A Countermeasure Against Power Analysis Attacks for FSR-Based Stream Ciphers

Shohreh Sharif Mansouri and Elena Dubrova

Department of Electronic Systems, School of ICT
KTH - Royal Institute of Technology, Stockholm
Email:{shsm,dubrova}@kth.se

Abstract. In this paper we analyze the power characteristics of Feedback Shift Registers (FSRs) and their effect on FSR-based stream ciphers. We introduce a technique to isolate the switching activity of a stream cipher by equalizing the current drawn from the cipher with lower power overhead compared to previously introduced countermeasures. By re-implementing the Grain-80 and the Grain-128 ciphers with the presented approach, we lower their power consumption respectively by 20% and 25% compared to previously proposed countermeasures.

1 Introduction

Constrained environments applications such as hardware authentication devices (RFID tags, etc.), smartcards, and wireless networks (Bluetooth, NFC, tags, etc.) require power-efficient, area-efficient and high-performance hardware encryption systems with large security margins. At present, FSR-based stream ciphers are one of the promising candidates into cryptographic primitives for advanced contact-less technologies like RFID because they have one of the smallest hardware footprints among all available cryptographic algorithms.

In cryptography, a side channel attack is an attack on the physical implementation of a cryptosystem [1]. Power consumption is one of the physical characteristics of a system which can be used as a source of information to reveal its structure. This leads to a type of side channel attacks called power analysis attacks [1]. One of the most popular power analysis attack is DPA (Differential Power Analysis) where an attacker records a large number of power traces while the device encrypts or decrypts data and analyzes them to reveal the secret key of the cryptographic system. As discussed in [2, 3] stream ciphers are vulnerable to DPA.

Several countermeasures to DPA attacks have been suggested for other cryptographic algorithms such as block ciphers. These methods alter the internal operation of the device under attack so that the secret information content in the power signature is reduced. While these countermeasures may be effective for stream ciphers, most of them have high area and power overheads. Due to the extreme power limitations in applications which use stream ciphers, all previous power masking solutions introduce high overheads. In addition, cost is another important factor in some of the target applications such as RFID tags. Although top security is desirable, the extra security should not come at the expense of excessively increasing the power consumption of the system. Therefore, there is a need for countermeasures which can achieve a high security margin with reasonable area, power and cost overheads.

In this paper we first analyze the specific power characteristics of FSRs and then introduce a countermeasure which exploits these characteristics to mask the power of FSR-based stream ciphers with a lower power overhead compared to previously introduced countermeasures [4–6]. Our idea is to mask the power consumption of a stream cipher by predicting its power consumption during real-time operation based on the switching activity of the FSR, where most of the power is consumed. This countermeasure does not make the DPA attack impossible, but it increases the effort required to break the cipher.

We demonstrate the advantages of the proposed approach on the stream cipher Grain-80 (Grain-128). Compared to previous methods we succeed to save in average 20% (25%) more power at the expense of 16% (14%) area overhead.

The remainder of the paper is organized as follows: in Section 2, related work is summarized; Section 3 gives an introduction to FSRs and analyzes their dynamic power consumption; Section 4 describes our approach; implementation details are discussed in Section 5 and the final results and security issues are considered respectively in Section 6 and 7. Section 8 concludes the paper.

2 Related work

Several countermeasures have been suggested to protect cryptographic algorithms from DPA attacks. Architecture level countermeasures concentrate on changing the quality of the power diagram so that it shows a complete different pattern compared to the original power diagram. Architecture level countermeasures can be implemented by noise insertion [7], random clock frequency [8], randomization of the instruction streams [9] or random insertion of dummy instructions into the execution sequence of the algorithm [10]. Although these methods succeed in changing the chip's power consumption pattern, it is still possible for an attacker to reconstruct the original pattern [8, 11].

In contrast to these solutions, other countermeasures mask the correlation between data and power consumption by using an isolation circuit [4–6, 12] or by using dual rail logic [13–15]. Two interesting isolation circuits can be found in [6] and [4]: [6] presents a special masking countermeasure for the AES block cipher while [4] presents a method which does not require any change in hardware implementation and masks the power variation by pushing the current consumption always to a constant value.

DPA analysis on FSR-based stream ciphers is a new research area and the countermeasures designed specifically for these ciphers are not many. While previous countermeasures may be effective on stream ciphers, most of them have high area and power overheads which make them unsuitable for FSR-based stream ciphers.

