Programmable Daisychaining of Microelectrodes to Secure Bioassay IP in MEDA Biochips

Tung-Che Liang, Krishnendu Chakrabarty, *Fellow, IEEE*, and Ramesh Karri, *Fellow, IEEE*

***Abstract*— As digital microfluidic biochips (DMFBs) make the transition to the marketplace for commercial exploitation, security and intellectual property (IP) protection are emerging as important design considerations. Recent studies have shown that DMFBs are vulnerable to reverse engineering aimed at stealing biomolecular protocols (IP theft). The IP piracy of proprietary protocols may lead to significant losses for pharmaceutical and biotech companies. The microelectrode dot array (MEDA) is a next-generation DMFB platform that supports real-time sensing of droplets and has the added advantage of important security protection. However, real-time sensing offers opportunities to an attacker to steal the biochemical IP. We show that the daisychaining of microelectrodes and the use of one-time programmability in MEDA biochips provides effective bitstream scrambling of biochemical protocols. To examine the strength of this solution, we develop a Satisfiability (SAT)-based attack that can unscramble the bitstreams through repeated observations of bioassays executed on the MEDA platform. Based on insights gained from the SAT attack, we propose an advanced defense against IP theft. Simulation results using real-life biomolecular protocols confirm that while the SAT attack is effective for simple instances, our advanced defense can thwart it for realistic MEDA biochips and real-life protocols.**

***Index Terms*— Computer security, control systems, microfluidics.**

I. Introduction

DIGITAL microfluidic biochips (DMFBs) have recently been transitioned to the marketplace for automating biomolecular protocols. In 2015, Illumina announced the use of DMFBs for sample preparation [2]. As another example of commercialization, Baebies [3] recently introduced a U.S. Food and Drug Administration (USDA)-approved product to screen newborns for diseases. These milestones highlight the emergence of digital microfluidic technology for commercial exploitation and its potential for point-of-care diagnosis, sample processing, and cell-based assays [4]–[6]. It is anticipated that companies will develop intellectual property (IP) in the form of kits for various biochemical applications, and the protocols executed by these kits will incorporate sensitive information, for example, advanced technology for accelerating deep deoxyribonucleic acid (DNA) sequencing or quantitative polymerase chain reaction (qPCR)-based gene-expression analysis.

Recent work has shown that an attacker can reverse engineer (RE) a proprietary protocol by analyzing the actuation sequence [7]. Chen *et al.* [7] assumed that the attacker knows the mapping between the actuation sequence and the electrodes. Based on this assumption, the attacker can RE the biomolecular protocols using consecutive actuation bitstreams. Therefore, reverse engineering aimed at stealing biomolecular protocols (IP theft) is of particular concern. The IP piracy of proprietary protocols may lead to significant losses for pharmaceutical and biotechnology companies.

Obfuscation is an attractive approach for IP protection. An obfuscation mechanism for DMFBs was proposed recently to secure IP protocols using a finite-state machine [8]. This article first proposes a physical unclonable function (PUF) whose response is based on the inherent variation in the fluidic operations of DMFBs. This article then utilizes the combination of PUF response bits and the license issued by the foundry as a key. Trusted users can gain access to the actuation sequences based on the challenge–response pairs associated with the PUF module. However, this article has two major drawbacks.

1) The PUF-based scheme is not secure against malicious adversaries who pretend to be legitimate users. Because commercialized biochips are sold in open markets, biochip developers cannot ensure that all users are trustworthy.
2) The PUF-based scheme may not be reliable after several bioassay executions because electrodes may degrade over time [8]. Chen *et al.* [7] had also shown that information about the proprietary protocol may be stolen through actuation sequences or sensor data. Therefore, actuation sequences and sensor data should be obfuscated in order to thwart RE attacks [7].

The microelectrode dot array (MEDA) is a next-generation DMFB platform that supports real-time sensing of droplets and has the added advantage of important security protections [9], [10]. A MEDA biochip is composed of an array of identical microelectrode cells (MCs). It has been fabricated at the Taiwan Semiconductor Manufacture Company (TSMC) using a mainstream 0.35-μm CMOS process, and an example of a fabricated chip is shown in Fig. 1. Each MC consists of a microelectrode, an electronic control circuit, and a sensing module that enables real-time sensing of droplets. The MCs are connected together to form a daisychain. Therefore, unlike electrodes in traditional DMFBs that are wired to distinct input ports for actuation, a MEDA biochip requires only one input port to actuate all the microelectrodes. The actuation pattern corresponding to all the microelectrodes is mapped to a bitstream, and this bitstream is scanned-in to the daisychain through the input port. The translation of the actuation pattern to the bitstream is based on the daisychain structure, and the

Manuscript received September 25, 2019; revised December 6, 2019; accepted January 5, 2020. Date of publication February 13, 2020; date of current version April 24, 2020. This work was supported in part by the Army Research Office under Grant W911NF-17-1-0320 and in part by the National Science Foundation under Grant CNS-1833622 and Grant CNS-1833624. *(Corresponding author: Tung-Che Liang.)*

Tung-Che Liang and Krishnendu Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: tung.che.liang@duke.edu; krish@duke.edu).

Ramesh Karri is with the Department of Electrical and Computer Engineering, New York University, New York, NY 10003 USA (e-mail: 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/TVLSI.2020.2967029

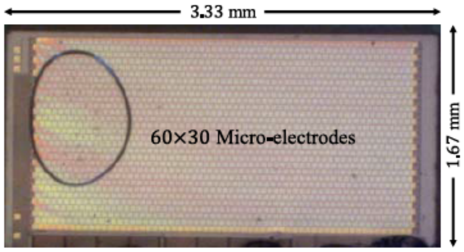


Fig. 1. Photograph of a MEDA biochip with a dispensed droplet. (Taken during our laboratory experiment.) The movement of the droplets can be monitored in real time during biochip operation.

ordering of the microelectrodes in a daisychain determines the mapping of the actuation sequence to the scanned-in bitstream. By exploiting this mapping in MEDA, the actuation sequences can be scrambled, and information leakage of the proprietary protocols can be avoided.

In this article, we present a programmable daisychain structure for MEDA biochips and show that the daisychaining and use of one-time programmability can protect biochemical protocol IP. The daisychain structure is designed to be one-time programmable after manufacturing and serves as a hidden key so that only authorized protocols are performed on the biochip. In addition, the scan-out data of the MEDA biochip is scrambled by the daisychain structure. To examine the strength of this solution, we develop a Boolean Satisfiability (SAT)-attack that can decipher the daisychain structure by observing the execution of the bioassays on the platform. Based on the insights gained from the SAT attack, we propose an enhanced daisychain structure. Simulation results using real-life biomolecular protocols and a theorem confirm that while the SAT attack is effective on simple instances, it can be thwarted by the advanced defense for realistic MEDA biochips and real-life protocols.

The key contributions of this article are as follows.

1) We describe a programmable daisychain structure that can be scrambled. The programmable structure can be mass produced in a typical semiconductor foundry. As a result, the cost of adding programmability to the daisychain is negligible.
2) We describe the translation processes corresponding to the actuation bitstreams and the sensed bitstreams using two matrix formulations. We discover that the actuation-bitstream translation matrix is orthogonal and that the transpose of this matrix is the sensed-bitstream translation matrix.
3) We formulate a SAT attack on the programmable daisychain. The objective of this attack is to decipher the microelectrodes-to-actuation sequence mapping by monitoring the inputs and outputs and the movement of the droplets on MEDA.
4) We show using real-life bioassay protocols that, even with a programmable daisychain, the SAT attack can decipher the mapping by observing bioassay execution.
5) We present an enhanced programmable daisychain structure and prove that this advanced structure can defend successfully against SAT attacks.

The remainder of this article is organized as follows. Section II explains the MEDA architecture and its working principle. Section III introduces the programmable daisychain. Section IV first presents the threat model and possible attacks and then shows that, among these attacks, only the SAT attack can successfully discover the structure of the daisychain. Section V presents an enhanced defense against the SAT attack. Section VI provides security assessment for the proposed defense. Section VII shows that the SAT attack can reveal the structure of the daisychain during bioassay execution, but an advanced daisychain structure can defend against the attack. Section VIII discusses the overheads of using the enhanced daisychain structure and compares it with prior work in integrated-circuit (IC) security. Finally, conclusions are drawn in Section IX.

II. Technology Overview

A. MEDA Biochip Architecture

A MEDA biochip platform is composed of a synthesis processor, a control unit, and a 2-D MC array, as shown in Fig. 2(b). All MCs in a MEDA are connected as a daisychain. Ideally, the MCs can be connected freely if a feasible routing can be found and there is no metal–layer constraint. However, in reality, to reduce the fabrication cost, a biochip engineer will try to reduce the use of mask layers. Therefore, from the cost-reduction perspective, two adjacent components in a daisychain should be physically adjacent on the MC array.

The MC integrates a microelectrode, an activation circuit, and a sensing circuit. The schematic of an MC is shown in Fig. 2(c). Each MC can be operated in three modes: 1) scanning; 2) actuation; and 3) sensing. In the scanning mode, the bit stored in an MC is passed on to the next MC when the clock signal of the flip-flop changes. In the actuation mode, the stored bit determines whether the microelectrode is actuated. If the stored bit is "1," the microelectrode is actuated; the microelectrode remains at low voltage otherwise. In the sensing mode, the presence of a droplet over the microelectrode is determined by the MC. If there is a droplet present over the microelectrode, a "1" is captured in the flip-flop, "0" otherwise.

All of the MCs in a MEDA biochip operate synchronously. Fig. 3 shows a droplet on a 4×4 MC array. An actuation sequence is applied to the array in order to transport the droplet to the right. First, all MCs are set to the scanning mode, and the translated bitstream, (0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0), is scanned to the array. Next, all MCs are set to the actuation mode, and the scanned-in actuation pattern is applied to the microelectrodes. The activated microelectrodes are marked in a dark-gray color. The droplet is thus transported to the right. Note that in the actuation mode, the top plate is actuated with a high voltage, that is, 25 V [10]. Therefore, the circuits that are associated with the nonactivated microelectrodes must resist the breakdown voltage of 25 V. An extended-drain MOS (EDMOS) transistor is used in the MC to increase the breakdown voltage.[1] After the actuation mode, all MCs are set to the sensing mode, and the droplet information is stored in all MCs. Finally, all MCs are set to the scanning mode again, and the sensor data is scanned out as a bitstream. These four steps compose an operation cycle. To perform a bioassay on a MEDA biochip, the operations of the bioassay are synthesized and translated into a sequence of bitstreams and these bitstreams are scanned in to the MEDA biochip cycle by cycle.

