Combined Side-Channel and Fault Analysis Attack on Protected Grain Family of Stream Ciphers

Abhishek Chakraborty, Bodhisatwa Mazumdar and Debdeep Mukhopadhyay
Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India.
Email: $\{abhishek.chakraborty, bodhisatwa, debdeep\}$@$cse.iitkgp.ernet.in$

***Abstract*—In this paper, we first demonstrate a new Differential Power Analysis (DPA) attack technique against the Grain family of stream ciphers (Grain v1 and Grain-128) by resynchronizing the cipher multiple times with the same value of the secret *key* and randomly generated different initialization vectors (IVs). Subsequently, we develop a combined side channel and fault analysis attack strategy targeting various fault attack countermeasures for the Grain cipher family. We considered clock glitch induced faults occurring in practice for a hardware implementation of the cipher to devise our novel attack technique. Our proposed combined attack strategy works well even if the *useful* ciphertexts are not available to the adversary. Further, the power trace classifications of a Grain cipher implementation on SASEBO G-II standard side channel evaluation board is shown in order to validate our proposed attack against the cipher. The captured power traces were analyzed using Least Squares Support Vector Machine (LS-SVM) learning algorithm based multiclass classifiers to classify the power traces into the respective Hamming distance (HD) classes. To extract power samples with high information about HD classes, Signal-to-noise ratio (SNR) metric was chosen for feature selection. The experimental results of power trace classifications of test set showed a high success rate of 98% when the five largest SNR sample instants over a clock cycle were chosen as features. Our proposed attack strategy can also be extended to other stream cipher designs based on Fibonacci configured shift registers.**

***Index Terms*—Grain stream cipher, Differential Power Analysis, Fault Attack Countermeasures, SASEBO G-II board, Clock glitch, Least Squares Support Vector Machine**

I. Introduction

CRYPTOGRAPHIC algorithms are extensively used in the modern era to ensure message confidentiality and integrity, secure computing, authentication of the communicating parties, digital signatures and several other applications. Traditionally, the robustness of cryptographic primitives has been determined using mathematical models and statistical analysis. However, the *real life implementations* of these ciphers can be studied and analyzed to mount Side Channel Attacks (SCAs) [1]–[3]. By effectively exploiting the unintentional leakage of information into the environment from the physical implementations, system breakdown can be achieved with a relatively less computational cost and in a shorter time compared to the conventional mathematical cryptanalysis. Fault analysis attacks are active side channel attacks in which an adversary induces erroneous computations and subsequently analyzes them to retrieve the secret key.

Stream ciphers are an important class of symmetric ciphers used extensively for encryption by hardware-based cryptographic systems. They are popular because of their simplicity, efficiency and performance. Secure realizations of stream ciphers, which are tolerant to both SCAs and fault analysis attacks is thus a crucial issue.

Many recent stream cipher designs are based on one or more Nonlinear Feedback Shift Registers (NLFSRs) along with Linear Feedback Shift Registers (LFSRs). NLFRs constitute a larger class and provide higher levels of security against algebraic attacks compared to LFSRs [4]. The Grain family of stream ciphers [5] are a part of the final hardware portfolio of eSTREAM project. The structure of Grain cipher consists of two Fibonacci configured shift registers, a LFSR and a NLFSR, along with a nonlinear output Boolean function. A DPA attack against Grain v1 is presented in [6]. In [7], although the authors mention about a possibility of an optimized known DPA attack by building templates, no explicit attack methodology is outlined. There are also several Differential Fault Analysis (DFA) attacks on Grain family of stream ciphers reported in literature [8]–[11].

In this work, we propose a new DPA attack strategy against Grain stream cipher which requires a low number of power traces corresponding to multiple resynchronizations of the cipher with the same secret *key* and different known IVs. Our attack technique does not demand any selection of specific IVs as required by the attack proposed in [6]. As a second contribution of this work, we also propose a novel combined side channel and fault analysis attack strategy against different fault attack countermeasures for Grain. Our developed algorithm to mount such a combined attack on the stream cipher during the initialization phase considers the biased nature of clock glitch induced faults occurring in its actual hardware design. In literature, such combined fault and side channel attacks have been applied against public key cryptosystems like RSA [12] and block ciphers like AES [13], [14].

We implemented Grain cipher on a SASEBO-GII board [15] in order to validate our proposed attack strategy against it. The Least Squares Support Vector Machine (LS-SVM) learning algorithm [16] was used as an analyzer of power traces collected from the hardware implementation of the cipher.

The organization of the paper is as follows: In section II, we present some basic concepts regarding power analysis of Feedback Shift Registers. Section III consists of a detailed description of our proposed DPA attack on Grain stream cipher. In section IV, we propose a new combined side channel and fault analysis attack on fault attack resistant countermeasures of the cipher. The experimental results of classifications of power traces using LS-SVM model for a

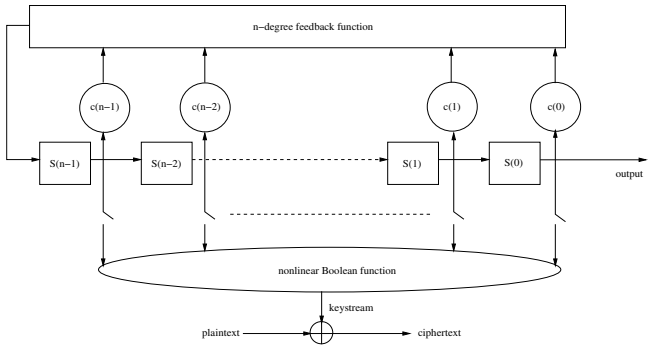

Fig. 1. An n-stage feedback shift register with a nonlinear filter function

Grain cipher implementation are reported in section V. Section VI concludes the paper.

II. Preliminaries

Shift registers can generate sequences at high speeds and can be easily implemented in both software and hardware. The contents of an n-stage binary shift register is referred to as its *state*. In this paper, we denote the states of a LFSR and a NLFSR at time instant t by ST_{LFSR_t} and ST_{NLFSR_t} respectively. We also assume that if the number of toggles in the state of a shift register in cycle t differs from that in cycle $t+1$, then the power dissipated by the register in the two cycles are also different, else they are same.