An architecture level countermeasure which can be used in new stream ciphers is [16]. The authors suggest a new implementation of FSRs which maximizes the switching activity in each cycle. The resulting FSR is twice larger compared to the original design and consumes more power. The only work in literature which specifically targets new stream ciphers such as Grain and Trivium is [13], in which the authors implement Grain and Trivium with SABL logic and manage to decrease the power variations of both ciphers. Although dual rail logic gates have less power variations compared to standard cells, they are larger and consume more power. The SABL based cipher in [13] has twice power and area compared to the standard cipher. In addition, SABL, TDPL [14] and 2N-2Np [15] cells are non-standard and therefore more difficult to design.

In this work we suggest a countermeasure specifically designed with standard cells for stream ciphers. We use the same suppression circuit suggested in [4] and in each cycle we choose the cipher's total current based on its switching activity, so that the power overhead of the method is reduced.

3 FSRs and Dynamic Power Consumption

Stream ciphers are symmetric key ciphers. FSR-based stream ciphers such as [17–20] consist of one or more linear or non-linear Feedback Shift Registers (FSRs) and some combinational blocks. Most of the area and power of FSR-based stream ciphers are consumed by the FSRs

themselves. FSRs are a chain of synchronous flip-flops connected back-to-back, with a feedback on the first flip-flop (in Fibonacci configuration). In this section we analyze the properties of FSRs and model their power consumption.

The total power of an FSR can be modeled as:

$$P = P_L + P_D = P_L + P_{CK} + P_{SA}$$

where P_L is the leakage power of the FSR, $P_D = P_{CK} + P_{SA}$ is its dynamic power, P_{CK} is the dynamic power consumed by the clock tree and P_{SA} is the dynamic power due to switching of the internal nets of the FSR.

For side channel attacks, the leakage power P_L of the FSR is of minor importance. This power remains constant during operation and therefore carries no information about the state of the FSR. On the other hand, the data-dependent dynamic power P_D of the FSR is the main source of information that can be exploited for power analysis attacks. P_{CK} also remains constant during operation. The power consumption P_{SA} of an FSR in one cycle can be modeled as:

$$P_{SA} = \sum_{n \in N} C_n V^2 SA_n$$

where N is the set of all nets in the FSR, C_n the capacitance of net n, V the power supply of the FSR and SA_n the switching activity of net n, i.e. a number which is 1 if the net toggles, and 0 otherwise. N should include all nets in the shift register and all nets in the feedback function. However, for security reasons most stream ciphers use long FSRs. For these FSRs the power consumption of the shift register is much higher compared to the power consumption of the feedback function, and the contribution of the feedback function to power consumption is negligible. Therefore, from the point of view of power consumption, an n-bits FSR can be seen as having n nets, each corresponding to the output of a flip-flop, i.e. to a state bit. Also, because of the regular structure of FSRs, in which all cells have the same structure and all are connected back-to-back, the capacitance of every net is the same, i.e. $C_n \simeq C$. The model is then reduced to

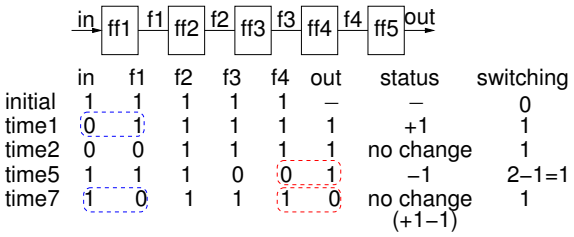
$$P_{SA} = CV^2 SA$$

where SA, the switching activity of the FSR, is the number of state bits in the FSR switching from 0 to 1 or from 1 to 0 during the clock cycle. There is some difference in power consumption when a net switches from 0 to 1 or from 1 to 0 [3]. However, this difference is small and is not taken into account here.

In contrast with other systems, with this model the switching activity of an FSR is determined solely by its state bits. As an example, if state bit i is at 1 in one cycle, and in the same cycle state bit $i-1$ is at 0, then state bit i will switch from 1 to 0 in the next cycle. The only exception to this rule is the first state bit, which is updated by the feedback function of the FSR. Neglecting the first bit, the switching activity of the FSR can be obtained by analyzing the sequence of its state bits $b_0, b_1, b_2, ...b_{n-1}$ and determining the number of $0,1$ and $1,0$ pairs in the sequence.

Figure 1 shows the operation of a 5-bits FSR initialized with the key 11111. Because the state bits are shifted by one position in every clock cycle, the switching activity of the system can only increase by one, decrease by one or remain constant, based on the bit that is output by the last flip-flop of the feedback shift register and the bit that enters the shift register. If a $0,1$ or $1,0$ sequence is eliminated, then the switching activity can remain constant or decrease by one, depending on the input of the shift register (time 5, 7). If a $0,0$ or $1,1$ sequence is eliminated, then the switching activity can remain constant or increase by one (time 1, 2).

