2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

A State Encoding Methodology for Side-Channel Security vs. Power Trade-off Exploration

Richa Agrawal[1], Mike Borowczak[2] and Ranga Vemuri[3]
[1,3] Digital Design Environments Laboratory, School of Electronics and Computing Systems
University of Cincinnati, Cincinnati, Ohio USA
Email: agrawara@mail.uc.edu, vemurir@ucmail.uc.edu
[2] College of Engineering and Applied Science, Department of Computer Science
University of Wyoming, Wyoming, USA
Email: mborowcz@uwyo.edu

***Abstract*—Power side-channel attacks have been shown to be effective against recovering protected information from integrated circuits. Existing defense methods are expensive in area, power or both. Small-scale ICs used in embedded systems and IoT devices are expected to be safe and secure, and yet cannot afford the area and power overheads of the sophisticated defense methods. This paper presents a design methodology for finite state controllers (FSMs) to defend against power analysis attacks while ensuring low power overhead. Further, a desired level of security can be achieved while minimizing power consumption. We formulate a set of constraints on state encoding based on security and power metrics. We express these constraints as a Boolean satisfiability (SAT) problem and use a SAT solver to generate constraint satisfying encodings. Using over 100 FSMs from BenGen and MCNC benchmark suites, experimental results show an average power reduction of up to 40% with respect to secure-only FSMs and 4-20% reduction with respect to minimal encoding strategy. Trade-off between security and power is demonstrated as mutual information between power side-channel and Hamming attacks models varies between 0 and 2.**

***Keywords*—Low Power, Finite State Controllers, Power Analysis, Boolean Satisfiability**

I. Introduction

Side-channel attacks are non-invasive hardware-based attacks that exploit the relationship between the operations of target device and measurable physical variables. Power analysis attacks use only power consumption information which makes them cost-efficient, easy and powerful against embedded devices [1], [2]. For small-scale finite state machines (FSM), reverse-engineering attacks are also known to be quite effective [3]. These attacks can be improved by combining side-channel information along with its functional output values to engineer the attack. Fault-detection attacks [4] already employ power and EM side-channel leakage to improve the attack efficiency, which makes a similar power analysis attack to reverse engineer an FSM quite plausible.

Current defense methods against power analysis attacks include using cryptographic subsystems [5] to encrypt the data flow to/from the device. But for small-scale devices, their large area and power overheads can render them unfeasible for practical use. Hardware protection schemes such as WDDL [6] which work on a low-level implementation also have high implementation and power costs.

High-level protection schemes hide or mask critical information to make side-channel measurements independent of input data and the device's computational trajectory. In finite state machines, the state registers hide information within their encodings [7]. This information can be subject to power analysis attacks since power consumption profiles can be correlated to data changes in registers and be used to reverse engineer the state machine. Secure design methodology using constrained state assignment has been proposed to thwart such reverse engineering attacks [8]. By appropriate state encoding, the method removes or reduces the correlation between the generated power footprint and the state sequence.

In this paper, we propose an improved power-efficient, low-cost secure design methodology for small-scale finite state controllers against power analysis attacks. Ability to reduce power with or without a trade-off against security is an essential tool for the designers. In Section II, we discuss the attack model and security constraints on state encodings to mitigate information leakage through power side-channel. In Section III, we introduce power reduction techniques with both compromised and uncompromised security levels. Section IV formulates the problem as a Boolean satisfiability (SAT) problem and in Section V we describe how to use a satisfiability solver to generate encodings. In Section VI we present experimental results using over 100 benchmarks to demonstrate the effectiveness of the proposed methods. We offer concluding remarks in Section VII.

II. Secure design for Finite State Machines

A. FSM State Encoding

Let $\mathcal{M} = (S, I, T, s_o)$ be a finite state machine where S is finite set of states, I is finite set of input symbols, $T : SXI \rightarrow S$ is a state transition function and $s_o \in S$ is the initial state. Let $s_1, s_2 \ldots s_M$, where $M = |S|$, denote the states.

States of an FSM are encoded as a set of Boolean state variables stored in flip-flops. Let $Q =< q_o, q_1, \ldots q_R >$ denote the Boolean valued vector stored in R flip-flops. Let $E : S \rightarrow (b_1, b_2, \ldots b_R)$, where $b_i \in \{0, 1\}$, be a state encoding function which maps each state to a Boolean vector of size R. The mapping should assign a unique vector to each state, that is, $\forall_{s_i, s_j \in S}, E(s_i) = E(s_j) \implies s_i = s_j$.

