Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 3, No. 3, 2026 (pp. 1-8)

RESEARCH ARTICLE ECEJOURNALS.IN

Hardware Watermarking and Obfuscation Techniques for IP Protection in SoC Design Flows

Nicolas Roux*

Lab Manager, Robotics, ECE Paris University in Paris, France.

KEYWORDS:

SoC Security,
Hardware Obfuscation,
Logic Locking,
IP Watermarking,
SAT Attack Resilience,
VLSI Design,
Hardware IP Protection.

ARTICLE HISTORY:

Submitted : 05.03.2026
Revised : 03.04.2026
Accepted : 10.05.2026

https://doi.org/10.31838/JIVCT/03.03.01

Abstract

The distributed semiconductor supply chain through globalisation of semiconductor design and outsourced fabrication has increased the risks of intellectual property (IP), unlicensed overproduction and reverse engineering in the distributed System-on-Chip (SoC) supply chains. With the growing investment of a third-party block IP blocks and heterogeneous block components in modern SoC platforms, safeguarding design properties in the entire RTL-to-GDSII implementation cycle has become a crucial issue. In this paper, a single, security aware design system, consisting of both proactive hardware obfuscation and reactive hardware watermarking in a typical, Electronic Design Automation (EDA) setting, is proposed. An RTL-level logic locking SAT resilience lock is implemented to create key-dependent functionality and output corruption in the maximum number of conditions with invalid keys. This is followed by inserting a 64-bit constraint-based digital watermark during physical synthesis and routing in order to be used in verification of post-fabrication ownership but without interference to actual functionality. This architecture has been measured to the ISCAS85 benchmark circuits and an AES-128 cryptography core being run on a 45 nm technology node. Experimental evidence shows that the average corruption of its output caused by an incorrect key is near to 50% Hamming Distance and the Power, Performance, and Area (PPA) overhead do not exceed 5%. Further security audit substantiates that sat protocol-based tests become more resistant and that it is resistant to watermark removal through resynthesis and optimization. The layered protection technique suggested has provided scalable and low overhead protection of IP protection in current SoC design flows, including the provision of both functional secrecy, as well as legal traceability.

Author's e-mail: n.roux@ece.fr

How to cite this article: Roux N. Hardware Watermarking and Obfuscation Techniques for IP Protection in SoC Design Flows. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 3, No. 3, 2026 (pp. 1-8).

Introduction

The semiconductor business has been changing dramatically to a horizontally distributed supply chain where design, verification, IP integration and fabrication are commonly handled by various parties in a multiple geographical footprint. However, even though this model will speed up innovation and shorten the time-to-market, it will pose significant security threats to hardware intellectual property (IP). Design safe assets are becoming more vulnerable to backwards-engineering, IP piracy, hardware emulation, overproduction, and malicious editing in all different phases of the System-on-Chip (SoC) lifecycle.[1, 2] Hardware compromises cannot be permanently removed as there is a chance to fix software vulnerability by providing updates or patches. An attack on a hardware IP core can come to pass and interfere with the silicon root of trust, exploiting what is known as secure boot, making it possible to engage in mass counterfeiting or unauthorised copying.[3] The increasing popularity of third party IP (3PIP), chiplet based, and third-party fabrication further expands the attack surface which needs to be effectively defended by strong hardware-level hardware defence mechanisms built right into the design flow.[4] Two major strategies have been actively researched to protect the hardware IP; logic lock-