In Figure 2 the SPICE simulation of a 5-bits FSR initialized with key value 11111 is shown. Current peaks (proportional to power) are proportional to the switching activity of the system. The highest peaks correspond to the time instants in which the sequence of state bits has the highest switching activity.



	in	f1	f2	f3	f4	out	status	switching
initial	1	1	1	1	1	–	–	0
time1	0	1	1	1	1	1	+1	1
time2	0	0	1	1	1	1	no change	1
time5	1	1	1	0	0	1	−1	2−1=1
time7	1	0	1	1	1	0	no change (+1−1)	1

Fig. 1. Switching activity of a 5-bits FSR with key value 11111.

Normally, n-bits FSRs used in stream ciphers iterate through $2^n - 1$ states during operation before repeating the same sequence (the state 00000.... is illegal). All 2^{n-1} states have therefore the same probability to occur when the FSR is initialized with a random key. The switching activity of the FSR can take any value between 0 and n. Opposite to most other systems, the switching activity in an FSR has a regular distribution. In particular, the number of internal states of an n-bits FSR with switching activity i is equal to the binomial coefficients of i out of n:

$$\binom{n}{i} = \frac{n!}{i! \times (n-i)!}, 0 \leq i \leq n$$

As shown in Figure 3 for a n-bits FSR, Although the switching activity can be between 0 and n, in practice most of the states have switching activity concentrated in a small range around $\frac{n}{2}$. As shown in Figure 3, for a 160-bits FSR 99.9% of the states have switching activity between 50 and 110. The figure also shows that the switching activity is more concentrated around $\frac{n}{2}$ for longer FSRs.

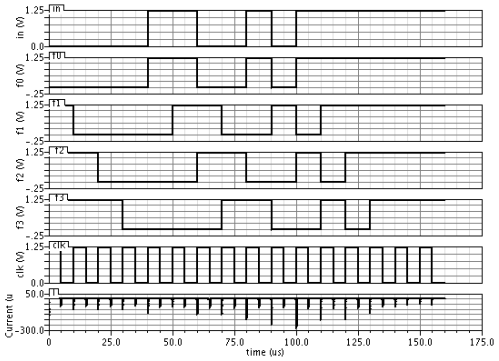


Fig. 2. The current peak is higher at times $90\mu s$ and $100\mu s$ when the sequence of state bits has a higher number of "10" or "01".

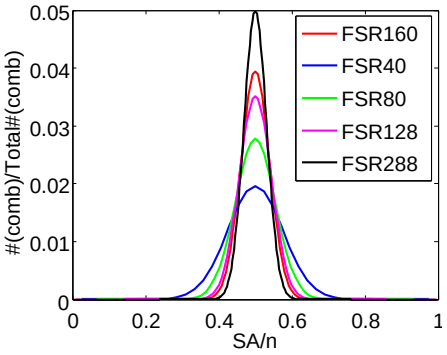


Fig. 3. Number of states with a given Switching Activity for 40, 80, 128, 160 and 288-bits FSRs.

4 Cipher Total Power and Suggested Approach

As mentioned, in FSR-based stream ciphers such as Grain [18], Trivium [20] and Mickey [17], most of the total area is occupied by FSRs and the total power consumption of the cipher is directly related to the power consumption of its FSRs which is related to its switching activity.

FSRs in stream ciphers normally have a large number of flip-flops. As an example, Grain-80 has two 80-bits FSRs, Grain-128 [19] has two 128-bits FSRs and Trivium has one 288-bits FSR. The power consumption of an FSR-based cipher is strongly influenced by the switching activity of its FSRs. Based on SPICE simulation of the FSRs of Grain-80, we obtain that their power consumption is 67% higher when their switching activity is maximal compared to when their switching activity is minimal, i.e. the total normalized power consumption of the two FSRs can be modeled as:

$$P = 0.33 + 0.67\frac{SA}{n}$$

where SA is the sum of the switching activities in the two FSRs of Grain and $n = 160$. Methods such as [4], when applied to FSR-based stream ciphers, mask the total current in hardware by pushing the power always to the maximum value, i.e. they force the current to be always equal to the current I_M which is consumed when the switching activity of the FSRs is maximal. When the switching activity is not maximal, a current sensor senses that less current is consumed by the cipher and shunts some current so that the total current consumed by the system is equal to I_M. The method introduces a very large power overhead when the FSRs have a non-maximal switching activity. In the Grain cipher, application of this masking method causes a 33% power overhead in average.

