A Practical Template Attack on MICKEY-128 2.0 Using PSO Generated IVs and LS-SVM

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

***Abstract*—The reported power analysis attacks on hardware implementations of the MICKEY family of streams ciphers require a large number of power traces. The primary motivation of our work is to break an implementation of the cipher when only a *limited* number of power traces can be acquired by an adversary. In this paper, we propose a novel approach to mount a Template attack (TA) on MICKEY-128 2.0 stream cipher using Particle Swarm Optimization (PSO) generated initialization vectors (IVs). In addition, we report the results of power analysis against a MICKEY-128 2.0 implementation on a SASEBO-GII board to demonstrate our proposed attack strategy. The captured power traces were analyzed using Least Squares Support Vector Machine (LS-SVM) learning algorithm based binary classifiers to segregate the power traces into the respective Hamming distance (HD) classes. The outcomes of the experiments reveal that our proposed power analysis attack strategy requires a much lesser number of IVs compared to a standard Correlation Power Analysis (CPA) attack on MICKEY-128 2.0 during the key loading phase of the cipher.**

***Index Terms*—Template attack, MICKEY-128 2.0 stream cipher, PSO, LS-SVM, FPGA.**

I. Introduction

The security of cryptosystems has traditionally been based on mathematically proven robust algorithms. However the real life implementations of ciphers can be analyzed to launch power analysis attacks [1]. Such attacks can be easily launched against a cipher's hardware implementation using widely available standard instruments. Although an active field of research, there are only a limited number of publications related to power analysis attacks on stream ciphers compared to various block ciphers and public key cryptosystems. In [2], the authors present a Differential Power Analysis (DPA) attack on Grain and Trivium stream ciphers where an adversary can select a large number of initialization vectors (IVs). In [3], the authors demonstrate a DPA attack against Grain stream cipher using machine learning based power trace template classifications. Some power analysis techniques against the MICKEY family of stream ciphers have been proposed in [4], [5].

In this paper, we present a new Template attack (TA) strategy against a hardware implementation of eSTREAM hardware portfolio finalist MICKEY-128 2.0 stream cipher. We used Particle Swarm Optimization (PSO) computational method to choose IVs of the cipher in such a way that it aids in our proposed attack technique to retrieve the secret key. The primary motivation of our proposed power analysis strategy is to reduce the number of *resynchronizations* of MICKEY-128 2.0 with different IVs so that a hardware implementation of the cipher can be broken with a *limited* number of captured power traces. We implemented MICKEY-128 2.0 cipher on Xilinx Virtex-5 FPGA device on SASEBO-GII development board [6]. Our proposed attack methodology is based on the Hamming distance (HD) power model, which is a standard model for CMOS logic based circuits. To validate our proposed power analysis attack strategy, we used Least Squares Support Vector Machine (LS-SVM) learning algorithm [7] as an analyzer of the collected power traces.

The organization of the paper is as follows: In section II, we provide a brief description of the MICKEY-128 2.0 stream cipher along with its hardware implementation. We report the results of Correlation Power Analysis (CPA) attacks on an FPGA implementation of the cipher in section III. Section IV describes our proposed power analysis attack strategy on MICKEY-128 2.0 using PSO generated IVs. In section V, we present the experimental results of our proposed attack on MICKEY-128 2.0 using LS-SVM learning algorithm. The final section concludes the paper.

II. Background

In this section, we first provide a brief description of MICKEY-128 2.0 stream cipher, followed by its implementation details on an FPGA platform. Finally, we briefly mention some reported CPA attacks on the MICKEY family of stream ciphers.

A. MICKEY-128 2.0 stream cipher

The stream cipher MICKEY-128 2.0 was designed by Babbage and Dodd [8]. It was selected as one of the hardware oriented algorithms in the final portfolio of eSTREAM project. MICKEY stands for **M**utual **I**rregular **C**locking **KEY**stream generator and it was designed for resource constrained hardware platforms.

