HARDWARE SECURITY THROUGH DESIGN OBFUSCATION

[1 Obfuscation-based secure SoC design 2](#_Toc163118718)

[1.1 Analysis of Netlist Obfuscation. Obfuscation Metric. 2](#_Toc163118719)

[1.2 System-level Obfuscation Methodology 2](#_Toc163118720)

[1.2.1 Choice of Optimal Set of Nodes for Modification 2](#_Toc163118721)

[1.2.2 The HARPOON Design Methodology 3](#_Toc163118722)

[1.3 Results 4](#_Toc163118723)

# Obfuscation-based secure SoC design

## Analysis of Netlist Obfuscation. Obfuscation Metric.

Let, be a function of the set P1 of primary inputs and state-element outputs and g be a function of a set P2 of primary inputs and state element (SE) outputs. Let , |P1| = p1, |P2| = p2, |P| = p, P1 P2 = Γ and |Γ| = γ = p1 + p2 − p. Further, let g be a Boolean OR function with p2 inputs. Then, for (2p2 − 1) of its input combinations, g is at logic-1. Consider en = 1. Then, for all these (2p2 − 1) input combinations of P2 , fmod = , causing a failing vector. Corresponding to each of these (2p2 − 1) combinations of P2, there are (p1 − p) other independent primary inputs to f. Hence, the total number of failing vectors when = 1 is:

|  |  |
| --- | --- |
|  | (1) |

For the other “all zero” input combination of P2, = 0. Let the number of possible cases where = 1 at = 0 be Ng0. Then, the total number of failing input patterns:

|  |  |
| --- | --- |
|  | (2) |

In the special case when , Ng0 is given simply by the number of possible logic-1 entries in the truth-table of .

The total input space of the modified function has a size 2γ. The *obfuscation metric* (M) is defined as:

|  |  |
| --- | --- |
|  | (3) |

The “+1” factor in the denominator is due to the *en* signal.

## System-level Obfuscation Methodology

### Choice of Optimal Set of Nodes for Modification

The practical level of obfuscation is estimated by the amount of verification mismatch reported by a Formal Verification based equivalence checker tool. Formal equivalence checker tools essentially try to match the input logic cones at the state-elements and the primary outputs of the reference and the implementation. Thus, in determining the suitability metric for a node as a candidate for modification, both these factors need to be considered. The following metric is proposed as the suitability metric for a node:

where Mnode - obfuscation metric for a node;

FI and FO - the number of nodes in the fan-in and the fan-out cone of the node, respectively;

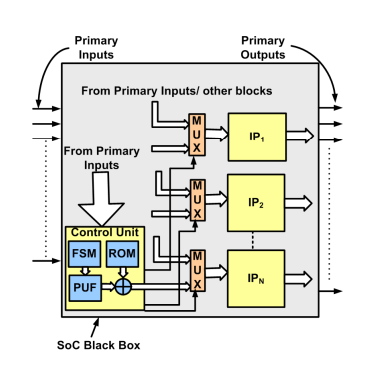
FImax and FOmax - the maximum number of fan-in and fan-out nodes in the circuit netlist and are used to normalize the metric;

ω1 and ω2 - weights assigned to the two factors, with 0≤ω1, ω2≤1 and ω1 + ω2 = 1. The values ω1 = ω2 = 0.5 were chosen because they gave the best results in terms of obfuscation, as shown in the next section.

1. The modifying node should have a very large fan-in cone, which in turn would substantial expand the logic cone of the modified node;
2. It should not be in the fan-out cone of the modified node;
3. It should not have any node in its fan-in cone which is in the fan-out cone of the modified node.

### The HARPOON Design Methodology

* A chip design house buys an IP core from an IP vendor (Figure 1), and makes an illegal copy or “clone” of the IP. The IC design house then sells it to another chip design house (after minor modifications) claiming the IP to be its own;
* An untrusted fabrication house makes an illegal copy of the GDS-II database supplied by a chip design house, and then illegally sells them as hard IP;
* An untrusted foundry manufactures and sells counterfeit copies of the IC under a different brand name.



Figure

1 – SoC design modification to support hardware obfuscation. An on-chip controller combines the input patterns with the output of a PUF block to produce the activation patterns

## Results

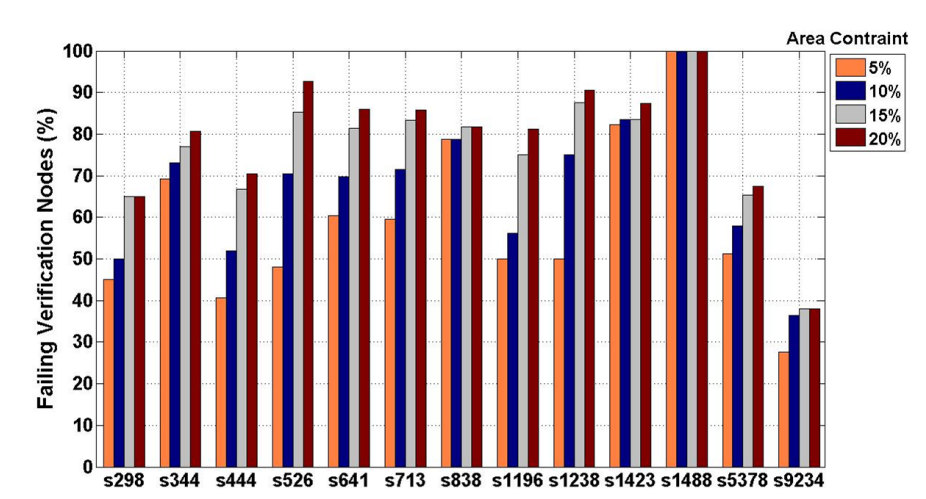
Choice of Scheme

Table

1 – Average Number of Failing Patterns for ISCAS-89 Benchmark Circuits for Different Modification Schemes

|  |  |  |  |
| --- | --- | --- | --- |
| Benchmark | Scheme-1 | Scheme-2 | Scheme-3 |
| S298 | 51 | 158 | 193 |
| S344 | 215 | 1093 | 1233 |
| S444 | 197 | 569 | 7732 |
| S526 | 146 | 485 | 1186 |
| S641 | 598 | 2491 | 5135 |
| S713 | 913 | 2918 | 3301 |
| S838 | 382 | 1757 | 5106 |
| S1196 | 2423 | 5382 | 9573 |
| S1238 | 2552 | 5157 | 9511 |
| S1423 | 6431 | 1816 | 28350 |
| S1488 | 333 | 18120 | 1156 |
| S5378 | 13311 | 29482 | 53066 |
| S9234 | 13862 | 30385 | 53365 |
| S3124 | 1344 | 32252 | 64573 |

Table 1 shows: the simple node modification scheme using only XOR gates (scheme 1), the theoretically suggested modification scheme employing OR-ing of selected primary inputs (scheme 2) and lastly the low-overhead modification scheme employing random selection of internal nodes avoiding combinational loops (scheme 3). The maximum number of modifiable nodes Nmax for each benchmark circuit was determined considering four different area constraints (5%, 10%, 15% and 20%) (Figure 2).



Figure

2 – Observed verification failures (with application of the HARPOON methodology) for ISCAS-89 circuits