Optimization Of Logic Obfuscation Technique For Hardware Security

K.N.Baluprithviraj, S.Vijayachitra

Abstract: Logic locking has been effectively utilized for ensuring digital circuits against IC theft. Present logic locking technique offer significant security advantage in terms their Hamming Distance((HD) which is the distance between original and logic locked outputs and resilent to key-sensitization attack. But no logic obfuscation methods available for considering HD, resilent to attacks, area, power and delay overhead. To solve this problem it is proposed a new Particle Swarm Optimization(PSO) optimized GDI Obfuscation Cell(OC) technique for identifying best location of obfuscation cell insertion considering HD and design overhead parameters of circuits. Simulation results on ISCAS–89 benchmark circuits show that high levels of security are achieved through a well formulated obfuscation scheme at less than 10% area, power and delay overheads

Index Terms: Gate Diffusion Input, Hardware Security, Intellectual Property, Logic Obfuscation, Particle Swarm Obfuscation, Reverse Engineering, System on Chip

1. INTRODUCTION

Hardware Intellectual Property (IP) cores have become known as an integral part of contemporary System-on-Chip (SoC) blueprints. Though, IP dealers are facing foremost challenges to defend hardware IPs and to avoid revenue loss owing to IP piracy. To thwart the IP disobedience various techniques have been recommended on mutually RTL- level and hardware implementation level. For instance, an unkind foundry may include hardware Trojans into fabricated chips. The distributed IP cores may well contain malicious logic and/or design errors which could be utilized by attackers after the IP cores are joined together into SoC platforms. The idea of hardware security was properly introduced after the appearance of hardware Trojans and the following countermeasures to take the edge off or prevent this sort of risk. Hardware security was a phrase which initially referred to hardware Trojan designs, classification, recognition, and separation where the independence foundries were treated as the major hazards. Consequently, the widened hardware Trojan recognition methods frequently focus on the post-silicon stage with importance on the security enrichment of testing methods. Given the fact that unauthorized IP cores might be an additional attack vector for malicious logic inclusion, the security of pre-synthesis designs turns out to be equally innermost. Following this demand, pre-silicon circuit protection techniques have also been developed. By this way the logic obfuscation plays an important role in the hardware security.In current scenario, the hardware securities mean to annoy the theft, overbuilding, and RE by obfuscating or masking. At any rate, they practice the ill effects of few concerns. For such concerns, an investigation of present-day anti-piracy, anti-overbuilding and anti-Reverse Engineering (RE) methods are executed by a proficient obfuscation technique. The principle of logic obfuscation is collapsed when if an attacker can find out the secured secret keys used for security obfuscation. By finding the keys, any individual can easily decode the functional netlist and also duplicate copies and then trade it as illegally. For the high efficient key insertion in the circuit or to perform the logic locking in the netlist, it is concentrated on the key gate locations and the strategies that an attacker may not decode the key bits. These types of key-insertions are driven by an interference graph techniques [2], which has taken the high execution time of key gate insertion. To handle such concerns, then the weighted logic locking [14] is by creating an immune to the key-sensitization attack. This is a result of the technique that key gate inputs are not directly driven to key gates, but are mutually shared in control gates first (i.e., each and every key input interferes straight away with at least another one). Likewise it thwarts the key sensitization attacks and produces an efficient output corruption of key gates.

To best of our knowledge, we have proposed a new trending concept for the obfuscation with an efficient key insertion based on the PSO technique on considering two phases: 1) identification of the best locations (circuit nodes) to insert key gates, and 2) key- and control-gate insertion. Based on the literatures like interference graph techniques [2] and weighted logic locking [14], it is improving an effectiveness of the key-gate insertion in the netlist. In this brief, an optimization technique which is named as Partial Swarm Optimization (PSO) algorithm to the digital logic obfuscation technique for the hardware security was proposed. This proposed system of PSO based Obfuscation Technique (PSO-OT) which gives the best solution of positions and the strategies for key insertion. This PSO-OT achieves a high hardware security and low area, power, and zero performance overheads to frustrate the embedded field and also care for the third party IP cores.

