Journal of Hardware and Systems Security
https://doi.org/10.1007/s41635-025-00158-w

Mitigating Electro-Optical Frequency Mapping Attacks on Logic-Locked Integrated Circuits

Thomas Wojtal[1] · **Robi Paul**[1] · **Michael Zuzak**[1]

Received: 1 July 2024 / Accepted: 17 January 2025

Abstract
The outsourcing of integrated circuit (IC) fabrication raises concerns of reverse-engineering, piracy, and overproduction of high-value intellectual property (IP). Logic locking was developed to address this by adding logic gates to a design to a chip's functionality during fabrication. However, recent advances have revealed that logic locking is susceptible to physical probing attacks, such as electro-optical frequency mapping (EOFM). In this work, we propose *Adjoining Gates*, a novel logic locking enhancement that places auxiliary logic gates near gates that leak key information when probed to obscure them, thereby mitigating EOFM-style attacks. To implement Adjoining Gates, we developed an open-source security verification and design automation algorithm that detects EOFM key leakage during placement and inserts Adjoining Gates in a design. Our evaluation shows that our proposed approach identified and mitigated all EOFM-extractable key leakage across 16 benchmarks of varying sizes, locking techniques, and probe resolutions with a 4.15% average gate count overhead.

Keywords Adjoining Gates · Logic locking · Electro-optical frequency mapping · Untrusted foundry

1 Introduction

The increasing complexity and cost of semiconductor manufacturing have led design houses to outsource integrated circuit (IC) fabrication. By outsourcing the fabrication process, design houses assume security risks including the potential for overproduction, reverse-engineering, and malicious modification of their design [1]. To mitigate these risks, logic locking was developed [2]. Logic locking is a combinational circuit obfuscation technique that inserts additional logic gates into a design to "lock" ICs prior to end-user shipment. These additional gates are driven by a new set of primary inputs, known as *key inputs*, that must have a secret value applied to enable a locked module to exhibit the intended functionality. The design house then withholds the secret key from untrusted fabrication partners, hiding the intended functionality of the device and protecting internal IP. After fabrication, the ICs are returned to the design house for *activation*, where the designer applies the secret key, allowing the device to function as intended [2].

Prior work has shown that electro-optical probes, which are used in IC failure analysis [3–5], can be used to infer the key of a logic-locked IC [6–9]. These attacks illuminate the backside of an IC with an electro-optical probing laser and measure the reflected power [5]. The reflected power is influenced by the number of free carriers in the illuminated silicon, which is correlated to the voltages applied to the transistors in the beam. By carefully selecting transistors whose sensitization is influenced by key inputs, details of the device state, such as the locking key, can be inferred. In this work, we consider electro-optical frequency mapping (EOFM) attacks in particular [3, 4]. EOFM attacks aggregate a series of probe measurements and perform frequency analysis to increase probe resolution. This allows fewer gates to be imaged simultaneously, limiting the interference/noise produced by nearby gates that act to obscure the measurement. Prior work has shown that EOFM-style attacks can infer the key of a locked IC, thereby rendering logic locking ineffective.

In this work, we develop a methodology to resist EOFM attacks against logic locking techniques during placement. A major aspect of EOFM attacks we focus on to achieve

✉ Michael Zuzak
mjzeec@rit.edu

Thomas Wojtal
tsw4235@rit.edu

Robi Paul
rp7248@rit.edu

[1] Rochester Institute of Technology, Rochester, NY, USA

Published online: 29 January 2025

this goal is the physical limitation of probing resolution [3–5]. Given a sufficiently small process technology, resolution limitations physically limit the ability of an electro-optical probing device to scan individual gates in an IC independently [4, 5, 8, 10]. We utilize this limitation to our advantage as the foundation of our logic locking enhancement. Our goal is to identify and mitigate EOFM-based attacks that attempt to recover the locking key by placing noisy gates near leaking gates that cannot be resolved separately, thereby obscuring the leakage. Specifically, we develop a security verification and design automation algorithm to detect regions of an IC that will leak the logic locking key when subjected to an EOFM-style attack and then automatically incorporate extra logic to remediate any leakage.

1.1 Contributions

In this work, we propose Adjoining Gates, a logic locking enhancement to mitigate EOFM attacks on logic locking. The fundamental idea of an Adjoining Gate is to place an additional noisy gate near leaking gates that cannot be resolved separately, thereby obscuring the leakage. The contributions of this work are summarized as follows:

1. We develop Adjoining Gates, a novel countermeasure against EOFM attacks on logic-locked ICs. Adjoining Gates serve as a logic locking enhancement that can be integrated alongside any conventional locking.
2. We develop an open-source security verification and design automation algorithm that detects areas of potential key leakage and automatically inserts Adjoining Gates. The code can be found in https://github.com/twojtal/Adjoining-Gate/.
3. We develop an overhead optimization to reduce the input fan-in of Adjoining Gates, thereby reducing the power, area, and delay overhead of the technique.
4. We evaluate Adjoining Gates across 16 benchmark circuits of varied size, locking technique, and EOFM resolution. Our open-source security verification and design automation algorithms identified and mitigated all EOFM-extractable key leakage across all benchmarks with a 4.15% average gate count overhead.

2 Preliminaries

2.1 Logic Locking

Logic locking obfuscates a chip's IP during fabrication by adding auxiliary combinational logic to a design that is driven both by internal logic signals as well as additional primary inputs added to a design, known as key inputs [2]. As a result, a locked circuit only exhibits its intended functionality when a correct value is applied to these added primary inputs, known as the secret key. By doing so, this key can be withheld from untrusted fabrication partners, hiding the intended functionality of the design. This helps protect design IP from reverse-engineering, overproduction, and theft [1]. After fabrication and testing, the design house *activates* the locked circuit by inserting the secret key into a tamper-proof memory, unlocking the intended functionality of the design for end-use. Fundamentally, the goal of locking is to make it infeasible for an untrusted foundry to deduce IC functionality or the secret key. Prominent locking techniques include Strong Logic Locking (SLL) [12], Anti-SAT [13], Full-Lock [14], Stripped Functionality Logic Locking (SFLL) [15], and others [2].