Let the Hamming weight of a Boolean vector B be denoted by HW(B) which equals the number of ones in B. Let the Hamming distance between two Boolean vectors B_1 and B_2 of the same size be denoted by HD(B_1, B_2) which is the number of positions in which the two vectors differ.

Minimal length encodings or simply minimal encodings have encoding length $R_{min} = \lceil log_2(M) \rceil$. Encodings with $R = M$ and HW($E(s)$) = 1, for all states s, are known as one-hot encodings. We will refer to this value of R as R_{max} for that FSM. Given a set of security and power constraints on the state encoding function, we are interested in finding encodings with minimum length of $R \in [R_{min}, R_{max}]$.

B. Hamming-model based Attacks

Power analysis attacks exploit the information leaked through power-side channel to reveal the data stored in internal registers. Attacker's intent is to reverse engineer the finite state controller by attacking the internal state registers. Strong correlation between the power profiles and data stored within the registers shows that variations in power consumption can be used to predict the changes in the state variables which can be used to construct the FSM.

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00031

IEEE computer society

A design implemented using CMOS technology is susceptible to Hamming model based attacks. The Hamming Weight model assumes that the power draw in a CMOS circuit is dependant on the status of each state bit which in turn depends on E(s) for each state s. Hence the power consumed when a circuit is in state s is correlated to the Hamming weight of the state under encoding E, or $HW(E(s))$.

The Hamming Distance model assumes that the dynamic power consumption in a CMOS circuit depends on the state transitions which are characterized by the switching activity in the state registers. Hence, the power consumed when a circuit transitions from state s_1 to state s_2 is correlated to the Hamming distance of the transition from s_1 to s_2 under encoding E, or $HD(E(s_1), E(s_2))$.

Transition	SD
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	1-δ
$1 \rightarrow 1$	0

TABLE I: Switching Distance Model

The Switching Distance model (or modified HD model) assumes CMOS gate consumes slightly more power during rise than during fall. The attack model can take this difference into account. For example, [9] introduced a parameter δ to capture the difference. Nominally, $\delta = 0.17$. In this model, power consumed when a circuit transitions from state s_1 to state s_2 is correlated to $SD(E(s_1), E(s_2))$, where the SD values shown in Table I for one bit are summed up across all the bits in the encoding. This is called the Switching distance of the transition from s_1 to s_2 under encoding E.

C. Secure State Assignment

One-hot encodings are an example of secure state assignment since all the states have the same HW (equal to 1) and all the transitions have the same HD (equal to 2) and same SD (equal to 2-δ). The FSM implementation under such an encoding, is unlikely to leak any information via power side-channel under the Hamming weight attack, the Hamming distance attack or Switching distance attack. While this is a possible solution, it has a large encoding size requirement.

Another way of securing the FSM against power attacks is by masking the information leakage with respect to different input patterns. A state s is said to be L-Reachable, if the machine can traverse from the start state s_o to state s in L clock cycles, or in other words with an input sequence of length L. By dividing all the states into different sets based on their clock-cycle reachability and making them indistinguishable by their power footprint, information about different paths invoked by different input patterns can be masked.

Since it is possible to reach a state through multiple paths, a state could be L-reachable for multiple values of L. In that case, state transition probabilities can be used to determine the most likely L value for which a state (transition) is reachable. Agrawal et al. [8] proposed a heuristic algorithm which finds the most probable states to reach after L cycles and assign state encodings with the same Hamming weight to all such states and ensure same Hamming distance to all of their outgoing transitions. This hides the state and transition information from being leaked through power and HW and HD power models. However, this algorithm requires restructuring of FSMs by replacing states with self-loops by a pair of states while maintaining functional equivalence. This leads to an increase in the area and power requirements due to the increase in the number of states and transitions.

In contrast, in this paper, we propose methods to achieve low power consumption while maintaining security and to trade power against security. Our encodings provide security against power attacks based on HD, HW as well as SD models which are widely used in power side-channel attacks.

III. Low Power Design for Secure FSMs

We can denote the states and transitions of an FSM as unique nodes and edges of a State Transition Graph (STG). For convenience, we refer to the STG as (S, T, s_o), where S denotes the set of state vertices, $T \subseteq (SXS)$ is the set of directed edges denoting the state transitions and $s_o \in S$ is the vertex representing the start state. Let L be a non-negative integer. We use the predicate $R_L(s)$ to denote the L-Reachability of state s.