To decrease the power overhead of the masking method, we suggest to mask the power to some specific levels depending on the switching activity of the cipher. In other words, based on the range of the switching activity SA we determine the masking current $I_M(SA)$. We introduce different levels, i.e. we divide the range of all possible switching activities in different intervals and associate to each of them one value of I_M. The levels are chosen based on the distribution of the switching activities. As an example, as shown in Table 1, for a 160-bits FSR, 49% of the internal states have switching activity between 50 and 80 (level 1), 49% have switching activity between 80 and 110 (level 2) and only 1% have switching activity lower than 50 or higher than 110. For a 160-bits FSR we can suggest 3 different power levels: all states with $SA \leq 80$ are masked with a current I_{M1} corresponding to the current consumed by the cipher when $SA = 80$; all states with $80 < SA \leq 128$ are masked with a current I_{M2} corresponding to the current consumed by the cipher when $SA = 128$; all states with $SA > 128$ are masked with the maximum current I_{M3} corresponding to the current consumed by the cipher when $SA = 160$.

With these values we have $I_{M1} = 67\%I_{M3}$ and $I_{M2} = 83\%I_{M3}$.

If the cipher changes too often between the three power levels, cryptanalysis attacks could be possible. We ran simulations of Grain-80 for 10^6 random keys and ran every simulation for 400000 cycles, corresponding to the output of a $50KB$ message, and found out that the difference between the maximal switching activity SA_{max} and the minimal switching activity SA_{min} in each run is less than 45 for 98% of the simulations.

FSR	Property	L1	L2	L3
160 bit	SA Levels	[0 : 80]	[81 : 127]	[128 : 160]
	#(state)	50%	49%	1%
	power μW	4.2	5.14	6.18
	P_L/P_{max}	67%	83%	100%
256 bit	SA Levels	[0 : 128]	[129 : 160]	[161 : 256]
	#(state)	50%	50%	~ 0
	power μW	6.40	7.59	10.02
	P_L/P_{max}	64%	75%	100%

Table 1. Relation between switching activity and total power consumption in 160 and 256-bits FSRs.

All the power reports in this article are estimated as a combination of dynamic and leakage power for operation at 27°, with a power supply of 1.2V at 1.1 MHz clock frequency.

5 Power Masking Algorithm and Implementation

Before they can generate a stream of data, stream ciphers must be initialized. During the initialization phase, the cipher does not produce any output. The initialization phase lasts 160 clock cycles for Grain-80 [18], 256 cycles for Grain-128 [19] and 1152 cycles for Trivium [20]. During this phase, the output value of the cipher is XOR-ed with the outputs of the FSR feedback functions and then fed into the inputs of the shift registers. After the initialization, the cipher enters the key generation phase, in which the loops are opened and the cipher produces 1 bit of data in each clock cycle.

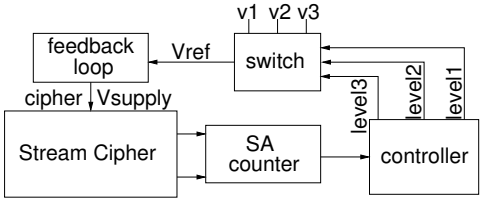


Fig. 4. Schematic diagram of the suggested countermeasure.

Most DPA attacks such as [2] or [3] happen during the initialization phase. A correlation DPA attack is performed for every round of the attack and the key bits are extracted.

In fact, masking the power consumption of the cipher is more critical during the initialization phase; therefore, during this phase, we suggest to check the current drawn by the cipher and to shunt an appropriate current so that the total current drawn from the supply shows only minimal variations and is equal to the current consumed by the cipher when the switching activity is maximal.

Later, during the key generation phase, in every cycle, the current still is checked and an appropriate current is shunted to make the total current equal to one of the defined levels.

Opposite to key generation phase, this level can be lower than the maximum current and can change based on the switching activity of the FSRs (see Algorithm 1).

As shown in Figure 4, our solution contains analog and digital blocks which both are integrated into the cipher chip; therefore the attacker cannot probe the data which is being exchanged between the blocks and the cipher.

5.1 Analog Blocks

Suppression Circuit For this block, we use the same circuit as the one presented in [4]. The circuit is based on a feedback loop made of a shunt transistor and an operational amplifier. This loop forms a high pass filter which senses current variations and draws an appropriate amount of current through the shunt transistor in order to keep the voltage at the source of the transistor at the defined voltage level (Vref).

In case the target cryptographic system is small and the overhead of the operational amplifier is too high, it is possible to design a suppression circuit based on diodes which draw an appropriate amount of current when the supply voltage at the cipher is higher than Vref. This solution is out of the scope of this paper.

In parallel to the feedback loop, the sense resistor and the capacitance C_{filter} form a low pass filter and consume the voltage at the source of the transistor if it is higher than Vref.