The input parameters of MICKEY-128 2.0 are (a) a 128 bit secret key K $(k_0, ..., k_{127})$ (b) an initialization vector or IV $(iv_0, ..., iv_{IVLENGTH-1})$ anywhere between 0 and 128 bits in length. The cipher is composed of two 160 bit long registers: a Linear Feedback Shift Register (LFSR) termed R and a Nonlinear Feedback Shift Register (NLFSR) termed S. The clocking of each register is dependent on the states of both the registers as detailed in [8].

The structure of MICKEY-128 2.0 is shown in **Fig. 1**. The stages of the registers R and S are denoted by $(r0, r1, ..., r159)$ and $(s0, s1, ..., s159)$ respectively. The variables $Control\_bit\_R$ and $Control\_bit\_S$ are used to regulate the clocking of the registers R and S respectively.

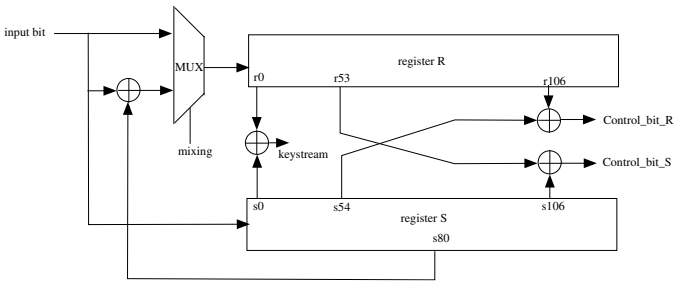

Fig. 1. Hardware realization of MICKEY-128 2.0 stream cipher.

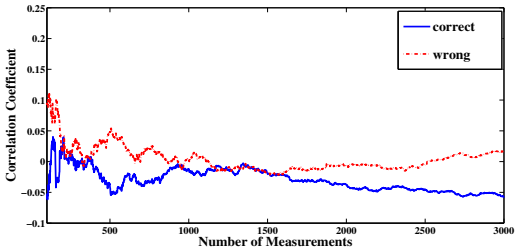

Fig. 2. Number of measurements vs. Correlation Coefficient targeting 64^{th} key bit using **random** IVs.

B. Related Works

In [5], the authors have outlined a Differential Power Analysis (DPA) attack technique against a MICKEY-128 cipher implementation in eSARGOt ASIC. A detailed study of various methodologies based on combinations of different intermediate values, power models and statistical analyses are presented in the work. In another hypothetical CPA attack on MICKEY v2 [4], the authors have considered the basic structure of MICKEY v2 as combinations of two and three input XOR gates and have provided some simulated results. In the next section, we present the results of a CPA attack mounted on an FPGA implementation of MICKEY-128 2.0 based on the HD power model, which is suitable for CMOS logic circuits.

III. CPA on an FPGA design of MICKEY-128 2.0

Unlike block ciphers, power analyses of stream ciphers usually require to capture the leakage associated with several consecutive clock cycles or rounds of operations rather than targeting a particular round. In a standard CPA framework against a stream cipher, an adversary resynchronizes its hardware implementation with several different Initialization Vectors (IVs) for a fixed secret key to obtain a sufficiently large number of power traces.

A. Attack Strategy

To execute a CPA attack, an adversary needs to focus on specific register(s) and find an *interesting* time interval where the register contents vary depending upon secret key bits. We targeted the registers R and S of MICKEY-128 2.0 during the key loading phase as the updates of both these registers are dependent on the input key bits. We used the Hamming distance (HD) power model to simulate the dynamic power consumption of the cipher implementation. At a particular clock cycle, a single key bit was targeted by randomly varying IVs for a fixed secret key as outlined in [5]. Once a key bit was determined from the correlation profile, the subsequent key bit was targeted using the same procedure. We can repeat this attack strategy for different key bit loading rounds to recover entire 128 bits of the secret key.