ing (hardware obfuscation) and hardware watermarking. In logic locking, the functionality of the circuit is limited to cases in which the correct secret key is presented, so that the design cannot be used by an unauthorised user. [5] But the initial XOR/XNOR-based locking schemes were demonstrated to be susceptible to attacks in the format of Boolean satisfiability (SAT) which could leak keys effectively.[6] More complex addressing SARLock, Anti-SAT and Stripped Functionality Logic Locking SFLL-HD enhanced SAT resistance but frequently at a huge overhead in area and timing.[7, 8] In addition, new machine-learning-mediated structural attacks have been able to forecast the location of key-gate, which poses more challenges to unstructured obfuscation techniques.[9] Instead, hardware watermarking is used in the design to integrate ownership signatures into watermarking in order to give forensic evidence in instances of, say, IP disputes.[10] Current watermarking techniques take advantage of constraint-based embedding at high-level synthesis or physical design to enhance indecisiveness to removal-by-resynthesis attacks.[11] However, watermarking will not definitely take away functional use under unauthorised use, they just make it traceable after infringement has happened. Although a lot of research has been carried out in both areas, most of the earlier studies accept the two processes of obfuscation and watermarking as independent processes which incur unnecessary hardware overhead, not co-optimised and not fully secured. The urgency is still on-the-need of having an integrated, security-conscious RTL-to-GDSII design model that optimally exerts proactive protection of functionality coupled with reactive ownership cheques and still achieves acceptable Power, Performance and Area (PPA) limits.[12]

The limitations mentioned by this paper are solved by suggesting a standard hardware security architecture that combines both the watermarking with SAT-resilient logic locking at the RTL stage and the embedding of watermarks into the physical synthesis and routing phases of hardware watermarks into a single framework. The offered strategy guarantees a high corruption of output during the case of wrong key settings, high watermark resistance against optimization attacks, and a low effect on the PPA metrics.

Contributions

The key contributions of the work are as follows:

1. The whole idea is to create a cohesive security-conscience SoC design flow, which involves logic locking and physical watermarking, as part of a conventional EDA pipeline.
2. Such a key-dependent structural locking method so far gives nearly 50% Hamming Distance when keys are used improperly, and therefore resists more efficiently extraction of key by SAT.
3. The 64-bit constraint based routing watermark which is resistant to resynthesis and optimization attacks.
4. Full experimental confirmation of less than 5% PPA overhead and quantitatively increasing complexity of computational attacks.

Related Work

Among the most popular methods of protecting intellectual property of the hardware type, there are logic locking, watermarking, and the related hardware obfuscation. Some of the early logic locking methods added XOR/XNOR key-gate counts to the gate-level netlists to implement key-dependent operations and avoid the unlicensed use of integrated circuits.[1] Although these techniques gave resistance against manual reverse engineering, they were subsequently established to be susceptible to his Boolean satisfiability (SAT)-based attacks, whereby distinguishing input patterns are systematically produced to restore the secret code at the expense of a many-input and many-output complex circuit in an equal measure of time.[2, 3] To help reduce the SAT vulnerabilities, advanced countermeasures were designed. SARLock and Anti-SAT frameworks proposed the minimal output corruption structures that aimed at making key recovering harder.[4] Stripped Functionality Logic Locking (SFLL-HD) also enhanced this further by deliberately taking away desired circuit functionality and, under key-controlled logic, as a method of imposing greater resistance against SAT solvers.[5] There was also the use of interdependent and cyclic locking which enhanced the complexity of the structure through the join of more than two modules in the design.[6] Although these advances have been made, most of the SAT-resilient schemes are heavy in Power, Performance and Area (PPA) overheads that do not readily fit in the resource-constrained SoC platform.[7] In addition, as it has been shown recently, machine-based structural attacks, especially those using Graph Neural Networks (GNNs), can locate key-gate locations and minimise the complexity of attacks, leaving additional vulnerabilities to standalone obfuscation systems.[8] In parallel with the study of obfuscation Hardware watermarking has also been studied as a companion to ownership verification. The signatures of early watermarking techniques were implemented at Register Transfer Level (RTL) or HDL by structural annotations or by adding redundant logic.[9] Nevertheless, these methods were very vulnerable to

