Reversible Circuits: IC/IP Piracy Attacks and Countermeasures

Samah Mohamed Saeed, *Member, IEEE,* Alwin Zulehner, *Member, IEEE,*
Robert Wille, *Senior Member, IEEE,* Rolf Drechsler, *Fellow, IEEE,* Ramesh Karri, *Senior Member, IEEE,*

***Abstract*—Reversible circuits employ a computing paradigm which is useful in a broad variety of applications. With increasing interest, also security concerns for those circuits will raise in the near future. At first glance, reversible circuits seem to be more secure to IC/IP piracy than conventional circuits since the target function is usually *embedded* in the reversible backbone circuit. This embedding adds ancillary inputs and garbage outputs that may appear to hide the target function. However, recent work showed that target function embedding and reversible synthesis methods leave telltale signs in the reversible circuits which allow for an easy extraction of the synthesis approach and the embedded circuit. In this paper, we perform an analysis of the IC/IP piracy attacks on reversible circuits. We focus on reversible circuits generated using QMDD- and BDD-based synthesis approaches as case studies. We show that most of the target function can be identified using the telltale signs of the synthesis approach. We then propose a cost-effective input/output scrambling scheme that wipes out these telltale signs, and thus, thwarts the considered attacks by adding reversible gates. Those additional gates yield efficient yet secure reversible circuits.**

***Index Terms*—Reversible logic, IC/IP piracy, Security, BDD-based synthesis, QMDD-based synthesis, Number of embeddings, Scrambling scheme.**

I. Introduction

Reversible computing has applications in quantum computing [1], [2], encoding/decoding devices [3]–[5], low-power design/adiabatic circuits [6]–[10], and optical computing [11]. Depending on the application, the fabrication processes may either be significantly different to conventional CMOS (e.g. in case of quantum computation where the technology is different) or similar to it (e.g. in case of encoding/decoding devices or low-power design/adiabatic circuits where established technologies are used).

For applications where the fabrication process is similar to conventional CMOS circuits, similar vulnerabilities will emerge. This includes Intellectual Property (IP)/Integrated Circuit (IC) piracy, reverse engineering, Hardware Trojan, and side-channel analysis [12]–[17]. Although fabricating reversible circuits based on those technologies is an on-going research problem (with a recent accomplishments reported e.g. in [10], [18]–[22]), its foreseeable impact is the motivation behind the study and the analysis of the security vulnerabilities. In fact, the difficulty of detecting malicious circuitry, i.e. Trojans, which have been inserted into reversible circuits was recently studied [23]. In this paper, we focus on recovering the functionality of the reversible circuit by launching an IC/IP piracy attack on reversible circuits.

At a first glance, reversible circuits appear to be more secure than conventional circuits because of the way they embed target functions into a reversible backbone circuit, in which the core of the reversible circuit is the target function. This process entails adding *ancillary inputs* and *garbage outputs* to the circuit. Without the knowledge of the ancillary inputs and garbage outputs, the target function seems to be hidden in the backbone reversible circuit. However, a recent analysis that considered IC/IP piracy of reversible circuits has shown that synthesis approaches leave telltale signs in the circuits, which enables an attacker to identify the synthesis approach [24], [25] and then reverse the synthesis process, to recover the target function.

In this work[1], we show how to reverse the synthesis process to retrieve the ancillary inputs and garbage outputs, and thus, the embedded function. By this, we show how an attacker can exploit the telltale signs of a synthesis approach in the gate level netlist to steal the IP. Furthermore, we propose an approach that wipes out the telltale signs and, hence, make it much harder to reverse engineer the function. The proposed method wipes out the telltale signs by adding reversible gates post-synthesis, which also target creating more optimization opportunities using design rules for optimization.

Experimental evaluations confirm the effectiveness of the proposed attacks and countermeasures in hiding the target function by wiping out the telltale signs. We shed light on QMDD- and BDD-based synthesis as examples of functional and structural synthesis approaches. While our discussion focuses on these two synthesis approaches, we emphasize that in a similar fashion telltale signs can be newly obtained for completely other synthesis approaches as well. We show the attack results on reversible circuits generated using QMDD- and BDD-based synthesis approaches before and after applying the proposed scrambling scheme. Results show that hiding the target function using additional gates offers an improved security level with limited hardware overhead.

The remainder of this paper is as follows: Section II provides the background on reversible logic and synthesis

Samah Mohamed Saeed is with the Department of Electrical Engineering, City College of New York, City University of New York, New York, NY, 10030 USA e-mail: ssaeed@ccny.cuny.edu.

Alwin Zulehner and Robert Wille are with Johannes Kepler University, Linz, Austria e-mail: azulehner@ica.jku.at, robert.wille@jku.at.

Rolf Drechsler is with University of Bremen/DFKI, Bremen, Germany e-mail: drechsler@uni-bremen.de.

Ramesh Karri is with New York University, New York, NY, 11201 USA e-mail: rkarri@nyu.edu.

[1] A preliminary version of this paper has appeared at IEEE/ACM International Conference On Computer Aided Design in [26].

approaches that generate reversible circuits. The motivation, the threat model, and the security metric are provided in Section III. The IC/IP piracy attacks on reversible circuits are provided in Section IV. The proposed scrambling scheme to hide the target function of a reversible circuit is described in Section V. Simulation results in Section VI demonstrate strength of the attacks and the effectiveness of the proposed countermeasure in terms of the amount of information leakage, the number of embeddings, and the hardware cost. Finally, we conclude the paper in Section VII.

II. Background

A. Reversible circuits

A reversible function has an equal number of inputs and outputs and maps each input to a unique output and vice versa. A reversible gate is used to build reversible circuits. Each reversible gate over the inputs $X = \{x_1, \ldots, x_n\}$ consists of a (possibly empty) set $C_i \subseteq \{x_j \mid x_j \in X\} \cup \{\overline{x}_j \mid x_j \in X\}$ of positive (x_j) and negative ($\overline{x}_j$) *control lines* and a set $T \subset X \setminus C$ of *target lines*. The *Toffoli* gate $TOF(C, x_t)$ [27] is a commonly used reversible gate, which consists of a single target line and positive/negative control lines. The value of the target line is inverted if all values on the positive (negative) control lines are set to 1 (0) or if $C = \emptyset$. The quantum cost is used to compute the hardware cost of the reversible circuit. For $|C|$ positive/negative control lines, the hardware cost of the Toffoli gate is computed as $2^{|C|+1} - 3$. If the Toffoli gate is entirely composed of negative control lines the quantum cost is increased by two [28][2].

Example 1. *Fig. 1 shows a* 4×4 *reversible circuit composed of four Toffoli gates. For the input* $x_1x_2x_3x_4 = 0011$*, the output* 1001 *is generated. Since the positive control lines of the first two gates from the left-side are assigned 1 using this input, the value of the target line is inverted. However, the third and the fourth gates keep the value of their target line since their positive control lines are assigned 0. In this example, the hardware cost of the reversible circuit is the summation of the hardware cost of all the Toffoli gates, which is 12.*

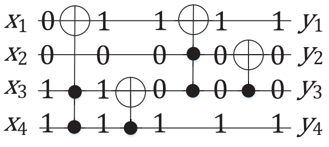


Fig. 1. Reversible circuit realizing a full adder.

Since most of the classical functions are non-reversible, they cannot be realized as reversible circuits. In these cases, the target function is *embedded* into a reversible backbone circuit [31]. *Ancillary inputs* and *garbage outputs*, respectively, are utilized for this purpose. These ancillary inputs and garbage outputs allow for a reversible function which, at the same time, implements the target function when the ancillary inputs are set to a specific constant value.

Example 2. *Consider a full adder, which is a non-reversible function. This target function can be realized as reversible circuit using garbage outputs and ancillary inputs as shown in Fig. 1. In fact, setting the input* x_1 *to* 0 *(i.e. making it an ancillary input) yields the sum on output* y_2 *and the carry-out on output* y_1*. At the same time,* y_3 *and* y_4 *become garbage outputs that can be ignored.*

B. Reversible synthesis and optimization

A non-reversible function is embedded into a reversible circuit. This embedding is done either explicitly or implicitly. *Functional synthesis schemes* [32], [33] which only accept reversible functions as input support explicit embedding. In contrast, *structural synthesis schemes* [34], [35] accept a non-reversible function to be synthesized as an input, generate the reversible circuit during the synthesis, and, by this, embed the function implicitly. The considerations in this paper hold for both types of synthesis schemes. As an illustration, we are focusing on representatives for each scheme:

- QMDD-based synthesis [33] is a functional synthesis approach. A reversible function, which embeds the target function, is provided as a permutation matrix and compactly represented by a *Quantum Multi-valued Decision Diagram* (QMDD) [36]. The permutation matrix maps each input combination (column) to a unique output combination (row). The reversible function is transformed to the identity matrix by considering one variable (i.e. circuit line) after another. This corresponds to swapping columns in the permutation matrix by adding Toffoli gates into the reversible circuit. While random embedding can be applied to convert a non-reversible function to a reversible one, automated embedding scales to large designs and minimizes the hardware overheard of the reversible circuits. We consider an efficient embedding approach [31] to construct the reversible function. In the proposed embedding, the minimum number of ancillary inputs and garbage outputs is added. Then, the columns of the permutation matrix with functional input assignment to the ancillary inputs are swapped with other functional/non-functional columns to form the identity matrix.
- BDD-based synthesis [34] is an example of a structural synthesis approach. The target function is represented as a *Binary Decision Diagrams* (BDD) [37], in which each function/sub-function is represented by a node controlled by input x_i and decomposed using Shannon decomposition
$$f = \overline{x}_i \cdot f_{x_i=0} + x_i \cdot f_{x_i=1},$$
where a function $f_{x_i=0}$ ($f_{x_i=1}$) is the negative (positive) co-factor of f obtained by assigning x_i to 0 (1). Each node of the BDD is mapped to a pre-defined reversible sub-circuit based on the type of the corresponding BDD node. Fig. 2 shows the pre-defined reversible sub-circuit for each type of the BDD node. The reversible sub-circuits are then composed to yield the complete reversible circuit.