B. Attack Results

We mounted a CPA attack on a MICKEY-128 2.0 implementation in Virtex-5 FPGA of SASEBO-GII board targeting two randomly selected key loading cycles (1^{st} and 64^{th} key bits). In our experimental setup, a higher power consumption results in dips towards negative direction for the captured power profiles. Therefore, in the correlation plots a more negative profile corresponds to the correct key bit. We noticed that with a few number of power traces, the correlation profiles of the correct and the wrong key bit guesses are hardly distinguishable. But, with an increase in the number of traces we observed a sharp distinction between the two correlation profiles at the *interesting point*. From our experiments, we observed that for the 1^{st} key bit loading round 1000 power traces were sufficient. On the other hand, while targeting the 64^{th} key bit loading, around 3000 power traces were required to clearly distinguish between the correct and wrong key bit guesses. In **Fig. 2**, we have plotted the values of correlation coefficients at the *interesting point* against the number of power traces captured by randomly varying the IVs targeting the 64^{th} key bit. The correlation profiles corresponding to correct and wrong key bit guesses are plotted in *blue* and *red* respectively. In the next section, we propose a new chosen-IV Template attack (TA) strategy which requires a much lesser number of IVs to retrieve the secret key of MICKEY-128 2.0 stream cipher.

IV. Proposed Attack on MICKEY-128 2.0

The primary motivation of our proposed power analysis attack strategy is to retrieve the secret key of MICKEY-128 2.0 with a *limited* number of power traces. As seen from the results of the CPA attack in section III, around 3000 different IVs are required to distinguish between correct and wrong key bit guesses. To reduce the number of such resynchronizations of MICKEY-128 2.0, we propose a Particle Swarm Optimization (PSO) based IV selection procedure. In this section, we first provide a brief overview of the PSO methodology, followed by detailed descriptions of the hypothetical power model considered and our proposed power analysis attack against MICKEY-128 2.0 stream cipher.

A. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based stochastic technique which optimizes a formulation through generations with regard to a given measure of quality [9]. In PSO, a population of particles is initialized with random positions which move around the search-space in quest for optimum by updating themselves through iterations. Each particle's movement is influenced by the position vector of the best solution it has achieved its so far (called *local best*) and the position vector of the best solution achieved by any particle in the swarm so far (called *global best*).

Such movements of several particles across the search-space is expected to attain an optimal solution for the problem being considered. The PSO computational method can be applied on optimization problems that are partially noisy or change over time. A drawback of metaheuristics like PSO is that it does not guarantee an optimal solution is found. However, in context to our proposed attack strategy against MICKEY-128 2.0 using PSO based IV selections, we only require to find solutions above a certain threshold (*predetermined margin* as defined in section IV-B), which may or may not be an optimum value.

B. Hypothetical Power Model

We considered the Hamming distance (HD) power model to estimate the power consumption of a CMOS based implementation of MICKEY-128 2.0 stream cipher. The overall power consumption during the key loading phase of the cipher can be modeled using the following expression:

$$P = P_{R_{comb}} + P_{S_{comb}} + \sum_{i=0}^{159} P_{R_i} + \sum_{i=0}^{159} P_{S_i} + \sigma$$

where, $P_{R_{comb}}$, $P_{S_{comb}}$, P_R, P_S and σ denote the power consumptions of the combinational circuits associated with the register R, combinational circuits associated with the register S, stages of the register R, stages of the register S, and implementation independent noise respectively. In our attack strategy, we considered the power dissipation due to the stages of registers R and S only as they are the dominating power contributing components.

We define *predetermined margin* (PM) as the minimum difference between any two HD classes that can be distinguished using preconstructed power trace templates from a cipher's hardware implementation.

The value of PM can be *set* by an adversary depending upon the noise characteristics of the captured power traces. If the noise levels are high, then it would be difficult to distinguish between two *close* HD values from the power templates. In such a case the adversary must *set* high values of PM to correctly identify the HD classes. On the other hand, if the influence of noise in the collected power traces are low, small values of PM would be sufficient. In our proposed power analysis technique, the parameter PM has a direct correlation with the number of IVs required to mount a Template attack (TA) on MICKEY-128 2.0 stream cipher.