Let $S_L \subseteq S$ be a subset of states which are L-reachable. Formally, $S_L = \{s \in S | R_L(s)\}$. Let T_L be the set of all transitions in the STG originating from the states in S_L. Formally, $T_L = \{(s_1, s_2) \in T | s_1 \in S_L\}$. In order to thwart information leakage through power side channel, all states and transitions occurring in the same clock-cycle should have the same power consumption to prevent the attacker from distinguishing them by their power footprint. Hence all states and transitions of an FSM are divided into S_L and T_L sets respectively, where, for different values of L, all the members of these sets are L-reachable. Multiple L-reachability of states are resolved using state transition probabilities, where the most probable path to the reach the state determines its S_L set.

Every FSM, based on its structure, has a maximum number of sets (L_{max}) that its states and transitions can be divided into. The designer can choose number of set divisions ($1 \leq l \leq L_{max}$) based on a desired level of security at the cost of power and area. $l = 1$ corresponds to maximal length encoding (such as one-hot). $l = L_{max}$ corresponds to minimum security as discussed before and usually, but not necessarily, yields minimal encoding length. Choice of l between these two extremes can be used to partition the states (transitions) into l groups so as to drive constrained encodings within each group such that the states (transitions) cannot be distinguished from each other and provide a trade-off point for security level vs. encoding length.

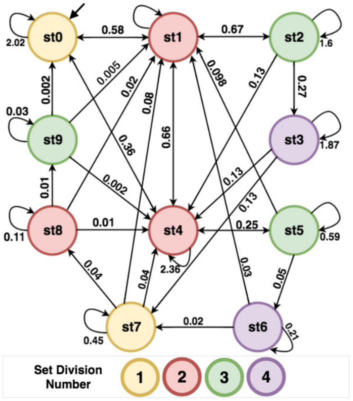

Fig. 1: MCNC Benchmark BBARA

As an example, consider *BBARA* MCNC benchmark FSM (consisting of 10 states and 37 transitions, with start state $st0$) shown in Figure 1 with weighted state transition probabilities (STPs) shown on their edges. Since self-loop transitions need to be restructured to determine L-reachability of a state, let's ignore their presence for the sake of explanation and assume that the FSM is to be divided into 4-l sets ($L_{max} = 7$). Starting with start state $st0$ (assigned with set division number 'l' = 1) and moving through the path of maximum STPs, the transition with maximum STP is used to assign the 'l' value to next-state of the transition, which is one greater than the 'l' value of the current-state. To restrict the division to 4-l sets, modulo operation is performed while assigning next-state 'l' values. The final division of sets in the *BBARA* FSM is shown in color-coded format in the figure.

Encodings generated fulfilling these conditions will be fully or partially secure (depending on l chosen) against HW, HD and SD attack models. Security against HW and HD attack models is quite intuitive. To understand security against SD attacks, consider two separate state transitions of an FSM, $\{St1 \to St2\}$ and $\{St3 \to St4\}$, where $St1$ and $St3$ belong to same L-reachable set S_{L1}, and $St2$ and $St4$ belong to set S_{L2}. Since states belonging to same S_L set must have equal HW and transitions originating from same set of states must have equal HD, let them be encoded such that:

$$HW(E(St1)) = HW(E(St3)) = W1$$
$$HW(E(St2)) = HW(E(St4)) = W2$$
$$HD(\{St1 \to St2\}) = HD(\{St3 \to St4\}) = D1$$

Let no. of (0 → 1) changes in transition $\{St1 \to St2\}$ be denoted by P and no. of (1 → 0) changes be denoted by Q:

$$P + Q = D1 \tag{1}$$

By simple math, it can be calculated that after transition $\{St1 \to St2\}$, number of 1's in state $St2$ is equal to the sum of number of 1's in state $St1$ and number of (0 → 1) changes subtracted by number of (1 → 0) changes. Therefore,

$$W2 = W1 + (P - Q) \tag{2}$$

which implies: $2P = D1 + (W2 - W1)$
$2Q = D1 - (W2 - W1)$

Similar results can be obtained for transition $\{St3 \to St4\}$ since values of P and Q depend on $D1$, $W1$ and $W2$. Hence transitions in the same T_L set have equal switching distance. Therefore the proposed design methodology provides security against SD attacks too.

