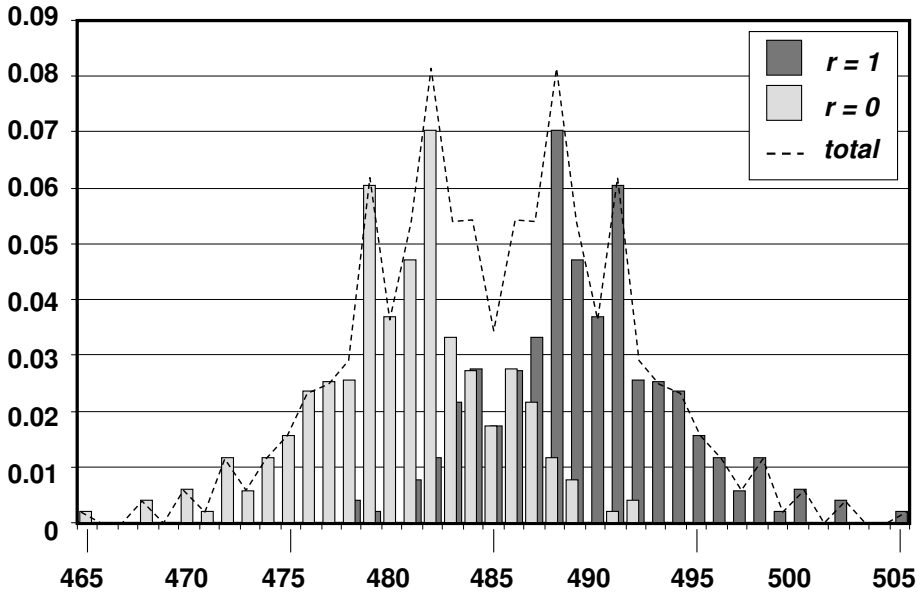

Fig. 2. Estimated Power Probability Density Function for RSL

with $0 \rightarrow 1$ transitions in a masked circuit. For example, if we find 490 $0 \rightarrow 1$ transitions, we may assume that mask $r = 1$. To remove the effect of the mask bit, we derive the equivalent toggle count for mask $r = 0$. This must be 970 - 490 = 480 toggles. In a practical implementation of this attack, we can measure a masked power trace, and then fold the resulting measurement around the average measured value when the measured value exceeds the average value. The folding technique only fails for a small part of the masked pdf, namely for the part where the mask is estimated incorrectly. Experimental results confirm that the DPA attack can find the key with only 30 power samples, when the attack is based on the Hamming weight of the input.

Table 1. Statistics of the RSL power pdf.

	$r = 0$	$r = 1$	overall
Min toggle	465	478	465
Max toggle	492	505	505
Average toggle	480.6	489.4	485
Stdev toggle	4.44	4.45	6.24
Entropy (bit)	4.07	4.07	4.61

Table 1 collects additional statistics of the RSL power pdf. The table highlights another interesting property: the pdf has a very low entropy. The entropy

or information content of a signal with N discrete values with probabilities p_i is defined as follows:

$$H(S) = \sum_{i=1}^{N} -p_i.log_2(p_i)$$

The overall power-trace has only 4.61 bit of information per power sample. The low entropy value is surprising because the circuit has 970 gates, and thus theoretically contains 2^{970} power states. The table also reveals that the addition of the random bit increases the entropy (from 4.07 to 4.61 bit). This is important as it indicates that it remains possible to strip the additional mask bit out.

Perfect masking should not modify the entropy. Indeed, under this condition, the masked signal does not carry any information about the masking scheme that was used to obtain it. The masking scheme itself thus is a secret variable, required to restore the original signal. However, it is presently not known how such a perfect masking scheme can be created. Recently, leakage functions were introduced to model the information flow from input to power-trace [10]. These leakage functions could be used to evaluate a masking scheme upfront.