2.2 Electro-Optical Probing and Frequency Mapping Attacks

Electro-optical probing (EOP) was developed for IC failure analysis. EOP involves applying an electro-optical laser, known as a *probe*, to illuminate a small region of an IC through the backside of the die and measure the reflected power. The reflected power is proportional to the number of free carriers contained in the illuminated substrate, which is influenced by the voltages applied to the device [3–5]. Hence, by measuring the reflected power, it is possible to infer information regarding the sensitization of logic gates in the device. Abnormal logic sensitization indicates a failure in the IC, facilitating the localization of manufacturing defects during a test. Prior work has shown that EOP can be used to infer the key of a logic-locked IC by probing key-dependent gates in a locked circuit [6, 8]. By sensitizing key-dependent gates to exhibit different logical states based on the key value, EOP can infer the key of a locked circuit [6–9].

To ensure a non-destructive measurement, the wavelength of the probing laser cannot exceed the bandgap of silicon. This requirement fundamentally limits the resolution of EOP to between 220 and 775 nm [4, 5]. Reducing the wavelength below this point, thereby increasing the resolution, will result in a destructive measurement where the voltage present in the substrate is influenced through the application of the probing laser. This fundamentally limits the resolution of electro-optical probes when applied to silicon ICs, ensuring that as technology nodes continue to shrink, more gates must be probed simultaneously. Electro-optical frequency mapping (EOFM) synthetically improves this resolution by performing multiple local measurements at a fixed interval [4, 5]. A large set of these measurements in varying device states is then converted to the frequency domain and band-pass filtered at the measurement frequency. By doing so, the switching activity of logic in a design can be isolated [4]. Prior work has shown that EOFM can be used to improve the

resolution of probing attacks on logic locking as well [9]. For example, the CLAP attack proposed a methodology to automatically identify a sequence of inputs that, when applied to the circuit, produce a different sensitization in the illuminated transistors based on the value of the key inputs fanning into the probed node (i.e., the set of illuminated gates in the circuit). This results in different frequency behavior based on the key value, allowing EOFM to infer the key.

2.3 Related Work

Prior works have proposed a variety of countermeasures to EOP attacks [7, 8, 16–18]. We separate prior countermeasures into three families and consider the limitations of each to motivate this work. The first countermeasure family consists of light-scattering coatings to diffuse reflected light. Doing so reduces the intensity of light emissions from the device, making it more challenging for attackers to extract key leakage [7, 17]. While these approaches are effective, they necessitate non-standard manufacturing processes, which may impact yield or increase the cost of fabrication. The second countermeasure family consists of electro-optical sensor circuits that detect the probing laser and disable or obfuscate the circuit [16, 18]. These sensors must be located near candidate probe points. However, as shown in [9], there may be hundreds or thousands of points in the circuit that could be probed to extract key leakage. Hence, a design may require a prohibitively large number of sensors.

Finally, we consider Concealing Gates, a logical mitigation strategy that most closely relates to our proposed approach [8]. Concealing Gates are inverters placed in close proximity to key gates in the circuit to mitigate side-channel leakage. However, the proposed approach protects only key gates (i.e., gates driven directly by a key input) and relies on the assumption that the secret key applied to the circuit is tied to the reset signal. While this approach was shown to be effective against considered attacks, prior work, such as [9], has proposed attacks that do not target key gates or require the use of a reset signal to extract key leakage. As such, Concealing Gates are insufficient to protect against such attacks. **In this work, we address this gap in prior art by proposing a countermeasure for EOFM attacks that do not extract leakage directly from key gates or rely on reset toggling, such as** [9].

2.4 Threat Model

We consider an adversary that aims to infer the key of a logic-locked IC. The adversary can take any strategy using (1) a locked netlist, which can be obtained via reverse-engineering the GDSII files provided for fabrication [1]; (2) a black-box oracle IC with the key applied, which can be obtained through the open market or test facilities; and (3) an EOFM probe station, which can be rented by the hour for a modest fee [6]. For this work, we consider only an EOFM-based attacker. Other probing schemes, such as electron beam probing, and fault injection attacks, such as [19, 20], are considered out-of-scope. This mirrors the attacker considered in prior work [6–9].

3 Adjoining Gates

3.1 Theoretical Foundation of Adjoining Gates

Prior work has shown that the logical and physical configuration of a circuit determines where EOFM leakage can be extracted from a design [9]. To do so, an electro-optical laser illuminates a small set of sensitized logic gates, referred to as a *node*, through the backside of the die and measures the reflected power (see Section 2.2 for more details). We consider the set of one or more gates that are simultaneously illuminated by the electro-optical probing laser (i.e., probed) to be a *target node* in the circuit. In this work, we consider the resolution of the probe (i.e., the number of illuminated gates) to be variable. Depending on the technology node as well as the resolution limitations of the probe itself, an attacker generally must probe multiple gates at once [4, 5]. The fan-in cone of a target node can contain many primary inputs (PI), some of which may be key inputs. Nodes that leak key information are either connected to key inputs or dependent on key inputs from at least one path in a node's fan-in [9]. Given a specific input, a node can be made controllable by (i.e., sensitive to) key inputs if different logical behavior is exhibited when different key values are applied to those inputs. In this scenario, EOFM can infer the key value by probing this target node. Prior work proved that a target node leaks key information if Eq. 1 is satisfied [9].

$$(N(I1, K1) = N(I2, K1)) \bigwedge (N(I1, K2) \neq N(I2, K2)) \quad (1)$$

In Eq. 1, N represents the logical function for the fan-in cone to the target node, $I1$ and $I2$ are PI vectors containing the values that sensitize the target node during cycles 1 and 2, and $K1$ and $K2$ are key input vectors containing the values that sensitize the target node during cycles 1 and 2. An EOFM attack aims to infer the subkey value by probing the target node in a correctly keyed oracle circuit. Equation 1 is satisfied if a set of two inputs can be identified that (1) produce the same reflected power (i.e., same voltages or logical state of the target node) for both inputs ($I1$ and $I2$) if the key value in the oracle is $K1$ and (2) produce a different reflected power (i.e., different voltages or logical state of the target node) for both inputs ($I1$ and $I2$) if the key value

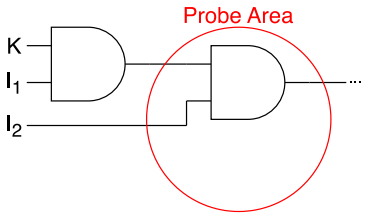

Fig. 1 Example of an EOFM attack extracting the locking key from a target node

Key (K)	I1	I2	Probe State
0	f	0	0
0	f	1	1
1	f	0	f
1	f	1	f

in the oracle is $K2$. In this case, when the oracle is probed, it can be observed whether the same or different reflected power is observed when the inputs $I1$ and $I2$ are applied to the circuit. This is observed as the presence or absence of a frequency component in the EOFM measurement of the target node. Because $K1$ and $K2$ produce different EOFM measurements, probing the oracle allows either $K1$ or $K2$ to be eliminated from consideration. Any target node in the circuit where values for $I1$, $I2$, $K1$, and $K2$ exist that satisfy Eq. 1 can be probed to infer information regarding the correct secret key. We refer to this as *key leakage*.