A. Power Analysis Attack on Fibonacci LFSR

A general structure of Fibonacci configured LFSR at time instant $t-1$ is shown in Fig.1. The state at time t is computed by right shifting the LFSR by one bit. The value shifted into the leftmost stage, denoted by $S(n)$, is a linear combination of the bit values of the n-stages of the LFSR as defined by its feedback polynomial. A power analysis attack on a standalone Fibonacci LFSR implementation is presented in [17].
If $ST_{LFSR_{t-1}} = (S(n-1), \cdots, S(0))$ then,
$ST_{LFSR_t} = (S(n), S(n-1), S(n-2), \cdots, S(1))$
where, $S(n) = c(n-1)S(n-1) \oplus c(n-2)S(n-2) \oplus \cdots \oplus c(0)S(0)$, $c(i) \in \{0,1\}$, $\forall i$, $1 \leq i \leq n$.
Let us denote the Hamming weight of a bit vector **A** by HW(**A**). Then the Hamming distance (HD) between the successive states of the LFSR at time instants $t-1$ and t (represented as HD_t) and that between the consecutive LFSR states at time instants t and $t+1$ (represented as HD_{t+1}) are calculated as follows:

$$\begin{aligned} HD_t &= HW(S(n) \oplus S(n-1), S(n-1) \oplus S(n-2), \\ &\cdots \quad , S(1) \oplus S(0)) \end{aligned} \tag{1}$$

$$\begin{aligned} HD_{t+1} &= HW(S(n+1) \oplus S(n), S(n) \oplus S(n-1), \\ &\cdots \quad , S(2) \oplus S(1)) \end{aligned} \tag{2}$$

Let PD_t denote the difference between HD_t and HD_{t+1}. From equations (1) and (2) we get the following:

$$\begin{aligned} PD_t &= HD_t - HD_{t+1} \\ &= HW(S(0) \oplus S(1)) - HW(S(n+1) \oplus S(n)) \\ &= \{0,1\} - \{0,1\} \\ &= \{-1,0,1\} \end{aligned} \tag{3}$$

TABLE I
Relations among the NLFSR state variables with PD values

PD_t	Relationship among state variables
$+1$	$L(0) \oplus L(1)$=1 and $L(n+1) \oplus L(n)$=0
-1	$L(0) \oplus L(1)$=0 and $L(n+1) \oplus L(n)$=1
0	$L(0) \oplus L(1) = L(n+1) \oplus L(n)$

From equation (3) it is evident that when the HDs in successive clock cycles are equal, the difference between power dissipations of the LFSR hardware implementation in corresponding clock cycles is very small (ideally zero), else the difference will be of a significant magnitude. The proposed attack technique in [17] utilizes Berlekamp-Massey algorithm to determine the initial state of a Fibonacci LFSR. A similar attack strategy against a standalone Galois configured LFSR is presented in [18].
Though LFSRs have been used extensively in many cryptographic applications, standalone implementations of such shift registers are not very secure. NLFSRs are known to provide higher levels of security against cryptanalytic attacks compared to LFSRs. Many modern stream cipher designs incorporate NLFSRs as primary building blocks. Therefore, in order to mount power analysis attacks on such stream ciphers, an adversary must be develop strategies which can effectively exploit the power side channel of an NLFSR implementation.

B. Power Analysis Attack on Fibonacci NLFSR

The structure of a Fibonacci NLFSR is similar to that of a Fibonacci LFSR with the only difference being that the feedback polynomial is nonlinear in the former case.
If $ST_{NLFSR_{t-1}} = (L(n-1), L(n-2), \cdots, L(0))$ then,
$ST_{NLFSR_t} = (L(n), L(n-1), L(n-2), \cdots, L(1))$
where, $L(n)$ denotes the value shifted into the leftmost stage of the NLFSR at time t. $L(n)$ is a nonlinear combination of the contents of the NLFSR stages as defined by its feedback connection polynomial.
A power analysis framework for a Fibonacci NLFSR based on differences of consecutive HDs is presented in [19].
Setting up equations similar to (1) and (2) we get the following expression of PD_t for a NLFSR:

$$\begin{aligned} PD_t &= HD_t - HD_{t+1} \\ &= HW(L(0) \oplus L(1)) - HW(L(n+1) \oplus L(n)) \\ &= \{0,1\} - \{0,1\} \\ &= \{-1,0,1\} \end{aligned} \tag{4}$$

The various values of PD_t correspond to different relations among the state variables $L(0)$, $L(1)$, $L(n)$ and $L(n+1)$ as shown in Table I.
From the table, it is to be noted that for $PD_t = 0$ the following two cases apparently cannot be distinguished.

- CASE I: $L(0) \oplus L(1) = L(n+1) \oplus L(n) = 0$
- CASE II: $L(0) \oplus L(1) = L(n+1) \oplus L(n) = 1$

In [19], a technique to distinguish between the above two cases for a L-bit NLFSR is proposed by considering a nonzero PD_{t+nL}, for the least integral value of n. However, an apriori determination of the number of additional clock cycles needed for such an analysis is not possible. In our proposed DPA

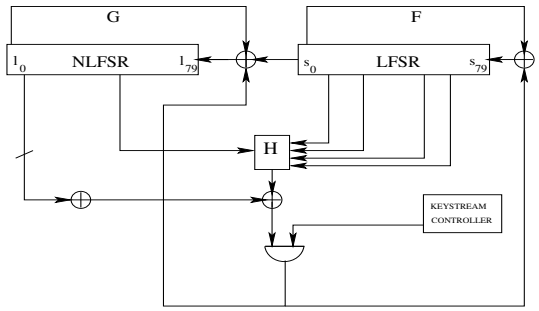

Fig. 2. Structure of Grain stream cipher

attack on Grain v1 stream cipher, described in section III, the above ambiguous case of NLFSR PD_t is not considered and thus eliminates the need to measure power consumptions of the cipher implementation in additional clock cycles.

III. DPA Attack on Grain stream cipher family

In this section, we first provide a brief description of Grain stream cipher [5] followed by a detailed explanation of our proposed DPA attack against the cipher. In [6], the authors presented a power analysis attack against Grain v1 in three different phases. The first two phases are DPA attacks with selectively chosen IVs to recover 34 and 16 bits of the key respectively. The last phase involves an exhaustive search for the remaining 30 bits. However, the selection of specific IVs makes the attack limited to certain favorable circumstances. In this work, we present a new DPA attack against Grain cipher (Grain v1 and Grain-128) which does not require any such careful selection of IVs. Moreover, our attack strategy offers a trade-off between the number of resynchronizations of the cipher and the computational of an exhaustive search for the remaining undetermined *key* bits.

A. Grain Stream Cipher

Grain family of stream ciphers is targeted towards hardware implementations requiring limited gate count, power consumption and memory. It is a synchronous stream cipher based on a LFSR, a NLFSR and a nonlinear output function. The internal state of Grain cipher is x bits ($x = 160$ for Grain v1 and $x = 256$ for Grain-128) and it supports a *key* of m bits ($m = 80$ for Grain v1 and $m = 128$ for Grain-128) and initialization vectors (IVs) of size v bits ($v = 64$ for Grain v1 and $v = 96$ for Grain-128). The m-bit *key* resides in the NLFSR bits $l_0, l_1, \ldots, l_{m-1}$ while the v-bit IVs are uploaded in the LFSR bits $s_0, s_1, \ldots, s_{v-1}$. The remaining $m - v$ bits of the LFSR $s_v, s_{v+1}, \ldots, s_{m-1}$ are filled with all ones. The structure of Grain v1 is shown in Fig.2. The expressions of the feedback functions for both the shift registers and the output function H can be found in [5].