3.2 Masked Dual-Rail Pre-Charge Logic (MDPL)

Our second simulation uses the same test circuit implemented with MDPL gates. In MDPL, each logic pair of wires can make two possible transitions: $(0,0) \rightarrow (0,1)$ and $(0,0) \rightarrow (1,0)$. When these differential nets are perfectly matched, we will measure a constant toggle count of 970 for the overall circuit. However, we performed a simulation with a small but uniform imbalance between the wires from each pair. The simulation was performed so that a $(0,0) \rightarrow (0,1)$ transition incremented the toggle count by 1, while a $(0,0) \rightarrow (1,0)$ transition incremented the toggle count by 0.99. This simulates a mismatching of 1%. While a uniform imbalance among all differential pairs is artificial, it allows us to clarify the method of our attack. In section 4, we will extend this attack to a non-uniform imbalance.

We did not include the toggle count graph for this simulation because it looks similar to the one shown in Figure 1. Only the X-axis (bin counts) is different. The bins for the MDPL simulation are numbered 964.95 to 965.35 (whereas the RSL bins go from 465 to 505). Even though for MDPL, the number of '1' data signals is constant, the distributions do not nicely overlap. Similarly as for RSL, the mask bit puts the circuit in one of the two complementary forms. The two complementary forms do not perform the same calculations on the internal data signals, and thus do not have the same transitions on the capacitances attached to the data signals, and as a result do not have the same characteristic average load.

Similar to the RSL case, the value of the mask bit can be estimated by considering if a power value is above or below the estimated average: When we see a toggle count below 965.15, we can assume that the mask bit has a particular value. When the toggle count is above 965.15, the mask bit has probably the opposite value. We can preprocess the power trace in a similar fashion as for

RSL: determine the average in the power-pdf and fold the upper part over the lower part. The resulting transformation removes the effect from the mask bit, and the resulting power-trace can be broken using DPA with only 30 samples. This illustrates a key point in this paper: the benefits of masking and dual-rail are not additive for side-channel resistance. Of course, the side-channel in MDPL is relatively smaller than in the case of RSL, and more sensitive power measurements must be made.

Can the above attack also work on a realistic circuit? In the next section, we show how to attack a large design in masked dual rail logic, which was implemented through layout-level using place-and-route. In this case, the imbalance among differential pairs is variable.

4 Applying the Attack on a Chip-level Layout of AES

In this section, we demonstrate that the conclusions from the previous chapter also hold for a large design with factual design data. We will show that the loading imbalances on masked dual-rail pre-charged gates in a chip-level layout are sufficient to enable the attack methodology described in this paper.

4.1 Device Under Test

We opted to demonstrate our findings on a large and practical circuit. The device under test is a complete AES core with encryption data path and key scheduling. The AES core is based on a single round of the AES-128 algorithm which consists of byte substitution, shift row, mix column and key addition phases along with on-the-fly key scheduling (see Figure 3). The byte substitution is implemented using look-up tables. A full encryption of 128-bit data using a 128-bit key takes precisely eleven clock cycles.

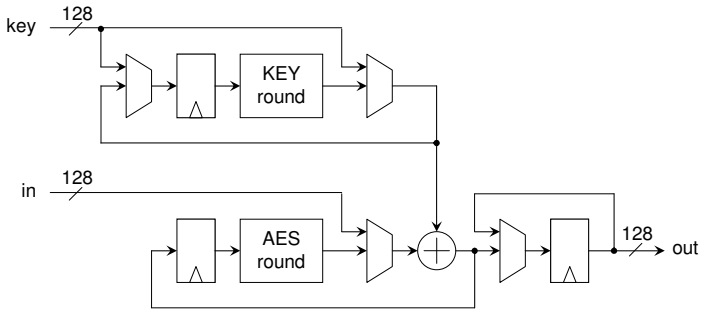

Fig. 3. Device under test: AES core with encryption data path and key scheduling

The gate level netlist describing the AES core contains just over 16K dual rail gates. The dual rail netlist has been placed-and-routed using Silicon Ensemble without any routing constraints. This means that the true net and the

false net of each differential pair are routed independently from each other. The lumped interconnect capacitances of the nets, which will be used in the power measurement simulations, have been extracted using the HyperExtract parasitic extractor, which takes second order and cross-coupling capacitances into account.

