Toward Increasing the Difficulty of Reverse Engineering of RSFQ Circuits

Harshit Kumar, Tahereh Jabbari, *Student Member, IEEE*, Gleb Krylov, *Student Member, IEEE*, Kanad Basu, Eby G. Friedman, *Fellow, IEEE*, and Ramesh Karri

***Abstract*—Integrated circuit (IC) camouflaging is a defense to defeat image-based reverse engineering. The security of CMOS ICs has been extensively studied and camouflage techniques have been developed. A camouflaging method is introduced here to protect superconducting electronics, specifically, rapid single flux quantum (RSFQ) technology, from reverse engineering. RSFQ camouflaged units have been developed by applying the structural similarity of RSFQ standard cells. A defense using camouflaged RSFQ cells combined with obfuscating the temporal distribution of inputs to the IC increases the attacker's effort to decamouflage. The approach establishes the complexity class of RSFQ decamouflaging and a model checker is applied to evaluate the strength of the defenses. These techniques have been evaluated on ISCAS'85 combinational benchmarks and the controllers of the OpenSPARC T1 microprocessor. A dummy Josephson junction fabrication process adds two additional mask steps that increase the cost overhead. Camouflaging 100% of the benchmark circuits results in an area and power overhead of almost 40%. In the case of the OpenSPARC processor, the approach requires near-zero area, power, and performance overhead even when 100% of the sensitive parts of the processor are camouflaged.**

***Index Terms*—Camouflaging, hardware security, rapid single flux quantum (RSFQ), reverse engineering, superconducting electronics.**

I. Introduction

With Moore's law faltering, mere scaling of silicon CMOS technology is no longer a viable pathway to high performance computing. Power-thermal dissipation is a limiting factor when determining processor performance [1]. The energy consumed by a modern supercomputer is 4–6 megawatts, sufficient to power 5000 homes in the US. Building a high performance exaflop class supercomputer using current resources would consume an estimated 1.5 gigawatts [2].

Manuscript received September 7, 2018; revised January 18, 2019; accepted January 24, 2019. Date of publication August 3, 2019; date of current version September 4, 2019. This article was recommended by Editor-in-Chief A. Polasek. *(Corresponding author: Harshit Kumar.)*

H. Kumar is with the Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721302, India (e-mail: harshitk@iitkgp.ac.in).

T. Jabbari, G. Krylov, and E. G. Friedman are with the Department of Electrical & Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: tjabbari@ur.rochester.edu; gkrylov@ur.rochester.edu; friedman@ece.rochester.edu).

K. Basu and R. Karri are with Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: kb150@nyu.edu; rkarri@nyu.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASC.2019.2901895

Energy efficient supercomputers based on superconducting electronics (SCE) have been pursued for about six decades [3]. Digital SCE can operate at ultra-high speed (near THz frequencies) [4], while consuming ultra-low power [5]. Popular SCE families include rapid single flux quantum (RSFQ) [6], reciprocal quantum logic [7], energy efficient single flux quantum (ERSFQ) [8], eSFQ [9], and adiabatic quantum flux parametron [10]. Simple RSFQ components can operate at near terahertz frequencies [11]. Larger RSFQ digital systems have been demonstrated to operate at 770 GHz, far beyond the projected capabilities of CMOS [12], [13].

Most SCE families use Josephson junctions (JJs) as the primary building block [6], [12]. JJs exhibit high switching speeds and ultra-low power dissipation. As technology matures, SCE researchers are focusing on designing very large scale integration (VLSI) SCE circuits by developing electronic design automation (EDA) tools to automate the RSFQ circuit design process [14]–[17].

The design–fabrication–test steps in an RSFQ design flow resemble that of CMOS. A mature SCE fabrication process for Nb/Al-AlO_X/Nb JJs, using 200 mm production class tools, has been developed [18]. The similarity to CMOS makes SCE a compelling, low power, and ultra-fast beyond CMOS alternative technology without requiring an overhaul of the CMOS ecosystem.

This similarity exposes SCE circuits to reverse engineering (RE) that is undermining CMOS-based circuits. While considerable research in the security of CMOS circuits exists, the security implications of SCE remain unexplored and is the focus of this study.

A. Contributions

This first-of-kind security assessment of RSFQ circuits makes three contributions.

1) Provides camouflage AND/OR gates and camouflage D flip-flop (DFF) in RSFQ logic to thwart image-based RE. Exploits the structural similarity of RSFQ cells to provide camouflaged cells with low overhead.
2) Exploits the clocked nature of RSFQ logic from a security point of view. The study leverages the synchronous nature of RSFQ to thwart satisfiability (SAT) attacks.
3) Obfuscates inputs to increase the attacker's effort. Camouflages flip-flops to thwart model checking attacks.

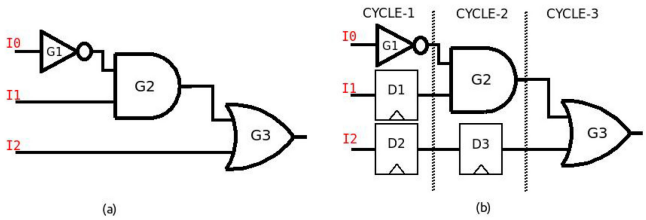

Fig. 1. Three-input circuit: (a) logic, and (b) equivalent RSFQ circuit.

B. Structure

The article is structured as follows. A brief background on both RSFQ logic and IC RE is provided in Section II. IC camouflaging is explained in Section II-D. The use of dummy JJs to create camouflaged gates and camouflaged flip-flops is described in Section III. Inputs are obfuscated to secure camouflaged gates in Section IV. Security analysis and hardware overhead are described in Section VI, and the article is concluded in Section VII.

II. Background

The basics of RSFQ circuit design, hardware security, IC RE, the target RE threat model, and IC camouflaging are explained in this section.

A. Fundamentals of RSFQ Logic

In RSFQ logic, information is conveyed as picosecond duration voltage pulses $V(t)$ [6], [8]. Switching a JJ is accompanied by a 2π phase leap and a voltage pulse equivalent to a quantum of flux ($\phi_\circ = 2.07 \times 10^{-15}$ V·s).

The presence of a flux quantum within a specified interval in time represents a logical "1," otherwise, a "0." RSFQ gates such as a Josephson transmission line (JTL), splitter, and confluence buffer do not require a clock, and the propagation delay is the delay of the output with respect to the input. RSFQ logic gates like NOT, OR, AND, and XOR are clocked [6]. The input pulses to these gates may change the internal state of the gate but do not change the output. The output associated with an input pulse is received when the clock pulse arrives at the logic gate. Hence, the propagation delay is measured as the time elapsed after the clock pulse arrives. RSFQ gates are, therefore, similar to CMOS gates with an edge-triggered flip-flop at the output.

Benchmark circuits, used by researchers as a basis for comparing results in the domain of testing and security, do not exist for the RSFQ logic family. To simulate different attacks on RSFQ circuits, existing CMOS benchmark circuits [19] are mapped onto RSFQ gates. Due to the gate-level pipelining inherent to RSFQ logic, a mapped RSFQ circuit functions correctly when all input pulses arrive during the same clock cycle.

A three-input logic circuit, shown in Fig. 1(a), is mapped to an RSFQ circuit [see Fig. 1(b)]. Path-balancing DFFs, D1, D2, and D3, are added at the input of gates G2 and G3 to equalize the logic depth. Rather than one clock cycle, as depicted in Fig. 1(a), the circuit shown in Fig. 1(b) requires three clock cycles to produce an output. SCE EDA tools have been developed to balance paths to minimize the circuit depth [15]–[20].

B. Hardware Security

An increase in design complexity of modern systems-on-chip (SoC) has resulted in higher manufacturing cost [21]. In an endeavor to regulate expenses, most design houses outsource portions of the design process to third parties. These cases include third party EDA software to third party foundries. Due to the increase in manufacturing cost, even large semiconductor companies are going fabless [22]. Although this distributed design flow reduces expenditure, it introduces security vulnerabilities into the design flow [23].

A malicious employee or a rogue foundry may introduce a hardware Trojan inside a circuit [24]. A Trojan is an external component introduced into a circuit to perform malicious operations like deny service or create a backdoor path. Other vulnerabilities with outsourcing fabrication include stealing of IC masks [25], IC counterfeiting [26], and overproduction of ICs [27]. IC counterfeiting and overproduction are feasible when a malicious entity can reverse engineer an IC.

Counterfeiting can be thwarted in two ways—either by IC camouflaging or logic locking to prevent RE, or by including a watermark to identify counterfeit ICs. The former technique is explained in Section II-D. An intellectual property (IP) watermark consists of a particular signature inside the IP, which is not part of the IP functionality but acts as proof of ownership. IP watermarks can be based on either finite state machines (FSM) or scan chains. FSM-based watermarks exploit undefined states [28] or transitions [29] in the FSM to create a watermark. Alternatively, scan chain watermarks are constructed by reordering the scan chains [30], [31]. Each of these approaches have specific disadvantages. While FSM-based watermarks require large area, scan chain watermarks produce routing congestion.

C. Reverse Engineering

RE is the process of analyzing the functionality, examining the manufacturing process, and extracting gate-level schematics and netlists. Companies use RE to gain a competitive advantage over rivals and to uncover evidence of patent infringement [32]. While RE is legal and accepted by the IC industry as a legitimate form of competition, an attacker may reverse engineer a gate-level netlist and use the netlist in the company's own design, make counterfeit ICs, or sell the IP to a rival. The semiconductor industry suffers losses in many billions of dollars due to IP infringement, most of which is due to RE.

A gate-level netlist can be extracted from a packaged IC as follows [33].

1) *Remove package* using corrosive acids.
2) *Delayer* using recipes for each layer (metal, insulator, semiconductor, or superconductor).
3) *Image* individual layers using SEM or optical instruments.
4) *Align* images, *annotate*, and infer the circuit schematic.
5) *Verify* and *analyze* the extracted netlist.