elimination by logic resynthesis and opticalizing. In an effort to increase the level of robustness, the later entrance of watermarking constraint-based approaches was presented at higher levels of abstraction with a signature being embedded in a scheduling constraint, resource allocation pattern or routing topology in a physical design.[10, 11] Physical watermarking enhances resistance to removal-by-resynthesis attacks, but it does not mean that only the legitimate functional use of the IP core can be used; it is just that it is used as a fact in the courts. However, the two aspects of hardware obfuscation and watermarking have developed independently, and the available literatures mostly consider them as independent protection systems. Few papers attempt to do them in a coordinated manner and those that do tend to use locking and watermark sequentially without coopting each other, creating unnecessary overhead, routing bandwidth problem, and timing loss.[12] Also, several earlier literatures do not fully test their procedure against the contemporary attack models, especially SAT based attacks with structural learning and AI based de-obfuscation methods. The difficulty remains then, to come up with a common RTL-to-GDSII security system, which can smoothly balance proactive functionality protection with reactive ownership traceability at reduced PPA overhead to be able to sensibly use available SoC frameworks. The current paper fills this gap, with an integrated, security-conscious design process that consists of SAT-resilient logic locking in conjunction with constraint-based physical watermarking on explicit PPA constraints.

Proposed Methodology

Unified Security-Aware Design Flow

The suggested methodology is based on the obfuscation of hardware and watermarking in a typical RTL-to-GDSII Electronic Design Automation (EDA) flow. The primitives of security are proposed in a coordinated way rather than in isolation of protection mechanisms, which is contrasted to the conventional methods of applying both mechanisms, which reduces redundancy to a minimum and power, performance area (PPA) overheads. Figure 1 depicts the general structure of the single security-aware flow. Design flow: The Register Transfer Level (RTL) The functional description of the System-on-Chip (SoC) is logic-locked at the Register Transfer Level (RTL) first. An RTL that has been locked is subsequently converted to a gate level netlist within given timing constraints. In the next physical design phase a watermark generator which is a constraint-based watermarking generator inserts a special digital signature by using routes and timing adjustments. The resultant product is a secured GDSII layout that has obfuscation and watermark protection. During synthesis and placement phases, structural sensitivity and timing slack information is analysed to come up with security co-optimization. It is in this way ensured that the key-gate insertion and watermark properties do not interfere with timing closure or result in over-routing congestion. The coherent flow is compatible with the traditional EDA toolchains, so that it can be deployed practically without substantial alterations of the tool.

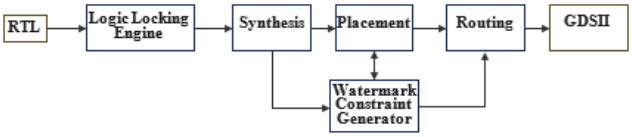

Fig. 1: Proposed Unified Security-Aware SoC Design Flow

Proactive Layer: SAT-Resilient Logic Locking

The proactive protection system works by the use of an RTL-presented SAT-resilient logic locking scheme. Structural concept Figure 2 is a diagram of the proposed strategy of key-gate insertion: original logic cones are transformed with an extra XOR/XNOR key-gate insertion controlled by a secret key vector. Mathematically, the functional behaviour of the defended circuit is expressed as:

$$Y=f(X,K) \quad (1)$$

X being the major input vector, K being the secret key vector of length k and Y being the output response. Only by matching the applied key with the embedded secret key is the proper functionality generated on the circuit. Any loss of the right key will lead to functional corruption.

Key-gate insertion is not done by chance. In its place, measures of structural sensitivity are calculated to calculate candidate sites of insertion. These measurement tools are controllability, observability, probability distribution of signals and path criticality. Gates that have a high influence on output propagation are deliberately altered in order to produce maximum output corruption under erroneous key conditions without unduly affecting vital timing paths.

Locking strength is determined by averaging the Hamming Distance between the output and correct and incorrect results;

$$HD_{avg} = \frac{1}{N}\sum_{i=1}^{N}\left|Y_{correct}^{(i)} \oplus Y_{wrong}^{(i)}\right|. \quad (1)$$

and N is the number of input vectors under consideration. The design goal will be to attain:

$$HD_{avg} \approx 50\% \quad (3)$$

Hamming Distance of 50 implies that the outputs during wrong key usage are statistically unrelated to its correct use, and thus, limits information leakage and approximation attacks. In addition, the interdependence between key structures is presented in more than two logic cones to become more complex to the structure and lower the performance of differentiating input creation in SAT-based assaults. The rate of distinction between the correct and incorrect key hypotheses is reduced and it becomes easier to extract key by increasing the computational complexity of key extraction.