2 RELATED WORKS

Some of the related techniques used for the hardware security in existing are given as follow. Reverse engineering: This technique [3] is exploration of the item, actions performed in the framework, examine the structure and stuffs used to construct framework and focus on interconnections and netlist. This is used in long drive equipment, for instance, military services frameworks, ships, atomic reactors. PC based obfuscation [4]: It is the principle of passive hardware metering approach that transfers a unique mark to every Integrated Circuit (IC) effectiveness by coordinating a slight programmable part to the ASIC to prevent IC overbuilding. The

- *K.N.Baluprithviraj, Assistant Professor, Department of EIE, Kongu Engineering College, Perundurai-638060, Erode, Tamil nadu, India, Ph.:04294226414. E-mail:baluprithviraj@gmail.com*
- *S.Vijayachitra, Professor, Department of EIE, Kongu Engineering College, Perundurai-638060, Erode, Tamil nadu, India, Ph.:04294226546. E-mail:dr.svijayachitra@gmail.com*

PC based obfuscation can be used to remain the IC from figuring out. HARPOON[6]: It is consist of two strategies i.e., obfuscation based IP security and validation based IP security. 1) In the obfuscation based IP security, a HDL code is used to exploit the influence of human readability. The code is reformatted by altering the interior net names and vacating the remarks. 2) In the validation based IP security, a digital mark is implanted in the design. Such automated marks are known as the IP trade. Practical Logic Obfuscation[9]: In the existing obfuscation structure as shown in Fig. 1, the secret key information is not revealed by the structure of OC even if the gate-level netlist was taken out by the image processing-based RE, so the attackers cannot easily know the logic function of OCs.

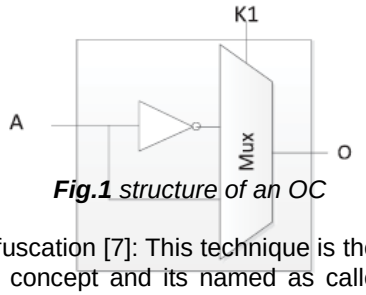

Fig.1 *structure of an OC*

GDI based obfuscation [7]: This technique is the previous work in obfuscation concept and its named as called configurable Gate Diffusion Input (GDI) which is developed to boost the security of hardware IP. Configurable GDI based obfuscated cell includes additional gates in the logic lane of the circuit with least amount of overheads to safe and secure an IC from piracy and overbuilding. This technique would substitute an inverter with the GDI cell or include the GDI cell into some wire of gate level netlist. . For the combination of {K1, K2} in the specified cell, {1 0} act as a inverter, and {0 1} act as a wire. GDI OC circuit is shown in fig.2.

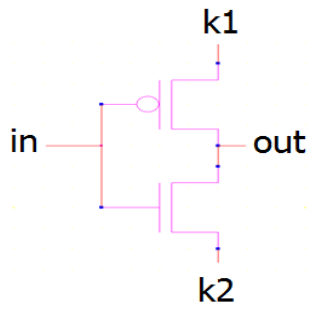

Fig.2 *GDI OC Circuit*

Interference Graph based Key insertion techniques [2]: This paper contributes

- a molest on logic obfuscation based on IC parts
- approaches used by an invader to decode keys based on their interference
- an algorithm for a logic obfuscation based on key gates interference graph

To place in key gates, form an interference graph of key gates in this paper. In this generated graph, every node symbolizes a key gate and a border connects to two nodes, when if two gates interfere. Inaccessible key gates are representing with secluded or isolated nodes. A run of key gates is denoting as a single node. Non-changeable key gates are connected with non-alterable edges. At the same time as changeable key gates are linked with variable edges. In sequence dynamic key gates are connected by two edges; a non- alterable edge arises from the key gate that is non-changeable and a variable edge arises from the key gate that is changeable.

3 PREFACE

From the above issues in the hardware security can be resolved by using the logic obfuscation techniques. To construct an obfuscation process in an effective manner, here, switching the novelty into the PSO algorithm for the hardware security. By using the optimization techniques like Particle Swarm Optimization, it can easily achieve the best solution factors for key gates insertion i.e., it shows the particular gates to be obfuscated in the circuits. So the level of computation complexity is reduced and produces the efficient security in the embedded systems.