[1] Researchers fabricated MEDA biochips using the technology node of 0.35 μm because the EDMOS transistor in this technology node can resist the breakdown voltage of 25 V [10].

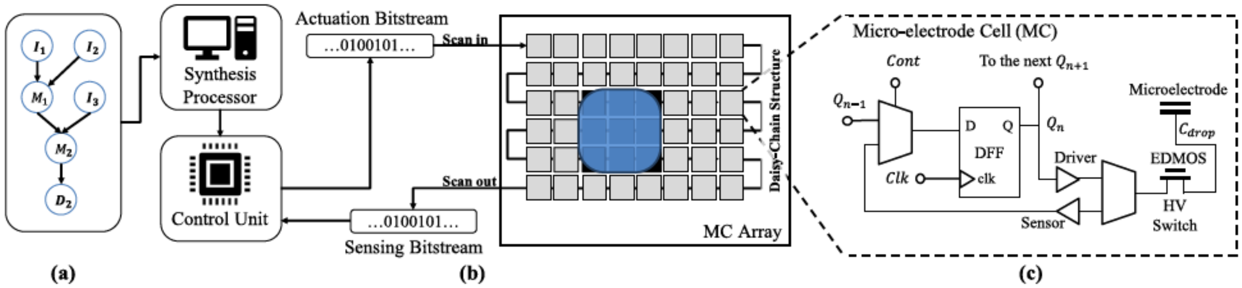


Fig. 2. (a) Illustrative sequencing graph. The nodes I_i, M_i, and D_i represent dispensing, merging, and detecting operations. (b) MEDA biochip platform consists of a synthesis processor, a control unit, and a 2-D MC array. (c) Circuit schematic of the sensor and control module of the MC. Q_n denotes the nth cell in the daisy-chain structure.

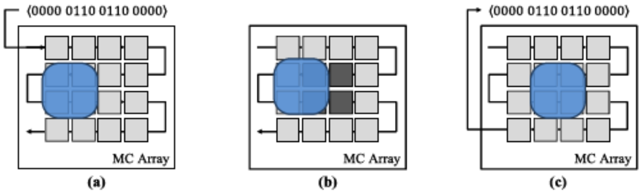


Fig. 3. Droplet on a 4 × 4 array is transported by the actuation sequence. (a) Scan in the actuation bitstream. (b) Apply actuation pattern to all microelectrodes. (c) Sense and scan out the bitstream.

B. MEDA Biochip Working Principle

To carry out biochemical assays on MEDA biochips, the given bioassay is interpreted as a sequencing graph that specifies the relationships between fluidic operations. An example of a sequencing graph is shown in Fig. 2(a). A synthesis tool loaded on the processor binds the operations to on-chip resources, generates an optimized schedule of these operations, and routes droplets on the biochip [11]. Based on the synthesis result, the actuation patterns for the microelectrodes are translated according to the daisychain structure of the MEDA biochip, and the actuation bitstreams are shifted to the MC array through the scan-in port of the MEDA biochip from the control unit sequentially. After an actuation pattern is activated on the MC array, a sensed bitstream is scanned out to the control unit as feedback corresponding to the scanned-in actuation pattern [10].

III. Daisychaining and Bitstream Translation

In this section, we explain the translation processes for the actuation bitstreams and the sensed outcomes using matrix formulations. Next, we present a programmable daisychain structure.

A. Daisychaining in MEDA

After high-level synthesis, droplet locations for all time steps during a bioassay execution are determined. In order to apply the actuation patterns to the MEDA, the determined droplet locations need to be translated into actuation bitstreams. The actuation-bitstream translation process can be modeled using a matrix formulation. Assume that a MEDA biochip contains a total of N microelectrodes. Let the droplet location at time t be D_t, where D_t is a Boolean vector of length N. The ith component of D_t, denoted as d_t^i, indicates that a droplet is present over the ith microelectrode at time t. Let B_t also be a Boolean vector of length N, in which a component b_t^i indicates that the ith microelectrode in the daisychain is activated with high voltage. Therefore, B_t can be described as

$$B_t = D_t \cdot K \tag{1}$$

where K is an $N \times N$ Boolean matrix. An entry $k_{i,j}$ within K indicates whether the ith bit component in D_t corresponds to the jth bit component in B_t.

Similar to the actuation-bitstream translation, the sensed outcomes (in forms of bitstreams) should also be translated properly so that the associated controller can comprehend the sensed results. The translation of the sensed bitstreams can also be formulated using a matrix. Because K is a one-to-one mapping matrix between the microelectrodes and the bitstreams, it is an orthogonal matrix, that is, $K^{-1} = K^T$. Let the sensed bitstream at time t be $\widehat{B_t}$ and the translated droplet location at time t be $\widehat{D_t}$, where $\widehat{B_T} \in \mathbb{B}^N$ and $\widehat{D_t} \in \mathbb{B}^N$. Therefore, to translate $\widehat{B_t}$ back to $\widehat{D_t}$, that is, to derive meaningful information, we can obtain from (1)

$$\widehat{B_t} \cdot K^T = \widehat{D_t} \cdot K \cdot K^T = \widehat{D_t}. \tag{2}$$

Note that $K \cdot K^T = K \cdot K^{-1} = I$, that is, an identity matrix.

An example of actuation-bitstream translation is shown in Fig. 4(a). Assume that at time t, four microelectrodes need to be activated based on the synthesis result, and the corresponding vector $D_t = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0)$. Suppose that two MEDA biochips, Chip 1 and Chip 2, contain the same number of microelectrodes, but their daisychain structures are different. The translation matrices for these two biochips are denoted as K_1 and K_2, respectively. Because $K_1 \neq K_2$, the translated bitstreams at time t for the two biochips are different according to (1). Let the actuation bitstream for Chip 1 at time t be B_t^1, and $B_t^1 = (0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. The actuation bitstream for Chip 2 at time t is given by $B_t^2 = (0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)$. Note that even though two biochips are activated with the same pattern, $B_t^1 \neq B_t^2$.

We next show that the translated actuation bitstreams for a daisychain structure cannot be used for another daisychain structure. If we exchange the bitstreams for the two chips, that is, scan in B_t^1 for Chip 2 and B_t^2 for Chip 1, the desired actuation patterns cannot be applied to the microelectrodes. Unexpected patterns are activated on the two biochips instead; these switched actuation patterns are shown in Fig. 4(b). During a bioassay execution, an actuation bitstream contributes to a fluidic operation. If an unintended bitstream is scanned in for microelectrode actuation, the corresponding fluidic operation will fail, and the overall bioassay execution will not complete.

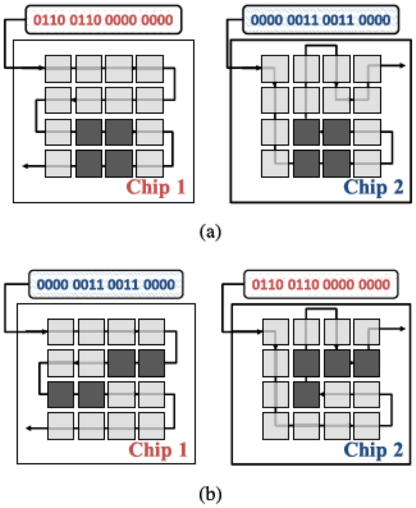


Fig. 4. Two distinct daisychain structures with their actuation sequences. (a) In order to actuate the same microelectrodes on two separate MC arrays, two different actuation bitstreams are required. (b) If we exchange the actuation bitstreams, the desired fluidic operations cannot be performed on the arrays.

For example, to dispense reagents from the reservoir on the biochip, the microelectrodes at the dispensing port should be activated; if other microelectrodes are activated instead, the dispensing operation fails. Likewise, without knowing the daisychain structure (or the translation matrix K), a user cannot execute any fluidic operations on the biochip using random actuation bitstreams, nor can the user comprehend the scanned-out bitstreams for a bioassay. By exploiting this characteristic, MEDA biochips provide an extra translation layer that can be used to scramble actuation bitstreams and the sensor data.

B. Programmable Daisychaining

We propose a programmable daisychaining solution for MEDA biochips so that bioassay IP in forms of bitstreams can be protected. For a given MC array, each MC is surrounded by two to four MCs (depending on whether the MC is on the edge). As shown in Fig. 5, the boundary of an MC only overlaps four MC boundaries. The input–output of an MC connects to the output/input of a surrounding MC. To make the daisychain programmable, a multiplexer (MUX) and a demultiplexer (DeMUX) are added at the input and the output of each MC, respectively, the structure of which is shown in Fig. 6. By carefully assigning control bits to MUX and DeMUX, the next MC and the previous MC are connected in a specific way. The control bits of all MCs can be wired to a read-only-memory (ROM) array that is one-time programmable after manufacturing so that the daisychain structure is also one-time programmable.

The original MC contains 36 CMOS transistors and an EDMOS [10]. The additional gates, a MUX and a DeMUX, require 36 CMOS transistors. Therefore, the new MC is twice as large as the original MC. However, the increased logic is acceptable because it does not affect the fluidic operations on MEDA biochips and neither does it increase the chip footprint. The original MC in [12] was designed in a layout area of $5050\,\mu\text{m} \times 50\,\mu\text{m}$ using the $0.35\,\mu\text{m}$ process. The proposed new MC requires an area of $(50\sqrt{2}) \times (50\sqrt{2}) \approx 70{\times}70\,\text{mm}^2$. According to [13], the radius of the smallest droplet that can be dispensed and moved on to MEDA biochips is $70\,\mu\text{m}$. The area of the new MC is approximately the same as the smallest area occupied by a droplet. As a result, all MEDA-enabled operations can be performed on an array made up of the new MCs.

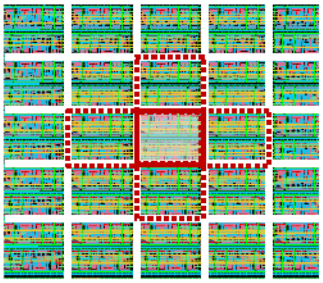


Fig. 5. Circuit layout of an MC array.

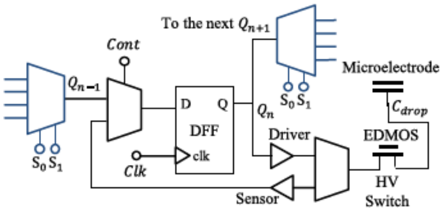


Fig. 6. Programmable MC.