Since the principles of fabricating RSFQ are similar to CMOS [18], the same RE methodology applies to the RSFQ

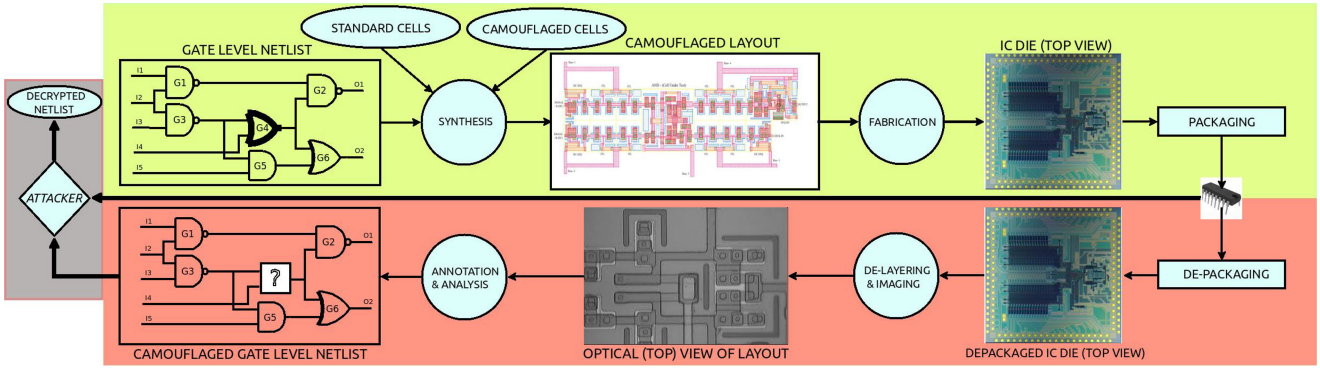

Fig. 2. RSFQ design flow is shown on the top half (shaded green). The bottom half (shaded red) depicts the reverse engineering flow employed by an attacker.

Logic family. RE is a significant threat to RSFQ circuits due to the relatively large size devices and small number of devices per IC, as compared to CMOS. The security of RSFQ circuits should be a concern as SCE is used in large data centers and supercomputers, both of which have national security implications [3].

The global scale semiconductor supply chain is making it easier for IP piracy while making it difficult to detect IP piracy. Hence, approaches are needed to thwart RE as part of the design process. In this article, defensive schemes are described for the emerging RSFQ SCE.

D. Camouflaging Thwarts Reverse Engineering

IC camouflaging thwarts image-based RE by introducing a set of *camouflaged cells* along with the standard cells during the synthesis process. Camouflaged cells appear indistinguishable from the top view of the different layers (encountered while delayering), yet perform different logical operations. For example, a NAND/NOR/XOR camouflaged cell can function as either a NAND, NOR, or XOR gate, depending upon the internal configuration. Merely imaging the layout is not sufficient to decipher the functionality of the camouflaged gates. Multiple strategies for designing camouflaged standard cells have been proposed. These methods include dummy contacts [34], dummy filler cells [35], programmable standard cells [36], or an amalgam of these approaches.

A list of terms used here is described in Table I. The RSFQ design flow with camouflaging (shaded green) is shown in Fig. 2. The NOR gate G4 in the original netlist is replaced with a camouflaged cell. The attacker procures the IC through spurious means or from the market (if available) and reverse engineers the IC to determine the gate-level netlist (shaded red). The attacker fails to determine the function of the camouflaged gate G4 solely by examining the circuit layout. To uncover the function of the camouflaged gates, the attacker can use the functional IC as a black box to determine the oracle (input–output pairs). With this oracle, the camouflaged netlist can be probed, allowing the function to be deciphered [37], [38].

The next example illustrates an oracle-based attack strategy [37]. The camouflaged netlist of a simple circuit is shown in Fig. 3(a). The function of the camouflaged gates, G2 and G3,

TABLE I
GLOSSARY OF TERMS

Keyword	*Description*
Functional IC	A fully functioning IC obtained after packaging (see Fig. 2).
Black box	Use of a functional IC to feed the input pattern to produce the corresponding output pattern.
Oracle	Input-output pairs obtained by querying the black box.
Unauthorized access	Unapproved access by a rogue agent in the supply chain.
Camouflage netlist	Netlist camouflaged by the defender.
Decamouflaging	Uncover function of camouflaged cells in a netlist.
Distinguishing input pattern	Set of inputs that can decamouflage an IC.
Assignment	Provisional assignment of a function to a camouflaged gate by attacker.
Completion	Confirmed assignment of function to each camouflaged gates.

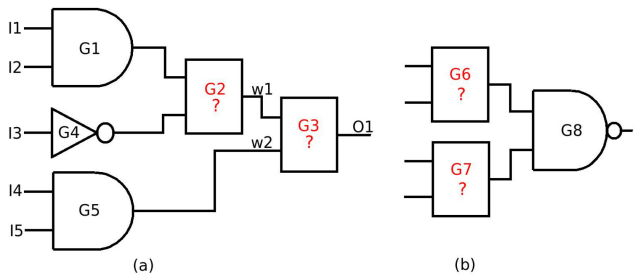

Fig. 3. Overview of IC camouflaging: (a) G2 and G3 are resolvable and (b) G6 and G7 are not resolvable [37].

is unknown to the attacker. The attacker exploits the following VLSI testing principles to decipher the functions:

1) *Justification:* The output of the camouflaged gate is justified to a known value by controlling the inputs. For example, the output of the NAND/NOR/XOR camouflaged gate is 0, irrespective of the functionality, NAND, NOR, or XOR, if both inputs are 1.
2) *Sensitization:* The output of a camouflaged gate is made observable at a primary output by placing non-controlling values in the remaining gates.

Consider the circuit shown in Fig. 3(a). The attacker can force the input of gate G3 to $w1 = 0$ and $w2 = 0$ by applying the input pattern 11 000 to resolve the function of G3 as XOR. If XOR

is eliminated, the attacker can justify the input of gate G3 to w1 = 0 and w2 = 1 by applying the input pattern 11011 to resolve the functionality of G3 to be either NAND or NOR. Once the function of G3 is known, the attacker can set non-controlling values to bypass G3 and decipher G2 in a similar manner.

The sensitization and justification conditions for the example circuit shown in Fig. 3(b) cannot be simultaneously satisfied. Consider justifying the output of either gates G6 and G7 to 0. This method prevents sensitization of the output of the gate that is not justified. This behavior occurs because 0 is a controlling value of G8 and sets the output to 1.

Similar attacks apply to RSFQ circuits with changes to accommodate a pipelined structure. The attacker waits for N clock cycles to receive the output from the black box; N is the number of pipeline stages in the circuit. N can be determined by analyzing the camouflaged gate-level netlist assuming that all gates introduce a single cycle delay. Similar to CMOS, camouflaging random gates does not secure an RSFQ circuit.

E. Threat Model

A GDSII file includes all of the masks, therefore, IC camouflaging assumes that the foundry is trustworthy and cooperates when fabricating ICs with camouflaged cells. Alternatively, post-fabrication entities in the supply chain are untrusted [39]. Since a fabricated IC is a tangible component in the supply chain, the attacker can be a rogue within the supply chain who can obtain an IC. Moreover, the attacker can procure an IC through illicit channels while or after the IC is deployed in the field. This threat model is consistent with the CMOS RE threat model, when the attacker has access to a functional IC. From the attacker's perspective, the assumptions are as follows:

1) the attacker has access to RE facilities allowing the steps listed in Section II-C to be performed;
2) the attacker can identify the camouflaged cell from a standard cell. The image of standard and camouflaged cells is assumed to be publicly available. While the attacker can differentiate a camouflaged cell from a standard cell, the function implemented by the camouflaged cell using image-based RE cannot be deciphered;
3) as a worst case scenario, it is assumed that the attacker can deduce the possible list of functions implemented by the camouflaged cells. For example, a camouflaged NAND/NOR/XOR cell can implement either a NAND, NOR, or XOR function. The attacker, however, cannot determine the exact function of the camouflaged cell using image-based RE.

III. Design of RSFQ Camouflaging Cells

In this section, dummy JJs, camouflaged RSFQ AND/OR gates, and camouflaged RSFQ flip-flops are discussed. The structure of dummy JJs to thwart RE is introduced in Section III-A. Considering a dummy JJ, RSFQ AND/OR gates are described in Section III-B. The function of the dummy DFF as a JTL is introduced and discussed in Section III-C.

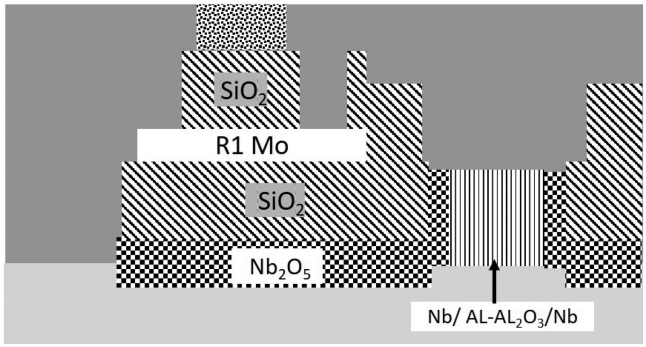

Fig. 4. Cross section of a normal and dummy JJ with a shunt resistor [40].

A. Dummy Josephson Junction

A dummy JJ never switches into the superconducting state and always behaves as a resistor. JJs are fabricated as a sequence of Nb–AlO_X–Nb layers where the AlO_X layer is the insulator [40]. The critical current density of a JJ depends upon the thickness of the AlO_X tunneling barrier [14]. Changing the thickness of the insulating layer and the quality of the superconducting material affects the switching characteristics of a JJ. Two approaches to fabricate a dummy JJ are considered. These approaches increase the cost by using two additional mask steps.