3.1 Particle Swarm Optimization

It is an advanced working out model developed by Kennedy and Eberhart [1]. A Particle Swarm Optimization (PSO) is a inhabitants-based stochastic optimization algorithm mock-up by the simulation of the common behavior of bird groups. Swarm Intelligence (SI) is a novel dispersed intelligent model for working out of optimization troubles that at very first took its brainwave from the biological illustrations by swarming, grouping and directing events in vertebrates. PSO integrates swarming deeds which is observed in flocks of birds, trains of fish, or flocks of bees, and also in human communal behavior, from which the proposal is emerged. Therefore, the PSO is a population-based optimization approach, which could be employed and also applied effortlessly to resolve a variety of optimization difficulties. In this approach, an each clarification of the optimization issues are regarded as a bird in the hunt space, which is called particle. Each particle has a speed by which the direction and distance travelled by the particle is determined, and a strength that is determined by the optimized task. The particles hunt in the solution room by tracking the optimal particle at present. Jinrong Zhu has suggested a modified particle swarm optimization approach. In this approach, each particle prefers its inertial aspect according to the forthcoming degree between the strength of itself and the optimal particle. The issues in the hardware security can be resolved by using the logic obfuscation techniques based on PSO algorithm. To construct an obfuscation process in an effective manner, switching the novelty into the PSO algorithm is carried out for hardware security. By using PSO, the best solution factors for key gates insertions are obtained, i.e., key gates shows the particular gates to be obfuscated in the circuits. So, the level of computation complexity is reduced and the efficient security is produced in the embedded systems. The proposed PSO procedure defines every particle as potential key to a crisis in D-dimensional space. Each one of the particles is familiar with its best value up to now (pbest) and its spot. Furthermore, every particle knows the best cost until now in the group (gbest) amongst pbest. This information is correlation of knowledge for how the further particles in the region have been performed. Each and every particle tries to alter its location and sites using the following in sequence:

- The distance among the both present position and pbest
- The distance among the both position and gbest

This amendment can be symbolized by the concept of speed

velocity. This speed Velocity of each and every mediator can be adapted according to the below equation (1) in Inactivity Weight Approach (IWA).

$V_{K+1} = W * V_K + C_1 * Rand1 * (P_K – X_K) + C_2 * Rand2 * (G_K – X_K)$ (1)

where, W – non-negative inactivity factor, V_K – speed velocity of article, X_K -present position of particle, C_1 - the cognitive component for relative influence, C_2- determine the communal component for relative influence of the, P_K - pbest of particle , GK -gbest of the particle, Rand1 , Rand2 - random numbers which are used to preserve the range of the population, and are consistently distributed in the interval [0,1].From the equation (1), a particle makes a decision where to shift next, considering its own knowledge, which is the recollecting memory of its best precedent position, and the skill of its most victorious particle in the swarm. In the particle swarm technique, the particle looks for the solutions in the crisis hole with a range [−s, s]. Each item updates its location according to equation (2).

Xk+1 = Xk + Vk+1
(2)

4 PROPOSED TECHINQUES

In this proposed method, it is chosen the inverters and wires for replacement with GDI OC [7] using a partial swarm optimization algorithm. The proposed PSO-OT algorithm is to find and select the inverters and wires locations for the replacement in the noncritical path of the original netlist. For an efficient optimization, it is decided to use as source for the metric as fitness, (i.e) the fault impact fitness metric, since it is very effective in identifying a particle that, when its complementary, affects most of the netlist outputs. To compute fitness for fault impact, a set of random patterns should be fault imitation for the both stuck-at 0 (s-a-0) and stuck-at-1 (s-a-1) faults, at all netlist inputs and gate outputs. For each one of these netlist nodes, the volume of patterns that identify the equivalent s-a-0 fault (NoP0). In addition, the total number of outputs that gets defected by that fault (NoO0) is calculated. Similarly, NoP1 and NoO1 are computed. From the weighted logic locking [14], an author solved a fault metric, in which NoP0, NoO0, NoP1 and NoO1 are calculated when only the outputs that have not been collapsed by any of the previously chosen particles. Such outputs are called as “unemployed” and the corresponding quantities are denoted as NoP0_unemp, NoO0_unemp, NoP1_unemp and NoO1_unemp. Particularly, to calculate a pattern in NoP0_unemp or NoP1_unemp it should identify the related fault at an unemployed netlist output. For the calculation of an output, where a fault is identified, in NoO0_unemp and NoO1_unemp should be unemployed. Therefore the fitness metric unemployed -output-aware fault impact (FIunemp for briefness), and compute it as follows:

FIunemp = NoP0_unemp · NoO0_unemp + NoP1_unemp · NoO1_unemp (3)

When all the outputs become employed, they are all reset to the unemployed state, and the next particle selection takes into account all outputs, it executes the identification of best key selection when the outputs are collapsed after this point are once again marked as employed and so on.

As considering a fault impact metric as a fitness function, this research work proposing the algorithm to identifying the best location for the key-gates insertion in the netlist which is mentioned as ‘pbest’. Furthermore, for the key and control gate insertion in the netlist, every node knows the best cost until now in the group (gbest) amongst pbests. Therefore the updating condition for identifying the key-gates and the control gate insertion are expressed below

i.e., if $f(P_K) > pbest$, then $pbest = P_K$.
if $f(G_K) > gbest$, then $gbest = G_K$.
(4)

where f(x) is the objective function to be optimized.

This proposed PSO algorithm can be performed searching for the optimal solution in each steps are given below.

Algorithm 1: proposed algorithm of PSO-OTs

1: Initialization a netlist of logic gates with the position and speed velocities on M dimensions in an any circuits.
2: loop
3: For each gate netlist, calculate the desired optimization fitness function as in (3).
4: Compare FIunemp fitness assessment with its local best fitness. If present FIunemp value is better, then set its local FIunemp best fitness equal to the present FIunemp position and P_K equal to the current location X_K.
5: Identify the FIunemp in all of the OC or key gate insertion with the best strength so far, and assign it to global best FIunemp computation.
6: Change the velocity and location of the OC or key gates according to (1) and (2).
7: Update pbest and gbest when condition is met.
if $f(P_K) > pbest$, then $pbest = P_K$.
if $f(G_K) > gbest$, then $gbest = G_K$.
8: The algorithm replicates the steps 3 to 7 until certain concluding conditions are fulfilled, such as a pre-defined number of iterations.
9: end loop

The proposed PSO-OT algorithm using this strategy to seek in huge range and the inexact position of the optimal insertion solution is confirmed rapidly and investigate in small range in the delayed iterations consecutively that the exact solution for key gates and a control gate insertions are found. A random number (Randn) is used in assessing the inactivity weight in the algorithm one by one to jump out from local optimum and a minimum inactivity weight factor is used to prevent the untimely function.

5 EXPERIMENTAL RESULTS

To evaluate the performance of proposed PSO-OT obfuscation technique, it performed a set of experiments performed on ISCAS benchmark circuits. MATLAB tool has been used for PSO and fault simulation experiment. Insertion of GDI OC cell is done on the basis to achieve 50% HD (hamming distance) between original and locked circuit for every benchmark circuit. A Synopsys software tool was used for obtaining the area, power and delay overhead of the proposed method. For fair comparison it compared with weighted logic locking method (insertion of key gates by key gate control algorithm), practical obfuscation (OC insertion done by time driven algorithm). Table 1 shows the comparison of area, delay and

power for the proposed technique with existing methods. The performance analysis of PSO based obfuscation technique on the area, power and delay consumption have been compared and shown in fig.3, fig. 4, fig. 5, fig. 6 and fig.7.

TABLE 1 COMPARISON *OF AREA, DELAY AND POWER FOR THE PROPOSED AND EXISTING METHOD*

Circuit	[14] Karousos et al.			[15] J. Rajendran et al.			Proposed		
	Area	Power	Delay	Area	Power	Delay	Area	Power	Delay
S27	27	7.30	0.30	27	7.2	0.30	37	6.9	0.27
S344	971.18	76	0.37	969.4	75.2	0.36	767.2	74.3	0.35
S386	693.79	75.2	0.50	682	76.6	0.48	660	73.2	0.45
S400	1189	74.1	0.32	1185	73.5	0.29	1165	71.9	0.26
S444	1194	80.3	0.12	1110	79.5	0.10	1090	78.3	0.08
S510	1235	95.5	0.91	1210	1232	0.88	1118	94.3	0.85