The peak and average supply current of a measurement sample, which are generally used in real life DPA attacks, are proportional to the sum of all the individual load capacitances being charged in the clock cycle of interest. Hence, we simulate the power consumption with weighted toggle counts. The simulation is cycle-based. Each power-consuming $0 \rightarrow 1$ transition contributes a weight to the overall toggle count equal to the extracted interconnect capacitance value of the switching net. The overall toggle count is thus equal to the total load capacitance.

To avoid biasing the power measurements, we do not take the load of the mask bit into account. This ensures that there is no direct observability of the mask bit value due to the very large capacitive load of the signals r and r_c, which are distributed to each gate. In other words, we assume that special precaution has been taken to match the capacitive loads between r and r_c. Instead, we simulate the masking of dual rail with precharge logic as follows. We simulate genuine dual rail logic having a single power transition per clock cycle. In each clock cycle the weighted toggle count of the circuit is recorded as is the weighted toggle count of the circuit in complementary form. Subsequently, in a post-processing step, a random mask bit is generated and one of the two weighted toggle counts is chosen based on the value of the mask bit. This simulates the correct cycle true behavior of both MDPL and Dual Rail RSL [5].

4.2 Power-based SCA Results

Figure 4 shows the probability density function of the weighted toggle counts based on the observations for 1,000,000 encryptions. As expected based on our observations in sections 3.1 and 3.2, the distributions for mask $r = 1$ and mask $r = 0$ do not nicely overlap. The mask bit thus also introduces a bias for a large circuit and for an in-depth analysis using actual extracted capacitances of a placed-and-routed circuit.

Furthermore, where there was some overlap between the distributions of the small circuit with uniform mismatch between the true and false nets (section 3.2), there is no overlap in this example. This is due to the following. The variation (i.e. the width) of a distribution is smaller than the distance between the two distributions. The variation is set by data signals which have a random transition for every observation. An example of such a signal is the input to the AES round. The distance is set by signals which always have the same transition for every observation. Examples of such signals are the control signals, which set the AES core for the last round, and the input to the KEY round, which calculates the last round key. When the mask bit changes, all those invariable signals make the opposite transition. The distance is thus set by the structural mismatch between two large capacitances, which has a binary effect. The variation, however, is set

by cumulated mismatch between many small capacitance pairs, which has a Gaussian effect.

The value of the mask bit can again be estimated by considering if a power value is above or below the calculated average. When we observe a weighted toggle count sample below 332.48, we can assume that the mask bit has a particular value. When the toggle count is above 332.48, the mask bit has the opposite value. Since there is no overlap between the distributions, the mask bit is always correctly deduced. A simple threshold filter allows separating the two power profiles and undoing the masking operation. Note that all measurement samples can be utilized during the DPA, by folding the resulting measurements around the average measured value as explained in section 3.1.