1) Method 1 — Vary Insulator Thickness of the JJ: The critical current density and thickness of the insulation layer depend upon the RSFQ fabrication technology and the physical design rules. The thickness of AlO_X is currently about 1 nm in a standard JJ technology [40]. By increasing the thickness of AlO_X beyond the ~38 nm coherence length of the Nb layer, a dummy JJ can be fabricated that always behaves as a resistor [41]. The magnitude of the resistance depends upon the insulator thickness.

While the reverse engineer can differentiate between a true JJ and a dummy JJ by slicing the die and imaging a side view, this strategy does not scale due to the large number of JJs in a typical RSFQ circuit [33], [37]. Hence, slicing an IC to decipher the function of every JJ is extremely challenging. Alternatively, a top view image of a dummy JJ is identical to a standard JJ. Hence, it is difficult to distinguish between two JJs using image-based RE.

To tune the McCumber damping parameter, most of the JJs in current fabrication processes are shunted with a resistance [42]. A cross section of a JJ with a shunt resistor is shown in Fig. 4. The structure is composed of two Nb layers, a stack of Nb–Al–Al_2O_3–Nb for the JJ, a Mo layer for the shunt resistors, and Nb_2O_5 and SiO_2 for the isolation layers [40]. A thicker insulator film yields a dummy JJ that cancels the superconducting current. The minimum thickness of Al–Al_2O_3 in a dummy JJ is 40 nm.

Dummy JJs are shunted with a resistor to appear identical to a normal JJ. A small shunt resistor with a dummy JJ can degrade the camouflaged cell. Decreasing the thickness of the Mo layer increases the shunt resistance and prevents deterioration of the camouflaged cell. Two approaches exist to tune the thickness of a JJ, either by addition or by elimination.

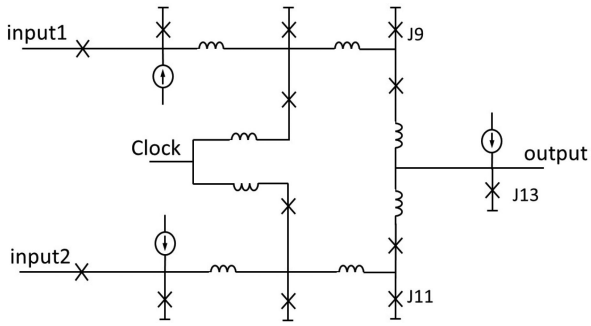

Fig. 5. Camouflaged RSFQ AND–OR cell; J9 and J11 are dummy JJs for the AND gate, and J13 is a dummy JJ for the OR gate.

a) Double deposition process: It can be used to fabricate JJs to achieve the proper critical current density for a normal JJ and to maintain the resistive behavior for a dummy JJ. The initial deposition process determines the critical current of a normal JJ. A thicker deposition layer can be used for a dummy JJ based on the coherence length. As compared to normal junctions, a dummy JJ increases the fabrication time by adding several fabrication steps. As compared to other methods to fabricate a dummy JJ, a double deposition process offers benefits that include an accurate thickness for the normal and dummy JJs and a shorter fabrication time.

b) Ion beam etching: A thick AlO_X layer is deposited for the dummy JJs. This step is followed by an ion beam etch to fabricate a normal JJ. The etching time, surface roughness, and insulator depth determine the switching characteristics of the JJ.

2) Method 2 — Damage the Nb Layer: In this method, the Nb layer is bombarded with an ion beam to damage the surface. The ion beam smooths the surface depending upon the energy, temperature, and angle of the beam [43], [44]. The properties of the ions used to bombard the surface are enhanced depending upon the thickness and material of the film affected by the ion beam. Various materials have different effects on the Nb layer. To remove an undesirable surface, an Ar or He ion beam is used. Dummy JJs, fabricated using an ion beam, have a Nb layer thickness identical to a normal JJ. Furthermore, bombarding the top Nb layer with carbon ions alters the superconductive properties (e.g., eliminates the superconducting current due to the large impurity concentration within the Nb). Based on the suggested methods, a normal JJ and dummy JJ can be separately fabricated with different and specific parameters. A dummy JJ can be included in a fabricated RSFQ circuit to thwart RE.

B. Camouflaged RSFQ AND/OR Cell

Dummy JJs are used to design camouflaged AND/OR RSFQ gates to behave as either a two-input AND or OR gate. The camouflaged AND/OR gate is based on the structural similarities of the AND and OR gates to ensure low overhead. A schematic of a camouflaged AND/OR gate with dummy JJs is shown in Fig. 5. For an AND gate, J9 and J11 are dummy JJs. Similarly, J13 is a dummy JJ for an OR gate. For simulation and analysis purposes, the dummy JJ is modeled as a large resistor in parallel with a small shunt resistor (a normal shunt resistor is 2 to 5 ohms). The high operating speed of the camouflaged cell is retained by reducing the thickness of the shunt resistor in the dummy JJ. The resistance of the shunted dummy JJ is 24 ohms. A simulation of the camouflaged AND/OR gate is shown in Fig. 6 characterizing the AND function, and in Fig. 7 characterizing the OR function. The camouflaged AND/OR cell layout is shown in Fig. 8. The layout follows 4.5 kA/cm^2 Hypres RSFQ design rules [45].

The output delay, power, and area depend upon the number of dummy JJs. Due to the small shunt resistor in a dummy JJ, the current passing through the JJ after each input pulse is significant. Hence, the power and output delay of a camouflaged gate can be reduced by optimizing the shunt resistance.

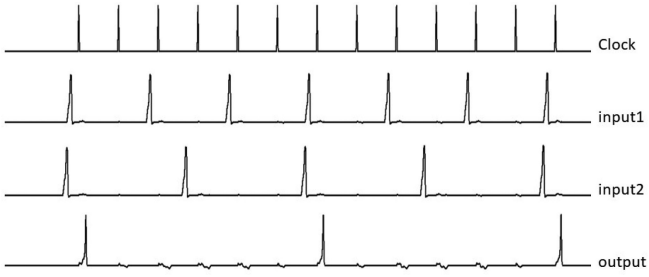

Fig. 6. AND function; J9 and J11 are dummy JJs.

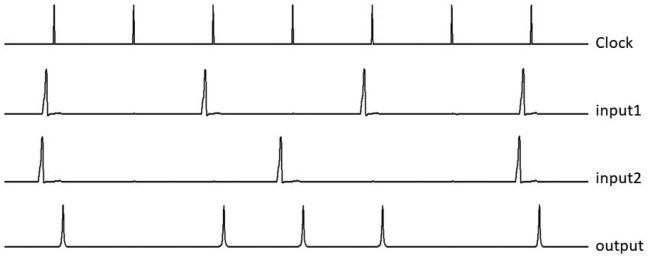

Fig. 7. OR function; J13 is a dummy JJ.

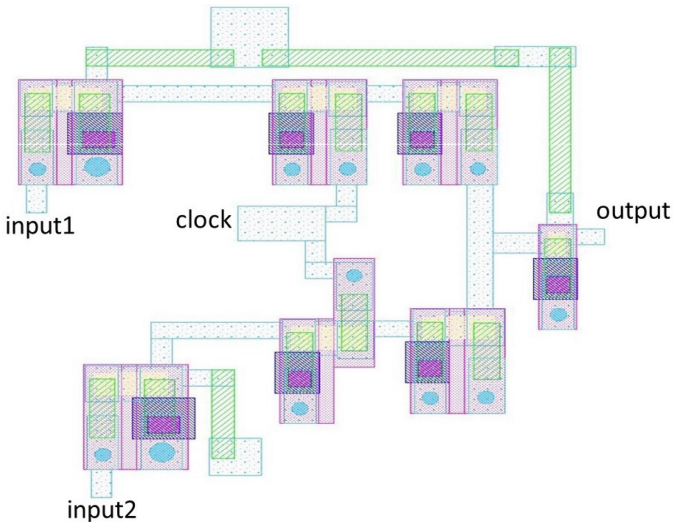

Fig. 8. Layout of a camouflaged RSFQ AND/OR cell.

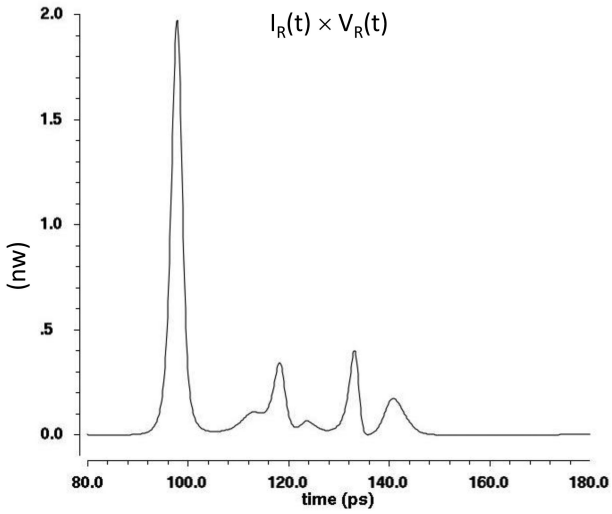

Fig. 9. Power dissipation of a dummy JJ.

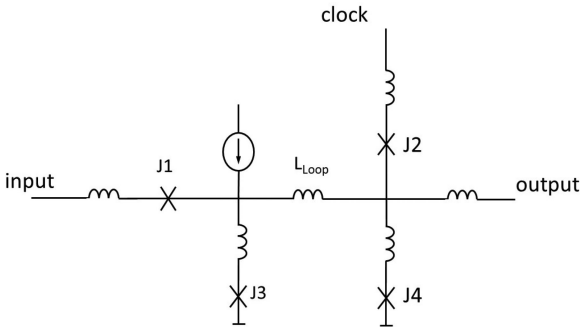

Fig. 10. Camouflaged RSFQ DFF.