Figure 1 contains a trivial example of an EOFM attack on a locked circuit. In this scenario, the subkey value K can be determined given the correct stimuli on inputs $I1$ and $I2$. The table in Fig. 1 demonstrates the truth table for this case. K is presented as the hypothetical subkey value applied to the leftmost AND gate, $I1$ and $I2$ are the inputs, and the probe measurement is represented by the absence or presence of a frequency component in the reflected power measurement. For inputs, the value f represents a toggling signal between logic "0" and "1" at a fixed frequency. For the probe state, the value f indicates that the sensitization of the target node changes with the frequency that the inputs are toggled. This can be observed as the presence of a frequency component in the reflected power measured by EOFM.

As shown by the rows marked in red, the probe measurement varies based on the key value applied to the target node. This discrepancy allows an attacker to infer the key value by simply applying a switching frequency on input 1 and maintaining input 2 constant at either logic "0" or "1." In this example, the lack of a frequency component in the reflected power measured by EOFM would indicate a key value of "0." Conversely, the presence of a frequency component in the reflected power matching the input frequency f would indicate a key value of "1." This example shows a scenario where key leakage occurs, as defined by Eq. 1.

3.2 Overview of Adjoining Gates

The core of Adjoining Gates is the addition of logic gates that obscure the reflected power measurement used to infer the key by EOFM attacks on logic locking. These relocated or added gates, which we call Adjoining Gates, are to produce a probe measurement at any target node that matches the switching frequency applied to the inputs in its fan-in. In terms of an EOFM attack, the node would appear to exhibit the same probe measurement, regardless of the key value applied. This effectively closes the EOFM side channel and prevents any key leakage from the node.

To achieve this, Adjoining Gates exploit the physical limitation of probing resolution (see Section 2.2). Given a sufficiently small process technology, resolution limitations prevent a probing device from observing individual gates in an IC independently [4, 5, 8, 10]. Adjoining Gates exploit this limitation by adding an additional logic gate near a leaking node. If these gates can be placed close enough together, the probe must measure them simultaneously.[1] This so-called Adjoining Gate is designed such that Eq. 1 can never be satisfied, thereby mitigating the EOFM key leakage in the device. Figure 2 depicts an example of the same circuit from Fig. 1 and a closely situated Adjoining Gate. By identifying and inserting Adjoining Gates alongside any target node with key leakage, EOFM-style attacks can be mitigated.

Consider the following example of an Adjoining Gate that mitigates key leakage in Fig. 2. The Adjoining Gate, in this case a NAND gate, has no key inputs in its fan-in cone (i.e., is non-key-dependent). Instead, this gate is dependent on the same primary inputs as the target node. As was done in our prior example, a frequency applied to $I1$ causes the target node to be sensitized differently based on the subkey value. Because the Adjoining Gate is located in close proximity to the target node, resolution limitations ensure that both gates must be probed together. Hence, any switching activity of the Adjoining Gate will also influence the EOFM measurement of the target node. This causes a frequency component to appear in the reflected power measured by EOFM regardless of the key value, K, applied to the circuit. The table on the right side of Fig. 2 contains the truth table for all input/key combinations and the resulting EOFM measurement.

Notice that the EOFM measurements in this example are identical regardless of the key value being "1" or "0." This indicates that no key leakage can be extracted by probing the target node. Hence, the use of an Adjoining Gate restricted the use of EOFM to extract key leakage from this target node, protecting the locking key. Given the ever-decreasing size of IC features and the physical limitations on electro-

[1] Concealing Gates have demonstrated that such physical proximity is achievable [8].

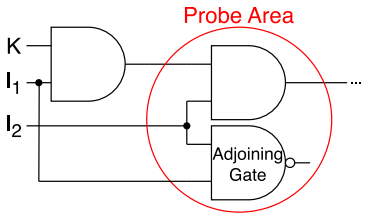

Fig. 2 Example of Adjoining Gate and corresponding EOFM attack behavior

Key (K)	I1	I2	Probe State
0	f	0	f
0	f	1	f
1	f	0	f
1	f	1	f

optical probe resolution [4, 5], this method of preventing EOFM attacks will remain effective as technology continues to shrink. By deploying these gates throughout the die in areas where EOFM key leakage exists, such attacks can be prevented.

3.3 Adjoining Gate Implementation

As defined in Section 3.1, a target node leaks key information when the PIs in its fan-in cone can cause the target node to produce different switching behavior for two possible key values. This behavior is formally defined by Eq. 1, derived in [9]. To ensure this never occurs, an auxiliary gate can be added in close proximity to a target node that produces switching activity whenever a PI in the fan-in cone of the target node changes, regardless of the key value applied to the circuit. We refer to this extra gate as an Adjoining Gate. By doing so, any change PIs that could be used to extract key leakage will necessarily produce a different sensitization in the Adjoining Gate. As a result, an EOFM measurement of the node will always contain a frequency component, preventing key values from being differentiated by obscuring any key leakage behind the switching of the Adjoining Gate. Note that this implementation of Adjoining Gates is designed to mitigate EOFM-based probing attacks on logic locking. Alternative probing strategies, such as electron beam probing that extract key leakage through different mechanisms, as well as fault injection attacks, such as [19, 20], are not mitigated by Adjoining Gates.

Specifically, we define an Adjoining Gate as a logic gate added alongside a target node that meets the following two criteria: (1) it is close enough to the target node so that both the Adjoining Gate and the target node must be probed (i.e., illuminated by the electro-optical probing laser) together, and 2) it is driven by all the PIs within the target node's fan-in cone. This obscures key leakage that may have otherwise been produced by the target node, ensuring that Eq. 1 will always be unsatisfiable. An example of an Adjoining Gate is shown in Fig. 3.

