Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 324-335

On DPA-Resistive Implementation of FSR-based Stream Ciphers using SABL Logic Styles

Reza Ebrahimi Atani, Sattar Mirzakuchaki, Shahabaddin Ebrahimi Atani, Willi Meier

Abstract:
The threat of DPA attacks is of crucial importance when designing cryptographic hardware. This contribution discusses the DPA-resistant implementation of two eSTREAM finalists using SABL logic styles. Particularly, two Feedback Shift Register (FSR) based stream ciphers, Grain v.1 and Trivium are designed in both BSim3 130nm and typical 350nm technologies and simulated by HSpice software. Circuit simulations and statistical power analysis show that DPA resistivity of SABL implementation of both stream ciphers has a major improvement. The paper presents the tradeoffs involved in the circuit design and the design for performance issues.

Keywords: DPA attack, Stream cipher, Grain v.1, Trivium, SABL, Standard CMOS.

1 Introduction

The term of security for a cryptographic primitive can be considered from two points of view: mathematical security (resistance against classical cryptanalysis) and the second one is physical security. Physical attacks on cryptographic devices take advantage of implementation-specific characteristics to recover the secret parameters. They are therefore much less general since they are specific to a given implementation but often much more powerful than classical cryptanalysis, and are considered very seriously by cryptographic devices implementors. A side-channel attack occurs when an attacker is able to use some additional information leaked from the implementation of a cryptographic function to cryptanalyze the function. Clearly, given enough side-channel information, it is trivial to break a cipher. One side channel attack in particular, namely the differential power analysis (DPA) is of great concern. It was first reported by Kocher et al. in 1998 that the power consumption of a smart card could reveal the secret key of the cryptographic algorithm [1]. DPA is a well-known and thoroughly studied threat for implementations of block ciphers (DES and AES), public key algorithms (RSA) and recently stream ciphers (Grain and Trivium [4]).

Stream ciphers as part of the symmetric key cryptography family, have always had the reputation of efficiency in hardware and speed. They have attracted much attention since the beginning of the eSTREAM project in 2004. Although there is vast literature about DPA on implementations of block ciphers and public key algorithms, only few publications can be found about DPA attacks on stream ciphers ([2], [3], [4], [8], [13], [14]).

In power analysis attacks, it is assumed that the power consumption of a circuit is correlated to the data handled. An attacker can therefore recover secret information by simply monitoring the power signals of a running device.

Stream ciphers require frequent synchronization to prevent synchronization loss between sender and receiver. Normally the initialization will be done with the same secret key and with a different initial value IV. So an attacker can disrupt the synchronization and apply a new known IV and measure the power traces in the initialization phase to apply a DPA on the embedded system of the stream cipher. So far, there is only one report on a practical DPA targeting hardware implementations of stream ciphers [4]. In that paper, a chosen IV DPA attack on Grain and Trivium stream ciphers has been described and executed. Protecting implementations against DPA attacks is usually difficult and expensive. The goal of countermeasures against DPA attacks is to make the power consumption independent of intermediate values of the stream cipher. In general, there are three basic groups into which these countermeasures can be

characterized: protocol countermeasures, algorithmic countermeasures, and hardware countermeasures [11].

The principles of the countermeasures can be implemented at different levels in a cryptographic device. In general, these techniques are theoretical countermeasures and only reduce the side channel leakage and do not fundamentally prevent a DPA. But the advantage of these countermeasures is to make the attack significantly harder. In this article, we provide a brief overview of hiding and masking logic styles (hardware countermeasures) and particularly we will use sense amplifier base logic (SABL) for secure implementation of stream ciphers. SABL is a logic style that uses a fixed amount of charge for every transition, including the degenerated events in which a gate does not change state. In every cycle, a SABL gate charges a total capacitance with a constant value.

So far, there has not been a unified architecture which can be used as a test bench for applicability of logic styles on stream ciphers. Regarding this, two FSR-based stream ciphers - Grain v.1 and Trivium stream ciphers - are implemented in cell level to find out the tradeoffs involved in designing the architecture and performance issues. Power traces of the resulting circuits exhibit that SABL significantly reduces signal to noise ratio (SNR). The rest of the paper is structured as follows: a general model of power analysis attack on stream ciphers is given in Section 2. Section 3 describes an overview of DPA Countermeasures on cell level. In sections 4 and 5 the descriptions of Grain v.1 and Trivium are explained. Design and simulation issues are described in section 6 and finally, conclusions are drawn in Section 7.

2 Differential Power Analysis of Stream Ciphers

DPA is based on the fact that CMOS logic and application specific details cause logic operations to have power characteristics that depend on the input data. It relies further on statistical analysis and error correction to extract the information from the power consumption that is correlated to the secret key [1]. In a DPA a hypothetical model of the device under attack is used to predict the power consumption. The classical setup for a DPA on stream ciphers is illustrated in Fig. 1. Output power traces are determined by the input data, IV, private key, output of the device and by many other parameters. An attacker to some extent has the potential knowledge of some of them (e.g. IV, input data and output data) while others are unknown. Regarding a DPA attack, multiple measurements of the power consumption of a cryptographic device are made. For each measurement, different chosen IV's are sent to the device. Since the cryptographic algorithm is known, a hypothesis on intermediate values can be used to calculate the targeted data values based on the random input values. If the correct hypothesis is used, the targeted data values are calculated correctly for all measurements. According to (1), the total power consumption of an embedded device depends on 3 factors:

$$P_{Total} = P_{Cons.} + P_{Noise} + P_{DD}. \tag{1}$$

With the help of statistical methods (calculation of correlations, mean values, etc.), the randomness of the data values that are not targeted ($P_{Const.}$: leakage currents and data independent power consumption and P_{Noise}: which comes from electrical noise) is exploited to reduce their effects on the power consumption traces. P_{DD} is the data dependent power consumption and is targeted in statistical analysis. After all, the result of the statistical operation indicates which key hypothesis is correct. Normally, a hamming distance power model is used to map the transitions that occur at the outputs of cells of a netlist to power consumption values. In CMOS gates, it is reasonable to assume that the main component of the data dependent power consumption is the dynamic power consumption which is the power dissipation of charging and discharging of output capacitance nodes ($P_{0\to1}$ or $P_{1\to0}$). In a CMOS gate, we can express dynamic power consumption by:

$$P_{Dynamic} = N \cdot C_L \cdot f \cdot V_{DD}^2 \tag{2}$$

where C_L is the gate load capacitance and N is the probability of a $0 \rightarrow 1$ or $1 \rightarrow 0$ output transition and f is the clock frequency. This equation shows that the power consumption of CMOS circuits is data dependent. Note that N is the most important factor in the hypothetical model. There are different techniques for calculation of it. For example, a variable gate delay model can be used for measuring the number of transitions and glitches of a circuit [7]. This technique can be easily applied to circuits by using a VHDL simulator in Register Transfer Level.

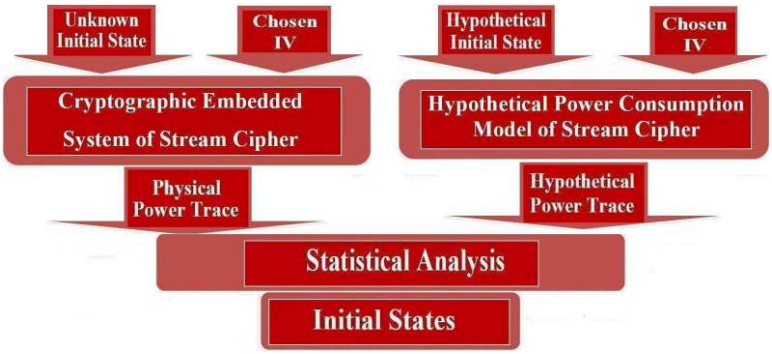

Figure 1: Differential power analysis model of stream ciphers.

3 DPA Countermeasures on Cell Level

So far, several methods in different ways have been proposed to counteract DPA attacks. In this section different known DPA countermeasures on cell level (hiding and masking techniques) are briefly presented and then their merits and disadvantages will be discussed.

The first structured approach to counteract DPA attacks at the cell level was the use of hiding logic styles. These styles try to break the correlation between an algorithm's intermediate results and the power consumption of the cryptographic device that executes this algorithm by making the instantaneous power consumption of the cells either random or the same in each clock cycle. The three major types of hiding logic styles are: Dual-Rail Precharge (DRP), Asynchronous, and Current Mode Logic (CML). DRP logic styles are the most popular types. For instance, SABL [10] and Wave Dynamic Differential Logic (WDDL) [15] are dual rail precharge logic styles whose logic gates are driven by a precharge signal to prevent glitches, and each logic signal is represented by two complementary wires. Other examples of DRP logic styles are Dual Spacer Dual Rail logic (DSDR), Three-phase Dual-rail Precharge Logic (TDPL) [16], and Three State Dynamic Logic (3SDL). Data dependent time of evaluation of the WDDL and its memory effect made it vulnerable to DPA attacks. One of the major drawbacks of hiding logic styles is the balancing of the cells and interconnect layouts to achieve constant power consumption. Since the charge and discharge of output nodes of dynamic and differential styles follow simple RC charge and discharge, cells and wires must mainly be balanced in a capacitive and resistive manner. But due to process variations, complex cross-coupling effects, and area limitations it is a hard task.

Besides hiding, masking at the cell level has become popular during the past few years. Using a masked logic style, designers also break the correlation between an algorithm's intermediate values and the power consumption of the cryptographic device that executes this algorithm. All intermediate values are masked by a random value. The cells then process only the masked intermediate values and their corresponding mask. Because the unmasked values and the masked value are uncorrelated, power consumption of the cell also remains uncorrelated to intermediate values. Generally there are two types of masking operation: boolean masking or arithmetic masking. If the masked cells are not activated in a data or operation dependent manner, masked logic styles counteract DPA attacks. There are two different

possible masking schemes: one mask per circuit (single masking) or one mask per signal. These masked bits are normally prepared by some random number/sequence generators.

Before, masking was mainly used at the architecture level. As a result, only a few practical results are available for this type of cell level countermeasure. For examples Masked Dual rail Precharge Logic (MDPL) [17] and Dual rail Random Switching Logic (DRSL) [18] were introduced by combining the masking scheme and dual rail precharge logic in order to use semi custom design tools without routing constrains. Designers can implement MDPL cells using commonly available conventional single rail standard cells. Only sequential cells are connected to the clock signal, and combinational cells precharge their outputs when their inputs have been set to the precharge value. The memory effect can reduce the DPA resistance of masked logic styles. Practical evaluations of the manufactured chips have also shown that early propagation is also a major threat to the DPA resistancy of masked logic styles.