C. Attacking with PSO based IV selections

In this subsection, we provide a detailed description of a chosen-IV power analysis attack strategy on MICKEY-128 2.0 using PSO based computational method.

1) PSO formulation: Each particle consists of an n bit binary array, n being equal to the length of IV. A '1' in the k^{th} bit of the array indicates that the k^{th} IV bit is 1, whereas a '0' in the k^{th} bit signifies that the k^{th} IV bit is 0. In case of MICKEY-128 2.0, since the n bit IV and the 128 bit secret key are loaded bit by bit, the total number of transitions in the registers R and S in the i^{th} round of key loading phase is a function of already loaded n bits of the IV, the preceding $i-1$ retrieved key bits and the unknown i^{th} key bit itself.

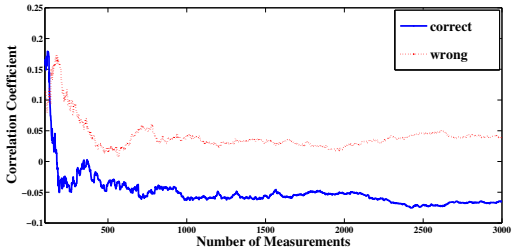

Fig. 3. Number of measurements vs. Correlation Coefficient targeting 64^{th} key bit using **PSO generated** IVs.

In our formulation, the value of *fitness function* $f(p)$ for a particle p in the swarm during the i^{th} round of key loading can be expressed as follows:

$$\boldsymbol{f(p) = |(HD^i_R(p) + HD^i_S(p))_{key\_bit\_guess\_0} - (HD^i_R(p) + HD^i_S(p))_{key\_bit\_guess\_1}|}$$

where, $HD^i_R(p)$ and $HD^i_S(p)$ denote the HDs of registers R and S respectively for the particle p during the i^{th} round of key loading phase. The value of $f(p)$ denotes the absolute difference between the sums of the aforementioned HDs for different guesses of the targeted key bit. For our experimentations, we initially generated 128 bit long random IVs (i.e., $n = 128$) for each particle in the swarm and the corresponding fitness values were evaluated. The *local best* of each particle was initialized with the corresponding IVs from the randomly generated population (typically of size 1000). The *global best* for the first generation was initialized with the particle having the maximum value of *fitness function* in the swarm for the first key bit loading round. The second generation was obtained through applications of sequences of *swap* operations, which determines the velocities with which different particles of the swarm approaches the optimum solution [10]. The *local best* of individual particles and the *global best* were updated if their values in the current generation were higher than their corresponding values in the prior generation. The objective of our formulation was to maximize the value of *fitness function* such that it can be exploited to mount a power analysis attack on MICKEY-128 2.0. The process of updating the *local best* and *global best* solutions was repeated for several generations till a *global best* solution with the *fitness* value above the adversary specified PM was obtained.

2) Application to CPA attack: A CPA attack can be mounted on MICKEY-128 2.0 with a low number of traces if such PSO generated IVs are used as the correlation profiles for the correct and wrong key bit guesses would be significantly different. In **Fig. 3**, we have plotted the correlation coefficient values at the *interesting point* by varying the number of measurements for PSO chosen-IVs with PM set to 30 and targeting the 64^{th} key bit. It was observed that about 500 traces were sufficient to distinguish between the right and the wrong key bit guesses in this case, compared to the requirement of about 3000 traces if the IVs were varied randomly for the *same* secret key. A certain fraction of the PSO generated IVs targeting a particular key bit will produce *fitness* values more than PM for the subsequent key loading rounds as well. However, for mounting a successful CPA attack in those

rounds might require the generation of additional IVs using PSO depending upon the magnitude of PM specified.