The energy dissipated by the dummy JJs in the camouflaged AND/OR is shown in Fig. 9. Averaging the energy over one clock cycle provides a power overhead of 100 pW for an AND gate with two dummy JJs, and 30 pW for an OR gate with one dummy JJ for an operating frequency of 10 GHz. The energy dissipated is approximately 2% to 5% higher than a standard RSFQ OR and AND gate. The output delay of the camouflaged AND/OR gates is 11 ps as compared to a delay of 10 ps for a standard AND and 7 ps for a standard OR gate. As compared to standard AND and OR gates, the area overhead of the camouflaged AND and OR gates is, respectively, approximately 15% and 10%. Since the energy dissipation is higher for camouflaged gates when compared to standard RSFQ gates, one might wonder whether a side-channel attack [46] can distinguish the two logic topologies. The energy dissipation due to JJ switching is low ($\sim 10^{-19}$ J) and, hence, power side channel attacks may be infeasible.

C. Camouflaged RSFQ D Flip-Flop

A camouflaged RSFQ DFF can function as a JTL or as a standard DFF. A schematic of a camouflaged DFF is shown in Fig. 10 which is the same as a standard DFF. In the camouflaged cell, J4 is a dummy JJ and behaves as a resistor. By adjusting the thickness of the insulating layer, the resistance is increased to lower the output delay and power. In a standard DFF, L_{Loop} is large, storing the information bit while the content is read. To achieve the same layout, the length of the inductor in the camouflaged DFF is maintained the same as a regular DFF. Consequently, the large kinetic inductance in the camouflaged DFF produces a large output delay when functioning as a JTL. To circumvent this effect, the inductance is reduced to decrease the delay. This smaller inductance can be achieved by increasing the thickness of the kinetic inductance, resulting in a decrease in the inductance [47]–[49]. Since a JTL is asynchronous and does not require a clock, the effect of the clock signal is eliminated by reducing the critical current through J2 by increasing the thickness of the insulator layer. A simulation of the camouflaged DFF functioning as a JTL is shown in Fig. 11. By changing the thickness of the Nb, Mo, and insulator layers to a standard thickness, the functionality of a standard DFF can be achieved.

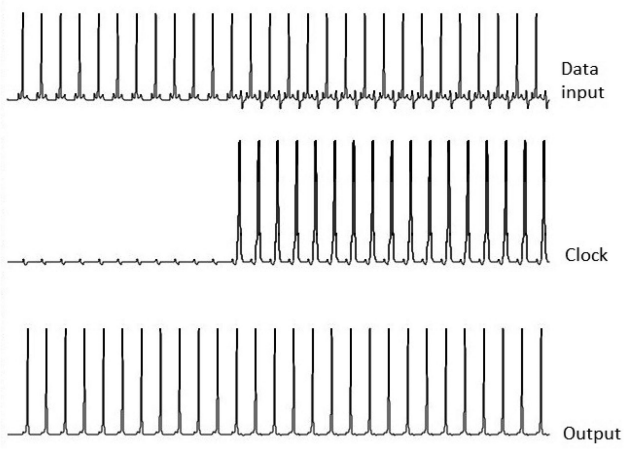

Fig. 11. Camouflaged DFF operating as a JTL. The JTL passes the input pulses regardless of the clock.

The output delay of a camouflaged DFF is approximately 11 ps which is roughly twice the delay of a standard JTL. This difference is attributed to the large inductance and two different input pulses—the clock and data signals. The throughput of the circuit is halved when the camouflaged DFF is part of the critical path. The current through the dummy JJ varies depending upon the clock frequency and the input signal in the camouflaged DFF. By assuming a frequency of 10 GHz for the input and clock pulses, the total power dissipated by the dummy JJ is approximately 100 pW. The energy dissipation of the camouflaged DFF is approximately the same as a standard DFF due to the different critical current of the JJs. The energy dissipated by a camouflaged DFF is approximately 2% more than a standard JTL. The camouflaged DFF has an identical top view and a different thickness for J2 and dummy J4. A camouflaged DFF therefore exhibits the same area as a standard DFF. Furthermore, the area of a camouflaged DFF is approximately twice as large as a standard JTL.

IV. Camouflaging RSFQ Circuits

Since camouflaging thwarts image-based RE, the attacker may resort to alternative means to decipher the function of the camouflaged gates. Query-based attacks such as brute force, test based [37], and SAT based [38] are popular. Brute force attacks, used as a last resort, become infeasible by increasing the number of camouflaged gates. Test-based attacks can be thwarted by selecting the location of the camouflaged gates to infer with justification and sensitization.

A SAT-based attack on a camouflaged IC uses a satisfiability solver to determine a set of discriminating input patterns (DIPs). By applying this attack to a functional IC, the function of the camouflaged gates is revealed [38]. Application of the inputs such as querying an IC to fetch the output bits requires significant time. An attacker's objective is, therefore, to minimize the size of the DIPs. Due to the small size of the set of DIPs, SAT-based attacks can decamouflage large benchmark circuits within minutes. Since these techniques are oracle guided, they require access to a functional IC which can be queried as a black box. Directly probing an IC is infeasible even for moderately sized circuits. Hence, the attacker uses a functional IC as a black box.

The objective of a defender is to increase the attacker's effort by increasing the size of the discriminating sets. This result is achieved by camouflaging a large number of gates. However, camouflaged cells exhibit additional area, power, and delay overhead as compared to standard cells. A tradeoff therefore exists between cost and security.

A. Camouflaging CMOS and RSFQ: Is It Really Different?

Given a combinational CMOS logic, the equivalent RSFQ Logic is sequential due to the clocked nature of RSFQ. Although clocked, the combinational behavior is retained by applying path-balancing flip-flops. These flip-flops plus the built-in latches in the gates transform the data paths into a multicycle pipeline with the same function. The RSFQ circuit shown in Fig. 1(b) behaves similarly to the CMOS circuit, except that the output appears after N clock cycles, where N is the depth of the balanced circuit ($N = 3$ in this case).

Hence, an equivalent netlist C_{combo} with a single cycle data path can be constructed from an RSFQ circuit by eliminating the path-balancing flip-flops and the built-in latches of the RSFQ gates. C_{combo} can be conceptualized as a technology independent netlist for digital logic. Hence, similar to CMOS circuits, RSFQ circuits are vulnerable to decamouflaging.

Decamouflaging sequential circuits requires the following alterations. A sequential circuit can be converted into a combinational circuit in one of two ways:

1) Access the internal state of the flip-flops via a scan chain. The flip-flops store the information of the respective cone of dependencies consisting of the standard combinational gates, thereby partitioning the circuit into smaller combinational modules. SAT solvers have been successfully employed on these circuits [38].
2) Eliminate the sequential nature by unrolling the circuit into the temporal domain. Bounded model checking can be used for this method [50]. Unrolling a circuit is computationally infeasible if the number of internal states relative to the number of inputs and outputs is large [50].

Modern ICs are fortified against the former method by blocking unauthorized access to scan chains [51]. Since established test structures do not exist in RSFQ, RSFQ test point and scan chain structures [52] can be similarly encrypted.

TABLE II
Temporal Distribution of Inputs in Fig. 1(b)

Input	t=0	t=1	t=2
I0	IN[0]	0	0
I1	0	IN[1]	0
I2	0	0	IN[2]

B. Does Removing Path-Balancing Flip-Flops Secure an IC?

Removing path-balancing flip-flops in a mapped RSFQ circuit reverts the function back to a sequential nature and loses functionality. This method can be managed by adding clock delays while feeding the input to the circuit.

Example: In Fig. 1(b), removing the path-balancing flip-flops, D1, D2, and D3, requires applying input IN[0:2] in a specific order, as listed in Table II, to maintain correct circuit function. As a result, the circuit becomes sequential with a large number of unknown internal states (due to the clocked nature of each gate) which makes unrolling impractical.

From an initial assessment, hiding the temporal distribution of the inputs from an adversary will secure an IC. However, every combinational RSFQ circuit has a unique mapping of path-balancing flip-flops which can be determined by examining the camouflaged layout, using the principle "*all inputs to a gate should arrive within the same clock cycle*." The attacker can decipher the temporal distribution of inputs to determine the circuit function.

C. Temporal Obfuscation of Inputs

The evolution of the strategy in Section IV-B is explained in this section. The camouflaged DFF described in Section III-C is used to hide the input timing. As explained in Section III-C, a camouflaged DFF functions similar to a JTL, introducing zero clock cycle delay, or a true DFF, which introduces a single clock cycle delay.

Consider the function of the circuit shown in Fig. 12(a). R1 to R6 depict the functional representation of the standard RSFQ gate, G1 to G6, with the respective built-in flip-flops, D1 to D6. The camouflaged flip-flop array consists of three stages of camouflaged DFF, (J1 to J3, F1 to F3, M1 to M3, H1 to H3, and K1 to K3). Let $|Camo\ stages|$ be the number of stages of camouflaged DFFs in the camouflaged flip-flop array: $|Camo\ stages| = 3$ in the current example. Each input port (I0 to I4) can exhibit a delay ranging from zero to three cycles due to the camouflaged flip-flop array. As a result, the total number of clock cycle delay combinations at the input is ζ, where

$$\zeta = (|Camo\ stages| + 1)^{|\text{Primary inputs}|}. \quad (1)$$

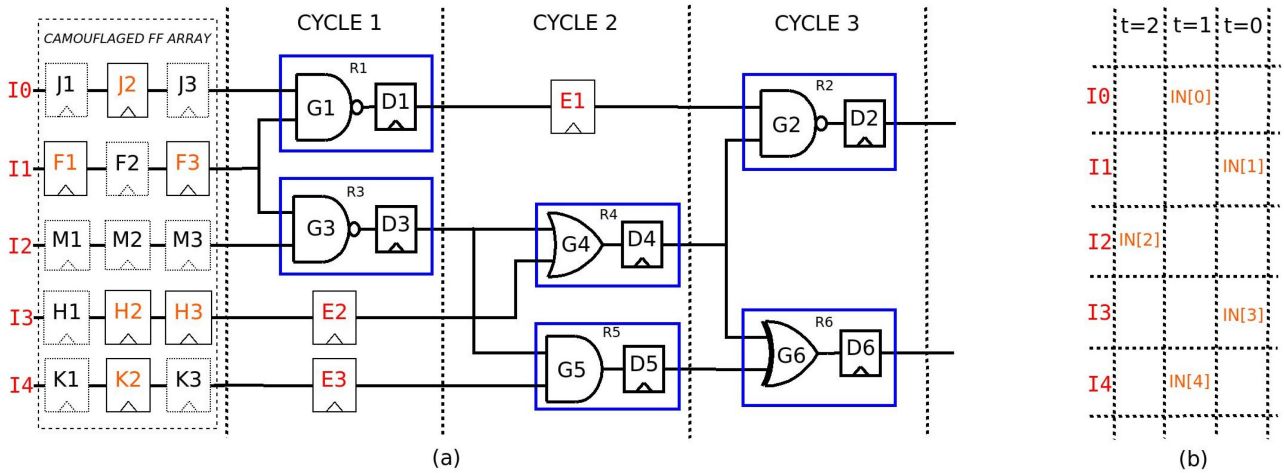

Fig. 12. Proposed defense using camouflaged gates: (a) camouflaged flip-flop array and (b) temporal distribution of inputs for the specific assignment to the camouflaged flip-flop array.