Although all these efforts, it has been shown ([19], [20], [21]) that MDPL leaks information. For example in [19], it has been shown that MDPL is susceptible to the early propagation effect. In order to combat the early propagation issues, the designers of MDPL introduced a so called improved MDPL (iMDPL). In each iMDPL gate there is an evaluation precharge detection unit, which consists of three (CMOS) AND gates and two (CMOS) OR gates. Hence it is not surprising that the area requirements for iMDPL gates increased significantly compared to MDPL gates. Another threat to masked circuits is the detection of the mask value, which lets attackers completely cancel out the effect of masking in a DPA attack. In particular, such an attack is dangerous for single masked circuits, where only one mask value is used for all signals in the circuit. Increasing the number of mask values per circuit is an option but it is impractical regarding its high complexity and area utilization.

4 Sense Amplifier Based Logic

In this paper we will concentrate on SABL [10] for DPA resistive implementation of stream ciphers. SABL is part of the DRP logic styles. Fig. 2 shows the transistor schematic of standard SABL gate library used for implementation of ciphers. Equation (3) illustrates the power consumption of a SABL gate,

$$P = C_L \cdot f \cdot V_{DD}^2 + C_{Clk} \cdot f \cdot V_{DD}^2 \tag{3}$$

where C_L represents the total output capacitance of the gate and C_{Clk} is the clock propagation circuitry capacitance. As can be seen in the Fig.2, SABL gates can be designed using Differential Pull Down Networks (DPDN) or Differential Pull Up Networks (DPUN), controlled respectively by clk and $\overline{clk}$. This allows two modes for cascading SABL gates: domino connection (by connecting the outputs of the gate to the inputs of the next gate through inverters) or NP-connection (N-gates followed by P-gates like in NP-logic).

In SABL, the concepts of dual rail and precharge logic are combined to achieve constant power consumption. Precharging breaks a signal's sequence of values by splitting each clock cycle into precharge and evaluation phases. In the precharge phase, the complementary wires encoding a signal are set to a predefined precharge value, such as 1. In the subsequent evaluation phase, one of the two complementary wires is set to 1 according to the actual value that is processed. As a result, for each signal in a circuit, exactly one $0 \rightarrow 1$ transition and one $1 \rightarrow 0$ transition occur in a clock cycle. By ensuring a balance between the complementary wires between cells on the one hand and a balance of the internal structure of the cells on the other hand, designers can achieve constant power consumption. The price is high power consumption and high current spikes of these gates which appear at the beginning of the precharge phase. By the use of delayed clock mechanism introduced in [13] and [14] we can reduce the peak of these spikes.

But in practice the throughput is highly dependent on layout design of the chip to have balanced complementary wires. Since the charge and discharge of output nodes of differential styles follow simple

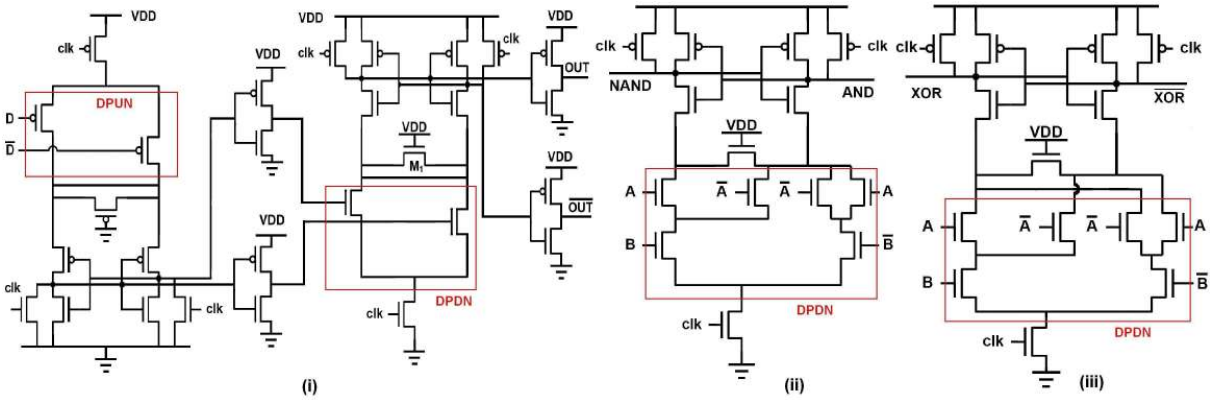

Figure 2: (i) SABL D-flip flop (ii) SABL Nand2 gate (iii) SABL Xor2 gate

RC charge and discharge, cells and wires must mainly be balanced in a capacitive and resistive manner. But due to process variations, complex cross-coupling effects, and area limitations this is hard to achieve. Avoiding these effects often requires custom cell design, which involves considerably more design effort than using available standard cells.

5 Grain Stream Cipher

Grain v.1 [5] is a stream cipher introduced in 2005 as a candidate for the hardware profile of eSTREAM project. Grain v.1 is a binary additive synchronous stream cipher with an internal state of 160 bits s_i, s_{i+1},..., s_{i+79} and b_i, b_{i+1},..., b_{i+79} residing in a linear feedback shift register (LFSR) and a nonlinear feedback shift register (NLFSR), respectively. The design of the algorithm mainly targets hardware environments where gate count, power consumption and memory is very limited. The key size of Grain is 80 bits (k_i, $0 \leq i \leq 79$). Additionally, an initial value of 64 bits (IV_i, $0 \leq i \leq 63$) is required. In initialization phase, all 80 NLFSR elements are loaded with the key bits, ($b_i = k_i$, $0 \leq i \leq 79$), then the first 64 LFSR elements are loaded with the IV bits, ($s_i = IV_i$, $0 \leq i \leq 63$). The last 16 bits of the LFSR are filled with ones. $f(x)$ and $g(x)$ are two polynomials used as feedback function for the LFSR and NLFSR.