Algorithm 1: Attack algorithm on MICKEY-128 2.0 with PSO chosen-IVs

Input: *predetermined margin*: PM, Power trace templates: $POW_{templates}$
Output: 128 bit secret key

```
IVpool = φ ; /* Contains a pool of PSO generated IVs */
for i = 1 to 128 do
    flag = 0 ;
    if (i > 1) then
        for IV ∈ IVpool do
            Clock the cipher with IV and the preceding i − 1 key bits ;
            if (fitness function > PM) then
                Determine the i^th key bit using POW_templates ;
                flag = 1 ; /* No new IV generation required */
                break ;
            end
        end
    end
    if (flag = 0) then
        Generate a new IV using PSO targeting the i^th key bit loading round ;
        Determine the i^th key bit using POW_templates;
        Add IV to IVpool;
    end
end
```

3) Application to Template attack: An adversary can successfully deduce the secret key with a *limited* number of power traces using PSO generated IVs and power trace templates as outlined in Algorithm 1. We constructed power trace templates for different HD values of a MICKEY-128 2.0 implementation (under the hypothetical power model considered) as outlined in Algorithm 2. Subsequently, we used LS-SVM learning algorithm to classify the power traces captured during attack phase for determining the secret key bits.

At first, an IV was generated using the aforementioned PSO formulation with a value of *fitness function* above PM for the 1^{st} key bit loading round. The cipher was then clocked with the PSO generated chosen-IV and the secret key loaded bit by bit. The power trace during the entire key loading phase of the implementation was captured. Subsequently, the 1^{st} bit of the secret key was determined by matching power trace of the corresponding round with the set of preconstructed power trace templates. It is to be noted that for the chosen-IV, a value of *fitness function* above PM for the 1^{st} key bit loading round empowers an adversary to easily distinguish between the HD classes due to the key bit guesses 0 and 1 using the power templates. Once the 1^{st} key bit was successfully determined, the 2^{nd} key bit loading round was targeted. In this phase, at first it was checked whether the prior PSO generated IV produced a *fitness* value above PM for the current targeted round also. If the outcome was positive, then no new IV generation was further required for the 2^{nd} key bit loading round, else a new IV was generated using the PSO formulation described earlier. This process was continued until all the 128 key bits were successfully recovered or a sufficiently large number of key bits were determined such that the remaining key bits can be retrieved by a brute force search. To devise our proposed attack strategy, we exploited the fact that the HDs between consecutive states of MICKEY-128 2.0 vary as a function of IV and the already loaded key bits during the key loading phase of the cipher.

TABLE I
NUMBER OF DIFFERENT PSO GENERATED IVS VS. *predetermined margin* (PM) TO RECOVER THE ENTIRE 128-BIT KEY

predetermined margin	Number of different IVs				
	key1	key2	key3	key4	key5
5	4	5	4	5	4
10	7	8	9	9	9
15	12	14	13	13	15
20	27	24	25	26	24
25	37	38	41	43	42
30	71	69	70	70	72

A PSO generated IV might produce values of *fitness function* above a specified PM for several key bit loading rounds of MICKEY-128 2.0 cipher but usually not for all the targeted rounds. In Table I, we summarize the simulation based results to estimate the number of PSO generated IVs required to recover the entire 128 bit key against various PM values for 5 different, randomly selected keys. It is quite evident that with an increase in PM, more number of PSO generated IVs are required to ensure that for every key bit loading clock cycles there is at least one IV which produces a value of *fitness function* above PM.

In actual power traces there are significant influences of various noise elements. This may lead to a wrong key recovery due to errors in power template matching. To enhance the confidence of a correct key recovery using our proposed attack strategy, we have considered a *majority voting* scheme in subsection V-A such that every key bit is determined by multiple PSO generated IVs.

V. EXPERIMENTAL RESULTS

In this section, we report the results of experimentations to demonstrate our proposed attack strategy against MICKEY-128 2.0 stream cipher. We first provide a detailed description of the techniques adapted to mount our proposed power analysis attack and then, we report the results based on power traces collected from an actual chip implementation.

A. Power Analysis of MICKEY-128 2.0 using 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. The Least Squares Support Vector Machine (LS-SVM) was originally proposed by Suykens and Vandewalle in [7]. 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 problems as solved by conventional SVMs.