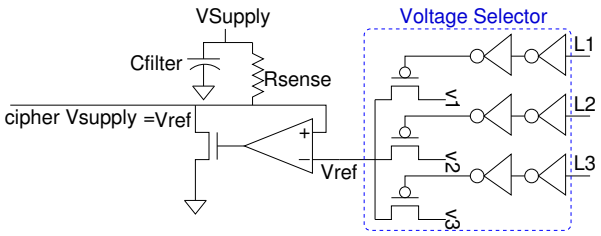


Fig. 5. Analog block containing the voltage selector and the suppression circuit.

Voltage Selector The Voltage Selector receives three input signals from the digital blocks. In each cycle, based on the switching activity of the cipher, only one of these input signals is active. The active signal corresponds to the appropriate voltage which is necessary as V_{ref} to guarantee that the correct current is shunted. For a 3-level voltage selector for a stream cipher with a 160-bits FSR, the voltage levels are given by:

$$Vref_i = V_{supply} - R_{sense}(Ifsr_i + Icount_i + Imax_{comb.})$$

where $i \in \{80, 128, 160\}$, $Imax_{comb.}$ is the sum of the maximum currents of all the combinational blocks in the cipher and controller, $Icount_i$ is the maximum power consumption of the SA counter (see Figure 4) while it counts to i and $Ifsr_i$ is the current consumption of the FSR with switching activity equal to i.

As shown in Figure 5 the 3-level voltage Selector is implemented using PMOS switches.

5.2 Digital Blocks

To keep track of the switching activity, we use an adder-subtractor which counts the series of 1,0 or 0,1 in the FSRs. Instead of calculating the switching activity of the FSR in each cycle, the switching activity counter tracks it by considering only the bit that exits the stream cipher and

the bit that enters it. If a series of $1,0$ or $0,1$ is inserted and a series of $1,0$ or $0,1$ is removed, the switching activity counter remains idle. If a series of $1,0$ or $0,1$ is inserted and a series of $1,1$ or $0,0$ is removed, the switching activity is increased by one. If a series of $1,1$ or $0,0$ is inserted and a series of $1,0$ or $0,1$ is removed, the switching activity is decreased by one.

As shown in Figure 6, based on the switching activity of the FSRs in each cycle, only one of the power level signals (level1, level2, level3) is active. When the cipher switching activity rises over a certain boundary, the power level jumps to the next higher level and stays in that level at least for some defined time k. k is chosen based on the degree of security. In this paper we choose k equal to 16 clock cycles. During this period, even if the switching activity gets lower than the boundary level, the controller does not change back to the lower level. This guarantees that the cipher does not change the power level too often and introduces an additional non linearity between power and switching activity. To simulate this behavior an additional counter is needed. For example, for waiting k= 16 cycles, a 4 bit counter is needed. Stream ciphers contain counters that are used only during the initialization phase (see Figure 7). To minimize the overheads, these counters can be reused for this purpose.

Algorithm 1 Power Masking Algorithm

```
1: if phase = initialization then
2:    CPLevel = level3
3:    PPLevel = initialState
4: else
5:    if CPLevel = PPLevel + 1 OR PPLevel = initialState then
6:       Wait for K clock cycles in CPLevel.
7:    else
8:       if SA < SA_l1 then
9:          CPLevel = level1
10:      end if
11:      if SA_l1 ≤ SA < SA_l2 then
12:         CPLevel = level2
13:      end if
14:      if SA_l2 < SA then
15:         CPLevel = level3
16:      end if
17:   end if
18: end if
```

The pseudo-code of the digital module is shown as algorithm 1 where PPLevel is the previous power level and CPLevel is the current power level.

The maximum switching activity is equal to the number of bits in the FSRs; therefore, Grain-80, with its two 80-bits FSRs needs a 8 bit counter and Trivium and Grain-128 need a 9 bit counter to keep track of the switching activity. The overall area overhead of the digital blocks with an 8 bit counter and the control block to determine the power level is 1030 μm^2. The total power is $0.88\mu W$ when the counter is working in all the cycles.

6 Experimental Results

In this section, we evaluate the presented approach on the stream ciphers Grain-80 and Grain-128. Grain-80 (128) consists of one 80-bits (128-bits) LFSR, one 80-bits (128-bits) NLFSR, one initialization counter and two combinational functions (see Figure 7).

Based on RTL synthesis [21], the FSRs take 80% of the total area of Grain-80 and 85% of Grain-128 and the two combinational functions occupy less than 10% of Grain's area. Inside each

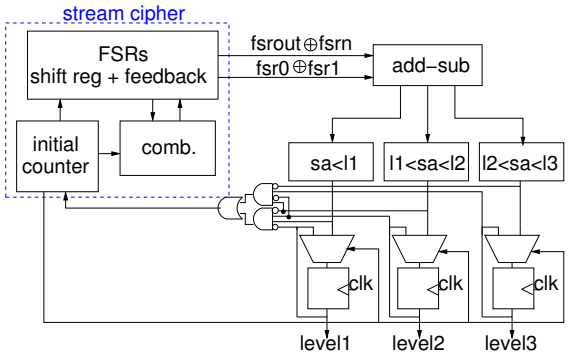


Fig. 6. Schematic diagram of the digital modules added to the original cipher.