$$f : s_{i+80} = s_{i+62} \oplus s_{i+51} \oplus s_{i+38} \oplus s_{i+23} \oplus s_{i+13} \oplus s_i \tag{4}$$

$$\begin{aligned} g : b_{i+80} = {} & s_i \oplus b_i \oplus b_{i+9} \oplus b_{i+14} \oplus b_{i+21} \oplus b_{i+28} \oplus b_{i+33} \oplus b_{i+37} \oplus b_{i+45} \oplus b_{i+52} \oplus b_{i+60} \oplus \\ & b_{i+62} \oplus b_{i+63} \cdot b_{i+60} \oplus b_{i+37} \cdot b_{i+33} \oplus b_{i+15} \cdot b_{i+9} \oplus b_{i+60} \cdot b_{i+52} \cdot b_{i+45} \oplus \\ & b_{i+33} \cdot b_{i+28} \cdot b_{i+21} \oplus b_{i+63} \cdot b_{i+45} \cdot b_{i+28} \cdot b_{i+9} \oplus b_{i+60} \cdot b_{i+52} \cdot b_{i+37} \cdot b_{i+33} \oplus \\ & b_{i+63} \cdot b_{i+60} \cdot b_{i+21} \cdot b_{i+15} \oplus b_{i+63} \cdot b_{i+60} \cdot b_{i+52} \cdot b_{i+45} \cdot b_{i+37} \oplus \\ & b_{i+33} \cdot b_{i+28} \cdot b_{i+21} \cdot b_{i+15} \cdot b_{i+9} \oplus b_{i+52} \cdot b_{i+45} \cdot b_{i+37} \cdot b_{i+33} \cdot b_{i+28} \cdot b_{i+21} \end{aligned} \tag{5}$$

The output function $h(x)$ uses as input selected bits from both feedback shift registers:

$$\begin{aligned} h(x) = {} & x_1 \oplus x_4 \oplus x_0 \cdot x_3 \oplus x_2 \cdot x_3 \oplus x_3 \cdot x_4 \oplus x_0 \cdot x_1 \cdot x_2 \oplus \\ + {} & x_0 \cdot x_2 \cdot x_3 \oplus x_0 \cdot x_2 \cdot x_4 \oplus x_1 \cdot x_2 \cdot x_4 \oplus x_2 \cdot x_3 \cdot x_4 \end{aligned} \tag{6}$$

where the variables x_0, x_1, x_2, x_3, and x_4 corresponds to the tap positions s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64} and b_{i+63} respectively. The output of the filter function is masked with the some state bits from the NFSR to produce the keystream z_i:

$$z_i = b_{i+1} \oplus b_{i+2} \oplus b_{i+4} \oplus b_{i+10} \oplus b_{i+31} \oplus b_{i+43} \oplus b_{i+56} \oplus h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$$

This output is used during the initialization phase as additional feedback to LFSR and NLFSR. During normal operation this value is used as key stream output. The generated output bits per clock cycle is called *Radix*. By implementing the small feedback functions, $f(x)$ and $g(x)$, and the output function several times the speed of Grain can easily reach up to Radix-32.

6 Trivium Stream Cipher

Trivium [6] is a stream cipher introduced in 2005 as a candidate for the hardware profile of the eSTREAM project. Trivium has an internal state of 288 bits $a_i, a_{i+1}, \ldots, a_{i+92}$, $b_i, b_{i+1}, \ldots, b_{i+83}$ and $c_i, c_{i+1}, \ldots, c_{i+110}$ - residing in three coupled NLFSRs A, B, and C of 93, 84, and 111 bits respectively. Trivium has a key $k = (k_0, \ldots, k_{79})$ of 80 bits as well as an initial value $IV = (IV_0, ..., IV_{79})$ of 80 bits. The initialization of the key and IV is done as follows:

$$\begin{cases} (a_0, \ldots, a_{92}) = (0, \ldots, 0, k_{79}, \ldots, k_0) \\ (b_0, \ldots, b_{83}) = (0, 0, 0, 0, IV_{79}, \ldots, IV_0) \\ (c_0, \ldots, c_{110}) = (1, 1, 1, 0, 0, \ldots, 0, 0) \end{cases} \tag{7}$$

Then, the state is updated over 4 full cycles, according to (3), but without generating key stream bits. After 1152 clocking it outputs a key stream bit z_i.

$$\begin{cases} a_{i+93} = a_{i+24} \oplus c_i \oplus (c_{i+1} \cdot c_{i+2}) \oplus c_{i+45} \\ b_{i+84} = b_{i+6} \oplus a_i \oplus (a_{i+1} \cdot a_{i+2}) \oplus a_{i+27} \\ c_{i+111} = c_{i+24} \oplus b_i \oplus (b_{i+1} \cdot b_{i+2}) \oplus b_{i+15} \end{cases} \tag{8}$$

$$z_i = a_i \oplus b_i \oplus c_i \oplus a_{i+27} \oplus b_{i+15} \oplus c_{i+45} \tag{9}$$

Trivium has a very simple structure that is well suited for different Radix implementations from Radix-1 to Radix-64 without noticeable hardware penalties.

The basic structure of the Grain v.1 and Trivium stream ciphers are shown in Fig. 3. In April 15, 2008, the eSTREAM competition was finished and according to the final report [12] both ciphers were selected among the four finalists of the H/W profile.