The states should be encoded in such a way that for any L, all states in S_L should have the same HW and all transitions in T_L should have the same HD and SD values. Since power has a strong correlation to HW, HD and SD, constrained encodings so produced ensure that the states (transitions) in the same S_L (T_L) cannot be distinguished from another. This is the minimum level of security an encoding should ensure. At the other extreme, no two states (transitions) would be distinguishable if all states (transitions) have the same HW (HD) values. One-hot encoding is an example of such an encoding.

A. Security vs Power Trade-off

Self-loop transitions in FSMs lead to zero Hamming distance value which reduces the power consumption. However, near-zero power consumption indicates to the attacker that the machine remained in the current state. To fully hide the transition information, secure FSM design methods restructure FSMs to eliminate self-loops [8], [10] at significant expense of power/area. In this paper, to save power/area we ignore self-loops and divide the states and transitions into desired number of S_L and T_L sets as discussed before.

Considering BBARA benchmark FSM with 4-l sets, as shown in Figure 1, the self-loop transitions are ignored while dividing the states and transitions into sets. A satisfying encoding strategy for this division is shown in Table II under Secure Encoding. Different HW, HD and SD values generated in the state register of the FSM implementation are shown in Table III. This FSM implementation gives the illusion of a generic 4-state FSM shown in Figure 2, which generates Hamming patterns similar to the benchmark FSM.

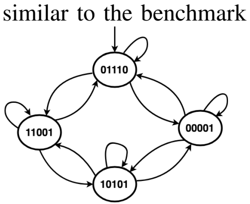

Fig. 2: Generic 4-state FSM

Proposed secure designs with intact self-loop transitions do no lead to any information leakage against HW power attacks. Against HD and SD power attacks, the proposed compromise might reveal some reachability information in the form of presence of a self-loop transition. However, it does not compromise on information regarding other transitions or reachability of rest of the states in any manner. The critical information is secure and as the results show the consequent impact on security due to proposed design trade-off is minimal.

B. Low Power Encodings for a Given Security Level

We now show a method to reduce power while maintaining the level of security afforded by the choice of l.

1) State Transition Probabilities: State Transition Probabilities (STPs) are used to resolve conflicts in reachability of states and also play an important role in average power consumption of a finite state machine. STP values can be obtained both theoretically and experimentally. We apply a theoretical method previously proposed by [11] wherein at any given clock cycle, the FSM must be in exactly one of its states. Probability that the FSM is in state s_n can be determined by the following system of equations:

$$P(s_n) = \sum_{m=1}^{M} P(s_m) * P(I_{mn}), \quad n \in \{1, M\} \tag{3}$$

where $P(I_{mn})$ is the probability of the input 'I_{mn}' to cause the state transition (s_m, s_n). The above equation calculates the sum of probabilities of all the transitions reaching state s_n from any state in the FSM.

Since the machine has to exist in at least one of the given 'M' states,

$$\sum_{m=1}^{M} P(s_m) = 1 \tag{4}$$

After solving for probabilities $P(s_n)$, STP for any transition (s_m, s_n) can then be calculated using,

$$P((s_m, s_n)) = P(s_m) * P(I_{mn}) \tag{5}$$

2) Switching Activity: Dynamic power consumption of a CMOS circuit is proportional to its average switching activity and switching activity of an FSM state register can be estimated using its STPs.

Let us assume that a state encoding E assigns bit-vectors of length R to every state. Let $b_n = E(s_n)$ be encoding of state s_n, where b_n^r is the value of r^{th} bit, $r \in [1, R]$. The switching activity (SA_r) of each flip-flop can be calculated using the STP and the toggle density of the flip-flops:

$$SA_r = \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{M} P((s_m, s_n)) * (b_m^r \bigoplus b_n^r) \tag{6}$$

The Switching Activity (SA) of the state register can be estimated as:

$$SA = \sum_{r=1}^{R} SA_r = \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{M} (w_{mn}) * HD(E(s_m), E(s_n)) \tag{7}$$

where, w_{mn} = P((s_m, s_n)) + P((s_n, s_m))

In order to obtain a secure and low power design, additional constraints are placed on state encodings to minimize average switching activity of the state register. This is achieved by assigning lower HD values to transitions with higher STPs.

Let the FSM transitions be divided into J T_L sets, $T_1, T_2, \ldots T_J$. Since all transitions in every T_L set must have the same HD value, SA estimation can be written as:

$$SA = \sum_{j=1}^{J} (HD_j * \sum_{t \in T_j} P(t)) \tag{8}$$

where, HD_j is the HD value associated by E with T_j and P(t) are the STPs of the transitions in set T_j.