B. Hypothetical power model for the attack

We considered the Hamming distance power model to estimate the power consumption of a CMOS based implementation of Grain v1. The overall power consumption of the cipher is given by the following expression as stated in [6].

$$P = P_G + P_F + P_H + \sum_{i=0}^{m-1} P_{N_i} + \sum_{i=0}^{m-1} P_{L_i} + \sigma$$

where, P_G, P_F, P_H, P_N, P_L and σ denote the power consumptions of the NLFSR feedback function, LFSR feedback function, nonlinear output function, the NLFSR stages, the LFSR stages, and implementation independent noise element respectively.

In our attack technique, we only consider the components $\sum_{i=0}^{m-1} P_{N_i}$ and $\sum_{i=0}^{m-1} P_{L_i}$ as they are major contributors to the total power consumption of the circuit [20]. All the remaining factors are treated as contributors of sources of noise.

C. Entropy Reduction of PD sequence of the LFSR section from known IV

The m-bit secret *key* and the v-bit known IV of a Grain stream cipher are loaded into the NLFSR and the LFSR respectively. Let us denote the initial contents of the LFSR bits $s_0, s_1, ..., s_{m-1}$ as $IV(0), \ldots, IV(v-1), s'(v), \ldots, s'(m-1)$ and the NLFSR bits $l_0, l_1, ..., l_{m-1}$ as $K(0), K(1), ..., K(m-1)$, where $(s'(v), \ldots, s'(m-1)) = \{1\}^{m-v}$.

Corresponding to the second clock cycle following the loading of the shift registers of Grain cipher at time t (say), we get the following expression for $PD_{LFSR_{t+1}}$:

$$\begin{aligned} PD_{LFSR_{t+1}} &= HD_{LFSR_{t+1}} - HD_{LFSR_{t+2}} \\ &= HW(IV(0) \oplus IV(1)) - HW(s_{m+1} \oplus s_m) \end{aligned} \quad (5)$$

where, HD_{LFSR} stands for HD between two consecutive states of the LFSR and PD_{LFSR} signifies the corresponding differences of HDs in successive clock cycles. s_m and s_{m+1} denote the values shifted into the LFSR bit s_{m-1} in the first and second clock cycles respectively, after the LFSR has been initialized. Since s_m and s_{m+1} are dependent on the internal state of the cipher in the previous clock cycle, we cannot ascertain whether $HW(s_{m+1} \oplus s_m)$ is 0 or 1 without knowing the content of the NLFSR. However, we know the value of $HW(IV(0) \oplus IV(1))$ since the IV is known. Therefore, we can classify $PD_{LFSR_{t+1}}$ into the following two cases depending upon the adjacent bits of IV.

- CASE Ia: If $HW(IV(0) \oplus IV(1))$ is 1 then, $PD_{LFSR_{t+1}}$ can take a value from the set $\{0, 1\}$.
- CASE Ib: If $HW(IV(0) \oplus IV(1))$ is 0 then, $PD_{LFSR_{t+1}}$ can take a value from the set $\{0, -1\}$.

This reduces the uncertainty (entropy) of $PD_{LFSR_{t+1}}$ from three to two possible values. This is subsequently exploited in the ensuing attack strategy.

D. Proposed attack strategy

We targeted the *initialization phase* of Grain stream cipher to recover the m-bit *key*. Each shift register configured in Fibonacci fashion can take a PD value from the set {-1,0,1} as shown in equations (3) and (4). We denote the overall PD value of Grain as PD_{total}, which is the sum of PD_{LFSR} and PD_{NLFSR}. Therefore PD_{total} can take any value from the set {-2,-1,0,1,2}. The possible combinations of PD_{total}, PD_{LFSR} and PD_{NLFSR} values are illustrated in Table II.

TABLE II
RELATIONS AMONG PD_{total}, PD_{LFSR}, AND PD_{NLFSR}

PD_{total}	PD_{LFSR}	PD_{NLFSR}
$+2$	$+1$	$+1$
$+1$	$+1$	0
	0	$+1$
0	0	0
	-1	$+1$
	$+1$	-1
-1	-1	0
	0	-1
-2	-1	-1

1) Uniquely determining PD_{LFSR} and PD_{NLFSR} : We considered the power consumption in the first m consecutive clock cycles after the resynchronization of Grain cipher to get the following sequence of $m-1$ consecutive PD_{LFSR} symbols:

$$\begin{aligned} PD_{LFSR_{t+1}} &= HW(IV(0) \oplus IV(1)) - HW(s_{m+1} \oplus s_m) \\ PD_{LFSR_{t+2}} &= HW(IV(1) \oplus IV(2)) - HW(s_{m+2} \oplus s_{m+1}) \\ &\vdots \\ PD_{LFSR_{t+v-1}} &= HW(IV(62) \oplus IV(63)) - HW(s_{m+v-1} \oplus s_{m+v-2}) \\ PD_{LFSR_{t+v}} &= -HW(s_{m+v} \oplus s_{m+v-1}) \\ &\vdots \\ PD_{LFSR_{t+m-1}} &= -HW(s_{2m-1} \oplus s_{2m-2}) \end{aligned} \quad (6)$$

where, s_{m-1+a} denotes the value shifted into the LFSR stage s_{m-1} in the a-th clock cycle after resynchronization for $1 \leq a \leq m$. In equation set (6), for the $PD_{LFSR_{t+k}}$ symbols, where $v \leq k \leq m-1$, the first component is $HW(1 \oplus 1) = 0$, as $(s'(v), \ldots, s'(m-1)) = \{1\}^{m-v}$. Each of the $PD_{LFSR_{t+j}}$ symbols, where $1 \leq j \leq v-1$, is either from the set $\{0, 1\}$ or from the set $\{0, -1\}$ depending upon consecutive IV bit contents. On the other hand, each of the $PD_{LFSR_{t+k}}$ symbols, where $v \leq k \leq m-1$, will only be from the set $\{0, -1\}$. Hence, using the knowledge of IV we can determine whether a $PD_{LFSR_{t+i}}$ symbol belongs to the set $\{0, 1\}$ or $\{0, -1\}$, where $1 \leq i \leq m-1$. However, the additional knowledge of corresponding PD_{total} symbols may lead to exact determination of PD_{LFSR} symbols in equation (6), which in turn will uniquely determine the corresponding PD_{NLFSR} values. Using Table II we get the following cases:

- CASE IIa: If $PD_{total} = \pm 2$, both the corresponding PD_{LFSR} & PD_{NLFSR} symbols are ± 1.
- CASE IIb: If $PD_{LFSR} \in \{0, 1\}$ and $PD_{total} = -1$ then, $PD_{LFSR} \neq 1$, i.e, $PD_{LFSR} = 0$ and $PD_{NLFSR} = -1$.
- CASE IIc: If $PD_{LFSR} \in \{0, -1\}$ and $PD_{total} = 1$ then, $PD_{LFSR} \neq -1$, i.e, $PD_{LFSR} = 0$ and $PD_{NLFSR} = 1$.