7 Design and Simulation Results

Both eSTREAM candidates are modeled at transistor level using a spice netlist. Circuit design of Grain v.1 and Trivium are mainly based on the techniques presented in [13] and [14]. In order to specify the impact of minimum feature size on the design, ciphers are designed using two technologies: typical BSIM3 $0.13\mu m$ CMOS SOI technology and typical $0.35\mu m$ CMOS SOI technology. Spice simulations were run to test the circuits by test vectors provided by the inventors of the ciphers using Hspice circuit simulator and C compiler. Domino cascading scheme is used for all SABL gate connections to make sure having a $0 \rightarrow 1$ transition in the input of all cascaded gates to prevent possible glitches. First a new standard gate library based on SABL logic is designed. Minimum possible sized transistors are used to lower the total capacitance to get lower dynamic power in (2). This will also minimize the charging time

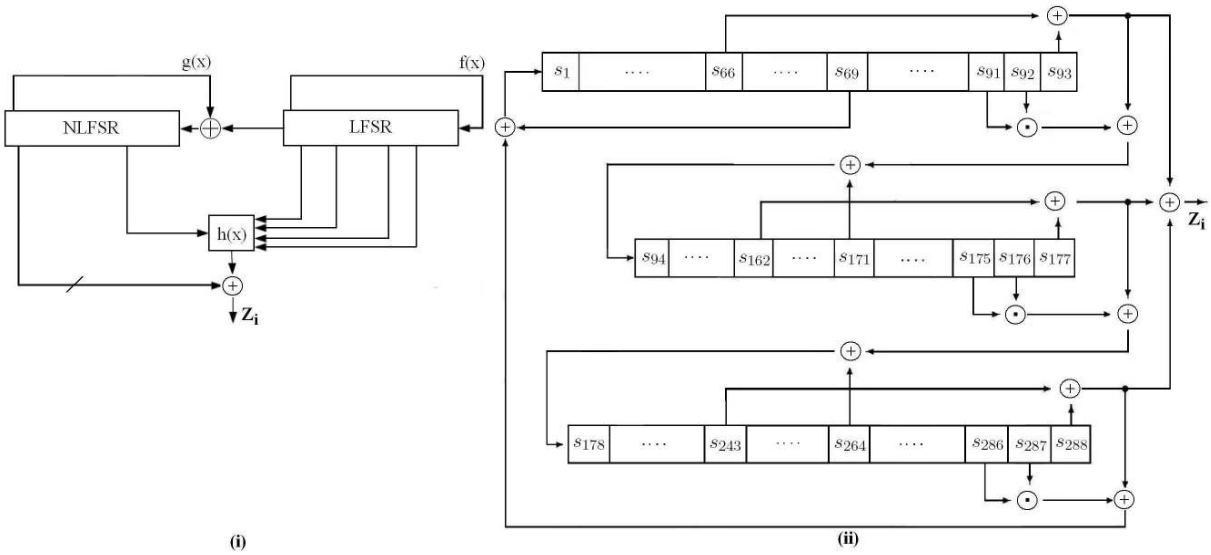

Figure 3: (i) Grain stream cipher (ii) Trivium stream cipher.

during precharge phase. Besides, this will help to cut the current spikes in the beginning of the precharge phase of each cycle. In order to get rid of the spikes, a delayed clocking mechanism is used [13], [14].

In order to increase security and speed of the initialization phase, a parallel data loading scheme is used, since in case of serial bit loading a straight forward simple power analysis attack is very likely to be successful in recovering all key bits. In parallel loading, key and IV will be loaded in the state bits after the first rising edge of the *CLK* signal. The gate level architecture of the parallel data loading scheme for standard CMOS is shown in Fig. 4. Note that in case of SABL all the wires and gates are dual rail. The area overhead is three Nand2 gates for each FlipFlop of the FSRs in the ciphers. Since all the components in the architecture need a *CLK* signal for switching from precharge phase into evaluation and vice versa, a chained buffer clock signal is needed. For the standard CMOS implementation of a the stream ciphers, standard two input Nand gates (4 Transistors), 8 transistor two input Xor gates, and the former 24 transistor, edge triggered D-FlipFlops (using eight Nand2), are used. In order to monitor all current variations, one sample has been taken every 50ps. Both simulations were run for four different 80-bit keys and IV's (64 bit IV for Grain v.1) in both SABL and standard CMOS designs. All power simulations are observed by $5MHz$ clock signal. The average power consumption per cycle was extracted by averaging the power consumption on 100 consecutive clock cycles. Then, the Mean Power Consumption (*MPC*), the Power Consumption Standard Deviation (*PCSD*), the Normalized Energy Deviation (*NED*) and Normalized Standard Deviation (*NSD*) were extracted for each simulated logic style (10). For example Supply current traces for standard CMOS design of Grain v.1 for the choice of K_2, IV_3 (in Table 1) in initialization phase is shown in Fig. 5.

$$NED = \frac{max(energy/cycle) - min(energy/cycle)}{max(energy/cycle)}, NSD = \frac{PCSD}{MPC} \tag{10}$$

In terms of transistor cost, the complete Trivium (including parallel data loading and clock buffering circuitary) required $\approx$ 23000 transistors for the SABL and $\approx$ 8500 transistors for the standard CMOS. In case of Grain v.1, $\approx$ 13500 transistors for the SABL and $\approx$ 6000 transistors for the standard CMOS are needed, confirming more than two times higher hardware cost for SABL styles. Table 1 shows the summary of final statistical power analysis results. For example in $0.13\mu m$ technology, and for K_1, IV_1, for Grain v.1, $\frac{PCSD_{SABL}}{PCSD_{SCMOS}} = 0.016$ which shows that the power consumption fluctuations of SABL implementation is nearly 1.6% of standard CMOS ($Power = Current \times Costant\ Supply\ Voltage$). This is a