In order to demonstrate our proposed power analysis strategy against MICKEY-128 2.0, we used LS-SVM based classifiers to classify the actual power traces corresponding to different HD classes. We implemented LS-SVM supervised learning classifiers using LS-SVMlab 1.8 [11].

In a standard template attack (TA) [12], an adversary first constructs multivariate Gaussian templates of noise within the collected power traces for all possible HD classes. This preliminary step is also termed as the *profiling phase*. 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

TABLE II

NUMBER OF DIFFERENT PSO GENERATED IVS VS. *predetermined margin* (PM) TO RECOVER THE ENTIRE 128-BIT KEY WITH *majority voting* SCHEME.

PM	Number of different IVs														
	key 1			key 2			key 3			key 4			key 5		
	VS 3	VS 5	VS 10	VS 3	VS 5	VS 10	VS 3	VS 5	VS 10	VS 3	VS 5	VS 10	VS 3	VS 5	VS 10
5	10	11	22	9	14	22	8	13	23	9	12	22	8	12	21
10	17	28	41	18	25	43	18	23	42	19	23	45	17	22	42
15	28	37	65	26	36	65	23	42	71	26	42	62	26	41	69
20	64	87	146	59	75	159	59	83	151	55	80	145	57	86	149
25	94	150	254	102	128	265	93	159	255	98	150	267	105	151	254
30	206	328	634	203	315	628	201	332	630	199	312	641	204	320	635

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 [13], [14], [3]. In [14], the authors introduce probabilistic multi-class SVMs and also show that SVM based template attacks outperform conventional TAs when the noise levels are significantly high in the collected power traces. In [3], the authors present a DPA on Grain v1 using the LS-SVM learning algorithm.

To mount our proposed power attack on MICKEY-128 2.0, an adversary needs to identify the HD classes corresponding to captured power traces using preconstructed power trace templates. We employed Algorithm 2 to determine the HD classes for which power traces should be collected in the *profiling phase*. The captured power traces would then be utilized to train the LS-SVM classifier.

Algorithm 2: Determining HD classes required to train LS-SVM classifier

Input: *predetermined margin*: PM, number of simulations: N, targeted key loading round: K

Output: HD classes to construct power trace templates: $HD_{template\_classes}$

```
for i = 1 to N do
    Clock MICKEY-128 2.0 with a 128 bit random IV ;
    Clock the cipher for K − 1 more rounds for a randomly generated key ;
    Compute (HD_R^K + HD_S^K)_key_bit_guess_0 and
    (HD_R^K + HD_S^K)_key_bit_guess_1 ;
    if (|(HD_R^K + HD_S^K)_key_bit_guess_0
        −(HD_R^K + HD_S^K)_key_bit_guess_1| ⩽ PM) then
        i = i − 1 ; /* Discard the randomly generated key-IV pair */
    end
    else
        Add both (HD_R^K + HD_S^K)_key_bit_guess_0 and
        (HD_R^K + HD_S^K)_key_bit_guess_1 to HD_template_classes;
    end
end
```

We generated 10000 HD class pairs and stored them in $HD_{template\_classes}$; for each pair, one HD class corresponded to a key bit guess 0 and the other corresponded to a key bit guess 1 for the K^{th} round of key loading. The HD classes belonging to a pair were separated by a magnitude greater than *predetermined margin* (PM) to ensure that they can be segregated from actual power traces. For our experiments, we set the values of PM and K to **30** and **1** respectively. We obtained the plot as shown in **Fig. 4**, where, *higher HD* and *lower HD* are defined as follows:

$$\begin{aligned} A &= (HD_R^K + HD_S^K)_{key\_bit\_guess\_0} \\ B &= (HD_R^K + HD_S^K)_{key\_bit\_guess\_1} \\ higher\ HD &= max(A, B) \\ lower\ HD &= min(A, B) \end{aligned}$$

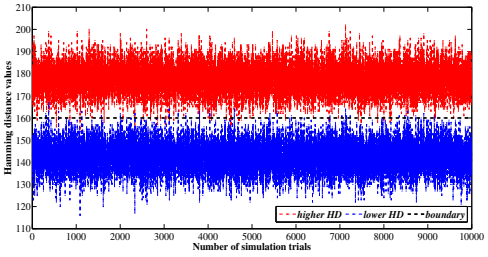

Fig. 4. Selection of *boundary* to partition HD classes for a PM=30.