IV. Attacks to Discover the Daisychain Structures

In this section, we present a threat model that benefits from discovering the structure of the daisychain. We then propose three attacks to discover the structure of a daisychain. We first investigate if a brute-force attack can discover the daisychain structure, that is, check if an attacker can enumerate all possible daisychains in an MC array and test each one of them to find out the correct structure. Second, we examine whether differential analysis can reveal the daisychain structure. Because these two attacks cannot effectively reveal the secret structure, we propose a powerful SAT-based attack to discover the daisychain structure.

A. Threat Model

To better motivate our threat model, we first examine the current business model of biotech companies such as Baebies [3]. Biotech companies sell full bio-protocol solutions to customers, and a bio-protocol solution includes a MEDA biochip platform and a bioassay in the form of actuation bitstreams. The biochip platform is opaque and connected to the Internet for software/firmware upgrade or for real-time online monitoring [14]. During the bioassay execution, the platform takes snapshots and sends the snapshots back to the biotech company through the Internet. The biotech company compares the snapshots with the golden execution to ensure system reliability. To avoid overwhelming transmitting data and Internet traffic, the snapshots are not captured and sent continuously (i.e., every clock cycle) during a bioassay execution. Instead, the platform takes a snapshot every few fluidic operations.

Similar to the reverse-engineering work in [7], we consider that the attack is not intrusive. Therefore, the attacker does not know the daisychain structure of the biochip. We assume that the attacker cannot observe all the fluidic operations during a bioassay execution because the platform is opaque. We assume that the attacker can obtain the bio-protocol in the form of bitstreams, which are stored in a storage unit in the platform. In addition, we assume that the attacker can intervene the communication from the biochip platform to the biotech company and acquire the execution snapshots. However, the attacker cannot RE the bio-protocol only using the snapshots because they are not consecutively captured. Therefore, the attacker's initial goal is to discover the daisychain structure, that is, the key matrix K, using the bitstreams and the snapshots. After discovering the daisychain structure, the attacker can RE the bio-protocol using the overall bitstreams.

Each fabricated MEDA biochip has a unique K based on the programmable daisychain described in Section III. The actuation bitstreams of a bioassay from the biotech company are translated according to the distinct daisychain structure. The biochips are designed in a way such that a biochip cannot execute more than one bioassay. The biotech company registers the matrix K of each biochip with the associated bioassay; a K can only be used to translate the actuation bitstreams of a bioassay. Therefore, if the attacker acquires more than one bioassay (along with more than one biochip), the analyzed data from one biochip cannot be used for another biochip.

As shown in Section III, the attacker cannot scan in any arbitrary bistreams to a MEDA to observe desired fluidic operations because the matrix K is unknown. In addition, the attacker cannot launch an attack by scanning an actuation bitstream that contains only one "1" bit. In MEDA biochips, droplets are much larger than a microelectrode. Even though the attacker shifts an actuation bitstream that contains only one "1" bit, the induced force associated with the only one actuated microelectrode is too small to move a droplet. The attacker still cannot observe any fluidic operations. Therefore, shifting such a bitstream does not benefit the attack.

B. Brute-Force Attack

The number of daisychains on a 2-D array of MCs can be formulated as the number of Hamiltonian paths in a 2-D grid graph. The daisychaining of MCs must satisfy two constraints. First, all MCs need to be connected as a chain for scanning in/out actuation/sensed bitstreams. Second, the MCs need to be connected using shortest wiring to obtain a small biochip. This can be accomplished by wiring physically adjacent MCs on the array as a daisychain. A MEDA biochip with $L \times W$ MC array can be modeled as an undirected grid graph $G_{L,W} = (V, E)$, where nodes in V are the MCs. Let the node corresponding to the ith row and jth column be denoted as $n_{i,j}$. For any given $n_{i,j}, n_{k,l} \in V$, $(n_{i,j}, n_{k,l}) \in E$ if $n_{i,j}$ and $n_{k,l}$ are physically adjacent on the MC array, i.e., ($|i - k| = 1$ and $j = l$) or ($i = k$ and $|j - l| = 1$). A Hamiltonian path is a path in the graph that visits each vertex exactly once [15]. As a result, a daisychain in an $L \times W$ MC array can be modeled as a Hamiltonian path in the corresponding graph $G_{L,W}$. Fig. 7 shows a 2×2 MC array and the corresponding graph $G_{2,2}$. Four Hamiltonian paths can be found in $G_{2,2}$.

It is well known that determining whether a graph contains a Hamiltonian path is an NP-complete problem [16]. As a result, it is even more difficult to determine how many distinct Hamiltonian paths exist in a given graph. To the best of our knowledge, a closed-form solution for this problem is not available, and the problem remains unsolved. However, algorithms have been proposed to exhaustively enumerate all paths for a given grid graph [16], [17]. Because of computation and memory limits, these algorithms can only enumerate paths for small-sized grid graphs. The enumeration of all paths for square grid graphs that are 17×17 or smaller is shown in Table I. For a grid graph that contains 17×17 vertices, there are 6.6×10^{43} distinct Hamiltonian paths [16], [17]. Because algorithms today can only enumerate Hamiltonian paths for less than 17×17 vertices in a grid graph, for a normal-sized MEDA biochip (with 60×30 MCs), it is not feasible for an attacker to exhaustively enumerate all possible daisychains.

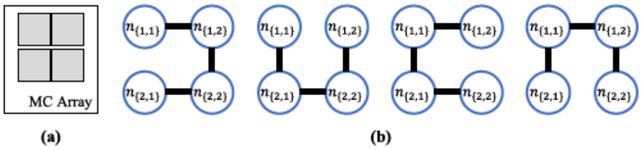


Fig. 7. Example of the daisychain enumeration. (a) MC array contains 2×2 microelectrodes. (b) Four Hamiltonian paths can be found in the corresponding grid graph $G_{2,2}$.

TABLE I

NUMBER OF HAMILTONIAN PATHS ON AN $L \times L$ SQUARE LATTICE, WHERE $1 \leq L \leq 17$ [16]

L	Number of paths
2	4
3	20
4	276
5	4324
6	299348
7	13535280
8	3023313284
9	745416341496
10	730044829512632
11	786671485270308848
12	3452664855804347354220
13	16652005717670534681315580
14	331809088406733654427925292528
15	7263611367960266490262600117251524
16	662634717384979793238814101377988786884
17	66428994739159469969440119579736807612665540

C. Differential Analysis

We propose a differential analysis to discover the mapping between the bitstreams and the MCs. Consider the example in Fig. 8 where three observations are made during a bioassay execution on a 3×3 MC array. Without any known mappings, the first bit in the bitstream can be mapped to any one of the nine microelectrodes, and the second bit can be mapped to 9–1 microelectrodes. Therefore, the number of possible daisychain structures is 9! Let the set of all microelectrodes be U and the set of microelectrodes that are under the droplets from the ith observation be S_i. In this example, $S_1 = \{e_1, e_2, e_4, e_5\}$, $S_2 = \{e_2, e_3, e_5, e_6\}$, and $S_3 = \{e_5, e_6, e_8, e_9\}$. The overlaps between these sets can be shown by a Venn diagram as in Fig. 9. Because $S_1 \cap S_2 \cap S_3 = \{e_5\}$ and $|S_1 \cap S_2 \cap S_3| = 1$, the mapping of e_5 to the corresponding bit in the bitstream can be found. Similarly, because $|U - (S_1 \cup S_2 \cup S_3)| = 1$, the mapping of e_7 to the corresponding bit in the bitstream can also be found. The five sets of microelectrodes of size 1 are colored green in Fig. 9, and the microelectrode mappings that correspond to these sets can be found. However, the remaining four microelectrode mappings cannot be found (even though all mappings are analyzed using the bitstreams). Therefore, a total of $(9 - 5)! = 24$ possible structures can be obtained

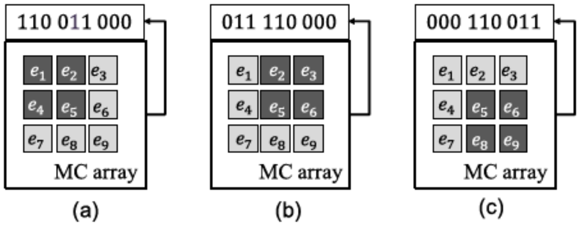


Fig. 8. Example of three observations. Actuated electrodes and the corresponding actuation bitstream at time (a) t, (b) $t + 1$, and (c) $t + 2$.

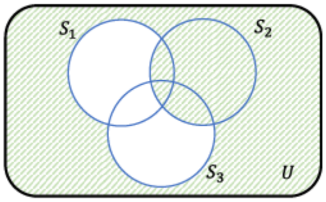


Fig. 9. Relationships between the microelectrode sets in Fig. 8.

using the differential analysis, and the probability of guessing the right structure is $1/24$.

From the above example, we can examine the best and the worst cases from the attacker's perspective for differential analysis if we obtain three observations on a 3×3 MC array. For the worst case scenario, $S_1 = S_2 = S_3$, and the sets in the corresponding Venn diagram contain either more than one or zero elements. Therefore, no microelectrode mappings can be found. The number of possible structures is 9! and the probability of guessing the right structure is $1/9! \approx 2.8 \times 10^{-6}$. On the other hand, the ideal scenario for the attacker is that seven out of eight sets in the corresponding Venn diagram contain only one element, and the other set contains two elements. In this case, the probability of guessing the right structure is $1/2!$ However, for this ideal case, the microelectrodes in any set (S_1, S_2, or S_3) form a nonrectangular shape on the MC array, which is impossible to obtain from the sensor result of a droplet. For example, the droplet cannot be an L shape on the MC array. As a result, it is not possible for an attacker to obtain the observations of the ideal case from bioassay execution.

Based on the analyses of the best-case and worst case scenarios, we learn that observations obtained from a bioassay execution impact the effectiveness of differential analysis. Therefore, if a bioassay is executed on an MC array in a particular way such that the observations obtained from the execution are not informative (i.e., deliberately differential analysis resistant), differential analysis does not prune away possible structures, and the probability of guessing the correct structure remains low. We simulate the executions of three real-life bioassays, namely the multiplexed *in vitro* bioassay [18], the chromatin immunoprecipitation protocol (ChIP) [19], and the gene-expression analysis [20], on a MEDA biochip with 60×30 MCs. Snapshots are taken during a given clock-cycle interval, for example, a snapshot is taken every five clock cycles. We simulated three different clock-cycle intervals. We then apply differential analysis to the observations obtained from these executions and show the results in Fig. 10. Because the multiplexed *in vitro* bioassay is the shortest bio-protocol among the three bioassays, we find the fewest discovered mappings using the associated observations. The simulation results show that when the interval cycle is longer, less observations can be made. Thus, fewer microelectrode mappings can be found using the differential analysis. The simulation results also show that the number of discovered microelectrode mappings is always smaller than 400.