[2] Note that advanced quantum cost metrics have been proposed e.g. in [29], [30]. However, for the purpose considered here, the definition from above is sufficient.

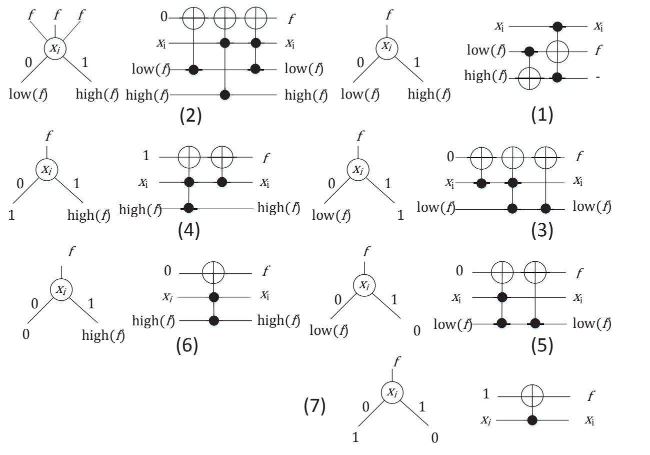


Fig. 2. The mapping of different BDD nodes to reversible sub-circuits.

Example 3. *Consider the function $f = x_1x_2\overline{x}_3x_4 + x_1x_2x_3$. The reversible circuit obtained by QMDD-based synthesis is shown in Fig. 3. Here, it can be seen how each variable is realized one after another (variable x_5 is first realized in region 1 and then x_1 is realized in region 2). For example, the proposed scheme swaps column* 11000 *with* 11001*, and then swaps column* 11001 *with* 01001 *(accomplished by $TOF\left(\{x_1, x_2, \overline{x}_3, \overline{x}_4\}, x_5\right)$ and $TOF\left(\{x_2, \overline{x_3}, \overline{x}_4, x_5\}, x_1\right)$, respectively). Accordingly, the BDD and the reversible circuit obtained by BDD-based synthesis are shown in Fig. 4(a) and 4(b), respectively. Sub-function f_1 represents the identity and, can be realized using the primary input x_4. For the sub-functions f_2, f_3, and eventually f, mappings of the BDD nodes can be applied based on Fig. 2 and composed as annotated in Fig. 4(b) – eventually realizing the overall function.*

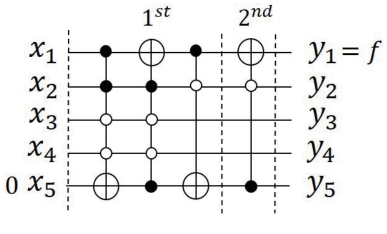
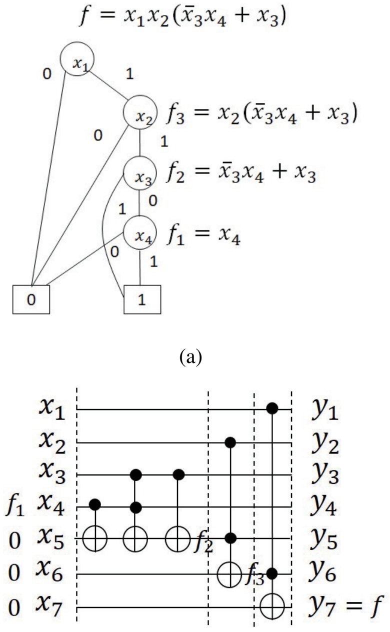


Fig. 3. Circuit obtained by QMDD-based synthesis.

Post-synthesis optimizations that reduce the cost of reversible circuits include templates/rules-based approaches [32], [38] and common control line reduction-based approaches [39]–[41]. Template matching reveals structured cascades of positively/negatively controlled Toffoli gates and map them to smaller optimized reversible sub-circuits. Template matching considers cascades of Toffoli gates composed of a different number of lines. As the number of control lines of Toffoli gates contributes to the hardware cost, sharing common control line reduces the hardware cost. The cascades of reversible gates with common control lines can be optimized by replacing each Toffoli gate in the cascade, except one that shares the largest number of control lines with its neighbors, with two copies of a cheaper gate.



(a)

(b)

Fig. 4. (a) BDD of $f = x_1x_2\overline{x}_3x_4 + x_1x_2x_3$. (b) Reversible circuit obtained by BDD-based synthesis.

III. Motivation, Threat Model, and Security metric

A. Motivation and threat model

Synthesis of reversible circuits explicitly or implicitly embeds the target function into a reversible function. The resulting reversible circuit never directly realizes the non-reversible target function, but naturally hides it by adding ancillary inputs (for which the respective positions and values have to be known in order to recover the target function) and the garbage outputs (for which the respective positions have to be known to read out the target function). At a first glance, this seems to make reversible circuits inherently secure to IC/IP piracy compared to conventional circuits.

To illustrate that, let's first consider the *threat model* assumed for this case: Consider an attacker in the foundry who has access to the gate-level implementation of the reversible circuit and wants to obtain its functionality. But since government agencies such as the Department of Defense use a strictly controlled distributed design flow, the attacker does not have access to the functional input/output of the chip. As the target function is implicitly or explicitly embedded into the reversible circuit, the attacker has to first determine the positions and values of the ancillary inputs and the positions of the garbage outputs – apparently hiding the target function.

Example 4. *Consider the reversible circuit in Fig. 1. From the attacker perspective, there are many functions embedded into this reversible circuit. For example, if x_1 is the ancillary input with value 1 and y_1 and y_2 are the primary outputs, then $y_1 = \overline{x_3x_4} \oplus x_2(x_3 \oplus x_4)$ and $y_2 = x_3 \oplus x_4 \oplus x_2$. On the other hand, if x_1 is the ancillary input with value 0 and y_1 and y_2 are the primary outputs, then $y_1 = x_3x_4 + x_2(x_3 \oplus x_4)$ and $y_2 = x_3 \oplus x_4 \oplus x_2$.*

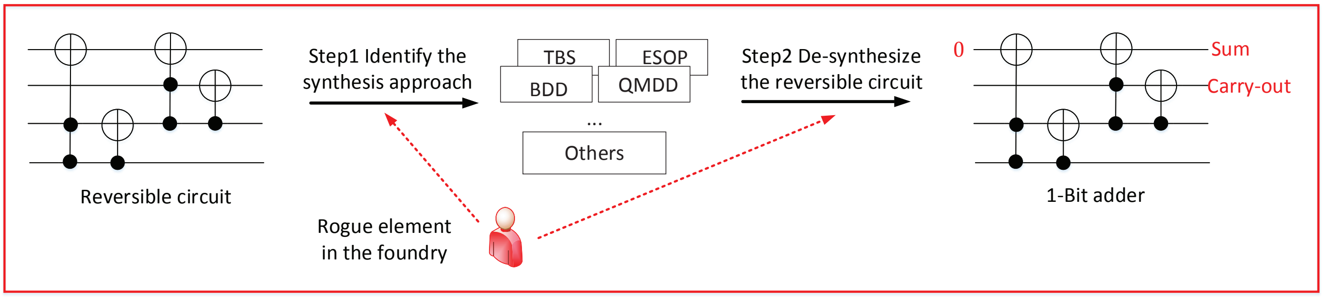

Reversible circuit
Step1 Identify the synthesis approach
TBS
ESOP
BDD
QMDD
...
Others
Rogue element in the foundry
Step2 De-synthesize the reversible circuit
0
Sum
Carry-out
1-Bit adder

Fig. 5. Steps for recovering the target function of a reversible circuit without access to the functional chip. This paper focuses on step 2 of the attack.

We consider IP piracy in which the value of the ancillary inputs enables identifying the functionality. Without proper values of the ancillary inputs, none of the reversible chips will function properly. If the fab overproduces the chips, at a very small additional cost, and sell them on the black market, these chips will not function without the value of the ancillary inputs. This will prevent IC piracy.

We focus on the gate-level of the reversible circuit which can be obtained from the layout in the case of untrusted foundry. We keep our analysis generic despite the underlying technology. The physical restrictions/characteristics of reversible computing applications frequently change (such as CMOS-based realization of reversible circuits [9], followed by several other realizations including Reversible Quantum-Flux-Parametron Logic (RQFP) [42] and most recently conditional reversible circuits [43]), while core concepts (such as relying on reversible computing) remain. For example, while our attack can be applied directly to CMOS-based reversible circuits, the technology of conditional reversible circuits is still under investigation. Hence, to remain applicable also for these changes, we focus on the core concepts which are required for the emerging applications.