For the remaining possible cases we cannot uniquely classify the shift register PD values, e.g. when $PD_{LFSR} \in \{0, -1\}$ and $PD_{total} = -1$, $PD_{LFSR} = 0$ implies $PD_{NLFSR} = -1$ whereas $PD_{LFSR} = -1$ implies $PD_{NLFSR} = 0$.

2) Retrieving the secret key from PD_{NLFSR} : Similar to the LFSR section, we can construct a set of PD symbols for the Grain stream cipher NLFSR as follows:

$$\begin{aligned} PD_{NLFSR_{t+1}} &= HW(K(0) \oplus K(1)) - HW(l_{m+1} \oplus l_m) \\ PD_{NLFSR_{t+2}} &= HW(K(1) \oplus K(2)) - HW(l_{m+2} \oplus l_{m+1}) \\ &\vdots \\ PD_{NLFSR_{t+m-1}} &= HW(K(m-2) \oplus K(m-1)) - HW(l_{2m-1} \oplus l_{2m-2}) \end{aligned} \quad (7)$$

where, l_{m-1+a} denotes the value shifted into the NLFSR stage l_{m-1} in the a-th clock cycle after resynchronization for $1 \leq a \leq m$. For a given IV, some of $PD_{NLFSR_{t+j}}$ can be uniquely determined depending upon the tuple, $\{PD_{total_{t+j}}, PD_{LFSR_{t+j}}\}$, where $1 \leq j \leq m-1$. To get all the PD_{NLFSR} symbols uniquely, one may have to resynchronize the cipher with multiple IVs. The steps for generating the PD_{NLFSR} symbols are explained in Algorithm. 1. A point to note here is that all the uniquely determined PD_{NLFSR} symbols are nonzero as shown in CASE IIa, IIb, and IIc.

Algorithm 1: Determining PD_{NLFSR} sequence

Input: PD_{total} sequence
Output: $PD_{NLFSR_{t+1}}, \ldots, PD_{NLFSR_{t+m-1}}$

1 Initialize Grain cipher with a new distinct IV and the same *key*.
2 Clock the cipher for m consecutive clock cycles and obtain the sequence of $m-1$ consecutive PD_{total} symbols.
3 From equation (6), CASE IIa, IIb, and IIc, determine the PD_{LFSR} symbols between instants $t+1$ and $t+m-1$ which can be uniquely determined from the set {-1,0,1}.
4 For every uniquely determined PD_{LFSR} symbol, obtain the corresponding unique nonzero value of PD_{NLFSR} symbols between instants $t+1$ and $t+m-1$.
5 Repeat Steps 1, 2, 3, and 4 until all the equations in (7) are generated.

If all the equations in (7) are uniquely identified, we can use Table I to construct the following set of equations depicting the relationships among the adjacent *key* bits, which was initially loaded into the NLFSR:

$$\begin{aligned} K(0) \oplus K(1) &= D_1 \\ K(1) \oplus K(2) &= D_2 \\ &\vdots \\ K(m-2) \oplus K(m-1) &= D_{m-1} \end{aligned}$$

where $D_i \in \{0, 1\}$, $1 \leq i \leq m-1$, stands for the determined cases. Therefore, there are only two such sets of $m-1$ equations corresponding to each guess of key bit $K(0)$. A DPA attack can be mounted against Grain cipher following Algorithm 2. Initially each of the targeted PD_{NLFSR} values (represented by an array 'valid_PD_{NLFSR}') are unknown, hence all elements of the array are assigned 0. An element will be assigned 1 when the corresponding XOR relation is uniquely determined from the information gained from either the favourable cases of power or combined analysis attack strategy. Hence, we can recover the secret *key* by measuring power consumption of the cipher implementation during its *initialization phase* for multiple resynchronizations of Grain with different IVs.

E. Reduction of classification errors

In actual power traces, large influences of various noise elements might often lead to wrong classifications of the aforementioned PD_{NFLSR} values and thereby making the attack fail. In order to successfully launch our proposed attack

on Grain v1, an adversary should minimize such classification errors as much as possible. In iterations of Steps 1, 2, 3, and 4 of Algorithm 1 some of the determined PD_{NFLSR} symbols may get repeated in some cases of subsequent iterations with new IVs, while for the remaining cases we get additional unique classifications between instants $t+1$ and $t+m-1$. Therefore, corresponding to every PD_{NLFSR} symbol we get a vector, whose each point denotes whether the PD_{NLFSR} symbol is uniquely determined or not for a particular IV. The length of such a vector is equal to the number of IVs used to resynchronize Grain stream cipher. These vectors are used to reduce classification errors (if any) by taking the majority of the determined points to ascertain the corresponding PD_{NLFSR} symbols. Therefore, we consider these vectors as an effective measure for correct classifications of different PD_{NLFSR} values obtained from real power traces.

F. Estimation of the number of IVs required

In this subsection, we present an estimation of the number of IVs required to resynchronize Grain v1 in order to determine the initial state of the NLFSR using Algorithm 2. The above mentioned problem of interest can be mapped to "The Coupon Subset Collection Problem" [21] as follows: Let $coupon_i$ denote a uniquely determined $PD_{NLFSR_{t+i}}$ value, where $1 \leq i \leq m-1$. For every resynchronization of the cipher we obtain a random subset of the coupons depending on the number of cases the PD_{NLFSR} values can be uniquely obtained from CASE IIa, IIb, and IIc. Let X denote the number of resynchronizations required until every element of the set $\{coupon_1, coupon_2, \ldots, coupon_{m-1}\}$ is contained at least once in a randomly obtained subset. Let us also assume the condition that if there are k types of coupon in a randomly chosen subset, the chosen set of size k is equally likely to be any one of the $\binom{m-1}{k}$ subsets of size k. Let ρ_k be the probability that the random subset is of size k. Then, we get the following expected value of X:

$$E[X] = \sum_{j=1}^{m-1} (-1)^{j+1} \binom{m-1}{j} / \{1 - \sum_{k=1}^{m-1} \rho_k \binom{m-1-j}{k} / \binom{m-1}{k}\}$$

From the numerical examples reported in [21], we can conclude that for a *general case* of randomly obtained subsets with resynchronizations of Grain v1, the expected number of different IVs required for our attack strategy to retrieve the entire secret *key* is less than a hundred.

A trade-off can be achieved between the number of times for which the cipher is rekeyed with new distinct IVs and the computation cost for the exhaustive search to determine the remaining *key* bits. If the number of rekeying attempts with distinct IVs is less, then the search key-space also increases significantly. Therefore, depending on the computational power of the adversary, our proposed attack technique can be mounted on Grain with a suitable number of IVs.

IV. Attacks on proposed fault attack countermeasures on Grain family

In this section, we show how our proposed power analysis attack strategy can be utilized to mount attacks against some reported fault tolerant countermeasures for the Grain family of stream ciphers. Depending upon the nature of countermeasures and the fault model assumed, some of the schemes can be broken using only side channel information or a combination of side channel and fault analysis techniques.