Let the number of mappings found from differential analysis be $DF(n)$, where n is the number of observations. Therefore, the number of possible structures obtained from differential analysis is $(60 \times 30 - DF(n))!$ and the probability that an attacker can decipher the scrambled microelectrode-to-bitstream structure is $1/(1800 - DF(n))!$ Because $DF(n) \leq 400$ for all n according to the simulation, the probability of guessing the right structure is less than $1/(1800 - 400)! = 1/1400! \approx 2.9 \times 10^{-3799}$. Thus, it is unlikely that an attacker can guess the correct structure using differential analysis on a MEDA biochip.

D. SAT-Based Attack to Discover the Daisychain Structure

In order to further evaluate the security strength of the programmable daisychaining, we propose a SAT-based attack to discover the daisychain structure. In this section, we first present the notation and the SAT model. We next show that even though the attacks described thus far in this section cannot unscramble the scanned-out bitstreams, the SAT attack can.

1) Notation and Formal Model: We adopt the attack model used by the brute-force and differential-analysis attacks. Let the ith microelectrode on the MEDA biochip be denoted as e_i, and the naming order is as shown in Fig. 8. Assume the attacker obtains vectors $\overrightarrow{X}$ and $\overrightarrow{Y}$ for each observation from a MEDA biochip with N microelectrodes, where $\overrightarrow{X} \subseteq \mathbb{B}^N$, $\overrightarrow{Y} \subseteq \mathbb{B}^N$, and $\mathbb{B} = \{0, 1\}$. The vector $\overrightarrow{X}$ represents the droplet locations; if a component $x_i = 1$, the ith microelectrode is under a droplet; if a component $x_j = 0$, the jth microelectrode is not covered by a droplet. The vector $\overrightarrow{Y}$ represents the scanned out bitstream. An example is shown in Fig. 8(a), where $\overrightarrow{X} = (1, 1, 0, 1, 1, 0, 0, 0, 0)$ and $\overrightarrow{Y} = (1, 1, 0, 0, 1, 1, 0, 0, 0)$.

We define a Boolean matrix $K_{N,N}$ to be a key matrix, where $k_{N,N} = [L(e_1), L(e_2), \ldots, L(e_N)]^T$. For a scanned-out bitstream, each microelectrode should map to a unique bit in it. The vector $L(e_i) \subseteq \mathbb{B}^N$ represents bits in $\overrightarrow{Y}$ that may map to the microelectrode e_i. Let l_i^j be the jth component of $L(e_i)$. If a component $l_i^j = 1$, the microelectrode e_i can be mapped to the jth bit in the bitstream. Without any observations, $l_i^j = 1$ for all i, j because each microelectrode can be mapped to any bit in the bitstream.

When $K_{N,N}$ is set to the correct matrix, $\overrightarrow{X}^T \cdot K = \overrightarrow{Y}^T$. For P consecutive observations, $\overrightarrow{X}_i^T \cdot K = \overrightarrow{Y}_i^T\ \forall i \in \{1, 2, \ldots, P\}$. If an attacker observes a small subset of $\overrightarrow{X}$ and $\overrightarrow{Y}$, that is, P is small, the attacker might obtain a key matrix $\widehat{K_{N,N}}$ that satisfies all P observations; however, $\widehat{K_{N,N}}$ may not apply to the new $(P + 1)$th observation [21]. Therefore, this key is not correct.

2) SAT Attack: Problem Formulation: Let the correct key be $K_{N,N}^c$. The attacker's goal is to find $K_{N,N} = K_{N,N}^c$. This is equivalent to solving the quantified Boolean formula: $\exists K_{N,N}\ \forall \overrightarrow{X}\ \forall \overrightarrow{Y} : \overrightarrow{X}^T \cdot K_{N,N} = \overrightarrow{Y}^T$.

3) Algorithm Overview: We can use a SAT solver to generate a $K_{N,N}$ to satisfy all observations. However, the solution $K_{N,N}$ may not satisfy a new observation. Consider the example in Fig. 8 from Section IV-C. Three observations are made during protocol execution.

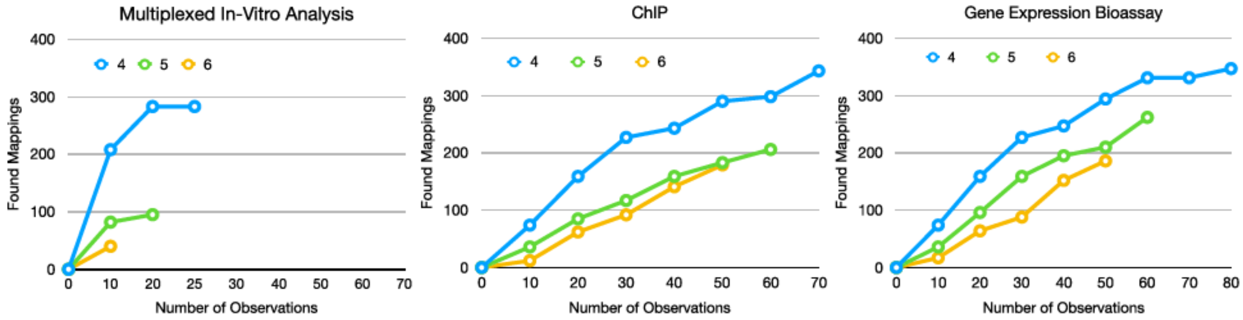


Fig. 10. Number of discovered microelectrode mappings using differential analysis on three real-life bioassays. Snapshots were made during bioassay executions in 4–6 clock-cycle intervals.

Let us consider the scenario when the attacker obtains one observation as shown in Fig. 8(a), that is, $\overrightarrow{X_1} = (1, 1, 0, 1, 1, 0, 0, 0, 0)$ and $\overrightarrow{Y_1} = (1, 1, 0, 0, 1, 1, 0, 0, 0)$. When these vectors are fed to the SAT solver with the proposed model, the solver returns possible solutions. However, some of these solutions are not legal for the daisychain structure. For example, in this case, the solver returns two illegal solutions $K'_{9,9}$ and $K''_{9,9}$, where

$$K'_{9,9} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$K''_{9,9} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The matrix $K'_{9,9}$ is not the key matrix $K^c_{9,9}$ because $\exists L(e_i)$ such that $\sum_j l_i^j \neq 1$, where l_i^j is the jth component in $L(e_i)$. This implies that the ith microelectrode can be mapped to more than one bit in the bitstream. As for the matrix $K''_{9,9}$, even though $\sum_j l_i^j = 1 \; \forall L(e_i)$, $K''_{9,9}$ is not a legal matrix because it represents an infeasible daisychain structure. Let the daisychain structure of $K''_{9,9}$ be denoted as a vector dc. Note that dc $= (e_1, e_5, e_3, e_6, e_4, e_2, e_7, e_8, e_9)$ because e_1 is related to the first bit of the bitstream, and e_5 is related to the second bit. For a legal daisychain structure, two adjacent components e_x and e_y in dc should be physically adjacent on the MC array because e_x should connect to e_y in the array to form a daisychain. While e_1 and e_5 are adjacent bits in dc, they are not physically adjacent on the MC array. Illegal key matrices can be automatically eliminated by: 1) checking whether the sum of each row equals 1 and 2) checking whether the components in the translated dc are adjacent.

By pruning away the illegal solutions obtained by the SAT solver, we can retain legal key matrices. Two legal matrices are listed below

$$K^3_{9,9} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$K^4_{9,9} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Both are legal for observations $\overrightarrow{X_1}$ and $\overrightarrow{Y_1}$. We cannot decipher the correct structure (i.e., the key matrix) because we have a limited number of observations.

Let us consider the case when the attacker is able to make two more observations, that is, the observations based on Fig. 8(b) and (c). These observations are $\overrightarrow{X_2} = (0, 1, 1, 0, 1, 1, 0, 0, 0)$, $\overrightarrow{X_3} = (0, 0, 0, 0, 1, 1, 0, 1, 1)$, $\overrightarrow{Y_2} = (0, 1, 1, 1, 1, 0, 0, 0, 0)$, and $\overrightarrow{Y_3} = (0, 0, 0, 1, 1, 0, 0, 1, 1)$. Using the SAT solver, we can obtain the key matrix $K^4_{9,9}$. Since this is the only legal key matrix, the secret key is deciphered with only three observations, that is, $K^4_{9,9} = K^c_{9,9}$. Recall that differential analysis cannot decipher the structure of the daisychain using the same three observations. Furthermore, this key matrix satisfies all new observations. Fig. 11 outlines the SAT attack to discover the mapping between bits in the bitstream and the MCs in the MEDA biochip.

An attacker requires three observations to decipher the daisychain structure in a 3 × 3 MEDA biochip. This example shows that the simple daisychain structure cannot defend against the SAT attack. Hence, we propose an enhanced daisychain structure.

V. Defense Against the SAT Attack

The SAT attack works on the simple daisychain because each microelectrode is associated with one fixed bit in the actuation and in the scanned-out sequence. Therefore, we propose to scramble the actuation and the scan-out bitstreams in each operation cycle, thwarting the SAT attack.

```
Input: $N$ observations of $\overrightarrow{X_i}$ and $\overrightarrow{Y_i}$
Output: Legal and possible key matrices $K^c_{N,N}$
Solutions = {}; Initialize a SAT solver $SAT$; $i = 0$;
for $i < N$ do
    $SAT(\overrightarrow{X_i}^T \cdot K_{N,N} = \overrightarrow{Y_i}^T)$;
end for
for a solution $K_{N,N}$ from $SAT$ do
    if ($K_{N,N}$ is legal) then
        Add $K_{N,N}$ to $Solutions$
    end if
end for
return $Solutions$;
```

Fig. 11. Pseudocode for the SAT attack.

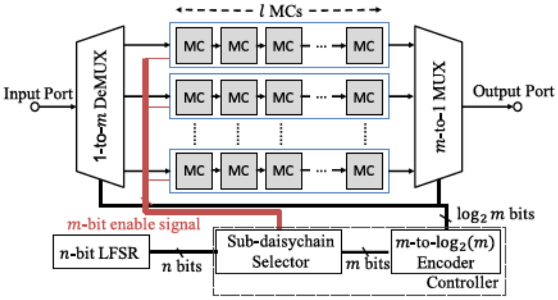


Fig. 12. Enhanced daisychain structure.