We note that the specific gate type used for Adjoining Gates does not impact their efficacy. Hence, choosing a small gate minimizes the overhead of integrating Adjoining Gates into a design. Additionally, while we leave the specific output termination used for Adjoining Gates to the designer, we consider the use of a small capacitive load as a baseline (e.g., the gate of a NMOS transistor). To prevent design automation tooling from optimizing away Adjoining Gates, attributes, such as "set_dont_touch" in Cadence Genus, can be used during synthesis.

3.4 Optimizations for Adjoining Gates

There is inherently design overhead (i.e., area, power, delay) associated with using Adjoining Gates to protect a circuit. In particular, PIs in the circuit must be routed to each Adjoining Gate, increasing routing pressure and PI fan-out in the circuit. In this section, we propose an optimization to limit the number of PIs that must fan into an Adjoining Gate to help alleviate design overhead.

Not all PIs in a target node's fan-in cone are necessary for Adjoining Gates to function properly. This occurs when the value for a specific PI is "don't-care" (i.e., the value does not matter) for any solution to Eq. 1. In this case, this PI does not contribute to the leakage at the target node. We propose the use of a SAT solver to determine the minimal subset of relevant PIs required to prevent leakage. Specifically, we propose a reductive approach. Initially, Adjoining Gates are added with all PIs that fan into the target node. Adjoining Gate inputs are then iteratively removed, and a SAT solver is used to solve Eq. 1 for the target node. If the removal of an input to the Adjoining Gate makes Eq. 1, indicating key leakage in the target node, then the PI cannot be removed from

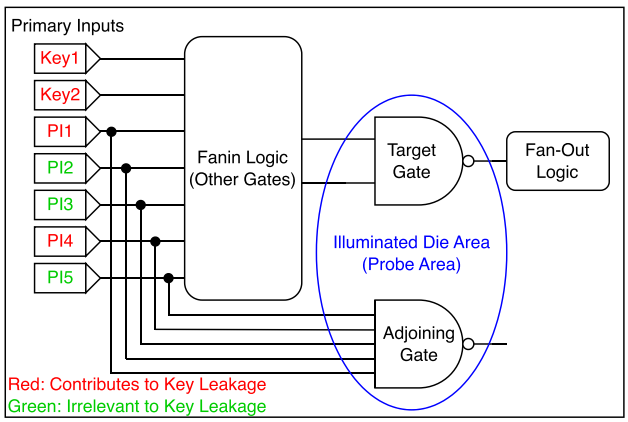

Fig. 3 Example of an Adjoining Gate

the Adjoining Gate and is added back. Conversely, if Eq. 1 remains unsatisfiable after removing the input, then the target node still does not contain key leakage. As a result, this PI can be removed from the Adjoining Gate without consequence. Figure 4 depicts the proposed PI reduction optimization.

Despite optimizing the number of inputs, Adjoining Gates may still require input signals to be routed to a high logical depth. This could lead to a long, high-capacitance interconnect being added to the design and increased routing pressure. To help alleviate this overhead, buffer chains can be inserted to reduce the capacitive load of long interconnects. Alternatively, nearby gates that are driven by required inputs (or intermediate signals) can be re-purposed to serve as an Adjoining Gate by re-locating them to be near the leaking node. This allows existing logic to serve as an Adjoining Gate, alleviating the need to route additional signals through the design.

3.5 Integrating Adjoining Gates into a Standard Design Flow

To implement Adjoining Gates, leaking nodes are identified within a circuit after initial placement. Adjoining Gates are then inserted during the iterative placement process within a standard design flow, similar to how decoupling capacitors are added to a design. Specifically, after the initial placement, nodes in the circuit are analyzed for leakage using Eq. 1. Adjoining Gates are then added and placed alongside any leaking nodes. Proximity constraints are applied to ensure that the design automation tools position the Adjoining Gates sufficiently close to the leaking nodes to obscure key leakage. This approach allows existing design automation tooling to be used to integrate Adjoining Gates into a design.

Adjoining Gates add logic to a design which may negatively impact other design factors, such as the power integrity, signal integrity, and thermal management. However, by integrating Adjoining Gate insertion into a conventional design flow, existing design automation tooling and optimization techniques can be used to minimize the potential impact of Adjoining Gates and ensure that design constraints are met.

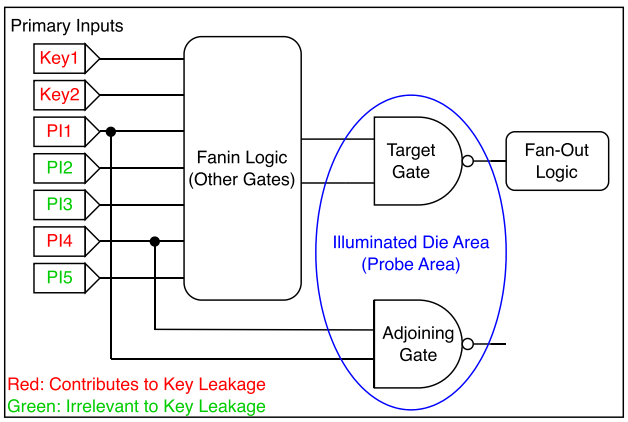

Fig. 4 Example Adjoining Gate with PI optimization

3.6 Integrating Adjoining Gates with Other Security Measures

Adjoining Gates are designed to mitigate EOFM leakage without altering a design's logical structure or the logic locking technique. This is achieved because Adjoining Gates are purely additive. They introduce auxiliary logic unrelated to the core design functionality. This makes Adjoining Gates both logic locking technique and locked module agnostic, requiring only proximity to leaking gates to operate effectively. This flexibility extends to other security measures incorporated solely into the logical function of the design (e.g., masking for side-channel mitigation). Hence, Adjoining Gates can be used with other security mechanisms without compromising their function.