FSR, the feedback functions consume much less power (in average $< 5\%$) compared to the FSRs. Therefore, for simplicity, during power masking, they are supposed to have always maximum power consumption. The same considerations are valid for the other two combinational functions. The NLFSR input is obtained by XORing the LFSR output bit and the feedback function; to

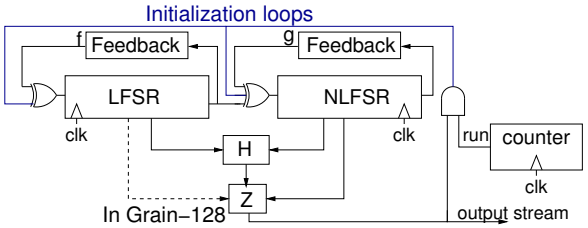


Fig. 7. Schematic structure of the Grain cipher.

simplify the hardware implementation, the two FSRs can be considered as a single 160-bit (256-bit) FSR; therefore only one counter is enough to calculate the switching activity. The power levels in Table 1 can be used for both Grain-80 and Grain-128. For the 160-bits FSR of Grain-80, the value 128 for level 3 is chosen to simplify the implementation of the digital controllers.

Grain-80					Grain-128				
Power μW			Area μm^2		Power μW			Area μm^2	
L1	L2	L3	ORG	WPM	L1	L2	L3	ORG	WPM
6.71	8.31	9.64	6849	7984	9.13	10.46	14.02	8576	9773
69%	86%	100%	100%	116%	65%	74%	100%	100%	114%

Table 2. Area comparison of the original (ORG) Grain-80 (128) and the same ciphers using our solution (WPM), and average power consumption of both ciphers in the three power levels.

For synthesis and power estimation, we used Cadence RTL Compiler [21] backannotated from gate level simulation in UMC 90nm ASIC technology library. The area overhead of the circuit in Figure 6 is 16% for Grain-80 and 14% for Grain-128 and the average power overhead is estimated as 9% for Grain-80 and 5% for Grain-128 in the worst case. Considering the results in Table 2, which include the overhead for the additional blocks, our method decreases in average the power 20% for Grain-80 and 25% for Grain-128 compared to the method in [4].

The working frequency for Grain is chosen equal to 1.1MHz to make both Grain-80 and Grain-128 decode 128 bits of data in less than $320\mu s$. $320\mu s$ is the time RFID tags have before starting to transmit data to the reader according to the ISO/IEC 18000 protocol. ISO/IEC 18000 is an ISO standard for passive RFID item level identification [22].

Figure 8 shows a SPICE simulation of the current pattern of Grain-80 using our countermeasure. The system runs for 230 μs at 1MHz clock frequency. In this article the focus is on the digital blocks and their effects on the cipher and no work is done on optimizing the suppression circuit. Therefore the operational amplifier overhead on the total power is not considered. In this example the cipher always has switching activity lower than 60. Therefore, after the initialization phase is completed at time $160\mu s$, the cipher switches to Level 1 and the current consumption decreases by 31%.

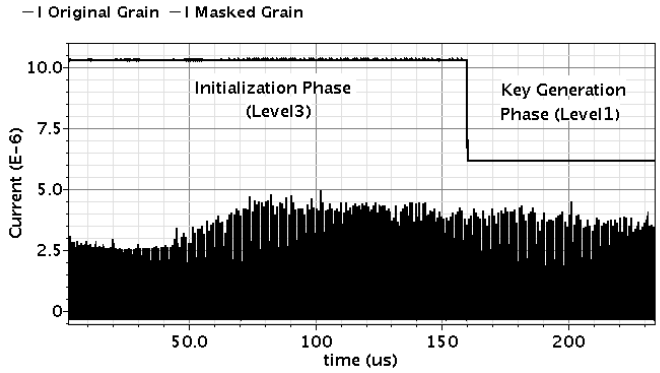


Fig. 8. Grain-80 with power masking blocks have smoother current pattern compared to the Grain-80 current peaks.

7 Security Considerations

We perform a DPA attack on the unprotected and the protected Grain-80 for 230 guessed keys. We consider two versions of protected Grain-80: one with three different power levels and another with two power levels. For the cipher with three power levels, $L1$ corresponds to $SA \leq 80$, $L2$ corresponds to $80 < SA \leq 128$ and $L3$ corresponds to $SA > 128$. For the cipher with two power levels, $L1$ corresponds to $SA \leq 110$ and $L2$ corresponds to $SA > 110$. The keys used for the attack differ from the correct one in at most two bits.