Algorithm 2: Power analysis attack algorithm against Grain stream cipher

Input: *CASEs IIa, IIb, IIc : power favourable cases*, Expected number of randomly generated IVs : NUM_IV
Output: m-bit *key* /* m=80 for Grain v1, m=128 for Grain-128*/

for $num = 1$ *to* $m-1$ **do**
  valid_$PD_{NLFSR}[num] \leftarrow 0$
end
for *each* $IV_{num} \in NUM\_IV$ **do**
  $NLFSR \leftarrow key$
  $LFSR \leftarrow [IV_{num}, \{1\}^i]$ // i=16 for Grain v1, i=32 for Grain-128
  Run Grain cipher for m clock cycles and capture the power trace.
  for $clock\_cycle_{num} = 1$ *to* $m-1$ **do**
    if *(*$PD_{total}[clock\_cycle_{num}] \in power\ favourable\ cases$*)* **then**
      valid_$PD_{NLFSR}[clock_cycle_{num}] \leftarrow 1$
    end
  end
end
if *(* $\sum_{i=1}^{m-1}$ valid_PD_{NLFSR}[i] = $m-1$) **then**
  Guess $key(0) = 0$ and get Key guess 1.
  Generate keystream 1.
  Guess $key(0) = 1$ and get Key guess 2.
  Generate keystream 2.
  Match keystreams 1 and 2 with the actual keystream to obtain the correct *key* guess
end
else
  Perform exhaustive search for the remaining *key* bits
end

In [8], the authors have outlined the use of an additional LFSR to thwart fault analysis attacks on Grain-128. The suggested countermeasure requires the use of two LFSRs with identical register contents. Therefore, according to our proposed attack technique PD_{total} for such a design can take any value from the set $\{-3, -2, -1, 0, 1, 2, 3\}$. The possible combinations of PD_{total}, PD_{LFSR_1}, PD_{LFSR_2} and PD_{NLFSR} values, such that $PD_{LFSR_1} = PD_{LFSR_2}$,are illustrated in Table III. From the table, it can be easily observed that except for the cases where $PD_{total} \in \{-1, 1\}$, the value of PD_{NLFSR} can be uniquely classified. Now, with the additional knowledge of IVs, an adversary can uniquely determine values of PD_{NLFSR} if the adjacent bit values of the IVs lie in the favourable cases as outlined in section III. Therefore, even with the incorporation of such redundant LFSR in the cipher design, the entire secret key can be recovered using our proposed power analysis strategy.

TABLE III
Relations among PD_{total}, PD_{LFSR_1}, PD_{LFSR_2} and PD_{NLFSR}

PD_{total}	PD_{LFSR_1}	PD_{LFSR_2}	PD_{NLFSR}
$+3$	$+1$	$+1$	$+1$
$+2$	$+1$	$+1$	0
$+1$	0	0	$+1$
	$+1$	$+1$	-1
0	0	0	0
-1	0	0	-1
	-1	-1	$+1$
-2	-1	-1	0
-3	-1	-1	-1

In [9], the authors have stated that incorporating higher degree feedback and output functions will enhance the complexity of their proposed DFA attack. However, the complexity of our proposed DPA attack technique remains unaltered even if higher degree feedback functions are considered.

A. Combined side channel and fault analysis on Grain

In [10], the authors have proposed the use of *affine differential resistant* output Boolean function for Grain v1 to make it resistant against fault analysis attacks even if the IV is public. Our proposed DPA attack against Grain v1 is independent of the output filter function and hence such a fault attack countermeasure is vulnerable to passive side channel attacks. However, the authors have considered a fault model where the cipher can be reinitialized several times with the same key-IV pair. On the other hand, our proposed power analysis attack technique against the Grain family of stream ciphers requires the use of multiple IVs. The use of a single key-IV pair to resynchronize the cipher multiple times might lead to a recovery of only few key bits using our power analysis strategy. To develop an attack under such a fault model, we injected clock glitch induced faults on a hardware implementation of the cipher to get a notion of the nature of faults occurring in practice. Thereafter, we developed a combined side channel and fault analysis attack against the Grain cipher family such that the key search space is significantly reduced.

1) Practical Faults: Faults can be injected in a register, which stores data or state in a hardware implementation using several techniques: ranging from low cost methods like power spiked, clock glitches to costly methods like optical injections via lasers, etc. Although a very powerful attack model, the practicality of a Differential Fault Analysis (DFA) depends on the cost of the setup and also on the type of faults which actually occur in a practical set-up.

Although several research work on DFA on stream ciphers have been performed, they have not been supported with real life experiments, as performed on other category of ciphers, like the Advanced Encryption Standard (AES). This lack of support with real experiments, have resulted in the absence of information on which fault models are practical and occurs in real life. We considered a low-cost fault injection technique using clock glitches to induce setup time violations on a Grain-128 implementation.

To the best of our knowledge, we report for the very first time the actual chip results for a fault attack on any stream cipher. An input clock was provided to a Grain-128 Spartan-3A (XC3S400A) FPGA implementation from an external function generator. A fast clock of 20 times the frequency compared to the input slow clock was used to introduce a clock glitch at a fixed PRGA round. The fast clock was derived from the slow clock using a Xilinx Digital Clock Manager (DCM) module in the design and the states of the registers were monitored using Chipscope Pro 12.3 analyzer. We obtained the correct ciphertext (corresponding to the fault free Grain-128 internal state) for input slower clock frequencies up to 7 MHz or fast clock frequencies up to 140 MHz. We gradually increased the input slower clock frequency in steps of 0.1 MHz and captured the corresponding states of the registers at each step. The number of attempts to inject a fault at each step was 1024. In **Fig. 3**, we plot the nature of the induced faults with respect to the frequency of the fast clock for different *key-IV* pairs.

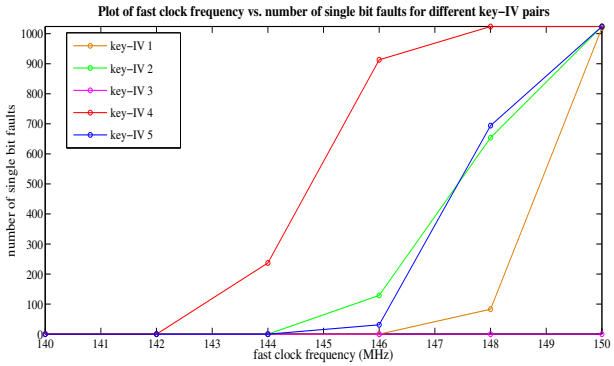

Fig. 3. Fast clock frequency vs. number of single bit faults

The faults observed were all single bit ones and in each case the faults affected only the 128^{th} bit of the NFSR ($bitNLFSR_{MSB}$) due to clock glitch introduced set up time violations. This is because the critical path of Grain-128 is through the NLFSR feedback [22]. From the experimental results we conclude that the single bit faults (if injected) were biased at a particular bit position irrespective of the initial register states. However, the frequency of occurrences of single bit faults varied for different *key-IV* pair initializations of the cipher. The reason for this variation may be due to the data dependent nature of fault sensitivity [23]. We utilized this biased nature of practical faults to develop an attack strategy against the countermeasure proposed in [10].