A. Reconfigurable Daisychain Structure

The enhanced daisychain structure divides the original daisychain into Q smaller subdaisychains, and the subdaisychains are enabled one after another based on the states of an n-bit linear feedback shift register (LFSR). The enhanced daisychain is shown in Fig. 12. When the biochip is in the actuation mode, only the enabled subdaisychain receives the scanned in bitstream. When the biochip is in the scan-out mode, only the enabled subdaisychain scans out the sensor data. The clock frequency of the LFSR is different from that of the daisychain. Let the clock cycle of the LFSR be clk_L, the clock cycle of the daisychain be clk_D, and the length of each subdaisychain be l. The relationship between the two clock cycles can be expressed as

$$clk_D = l \times clk_L.$$

Example: Consider an enhanced daisychain with three subdaisychains (each with 600 MCs) and a 4-bit LFSR. The states of the LFSR are shown in Table II. The controller is designed such that each LFSR state enables only one of the subdaisychains. The controller contains two modules: a subdaisychain selector and a 4-to-2 encoder. Let L_i be the ith pin from the LFSR and O_i be the ith output pin of the subdaisychain selector. This controller can be expressed as

$$\begin{aligned}O_1 &= (\overline{L_1\ L_2\ L_3}L_4) + (\overline{L_1\ L_2}L_3\overline{L_4}) + (L_1\overline{L_2}L_3\overline{L_4})\\ &\quad + (L_1\ L_2\overline{L_3}L_4) + (\overline{L_1}L_2\ L_3\ L_4)\\ O_2 &= (L_1\overline{L_2\ L_3\ L_4}) + (L_1\ L_2\overline{L_3\ L_4}) + (\overline{L_1}L_2L_3\overline{L_4})\\ &\quad + (L_1\overline{L_2}L_3\overline{L_4}) + (\overline{L_1\ L_2}L_3\ L_4)\\ O_3 &= (\overline{L_1}L_2\overline{L_3\ L_4}) + (L_1\overline{L_2\ L_3}L_4) + (L_1\overline{L_2}L_3\ L_4)\\ &\quad + (L_1\ L_2\ L_3\overline{L_4}) + (L_1\ L_2\ L_3\ L_4).\end{aligned}$$

Here, we consider the scenario when the biochip is in the sensing (i.e., scan-out) mode. The same working principle applies to the actuation (i.e., scan-in) mode. When the biochip is activated, the LFSR is initialized with a seed of 0001 from an on-chip memory, and the controller enables the first subdaisychain based on this LFSR state. When in the scanning mode, the MEDA biochip is operated at 1-MHz frequency [10], and the clock cycle of the D flip-flops (DFFs) in MCs is 1 μs. Because the LFSR must change its state after all the bits are scanned into a subdaisychain, the LFSR clock cycle is set as 600 μs. After the first 600 bits of the sensor data are scanned out from the first subdaisychain, the LFSR enters the next state 1000, and the second subdaisychain is enabled by the controller. For the subsequent LFSR state, the third subdaisychain is enabled. After three states, the full sensor bitstream is scanned out. Let the subbitstreams from the first, the second, and the third subdaisychain be sub_1, sub_2, and sub_3, respectively. The first full scanned-out bitstream can be denoted as (sub_1, sub_2, sub_3). Similarly, the second full bitstream can be scanned out as (sub_1, sub_3, sub_2). The remaining full bitstreams are shown in Table II.

TABLE II

EXAMPLE OF THE ENHANCED DAISYCHAIN

LFSR states	Enabled sub-daisychain	Scanned-out bitstream
0001, 1000, 0100	1, 2, 3	(sub_1, sub_2, sub_3)
0010, 1001, 1100	1, 3, 2	(sub_1, sub_3, sub_2)
0110, 1011, 1010	2, 3, 1	(sub_2, sub_3, sub_1)
1010, 1101, 1110	2, 1, 3	(sub_2, sub_1, sub_3)
1111, 0111, 0011	3, 1, 2	(sub_3, sub_1, sub_2)

In the above case, the initial seed is programed in a tamper-resistant memory as 0001 [22], [23]. For each fabricated MEDA biochip, the initial seed is programed differently. For example, if the initial seed is chosen as 1010, then the first full bitstream is (sub_2, sub_1, sub_3). Therefore, by assigning distinct initial seeds to different MEDA biochips, the order of scrambled patterns for each biochip can be unique.

B. Bitstream Translation

For the programmable daisychain structure described in Section III, a translation matrix K is sufficient to translate all the actuation bitstreams from the synthesis result. However, for the enhanced daisychain structure, a new translation procedure is needed because the daisychain structure is reconfigured during bioassay execution.

We first introduce the translation procedure for the actuation bitstreams. Assume that an enhanced daisychain structure has a total of P scrambled patterns. After high-level synthesis, all droplet locations during the bioassay execution are determined. Assuming that there are a total of T time stamps for the bioassay execution, the determined droplet locations at time t are denoted as a vector D_t, where $1 \leq T \leq T$. To translate all the determined droplet locations D_t into bitstreams, the translation tool that is loaded on the connected processor stores a series of P translation matrices, K_1 to K_P. Matrix K_i is associated with the ith scrambled pattern. For example, if the subdaisychains are enabled in the order as described in Table II, K_1 translates the bitstreams in an order of (sub_1, sub_2, sub_3). Let the translated bitstream at time t be B_t. Note that B_1 can be derived using (1) as $B_1 = D_1 \cdot K_1$. All the bitstreams are translated according to a distinct translation matrix K_i. The overall actuation-bitstream translation procedure is shown in Fig. 13.

The translation procedure for the sensed bitstreams is similar to that of the actuation bitstreams, but the translation matrices used here are different. Similar to the notation used in Section III, we define the sensed bitstream at time t as $\widehat{B_t}$ and the translated droplet location at time t be $\widehat{D_t}$, where

```
Input: $T$ droplet locations ($D_1$ to $D_T$) and $P$ translation matrices ($K_1$ to $K_P$)
Output: A list of translated bitstreams $TB$
$TB = []$;
for $i$ in range(0, $T$) do
    $j = i \bmod P$;
    $B_i = D_i \cdot K_j$;
    $TB$.add($B_i$)
end for
return $TB$
```

Fig. 13. Pseudocode for the actuation-bitstream translation.

```
Input: $T$ sensed bitstreams ($\widehat{B_1}$ to $\widehat{B_T}$) and $P$ translation matrices ($K_1$ to $K_P$)
Output: A list of translated droplet locations $TD$
$TD = []$;
for $i$ in range(0, $T$) do
    $j = i \bmod P$;
    $\widehat{D_i} = \widehat{B_i} \cdot K_j^T$;
    $TD$.add($\widehat{D_i}$)
end for
return $TD$
```

Fig. 14. Pseudocode for the sensed-bitstream translation.

$1 \leq t \leq T$. According to (2), the translation matrix for the sensed bitstreams is the transpose of the matrix for the actuation bitstreams. Therefore, because the scrambled pattern for the sensed bitstreams are the same as that for the actuation bitstreams, the droplet location at time t can be expressed as $\widehat{D_t} = \widehat{B_t} \cdot K^T$. For example, assume that the subdaisychains are enabled in the order as described in Table II. The bitstream $\widehat{B_1}$ is scanned based on the order of (sub_1, sub_2, sub_3), and this order is the same as that in the example in actuation-bitstream translation (in the previous paragraph). Therefore, the translated $\widehat{D_1}$ can be derived as $\widehat{D_1} = \widehat{B_1} \cdot K_1^T$. The overall sensed-bitstream translation procedure is shown in Fig. 14.

VI. Security Assessment of the Proposed Defense

We consider attacks on the enhanced daisychain structure when: 1) an attacker does not know the architecture of the enhanced daisychain structure and 2) an attacker knows the structure of the LFSR.

A. Attacks Without Knowing the Architecture

Because the attacker does not know the architecture, we assume s/he can only use the SAT attack with all observations. The subbitstream order in the scanned-out bitstreams is randomized based on the LFSR states, that is, the bits in the scanned-out bitstreams are not uniquely mapped to specific microelectrodes. The SAT attack in Section IV is effective because each bit in the bitstream is associated with a unique microelectrode. When we randomize the association of each bit with a microelectrode in the scanned-out bitstreams, the SAT constraints from the observations become inconsistent. The SAT solver cannot generate a satisfying solution. To establish this claim, we present the following theorem. The proof is given in Appendix.

Theorem 1: When we adopt the enhanced daisychain, the SAT attack cannot unravel the structure of the daisychain based on the continuous observations of bioassay execution.

B. Attacks With the Knowledge of the Architecture

We next assume that an attacker knows the structure of the LFSR as well as the number of subdaisychains, that is, the attacker knows the number of scrambled patterns. For example, s/he can de-layer the MEDA biochip in order to discover the architecture of the MEDA biochip, including the LFSR size. Consequently, the attacker can employ the same SAT-based attack on the scanned-out bitstreams that correspond to the same scrambled pattern.

We quantify the effectiveness of our defense by defining a probability metric P, which describes the probability that an attacker can discover all the scrambled patterns corresponding to an enhanced daisychain structure. Let the number of microelectrodes in a MEDA biochip be N_E, the discovered mappings obtained by the attacker be d, and the number of scrambled patterns in the proposed defense be N_P. According to the analysis in Section IV, the best-case scenario is that $(N_E - 2^{d-1})!$ possible key matrices are generated for a pattern. For an attacker to unscramble all N_P scrambled patterns, the probability P is derived as

$$P = \left(\frac{1}{(N_E - 2^{d-1})!}\right)^{N_P}. \tag{3}$$

From this equation, to successfully defend the attacks described in Section IV, we have two approaches: 1) increasing the number of scrambled patterns (N_P) and 2) reducing the discovered mappings (d).

1) Increasing the Number of Scrambled Patterns: Assume that N_o observations are obtained from a bioassay execution, the LFSR has S states, the enhanced daisychain structure has D subdaisychains, and there are a total of N_P scrambled patterns [i.e., the ith bitstream and the $(i + N_P)$th bitstream correspond to the same scrambled pattern]. Consider the example provided in Table II. In this example, $S = 15$, $D = 3$, and the number of scrambled patterns is $N_P = S/D = 5$. If a lengthy bioassay with many operations is executed on a MEDA biochip using the enhanced daisychain, an attacker can acquire a sufficient number of observations to unscramble the bitstreams.

Example: we consider the execution of the gene-expression analysis bioassay, which has 317 actuation bitstreams, that is, $N_o = 317$. For each scrambled pattern, an attacker can obtain $N_o/N_P = 317/5 \approx 63.4$ observations. In our simulation, which will be described in detail in Section VII, an attacker can discover all the mappings using the SAT attack with only 45 observations. Therefore, the attacker can unscramble all the bitstreams, that is, $P = 1$.