The bounded range [SA_{min}, SA_{max}] can be calculated as follows. The lower bound of SA equates to minimum possible value of HD, that is '0' for a self-loop transition:

$$SA_{min} = \sum_{j=1}^{J} (0) * \sum_{t \in T_j} P(t) = 0 \tag{9}$$

SA_{max}, the upper bound of SA, corresponds to every flip-flop switching in every clock cycle or the HD of every valid transition is equal to the number of encoding bits R:

$$SA_{max} = \sum_{j=1}^{J} R * \sum_{t \in T_j} P(t) = R * \sum_{m=1}^{M} \sum_{n=1}^{M} P((s_m, s_n)) = R \tag{10}$$

These parameters will be used to generate encodings as described in the next section.

3) State Encodings Constraints: The problem of generating the desired state encodings can be stated in terms of set of constraints. For the given level of security, all the states (transitions) of the FSM can be divided into appropriate number of S_L (T_L) sets. The following set of constraints are applied to find a valid set of state encodings within the range $[R_{min}, R_{max}]$:

1. S_L Set Constraints: States with the same S_L set number must have the same Hamming weight.

2. T_L Set Constraints: Transitions with the same T_L set number must have the same Hamming distance.

Additional constraint to minimize switching activity is applied to reduce average power consumption:

3. Switching Activity Constraint: Switching Activity (SA) of the state register in the FSM should be as low as possible.

In benchmark FSM *BBARA* shown in Figure 1, we can observe an increase in encoding length from Minimal encoding (ME) *to* Secure encoding (SE) *to* Low power & Secure encoding (LPSE), while noting an improvement in security and power respectively. Table II shows different encodings for the benchmark FSM and Table III shows possible Hamming patterns assumed by them. Minimal Encodings are unsecure encodings which are used for comparison. As can be seen, the state registers can assume only *two* unique HW values for SE and LPSE whereas *five* unique HW values for ME. The transitions can also assume only *two* or *three* unique HD values for SE and LPSE in comparison to *five* unique HD values assumed by ME strategy. Similarly, unique SD values for SE and LPSE range within *four* and *six* whereas for ME there can be *eleven* unique values. Moreover, the estimated Switching Activity and Simulated Power in Table III show improvement for Low Power & Secure encodings over Secure encodings.

Therefore, security for FSMs can be achieved along with low power consumption and the designer can choose the level of security at the cost of area and power.

States	Minimal Encoding	Secure Encoding	Low Power & Secure Encoding
st0	1010	01110	101101
st1	1100	00100	100100
st2	0100	10110	100001
st3	0111	11001	100111
st4	1111	01000	000101
st5	0001	11010	000110
st6	1000	01101	110110
st7	0011	11100	110101
st8	1110	10000	010100
st9	0000	10011	001100
Estimated SA	**0.443**	**0.5148**	**0.445**
Avg. Simulated Power(uA)	**1.59**	**1.56**	**1.41**

TABLE II: Example State Assignment for BBARA for different encoding strategies

Attack Model	Different Values Assumed by State Register		
	Minimal	Secure	Low-power & Secure
Hamming Weight	(0, 1, 2, 3, 4)	(1, 3)	(2, 4)
Hamming Distance	(0, 1, 2, 3, 4)	(0, 2, 4)	(0, 2)
Switching Distance	(0, 1, 1-δ, 2-δ, 2-2δ, 3, 3-δ, 3-2δ, 3-3δ, 4-δ, 4-4δ)	(0, 2, 2-δ, 2-2δ, 4-δ, 4-2δ)	(0, 2, 2-δ, 2-2δ)

TABLE III: Values assumed by *BBARA* FSM State Register for three different encoding strategies

IV. Boolean SAT Formulation

Given the security and power constraints, the problem of generating the state assignment can be transformed into a Boolean satisfiability (SAT) problem. Let's assume that the given FSM has a set of M states:

$$S = \{s_1, s_2, s_3, ..., s_M\}$$

Let the division of 'M' states into 'I' S_L sets and transitions into 'J' T_L sets be obtained. Given the binary encodings $b_m = E(s_m)$ of state s_m of bit-length R, let's define a predicate $DistinctStates$ for states s_1 and s_2 being distinct by defining $HD(s_1, s_2)$ should be a positive integer:

$$DistinctStates(s_1, s_2) = \sum_{r=1}^{R} (b_1^r \bigoplus b_2^r) \geq 1 \tag{11}$$