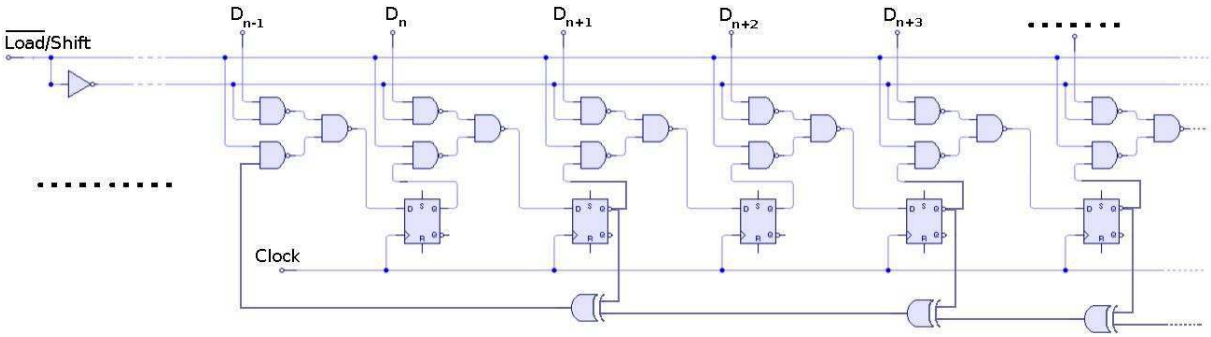

Figure 4: Parallel data loading scheme in FSRs (standard CMOS)

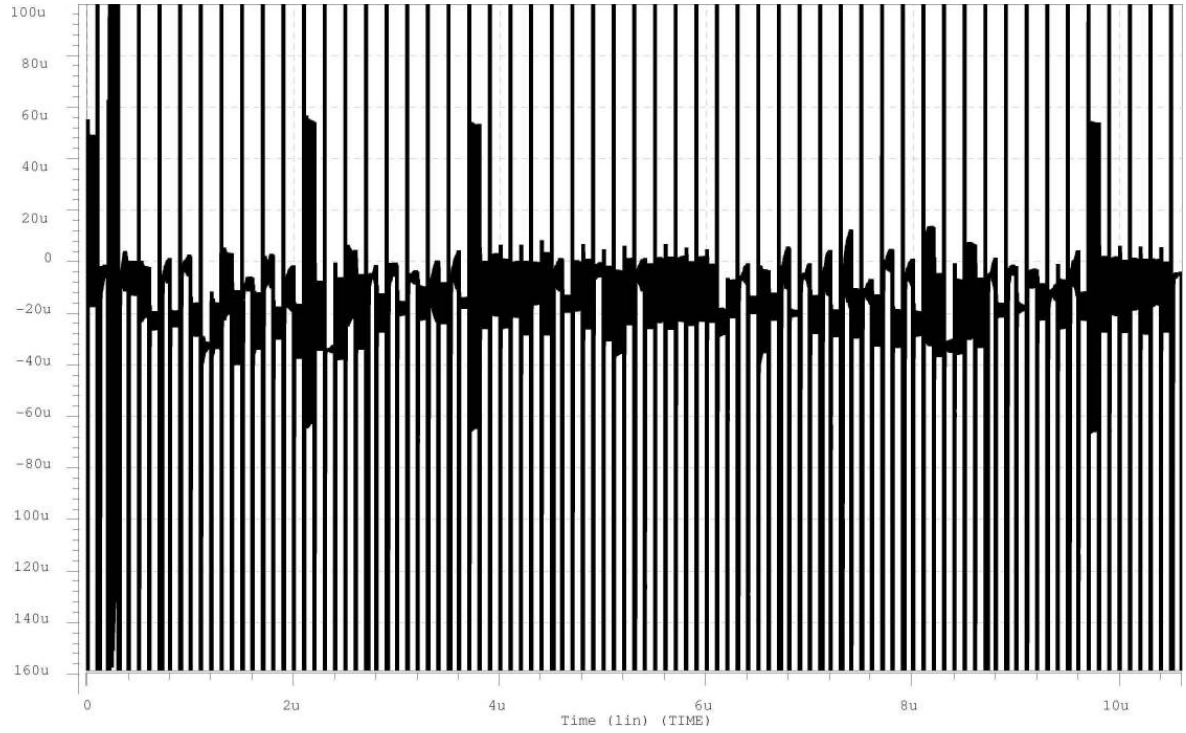

Figure 5: supply current variation of Standard CMOS design of Grain v.1 in 350nm technology

Table 1: Statistical power analysis of Trivium and Grain v.1 for different 80 bit hexadecimal key and IV's ($K_1 = AA\ldots A,\ K_2 = 80\ldots0, IV_1 = 55\ldots5, IV_2 = FF\ldots F, IV_3 = 00\ldots0, IV_4 = 11\ldots1$). [Note that in case of Grain v.1 the IV's are 64 bits]