The above example shows that P increases as N_o increases. Conversely, the fewer observations an attacker can make, the less likely it is that s/he can unscramble the bitstreams. Therefore, a secure daisychaining should incorporate many scrambled patterns, that is, N_P should be sufficiently large. Since $N_P = S/D$, an ideal daisychain structure should contain a large number of the LFSR states (S) and a small number of subdaisychains (D). Consider an enhanced daisychain such that an LFSR has 1023 states and 3 subdaisychains, that is, $S = 1023$ and $D = 3$. The number of the scrambled patterns is $N_P = S/D = 341$. As a result, for a bioassay that allows N_o observations, an attacker can acquire on average $N_o/341$ observations for each scrambled pattern.

We consider three representative bioassays (multiplexed *in vitro* [18], ChIP [19], and gene-expression analysis [20]) and these offer 109, 285, and 317 observations, respectively, based on their corresponding actuation sequences. Assuming that an attacker aims at unscrambling the bitstreams of the gene-expression protocol (with the largest number of observations among three bioassays), s/he can only obtain $317/341$ observations on average for a scrambled pattern, which is less than 1 for each scrambled pattern. Therefore, the attacker cannot unscramble the bitstreams of these existing bioassays. To unscramble a series of bitstreams that correspond to a scrambled pattern, our experiments (reported in Section VII) show that there should be at least 45 observations, that is, $N_o/341 \geq 45$. As a result, for the attacker to unscramble the bitstreams on this enhanced daisychain structure, s/he needs at least $N_o \geq 13\,860$ observations from many bioassay executions.

2) Reducing the Discovered Mappings: To prevent an attacker from discovering microelectrode mappings from a MEDA biochip, we further propose a physical mechanism that periodically disables and resets the daisychain so that only a limited number of observations can be made for a single MEDA biochip. A counter is added and integrated with the enhanced daisychain, and the counter records the number of observations that the biochip scans out. Once the number of observations exceeds a threshold, for example, 2000, which is much larger than that of representative bioassays (<350), the LFSR is initialized with an all-zeros seed. The daisychain is no longer functional, and the biochip cannot execute any bioassay.

Based on the proposed structure with the counter, an attacker can get at most $2000/341 \approx 6$ observations for each scrambled pattern. To unscramble all $N_P = 341$ series of bitstreams (from all scrambled patterns), the probability $P = (1/1768!)^P = (1/1768!)^{341}$. As a result, it is extremely unlikely that an attacker can unscramble the bitstreams using a single MEDA biochip with the enhanced daisychain.

Now consider what might happen if an attacker fails in unscrambling bitstreams using one MEDA biochip. S/he may want to reproduce the attack on another MEDA biochip. Recall that each MEDA biochip has unique scrambled patterns based on the different seeds for the LFSR, that is, the bitstreams of a bioassay is different for each MEDA biochip. Even if the attacker can afford the cost of new biochips, the acquired bitstreams of the same bioassay are different for the new biochips. Therefore, the attack information obtained from a biochip cannot be combined with that from attacking another biochip. Hence, the probability of unscrambling a bioassay does not increase using multiple MEDA biochips.

VII. Experimental Results

We simulated the execution of three real-life biochemical protocols on a 60×30 MEDA biochip: multiplexed *in vitro* diagnosis [18], gene-expression analysis [20], and ChIP [19]. We consider normal-sized droplets which occupy 4×4 microelectrodes. We implemented the simulator using Python on a workstation with 2.5 GHz Xeon processor and 2 GB memory, and we employed the Minisat solver for the SAT attack [24]. During bioassay execution, snapshots of droplet locations are made during a given clock-cycle interval C, where $4 \leq C \leq 6$. The snapshots and the scanned-out bitstreams are used as inputs to the SAT solver, that is, $\overrightarrow{X_i}$ and $\overrightarrow{Y_i}$. Because the numbers of fluidic operations in the above bioassays are different, the numbers of generated observations are also different. Details are shown in Table III.

TABLE III
Details of the Biomolecular Protocols

	Number of Operations	Number of Operation Cycles
Multiplexed in-vitro bioassay	12	109
ChIP	16	285
Gene-expression analysis	18	317

TABLE IV
Results for Simple and Advanced Daisychain Scrambling

Simple Daisychain scrambling	Multiplexed In-Vitro Analysis				SAT attack cannot unscramble the advanced daisychain no matter the number of observations; see Theorem 1.
	Clock-Cycle Interval	4	5	6	
	Number of Observations	27	21	18	
	Number of Found Keys	342	487	754	
	Chromatin Immunoprecipitation Protocol				
	Clock-Cycle Interval	4	5	6	
	Number of Observations	71	57	47	
	Number of Found Keys	1	1	114	
	Gene-Expression Analysis				
	Clock-Cycle Interval	4	5	6	
	Number of Observations	79	63	52	
	Number of Found Keys	1	1	28	

The simulation results for a simple daisychained MEDA biochip are shown in Table IV. We record the number of legal key matrices from the SAT solver after all the observations are fed to the solver. Even though the SAT attack cannot reveal the daisychain structure using the observations associated with the multiplexed *in vitro* bioassay, the SAT attack can reveal the daisychain structure using observations from ChIP or gene-expression analysis. The results confirm that the simple daisychain structures can be deciphered using the SAT attack.

We repeated this experiment for a MEDA biochip with an enhanced daisychain. We considered three subdaisychains and employed a 10-bit LFSR. We used the randomized bitstreams as the input to the SAT model across the observations. As expected from Theorem 1, the SAT attack failed to provide any legal key matrices for all three real-life biomolecular protocols.

VIII. Discussion

In this section, we examine the timing and area overheads associated with the enhanced daisychain. We next present comparisons between our enhanced security solution with existing security methods.

A. Timing, Area, and Power Overheads

Timing Overhead: The LFSR and the controller in the enhanced daisychain are designed in such a way that the scan mode does not require extra time. Recall that we assume each subdaisychain contains l MCs, and thus the LFSR changes its state when l bits of the bitstream are scanned out. When the LFSR state changes, the controller enables one subdaisychain and disables the previous subdaisychain simultaneously. Therefore, bitstream scanning operates as in normal daisychaining, that is, the enhanced daisychain does not incur any time overhead.

As an example, consider the first three scanned-out bitstreams from the enhanced daisychain in Table II. We provide the corresponding timing diagram that corresponds to these bitstreams and the enable signals for the three subdaisychains in Fig. 15. Each bitstream can be divided into three parts, and each part is scanned out from a subdaisychain. At any time, only a subdaisychain is enabled, and the sensed data is

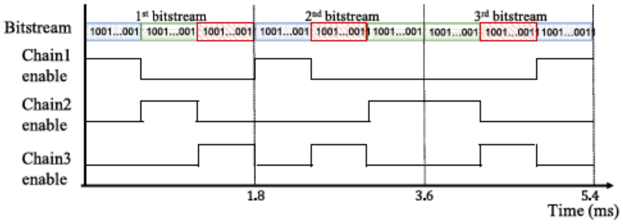


Fig. 15. Timing analysis results for the enhanced daisychain of Table II, where the scanning clock cycle is 1 μs.

TABLE V

TRANSISTOR COUNTS FOR THE ADDED MODULES

Module	10-Bit LFSR	Decoder	MUX	deMUX	Counter	Total
Number of Transistors	126	72	18	18	176	410

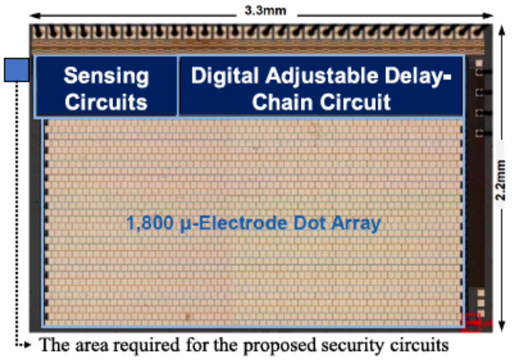

3.3mm
2.2mm
Sensing Circuits
Digital Adjustable Delay-Chain Circuit
1,800 μ-Electrode Dot Array
The area required for the proposed security circuits

Fig. 16. Microphoto of a state-of-the-art MEDA biochip [12]. Sensing modules have already been integrated in the MEDA biochip.

scanned out from the enabled subdaisychain. We can see that, by carefully designing the enable patterns, we can ensure that the enhanced daisychain does not incur any time overhead.

Area Overhead: The addition of the LFSR and the controller introduces area overhead. The area overhead depends on the size of the LFSR. Increasing the LFSR size increases the number of states and thus offers security, but it also has a larger area overhead due to the complex decoder in the controller. In Table V, we present the number of CMOS transistors for the added circuits if a 10-bit LFSR is used in the enhanced daisychain. A total of 410 transistors need to be integrated. The MC of a state-of-the-art MEDA biochip consists of 37 transistors and is fabricated in an area of $50\times50\,\mu\text{m}^2$ [12]. Since the security circuits are fabricated (along with the MC array) using the same 0.35 μm fabrication process as the MCs, we can estimate the required area for the extra circuits to be $0.05\times0.05\times410/37 = 0.028\,\text{mm}^2$. A microphoto of the state-of-the-art MEDA biochip is shown in Fig. 16. The area of the MEDA biochip is $3.3\times2.2 = 7.26\,\text{mm}^2$. Hence, the area overhead of the security modules is $0.028/7.26\times100\% = 0.4\%$. Several sensing circuits have been fabricated with the MC array in MEDA biochips, such as high-resolution droplet sensing [10], and these circuits are shown in Fig. 16. Likewise, the extra circuits for the enhanced daisychain can be placed around the MC array so that they do not affect the MC array and the fluidic operations.

Power Overhead: In order to estimate the power overhead of the proposed defense, we simulated a MEDA biochip with 1800 microelectrodes using HSPICE. Two designs were considered: an original daisychain structure and an enhanced daisychain structure. The defense circuits in the enhanced daisychain structure are composed of a 4-bit LFSR and an associated controller, which has been described in the example in Section V. The added security circuits are active only when the biochip functions in scanning mode. Therefore, when the biochip is in the sensing mode or the actuation mode, the added circuits do not contribute to power consumption. While shifting a bitstream of length 1800, the power consumptions for the added security circuits and the MEDA are listed in Table VI. Even though the added circuits introduce extra power of 1.18 mW, the overall power consumption of the defense daisychain structure is lower when compared to the original daisychain structure. This is because less DFFs in the MCs of MEDA are enabled and shifting data. In the original design, when we shift a bitstream, the DFFs in all 1800 MCs are toggling. On the other hand, the enhanced daisychain structure has several shorter subchains. When we shift a bitstream, only a subchain is enabled, and only those DFFs in the subchain shift a segment of the bitstream. In the example of our simulation, only 600 DFFs (of 1800 DFFs) are enabled and shifting data. Therefore, the power consumption of MEDA can be greatly reduced.