2) Proposed Combined Attack: We assumed that an adversary can monitor the associated side channel leakage (power in our case) in addition to the ability to inject single bit faults on a Grain stream cipher (Grain v1 or Grain-128) implementation. We get additional favourable cases where the targeted PD values of NLFSR can be uniquely classified by incorporating single bit fault injection capability at $bitNLFSR_{MSB}$ as outlined in Table IV. In the table, we represent the conditions based on the state of the cipher during the first clock cycle of operation. As evident, the conclusions drawn from such favourable cases are equally valid for the remaining key loading phase clock cycles as well.

TABLE IV
ADDITIONAL FAVOURABLE CASES FROM COMBINED SIDE CHANNEL AND FAULT INFORMATION

CASE	*IV Condition*	*Before Fault*	*After Fault*	*Conclusions*
IIIa	$IV(0) \oplus IV(1) = 0$	$PD_{total} = -1$	$PD^*_{total} = -2$	$K(0) \oplus K(1) = 0$
IIIb	$IV(0) \oplus IV(1) = 1$	$PD_{total} = +1$	$PD^*_{total} = +2$	$K(0) \oplus K(1) = 1$
IIIc	$IV(0) \oplus IV(1) = 0$	$PD_{total} = 0$	$PD^*_{total} = +1$	$K(0) \oplus K(1) = 1$
IIId	$IV(0) \oplus IV(1) = 1$	$PD_{total} = 0$	$PD^*_{total} = -1$	$K(0) \oplus K(1) = 0$

Detailed analysis of CASE IIIa :

Let us suppose that for a known IV we have $IV(0) \oplus IV(1) = 0$ and for a *fault-free* operation of the cipher $PD_{total_{t+1}} = -1$. The corresponding PD values for LFSR and NLFSR sections are as follows:

$$\begin{aligned} PD_{LFSR_{t+1}} &= HD_{LFSR_{t+1}} - HD_{LFSR_{t+2}} \\ &= HW(IV(0) \oplus IV(1)) - HW(s_{81} \oplus s_{80}) \end{aligned} \quad (8)$$

$$PD_{NLFSR_{t+1}} = HD_{NLFSR_{t+1}} - HD_{NLFSR_{t+2}} = HW(K(0) \oplus K(1)) - HW(l_{81} \oplus l_{80}) \quad (9)$$

The possible combinations of PD values of the shift registers such that $PD_{total_{t+1}} = -1$ are as follows:
$\{PD_{LFSR_{t+1}}, PD_{NLFSR_{t+1}}\} \in \{\{0, -1\}, \{-1, 0\}\}$
As seen in section IV-A1, a clock glitch in the the cipher implementation will produce a bit flip at $bitNLFSR_{MSB}$. Therefore, we get the following two scenarios:

Scenario 1: Let the PD combination for *fault-free* operation be $\{PD_{LFSR_{t+1}}, PD_{NLFSR_{t+1}}\} = \{0, -1\}$. From equation (9), it can be seen that $K(0) \oplus K(1) = 0$ and $l_{81} \oplus l_{80} = 1$ for such a combination. After fault injection via clock-glitch on the cipher at time $t+2$, the contents of l_{81} gets flipped. Hence, we get $l^*_{81} \oplus l_{80} = 0$, $PD^*_{NLFSR_{t+1}} = 0$ and $PD^*_{total_{t+1}} = 0$. It is to be that such a fault injection does not have any effect on key bits $K(0)$ and $K(1)$; thus, $K(0) \oplus K(1) = 0$ remains unchanged. To summarize, in Scenario 1 the value of $PD_{total_{t+1}}$ changes from -1 to 0 after fault injection.

Scenario 2: Let the PD combination for *fault-free* operation be $\{PD_{LFSR_{t+1}}, PD_{NLFSR_{t+1}}\} = \{-1, 0\}$. From equation (9), it is evident that for $PD_{NLFSR_{t+1}} = 0$ we get $K(0) \oplus K(1) = l_{81} \oplus l_{80}$. Therefore, we get the following two sub-scenarios:

Scenario 2a: Let $K(0) \oplus K(1) = l_{81} \oplus l_{80} = 1$. After inducing clock glitch at time $t + 2$, the contents of l_{81} gets flipped. Hence, we get $l^*_{81} \oplus l_{80} = 0$, $PD^*_{NLFSR_{t+1}} = 1$, $PD^*_{total_{t+1}} = 0$ and $K(0) \oplus K(1) = 1$. To summarize, in Scenario 2a the value of $PD_{total_{t+1}}$ changes from -1 to 0 after fault injection.

Scenario 2b: Let $K(0) \oplus K(1) = l_{81} \oplus l_{80} = 0$. After introduction of clock glitch at time $t+2$ due to flipping of l_{81} we get $l^*_{81} \oplus l_{80} = 1$, $PD^*_{NLFSR_{t+1}} = -1$, $PD^*_{total_{t+1}} = -2$ and $K(0) \oplus K(1) = 0$. To summarize, in Scenario 2a the value of $PD_{total_{t+1}}$ changes from -1 to -2 after fault injection.

Remarks: The transitions of $PD_{total_{t+1}}$ are identical for scenarios 1 and 2a though the values of $K(0) \oplus K(1)$ are complementary. Therefore, in such cases the values of the XORs between adjacent key bits cannot be uniquely determined. On the other hand, when the value of $PD_{total_{t+1}}$ changes from -1 to -2 for $IV(0) \oplus IV(1) = 0$, we get $K(0) \oplus K(1) = 0$. This result corresponds to CASE IIIa in Table IV. Using similar arguments, we can arrive at CASEs IIIb, IIIc and IIId.

The favourable cases due to combined fault and side channel analysis in addition to the favourable cases obtained from side channel leakages only significantly reduces the key search space. The combined fault and side channel attack strategy against the Grain family of stream cipher is outlined in Algorithm 3. The main idea behind our proposed attack is to first exploit the leakages associated with the power side channel during the initialization phase of the cipher implementation and thereafter combine the single bit biased fault injection model along with associated power leakages to retrieve the secret key. Initially each of the adjacent key bits XOR relations (represented by array 'valid_adj_XOR') are unknown, hence all elements of the array are assigned 0. An element will be assigned 1 when the corresponding XOR relation is uniquely determined from the information gained from either the favourable cases of power or combined analysis attack strategy. The remaining undetermined adjacent key XOR relations are exhaustively search to retrieve the secret key. If we assume a fault model that empowers an adversary to run a Grain cipher implementation with different IVs for the same secret key in addition to reinitializing the cipher multiple times with same key-IV pair [9], then the key search space would be reduced to only two guesses using our proposed combined side channel and fault analysis attack strategy. Moreover, our proposed attack technique is during the initialization phase of the cipher and works even if the useful faulty ciphertexts are not available due to incorporation of some randomized countermeasures during the keystream generation rounds.