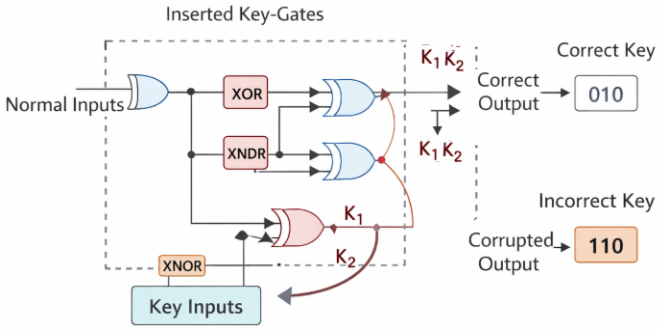

Fig.2: Logic-Level Schematic of the Proposed SAT-Resilient Key-Gate Insertion Mechanism

Reactive Layer: Constraint-Based Watermarking

Besides functional obfuscation, reactive ownership verification mechanism is also incorporated at physical that time. The physical-level embedding approach used in this paper is shown in Figure 3 with the watermark bits embedded in routing layers in timing slack areas and retrieved in the final layout. The layout is programmed with a special 64-bit digital signature when encouraging it in the place-and-route stage through constraint-based watermarking. As compared to the HDL-level watermarking which can be removed by resynthesis, physical-level embedding performs at a persistent level of across optimization pass. The encoder of the watermark commences by encoding every bit of the signature to a randomised routing setup. Adequate slack is found so that timing-safe regions are established in order to forestall performance degradation. In every watermark bit, routing topology rules in the form of predefined patterns of metal-layer usage, via preference routing, or controlled detours in the non-essential interconnect. These limitations are imposed when routing in details to create a distinct and repeatable physical pattern.

The watermark is not extracted using a layout scan-chain, but rather the post fabrication scan-chain. It is reconstructed to give the signature:

$$S = Extract(GDSII) \quad (4)$$

The probability of a false ownership claim is bounded by:

$$P_{false} = 2^{-64} \quad (5)$$

otherwise, uniformity of distribution of the 64-bit signature space is assumed. The tests used to gauge robustness include logic resynthesis, incremental optimization and routing rerouting. This is because the watermark is injected in timing constrained routing structures and any effort to eliminate this without being aware of the encoding scheme results in the loss of timing errors or delivery incompatibilities hence maintaining ownership support.

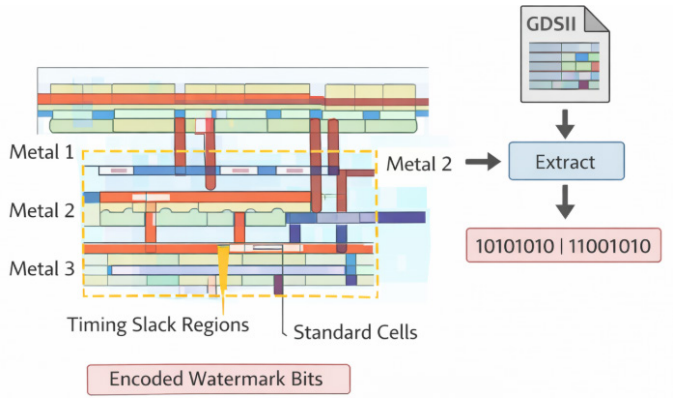

Fig.3: Physical Layout Illustration of Routing-Level Watermark Embedding and Extraction