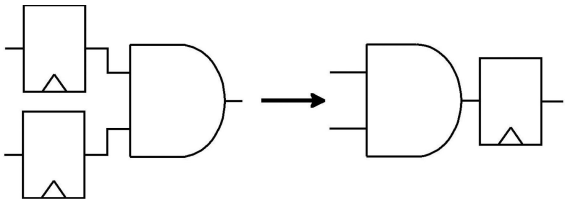

Fig. 13. Bubble pushing of flip-flops.

Thus, 4^5 input combinations exist for the example shown in Fig. 12. One assignment to this array is considered wherein the camouflaged DFFs, shown in orange (J2, F1, F3, H2, H3, K2), function as a true DFF, introducing a single clock delay. The remaining camouflaged DFFs function as a JTL passing the SFQ pulse to the output. For this circuit to function as intended, the inputs are temporally arranged, as shown in Fig. 12(b).

The designer hides this unique temporal distribution of the inputs from the attacker. Without this information, the attacker cannot deduce the function of the camouflaged netlist due to the exponential number of possible input combinations introduced by the camouflaged flip-flop array.

D. Sustainability Against Removal Attack

If the entire camouflaged flip-flop array is intact, an attacker can isolate the chain of camouflaged DFFs to bypass the defense. To prevent this attack, a concept similar to bubble pushing can be used [53]. As illustrated in Fig. 13, a unit cycle delay in all of the inputs of a logic gate is functionally equivalent to a unit cycle delay at the output. The camouflaged DFFs can be pushed to different pipeline levels within the circuit, creating a uniform distribution of camouflaged DFFs.

E. Output Corruption

One of the basic requirements of a robust camouflaging technique is to ensure that incorrect assignments to the camouflaged netlist produces incorrect outputs. The objective is to achieve

Algorithm 1: Identification of Camouflaged Gate Locations Ensuring Maximum Output Corruptibility.

Input : Netlist, $\#Camo\_FF\_stages$, $\#gates\_to\_camo$
Output : Camouflaged netlist
Calculate output corruptibility for all gates;
$Camo\_Location$ = Gates with highest output corruptibility;
/*CBS()– clique-based strategy [37]*/
Netlist $AND\_OR\_camouflaged \leftarrow$ CBS (netlist, $Camo\_Location$, $\#gates\_to\_camo$);
for $i \leftarrow 1$ *to* $\#Camo\_FF\_stages$ **do**
  Insert a FF at all inputs;
while $True$ **do**
  for $GATE$ *in* AND_OR_camo **do**
    if *all inputs of* $GATE$ *have camouflaged DFF with same function* **then**
      if *input of* $GATE \notin Camo\_Location$ **then**
        Bubble push camouflaged DFF to output of $GATE$;
      else
        pop input of $GATE$ in $Camo\_Location$;
  if *Number of Bubble push* == *0* **then**
    break;
Camouflaged netlist $\leftarrow AND\_OR\_camo$;

50% output corruption for minimum correlation. The following principles from VLSI testing are used [54]:

1) *Excitation* of either a stuck-at-1 or stuck-at-0 fault is analogous to an incorrect assignment of a camouflaged gate (camouflaged DFF or camouflaged AND/OR gate).
2) *Propagation* of the activated faults to the output ensures that an incorrect assignment necessarily sensitizes the output, thereby corrupting the signals.

Based on a fault impact-based metric [55], those locations where a fault is inserted propagates to the output, allowing corrupt output bits to be identified. From these locations, gates or flip-flops are selected to camouflage using Algorithm 1.

V. Security Analysis of RSFQ Camouflaging

A strong camouflaging scheme should be resilient against all types of decamouflaging and offer high output corruption for incorrect assignment to camouflaged gates.

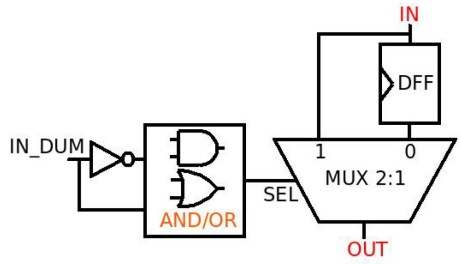

Fig. 14. $dum_{\mathbf{SAT}}$ which replicates the function of a dummy flip-flop in $C_{\mathbf{SAT}}$.

A. Security: A Complexity Theoretic Perspective

The attacker obtains two copies of an IC. The first IC $\bar{C}$ is used as a black box to produce the oracle. The second IC is reverse engineered to obtain the gate-level netlist C_{camo}. The attacker deciphers the function of the camouflaged gates in C_{camo} by using the oracle obtained from querying the black box $\bar{C}$. The problem is formalized as follows:

Problem 1. Let n be the number of primary inputs, o be the number of primary outputs, k_{camo} be the number of AND/OR camouflaged gates, and k_{dum} be the number of camouflaged DFFs in C_{camo} (and $\bar{C}$). A completion X is the assignment of boolean identities to all of the camouflaged gates of C_{camo}, and is denoted by C_{camo}^{X}. The attacker's objective is to determine the completion X^* which assigns correct boolean functionality to all of the camouflaged gates, satisfying the following relation:

$$C_{\text{camo}}^{X^*}(I) = \bar{C}(I) \quad \forall I \in \mathcal{I} \tag{2}$$

where $\mathcal{I}$ is the universal set of input sequences. Equation (3) implies that a completion is correct if the output of the completed circuit agrees with the output of the black box for all possible input sequences.

Complexity of Problem 1 : Given a set of input sequences $\mathbf{I}$, the certificate (guess) for the stated problem in (3) is a completion X such that C_{camo}^{X} concurs with $\bar{C}$ on all input sequences in $\mathbf{I}$. The maximum number of camouflaged gates depends upon the circuit C_{camo} (or $\bar{C}$) which prompts the linear dependency of X on the size of C_{camo}. The verification of the given certificate can be accomplished in $O(|C_{\text{camo}}||\mathbf{I}|)$ which is polynomial in $|\mathbf{I}|$. Moreover, given an input sequence $i \in \mathbf{I}$, the corresponding output can be obtained in a time polynomial in the size of C_{camo}. Hence, this problem is NP and can be solved using the model checker based attack which tackles problems of the same complexity class.

The camouflaged netlist C_{camo} is modified to simulate a camouflaged DFF. Each camouflaged DFF in C_{camo} is replaced with the structure shown in Fig. 14. The structure, $cam_{\mathbf{SAT}}$, models the function of the camouflaged DFF in a circuit using AND/OR camouflaging. Depending upon whether the camouflaged gate (AND/OR) performs the function AND or OR, the select line (SEL) of MUX is 0 or 1, introducing a delay of one clock cycle between the input (IN) and output (OUT) or no delay at all. The boolean circuit resulting from the reduction of C_{camo} is $C_{\mathbf{SAT}}$, where all camouflaged DFFs are replaced by the corresponding $cam_{\mathbf{SAT}}$. $C_{\mathbf{SAT}}$ has $k_{\text{total}} (= k_{\text{camo}} + k_{\text{dum}})$ number of AND/OR camouflaged gates, $n + k_{\text{dum}}$ is the number of primary inputs, and o is the number of primary outputs. The model checking framework is used to launch a decamouflaging attack on $C_{\mathbf{SAT}}$ [50].

TABLE III
BENCHMARK CHARACTERISTICS (ITALICS: *ISCAS'85 Benchmarks* [19]; BOLD: *OpenSPARC Controllers* [56])

Benchmark	#PI	#PO	#Gates
c880	60	26	383
c2670	233	140	1193
c3540	50	22	1669
c5315	178	123	2307
c6288	32	32	2406
c7552	207	108	3512
Fpu input	223	176	916
Fpu divider	194	180	1197
Store buffer	170	153	1360
IFQ	227	226	2027
TLU	490	400	2724

B. Benchmarks and Experimental Setup

ISCAS'85 combinational benchmarks [19] and OpenSPARC T1 microprocessor controllers [56] are camouflaged using the proposed approach. Due to insufficient candidate gates for camouflaging, the smaller benchmark circuits and controllers are excluded from this analysis. The number of gates and the primary inputs (PI) and outputs (PO) of the benchmarks are listed in Table III. Approximately 200 lines of Python code convert the CMOS combinational circuits to RSFQ circuits using path-balancing flip-flops. The attack framework is in C++ in ~700 lines of code using NuSMV [57] as the back-end model checker. The corresponding $C_{\mathbf{SAT}}$ formulation is based on the principles described in Section V-A. The AND-OR and DFF-JTL cells are inserted by Algorithm 1. All experiments are performed on an Intel i7-4510U processor. The bounded model checking-based decamouflaging attacks are launched on these netlists.