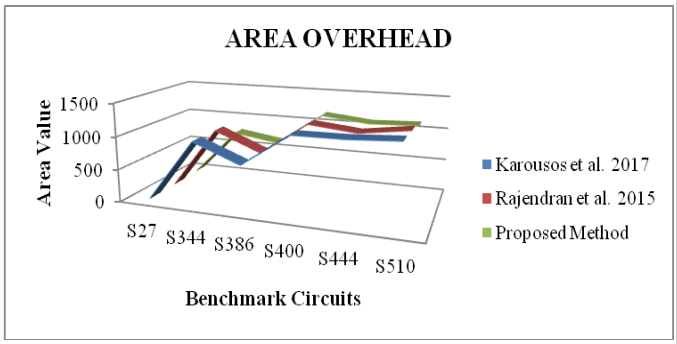

Fig. 3 *Comparison of benchmark circuits based on area consumption*

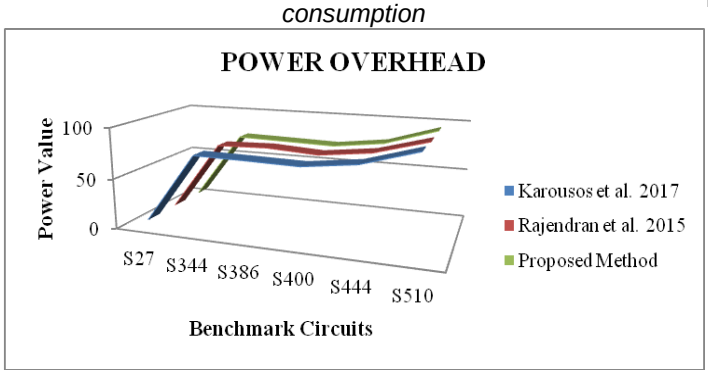

Fig. 4 *Comparison of benchmark circuits based on power consumption*

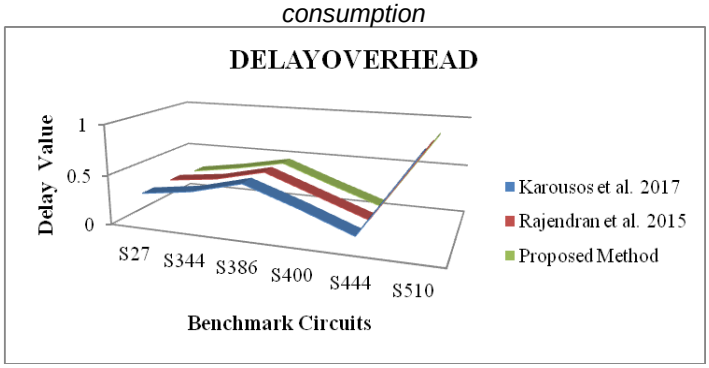

Fig. 5 *Comparison of benchmark circuits based on delay consumption*

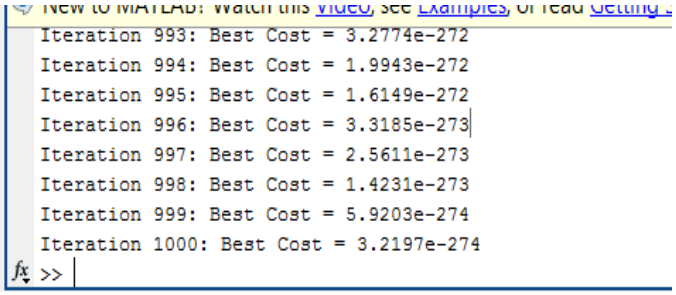

Fig. 6 *Best Cost for the location of key gates in benchmark circuit*

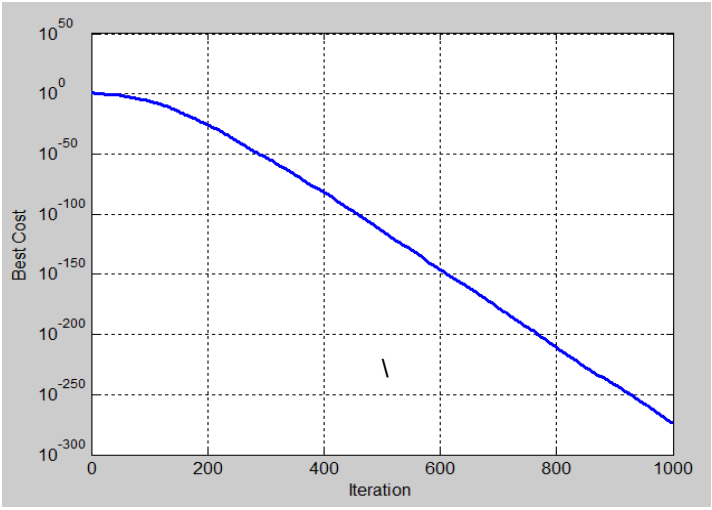

Fig. 7 *Comparison of iteration with best cost*

From the Table 1 it shows that proposed PSO technique for identifying best location of OC achieves high hamming distance with minimum number of OC requirements. The proposed technique based on PSO gives minimum overhead in area, power and delay parameters.

6 CONCLUSION

A PSO optimized logic obfuscation technique was presented in this paper. Based on optimized best location identified by swarm intelligence used to replace a inverter and wires in the circuit, which target high HD rates and minimum area and delay overhead. Designers can increase security level by setting HD levels without raise in the area, delay and power overhead.

REFERENCES

[1] J. Kennedy, R. Eberhart, “Particle Swarm Optimization”, in proceeding of IEEE International Conference of Neural Networks, pp. 1942-1948,1995.

[2] J.Rajendran, pino. Y, O.Sinanoglu and R.Karri, “Security Analysis of Logic Obfuscation”, proceedings of the 49th Annual Design Automation Conference, 2012

[3] R. Torrance and D. James, “The State-of-the Art in Semiconductor Reverse Engineering”, in proceeding of 48th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 333-338, 2011.

[4] A. Baumgarten, A.Tyagi and J.Zambreno, “Preventing IC piracy using reconfigurable logic barriers”, IEEE Des. Test. Computer, vol. 27, no.1, pp.66-75, 2010.