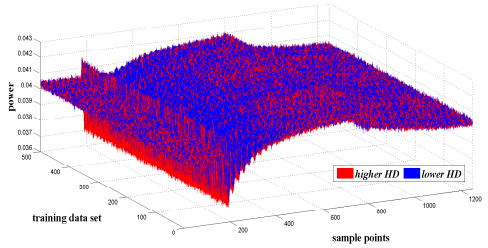

Fig. 5. Training data set for LS-SVM based binary classifier

From the nature of the plot, we selected HD value 160 as the *boundary* value to partition the training HD classes into two distinct sets: (i) *higher HD* class and (ii) *lower HD* class. It is to be noted that since the total length of the register stages in MICKEY-128 2.0 is 320, the frequencies of occurrences of HD classes around mid-value of 160 is expected to be much higher than the extremal values. Therefore, with PM set to 30 we devised a LS-SVM based binary classifier to classify a power trace belonging to either a *higher HD* class or a *lower HD* class. Since the values of the HD classes corresponding to the key bit guesses 0 and 1 are calculated beforehand, the classification outcome of an unknown power trace determined the value of the key bit in the targeted round. However, with lower PM values an adversary needs to employ LS-SVM based multiclass classifiers [3] depending upon the frequencies of HD classes occurring.

We constructed a training data set consisting of 10000 *aligned* power traces corresponding to *higher HD* and *lower HD* classes with *known* key-IV pairs (using Algorithm 2). Each of the training HD classes consisted of 5000 power traces as shown in **Fig. 5**. After forming the training data set, an adversary can deduce the secret key using PSO generated IVs as outlined in Algorithm 1.

However, a power trace collected from actual experimental setup is disturbed by various noise elements and this may lead to faulty classifications. Therefore, to increase our confidence of a successful key bit recovery, we employed a *majority voting* scheme where multiple number of PSO generated IVs

TABLE III
RESULTS OF RBF KERNEL BASED CLASSIFICATIONS FOR ROUND 1.

No. of features	SR for different parameter combinations			
	$\gamma = 1, \sigma^2 = 0.1$	$\gamma = 1, \sigma^2 = 1$	$\gamma = 10, \sigma^2 = 0.1$	$\gamma = 10, \sigma^2 = 1$
1	90	90	90	90
2	90	90	80	90
3	80	90	70	90
4	80	90	80	90
5	70	90	70	80

TABLE IV
RESULTS OF RBF KERNEL BASED CLASSIFICATIONS FOR ROUND 64.

No. of features	SR for different parameter combinations			
	$\gamma = 1, \sigma^2 = 0.1$	$\gamma = 1, \sigma^2 = 1$	$\gamma = 10, \sigma^2 = 0.1$	$\gamma = 10, \sigma^2 = 1$
1	20	10	20	20
2	20	20	20	20
3	90	30	90	40
4	100	90	100	90
5	100	90	100	90

decide the value of each key bit. As evident, with inclusion of *majority voting* scheme, the number of different IVs required to resynchronize the cipher operation increases. We denote the minimum number of PSO generated IVs per key loading round considered for *majority voting* by a parameter *vector size* (VS). In Table II, we report the simulation based results for the total number of PSO generated IVs required for different PM and VS values to entirely recover the 5 randomly selected keys (as used in Table I) after incorporating the *majority voting* scheme. Also for our experimentations, we considered the *mean value* of every 10 power traces for a particular chosen IV to further reduce the effect of noise. Similarly, we took means of every 10 power traces belonging to the same training HD class and hence, the size of the training set was transformed to 1000 *mean* traces.