Stream Cipher	Trivium				Grain			
Trivium	MPC [μW]	PCSD [μW]	NED	NSD	MPC [μW]	PCSD [μW]	NED	NSD
SABL	$0.35\mu m, V_{dd} = 3.3V, V_{TN} = 0.6V, V_{TP} = -0.85V$							
K_1, IV_1	949	1.2632	0.0091	0.00133	616	0.9497	0.0136	0.00154
K_1, IV_2	940	1.2469	0.0106	0.00132	605	0.9375	0.0111	0.00155
K_2, IV_3	938	1.2403	0.0089	0.00131	601	0.9318	0.0122	0.00155
K_2, IV_4	943	1.2531	0.0117	0.00133	611	0.9393	0.0120	0.00153
S-CMOS	$0.35\mu m, V_{dd} = 3.3V, V_{TN} = 0.6V, V_{TP} = -0.85V$							
K_1, IV_1	641	49.1	0.3292	0.0766	421	26.8	0.2784	0.0636
K_1, IV_2	637	19.3	0.2351	0.0303	402	18.5	0.1905	0.0460
K_2, IV_3	629	23.5	0.3139	0.0374	397	17.1	0.2187	0.0430
K_2, IV_4	635	41.7	0.2842	0.0657	415	29.6	0.3882	0.0713
SABL	$0.13\mu m, V_{dd} = 1.2V, V_{TN} = 0.4V, V_{TP} = -0.39V$							
K_1, IV_1	545	0.8435	0.0061	0.0015	378	0.7610	0.0124	0.0020
K_1, IV_2	537	0.7927	0.0054	0.0014	371	0.7424	0.0082	0.0020
K_2, IV_3	536	0.7831	0.0050	0.0014	369	0.7291	0.0079	0.0019
K_2, IV_4	541	0.8237	0.0059	0.0015	374	0.7482	0.0101	0.0020
S-CMOS	$0.13\mu m, V_{dd} = 1.2V, V_{TN} = 0.4V, V_{TP} = -0.39V$							
K_1, IV_1	337	31.20	0.8945	0.0926	258	22.50	0.7889	0.0872
K_1, IV_2	321	16.12	0.8191	0.0190	246	14.14	0.8026	0.0545
K_2, IV_3	319	17.14	0.8614	0.0537	242	12.31	0.8402	0.0509
K_2, IV_4	326	29.94	0.9218	0.0918	252	21.72	0.7924	0.0862

major improvement but still $PCSD_{SABL} \neq 0$ and very small current variations are detectable. The overall comparison between SABL and standard CMOS design for 4 different key and IV choices (Table 1) is shown in Table 2. DPA resistivity factor is calculated in (11):

$$DPA\ Resistivity \propto \frac{1}{PCSD} \tag{11}$$

One of the fundamental parameters of a cryptographic algorithm is the amount of data it can process within a given period. The total throughput of the algorithm is expressed as $Mbits/s$ and can be calculated from $T = f \times Radix$ where f is the clock frequency of the design (e.g. $5MHz$). Since Trivium throughput rate for SABL and S-CMOS designs are equal, in order to make a fair comparison, a new normalized

Table 2: Overall comparison for SABL and S-CMOS design of Trivium (All data are normalized)

Cipher	Trivium				Grain			
Logic Style	SABL	S-CMOS	SABL	S-CMOS	SABL	S-CMOS	SABL	S-CMOS
Technology	$0.13\mu m$		$0.35\mu m$		$0.13\mu m$		$0.35\mu m$	
Transistor Cost (A)	1	0.37	1	0.37	1	0.44	1	0.44
Power Consumption (P)	1	0.60	1	0.67	1	0.66	1	0.67
DPA Resistancy (DR)	1	0.0374	1	0.0435	1	0.045	1	0.043
Qualifying Factor (QF)	1	0.062	1	0.065	1	0.068	1	0.064

Qualifying Factor (QF) is defined in (12):

$$QF = \frac{DR \times T}{P \times A} \tag{12}$$

Where, A, P, and DR corresponds to transistor cost, power consumption, and DPA resistancy respectively. At the end of the simulations and data analysis we exhibited that SABL logic styles allow to significantly decrease the supply current variations of both eSTREAM circuits. But still in both designs very small current variations are detectable. As a disadvantage this could be a start of a DPA attack since the predictability of the energy variations is more critical than their amplitude. It is clear that decreasing the power consumption variations will affect all the stream cipher design components in exactly the same way, and therefore not affect the SNR. Since DPA efficiency depends on the possibility to predict the power consumption of a device in function of its input data and the value of the correlation coefficient, the attack is still theoretically feasible against SABL circuits. But these current differences are due to the presence of parasitic capacitances in the design and therefore, they cannot be predicted without a precise transistor level knowledge of the circuit. As a consequence, an attacker can only target one specific implementation and preliminarily needs to build a table containing the power consumption differences in function of the circuit input data. These informations are not usually made available to the users in full custom design. Moreover, under the assumption that we can perfectly predict and measure the power consumption, a circuit resistance is equal for any logic style. Nevertheless, in practice, measurements are not perfect and induce noise, independently of the logic style considered. This will cause a reduction of the correlation values, depending on the power consumption variances, although it is hard to evaluate and highly depends on the attacker measurement setup.

As can be seen in the Table 1 and Table 2 the DPA resistancy is improved for smaller minimum feature sized designs. Although, current variations do not follow the scaling rules. This is mainly because of clock feedthrough effect and also the former subthreshhold leakages which play a big roll in deep submicron designs.

Stream ciphers always had the reputation of efficiency in hardware. Their smaller architecture helps to use full custom design flow in order to have balanced routing of component wires. So simpler stream cipher designs would have lower design costs. Regarding design flow, Trivium has lower hardware complexity and circuit design is easier. Although Trivium has bigger architecture, timing constraints and clock distribution of Trivium are the same as Grain. Comparing resistance against DPA attacks of the two eSTREAM candidates, simulations show Grain has lower current spikes and smaller current variations. This is thanks to the higher circuit complexity of Grain which combines different current variation of gates to achieve a semi random supply current variation. Current spikes in Trivium are due to the higher number of flip flops. Another disadvantage of Trivium is its large number of iterations in initialization phase (1152 rounds) which let attackers to have more power traces.