Lacking a manufactured circuit, we execute the attack on the power consumption results obtained from RTL compiler, estimated based on gate-level switching activity back-annotated through a VCD file. To make the attack realistic, a white noise signal up to 0.5% of the maximum power is added to the power consumption results in each sampling.

Figure 10 shows the correlation coefficients of the guessed keys for the DPA attack on the unprotected Grain-80 after 5K encryptions. In contrast, as shown in Figure 9, the two-levels protected Grain-80 is still resistant against DPA attack after 1M encryptions.

The Measurements To Disclosure (MTD) is defined as the minimum number of measurements which are necessary to distinguish the correct key from all the other guessed keys [23].

MTD is defined as the number of encryptions for which the correct key's correlation coefficient diagram intersects the maximum correlation coefficient diagram of all the wrong guessed keys.

As shown in Figure 11, for the unprotected Grain-80 MTD occurs at 188 runs where the correlation coefficient of the correct key becomes higher than the maximum correlation coefficient

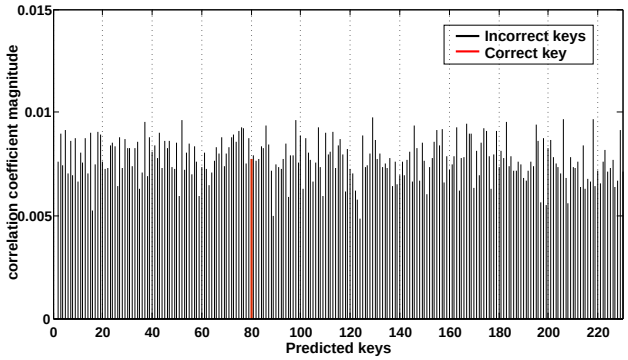


Fig. 9. Correlation coefficients of the 230 guessed keys on 3-levels protected Grain-80 after 1M encryptions.

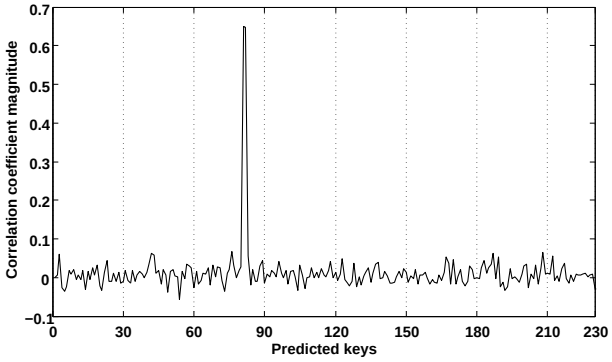


Fig. 10. Correlation coefficients of the 230 guessed keys on unprotected Grain-80 after 5k encryptions.

of all incorrect keys. As shown in Figure 12 the protected Grain-80 with 2 power levels has MTD higher than 20K runs.

Grain-80 with three power levels, as suggested in Figure 6, has MTD equal to 556 (see Figure 13). Although this MTD is lower than MTDs of Grain with one or two different power levels, it is still higher than the unprotected Grain-80 with MTD equal to 188. If Grain-80 with three power levels and higher MTD is necessary, the power levels can be redefined such as: $L1 : SA < 64$, $L2 : 64 \leq SA < 128$ and $L3 : 128 \leq SA < 160$. In this case, as shown in Figure 14, the MTD increases to 8K.

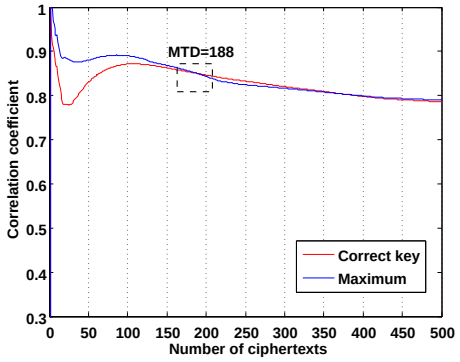


Fig. 11. Key disclosure in unprotected Grain-80 occurs at 188 ciphertexts.

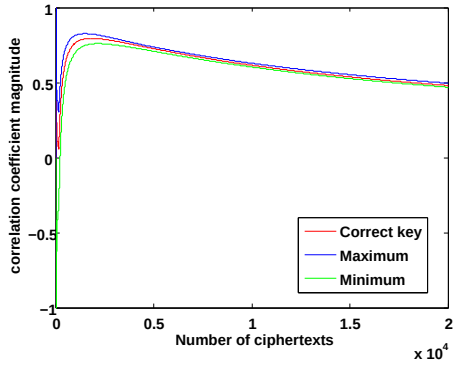


Fig. 12. Key disclosure in Grain-80 with two power levels can occur for ciphertexts larger than 20k.