C. Security Analysis: Resilience to Practical Attacks

This defense offers resilience against two major decamouflaging attacks: test-based and SAT-based. A test-based attack sensitizes the output of the camouflaged gate, as illustrated in the example described in Section II-D. This attack can, however, be thwarted when a clique-based selection (CBS) strategy is used [37]. CBS ensures that the camouflaged gates interfere with each other, preventing simultaneous sensitization and justification. CBS can be coupled with output corruption-based selection (OCS) (see Section IV-E) to provide a defense against a test-based attack and ensure high output corruption.

An attacker can decamouflage a netlist protected by CBS + OCS with a SAT solver. SAT resilient schemes exploiting the clocked nature of RSFQ logic do not exist. Temporal obfuscation of the inputs (see Section IV) reverts the multicycle pipelined structure of the combinational logic into a clocked system, preventing the attacker from creating an equivalent C_{combo}, as described in Section IV-A.

TABLE IV
RESULTS OF THE DECAMOUFLAGING (■: SUCCESSFUL; UNSUCCESSFUL)

#camouflaged AND/OR gates	32			64			128		
Benchmark	*#camouflaged DFF stages*								
	0	*1*	*2*	*0*	*1*	*2*	*0*	*1*	*2*
c880	■	■	■	■	■	■	■	■	■
c2670	■	■	×	■	×	×	■	×	×
c3540	■	■	×	■	■	×	■	×	×
c5315	■	×	×	■	×	×	■	×	×
c6288	■	■	×	■	×	×	■	×	×
c7552	■	×	×	■	×	×	■	×	×
FPU Input	■	■	■	■	■	■	■	■	■
FPU Divider	■	■	×	■	×	×	■	×	×
Store buffer	■	×	×	■	×	×	■	×	×
IFQ	■	×	×	■	×	×	■	×	×
ILU	■	×	×	■	×	×	■	×	×

Decamouflaging sequential circuits using a SAT solver requires controllability and observability of the internal states within a circuit. The scan chains are design-for-test structures which provide the ability to set and observe each on-chip flip-flop. Modern ICs, however, do not provide unauthorized access to scan chains; the attacker is therefore left with the sole option of "unrolling" the circuit using a concept similar to time frame expansion [50]. This action can be achieved by using a model checker which performs a reachability analysis based on SAT. Bounded model checking [50] discovers a discriminating set of input sequences which are sufficient to determine the functionality of the camouflaged gates within a sequential circuit without scan access.

Inserting one stage of camouflaged DFF requires a camouflaged DFF to be added to every primary input within the circuit. These camouflaged DFFs are "bubble pushed" to those locations which offer maximum output corruption, thwarting removal attacks. The results of decamouflaging are summarized in Table IV.

1) *Case 0. The number of camouflaged DFF stages = 0:* No camouflaged DFFs are introduced into the circuit. Only AND-OR camouflaging is performed using the CBS + OCS strategy. The model checker decamouflages all circuits, confirming the weakness of CBS.
2) *Case 1. The number of camouflaged DFF stages = 1:* One stage of camouflaged DFFs is introduced along with camouflaged AND/OR gates. Decamouflaging fails in some circuits due to the large number of circuit inputs (i.e., a large number of camouflaged DFFs).
3) *Case 2. The number of camouflaged DFF stages = 2:* Two stages of camouflaged DFFs are introduced along with AND-OR camouflaging. The attack is unable to decamouflage all of the circuits except c880 and FPU Input controller. Successful decamouflaging in these circuits is due to the small size. On benchmark circuits such as c5315, c6288, c7522, and the larger OpenSPARC controllers, the attack crashes due to the inability of NuSMV to support large model checking instances. This behavior demonstrates the infeasibility of unrolling introduced with this defense.

On increasing the number of camouflaged DFF stages to three, the defense can thwart decamouflaging on the FPU Input controller. Due to the small size of c880, seven stages are required to thwart decamouflaging.

TABLE V
OVERHEAD OF CAMOUFLAGED CELLS AS COMPARED TO BASELINE STANDARD CELLS

Function	Camouflaged DFF			Function	Camouflaged AND/OR		
	Power	Delay	Area		Power	Delay	Area
DFF	0%	0%	0%	AND	2-5%	10%	15%
JTL	2%	100%	100%	OR	2-5%	57%	10%

VI. OVERHEAD OF RSFQ CAMOUFLAGING

The overhead of the camouflaged cells is summarized in Table V. CMOS tools are used to synthesize the benchmarks. Selected gates are replaced by camouflage gates and dummy DFFs are inserted at the inputs. The area, power, and delay overhead of the camouflaged circuits is listed.

The area and power overhead for the benchmark circuits is shown in Fig. 15. The NAND and NOR gates generated by the CMOS synthesis tools are replaced with AND and OR plus NOT gates (e.g., NAND = AND + NOT). This approach camouflages the maximum number of gates, offering maximum security. Three stages of camouflaged DFF are introduced for each benchmark circuit. For circuits of comparable size, the circuit with a greater number of primary inputs has more overhead, e.g., C6288 and TLU. Camouflaging all of the gates incurs an overhead of about 50%. This approach contrasts with the large overhead associated with CMOS camouflaging; camouflaging 5% of CMOS gates entails over 100% overhead.

The area overhead of camouflaging flip-flop stages is shown in Fig. 16. For a small circuit with a large number of inputs (FPU Input), the overhead rises steeply as compared with a circuit with a small number of inputs (c6288). Although the overhead is linear in the number of camouflaged DFF stages, the search space for input patterns is exponential.

Since RSFQ circuits are gate-level pipelined, the throughput of the system depends upon the maximum delay of the intermediate cells. While the camouflaged DFF and camouflaged AND/OR cell incur a delay overhead of, respectively, 100% and 50%, the delay is less than with complex standard cells. The throughput of the circuit is halved in the worst case if the camouflage cells are part of a critical path. *Camouflaging the OpenSPARC T1 processor:* Modern ICs are composed of two types of logic—a data path and a controller. The controller determines the sequence of operations executed during each clock cycle, and hence realizes the secret IP of the circuit. Moreover, since the controllers occupy less than 1% of the total design area, the overhead to camouflage a controller is negligible [37].

Consider the OpenSPARC T1 processor shown in Fig. 17. While the T1 processor has a six-stage data path, the controllers listed in Table VI orchestrate the flow of operations. While the controllers of the OpenSPARC T1 processor require 1.2×10^4 gates, the processor has 11×10^6 gates, accounting for less than 1% of the overall system [58], [59]. Hence, camouflaging can be efficiently applied to the controllers. The area and power of camouflaging all of the gates in the controllers are negligible as compared to the area and power consumed by the baseline processor.

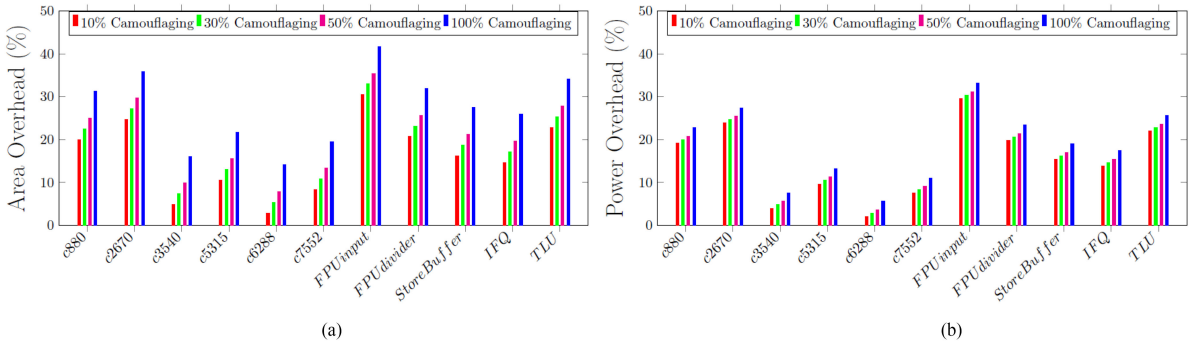

Fig. 15. Different amounts of camouflaging: (a) area and (b) power overhead.

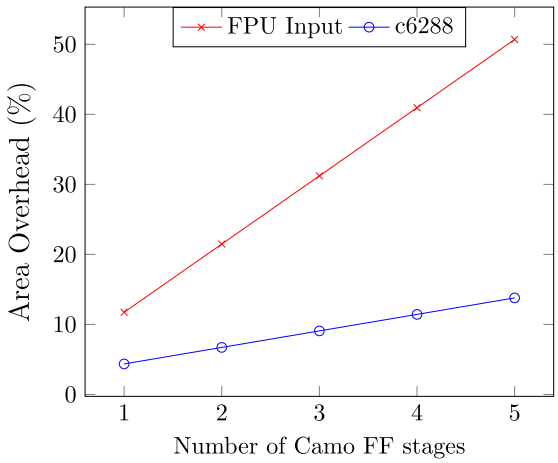

Fig. 16. Area overhead in terms of number of camouflaged DFF stages.

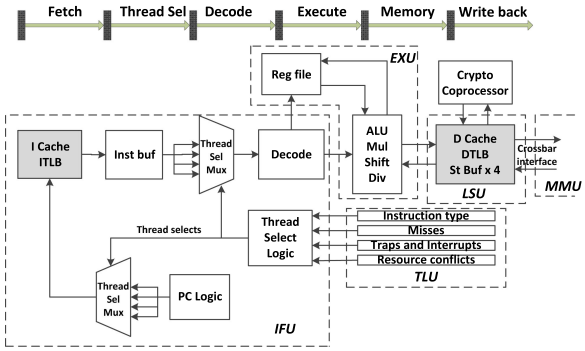

Fig. 17. Five-stage pipeline of the OpenSPARC processor.