Let's define a predicate $EqualHW$ for two states s_1 and s_2 having the same Hamming Weight:

$$EqualHW(s_1, s_2) = \sum_{r=1}^{R} (b_1^r) \equiv \sum_{r=1}^{R} (b_2^r) \tag{12}$$

For any S_i, all $s_m \in S_i$ (∀i $\in [1, I]$) should have equal Hamming weight i.e. every pair in S_i must satisfy predicate $EqualHW$. Similarly let's define predicate $EqualHD$:

$$EqualHD((s_1, s_2), (s_3, s_4)) = \sum_{r=1}^{R} (b_1^r \bigoplus b_2^r) \equiv \sum_{r=1}^{R} (b_3^r \bigoplus b_4^r) \tag{13}$$

All transitions in T_j set must have equal Hamming distance i.e. every pair of transitions in T_j (∀j $\in [1, J]$) must satisfy $EqualHD$.

To obtain minimum switching activity in the FSM transitions, let's define predicate *CorrectSA*. SA is defined using equation 8 and our aim is to minimize it:

$$CorrectSA(SA) = \sum_{j=1}^{l} HD(T_j) \times W_j \equiv SA \tag{14}$$

where, W_j is the sum of w_{mn}'s of all transitions in T_j.

For a valid solution, we need to find state-encodings for a given R bit-encoding and a given SA (Switching Activity) value, such that it satisfies all the above defined predicates.

In this research we have used the Z3 SMT (Satisfiability Modulo Theories) solver [12] to solve for a valid state assignment, although any SAT solver can be used. Z3 is a high performance solver which has the ability to solve models comprising of bit-vector variables and constraints in terms of those vectors.

V. Algorithm for Secure Low-Power State Assignment

The obtained constraints are applied to the SMT Solver to find a valid set of state encodings within the range [R_{min}, R_{max}] discussed in Section IV.

Note that, the SMT solver doesn't accept fractional values. Hence, we have scaled all the STP values by a common factor so that they can be represented as integers. This results in scaling up of SA, SA_{min} and SA_{max} to larger integer values, minimizing errors introduced due to rounding off fractional STP values.

Algorithm 1 finds secure and low power FSM state encodings. For each encoding size R, the algorithm finds an encoding (if it exists) to yield the minimum possible switching activity. Upon finding a satisfiable solution, line 17 updates SA_{max} to current SA value, which reduces search time by excluding encoding solutions with higher switching activity.

Algorithm 1 Generate Secure and Low Power Encodings

```
 1: Inputs: FSM with $M$ states and Its STG
 2:         $S_L$ and $T_L$ sets Generated for given $l$ value
 3: Output: Low Power & Secure State Encodings
 4:
 5: calculate $SA_{min}$ using scaled STP values
 6: calculate $SA_{max}$ using scaled STP values
 7: for $R = R_{max}$ to $R_{min}$ step $-1$ do
 8:     for $SA = SA_{min}$ to $SA_{max}$ step $+1$ do
 9:         initialize SMT solver S
10:         initialize State $s_i, i \in [1, M]$: bit-vector of size R
11:         S ← Add 'Distinct States' Constraint
12:         S ← Add '$S_L$ Set' Constraints
13:         S ← Add '$T_L$ Set' Constraints
14:         S ← Add Switching Activity Constraint(Eqn. 8)
15:         if (S is Satisfiable) then
16:             print State Encoding Solution
17:             $SA_{max} = SA$
18:             break (from the inner for-loop)
19:         end if
20:     end for
21: end for
```

VI. Experimental Results and Analysis

All experiments were performed using over 100 benchmarks from BenGen [13] and MCNC [14] suites. Each FSM was encoded with different encodings, minimal binary and secure encodings with different levels of security, using restructured method and the two proposed low-power methodologies. These encodings were converted to Verilog and then synthesized in 90nm CMOS technology using Synopsys DC Compiler. After converting these gate-level netlists to Spice using a Verilog-to-Spice converter, power simulation was performed using 1000 random input vectors generated using a stimuli-generator. Every FSM implementation resulting from different state encodings for all three security-methodologies were simulated using Nanosim to obtain power traces.

Best-case attack model data was generated using the same input stimuli to perform statistical analysis. Hamming weight, Hamming distance and Switching distance data were calculated for the 1000-vector input stimuli for every implementation of the FSM. Perl scripting was used to calculate mutual Information (MI) between the power traces obtained using Nanosim and the attack model data generated. Figure 3 shows the range of sizes of the benchmark FSMs used in the experimentation, in terms of number of states and transitions. Input bit-length for these FSMs ranged between 1 and 16.