Finally, while this threat model is inapplicable to quantum computing, under different threat modes, the applied synthesis approach can imply the value of the ancillary inputs and thus, the Boolean function (oracle) of the quantum circuit.

B. Security metric

In order to recover the target function of a reversible circuit, an attacker is forced to explore the search space of all possible non-reversible functions. Hence, we propose the number of embeddings as the baseline security metric, which depends on the unknown ancillary inputs and garbage outputs.

A reversible function f can be represented as $f(x_1, x_2, \cdots, x_n) = (y_1, y_2, \cdots, y_n)$. Let k_i be the number of inputs that drive an output bit y_i but not y_p where $1 \leq p < i$. The number of functions embedded into n primary outputs circuit is $\prod_{i=1}^{n}(\sum_{j=0}^{k_i} C(k_i, j) \times 2^j)$, where the binomial coefficient $C(k_i, j)$ refers to the number of ways of selecting j $(0 \leq j < k_i)$ un-ordered ancillary inputs from k_i inputs that drive output y_i. An upper bound of the number of all possible non-reversible functions embedded into an n input/output reversible circuit can be derived as: # of embeddings $\leq (2^n - 1) \times \prod_{i=1}^{n}(\sum_{j=0}^{k_i} C(k_i, j) \times 2^j)$, where $(2^n - 1)$ is the number of all possible sets of the primary outputs of the target function[3]. If the location of the ancillary inputs is known, the upper bound of the number of embedding will be reduced to $2^m \times (2^n - 1)$, where m is the number of ancillary inputs. If the location of both the ancillary inputs and garbage outputs are known, the upper bound of the number of embedding will be further reduced to 2^m.

Example 5. *Consider the reversible circuit in Fig. 1, in which* $k_1 = 4, k_2 = 0, k_3 = 0, k_4 = 0$ *and* $n = 4$. *The number of embeddings is* $(2^n - 1) \times \prod_{i=1}^{n}(\sum_{j=0}^{k_i} C(k_i, j) \times 2^j) = (2^4 - 1) \times \prod_{i=1}^{4}(\sum_{j=0}^{k_i} C(k_i, j) \times 2^j) = (2^4 - 1) \times 81 = 1215.$

IV. IC/IP Piracy of Reversible Circuits

In order to recover the target function of the reversible circuit, an attacker should first identify the synthesis approach that generated the reversible circuit. Next, this information can be used to reverse the synthesis process (referred as de-synthesis) to reveal the ancillary inputs as well as the garbage outputs. This is illustrated in Fig. 5.

Recent studies showed that the embedding and synthesis methods such as those reviewed in this paper leave telltale signs [24], [25]. These telltale signs can be mapped to features that a machine learning scheme can use to identify the synthesis approach[4]. Once the synthesis approach has been identified, the ancillary inputs and the garbage outputs can be determined and, hence, the target function can be extracted. This is illustrated by means of the two considered synthesis approaches:

QMDD-based synthesis: Reversible circuits generated using QMDD-based synthesis can be partitioned into n regions, where n is less than or equal to the number of variables in the circuit. In each region, one unique variable is transformed to the identity by adding additional Toffoli gates, and thus, used as either a control or a target line in each gate within the region. Toffoli gates are inserted to convert a non-reversible function into a reversible one by swapping functional input assignment columns with functional/non-functional columns. In other words, each swap operation is represented by a Toffoli gate. Thus, the main telltale sign of QMDD-based synthesis is that the functional assignments to the ancillary inputs activate the largest number of Toffoli gates. An attacker can traverse the reversible circuit to generate input patterns that

[3] One is subtracted to exclude the case where all the output bits are garbage.

[4] For a detailed discussion refers to [24], [25].

activate each reversible gate. This is analogous to test pattern generation of input test patterns that activate and propagate missing target line faults [44]. Next, the attacker analyzes the test patterns to recover the target function. If the location of the ancillary inputs is known to the attacker, he/she can identify the functional assignment of the ancillary inputs by monitoring the ancillary inputs value in the input patterns that activate the maximum number of the reversible gates. Otherwise, the attacker identifies the location of the primary inputs by extracting the unstable bits in the input patterns that activate the maximum number of the reversible gates. In other words, the attacker observes the test patterns with maximum number of Toffoli gates and identifies the location of the bit flips (i.e. primary inputs).

Example 6. *Consider the reversible circuit of function $f = x_1x_2\overline{x}_3x_4 + x_1x_2x_3$ realized by QMDD-based synthesis as shown in Fig. 3. As discussed before in Example 3, circuits obtained by QMDD-based synthesis realize one variable after another using Toffoli gates. The following input patterns $x_1x_2x_3x_4x_5 = 11000$ and $x_1x_2x_3x_4x_5 =$ 10--0, where - can be zero or one, activate the maximum number of Toffoli gates. If the location of the ancillary input x_5 is known, the attacker can easily identify its ancillary value, $x_5 = 0$, and thus, most of the target function. Thus, the number of embeddings is 2^2-1 since there are only two potential primary outputs. However, if the location of the ancillary input is unknown, the attacker can identify most of the primary inputs by observing the input bits differences in these patterns. In this example, x_2, x_3 and x_4 exhibit bit flips in the top input patterns, and thus classified as primary inputs. The number of embeddings is 12, which is a function of the number of all possible primary outputs and ancillary inputs (two possible primary outputs (y_1, y_5) and ancillary inputs (x_1, x_5)[5]).*

BDD-based synthesis: Reversible circuits generated using BDD-based synthesis can be identified by the respective sub-circuits used to realize each BDD node as shown in Fig. 2. One particular telltale sign here is that primary inputs are never used as target lines – clearly identifying the ancillary inputs and some of the garbage outputs. In addition, the target line of a reversible gate that does not control any successor gate is connected to a primary output. Finally, the type of the reversible sub-circuit indicates the associated ancillary input value, which is shown in Fig. 2. Using these telltale signs the attacker first differentiates between primary and ancillary inputs and primary and garbage outputs. Next, the attacker partitions the reversible circuits into sub-circuits. Each sub-circuit consists of the maximum number of gates that map to a pre-defined reversible sub-circuit as shown in Fig. 2, which often determines the associated ancillary input value.

Example 7. *Consider again the reversible circuit of function $f = x_1x_2\overline{x}_3x_4 + x_1x_2x_3$ realized by BDD-based synthesis as shown in Fig. 4(b). The primary inputs of the reversible circuit correspond to control-only variables, which are connected to garbage outputs. Thus, x_1, x_2, x_3, and x_4 are primary inputs and the remaining input variables are ancillary inputs, while y_1, y_2, y_3, and y_4 are garbage outputs. The sub-circuit type indicates the corresponding ancillary input value. Based on Fig. 2, the first sub-circuit that represents f_2 is associated with ancillary input of value zero ($x_5 = 0$). Similarly, the second, and third sub-circuits that represent f_3 and f are associated with zero too ($x_6 = 0, x_7 = 0$). y_7 is primary output while y_5 and y_6 are potential primary outputs. Thus, the number of embeddings is 4, which can be reduced to 1 if we considered the potential primary outputs as garbage outputs.*

V. Hiding the Function of the Circuit

We have shown that reversible circuits generated using different synthesis approaches are vulnerable to IC/IP piracy attacks that reveal the target function of the reversible circuit. A potential approach to hide the target function of the reversible circuit is by adding extra ancillary inputs and garbage outputs prior to synthesis, which embeds an arbitrary function. While this approach can partially hide the target function, the telltale signs of the synthesis approach can identify the ancillary inputs and garbage outputs added by the synthesis approach. The extra ancillary inputs and garbage outputs added to the target function prior to the synthesis approach are treated as primary inputs and outputs, respectively. Therefore, although the attacker can identify most of the ancillary inputs and garbage outputs added by the synthesis approach, the number of embeddings is larger since every primary input is a potential ancillary input, and every primary output is a potential garbage output. That is, adding further ancillary inputs and garbage outputs increase the complexity of IC/IP piracy attacks. However, this also comes with a significant increase in the hardware cost.

Table I and II provide the attack results on reversible circuits generated using QMDD- and BDD-based synthesis approaches, respectively, in the presence of extra ancillary inputs added prior to the synthesis/embedding approach. For each n input/output reversible circuit in Table I, we created four variants of the reversible circuit with 0 (original reversible circuit), 0.5×n, 1×n, and 2×n additional ancillary inputs, where n is the number of circuit lines of the original reversible circuit. We report the number of embeddings (#Embeddings), percentage of primary input leakage (%L_P) and the hardware cost (Cost) defined in Section II-A. Additional ancillary inputs significantly inflate the number of embeddings. While an attacker can not identify most of the primary inputs of the original reversible circuit, he/she can reveal the ancillary inputs value for known location of the ancillary inputs, which necessitates a countermeasure to thwart this attack. However, input scrambling using additional ancillary inputs leaks more information about the location of the primary inputs.

We created four variants of the reversible circuits generated using BDD-based synthesis with 0 (original reversible circuit), 0.1×n, 0.2×n and 0.5×n extra ancillary inputs[6] as shown in Table II. We reduce the ratio of the additional ancillary inputs

[5] Each potential ancillary input has an expected value ($x_1 = 1, x_5 = 0$), which is obtained from the input pattern with the maximum number of activated Toffoli gates.