TABLE VI

POWER CONSUMPTION (IN mW) FOR THE ADDED CIRCUITS AND THE MEDA

	Security Circuits	MEDA	Total Power
Original daisychain	N/A	168.73	168.73
Enhanced daisychain	1.18	19.44	20.62

B. Comparison With Prior Work in Scan-Chain Security

Related to the daisychains in this article, scan-chains of flip flops in an IC are used for testing. Scan chains enhance controllability and observability. However, they allow a malicious adversary access to confidential data [25]. Therefore, countermeasures have been developed to enable authorized users to operate the test mode. For example, VIm-Scan [26] requires users to scan in the secret keys in several iterations to unlock the chip for testing. In our enhanced daisychaining method, the correct daisychain structure is the secret. Without knowing the correct daisychain, the user cannot exploit the biochip.

Another authorization-based method, named Lock and Key, was proposed in [27]. This method requires that the chip with scan chains operate in two modes: secure and insecure. The chip is initialized (after reset) in the insecure mode. When the chip is in the insecure mode, the scanned-out results are randomly scrambled by the states of an LFSR and the scanned-out data is unpredictable. However, when a correct key is applied one clock cycle after the initialization, the chip switches to the secure mode and allows predictable operation of the scan chains. The chip remains in the secure mode until it is reset. Unauthorized users cannot exploit the scan chains without knowing the secret key. Even though this work and our enhanced daisychain employ LFSRs, there are two main differences.

1) The initial seed is programed in the biochip so that the order of scrambling is fixed. This seed is provided by the authorized users for Lock and Key.
2) The LFSR in the enhanced daisychain determines the order of scan-in/scan-out bitstreams. On the other hand, the LFSR in [27] enables specific scan chains for testing.

Scrambling of scan chain was proposed in [28]. Here, extra logic circuits are added in the scan chain. The scanned-out

TABLE VII

SAT ATTACK ON LOGIC LOCKING AND DAISYCHAIN SCRAMBLING

	Logic locking	Daisychain scrambling
Protection	Add key inputs to corrupt digital design outputs.	Scramble daisychain ↔ actuation/scan-out bitstream map.
SAT inputs	Search for distinguishing inputs (DI) and create SAT constraints.	Observe droplet locations and the scanned out bitstreams during bioassay executions and create SAT constraints.

data is locked in a specific way that only authorized users can comprehend. Similar to our enhanced daisychain scrambling, Hely *et al.* [28] also scramble the scanned-out result. However, it uses multiplexers in the scan chains to scramble the output. Our solution uses an LFSR to scramble the scan-in and scan-out bitstreams across several operation cycles.

Recently, a SAT-based attack, named scanSAT, was proposed to attack obfuscated scan chains [29]. This articlefirst models the obfuscated scan chain as a combinational circuit with logic locking. It then launches the SAT attack on the equivalent combinational circuit. The experimental results in [29] show that obfuscated scan chains can be successfully de-obfuscated even in the presence of a scan compression infrastructure. However, this attack cannot be used to decipher our proposed MEDA daisychain due to two main reasons.

1) The targeted chain structures are different. The model in scanSAT assumes that the order of the flip flops in scan chains is fixed. However, the order of the flip flops in our proposed daisychain changes dynamically during bioassay execution. Therefore, there is no equivalent combinational circuit for the scanSAT attack.
2) Attack objectives are different. The scanSAT attack aims at retrieving the information hidden in the obfuscated netlist. On the other hand, the proposed SAT attack in Section IV aims at discovering the mapping relationship between actuation bitstreams and microelectrodes.

C. Comparison With (Un)related Work on SAT Attack on Logic Locking

Logic locking has been proposed to combat IC piracy and counterfeiting [30]. Logic locking inserts extra gates—the key gates—into the circuit to potentially corrupt the functionality [30]–[32]. Logic locking of digital designs is not related to the problem of scrambling of the mapping between the actuation and sensor bitstreams and the MCs in a biochip. Different SAT formulations are used to attack logic locking and biochip bitstream scrambling [21]. Table VII summarizes the key differences between the SAT attacks on digital designs and on MEDA biochip daisychain structures.

D. Comparison With Software Encryption Methods

Many well-known data encryption methods, such as Advanced Encryption Standard (AES), are used today [33], [34]. The actuation bitstreams of bioassays can also be encrypted using these methods. However, these methods are not adequate for the threat model described in Section IV in microfluidic systems. As mentioned in Section II, a MEDA biochip is controlled by a microcontroller unit connected to a synthesis processor. The processor runs CAD tools to synthesize bioassays and generates the translated actuation sequence. Even though the actuation bitstreams are encrypted and stored in the processor, to execute the bioassays, the bitstreams need to be decrypted before being sent to the MEDA biochip. The decrypted actuation sequences are sent to the controller and transmitted to the scan-in port of the MEDA biochip. The communication channels between the processor and the biochip may be insecure and the attacker can eavesdrop on the channels. The attacker can acquire the decrypted bitstreams and comprehend the encryption method. Therefore, the encryption cannot thwart the attack. On the other hand, the scrambled actuation bitstreams based on our proposed defense are fed directly to the biochips. Even if the attacker can observe the bitstreams, the bioassay IP has been proven to be secure (in Section VI).

E. Security Concerns in Biochip Supply Chain and Probing

Previous work has pointed out that supply chain attacks may compromise VLSI hardware integrity [35]. Therefore, we consider a case when a malicious adversary launches supply chain attacks in the biochip industry, for example, the attacker colludes with rogue insiders in the biochip foundry to leak the biochip design [36]. Under such an attack, the malicious adversary can comprehend the design of the fabricating biochip. Furthermore, the attacker may introduce back doors in the biochip to leak sensitive information, that is, bioassay IP in the form of bitstreams [37]. However, the supply chain attack is not much of a concern for bioassay IP protection. In the biochip supply chain, foundries do not have access to the bioassay IP [38], [39]. Instead, bioassay IP is stored in the synthesis processor that is associated with the biochips. When bioassays are being executed, the IPs (in the form of bitstreams) are scanned to the MEDA biochips. Assuming that, during bioassay execution, the bitstreams are leaked through the back door to the attacker, our proposed method can thwart such attack because the bitstreams are scrambled randomly. Without understanding the patterns, the attacker cannot RE the bitstreams, that is, the attacker cannot steal the IPs.

Prior work has also pointed out that probing attacks may compromise VLSI hardware integrity [40], [41]. In this article, we only consider nonintrusive attacks. However, if an attacker is able to delayer MEDA biochips in a sophisticated way such that the biochip is still functional, s/he can observe the signals in the daisychain. As a result, the attacker can discover the daisychain structure and RE the bio-protocol. Because the underlying circuits in MEDA are also fabricated using the semiconductor technology, the probing attack for MEDA biochips is similar to that for ICs. A biochip developer can use the defenses proposed for ICs to secure MEDA biochip [40], [42]–[44].

IX. Conclusion

We have presented the use of daisychaining to secure IP protocols that are executed on DMFBs. We have also presented a SAT attack that can discover the simple daisychain scrambling, and simulations confirmed the effectiveness of the attack. An advanced daisychain structure was proposed to scramble the scan-in and scan-out data, and the simulations have shown that this defense is very effective against the SAT attack. We have also evaluated attacks on the defense as well as the overheads of using this defense.

Appendix

Proof of Theorem 1

This proof is based on some lemmas, which we prove first. Let X_i be the droplet location vector in observation i, Y_i be the scanned-out bitstream in observation i, and $K_{N,N}$ be a legal key matrix returned by the SAT solver.

Lemma 1: If the enhanced daisychain is used and $X_i = X_{i+1}$, then $Y_i \neq Y_{i+1}$.

Proof: The proof is straightforward because the bits associated with a specific microelectrode are randomized in the scanned-out bitstreams using the enhanced daisychain across the ith and the $(i+1)$th observations. Therefore, if $X_i = X_{i+1}$, that is, the on-chip droplets are not moved for the ith and the $(i+1)$th observations, and the scanned-out bitstreams $Y_i \neq Y_{i+1}$. □

Lemma 2: If X_i can be partitioned into two vectors X_i^1 and X_i^2 $(X_i = X_i^1 + X_i^2)$, then Y_i can also be partitioned into two vectors Y_i^1 and Y_i^2 $(Y_i = Y_i^1 + Y_i^2)$, where $(X_i^1)^T \cdot K_{N,N} = (Y_i^1)^T$ and $(X_i^2)^T \cdot K_{N,N} = (Y_i^2)^T$.

Proof:

Consider the following relationships:

$$\begin{aligned} X_i^T \cdot K_{N,N} &= (X_i^1 + X_i^2)^T \cdot K_{N,N} \\ &= (X_i^1)^T \cdot K_{N,N} + (X_i^2)^T \cdot K_{N,N} \\ &= (Y_i^1)^T + (Y_i^2)^T. \end{aligned} \tag{4}$$

Also, we know that $X_i^T \cdot K_{N,N} = Y_i^T$. From (2) and (3), we obtain $Y_i = Y_i^1 + Y_i^2$. □

Lemma 3: $X_i \cdot X_{i+1}^T \neq (0,0,0,\ldots,0)_N$, where $X_i \cdot X_{i+1}^T$ is the matrix multiplication of X_i and X_{i+1}.

Proof:

Let fp_i^d be the microelectrode set that a droplet d occupies at cycle i. Assuming the droplet exists across observations i and $i+1$. We know $fp_i^d \cap fp_{i+1}^d \neq \emptyset$. Therefore, $X_i \cdot X_{i+1}^T \neq (0,0,0,\ldots,0)_N$, where the vector X_i is another representation for the droplet locations. □

Theorem 1: When we adopt the enhanced daisychain, the SAT attack cannot unravel the structure of the daisychain based on the continuous observations of bioassay execution.