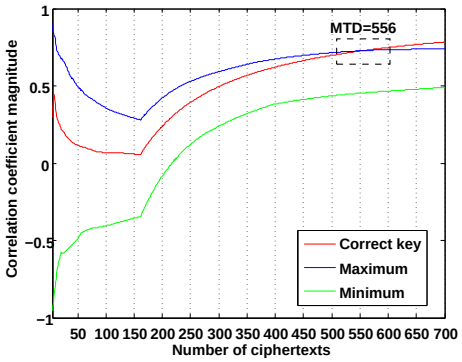


Fig. 13. Key disclosure in Grain-80 with three power levels of Table 2 occurs at 556 ciphertexts.

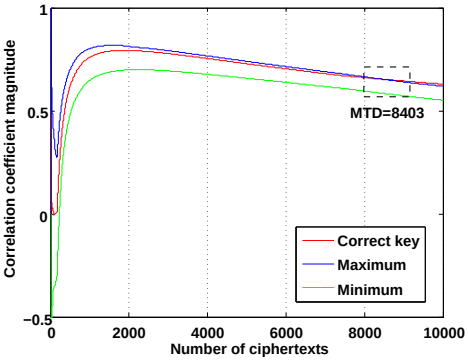


Fig. 14. Key disclosure in Grain-80 with the three alternative power levels occurs at 8k ciphertexts.

8 Conclusion

We suggested a countermeasure designed with standard CMOS cells for differential power attack analysis on stream ciphers. The power of Grain-80 (128) decreased by 20% (25%) in average during the key generation phase at the expense of 16% (14%) area overhead.

9 Acknowledgement

This work was supported on part by the Swedish Governmental Agency for Innovation Systems (Vinnova) through the Vinn Excellence centers program and a research grant No. 621-2010-4388 from the Swedish Research Council.

References

1. S. Mangard and et Al., Eds., *Power Analysis Attacks Revealing the Secrets of Smart Cards*, 2007.
2. W. Fischer and et Al., "Differential power analysis of stream ciphers," in *Lecture Notes in Computer Science*, 2007.
3. D. Strobel, "Side channel analysis attacks on stream ciphers," *master thesis*, 2009.
4. G. B. Ratanpal and et Al., "An on-chip signal suppression countermeasure to power analysis attacks," *TDSC 2004*.
5. A. Shamir, "Protecting smart cards from passive power analysis with detached power supplies," in *CHES 2000*.
6. C. Tokunaga and D. Blaauw, "Secure aes engine with a local switched-capacitor current equalizer," in *ISSCC 2009*.
7. P. Kocher and et Al., "Differential power analysis," in *CHES 1999*.
8. M.-L. Akkar and et Al., "Power analysis, what is now possible," in *ASIACRYPT 2000*.
9. P. Grabher and et Al., "Non-deterministic processors: Fpga-based analysis of area, performance and security," in *WESS 2009*.
10. C. Clavier and et Al., "Differential power analysis in the presence of hardware countermeasures."
11. T. S. Messerges and et Al., "Investigations of power analysis attacks on smartcards," in *In USENIX Workshop on Smartcard Technology*, 1999.
12. P. Rakers and et Al., "Secure contactless smartcard asic with dpa protection," in *CICC 2000*.
13. R. E. Atani and et Al., "On dpa-resistive implementation of fsr-based stream ciphers using sabl logic styles," *ijccc 2008*, December.
14. M. Bucci and et Al., "Three-phase dual-rail pre-charge logic," in *CHES 2006*.
15. A. Moradi and et Al., "Charge recovery logic as a side channel attack countermeasure," in *ISQED 2009*.
16. S. Burman and et Al., "Lfsr based stream ciphers are vulnerable to power attacks," in *AFRICACRYPT 2007*.
17. S. Babbage and M. Dodd, "The mickey stream ciphers," *New Stream Cipher Designs: The eSTREAM Finalists*, 2008.
18. M. Hell and et Al., "The Grain family of stream ciphers," *New Stream Cipher Designs: The eSTREAM Finalists*, 2008.
19. ——, "A stream cipher proposal: Grain-128," in *Information Theory, 2006*.
20. C. Cannière and B. Preneel, "Trivium," *New Stream Cipher Designs: The eSTREAM Finalists*, 2008.
21. Cadence, "Using encounter rtl compiler, product version 9.1," 2009.
22. "International organization for standardization. ISO/IEC 18000-3. information technology aidc techniques RFID for item management," 2003.
23. K. Tiri, Hwang, and et Al., "Prototype ic with wddl and differential routing dpa resistance assessment," in *Cryptographic Hardware and Embedded Systems CHES 2005*.