[6] The number of extra ancillary inputs in QMDD- and BDD-based reversible circuits varies from 1 to 21.

TABLE I
IC/IP PIRACY ATTACKS ON REVERSIBLE CIRCUITS GENERATED USING QMDD-BASED SYNTHESIS WITH ADDITIONAL ANCILLARY INPUTS. %L_P IS THE PERCENTAGE OF LEAKED PRIMARY INPUTS.

Benchmark	Original reversible circuit			+Additional Ancillary Inputs								
				0.5×			1×			2×		
	%L_P	#Embeddings	Cost	%L_P	#Embeddings	Cost	%L_P	#Embeddings	Cost	%L_P	#Embeddings	Cost
4mod5	0	192	78	75	1.12E+07	1226	75	4.08E+12	5798	75	9.58E+13	34232
4mod7	0	1.20E+04	646	0	2.66E+18	39114	0	1.44E+24	317998	100	4.12E+44	2179186
C17	0	2.44E+09	1245	0	1.09E+22	36144	80	2.86E+27	187242	40	3.44E+138	41803462
decode-en	0	2.44E+09	750	0	2.11E+16	54899	0	1.09E+22	190790	0	5.66E+55	1465585
cm150	61.9	3.70E+47	6.41E+07	29	9.51E+176	2.76E+09	24	2.37E+285	4.03E+08	24	1.88E+64	2.12E+09

TABLE II
IC/IP PIRACY ATTACKS ON REVERSIBLE CIRCUITS GENERATED USING BDD-BASED SYNTHESIS WITH ADDITIONAL ANCILLARY INPUTS. %L_A DENOTES THE PERCENTAGE OF LEAKED ANCILLARY INPUTS.

Benchmark	Original reversible circuit			+Additional Ancillary Inputs								
				0.1×			0.2×			0.5×		
	%L_A	#Embeddings	Cost	%L_A	#Embeddings	Cost	%L_A	#Embeddings	Cost	%L_A	#Embeddings	Cost
4mod7	100	1.00E+00	86	75	844	103	75	844	103	75	1.33E+03	122
5xp	91.3	4.00E+00	254	82.46	3.20E+06	689	77.78	9.80E+06	696	73.91	1.80E+08	707
adr4	100	1.00E+00	74	64.29	3.07E+05	109	60	3.70E+06	233	58.33	3.40E+06	247
alu1	93.8	2.00E+00	139	81.8	1.91E+07	192	83.3	2.87E+07	200	66.7	6.20E+09	218
sqrt	72.7	6.40E+01	240	70.4	2.45E+06	266	71.4	3.71E+06	317	68.8	3.38E+07	320

due to the large number of circuits lines of reversible circuits generated using BDD-based synthesis. We report the number of embeddings (#Embeddings), the percentage of leaked ancillary inputs (%L_A), and the hardware cost (Cost). The additional ancillary inputs increase the number of embeddings and the quantum cost while reducing the ancillary input leakage. In few cases, increasing the number of additional ancillary inputs also increases the ancillary inputs leakage due to the random value of the additional ancillary inputs, which impacts the structure of the BDD, the selected pre-defined sub-circuits to construct the reversible circuit, and thus, the ancillary inputs value. We conclude that, while adding new ancillary inputs prior to the synthesis approach increases the number of embeddings, most of the ancillary inputs added by the synthesis approach itself can still be recovered using the telltale signs of the synthesis approach. In addition, this naive scrambling scheme results in a significant hardware overhead.

An alternative procedure to thwart IC/IP piracy attacks in a cost-effective manner without leaking any information about the ancillary inputs and the garbage outputs is to destroy the telltale signs of the synthesis approaches while, preserving the functionality of the circuit. We propose a *scrambling* scheme to destroy the telltale signs introduced by the synthesis approaches making it harder to extract the target function. Our scrambling proposal adds reversible gates to the circuit to destroy its telltale signs post-synthesis. Since additional reversible gates increase the hardware cost, we propose to use post-synthesis optimization (e.g. in [32], [38]–[41]) to add reversible gates so that the telltale signs are destroyed *and* optimization opportunities are created.

Our proposed scrambling scheme is illustrated in Algorithm 1. The number of telltale signs depends on the synthesis approach itself, and thus, varies from one synthesis to the other. These telltale signs are extracted from reversible circuits generated using the corresponding synthesis approach prior to the scrambling scheme.

Algorithm 1: The scrambling approach to erase the telltale signs of the reversible synthesis approach.

```
Input: Reversible circuit G, telltale signs of the synthesis
       approach TS
Output: Modified reversible circuit

Cost($x_i$) is the hardware cost of the reversible circuit;
#TS(G) is the number of telltale signs of reversible circuit G;

// Reorder reversible gates
for each reversible gate $g_i$ do
    identify a gate $g_j$ that can be swapped with $g_{i+1}$[7] to
      construct a sub-circuit that partially matches a design rule
      for optimization
end

// Add non-functional reversible gate for scrambling
for each telltale sign $ts_i$ do
    for each sub-circuit that is partially optimized do
        if ($\exists$ $TOF(C_i, t_i)$ : $G^{'}$ = $TOF(C_i, t_i) \cup G$) &&
          #TS($G^{'}$) <#TS(G) && Cost($G^{'}$) <Cost(G) then
            G= $G^{'}$;
            #TS($G$)= #TS($G$) -1;
            break;
        else
    end
end
for each of the remaining telltale signs $ts_i$ do
    for each possible new $TOF(C_i, t_i)$ that maintains the
      functionality of the target function do
        if ( $G^{'}$ = $TOF(C_i, t_i) \cup G$) && #TS($G^{'}$) <#TS(G)
          then
            G= $G^{'}$;
            #TS($G$)= #TS($G$) -1;
            break;
        else
    end
end
return $G$;
```

[7] A Toffoli gate $TOF(C_i, t_i)$ can be swapped with the adjacent Toffoli gate $TOF(C_{i+1}, t_{i+1})$ if $t_{i+1} \notin C_i$, and $t_i \notin C_{i+1}$.

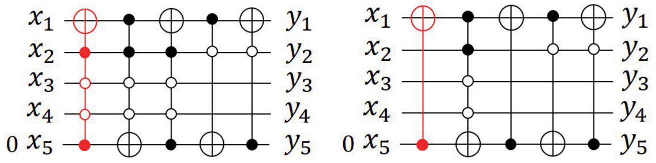
We use templates/rules-based [32], [38] and common control line reduction-based post-synthesis optimization approaches [39]–[41], which significantly reduce the cost of reversible circuits generated using BDD- and QMDD-based synthesis. In templates/rules-based optimization, we reorder adjacent reversible gates to partially match any optimization template, while in common control line reduction, we reorder adjacent reversible gates to increase the number of common control lines. To remove the telltale signs at minimum cost, we first target adding reversible gates that either form an optimization template or share control lines with adjacent reversible gates.

We customized this generic method to reversible circuits generated by BDD- and QMDD-based synthesis.

QMDD-based synthesis: Reversible circuits generated using QMDD-based synthesis consist of Toffoli gates with large number of control lines. Common control lines can be exploited by optimization approaches to reduce the hardware cost. We combine common control line reduction and templates-based optimization with the scrambling scheme as follows:

1) Reorder reversible gates to maximize the number of common control lines of adjacent gates, while preserving the reversible circuit function.
2) Identify the non-functional input patterns that activate the second largest number of reversible gates. Out of these patterns balance the non-functional input pattern with the functional input pattern that activates the maximum number of reversible gates. This is done by adding reversible gates which create new optimization opportunities (templates or common control lines), and thus, minimize hardware cost.
3) If all the optimization opportunities are exhausted, add reversible gates that balance a non-functional input pattern with the functional input pattern that activates the maximum number of gates.

Example 8. *Consider the circuits realized by QMDD-based synthesis shown in Fig. 3. The cost of the QMDD-based reversible circuit is 68. To scramble the QMDD-based reversible circuit, we apply the three steps of our scrambling algorithm. First, we attempt to reorder reversible gates to maximize the number of common control lines of adjacent gates. In this example, gates can not be reordered without violating the functionality of the design. In the next step, we balance the non-functional input pattern* $x_1x_2x_3x_4x_5 = 11001$ *with the functional input patterns* $x_1x_2x_3x_4x_5 = 11000$ *and* $x_1x_2x_3x_4x_5 =$ 10--00 *that activate the maximum number of Toffoli gates by adding* $TOF\left(\{x_2, \overline{x_3}, \overline{x}_4, x_5\}, x_1\right)$ *highlighted in red color in Fig. 6(a). The selected Toffoli gate creates a new optimization opportunity. The reversible circuit after optimization is shown in Fig. 6(b). The hardware cost of the scrambled reversible circuit is 41.*

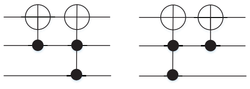


(a) (b)

Fig. 6. Scrambling the QMDD-based reversible circuit from Fig. 3: (a) Balancing functional and non-functional input patterns. (b) Applying post-synthesis optimization.