Algorithm 3: Combined side channel and fault analysis attack algorithm against Grain stream cipher

Input: *CASEs IIa, IIb, IIc : power favourable cases,*
CASEs IIIa, IIIb, IIIc, IIId : combined favourable cases
Output: m-bit *key* /* m=80 for Grain v1, m=128 for Grain-128*/

for $num = 1$ *to* $m - 1$ **do**
  valid_adj_XOR[num] $\leftarrow 0$
end
$NLFSR \leftarrow key$
$LFSR \leftarrow [IV, \{1\}^i]$ // i=16 for Grain v1, i=32 for Grain-128
for x cycles and capture power. // x=81 (Grain v1), x=129 (Grain-128)
for $clock\_cycle_{num} = 1$ *to* $m - 1$ **do**
  if *(*$PD_{total}[clock\_cycle_{num}] \in power\ favourable\ cases$*)* **then**
    valid_adj_XOR[$clock\_cycle_{num}$] $\leftarrow 1$
  end
end
for *each* valid_adj_XOR[num]= 0 **do**
  Clock cipher for $num - 1$ cycles and then inject clock glitch.
  if *(*$PD^*_{total}[clock\_cycle_{num}] \in combined\ favourable\ cases$*)* **then**
    valid_adj_XOR[$clock\_cycle_{num}$] $\leftarrow 1$
  end
end
if *(* $\sum_{i=1}^{m-1}$ valid_adj_XOR[i] = $m - 1$) **then**
  Guess $key(0) = 0$ and get Key guess 1.
  Generate keystream 1.
  Guess $key(0) = 1$ and get Key guess 2.
  Generate keystream 2.
  Match keystreams 1 and 2 with the actual
  keystream to obtain the correct *key* guess
end
else
  Perform exhaustive search for the remaining *key* bits.
end

V. Experimental Results

In this section, we provide detailed descriptions of our experimental strategy, results and their analysis to demonstrate our proposed attack technique against a Grain v1 implementation on Xilinx Virtex-5 (XC5VLX50) FPGA of SASEBO-GII board. The power traces of the cipher circuit were captured using a Tektronix digital oscilloscope DPO 4034B (2.5 GSa/s). We used LS-SVM algorithm to analyze the power traces.

A. LS-SVM

Support Vector Machines (SVMs) are powerful supervised learning models for data analysis and pattern recognition. They are widely used for problems of classification and regression analysis. LS-SVM is a kind of kernel based learning method in which a solution is obtained by solving a set of linear equations instead of convex quadratic programming problem as solved by conventional SVMs [16] .

In LS-SVM we employed Radial Basis Function (RBF) kernel, which involves two hyperparameters - a regularization parameter γ and a parameter σ^2 related to the shape of the decision boundary. The optimization of these parameters is crucial to get high success rates of the classifier. The classification problems in which there are more than two classes can be modeled using multiclass LS-SVM which can be generated by combining binary SVMs.

B. Template Attack using LS-SVM

In this subsection we provide the details of building templates of real power traces collected from hardware implementation of Grain v1 stream cipher on SASEBO G-II board. We implemented the LS-SVM supervised learning classifiers using the LS-SVMlab 1.8 [24].

In a standard template attack (TA) [25] the adversary first constructs multivariate Gaussian templates of noise within the collected power traces for all possible Hamming distance (HD) classes. In the subsequent characterization phase, the attacker classifies a new power trace by calculating multivariate Gaussian probability density functions for all the templates and applying maximum likelihood approach. This technique thus relies on the assumption of a particular noise model to mount a successful attack. To overcome this issue, recent works suggest a noise distribution independent SVM based approach as one of the most promising alternatives [26]–[28].

In order to apply our proposed attack strategy against Grain v1, we must be able to distinguish among its power consumption differences in successive clock cycles (i.e., the PD values). However, even if the HD classes are separated by the same distance, the distances among corresponding mean power trace values usually differ [29]. Therefore, in order to successfully implement our proposed attack technique, one must first construct power trace templates corresponding to all possible HD classes and then classify an unknown power trace with respect to the preconstructed templates.

We present the basis of our attack approach by considering the power dissipations of Grain v1 in two successive clock cycles. Since the internal state of Grain v1 is 160 bits in length, the frequency of HDs around the HD class 80 is much higher than the extremal HD values. Let us consider that its power consumption in the i-th clock cycle corresponds to HD class 85 ($HD(85)$), which belongs to the category of high frequency HDs. According to our analysis in section III-D, the set of possible PD_{total} values for Grain v1 is {-2,-1,0,1,2}. Therefore, the power consumption of the cipher implementation in $i+1$-th clock cycle will correspond to one of the elements of the set $HD_{set} = \{HD(83), HD(84), HD(85), HD(86), HD(87)\}$.

We constructed a data set consisting of 3500 *aligned* power traces (each having 1250 time samples) for different *key*-IV pairs. Out of the collected power traces, 2500 constituted the training set and the remaining 1000 made up the test set. In the training set 500 power traces were collected for each of the possible HD classes belonging to HD_{set}. In the test set we collected 500 power traces for $HD(83)$ and another 500 traces for $HD(85)$ with different *key*-IV pairs of Grain v1.

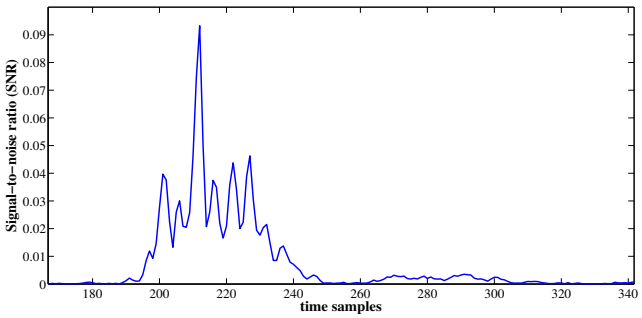

Fig. 4. SNR plot of power trace time sample instants in a clock cycle

1) Feature Selection: The majority of the large number of time samples of a power trace do not contain any relevant information with respect to the targeted register update value and thus represent noise. Therefore, feature selection is an inevitable option to extract the useful time samples having high information content as well as to reduce the computational burden of the classifier. For feature selection we used the Signal-to-noise (SNR) ratio as a metric [29] based on the collected power traces constituting the training set. The plot of SNR corresponding to various time sample components of a power trace belonging to the same clock cycle is shown in Fig.4. The sharp peak around sample instant 212 corresponds to the rising edge of the clock, during which state change of the shift registers of Grain v1 takes place.

2) Results of multiclass classifications: We used the RBF kernel having hyperparameters γ and σ^2 for multiclass classifications. A combination of cross-validation and simplex algorithms was used for tuning the RBF kernel hyperparameters. In order to minimize the effect of noise to a greater extent, we took means for every 10 power traces belonging to the same HD classes to get a representative power trace. Thus, our training and test sets were transformed to 250 and 100 representative power traces respectively. We used LS-SVM algorithm to identify the two HD classes constituting the test set. We report the success rate by the average correct classifications of the two HD classes. The results of these classifications are presented in Table V. The classification of the test set using five largest SNR time sample points as components of LS-SVM model led to a high success rate of 98%. When the number of features were increased to 6 (i.e., the six largest SNR sample instants), the classification outcome showed a lower success rate of 97%. The reason behind this degradation being that the extra feature considered did not impart any added valuable information to the classifier. In fact the added feature corresponds to a noisy sample making the classification task more difficult. Therefore, a template attack using LS-SVM with suitable features will empower an adversary to successfully attack Grain v1 using our proposed attack strategy. However, the success rates of classifications using LS-SVM learning algorithm largely depends on the quality of power traces collected. The lesser the influence of noise elements, the higher is the success rate.