4 Evaluation of Adjoining Gates

4.1 Open-Source Adjoining Gate Insertion Tool

To facilitate the evaluation of Adjoining Gates, we developed an open-source security verification and automation tool. This tool first performs a security verification of a circuit to identify locations where EOFM could be used to extract key leakage based on a pre-specified probe resolution. If desired, Adjoining Gates are then automatically added to the design to mitigate identified leakage. This tool is open-source and can be found at [11]. The tool contains two primary routines: *scanning* and *adding*. *Scanning* performs a security verification of the design whereby a circuit is evaluated for EOFM leakage. To do so, a graph representation of the circuit is traversed and Eq. 1 is solved at each possible target node to determine if any key leakage could be extracted via EOFM. Leaking target nodes are marked and returned to the user. *Adding* inserts Adjoining Gates at leaking target nodes in the circuit.

To evaluate Adjoining Gates, we applied our tool to 16 benchmark circuits that were developed to evaluate EOFM-

Table 1 Characteristics of benchmark circuits used for evaluation

Circuit	PIs	Gates	POs	LL Key Length			
				AntiSAT	Full-Lock	SFLL	SLL
b14	277	9767	299	404	540	277	256
c1908	33	880	25	78	384	33	88
c5315	178	2307	123	156	540	178	231
des	256	5104	245	368	540	256	256

style attacks [9]. The benchmarks consisted of 4 varying size benchmark circuits (c1908, c5315, des, and b14), locked with 4 locking techniques each (SLL [12], Anti-SAT [13], Full-Lock [14], and SFLL [15]). For reference, the characteristics and description of each benchmark circuit are in Table 1. Complete details of the open-source suite can be found in [9]. For our evaluation, a security verification was performed to identify leaking nodes, and Adjoining Gates were inserted in each circuit. The runtime, number of Adjoining Gates, and the number of PIs per Adjoining Gate were assessed over varying EOFM probe resolutions. We note that there are two scenarios where the effective resolution of the EOFM probe changes, causing more gates to be illuminated (i.e., probed) simultaneously: (1) the wavelength of the electro-optical laser used by the probe is reduced, and (2) a smaller technology node is used. The technology node size is our primary consideration for this work because it considers the efficacy of Adjoining Gates in the future as technology nodes continue to shrink, resulting in smaller logic gates and a coarser effective probe resolution. To simulate changes in probe resolution for our evaluation, we consider between 2 and 10 logic gates in the circuit being simultaneously illuminated by the probe.

4.2 Effectiveness of EOFM Attacks on Adjoining Gates

Let us consider the ability of Adjoining Gates to mitigate EOFM attacks against logic locking. To do so, we inserted Adjoining Gates into any leaking node of the benchmark circuits and assessed the protected circuit using Eq. 1 to determine if any target nodes exhibited leakage after Adjoining Gate insertion. We again emphasize Eq. 1 was proved to determine whether the logic locking key can be extracted from a target node using EOFM in [9]. Table 2 contains the results of this experiment, categorized by probe resolution. In all cases, the number of target nodes in the circuit that leaked after Adjoining Gate insertion was 0. This supports the capability of Adjoining Gates to mitigate EOFM key leakage across locking techniques, circuits, and probe resolutions. Additionally, the number of inserted Adjoining Gates, which is equivalent to the number of target nodes leaking in a circuit, was between 50 and 475 gates on average. This indicates that a relatively small number of Adjoining Gates are necessary to mitigate EOFM attacks against a locked circuit. Finally, we observe that the number of Adjoining Gates does not appear to strongly correlate to circuit size. This can be observed in Table 2, where the circuits with a higher number of visited nodes (e.g., des) require more Adjoining Gates than circuits with the most total nodes (e.g., b14). During Adjoining Gate insertion, nodes are only visited if they have key inputs in their fan-in (i.e., they potentially leak key information). Hence, the number of nodes visited is influenced by the circuit topology. In circuits where key inputs quickly diffuse into a large portion of the logic, more nodes are visited, resulting in more Adjoining Gates being added.

4.3 Impact of Probe Resolution on Adjoining Gates

The results in Table 2 indicate that when more gates are illuminated by the probe (i.e., a coarser effective resolution), the average number of leaking nodes is reduced. This also correlates to a decrease in the number of Adjoining Gates added to a circuit. Once again, we note that the number of inserted Adjoining Gates corresponds to the number of target nodes leaking before Adjoining Gate insertion. This is because an Adjoining Gate must be added to mitigate the leakage in the target node. As noted in Sect. 2.2, to ensure non-destructive measurements, the resolution of an electro-optical probe is limited to between 220 and 775 nm [4, 5]. Hence, these results indicate that Adjoining Gates will not only scale as technology size continues to decrease (i.e., a

Table 2 Effectiveness of Adjoining Gates at mitigating EOFM leakage in benchmark circuits across EOFM probe resolution

Res	b14		c1908		c5315		des		All
	$\overline{V}$	L	$\overline{V}$	L	$\overline{V}$	L	$\overline{V}$	L	L (w/ AG)
2	495.75	212.75	251.50	114.25	816.50	546.00	1339.25	994.00	0.00
3	425.75	148.75	202.50	79.75	637.25	385.00	1108.00	798.50	0.00
4	298.25	93.50	163.75	62.25	537.75	312.75	878.00	627.50	0.00
5	246.00	71.25	153.75	52.50	416.75	201.25	682.50	472.75	0.00
6	220.50	55.25	124.50	31.75	356.00	158.00	606.50	422.75	0.00
7	195.00	40.25	104.25	15.75	271.50	99.00	444.75	271.25	0.00
8	61.00	19.00	42.00	10.00	164.75	69.25	322.00	267.00	0.00
9	45.50	11.50	45.00	11.50	140.25	62.50	240.50	186.50	0.00
10	30.75	11.25	32.00	11.00	109.50	47.25	205.75	160.25	0.00

Results are averaged over all evaluated locking techniques
Abbreviations: *AG* Adjoining Gate, *Res* electro-optical probe resolution (in gates), *V* visited, *L* leaking, $L(w/AG)$ leaking with Adjoining Gates implemented