BDD-based synthesis: While most of the pre-defined reversible sub-circuits used by BDD-based synthesis are associated with unique ancillary input value, there are some pre-defined reversible sub-circuits that can not determine the associated ancillary input value. Lets call them universal sub-circuits. This is illustrated in Fig. 2 in which two pre-defined sub-circuits in case 4 and 5 have the same structure, however, they are connected to different values of the ancillary input. We maximize the number of universal sub-circuits used in the backbone reversible circuit to prevent recovering the ancillary inputs value. We also add non-functional pre-defined reversible sub-circuits to prevent leaking the ancillary inputs value associated with non-universal sub-circuits. We adapt our proposed scrambling algorithm for BDD-based reversible circuits as follows:

1) Reorder the reversible gates to construct more universal gates, while preserving the reversible circuit function. Fig. 7 provides two identical sub-circuits– the first one can uniquely determine the associated ancillary input value, while the second one is a universal sub-circuit. According to Fig. 2, the reversible sub-circuit in Fig. 7(a) consists of two pre-defined sub-circuits, in which the first sub-circuit from the left determines the ancillary input value associated with the target line. On the other hand, the reversible sub-circuit in Fig. 7(b) is mapped to two different pre-defined sub-circuits. Each has a different constant value associated with the target line.
2) For 50% of the uniquely identified ancillary inputs, add non-functional reversible sub-circuits that are associated with the opposite value of the functional ancillary inputs according to Fig. 2. These sub-circuits share common control lines if possible for optimization purpose. The random selection of the 50% of the uniquely identified ancillary inputs prevents the attacker from differentiating between fake and original reversible sub-circuits connected to the ancillary inputs. Therefore, from the attacker perspective, each pre-defined reversible sub-circuits can be associated with any ancillary input value. If we add non-functional pre-defined reversible sub-circuits to all the ancillary inputs, an attacker can guess the ancillary inputs value by inverting the one obtained from Fig. 2.
3) For each control-only circuit line x (i.e. primary input), add a reversible gate in which the target line is also x, while preserving the functionality of the circuit. Thus, there are no outputs directly connected to the inputs of the reversible circuits.

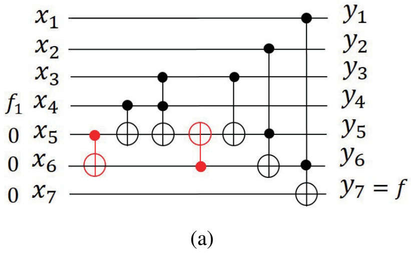


(a) (b)

Fig. 7. Identical reversible sub-circuits (a) and (b).

Example 9. *The hardware cost of the reversible circuit in Fig. 4(b) generated using BDD-based synthesis is 17. To hide the target function, we first attempt to reorder reversible gates to construct universal sub-circuits. In this example, reordering doesn't create any universal reversible sub-circuit. Next, we add non-functional pre-defined reversible sub-circuits to* 50% *of the ancillary inputs that implies the opposite value of randomly selected* 50% *of the ancillary inputs. These gates should be activated by non-functional assignment to the ancillary inputs. For ancillary input* x_5 *we add* $TOF(\{x_6\}, x_5)$*, which violates the structure of the pre-defined reversible sub-circuit in case 3 of Fig. 2. Thus,* $TOF(\{x_4\}, x_5)$ *and* $TOF(\{x_3, x_4\}, x_5)$ *form a universal sub-circuit after reordering. For ancillary input* x_6*, we add a pre-defined reversible sub-circuit* $TOF(\{x_5\}, x_6)$*, which is associated with the opposite value of the ancillary input. These additional Toffoli gates are highlighted in red color in Fig. 8(a). The hardware cost of the resulting reversible circuit is 19. To hide the location of the primary inputs and garbage outputs of the reversible circuit, we add a Toffoli gate to each control-only line as shown in Fig. 8(b). The hardware cost of resulting reversible circuit is 22.*

Even if an attacker has a prior knowledge of the location of the ancillary inputs of the reversible circuit, the proposed scrambling scheme hides the value of the ancillary inputs.



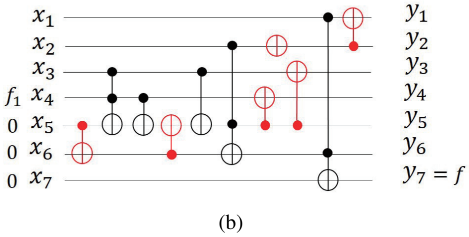


Fig. 8. Scrambling the BDD-based reversible circuit from Fig. 4(b): (a) Adding fake pre-defined sub-circuit. (b) Reordering Toffoli gates and hiding the location of the primary inputs and garbage outputs.

VI. Simulation Results

We evaluate the strength of the IC/IP piracy attacks and the effectiveness of our proposed countermeasure. First, we launch IC/IP piracy attacks on reversible circuits generated using QMDD- and BDD-based synthesis approaches. We report the number of embeddings and the amount of information leakage (ancillary/primary inputs) to evaluate the difficulty of the attack. Next, we show the effectiveness of our proposed countermeasure by launching IC/IP piracy attacks on scrambled reversible circuits. We implemented the scrambling scheme and checked its potency in wiping out the telltale signs of the synthesis algorithms in the resulting reversible circuits. We use templates- [32] and common control line reduction-based [39] optimization approaches. We calculated how likely it is for an attacker to locate the ancillary inputs and the garbage outputs and identify the value of the ancillary inputs, which are used to compute the number of embeddings. We also computed the hardware overhead of the proposed scrambling scheme. Furthermore, we calculated the run time of the proposed scrambling scheme. As for benchmarks, we use circuits from RevLib realized using QMDD- and BDD-based synthesis [45]. We present the results of both synthesis approaches. Finally, we provide a comparison of the scrambling scheme with the approach based on adding extra ancillary inputs.

A. IC/IP piracy attacks

The attack results on QMDD- and BDD-based reversible circuits are summarized in Table III and IV, respectively. In each table, the first column identifies the benchmark. Columns 2-4 provide the number of circuit lines, garbage outputs, and ancillary inputs, of reversible circuits synthesized using the corresponding synthesis approach. Column five refers to the hardware cost of the reversible circuit defined in Section II-A. Columns 6-7 provide percentage of leaked garbage outputs and the number of emebeddings for the base-case prior to the IC/IP piracy attack.

For QMDD-based reversible circuits, columns 8 and 9 of Table III provide the percentage of leaked primary inputs and the number of embeddings, respectively, after running the IC/IP piracy attack. The attacker tries to learn the location of the primary inputs and outputs and the value of the ancillary inputs to recover the target function. We use a naive test pattern generation algorithm [44] to generate test patterns that detect the maximum number of missing target line faults. On average, the attacker can identify 10.5% of the primary inputs, which can be increased if an advanced test pattern generation algorithm using SAT-solver [44] is applied. The attack results indicate that if the location of the ancillary inputs is unknown to the attacker, it is difficult to recover the target function. However, if the location of the ancillary inputs of the reversible circuit is known, the attacker can recover the their value. For 131 benchmarks, the ancillary inputs of 94% (i.e.125-out-of-131) of the reversible circuits are recovered, while the ancillary inputs of the remaining reversible circuits are partially recovered. Thus, the attacker can recover most of the target function. In this case, the number of embeddings primarily depends on the number of potential primary outputs.

For BDD-based reversible circuits, column 8-10 of Table IV provide the percentage of leaked ancillary inputs, number of embeddings, and the number of reduced embeddings after running IC/IP piracy attack. On average, the attacker can recover 81.6% ancillary inputs and 33.9% of the garbage outputs that are directly connected to the inputs. Thus, the resulting number of embeddings is significantly reduced. Experimental results showed that primary outputs that satisfy the second telltale sign of the BDD-based synthesis approach in section IV are the only primary outputs of the reversible circuit. In other words, the output of sub-circuits that are connected to the reversible circuit outputs and control other reversible sub-circuits are considered as garbage outputs. Therefore, the reduced number of embeddings considers only primary outputs that satisfy the second telltale signs of BDD-based reversible circuits.

B. Scrambling-based countermeasure

Table III and IV also summarize the attack results on QMDD- and BDD-based reversible circuits, respectively, after applying our proposed scrambling scheme. In each table, the last three columns (#Embeddings, Cost, and Scrambling run_time), indicate the number of embeddings, the percentage of hardware overhead of the proposed countermeasure (Cost↑), and the run time of the scrambling scheme, respectively. The increase in the hardware cost is computed as follows:

$$Cost \uparrow = 100 \times \frac{(cost\,of\,scrambled\,ckt - cost\,of\,original\,ckt)}{(cost\,of\,original\,ckt)}.$$

The attackers can not identify the primary inputs and outputs of the scrambled QMDD-based reversible circuits. According to the telltale signs of QMDD-based reversible circuits, an input is classified as a primary input if the maximum number of Toffoli gates is activated by all of its possible values. In addition, an output is considered as a garbage output when it is directly connect to the input. In the presence of the scrambling scheme, both primary inputs and ancillary inputs exhibit bit flips. The attacker can not differentiate between primary and ancillary inputs, in which functional and non-functional input patterns activate the maximum number of Toffoli gates. As a result, to compute the number of embeddings, every input is considered as a potential ancillary input. If the additional Toffoli gates does not reduce the percentage of leaked garbage outputs, the number of embeddings of a scrambled reversible circuit will be equal to the corresponding one prior to the IC/IP piracy attack (basic). Since most of the QMDD-based reversible circuits leak 0% of the garbage outputs as shown in Table III, the number of embeddings prior to the IC/IP piracy attack is mostly equal to the corresponding one after applying the proposed countermeasure. Even if the attacker knows the location of the ancillary inputs, he/she leaks 0% of the ancillary inputs due to the additional Toffoli gates that balance the maximum number of reversible gates for both functional and non-functional assignments to the ancillary inputs. In this case, the number of embeddings is a function of the number of all possible values of the ancillary inputs and locations of the garbage outputs. To hide the telltale signs of QMDD-based reversible circuits, on average, 0.2% extra hardware cost has to be expended. Optimization opportunities are created by reordering Toffoli gates. Moreover, few Toffoli gates are required to balance the maximum number of gates that are activated by functional and non-functional ancillary inputs assignments.