TABLE VI
OPENSPARC CONTROLLERS AND THEIR CORRESPONDING IP SECRET [37]

Controller	IP secret
Instruction decoder	Decoding
Load Store unit read/write	Read/write priorities
Floating point input	Schedule FPU ops
Exception Handler	Supported exceptions
Floating point divider	Division
Load store unit store buffer	Ordering stores
Instruction queue	Cache line validation
Trap logic unit	Interrupt handler

VII. CONCLUSION

In this article, methods for camouflaging RSFQ circuits using two types of camouflaging cells are presented. The proposed camouflaged AND/OR cell can represent either AND or OR boolean functionality with the same layout, increasing ambiguity during RE. A camouflaged DFF has been proposed which can function as a DFF or JTL. The flip-flop is used to obfuscate the temporal distribution of the inputs to the circuit. The camouflaged AND/OR cell requires an area overhead of up to 15%, power overhead of up to 5%, and delay overhead of up to 57%. The camouflaged DFF uses the same layout as a standard DFF. While functioning as a JTL, the camouflaged DFF offers an area and delay overhead of 100%, and a power overhead of 2%. The camouflaged cells obfuscate large portions of a circuit, increasing the attacker's effort.

Model checking-based decamouflaging attacks are performed on the ISCAS benchmarks and the controllers of the OpenSPARC T1 processor. The efficacy of the model checking–based attack is limited by the size of the circuit and the complexity of the model checking problem. As demonstrated by the results of the attack (see Table IV), the defense is able to thwart this attack. A test-based attack is thwarted through the use of a CBS + OCS strategy while selecting the location of the camouflaged AND/OR gates. The defense can provide immunity against state-of-the-art attacks when integrated into commercial ICs, whose size is orders of magnitude greater than the ISCAS benchmark circuits.

In this article, the analysis is focused on RSFQ circuits whose CMOS counterpart is combinational in nature. The security implications of the equivalent RSFQ logic for sequential CMOS circuits will be pursued in future work. In addition to RE, vulnerabilities due to RSFQ design-for-test structures will also be assessed.

REFERENCES

[1] S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, no. 5, pp. 67–77, May 2011. [Online]. Available: https://cacm.acm.org/magazines/2011/5/107702-the-future-of-microprocessors/fulltext

[2] P. Kogge, "Next-generation supercomputers," Jan. 2011. [Online]. Available: https://spectrum.ieee.org/computing/hardware/nextgeneration-supercomput ers

[3] D. C. Brock, "The NSA's frozen dream," *IEEE Spectrum*, vol. 53, no. 3, pp. 54–60, Mar. 2016.

[4] D. K. Brock, "RSFQ technology: Circuits and systems," *Int. J. High Speed Electron. Syst.*, vol. 11, no. 01, pp. 307–362, 2001.

[5] K. Gaj, E. G. Friedman, and M. J. Feldman, "Timing of multi-gigahertz rapid single flux quantum digital circuits," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 16, no. 2–3, pp. 247–276, 1997.

[6] K. Likharev and V. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.

[7] Q. P. Herr, A. Y. Herr, O. T. Oberg, and A. G. Ioannidis, "Ultra-low-power superconductor logic," *J. Appl. Phys.*, vol. 109, no. 10, 2011, Art. no. 103903.

[8] O. Mukhanov, V. Semenov, and K. Likharev, "Ultimate performance of the RSFQ logic circuits," *IEEE Trans. Magn.*, vol. M-23, no. 2, pp. 759–762, Mar. 1987.

[9] O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 760–769, Jun. 2011.

[10] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Supercond. Sci. Technol.*, vol. 26, no. 3, 2013, Art. no. 035010.

[11] H. Engseth, S. Intiso, M. Rafique, E. Tolkacheva, and A. Kidiyarova-Shevchenko, "A high frequency test bench for rapid single-flux-quantum circuits," *Supercond. Sci. Technol.*, vol. 19, no. 5, 2006, Art. no. S376.

[12] O. Mukhanov, S. Sarwana, D. Gupta, A. Kirichenko, and S. Rylov, "Rapid single flux quantum technology for SQUID applications," *Physica C: Superconduct.*, vol. 368, no. 1–4, pp. 196–202, 2002.

[13] W. Chen, A. Rylyakov, V. Patel, J. Lukens, and K. Likharev, "Rapid single flux quantum T-flip flop operating up to 770 GHz," *IEEE Trans. Appl. Superconduct.*, vol. 9, no. 2, pp. 3212–3215, Jun. 1999.

[14] S. K. Tolpygo *et al.*, "Advanced fabrication processes for superconducting very large-scale integrated circuits," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 3, Apr. 2016, Art. no. 1100110.

[15] G. Pasandi, A. Shafaei, and M. Pedram, "SFQmap: A technology mapping tool for single flux quantum logic circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2018, pp. 1–5.

[16] K. Gaj, Q. P. Herr, V. Adler, A. Krasniewski, E. G. Friedman, and M. J. Feldman, "Tools for the computer-aided design of multi-gigahertz superconducting digital circuits," *IEEE Trans. Appl. Superconduct.*, vol. 9, no. 1, pp. 18–38, Mar. 1999.

[17] C. J. Fourie, "Digital superconducting electronics design tools status and roadmap," *IEEE Trans. Appl. Superconduct.*, vol. 28, no. 5, Jan. 2018, Art. no. 1300412.

[18] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, L. M. Johnson, M. A. Gouker, and W. D. Oliver, "Fabrication process and properties of fully-planarized deep-submicron Nb/Al-AlO_x/Nb josephson junctions for VLSI circuits," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 3, Jun. 2015, Art. no. 1101312.

[19] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1985, pp. 685–698.

[20] N. Katam, A. Shafaei, and M. Pedram, "Design of complex rapid single-flux-quantum cells with application to logic synthesis," in *Proc. Int. Superconductive Electron. Conf.*, Jun. 2017, pp. 1–3.

[21] S. Heck, S. Kaza, and D. Pinner, "Creating value in the semiconductor industry," *McKinsey Semiconductors*, Oct. 2011, pp. 5–144.

[22] J. Hurtarte, E. Wolsheimer, and L. Tafoya, *Understanding Fabless IC Technology*. New York, NY, USA: Elsevier, Aug. 2007.

[23] I. Polian, G. T. Becker, and F. Regazzoni, "Trojans in early design steps an emerging threat," presented at the 6th Conf. Trustworthy Manufact. Utilization Secure Devices (TRUDEVICE 2016), Barcelona, Spain, Nov. 14–16, 2016.

[24] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Comput.*, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[25] T. Huffmire *et al.*, "Managing security in FPGA-based embedded systems," *IEEE Des. Test Comput.*, vol. 25, no. 6, pp. 590–598, Nov./Dec. 2008.

[26] M. M. Tehranipoor, U. Guin, and D. Forte, Eds., "Counterfeit integrated circuits," in *Counterfeit Integrated Circuits*. Springer, 2015, pp. 15–36.

[27] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in *Proc. Des., Autom. Test Europe*, Mar. 2008, pp. 1069–1074.

[28] A. L. Oliveira, "Techniques for the creation of digital watermarks in sequential circuit designs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1101–1117, Sep. 2001.

[29] I. Torunoglu and E. Charbon, "Watermarking-based copyright protection of sequential functions," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 434–440, Mar. 2000.

[30] A. Cui and C.-H. Chang, "Intellectual property authentication by watermarking scan chain in design-for-testability flow," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 2645–2648.

[31] A. Cui and C. Chang, "A post-processing scan-chain watermarking scheme for VLSI intellectual property protection," in *Proc. IEEE Asia–Pacific Conf. Circuits and Syst.*, Dec. 2012, pp. 412–415.

[32] Grey B, "How we used reverse engineering to confirm patent infringement?" August 2017. [Online]. Available: https://www.greyb.com/reverse-engineering-patent-infringement

[33] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Proc. 48th Des. Autom. Conf.*, San Diego, CA, USA, Jun. 5–10, 2011, doi: 10.1145/2024724.2024805.

[34] SypherMedia, "Syphermedia library circuit camouflage technology." [Online]. Available: http://www.smi.tv/camodata-sheet.pdf

[35] J. P. Baukus, L. W. Chow, R. P. Cocchi, P. Ouyang, and B. J. Wang, "Camouflaging a standard cell based integrated circuit," phUS Patent no. 8151235, 2012.

[36] J. P. Baukus, L. W. Chow, R. P. Cocchi, P. Ouyang, and B. J. Wang, "Building block for a secure CMOS logic cell library," phUS Patent no. 8111089, 2012.

[37] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 709–720.

[38] M. E. Massad, S. Garg, and M. V. Tripunitara, "Integrated circuit (IC) decamouflaging: Reverse engineering camouflaged ICs within minutes," in *Proc. 22nd Annu. Netw. Distrib. Syst. Secur. Symp.*, San Diego, California, USA, Feb. 8–11, 2015.

[39] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Camoperturb: Secure IC camouflaging for minterm protection," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2016, pp. 1–8.

[40] J. Kunert, O. Brandel, S. Linzen, O. Wetzstein, H. Toepfer, T. Ortlepp, and H. Meyer, "Recent developments in superconductor digital electronics technology at fluxonics foundry," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 5, Oct. 2013, Art. no. 1101707.

[41] J. F. Annett, *Superconductivity, Superfluids and Condensates*. Oxford University Press, 2004.

[42] B. Dimov *et al.*, "Tuning of the RSFQ gate speed by different Stewart-McCumber parameters of the Josephson junctions," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 2, pp. 284–287, Jun. 2005.

[43] F. Frost, R. Fechner, B. Ziberi, J. Vllner, D. Flamm, and A. Schindler, "Large area smoothing of surfaces by ion bombardment: fundamentals and applications," *J. Phys.: Condens. Matter*, vol. 21, no. 22, May 2009, Art. no. 224026.