Security-PPA Co-Optimization Strategy

The major issue in the integration of obfuscation and watermarking is the management of PPA overhead. In order to deal with this, co-optimization strategy is adopted in the process of synthesis and physical design. Timing analysis is also carried out during key-gate insertion to rule out gates, which reside in critical paths, to limit the number of delay penalties. Likewise, the watermark embedding is done on areas that have slack to ensure the non-degradation of frequencies. A timing-closure loop is used on an iterative basis where the synthesis and placement outcome is analysed and security constraints are tightened as a result. The predetermined quantity of key-gates inserted reduces the area overhead. Impact of power is managed by eliminating redundant switching in logic wads of high switching frequency. With this concerted optimization procedure, the combined structure

ensures that the total area overhead is reduced to less than 5 percent and delay increment is within manageable design thresholds and it becomes much harder to resist reverse engineering, SAT-based key recovery, and watermark removal attacks.

Experimental Setup and Security Evaluation

In order to justify the effectiveness of the proposed unified hardware protection framework, the functional security resilience and implementation overhead were analysed in a realistic VLSI design environment. Experiments performed were based on 45nm CMOS standard cell library at a clock rate of 500MHz. The design, which was being synthesised using an instance of an industry-standard RTL synthesis engine, physical implementation occurred using a constraint-aware place-and-route (P&R) engine. Representative ISCAS c432 and c1355 combinational designs were also benchmarked to test how scalable the tools were to useful SoC subsystems and a cryptographic core (for AES-128) was benchmarked. The analysis framework took three major security aspects into account, including resilience to extraction of keys using SAT, persistence to structural assault by AI and watermark holding as the flow of watermark optimization.

SAT Attack Resistance Evaluation

To test resilience to attacks based on Boolean Satisfiability (SAT), locked netlists were tested with a SAT-based framework of key-recovery. The attack presupposes access to oracles of a working chip and complete access to the locked netlist in the form of the gate-level. The time taken to extract key was timed where there was a varying key length k. As the size of keys grows, experimental results reveal that solver runtime grows exponentially, which is a better resistance than the traditional locking of XOR based on randomness. The trend of scalability is given in Figure 4 where the SAT runtime is shown to be growing exponentially with the key length that is increasing 32-128 bits. The efficiency of distinguishing the input generation is minimised by the introduction of structural interdependencies among several logic cones, and thus more SAT iterations take place at the convergence points. Since the placement of key-gate switches is determined by structural sensitivity and not random insertion, the measured obfuscation structure has a lower susceptibility of obfuscation to iterative pruning techniques by current SAT solvers. The trend towards complexity of the attacks has confirmed that the length of key and the interdependency of structure play a major role in increasing the level of required computational effort in key recovery success.

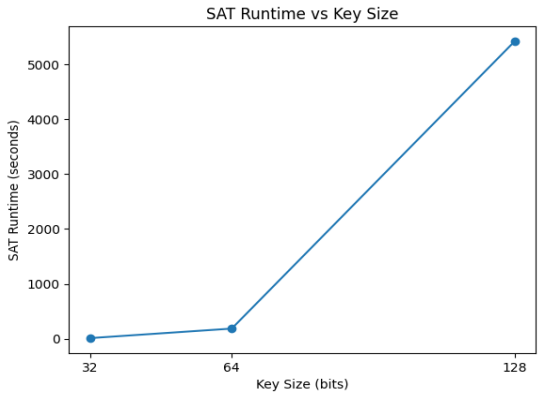

Fig.4: SAT Solver Runtime as a Function of Key Size Demonstrating Exponential Attack Complexity

AI-Based Structural Attack Resistance

Non-SAT attacks were also assessed in which structural vulnerability to machine-learning-based attacks was considered. In particular, the entropy-based structural analysis was used to evaluate the randomness that was introduced by insertions of key-gates. The measures of structural entropy of the locked netlist were calculated according to the connectivity distribution of the gates and variability of signal propagation. Findings show that baseline unlocked designs have higher structural entropy than baseline locked designs, which is to say that the position of key gates is less predictable. This increase in entropy is a limiting factor of the performance of the Graph Neural Network (GNN)-based key-gate classification methods, which are based on structural regularities. The dispersed pattern of interdependent key structures between various logic cones introduced by the proposed mechanism derails personalised structural signatures that are frequently used by AI-based assaults.