In the presence of the scrambling scheme for BDD-based reversible circuits, primary inputs are misclassified as ancillary inputs and garbage outputs are misclassified as primary outputs based on the telltale signs of the BDD-based synthesis approach. Similar to QMDD-based reversible circuits, the attacker can not locate the ancillary inputs. In addition, the scrambling scheme connects each ancillary input with either a universal, a fake (50%), or an original sub-circuit. Therefore, the attacker can recover at most 50% of the ancillary inputs value without knowing the location of the ancillary inputs and garbage outputs. The attacker can not recover the primary inputs and the the garbage outputs ($\%L\_P = 0$ and $\%G = 0$). From the attacker perspective every input behaves as an ancillary input and every output behaves as a primary output. Therefore, the number of embeddings, which is a function of the number of all possible ancillary inputs and primary outputs of the reversible circuit, is very large at the expense of, on average, 12.7% increase in the hardware cost as illustrated in Table. IV. The hardware overhead can be reduced by exploiting the trade-off between the number of embeddings and the hardware cost. This can be done by hiding the location of a subset of the primary inputs in step 3 of the countermeasure instead of considering all the primary inputs. In addition, we observe that for reversible circuits with a large number of lines, the increase in the hardware cost is in the range of 3% to 10%. In other words, for larger reversible circuits, it is expected that the hardware overhead will be smaller. Moreover, the cost of adding Toffoli gates to the primary inputs to hide their location is much smaller compared to the cost of hiding ancillary inputs value. For example, alu2 reversible circuit has the minimum hardware overhead (2.9%), the maximum number of embeddings (13.5E+2602), and a large number of primary inputs (10). However, reversible circuit mlp4, which leaks more ancillary inputs values prior to the countermeasure, but has smaller number of primary inputs (8), requires more hardware overhead (10.2%) to scramble the reversible circuit. Finally, we observe that the number of embeddings after applying the proposed countermeasure is significantly higher than the one in the base-case prior to the IC/IP piracy attack, which is due to outputs scrambling. An attacker can no longer recover any garbage output.

The run time of the scrambling scheme applied to QMDD- and BDD-based reversible circuits is a fraction of seconds, on average, which is a small number and primarily depends on the number of gates (O(N^2), where N is the number of gates). Despite that, the run time of the scrambling scheme applied to QMDD-based reversible circuits is slightly higher than BDD-based reversible circuits due to the large number of reversible gates in QMDD-based reversible circuits.

Table V and VI provide a comparison between adding extra ancillary inputs and the scrambling countermeasures, respectively, in terms of the number of embeddings, the hardware cost, and the amount of information leakage. In Table V(VI), Red. L_P (Red. A_P), # Embeddings, and Red. Cost, refer to the percentage of leaked primary inputs (ancillary inputs), the number of embedding, and the percentage of hardware cost reduction, respectively, of the scrambling approach applied to QMDD- (BDD-)based reversible circuits compared to adding extra ancillary inputs. For example, the number of embeddings of 4mod7 scrambled reversible circuit generated using QMDD-based synthesis is 1.9 multiplied by the number of embeddings when 0.1 x additional ancillary inputs are added prior to the synthesis approach as shown in Table V. While the number of embeddings after applying the scrambling approach to QMDD-based reversible circuits is less than the number

TABLE III
SCRAMBLING USING ADDITIONAL REVERSIBLE GATES FOR REVERSIBLE CIRCUITS GENERATED USING QMDD-BASED SYNTHESIS.

Benchmark	#			Cost	Basic		Attack without scrambling		Attack with scrambling		Scrambling
	I/O	G	A		%G	#Embeddings	%L_P	#Embeddings	#Embeddings	Cost ↑	run_time(s)
4mod5	5	4	1	8.00E+01	75	3.60E+05	0	1.90E+02	2.10E+07	3.80E-02	0.001
4mod7	5	2	1	6.40E+02	50	2.10E+07	0	1.20E+04	3.90E+08	4.70E-03	0.002
adr4	9	4	1	2.30E+05	0	1.50E+24	100	1.80E+16	1.50E+24	4.40E-06	0.088
alu1	18	10	6	5.90E+08	0	1.00E+87	0	7.80E+56	1.00E+87	1.40E-08	3.193
alu2	14	8	4	2.40E+07	0	2.00E+54	0	6.60E+35	2.00E+54	5.00E-07	1.071
alu3	14	6	4	1.60E+07	0	2.00E+54	0	6.60E+35	2.00E+54	4.90E-07	0.833
C7552	20	4	15	6.80E+07	0	1.60E+106	0	1.70E+69	1.60E+106	1.60E-07	0.025
clip	11	6	2	1.70E+06	0	6.30E+34	0	1.50E+23	6.30E+34	3.00E-06	0.339
cm150	22	21	1	6.40E+07	9.5	4.10E+121	61.9	3.70E+47	1.20E+122	6.20E-08	0.168
cm42	13	3	9	1.90E+05	0	2.10E+47	0	2.00E+31	2.10E+47	4.70E-05	0.009
cm82	6	3	1	2.80E+03	0	6.00E+11	0	1.30E+08	6.00E+11	1.10E-03	0.013
cm85	13	10	2	1.40E+06	10	2.40E+46	0	2.50E+30	2.10E+47	5.80E-06	0.078
cmb	20	16	4	3.10E+08	6.3	2.30E+103	0	1.10E+68	3.80E+102	6.50E-09	1.184
con1	8	6	1	9.60E+03	0	3.70E+19	14.3	1.70E+13	3.70E+19	1.00E-04	0.021
dc1	10	3	6	2.20E+04	0	1.80E+29	0	3.70E+19	1.80E+29	4.10E-04	0.018
dk17	19	8	9	1.40E+09	0	2.40E+96	0	8.20E+62	2.40E+96	1.60E-08	7.245
example2	14	8	4	2.40E+07	0	2.00E+54	0	6.60E+35	2.00E+54	5.00E-07	1.142
mlp4	13	5	5	5.00E+06	0	2.10E+47	0	2.00E+31	2.10E+47	4.60E-06	0.162
sqrt8	9	5	1	2.00E+04	40	4.90E+23	25	1.40E+11	4.90E+23	5.10E-05	0.018
squar5	9	1	4	2.10E+04	0	1.50E+24	0	1.80E+16	1.50E+24	1.40E-04	0.021
sym9	10	9	1	2.70E+05	0	1.80E+29	0	3.70E+19	1.80E+29	1.80E-05	0.030
x2	16	9	6	1.40E+07	0	5.10E+69	30	1.40E+45	5.10E+69	5.00E-07	0.286

of embeddings when extra ancillary inputs are added, there is a very large reduction in the hardware cost of scrambled QMDD-based reversible circuits and the location of all their primary inputs are hidden.

VII. CONCLUSION

In this paper, we analyze IC/IP piracy attacks on reversible circuits. We focus on non-reversible functions embedded into reversible circuits. We use the number of embeddings as a security metric to evaluate the difficulty of extracting the target function. We propose IC/IP piracy attacks on reversible circuits generated using QMDD- and BDD-based synthesis approaches as case studies. We also propose a scrambling-based defense to thwart the IC/IP piracy attacks on reversible circuits by adding reversible gates after the synthesis step to erase the telltale signs of the synthesis approach of the reversible circuit. Experimental results show that the proposed countermeasure hides the location of the ancillary inputs and garbage outputs in addition to the value of the ancillary inputs at the expense of limited hardware overhead.

REFERENCES

[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[3] A. Zulehner and R. Wille, "Taking one-to-one mappings for granted: Advanced logic design of encoder circuits," in *Proceedings of IEEE Design Automation and Test in Europe*, 2017.

[4] R. Wille, R. Drechsler, C. Osewold, and A. G. Ortiz, "Automatic design of low-power encoders using reversible circuit synthesis," in *Proceedings of IEEE Design Automation and Test in Europe*, 2012, pp. 1036–1041.

[5] R. Wille, O. Keszocze, S. Hillmich, M. Walter, and A. G. Ortiz, "Synthesis of approximate coders for on-chip interconnects using reversible logic," in *Proceedings of IEEE Design Automation and Test in Europe*, 2016, pp. 1140–1143.

[6] C. H. Bennett, "Logical reversibility of computation," *IBM J.Res.Dev*, vol. 17, no. 6, pp. 525–532, 1973.

[7] A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, and E. Lutz, "Experimental verification of Landauer's principle linking information and thermodynamics," *Nature*, vol. 483, pp. 187–189, 2012.