8 Summary and Conclusions

This paper investigated the use of SABL logic to counteract power analysis attacks. In particular, an efficient DPA resistive circuit for Grain v.1 and Trivium stream ciphers have been designed and compared with their standard CMOS implementations. First we exhibited that SABL allow to significantly decrease the circuit energy variations. This is due to equal amounts of power consumption in each clock cycle of SABL gates. All implementations have been done on transistor level but in practice the cipher itself is part of a system on chip with lots of other circuits which can increase $P_{Cons.} + P_{Noise}$ in (1) to achieve lower SNR. Although SABL cannot be completely tamper resistant, this logic probably presents acceptable security margins for general applications of stream ciphers. For future work interested researchers can investigate some circuit changes in SABL styles to counteract other side channel attacks such as fault attacks to obtain more security.

Acknowledgment

Reza Ebrahimi Atani wishes to thank the Iran Telecommunication Research Center (ITRC) for their financial support (www.itrc.ac.ir).

Bibliography

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology - CRYPTO'99*, Springer-Verlag, LNCS Vol. 1666, pp. 388–397, 1999.

[2] Ch. Rechberger and E. Oswald, "Stream Ciphers and Side-Channel Analysis" *In SASC 2004 - The State of the Art of Stream Ciphers*, Brugge, Belgium, Workshop Record, pp. 320–326, Oct. 14-15, 2004.

[3] J. Lano, N. Mentens, B Preneel, and I. Verbauwhede, "Power Analysis of Synchronous Stream Ciphers with Resynchronization Mechanism" *In SASC 2004 - The State of the Art of Stream Ciphers*, Brugge, Belgium, Workshop Record, pp. 327–333, Oct. 14-15, 2004.

[4] W. Fischer, B. M. Gammel, O. Kniffler, J. Velton, "Differential Power Analysis of Stream Ciphers," *Topics in Cryptology - CT-RSA 2007*, Springer-Verlag, LNCS, Vol. 4377, pp. 257–270, 2007.

[5] M. Hell, Th. Johansson, A. Maximov, and W. Meier, "Grain - A Stream Cipher for Constrained Environments," 2006, eSTREAM project website.

[6] C. De Canniere, and B. Preneel, "Trivium Specifications," 2005, eSTREAM project website.

[7] T. Seko, A. Nakamura, and T. Kikuno, "Measurement of glitches based on variable gate delay model using VHDL simulator," *Asia-Pacific Conference on Circuits and Systems,* Nov. 1998, PP. 767 – 770.

[8] B. Gierlichs et. al., "Susceptibility of eSTREAM Candidates towards Side Channel Analysis," *SASC 2008,* Switzerland, Feb. 13-14, 2008, Workshop Record, pp. 320 – 326.

[9] K. Tiri, and I. Verbauwhede, "Charge recycling sense amplifier based logic: securing low power security ICs against DPA" *30th European Conference on Solid-State Circuits,* 21-23 Sept. 2004, pp. 179 – 182.

[10] 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," *28th European Solid State Circuits Conference*, IEEE Press, pp. 403 – 406, , 24-26 Sep. 2002.

[11] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards,* Springer, 2007.

[12] S. Babbage et. al., *The eSTREAM Portfolio,* April 2008, eSTREAM project website.

[13] R.E. Atani, W. Meier, S. Mirzakuchaki, and S.E.Atani, "Design and Implementation of DPA Resistive Grain-128 Stream Cipher Based on SABL Logic", *International Journal of Computers, Communications & Control*, Vol. III (supl. issue), pp. 293 – 298, 2008.

[14] R.E. Atani, W. Meier, S. Mirzakuchaki, and S.E.Atani, "Design and simulation of a DPA resistive circuit for Trivium stream cipher based on SABL styles" *Mixdes 2008,* 19-21 June. 2008, pp. 203 – 208.

[15] K. Tiri, and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation" *DATE 2004,* 2004, pp. 246–251.

[16] M. Bucci, L. Giancane, R.o Luzzi, and A. Trifiletti, "Three-Phase Dual-Rail Precharge Logic" *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 232–241.

[17] T. Popp, and S. Mangard, "Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints" *In Cryptographic Hardware and Embedded Systems CHES 2005,* Vol. 3659 of LNCS, Springer, 2005, pp. 172–186.

[18] Z. Chen, and Y. Zhou, "Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage," *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 242–254.

[19] D. Suzuki, and M. Saeki, "Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style" *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 255–269.

[20] P. Schaumont, and K. Tiri, "Masking and Dual-Rail Logic Dont Add Up," *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 95–106.

[21] B. Gierlichs, "DPA-Resistance Without Routing Constraints?" *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 107–120.

Reza Ebrahimi Atani
Electrical Engineering Department
Iran University of Science and Technology (IUST),
Narmak,16846, Tehran, Iran.
E-mail: rebrahimi@iust.ac.ir

Sattar Mirzakuchaki
Electrical Engineering Department
Iran University of Science and Technology (IUST),
Narmak,16846, Tehran, Iran.
E-mail: m-kuchaki@iust.ac.ir

Shahabaddin Ebrahimi Atani
Mathematics Department
University of Guilan
P.O.Box 1914, Rasht, Iran.
E-mail: ebrahimi@guilan.ac.ir

Willi Meier
IAST Institute
FHNW
CH 5210, Windisch, Switzerland.
E-mail: willi.meier@fhnw.ch