Watermark Robustness Evaluation

The robustness of UK 64-bit watermark was tested in numerous design alteration conditions on the physically embedded signature. These were logic resynthesis, incremental optimization pass and complete P&R re-execution with changed routing constraints. Extraction of the watermark of the resulting GDSII layout was carried out by an applicable extraction function Equation 4. The watermark has been preserved in all the consideration scenarios. Since the bits in watermark were integrated into routing areas that had timing constraints, any effort to delete or modify the routing topology of the encoding scheme to encode this data without understanding the encoding algorithm led to a timing infringement or operational anomaly. However, the likelihood of wrongly assigning property of a unique signature is limited by Equation 5, where there is a uniform distribution of the

64-bit signature space. The findings prove the assertion that the physical watermark remains resilient to the process of common optimization flows and resynthesis to provide credible ownership traceability.

Results and Discussion

PPA Overhead Analysis

The overhead added by the adopted integrated obfuscation and watermarking system in implementation was measured on representative benchmark circuits and one cryptographic core. The effect on the gate count, area, dynamic power and critical path delay is summarised in Table 1.

The findings show that the total area overhead is less than 5 percent in all the assessed designs including the much larger AES-128. The power increase is low and under 2 percent in any of the cases whereas delay degradation is small and it is within suitable time margins of a 500 MHz operating frequency. Comparative measurements of area overhead of standalone locking, standalone watermarking and the proposed integrated framework are depicted in Figure 5. The co-optimised integration, as demonstrated, is much more cost-effective than the independent protection mechanisms in the aspect of redundant hardware insertion. The proposed framework is more efficient in comparison with previous methods that use SAT-resilience locking like SFLL-HD that has recorded area overheads of between 8 and 15 percent in real-world applications. This low overhead can be explained by sensitivity-guided placement of the key-gate and use of slack watermark embedded, to ensure the process does not introduce any content into timing-sensitive areas. Such results verify that integrated protection can indeed be prohibitively PPA-penalised when it is optimally co-optimised.

Output Corruption and Functional Security

Output corruption on wrong key application was used to assess the strength of functional protection by scoring on average Hamming Distance which was as given in Equation(2). The observed value was:

$$HD_{avg} = 49.6\% \quad (6)$$

Figure 6 provides a distribution of Hamming Distance using benchmark circuits with all the evaluated designs showing 50% of the corruption as the theoretical value. This value is quite close to what would be expected of statistically uncorrelated outputs when erroneous keys are used. The near-random distribution of output is an important step to reducing information that can be exploited to leakage and also to restrict the usefulness of approximation and pruning-based attacks. The sensitivity-driven insertion strategy portrays a homogenous corruption of many output cones as opposed

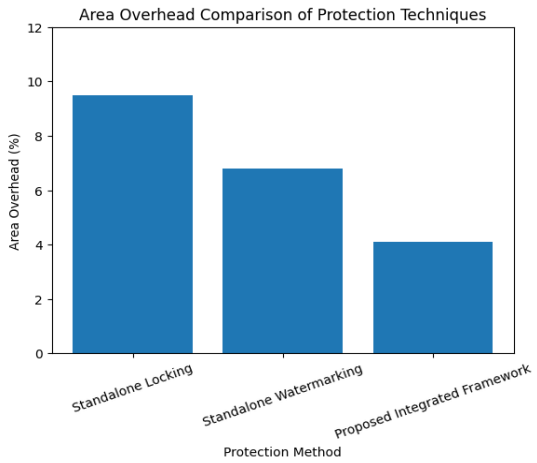

Fig. 5: Area Overhead Comparison Between Standalone Protection Techniques and the Proposed Integrated Framework