[8] A. Zulehner, M. P. Frank, and R. Wille, "Design automation for adiabatic circuits," in *Proceedings of Asia and South Pacific Design Automation Conference*, 2019, pp. 669–674.

[9] W. C. Athas and L. J. Svensson, "Reversible logic issues in adiabatic cmos," in *Proceedings of Workshop on Physics and Computation*, 1994, pp. 111–118.

[10] M. P. Frank, "The future of computing depends on making it reversible," *IEEE Spectrum*, 2017.

[11] R. Cuykendall and D. R. Andersen, "Reversible optical computing circuits," *Opt. Lett.*, vol. 12, no. 7, pp. 542–544, Jul 1987.

[12] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[13] J. Rajendran, E. Gavas, J. Jimenez, V. Padman, and R. Karri, "Towards a comprehensive and systematic classification of hardware Trojans," in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2010, pp. 1871–1874.

[14] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference*, 2011, pp. 333–338.

[15] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan attacks: Threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014.

[16] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the International Cryptology Conference on Advances in Cryptology*, 1999, pp. 388–397.

[17] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proceedings of the International Cryptology Conference on Advances in Cryptology*, 1996, pp. 104–113.

[18] W. Wustmann and K. Osborn, "Reversible fluxon logic: Topological particles allow gates beyond the standard adiabatic limit," in *arXiv:1711.04339*, 11 2017.

[19] M. S. M. Hogg, Tad and D. G. Allis, "Mechanical computing systems using only links and rotary joints," *Molecular Systems Design & Engineering*, vol. 2, no. 3, pp. 235–252, 2017.

[20] I. Phillips and H. Rahaman, Eds., *Reversible Computation - 9th International Conference, RC 2017, Kolkata, India, July 6-7, 2017, Proceedings*, ser. Lecture Notes in Computer Science, vol. 10301. Springer, 2017.

[21] S. J. Devitt and I. Lanese, Eds., *Reversible Computation - 8th International Conference, RC 2016, Bologna, Italy, July 7-8, 2016, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9720. Springer, 2016.

[22] "Special issue: Design of reversible computing systems," in *IEEE Transactions on Emerging Topics in Computing*, 2019.

TABLE IV

SCRAMBLING USING ADDITIONAL REVERSIBLE GATES FOR REVERSIBLE CIRCUITS GENERATED USING BDD-BASED SYNTHESIS.

Benchmark	#			Cost	Basic		Attack without scrambling			Attack with scrambling		Scrambling
	I/O	G	A		%G	#Embeddings	%L_A	#Embeddings	#Reduced Embeddings	#Embeddings	Cost ↑.	run_time(s)
4mod7	12	10	8	86	40	1.40E+08	100	3.20E+01	1.00E+00	4.20E+34	14	0.002
5xp1	30	20	23	254	30	3.50E+21	91.3	6.60E+04	4.00E+00	9.20E+218	8.3	0.003
adr4	16	11	8	74	72.7	1.10E+10	100	8.00E+00	1.00E+00	1.90E+68	21.6	0.001
alu1	28	20	16	139	60	1.50E+18	93.8	5.10E+02	2.00E+00	1.70E+179	13.7	0.003
alu2	105	99	95	1436	10.1	2.70E+76	89.5	6.30E+29	1.00E+03	3.5E+2602	2.9	0.056
alu3	66	58	56	644	17.2	2.20E+48	76.8	2.30E+18	8.20E+03	3.1E+1049	5.3	0.009
apla	103	91	93	1002	11	1.40E+77	82.8	1.60E+29	6.60E+04	6.0E+2323	4.4	0.025
C7552	35	19	30	202	26.3	5.40E+25	50	5.40E+08	3.30E+04	5.70E+245	6.9	0.002
clip	66	61	57	704	14.8	4.50E+48	78.9	1.80E+19	4.10E+03	6.5E+990	4.7	0.009
cm150	37	36	16	186	58.3	3.00E+22	0	2.20E+09	6.60E+04	3.6E+346	11.3	0.001
cm42	22	12	18	117	33.3	8.20E+15	50	1.30E+05	5.10E+02	1.80E+87	8.5	0.001
cm82	13	10	8	82	50	4.10E+08	100	3.20E+01	1.00E+00	3.30E+43	18.3	0.012
cm85	36	33	25	275	33.3	5.00E+24	56	8.60E+09	2.00E+03	6.4E+313	14.2	0.014
cmb	43	40	27	158	40	4.40E+28	100	1.70E+07	1.00E+00	4.7E+435	60.8	0.009
co14	27	26	13	159	53.8	6.30E+16	100	4.10E+03	1.00E+00	3.20E+177	11.9	0.002
con1	16	14	9	96	50	2.20E+10	77.8	5.10E+02	4.00E+00	5.10E+69	17.7	0.005
dc1	20	13	16	160	30.8	2.30E+14	87.5	2.00E+03	4.00E+00	3.70E+88	15	0.002
decod	35	19	30	202	26.3	5.40E+25	50	5.40E+08	3.30E+04	3.90E+280	16.3	0.009
dist	79	74	71	975	10.8	1.20E+59	100	7.40E+19	1.00E+00	6.1E+1519	12.1	0.018
dk17	58	47	48	426	21.3	1.30E+42	77.1	2.80E+14	2.00E+03	6.2E+698	7	0.006
f2	16	12	12	113	33.3	1.80E+11	58.3	8.20E+03	3.20E+01	7.70E+65	9.7	0.015
mlp4	103	95	95	1158	8.4	5.50E+77	94.7	5.00E+27	3.20E+01	1.0E+2311	10.2	0.028
sqn	40	37	33	426	18.9	1.00E+29	69.7	1.10E+12	1.00E+03	1.6E+384	13.1	0.004
sqrt8	30	26	22	240	30.8	8.60E+20	72.7	1.70E+07	6.40E+01	3.40E+217	9.6	0.024

TABLE V

SECURITY AND COST ENHANCEMENT OF THE PROPOSED SCRAMBLING APPROACH APPLIED TO QMDD-BASED REVERSIBLE CIRCUITS WITH RESPECT TO ADDING ANCILLARY INPUTS COUNTERMEASURE.

Benchmark	+Additional Ancillary Inputs								
	0.5×			1×			2×		
	%Red. L_P	#Embeddings	% Red. Cost	%Red. L_P	#Embeddings	% Red. Cost	%Red. L_P	#Embeddings	% Red. Cost
4mod5	100	1.9x	92.9	100	5.1E-06x	98.5	100	2.2E-07x	99.7
4mod7	0	1.5E-10x	98.3	0	2.7E-16x	99.8	100	9.5E-37x	99.97
C17	0	5.5E-11x	96.5	100	2.1E-16x	99.3	100	1.7E-127x	99.997
decode-en	0	2.8E-05x	98.6	0	5.5E-11x	99.6	0	1.1E-44x	99.9
cm150	100	1.2E-57x	97.7	100	4.6E-166x	84.1	100	5.9E+55x	97

TABLE VI

SECURITY AND COST ENHANCEMENT OF THE PROPOSED SCRAMBLING APPROACH APPLIED TO BDD-BASED REVERSIBLE CIRCUITS WITH RESPECT TO ADDING ANCILLARY INPUTS COUNTERMEASURE.

Benchmark	+Additional Ancillary Inputs								
	0.1×			0.2×			0.5×		
	%Red. L_A	#Embeddings	% Red. Cost	%Red. L_A	#Embeddings	% Red. Cost	%Red. L_A	#Embeddings	% Red. Cost
4mod7	100	5.0E+31 x	4.8	100	5.0E+31x	4.8E+00	100	3.2E+31x	19.6
5xp	100	2.9E+212x	60.1	100	9.4E+211x	6.0E+01	100	5.1E+210x	61.1
adr4	100	6.2E+62x	17.4	100	5.1E+61x	6.1E+01	100	5.6E+61x	63.6
clip	100	1.4E+211x	1.1	100	9.2E+210x	1.7E+01	100	1.0E+210x	17.8
alu1	100	8.9E+171x	17.7	100	5.9E+171x	2.1E+01	100	2.7E+169x	27.5

[23] X. Cui, S. M. Saeed, A. Zulehner, R. Wille, K. Wu, R. Drechsler, and R. Karri, "On the difficulty of inserting trojans in reversible computing architectures," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2018.

[24] S. Saeed, N. Mahendran, A. Zulehner, R. Wille, and R. Karri, "Identifying reversible circuit synthesis approaches to enable IP piracy attacks," in *Proceedings of IEEE International Conference on Computer Design*, 2017, pp. 537–540.

[25] S. M. Saeed, N. Mahendran, A. Zulehner, R. Wille, and R. Karri, "Identification of synthesis approaches for IP/IC piracy of reversible circuits," *J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 3, pp. 23:1–23:17, Apr. 2019.

[26] S. M. Saeed, X. Cui, A. Zulehner, R. Wille, R. Drechsler, K. Wu, and R. Karri, "IC/IP piracy assessment of reversible logic," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2018, pp. 5:1–5:8.

[27] T. Toffoli, "Reversible computing," in *Proceedings Automata, Languages and Programming*, J. de Bakker and J. van Leeuwen, Eds. Springer Berlin Heidelberg, 1980, pp. 632–644.