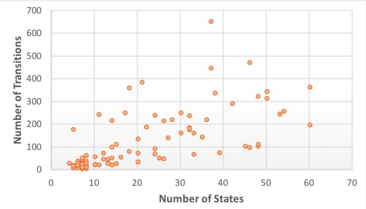

Fig. 3: Size of Benchmark FSMs as States vs Transitions

We report results for (1) the original FSMs without restructuring self-loops, and (2) low-power design for original FSMs, and compare them both with restructured FSMs with transformed self-loops [8].

A. Security Analysis

We use Mutual Information (MI) (between the power traces and the HW, HD or SD models) as a distinguisher to measure security ($MI \geq 0$) against all three forms of attacks [15].

In FSMs without restructuring (i.e. original FSMs), perfect security ('Zero' MI) can only be achieved against HW attacks due to presence of self-loops in FSMs. HD and SD attacks can reveal self-loops within the FSMs. Figure 4 shows reduction in information leakage (in terms of MI) as security increases. Since encoding length requirement increases for higher security, the plot demonstrates the basic trade-off between security and area.

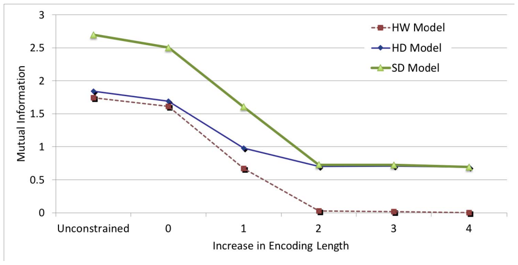

Fig. 4: Average Mutual Information between Power Side-channel and Hamming Models

Figure 5 shows the increase in the encoding bit-length (with respect to unconstrained minimal length encoding) for maximum possible security, where all states and transitions have the same Hamming weight, Hamming distance and Switching distance for restructured, original and original low power FSMs. Difference in encoding length (R) requirement can be seen when loops are restructured. Restructuring the FSMs increases the encoding length by 40-70% depending on the level of security chosen, whereas original FSMs only increase encoding length by 15-40%.

B. Area Analysis

Encoding with varying bit-lengths determine the synthesized FSMs areas. Figure 6 shows the normalized area increase for maximum security for both original and restructured FSMs with respect to the unconstrained minimal length encodings for the original FSMs. On average, for maximum security (ie. MI(power,HW)=0), area increases only by a factor of 1.37 for the original FSMs. For restructured FSMs, an increase in area by a factor of 2.04 was observed, for maximum security (ie. MI(power,HW)=0 and MI(power,HD)=0).

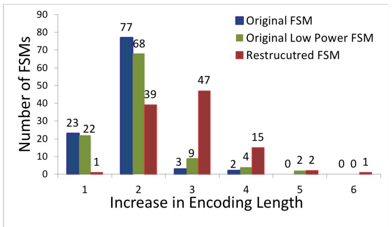

Fig. 5: Encoding Length for Maximum Security

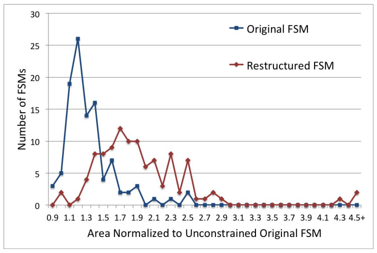

Fig. 6: Normalized Area for Maximum Security

Additional low-power switching activity constraint introduces a slight increase in the encoding bit-requirement. 30% of benchmark FSMs observed no increment, while the rest observed a 5% increase on average. The average layout area requirement increased by 4% due to these constraints.

C. Power Analysis

Average power determined using NanoSim simulations shows a reduction when low power techniques are implemented along with security measures. Figure 7 shows the average normalized power consumption with respect to unconstrained minimal length encodings for three types of encodings. Restructured FSMs require much higher power than original FSMs due to increase in area. On average, power consumption increases by a factor of 3.4 for secure restructured FSMs [8], compared to a factor of 1.6 for secure original FSMs. Switching Activity constraint further reduces power consumption for every benchmark within the range of 0-40%. For maximum security (ie. MI(power,HW)=0), on average a reduction of 15% is observed. It should be noted that these low-power techniques do not result in any security trade-off and have no substantial increase in synthesized area.