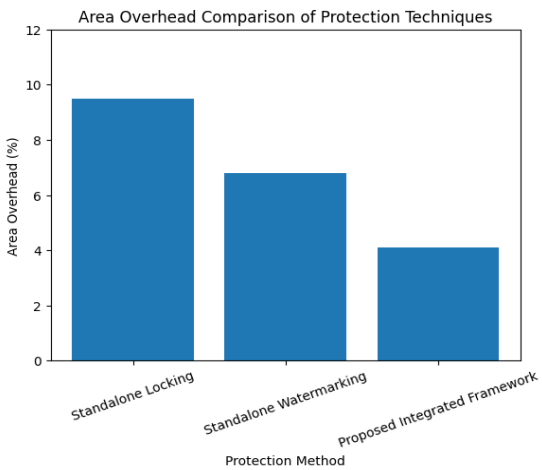

Fig.6: Average Hamming Distance (HD_avg) Across Benchmark Circuits Under Incorrect Key Conditions

Table 1: PPA Overhead of the Proposed Unified Security Framework

Benchmark	Gate Count	Area Overhead (%)	Power Increase (%)	Delay Increase (ns)
c432	160	3.2	1.1	0.04
c1355	546	2.8	0.9	0.02
AES-128	14,200	4.1	1.8	0.12

to conventional random XOR-based locking wherein the selection among gates is suboptimal thereby resulting in biassed corruption. This consistency gives it strong resistance to brute-force key-search attacks as well as structural inference attacks.

Watermark Extraction and Robustness

The strength of the 64-bit routing level watermark was tested by a variety of post-implementation situations of modifications. The accuracy of watermark extraction was 100 percent in all cases of tests, which proves that it is reliable to recover watermark form the final GDSII layout. The watermark removal success rate was less than 5% using removal techniques based on logic resynthesis, incremental optimization passes and complete place-and-route re-execution. In other scenarios where routing constraints have been changed without being informed of the encoding scheme, timing violations could be noted suggesting the tight association between watermark embedding and timing-sensitive routing structures. The chance of a false attribution of ownership is still limited to 2-64, which makes the event of the accidental duplication of signatures statistically insignificant. The physical constraint-based method exhibits a much greater persistence and forensic integrity as compared to HDL-level watermarking methods, which had been reported in previous research, and were extremely susceptible to removal-by-resynthesis attacks.

Comparative Analysis and Discussion

The integrated framework is seen to have two main benefits when compared to standalone logic locking strategies. To start with the PPA overhead is minimised by using synchronised additions of key-gates and watermark constraints which do not cause unnecessary revisions. Second, layered defence leads to security resilience: logic locking could have been partially compromised, however, ownership verification would be maintained with physical watermarking. When compared to the independent implementations of locking and watermarking that have been conducted in the past, the integrated method offers a trade-off that balances the implementation cost and security strength. The experimental findings prove that the layered architecture adds complexity to the attacks and at the same time maintains performance limits needed in the deployment of SoCs in practise. On the whole, the results show that the integration of the proactive obfuscation and the reactive watermarking, as a part of the single RTL-to-GDSII flow, can present substantial security improvements without unreasonable implementation costs. This justifies the practicability of the suggested methodology on actual semiconductor supply chains, specifically, on distributed manufacturing as well as third-party IP integration settings.