B. Attack Results on SASEBO-GII

Our experimental setup consisted of a SASEBO-GII evaluation board [6], a Tektronix digital oscilloscope DPO 4034B and a PC. We implemented the cipher and control modules on Xilinx Virtex-5 (XC5VLX50) and Xilinx Spartan-3A (XC3S400A) FPGAs of SASEBO-GII respectively. The Virtex-5 FPGA was operated at 2 MHz clock frequency and the power traces were captured at a sampling rate of 2.5 GSa/s. We used LS-SVM classifier to analyze the power traces.

1) Feature Selection: In a captured power trace, the majority of sample points do not contain any valuable information regarding the targeted register updates of MICKEY-128 2.0 and hence represent noise. We used Pearson's correlation coefficient metric to select the most relevant components of the power traces. Such a feature selection technique helps to reduce the computational burden of a classifier as well as it avoids training the classifier with wrong data set features.

2) Results of classifications: In Tables III and IV, we report the results of LS-SVM based binary classifications targeting two randomly selected key loading clock cycles (1^{st} and 64^{th} round) with PM set to 30. We used RBF kernel-based LS-SVM classifiers with hyperparameter combinations of $\gamma \in \{1, 10\}$ and $\sigma^2 \in \{0.1, 1\}$. In addition, we employed the *majority voting* scheme so that the value of each targeted key bit is determined by the majority outcome of 10 such PSO chosen-IVs. By *success rate* (SR) we refer to the percentage of IVs out of 10 that produce the correct key bit prediction using the classifier. From the results of the experiments based on various combinations of the RBF kernel hyperparameters, it can be stated that by selecting a suitable number of features an adversary can successfully attack a MICKEY-128 2.0 implementation using our proposed method. However, the success rate of classifications using LS-SVM learning algorithm depends on the quality of power traces collected. If high noise levels are associated with the collected power traces, then the aforementioned *majority voting* scheme should be incorporated.

VI. CONCLUSIONS

In this paper, we presented a new Template attack strategy against MICKEY-128 2.0 stream cipher. We used PSO computational method to generate IVs for power analysis of the cipher. The primary advantage of our proposed technique is that it empowers an adversary to attack a hardware implementation of MICKEY-128 2.0 using a *limited* number of resynchronizations of the cipher with different IVs. At first, we performed a standard CPA attack targeting an arbitrarily chosen key loading round and observed that around 3000 random IVs were required, whereas using PSO generated IVs around 500 power traces were sufficient. Finally, we proposed a Template attack on MICKEY-128 2.0 using PSO selected IVs and preconstructed power trace templates using a *limited* number of power traces. To demonstrate our proposed method, we used LS-SVM based binary classifiers as an analyzer of the captured power traces from an FPGA implementation of the cipher.

REFERENCES

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

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

[3] A. Chakraborty, B. Mazumdar, and D. Mukhopadhyay, "A practical dpa on grain v1 using ls-svm," in *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*, pp. 44–47.

[4] J. Liu, D. Gu, and Z. Guo, "Correlation power analysis against stream cipher mickey v2," in *Computational Intelligence and Security (CIS), 2010 International Conference on*, IEEE, 2010.

[5] H. Zhao, *Power analysis attacks on a hardware implementation of the stream cipher MICKEY*. PhD thesis, K. U. LEUVEN, 2009.

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

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

[8] S. Babbage and M. Dodd, "The stream cipher mickey-128 2.0. estream, ecrypt stream cipher project, 2006," *Avaiable at: http://www. ecrypt. eu. org/stream/p2ciphers/mickey128/mickey128_p2. pdf*, 2013.

[9] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5, pp. 4104–4108, IEEE, 1997.

[10] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," in *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 3, IEEE, 2003.

[11] K.Brabanter, P.Karsmakers, C. F.Ojeda, J.Brabanter, K.Pelckmans, B.Moor, J.Vandewalle, and J.Suykens, "LS-SVMlab Toolbox."

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

[13] 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.

[14] A. Heuser and M. Zohner, "Intelligent machine homicide," in *Constructive Side-Channel Analysis and Secure Design*, Springer, 2012.