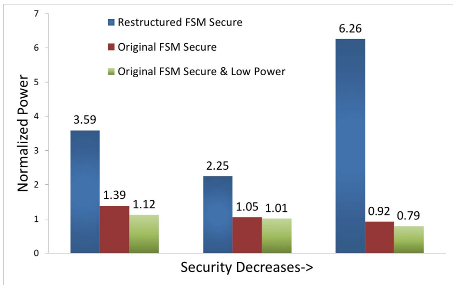

Fig. 7: Normalized Power for Graded Security

Figure 8 compares power profiles for different security levels in a single *'STYR'* MCNC benchmark FSM, which consists of 30 states and 93 transitions. Restructured secure designs have much higher power consumption than original secure designs. Power consumption further decreases in original FSM for low power & secure implementation, even though encoding length increases for maximum security (ie. MI(power,HW)=0).

VII. Conclusion

This work proposed low power FSM encoding methods on a power attack resistant design methodology with user-defined security metric. Two kinds of improvements with and without security trade-off were achieved using constrained state assignment. The security against HW model range between 18-100%, while it ranged between 30-58% against HD model and between 27-60% against SD model. The overall increase in encoding length of a typical benchmark ranges from 15-55%, depending on security level and power optimization. Reduction in power consumption by more than 50% is observed for all FSMs with respect to restructured FSMs, and additional low-power constraint introduces further reduction in power ranging between 4-20% depending on security level.

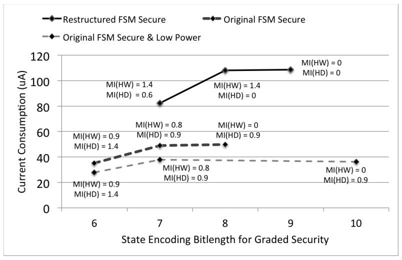

Fig. 8: Power Profile of *'STYR'* MCNC benchmark FSM

References

[1] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer Science & Business Media, 2008.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in cryptology CRYPTO99*. Springer, 1999, pp. 789–789.

[3] J. Smith, K. Oler, C. Miller, and D. Manz, "Reverse engineering integrated circuits using finite state machine analysis," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[4] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware Trojan horse detection using gate-level characterization," in *Design Automation Conference, 2009. 46th ACM*. IEEE, 2009, pp. 688–693.

[5] M. A. Bahnasawi, K. Ibrahim, A. Mohamed, M. K. Mohamed, A. Moustafa, K. Abdelmonem, Y. Ismail, and H. Mostafa, "ASIC-oriented Comparative Review of Hardware Security Algorithms for Internet of Things Applications," in *Microelectronics (ICM), 2016 28th International Conference on*. IEEE, 2016, pp. 285–288.

[6] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," in *Proceedings of the conference on Design, automation and test in Europe-Volume 1*. IEEE Computer Society, 2004, p. 10246.

[7] L. Yuan and G. Qu, "Information hiding in finite state machine," in *Int. Workshop on Information Hiding*. Springer, 2004, pp. 340–354.

[8] R. Agrawal and R. Vemuri, "On State Encoding Against Power Analysis Attacks for Finite State Controllers," in *International Symposium on Hardware Oriented Security and Trust*. IEEE, 2018, pp. 181–186.

[9] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration, the VLSI journal*, vol. 40, no. 1, pp. 52–60, 2007.

[10] M. Borowczak and R. Vemuri, "S* FSM: A Paradigm Shift for Attack Resistant FSM Designs and Encodings," in *BioMedical Computing,2012 ASE/IEEE International Conference on*. IEEE, 2012, pp. 96–100.

[11] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, "Power Estimation Methods for Sequential Logic Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 3, pp. 404–416, 1995.

[12] M. LLC. (2005) Z3 an efficient theorem prover. [Online]. Available: http://www.rise4fun.com/z3/tutorial

[13] L. Jozwiak, D. Gawlowski, and A. Slusarczyk, "An Effective Solution of Benchmarking Problem:FSM Benchmark Generator and its Application to Analysis of State Assignment Methods," in *Digital System Design,2004. Euromicro Symposium on*. IEEE, 2004, pp. 160–167.

[14] S. Yang, *Logic Synthesis and Optimization Benchmarks User Guide: version 3.0*. Microelectronics Center of North Carolina (MCNC), 1991.

[15] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual Information Analysis," *Cryptographic Hardware and Embedded Systems–Cryptographic Hardware and Embedded Systems 2008*, pp. 426–442, 2008.