Proof: Assume that a legal key $K_{N,N}^c$ is returned by the SAT solver that attacks the enhanced daisychain. Therefore, for any two observations X_i and X_{i+1}, $X_i^T \cdot K_{N,N}^c = Y_i^T$ and $X_{i+1}^T \cdot K_{N,N}^c = Y_{i+1}^T$. Based on Lemma 3, $X_i \cdot X_{i+1}^T = I_{i+1}^i$, where $I_{i+1}^i \neq (0,0,0,\ldots,0)_N$. We can express X_i and X_{i+1} as $X_i = I_{i+1}^i + R_i$ and $X_{i+1} = I_{i+1}^i + R_{i+1}$, respectively, where $R_i \subseteq \mathbb{B}^N$ and $R_{i+1} \subseteq \mathbb{B}^N$. According to Lemma 2, $Y_i = Y_i^1 + Y_i^2$ and $Y_{i+1} = Y_{i+1}^1 + Y_{i+1}^2$. We find out that $Y_i^1 = (I_{i+1}^i)^T \cdot K_{N,N}^c = Y_{i+1}^2$, that is, for the same I_{i+1}^i from two observations i and $i+1$, the scanned-out bitstreams are the same. This contradicts Lemma 1. □

Acknowledgment

A preliminary version of this paper appeared in [1].

References

[1] T.-C. Liang, K. Chakrabarty, and R. Karri, "Programmable daisychaining of microelectrodes for IP protection in MEDA biochips," in *Proc. IEEE Int. Conf. Test*, 2019.

[2] (2014). *Neoprep NFS Library Prep With Digital Microfluidics by Illumina*. Accessed: Apr. 1, 2019. [Online]. Available: https://support.illumina.com/content/dam/illumina-support/documents/documentation/system_documentation/neoprep/neoprep-system-guide-15049720-01.pdf

[3] (2016). *FDA Advisors Approve of Baebies Seeker Analyzer for Newborns*. Accessed: Apr. 1, 2019. [Online]. Available: https://www.baebies.com/fda-advisors-back-approval-baebies-seeker-analyzer-newborns/

[4] T.-Y. Ho, K. Chakrabarty, and P. Pop, "Digital microfluidic biochips: Recent research and emerging challenges," in *Proc. 7th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, 2011, pp. 335–344.

[5] J. Fiske, D. Grissom, and P. Brisk, "Exploring speed and energy tradeoffs in droplet transport for digital microfluidic biochips," in *Proc. 19th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2014, pp. 231–237.

[6] W.-L. Chou, P.-Y. Lee, C.-L. Yang, W.-Y. Huang, and Y.-S. Lin, "Recent advances in applications of droplet microfluidics," *Micromachines*, vol. 6, no. 9, pp. 1249–1271, Sep. 2015.

[7] H. Chen, S. Potluri, and F. Koushanfar, "BioChipWork: Reverse engineering of microfluidic biochips," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Nov. 2017, pp. 9–16.

[8] C.-W. Hsieh, Z. Li, and T.-Y. Ho, "Piracy prevention of digital microfluidic biochips," in *Proc. 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 512–517.

[9] G. Wang, S.-K. Fan, and D. Teng, "Digital microfluidic operations on micro-electrode dot array architecture," *IET Nanobiotechnol.*, vol. 5, no. 4, pp. 152–160, Dec. 2011.

[10] K. Y.-T. Lai, Y.-T. Yang, and C.-Y. Lee, "An intelligent digital microfluidic processor for biomedical detection," *J. Signal Process. Syst.*, vol. 78, no. 1, pp. 85–93, Jan. 2015.

[11] Z. Li, K. Y.-T. Lai, P.-H. Yu, K. Chakrabarty, T.-Y. Ho, and C.-Y. Lee, "Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 3, pp. 612–626, Jun. 2017.

[12] Y. Ho *et al.*, "Design of a micro-electrode cell for programmable lab-on-CMOS platform," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2871–2874.

[13] Z. Zhong, Z. Li, and K. Chakrabarty, "Adaptive error recovery in MEDA biochips based on droplet-aliquot operations and predictive analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 615–622.

[14] (2018). *Laboratory Monitoring by Notable Labs*. Accessed: Feb. 5, 2020. [Online]. Available: http://tetrascience.com/case-studies/laboratory-monitoring-notable-labs

[15] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*. Boca Raton, FL, USA: CRC Press, 2005.

[16] J. L. Jacobsen, "Exact enumeration of Hamiltonian circuits, walks and chains in two and three dimensions," *J. Phys. A, Math. Theor.*, vol. 40, no. 49, pp. 14667–14678, Dec. 2007.

[17] O. Bodroza-Pantic *et al.*, "Enumeration of Hamiltonian cycles in some grid graphs," *MATCH Commun. Math. Comput. Chem.*, vol. 70, no. 1, pp. 181–204, 2013.

[18] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 4, pp. 1–32, 2008.

[19] M. Ibrahim, K. Chakrabarty, and U. Schlichtmann, "Synthesis of a cyberphysical hybrid microfluidic platform for single-cell analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 7, pp. 1237–1250, Jul. 2019.

[20] M. Ibrahim, K. Chakrabarty, and K. Scott, "Synthesis of cyberphysical digital-microfluidic biochips for real-time quantitative analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 5, pp. 733–746, May 2017.

[21] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2015, pp. 137–143.

[22] J. Leon, W. E. Boyd, S. He, and C. Krutzik, "Tamper-resistant memory device with variable data transmission rate," U.S. Patent Appl. 13/363 571, Jul. 19, 2012.

[23] S. R. Walmsley, "Tamper resistant shadow memory," U.S. Patent 7 188 282, Mar. 6, 2007.

[24] A. Ignatiev, A. Morgado, and J. Marques-Silva, "PySAT: A Python toolkit for prototyping with SAT oracles," in *Proc. SAT*, 2018, pp. 428–437, doi: 10.1007/978-3-319-94144-8_26.

[25] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on dedicated hardware implementations of data encryption standard," in *Proc. Int. Conf. Test*, Mar. 2005, pp. 339–344.

[26] S. Paul, R. S. Chakraborty, and S. Bhunia, "VIm-Scan: A low overhead scan design approach for protection of secret key in scan-based secure chips," in *Proc. 25th IEEE VLSI Test Symp. (VTS)*, May 2007, pp. 455–460.

[27] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing scan design using lock and key technique," in *Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (DFT)*, Mar. 2006, pp. 51–62.

[28] D. Hely, M. Flottes, F. Bancel, B. Rouzeyre, N. Berard, and M. Renovell, "Scan design and secure chip," in *Proc. IOLTS*, vol. 4, Jul. 2004, pp. 219–224.

[29] L. Alrahis, M. Yasin, H. Saleh, B. Mohammad, M. Al-Qutayri, and O. Sinanoglu, "ScanSAT: Unlocking obfuscated scan chains," in *Proc. 24th Asia South Pacific Design Autom. Conf. (ASPDAC)*, 2019, pp. 352–357.

[30] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," *IEEE Design Test. Comput.*, vol. 27, no. 1, pp. 66–75, Jan. 2010.

[31] J. A. Roy, F. Koushanfar, and I. L. Markov, "Ending piracy of integrated circuits," *Computer*, vol. 43, no. 10, pp. 30–38, Oct. 2010.

[32] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proc. 49th Annu. Design Autom. Conf. (DAC)*, 2012, pp. 83–89.

[33] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. New York, NY, USA: Springer, 2013.

[34] M. Hell, T. Johansson, and W. Meier, "Grain: A stream cipher for constrained environments," *Int. J. Wireless Mobile Comput.*, vol. 2, no. 1, p. 86, 2007.

[35] B. Liu and B. Wang, "Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2014, pp. 1–6.

[36] M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2013, pp. 819–823.

[37] C. E. Irvine and K. Levitt, "Trusted hardware: Can it be trustworthy?" in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 1–4.

[38] S. S. Ali, M. Ibrahim, J. Rajendran, O. Sinanoglu, and K. Chakrabarty, "Supply-chain security of digital microfluidic biochips," *Computer*, vol. 49, no. 8, pp. 36–43, Aug. 2016.

[39] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security assessment of cyberphysical digital microfluidic biochips," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 3, pp. 445–458, May 2016.

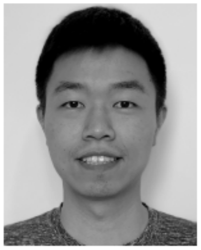
[40] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Proc. Annu. Int. Cryptol. Conf.* New York, NY, USA: Springer, 2003, pp. 463–481.

[41] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J.-P. Seifert, "Breaking and entering through the silicon," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2013, pp. 733–744.

[42] P. Mishra, S. Bhunia, and M. Tehranipoor, *Hardware IP Security and Trust*. New York, NY, USA: Springer, 2017.

[43] D. Shahrjerdi, J. Rajendran, S. Garg, F. Koushanfar, and R. Karri, "Shielding and securing integrated circuits with sensors," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2014, pp. 170–174.

[44] E. Amini, A. Beyreuther, N. Herfurth, A. Steigert, B. Szyszka, and C. Boit, "Assessment of a chip backside protection," *J. Hardw. Syst. Secur.*, vol. 2, no. 4, pp. 345–352, Dec. 2018.



Tung-Che Liang received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2014. He is currently pursuing the Ph.D. degree at Duke University, Durham, NC, USA.

He was with Synopsys Inc., Hsinchu, Taiwan, as a Research and Development Engineer. He was a Yield and Diagnosis Intern with Intel, Santa Clara, CA, USA, and a Design-for-Testing (DFT) Intern with NVIDIA Inc., Santa Clara. His current research interests include design automation and security for microfluidic systems.

Krishnendu Chakrabarty (Fellow, IEEE) received the B. Tech. degree from IIT, Kharagpur, Kharagpur, India, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1992 and 1995, respectively.

He is currently the John Cocke Distinguished Professor, the Department Chair of electrical and computer engineering, and a Professor of computer science with Duke University, Durham, NC, USA. His current research projects include testing and design-for-testability of integrated circuits and systems, microfluidic biochips, hardware security, machine learning for fault diagnosis and failure prediction, and neuromorphic computing systems.

Dr. Chakrabarty is a Fellow of ACM and AAAS. He is a Golden Core Member of the IEEE Computer Society and a Senior Member of the National Academy of Inventors.

Ramesh Karri (Fellow, IEEE) received the B.E. degree in electronics and communication engineering from Andhra University, Visakhapatnam, India, in 1985, and the Ph.D. degree in computer science and engineering from the University of California at San Diego, San Diego, CA, USA, in 1993.

He is currently a Professor of electrical and computer engineering with New York University (NYU), New York, NY, USA, where he co-directs the Center for Cyber Security. He also leads the Cyber Security thrust of the NY State Center for Advanced Telecommunications Technologies with NYU. He co-founded the Trust-Hub, Gainesville, FL, USA. He organizes the Embedded Systems Challenge and the global red-team-blue-team hardware hacking event. His research and education activities in hardware cybersecurity include trustworthy integrated circuits (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.