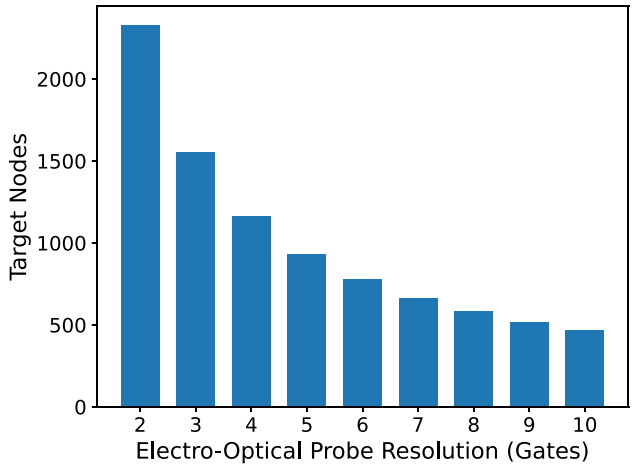

Fig. 5 Average target nodes present in benchmark circuits by probe resolution

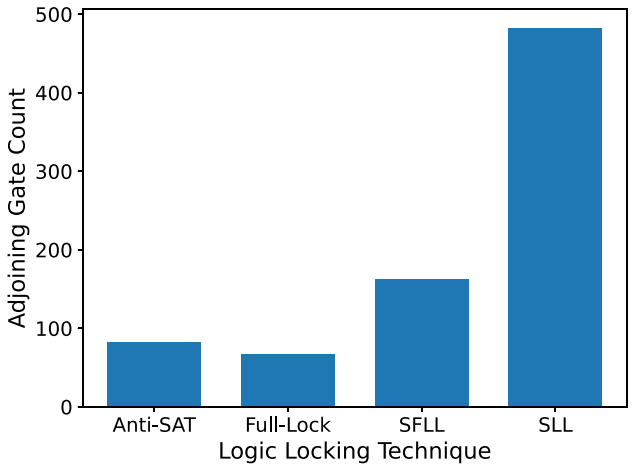

Fig. 7 Average number of Adjoining Gates added by logic locking technique

coarser effective resolution), but may become more effective, requiring fewer gates to mitigate EOFM-based leakage in the design.

The observed reduction in Adjoining Gates required to protect the circuit for coarser probe resolutions is caused by the corresponding reduction in target nodes present in the circuit. This is because a larger electro-optical probe beam must image multiple gates simultaneously, resulting in fewer overall target nodes that can be imaged. This can be seen in Fig. 5, which depicts the decrease in the total target nodes in the benchmark circuits as the resolution becomes coarser. The runtime of scanning (security verification) and adding Adjoining Gates to a circuit also followed a similar trend. As shown in Fig. 6, detecting key leakage and inserting Adjoining Gates decreased with coarser resolutions. Once again, this is caused by the decrease in the number of target nodes that must be analyzed for key leakage. However, because coarser resolutions require more logic gates to be considered simultaneously, a more complex SAT formulation must be solved to identify a solution to Eq. 1. This caused the slightly increased scan runtime for coarser probe resolutions. We emphasize that both the scan (security verification) and add (Adjoining Gate insertion) phase of the developed Adjoining Gate tool ran in under 30 s on average, regardless of resolution, supporting the efficiency of the proposed Adjoining Gate approach.

4.4 Effectiveness of Adjoining Gates Across Locking Techniques

In this section, we consider the ability of Adjoining Gates to enhance varied locking techniques and evaluate the effectiveness of Adjoining Gates when paired with each technique. To do so, we implemented Adjoining Gates alongside 4 locking

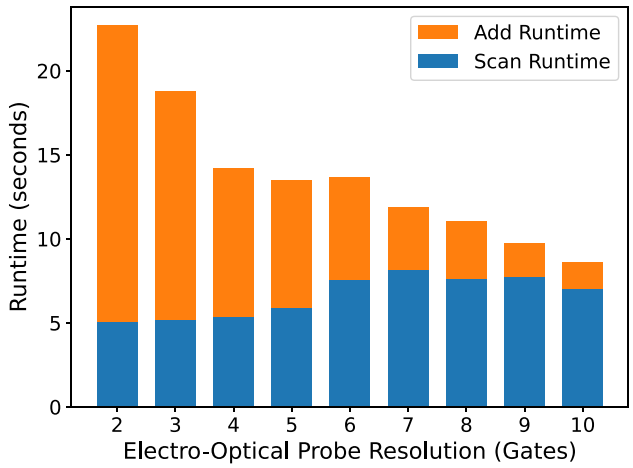

Fig. 6 Average runtime to scan and add Adjoining Gates to circuit by probe resolution

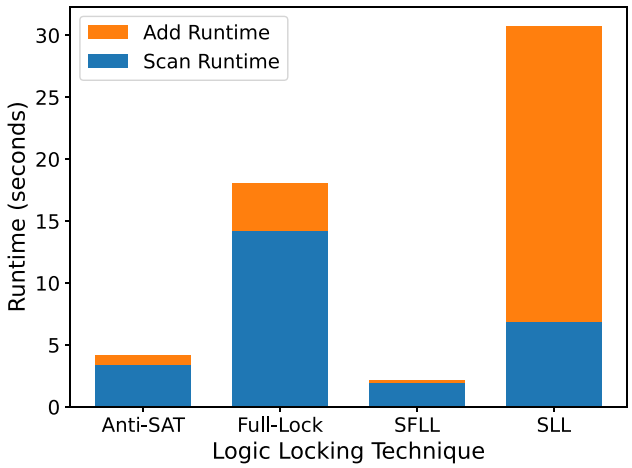

Fig. 8 Average runtime of automated Adjoining Gate insertion by logic locking technique

Table 3 Overhead in gate count for Adjoining Gates (AG) in each benchmark

Circuit	LL Tech.	AG Count	AG Overhead (%)	Circuit PI Count	Avg PIs per AG	Avg PI Overhead (%)
b14	AntiSAT	50.22	0.66%	681	1.47	0.22%
	Full-Lock	147.89	1.52%	817	1.76	0.22%
	SFLL	40.22	0.52%	554	2.09	0.38%
	SLL	56.56	0.76%	533	11.58	2.17%
c1908	AntiSAT	15.89	1.26%	111	1.76	1.59%
	Full-Lock	105.44	4.22%	417	2.04	0.49%
	SFLL	4.44	0.34%	66	4.00	6.06%
	SLL	47.00	3.37%	121	1.57	1.30%
c5315	AntiSAT	241.33	10.05%	334	1.51	0.45%
	Full-Lock	111.22	2.17%	718	1.65	0.23%
	SFLL	33.56	1.82%	356	1.79	0.50%
	SLL	449.89	15.12%	409	2.60	0.63%
des	AntiSAT	24.33	0.31%	624	1.41	0.23%
	Full-Lock	172.67	2.22%	796	8.93	1.12%
	SFLL	29.56	0.40%	512	2.02	0.39%
	SLL	1640.33	21.61%	512	5.68	1.11%