[28] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction," *IEEE Transactions on CAD*, vol. 27, no. 3, pp. 436–444, 2008.

[29] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffoli gates," in *Proceedings of IEEE International Symposium on Multiple-Valued Logic*, May 2011, pp. 288–293.

[30] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 6, pp. 818–830, June 2013.

[31] A. Zulehner and R. Wille, "Make it reversible: Efficient embedding of non-reversible functions," in *Proceedings of IEEE Design Automation*

and Test in Europe, 2017, pp. 458–463.

[32] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference*, 2003, pp. 318–323.

[33] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler, "Synthesis of reversible circuits with minimal lines for large functions," in *Proceedings of Asia and South Pacific Design Automation Conference*, 2012, pp. 85–92.

[34] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference*, July 2009, pp. 270–275.

[35] K. Fazel, M. Thornton, and J. Rice, "ESOP-based Toffoli gate cascade generation," in *Proceedings of IEEE PacRim*, 2007, pp. 206 –209.

[36] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler, "QMDDs: Efficient quantum function representation and manipulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 86–99, 2016.

[37] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. 35, no. 8, pp. 677–691, 1986.

[38] K. Datta, G. Rathi, R. Wille, I. Sengupta, H. Rahaman, and R. Drechsler, "Exploiting negative control lines in the optimization of reversible circuits," in *Proceedings of the International Conference on Reversible Computation*, ser. RC'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 209–220.

[39] A. Deb, R. Wille, R. Drechsler, and D. K. Das, "An efficient reduction of common control lines for reversible circuit optimization," in *Proceedings of IEEE International Symposium on Multiple-Valued Logic*, May 2015, pp. 14–19.

[40] D. M. Miller, R. Wille, and R. Drechsler, "Reducing reversible circuit cost by adding lines," in *Proceedings of IEEE International Symposium on Multiple-Valued Logic*, May 2010, pp. 217–222.

[41] R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler, "Improving the mapping of reversible circuits to quantum circuits using multiple target lines," in *Proceedings of Asia and South Pacific Design Automation Conference*, Jan 2013, pp. 145–150.

[42] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Reversible logic gate using adiabatic superconducting devices," in *Scientific reports*, 2014.

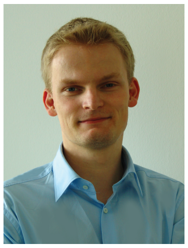
[43] M. P. Frank, "Foundations of generalized reversible computing," in *Reversible Computation*, I. Phillips and H. Rahaman, Eds. Springer International Publishing, 2017, pp. 19–34.

[44] R. Wille, H. Zhang, and R. Drechsler, "ATPG for reversible circuits using simulation, boolean satisfiability, and pseudo boolean optimization," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI*. IEEE, 2011, pp. 120–125.

[45] R. Wille, D. Grosse, L. Teuber, G. W. Dueck, and R. Drechsler, "Revlib: An online resource for reversible functions and reversible circuits," in *Proceedings of International Symposium on Multiple Valued Logic*, 2008, pp. 220–225.

Samah Mohamed Saeed received the BS and MS degrees from the Department of Computer Science, Kuwait University, Kuwait City, Kuwait, in 2008 and 2010, respectively, and the PhD degree from the Department of Computer Science and Engineering, New York University (NYU) Polytechnic School of Engineering, Brooklyn, NY, in 2015. She is currently an assistant professor with the City College of New York, City University of New York. Her current research interests include side-channel analysis, design-for-secure test for VLSI circuits, security of emerging technology, and security of reversible computing. She received the Best Paper Award in VLSI Test Symposium 2011, the Pearl Brownstein Doctoral Research Award by NYU Polytechnic School of Engineering in 2013, and the TTTCs E. J. McCluskey Best Doctoral Thesis Award at the IEEE International Test Conference in 2014

Alwin Zulehner received his BSc and MSc degree in computer science from the Johannes Kepler University Linz, Austria in 2012 and 2015, respectively. He is currently a Ph.D. student at the Institute for Integrated Circuits at the Johannes Kepler University Linz, Austria. He has published several papers on international conferences such as the ASP Design Automation Conference, Design, Automation and Test in Europe, and Reversible Computation. His research interests include design automation for emerging technologies, focusing currently on reversible circuits, quantum circuits, and adiabatic circuits.



Robert Wille (SM'15) received the Diploma and Dr.-Ing. degrees in computer science from the University of Bremen, Bremen, Germany, in 2006 and 2009, respectively. He is now a Full Professor and Head of the Institute for Integrated Circuits at the Johannes Kepler University Linz, Austria. Before, he was with the Group of Computer Architecture at the University of Bremen, Germany, and the German Research Center for Artificial Intelligence. Furthermore, he was a Lecturer with the University of Applied Science Bremen as well as a Visiting Professor at the University of Potsdam and the Technical University Dresden, Germany. His current research interests include the design of circuits and systems for both conventional and emerging technologies, including research in the domain of verification and proof engines as well as the synthesis and optimization of quantum circuits, reversible circuits, optical circuits, and microfluidic biochips. He has published over 200 journal and conference papers in the above areas.

Robert Wille was a recipient of the Best Paper Award from the International Conference on Computer-Aided Design (ICCAD) in 2013 and the Forum on specification and Design Languages (FDL) in 2010 as well as the Young Researchers Award from the International Symposium on Multiple-Valued Logic (ISMVL) in 2008. He serves as Editor e.g. for the Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), was the General Chair for ISMVL and FDL, the PC Chair for ISMVL, FDL, as well as frequent PC Member and the Track Chair for conferences, such as the ASP-DAC, DAC, DATE, and ICCAD.

Rolf Drechsler (F'15) received the Diploma and Dr.Phil.Nat. degrees in computer science from J. W. Goethe University Frankfurt am Main, Frankfurt am Main, Germany, in 1992 and 1995, respectively.

He was with the Institute of Computer Science, Albert-Ludwigs University, Freiburg im Breisgau, Germany, from 1995 to 2000, and with the Corporate Technology Department, Siemens AG, Munich, Germany, from 2000 to 2001. Since 2001, he has been with the University of Bremen, Bremen, Germany, where he is currently a Full Professor and the Head of the Group for Computer Architecture, Institute of Computer Science. In 2011, he became the Director of the Cyber-Physical Systems Group, German Research Center for Artificial Intelligence, Bremen. His current research interests include the development and design of data structures and algorithms with a focus on circuit and system design.

Prof. Drechsler was a recipient of the best paper awards at the Haifa Verification Conference in 2006, the Forum on Specification Design Languages in 2007 and 2010, the IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems in 2010, and the IEEE/ACM International Conference on Computer-Aided Design in 2013. He was a member of Program Committees of numerous conferences including, Design Automation Conference (DAC), International Conference on Computer-Aided Design, Design, Automation and Test in Europe (DATE), Asia and South Pacific Design Automation Conference, Forum on specification and Design Languages, MEMOCODE, and Formal Methods in Computer-Aided Design, the Symposiums Chair ISMVL 1999 and 2014, and the Topic Chair for "Formal Verification" DATE 2004, DATE 2005, DAC 2010, as well as DAC 2011. He is a Co-Founder of the Graduate School of Embedded Systems and the Coordinator of the Graduate School "System Design" funded within the German Excellence Initiative.

Ramesh Karri is a Professor of Electrical and Computer Engineering at Tandon School of Engineering, New York University. He has a Ph.D. in Computer Science and Engineering, from the University of California at San Diego. His research and education activities span hardware cybersecurity including trustworthy ICs, processors and cyberphysical systems; security-aware computer aided design, test, verification, validation and reliability; nano meets security; metrics; benchmarks; hardware cybersecurity competitions; additive manufacturing security.

He has over 200 journal and conference publications including tutorials on Trustworthy Hardware in IEEE Computer (2) and Proceedings of the IEEE (5). His groups work on hardware cybersecurity was nominated for best paper awards (ICCD 2015 and DFTS 2015) and received awards at conferences (ITC 2014, CCS 2013, DFTS 2013 and VLSI Design 2012) and at competitions (ACM Student Research Competition at DAC 2012, ICCAD 2013, DAC 2014, ACM Grand Finals 2013, Kaspersky Challenge and Embedded Security Challenge).

He was the recipient of the Humboldt Fellowship and the National Science Foundation CAREER Award. He is the area director for cyber security of the NY State Center for Advanced Telecommunications Technologies at NYU-Poly; Co-founded the NYU Center for CyberSecurity -CCS (http://cyber.nyu.edu/), co-founded the Trust-Hub (http://trust-hub.org/) and founded and organizes the Embedded Security Challenge, the annual red team blue team event at NYU, (http://www.nyu.edu/csaw2016/csaw-embedded).

He co-founded the IEEE/ACM Symposium on Nanoscale Architectures (NANOARCH). He served as program/general chair of conferences including IEEE International Conference on Computer Design (ICCD), IEEE Symposium on Hardware Oriented Security and Trust (HOST), IEEE Symposium on Defect and Fault Tolerant Nano VLSI Systems (DFTS) NANOARCH, RFIDSEC 2015 and WISEC 2015. He serves on several program committees (DAC, ICCAD, HOST, ITC, VTS, ETS, ICCD, DTIS, WIFS).