[44] T. Kanayama, H. Tanoue, and T. Tsurushima, "Niobium silicide formation induced by arion bombardment," *Appl. Phys. Lett.*, vol. 35, no. 3, pp. 222–224, Aug. 1979.

[45] "HYPRES Design Rules HYPRES, Inc.," Mar. 2015. [Online]. Available: https://www.hypres.com/wp- content/uploads/2010/11/DesignRules-6.pdf

[46] S. B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, "Power-analysis attack on an ASIC AES implementation," in *Proc. Int. Conf. Inf. Technol.: Coding Comput. (ITCC04)*, vol. 2, Apr. 2004, pp. 546–552.

[47] D. Sheen, S. Ali, D. Oates, R. S. Withers, and J. Kong, "Current distribution, resistance, and inductance for superconducting strip transmission lines," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 2, pp. 108–115, Jun. 1991.

[48] R. Meservey and P. M. Tedrow, "Measurements of the kinetic inductance of superconducting linear structures," *J. Appl. Phys.*, vol. 40, no. 5, pp. 2028–2034, Apr. 1969.

[49] D. Niepce, "Fabrication and characterisation of thin-film superconducting nanowire superinductors for novel quantum devices," Master's thesis, Dept. Microtechnol. Nanosci., Quantum Technol. Lab., Chalmers Univ. Technol., Goteborg, Sweden, 2014.

[50] M. E. Massad, S. Garg, and M. Tripunitara, "Reverse engineering camouflaged sequential circuits without scan access," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, Nov. 2017, pp. 33–40.

[51] U. Guin, Z. Zhou, and A. Singh, "Robust design-for-security architecture for enabling trust in IC manufacturing and test," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 5, pp. 818–830, May 2018.

[52] G. Krylov and E. G. Friedman, "Design for testability of SFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 8, Dec. 2017, Art. no. 1302307.

[53] D. Harris and S. Harris, *Digital Design and Computer Architecture*. San Mateo, CA, USA: Morgan Kaufmann, 2010.
[54] L. T. Wang, C. Wu, and X. Wen, *VLSI Test Principles and Architectures: Design for Testability*, 1st ed. Elsevier, Jul. 2006.
[55] J. Rajendran *et al.*, "Fault analysis-based logic encryption," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 410–424, Feb. 2015.
[56] "Sun microsystems, OpenSPARC T1 processor." [Online]. Available: http://www.opensparc.net/opensparc-t1/index.html
[57] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "NUSMV: A new symbolic model checker," *Int. J. Softw. Tools Technol. Transfer*, vol. 2, no. 4, pp. 410–425, Mar. 2000.
[58] A. Waksman, S. Sethumadhavan, and J. Eum, "Practical, lightweight secure inclusion of third-party intellectual property," *IEEE Des. Test*, vol. 30, no. 2, pp. 8–16, Apr. 2013.
[59] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proc. USENIX Secur. Symp.*, 2007, pp. 20:1–20:16.

Harshit Kumar is currently working toward the dual degree (B.Tech. + M.Tech) with the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, West Bengal, India. He is currently pursuing a specialization in VLSI & Microelectronics.

His current research interests include hardware security and brain–machine interfaces.

Tahereh Jabbari (S'15) received the B.Sc. and M.Sc. degrees in electrical engineering in, respectively, 2012 and 2015, from the Sharif University of Technology in Tehran, Iran. She also received the M.Sc. degree in electrical engineering from the University of Rochester in Rochester, NY, USA, in 2019. She is currently working toward the Ph.D. degree in electrical engineering with the University of Rochester.

She was a Researcher with the Superconductor Electronics Research Laboratory, Department of Electrical Engineering, Sharif University of Technology, from 2015 to 2017, to research and develop superconducting digital electronics. In 2017, she joined the graduate program in the Electrical and Computer Engineering Department at the University of Rochester. Her research interest focuses on superconducting digital electronics, electronic design automation, global signaling, and synchronous clocking of SFQ VLSI circuit, and security analysis of superconductive electronics.

Gleb Krylov (S'16) received the Specialist degree in computer engineering from the National Research Nuclear University MEPhI, Moscow, Russia, in 2014, and the M.S. degree in electrical engineering from the University of Rochester, NY, USA, in 2017, where he is currently working toward the Ph.D. degree.

In 2017, he interned with Hypres, Inc., Elmsford, NY, USA. In 2018, he interned with Synopsys, Inc., Mountain View, CA, USA. His current research interests include superconducting digital electronics and electronic design automation.

Kanad Basu received the Ph.D. degree in computer engineering from the Department of Computer and Information Science and Engineering, University of Florida, FL, USA, in 2012. His dissertation was focused on improving signal observability for post-silicon validation.

He worked in various semiconductor companies like IBM and Synopsys. At IBM, he was responsible for the design on IBM Power and Z Processors. At Synopsys, he helped in development of DFTMAX Ultra, the state-of-the-art low-pin hardware test solution. While working toward the Ph.D. degree, Kanad was a graduate intern at Intel Corporation. Currently, he is an Assistant Research Professor with the Electrical and Computer Engineering Department, New York University (NYU), NY, USA. His research interests include hardware security, malware detection, high-level synthesis, post-quantum cryptography and machine learning. He holds 2 U.S. patents, and has authored 2 book chapters, 6 journal articles, and 26 articles in peer-reviewed conference proceedings. He has won several awards in both academia and industry, including the "Best Paper Award" at the International Conference on VLSI Design 2011.

Eby G. Friedman (F'00) received the B.S. degree from Lafayette College, Easton, PA, USA, in 1979, and the M.S. and Ph.D. degrees from the University of California, Irvine, CA, USA, in 1981 and 1989, respectively, all in electrical engineering.

He was with Hughes Aircraft Company from 1979 to 1991, and, as Manager of the Signal Processing Design and Test Department, was responsible for the design and test of high-performance digital and analog ICs. He has been with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA, since 1991, where he is a Distinguished Professor, and the Director of the High Performance VLSI/IC Design and Analysis Laboratory. He is also a Visiting Professor at the Technion–Israel Institute of Technology. He is the author of more than 500 papers and book chapters, 17 patents, and the author or editor of 18 books in the fields of high-speed and low-power CMOS design techniques, 3-D design methodologies, high-speed interconnect, and the theory and application of synchronous clock and power distribution networks. His current research and teaching interests include high-performance synchronous digital and mixed-signal microelectronic design and analysis with application to high-speed portable processors, low-power wireless communications, and server farms.

Dr. Friedman is the Editor-in-Chief of the *Microelectronics Journal*, a Member of the editorial board of the *Journal of Low Power Electronics* and *Journal of Low Power Electronics and Applications*, and a Member of the technical program committee of numerous conferences. He previously was the Editor-in-Chief and Chair of the steering committee of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, the Regional Editor of the *Journal of Circuits, Systems and Computers*, a Member of the editorial board of the PROCEEDINGS OF THE IEEE, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, *Analog Integrated Circuits and Signal Processing*, IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, and *Journal of Signal Processing Systems*, a Member of the Circuits and Systems (CAS) Society Board of Governors, Program and Technical Chair of several IEEE conferences, and a recipient of the IEEE Circuits and Systems Mac Van Valkenburg Award, IEEE Circuits and Systems Charles A. Desoer Technical Achievement Award, a University of Rochester Graduate Teaching Award, and a College of Engineering Teaching Excellence Award. Dr. Friedman is a Senior Fulbright Fellow.

Ramesh Karri received the B.E. degree in electronics and communication engineering from Andhra University, Visakhapatnam, India, in 1985, M.Tech. degree in computer science from University of Hyderabad, in 1988, and the M.Sc. degree in computer engineering and the Ph.D. degree in computer science and engineering from the University of California at San Diego, La Jolla, CA, USA, in 1992 and 1993, respectively.

He is currently a Professor of Electrical and Computer Engineering with New York University (NYU), NY, USA. He codirects the NYU Center for Cyber Security (http://cyber.nyu.edu) and leads the Cyber Security thrust of the NY State Center for Advanced Telecommunications Technologies at NYU. He cofounded the Trust-Hub (http://trust-hub.org). He organizes the global red-team-blue-team hardware hacking event, the Embedded Systems Challenge (https://csaw.engineering.nyu.edu/esc). He served/serves as the Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (2010–2014), IEEE TRANSACTIONS ON CAD (2014–present), *ACM Journal of Emerging Computing Technologies* (2007–present), *ACM Transactions on Design Automation of Electronic Systems* (2014–present), IEEE ACCESS (2015–), IEEE TRANSACTIONS ON EMERGING TECHNOLOGIES IN COMPUTING (2015–present), IEEE DESIGN AND TEST (2015–present), and IEEE EMBEDDED SYSTEMS LETTERS (2016–present). He served as an IEEE Computer Society Distinguished Visitor (2013–2015). He served in the Executive Committee of the IEEE/ACM Design Automation Conference leading the Security@DAC initiative (2014–2017). He has given invited keynotes, talks, and tutorials on Hardware Security and Trust (ESRF, DAC, DATE, VTS, ITC, ICCD, NATW, LATW, CROSSING, HIPEAC). He has published over 200 articles in leading journals and conference proceedings. His research and education interests are in hardware cybersecurity and include trustworthy ICs; processors and cyber–physical systems; security-aware computer-aided design, test, verification, validation, and reliability; nano meets security; hardware security competitions, benchmarks, and metrics; biochip security; and additive manufacturing security.

Dr. Karri cofounded 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, and WISEC. He serves on several program committees (HOST, ITC, VTS, ETS, ICCD, DTIS, WIFS). His work on hardware cybersecurity received Best Paper Award nominations (ICCD 2015 and DFTS 2015) and awards (ITC 2014, CCS 2013, DFTS 2013, and VLSI Design 2012, ACM Student Research Competition at DAC 2012, ICCAD 2013, DAC 2014, ACM Grand Finals 2013, Kaspersky Challenge and Embedded Security Challenge). He received the Humboldt Fellowship and the National Science Foundation CAREER Award.