All benchmark circuits can be found at [11]

techniques (SLL [12], Anti-SAT [13], Full-Lock [14], and SFLL [15]). The average number of Adjoining Gates added by locking technique averaged over the 4 benchmark circuits is in Fig. 7. The corresponding *scanning* and *adding* runtime for each benchmark, averaged by locking technique, is in Fig. 8. Based on these results, we observe that Adjoining Gates were added to each benchmark circuit in under 30 s, regardless of locking technique. Moreover, in each case, Adjoining Gates fully mitigated all EOFM-based key leakage in the circuit. This experimentally supports the ability of Adjoining Gates to enhance logic locking techniques across a variety of constructions.

Based on Fig. 7, we note that the number of Adjoining Gates added to the circuit varies by locking technique. This variance by technique is unsurprising given that each technique modifies the circuit differently. This result appears to be caused by how distributed the locking technique is throughout the circuit. This is because target nodes without any key inputs in their fan-in cone cannot leak key information (i.e., Eq. 1 cannot be satisfied). Hence, SLL, where locking gates are distributed throughout the entire design, requires the highest number of Adjoining Gates. Conversely, the other three locking techniques which rely on a unified locking structure being inserted in the design require far fewer Adjoining Gates to mitigate leakage.

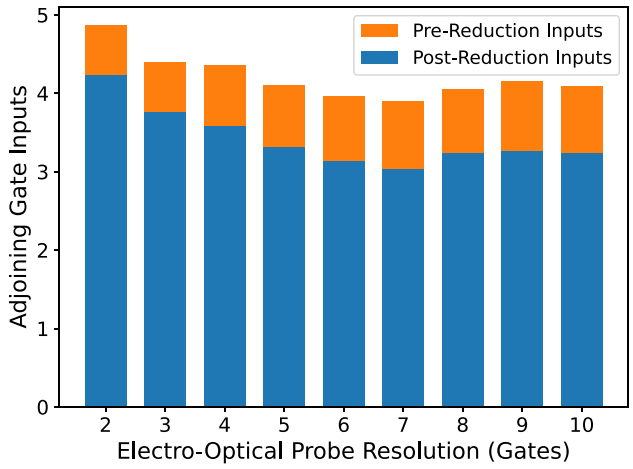

Fig. 9 Average number of Adjoining Gate inputs before/after input optimization

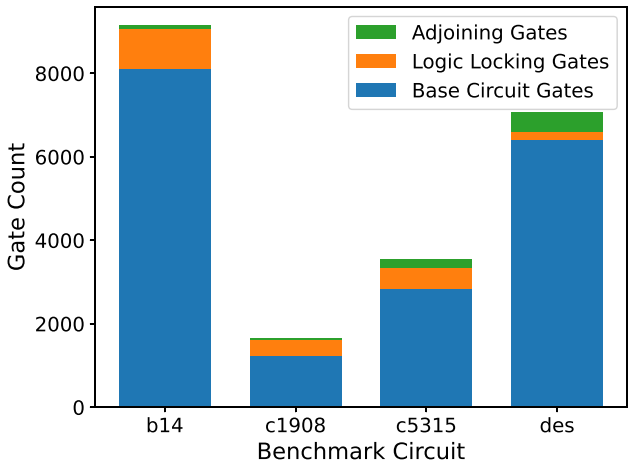

Fig. 10 Average gate count overhead of Adjoining Gates and locking by benchmark circuit

A similar result can be seen in Fig. 8. SLL and Full-Lock had distinctly higher runtime than Anti-SAT and SFLL. This can once again be attributed to the topology of the locking technique. SFLL and Anti-SAT rely on point-function circuits integrated alongside the original circuit that receive the key inputs. Hence, only this added point-function circuit can leak key information because it is the only logic receiving the key inputs. As a result, only target nodes in this region of the design must be analyzed and protected by Adjoining Gates. Conversely, Full-Lock integrates a single locking structure into the core logic of the circuit. This requires that the circuit after this locking structure be analyzed. SLL distributes key inputs throughout the entire design, requiring that nearly the entire design be analyzed and protected.

4.5 Adjoining Gate Input Reduction Optimization

The input reduction optimization was evaluated by comparing the average number of Adjoining Gate inputs before and after applying the overhead optimization proposed in Section 3.4. These results are aggregated in Table 3 and further visualized in Fig. 9. Based on these results, the proposed optimization reduced the average number of inputs required for an Adjoining Gate by 13–20% based on EOFM probe resolution. Across all resolutions, the average Adjoining Gate input reduction was 15.34%. Hence, the proposed optimization achieved a modest improvement in the number of inputs required by Adjoining Gates, thereby reducing the routing pressure and design overhead. We note that this optimization does not impact the effectiveness of Adjoining Gates, with testing both prior to and after the implementation of this optimization achieving the full cessation of EOFM leakage in all benchmark circuits.

4.6 Gate Overhead of Adjoining Gates

To assess the overhead of inserting Adjoining Gates to mitigate EOFM leakage, we quantify the number of Adjoining Gates added to a design. This metric is used because Adjoining Gates can be any gate, or a single pass transistor, as long as they produce switching behavior whenever PIs that fan into the target node are switched. The results of this analysis are in Table 3 and further visualized in Fig. 10. Based on these results, the number of Adjoining Gates added represents an average of 4.15% of the total gate count of the evaluated benchmark circuits.

Based on Table 3, the overhead of Adjoining Gates varies among locking techniques as well, with SLL incurring the highest overhead and SFLL the lowest on average. This makes sense. In SLL, key gates are distributed throughout the circuit, causing many nodes to have key inputs in their fan-in cones, which may cause the nodes to leak. Conversely, SFLL uses the secret key in the restore unit placed alongside the circuit, rather than integrated into it. This results in most of the nodes in the original circuit not having key inputs in their fan-in, reducing the number of potential leaking nodes and requiring fewer Adjoining Gates.