[5] A.Kumar , “Area-delay-power Efficient PSO based full

adder in different technologies", International Conference on Communication and Signal Processing (ICCSP), 2016

[6] R.S. Chakraborty and S. bhunia, "HARPOON: An obfuscation based SoC design methodology for Hardware Protection", IEEE Trans. Comput. Aided Design Integrated Circuits System vol.28, no.10, pp.1493-1502, 1999.

[7] Baluprithviraj Krishanaswamy Natarajan and Vijayachitra Senniappan, "Logic Obfuscation Technique using Confiigurable Gate Diffusion Input for Improved Hardware Security", IEICE Electronics Express, vol.15, no.19, pp.1-10, 2018.

[8] J.O. Ushie, O.J.A Etim, I. Prosper, "Optimization Digital Combinational Circuit using Particle Swarm Optimization Technique", Latin American Journal of Physics Education, vol.6, no.1, pp.72-77, 2012.

[9] Z.Jhang, "A Practical Logic Obfuscation Technique for Hardware Security", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.24, no.3, pp.1193-1197, 2016

[10] I.S.Abukhater, "Circuit Techniques for CMOS Low Power High Performance Multiplers", IEEE Journal Solid-State Circuits, vol.31, 1996

[11] K.Yano, "Top-down pass transistor Logic Design", IEEE Journal Solid State Circuits, vol.31, 1996

[12] Y.Lao, "Obfuscation DSP circuits via High-Level Transformations", IEEE Trans. Very Large Scale Integration(VLSI) system, vol.23, 2015

[13] V.G.Gudisc and G.K.Venayagamoorthy, "Evolving Digital Circuits using Particle Swarm", Proceedings of IEEE Joint Conference, vol.1, pp.468-472, 2003

[14] Karousos et al., "Weighted Logic Locking: A New Approach for IC Piracy Protection", IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), 2017.

[15] J.Rajendran et al., "Fault Analysis-based Logic Encryption", IEEE Trans. Computer, vol.64, no.2, pp.410-424, 2015