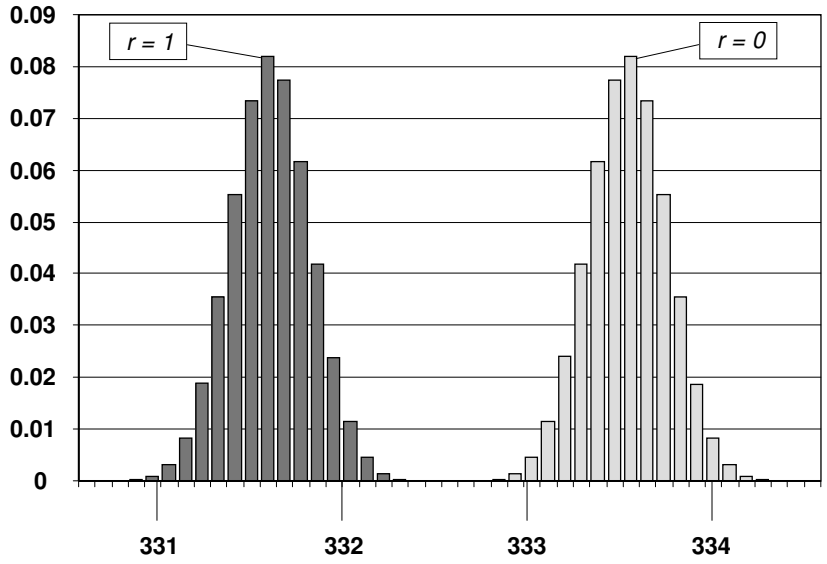

Fig. 4. Estimated power probability density function of device under test

Without the filtering operation, a DPA is unsuccessful. None of the key bytes was disclosed even after all of the 1,000,000 measurement samples are taken into account. Once the mask bit is removed, however, unbalanced capacitances caused by routing differences make a DPA possible: 2,000 measurement samples are sufficient to disclose the first key bytes. This confirms that masked logic styles remain susceptible to classic power attacks after only a simple filtering operation. For masked logic, whether it is single ended or dual rail logic, to work, the power probability density function should not disclose any information regarding the mask bit value. The distributions for the different mask bit values can be made more difficult to distinguish by increasing their overlap, for example by *not* masking constant signals.

5 Conclusions

Masking and dual-rail logic do not add up. We have shown with simulations as well as analysis of design data from an AES chip that individual weaknesses remain if both are blindly combined. Indeed, the mask bit puts the circuit into one of two dual configurations. Without routing constraints, loading imbalances will be present between differential wires and the two dual configurations can not have the same characteristic power consumption. As a result, the masking can easily be undone by observing whether the measurement sample is below or above the average power consumption. Once the mask bit has been removed, a power attack easily discloses the key due to loading imbalances between the differential wires.

Acknowledgements. Patrick Schaumont was supported in part by the National Science Foundation (CCR-0644070).

References

1. S. Mangard, E. Oswald, T. Popp, "Power Analysis Attacks: Revealing the Secrets of Smart Cards," 2007, ISBN 978-0-387-30857-9, 2007, XXIV + 338 p, Springer-Verlag.
2. K. Tiri, P. Schaumont, "Changing the Odds against Masked Logic", Selected Areas of Cryptography 2006 (SAC), LNCS, Springer-Verlag, to appear.
3. D. Suzuki, M. Saeki, T. Ichikawa, "Random Switching Logic: A Countermeasure against DPA based on Transition Probability," Cryptology ePrint Archive, Report 2004/346, 2004.
4. T. Popp, S. Mangard, "Masked Dual-Rail Pre-charge Logic: DPA Resistance without the Routing Constraints," Proc. of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), LNCS 3659, p. 172-186, August 2005.
5. Z. Chen, Y. Zhou, "Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage," Proc. of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2006), LNCS 4249, p. 242-254, October 2006.
6. S. Mangard, T. Popp, B. Gammel, "Side-channel Leakage of Masked CMOS Gates," Proc. of the RSA Conference 2005 Cryptographers' Track, LNCS 3376, p. 351-365, February 2005.
7. D. Suzuki, M. Saeki, "Security Evaluation of DPA Countermeasures using Dual-Rail Pre-charge Logic Style," Proc. of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2006), LNCS 4249, p. 255-269, October 2006.
8. D. Suzuki, M. Saeki, T. Ichikawa, "DPA Leakage Models for CMOS Logic Circuits", Proc. of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), Lecture Notes in Computer Science, vol. 3659, pp. 366-382, August 2005.
9. K. Tiri, I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," Design, Automation and Test in Europe Conference (DATE 2004), p. 246-251, 2004.
10. F.-X. Standaert, E. Peeters, C. Archambeau, J.J. Quisquater, "Towards Security Limits of Side-Channel Attacks," Proc. of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2006), LNCS 4249, p. 30-45, October 2006.