We note that logic locking is a combinational, module-level security scheme [2]. As a result, Adjoining Gates must consider only a combinational cloud, whose size is often limited by timing constraints. The largest evaluated benchmark circuit, b14, exhibits a modest Adjoining Gate overhead. This suggests that Adjoining Gates scale effectively to large combinational circuits. Moreover, prior work indicates that logic locking is applied only to a small subset of carefully selected modules, rather than all modules in a design [21]. This indicates that Adjoining Gates can be applied to locked modules with similar overheads, even in large SoCs.

5 Conclusion

We propose Adjoining Gates, a novel logic locking enhancement to mitigate EOFM attacks on aimed at inferring the secret key. Adjoining Gates are logic gates with a specific functionality that are added in close proximity to nodes leaking key information when illuminated by an electro-optical probe to obfuscate EOFM measurements. They can be implemented with any existing logic locking technique. We then developed an open-source security verification and automation tool to implement Adjoining Gates. This tool scans arbitrary circuits, detects target nodes with EOFM key leakage, and inserts Adjoining Gates to mitigate identified leakage. To evaluate Adjoining Gates, we applied this tool to 16 benchmark circuits of varying size, locking technique, and probe resolution. Adjoining Gates mitigated all identified key leakage in each benchmark circuit with a corresponding 4.15% average increase in gate count.

Declarations

Funding This work was supported by National Science Foundation Grant 2245573.

Conflict of Interest The authors declare no competing interests.

Author Contribution T.W. and M.Z. developed the concept. T.W. developed the software tools. T.W. and R.P. performed the data generation and analysis. T.W., R.P., and M.Z. wrote the manuscript. All authors reviewed the manuscript.

Data Availability The source code, benchmarks, and data generated for this manuscript can be found at https://github.com/twojtal/Adjoining-Gate.

References

1. Rostami M, Koushanfar F, Karri R (2014) A primer on hardware security: models, methods, and metrics. Proc IEEE 102(8):1283–1295
2. Kamali HM, Azar KZ, Farahmandi F, Tehranipoor M (2022) Advances in logic locking: past, present, and prospects. Cryptology
3. Ng YS, Lundquist T, Skvortsov D, Liao J, Kasapi S, Marks H (2010) Laser voltage imaging: a new perspective of laser voltage probing. In: International symposium for testing and failure analysis (ISTFA), pp 5–13
4. Liu SY, Chou HH, Pang MT, Chao KY, Chang JC, Lin JC, Chen CM (2017) Laser voltage imaging and probing, efficient techniques for scan chain verification in advanced node. In: 2017 IEEE 24th international symposium on the physical and failure analysis of integrated circuits (IPFA), pp 1–4. IEEE
5. Kindereit U (2014) Fundamentals and future applications of laser voltage probing. In: 2014 IEEE international reliability physics symposium, pp 3–1. IEEE
6. Rahman MT, Tajik S, Rahman MS, Tehranipoor M, Asadizanjani N (2020) The key is left under the mat: on the inappropriate security assumption of logic locking schemes. In: 2020 IEEE international symposium on hardware oriented security and trust (HOST), pp 262–272. IEEE
7. Shen H, Asadizanjani N, Tehranipoor M, Forte D (2018) Nanopyramid: an optical scrambler against backside probing attacks. In: International symposium for testing and failure analysis (ISTFA)
8. Rahman M, Dipu N, Mehta D, Tajik S, Tehranipoor M, Asadizanjani N (2021) Concealing-gate: optical contactless probing resilient design. ACM J Emerging Technol Comput Syst 17(3):1–25
9. Zuzak M, Liu Y, McDaniel I, Srivastava A (2022) A combined logical and physical attack on logic obfuscation. In: IEEE/ACM international conference on computer-aided design. https://doi.org/10.1145/3508352.3549349
10. Parvin S, Krachenfels T, Tajik S, Seifert J-P, Torres F, Drechsler R (2022) Toward optical probing resistant circuits: a comparison of logic styles and circuit design techniques. In: 2022 27th Asia and south pacific design automation conference (ASP-DAC)
11. https://github.com/twojtal/Adjoining-Gate/
12. Yasin M, Rajendran JJ, Sinanoglu O, Karri R (2015) On improving the security of logic locking. IEEE Trans Comput Aided Des Integr Circ Syst 35(9):1411–1424
13. Xie Y, Srivastava A (2019) Anti-sat: mitigating sat attack on logic locking. IEEE Trans Comput Aided Des Integr Circ Syst 38(2):199–207. https://doi.org/10.1109/TCAD.2018.2801220
14. Kamali H, Azar K, Homayoun H, Sasan A (2019) Full-lock: hard distributions of sat instances for obfuscating circuits using fully configurable logic and routing blocks. In: ACM/IEEE design automation conference (DAC), pp 1–6
15. Sengupta A, Nabeel M, Limaye N, Ashraf M, Sinanoglu O (2020) Truly stripping functionality for logic locking: a fault-based perspective. IEEE Trans Comput Aided Des Integr Circ Syst 39(12):4439–4452
16. Li J, Shi Z, Jin Y (2021) An energy-efficient sensor circuit for preventing electro-optical probing attacks. IEEE Trans Circ Syst I: Regular Papers 68(1):183–194. https://doi.org/10.1109/TCSI.2020.3037861
17. Wang X, Lu Y (2019) Light-scattering coatings for counteracting electro-optical probing attacks. IEEE Trans Inf Forensics Secur 14(10):2689–2698. https://doi.org/10.1109/TIFS.2019.2911476
18. Roy S, Farheen T, Tajik S, Forte D (2022) Self-timed sensors for detecting static optical side channel attacks. In: International symposium on quality electronic design (ISQED)
19. Jain A, Rahman MT, Guin U (2020) ATPG-guided fault injection attacks on logic locking. In: 2020 IEEE physical assurance and inspection of electronics (PAINE), pp 1–6. IEEE
20. Zhong Y, Jain A, Rahman MT, Asadizanjani N, Xie J, Guin U (2022) AFIA: ATPG-guided fault injection attack on secure logic locking. J Electr Test 38(5):527–546
21. Sengupta A, Ashraf M, Nabeel M, Sinanoglu O (2018) Customized locking of IP blocks on a multi-million-gate SOC. In: 2018 IEEE/ACM international conference on computer-aided design (ICCAD), pp 1–7. IEEE