TABLE V
RESULTS OF MULTI-CLASS CLASSIFICATION USING RBF KERNEL

Number of features	Success Rate	γ	σ^2
1	47	6.93	1.61
2	70	8.37	1.13
3	77	4.20	0.93
4	82	5.03	1.09
5	98	6.88	1.27
6	97	8.21	1.37

VI. CONCLUSION

In this paper, we present a new DPA attack against Grain based on the Hamming distance power model. Our proposed attack technique does not require any selection of specific IVs like the DPA attack presented in [6]. Our attack strategy also empowers an adversary to trade-off between the number of resynchronizations of Grain cipher with new distinct IVs and the computation cost for an exhaustive search to retrieve the undetermined *key* bits. We present a novel combined side channel and fault analysis scheme against fault attack countermeasures for Grain stream cipher utilizing the biased nature of clock glitch induced single bit faults occurring in practice. A similar attack can also be mounted on other Fibonacci configured shift registers based stream ciphers like Trivium [30]. To demonstrate our proposed power analysis technique, we report the experimental results of power trace classifications using LS-SVM learning algorithm for a Grain stream cipher implementation on SASEBO G-II board.

REFERENCES

[1] P.Kocher, J.Jaffe, and B.Jun, "Differential Power Analysis," in *CRYPTO*, 1999, pp. 388–397.

[2] P.Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *CRYPTO*, 1996, pp. 104–113.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *CHES 2001*. Springer, 2001, pp. 251–261.

[4] E. Dubrova, "A transformation from the fibonacci to the galois nlfsrs," *Information Theory, IEEE Transactions on*, vol. 55, no. 11, pp. 5263–5271, 2009.

[5] M. Hell, T. Johansson, and W. Meier, "Grain-a stream cipher for constrained environments. ecrypt stream cipher project report, 2005."

[6] W. Fischer, B. M. Gammel, O. Kniffler, and J. Velten, "Differential power analysis of stream ciphers," in *Topics in Cryptology–CT-RSA 2007*. Springer, 2006, pp. 257–270.

[7] B. Gierlichs, L. Batina, C. Clavier, T. Eisenbarth, A. Gouget, H. Handschuh, T. Kasper, K. Lemke-Rust, S. Mangard, A. Moradi *et al.*, "Susceptibility of estream candidates towards side channel analysis," *Proceedings of SASC*, pp. 123–150, 2008.

[8] A. Berzati, C. Canovas, G. Castagnos, B. Debraize, L. Goubin, A. Gouget, P. Paillier, and S. Salgado, "Fault analysis of grain-128," in *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*. IEEE, 2009, pp. 7–14.

[9] S. Karmakar and D. R. Chowdhury, "Fault analysis of grain-128 by targeting nfsr," in *Progress in Cryptology–AFRICACRYPT 2011*. Springer, 2011, pp. 298–315.

[10] S. Banik, S. Maitra, and S. Sarkar, "A differential fault attack on the grain family of stream ciphers," in *Cryptographic Hardware and Embedded Systems–CHES 2012*. Springer, 2012, pp. 122–139.

[11] P. Dey, A. Chakraborty, A. Adhikari, and D. Mukhopadhyay, "Improved practical differential fault analysis of grain-128," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 459–464.

[12] F. Amiel, K. Villegas, B. Feix, and L. Marcel, "Passive and active combined attacks: Combining fault attacks and side channel analysis," in *Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on*. IEEE, 2007, pp. 92–102.

[13] C. Clavier, B. Feix, G. Gagnerot, and M. Roussellet, "Passive and active combined attacks on aes combining fault attacks and side channel analysis," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2010 Workshop on*. IEEE, 2010, pp. 10–19.

[14] A. Dehbaoui, J.-M. Dutertre, B. Robisson, and A. Tria, "Electromagnetic transient faults injection on a hardware and a software implementations of aes," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012 Workshop on*. IEEE, 2012, pp. 7–15.

[15] "Sasebo g-ii , http://www.rcis.aist.go.jp/special/sasebo/index-en.html."

[16] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[17] S. Burman, D. Mukhopadhyay, and K. Veezhinathan, "Lfsr based stream ciphers are vulnerable to power attacks," in *INDOCRYPT 2007*.

[18] A. Chakraborty, B. Mazumdar, and D. Mukhopadhyay, "Fibonacci lfsr vs. galois lfsr: Which is more vulnerable to power attacks?" in *Security, Privacy, and Applied Cryptography Engineering*. Springer, 2014, pp. 14–27.

[19] A. A. Zadeh and H. M. Heys, "Simple power analysis applied to nonlinear feedback shift registers," *Information Security, IET*.

[20] S. Mansouri and E. Dubrova, "An architectural countermeasure against power analysis attacks for fsr-based stream ciphers," in *COSADE 2012*. Springer, 2012, pp. 54–68.

[21] I. Adler and S. M. Ross, "The coupon subset collection problem," *J.Appl.Prob.*, pp. 737–746, 2001.

[22] S. S. Mansouri and E. Dubrova, "An improved hardware implementation of the grain stream cipher," in *Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on*. IEEE, 2010, pp. 433–440.

[23] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, "Fault sensitivity analysis," in *Cryptographic Hardware and Embedded Systems, CHES 2010*. Springer, 2010, pp. 320–334.

[24] K.Brabanter, P.Karsmakers, C. F.Ojeda, J.Brabanter, K.Pelckmans, B.Moor, J.Vandewalle, and J.Suykens, "LS-SVMlab Toolbox User's Guide version 1.8."

[25] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *CHES 2002*. Springer, 2003, pp. 13–28.

[26] G. Hospodar, E. Mulder, B. Gierlichs, I. Verbauwhede, and J. Vandewalle, "Least squares support vector machines for side-channel analysis," *Center for Advanced Security Research Darmstadt*, pp. 99–104, 2011.

[27] L. Lerman, G. Bontempi, and O. Markowitch, "Side channel attack: an approach based on machine learning," *CASED*, pp. 29–41, 2011.

[28] T. Bartkewitz and K. Lemke-Rust, *Efficient template attacks based on probabilistic multi-class support vector machines*. Springer, 2013.

[29] T. P. Stefen Mangard, Elisabeth Oswald, *Power Analysis Attacks revealing the secrets of Smart Cards*. Springer, 2007.

[30] C. De Canniere and B. Preneel, "Trivium," in *New Stream Cipher Designs*. Springer, 2008, pp. 244–266.