Conclusion

The paper has demonstrated an integrated hardware security structure which incorporates SAT-resilient logic locking in conjunction with constraint-based watermarking into a conventional RTL-to-GDSII SoC system design flow. In contrast to more traditional methods of obfuscation and watermarking, the presented methodology proposes a security-aware co-optimization approach in terms of which minimising unnecessary overhead and enhancing the trustworthiness of the golden protection are the main priorities. The framework supports the physical blocking of places by employing the key-dependent logic on-the-fly at the RTL level and physically entrenching the 64-bit watermark on the output of place-and-route, which provides simultaneously the proactive functionality protection and reactive traceability of ownership. Experimental testing of the built-in structure with ISCAS85 test circuits and an AES-128 cryptographic core modelled in 45 nm CMOS technology shows that the integrated structure incurs less than 5 percent Power, Performance and Area overheads in addition to nearly optimal 50 percent corruption of output when using the wrong key. Simulations of SAT attacks show that key-recovery update time grows exponentially with key size, the performance of SAT attacks based on key-recovery is more than that based on simple one-way verification of the key, and its resistance to attacks on Boolean satisfiability. In addition, watermark embedding at routing level is highly resistant to resynthesis, optimization, and making layout re-execution, which guarantees effective post fabrication ownership verification. These findings substantiate the evaluation of the hypothesis that layered hardware protection can dramatically increase attack complexity without the prohibitive implementation costs such that the proposed solution is feasible to the current distributed semiconductor supply chains and hardware-based ecosystems of third-party IP integration market offerings. The next steps of the research will be to incorporate Physically Unclonable Functions (PUFs) so that chip-specific dynamic key generation can be achieved, improving overproduction and cloning resistance. Other extensions can be AI-adaptive locking, split-manufacturing-sensitive watermarking, and blockchain-based support IP traceability infrastructure in safe multi-party SoC design setting.

References

1. Abdel-Hamid, A. T., Tahar, S., & Aboulhamid, E. M. (2005). A survey on IP watermarking techniques. *Design Automation*

for Embedded Systems, 9(3), 211–227. https://doi.org/10.1007/S10617-005-1395-X

2. Chang, C.-H., & Cui, A. (2010). Synthesis-for-testability watermarking for field authentication of VLSI intellectual property. *IEEE Transactions on Circuits and Systems I: Regular Papers, 57*(7), 1618–1630. https://doi.org/10.1109/TCSI.2009.2035415
3. Karmakar, R., Suman, S. J., & Chattopadhyay, S. (2020). A cellular automata guided finite-state-machine watermarking strategy for IP protection of sequential circuits. *IEEE Transactions on Emerging Topics in Computing, 10*(2), 806–823.
4. Katkoori, S., & Sheikh, A. I. (2022). *Behavioral synthesis for hardware security*. Springer.
5. Kirovski, D., Hwang, Y.-Y., Potkonjak, M., & Cong, J. (2006). Protecting combinational logic synthesis solutions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25*(12), 2687–2696. https://doi.org/10.1109/TCAD.2006.882490
6. Lewandowski, M., Meana, R., Morrison, M., & Srinivas, K. (2012). A novel method for watermarking sequential circuits. In *Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust* (pp. 21–24). https://doi.org/10.1109/HST.2012.6224313
7. Oliveira, A. L. (2001). Techniques for the creation of digital watermarks in sequential circuit designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20*(9), 1101–1117. https://doi.org/10.1109/43.945306
8. Rathor, M., Anshul, A., Bharath, K., Chaurasia, R., & Sengupta, A. (2023). Quadruple phase watermarking during high level synthesis for securing reusable hardware intellectual property cores. *Computers & Electrical Engineering, 105*, 108476. https://doi.org/10.1016/j.compeleceng.2022.108476
9. Sengupta, A., Bhadauria, S., & Mohanty, S. P. (2016). Embedding low cost optimal watermark during high level synthesis for reusable IP core protection. In *Proceedings of the IEEE International Symposium on Circuits and Systems* (pp. 974–977). https://doi.org/10.1109/ISCAS.2016.7527405
10. Sengupta, A., Roy, D., & Mohanty, S. P. (2018). Triple-phase watermarking for reusable IP core protection during architecture synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37*(4), 742–755. https://doi.org/10.1109/TCAD.2017.2729341
11. Torunoglu, I., & Charbon, E. (2000). Watermarking-based copyright protection of sequential functions. *IEEE Journal of Solid-State Circuits, 35*(3), 434–440. https://doi.org/10.1109/4.826826
12. Zhuang, X., Zhang, T., Lee, H. H. S., & Pande, S. (2004). Hardware assisted control flow obfuscation for embedded processors. In *Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems* (pp. 292–302). https://doi.org/10.1145/1023833.1023873