Security of Microfluidic Biochip: Practical Attacks and Countermeasures

HUILI CHEN, University of California, San Diego, USA
SEETAL POTLURI, North Carolina State University, USA
FARINAZ KOUSHANFAR, University of California, San Diego, USA

With the advancement of system miniaturization and automation, Lab-on-a-Chip (LoC) technology has revolutionized traditional experimental procedures. Microfluidic Biochip (MFB) is an emerging branch of LoC with wide medical applications such as DNA sequencing, drug delivery, and point of care diagnostics. Due to the critical usage of MFBs, their security is of great importance. In this article, we exploit the vulnerabilities of two types of MFBs: Flow-based Microfluidic Biochip (FMFB) and Digital Microfluidic Biochip (DMFB). We propose a systematic framework for applying Reverse Engineering (RE) attacks and Hardware Trojan (HT) attacks on MFBs as well as for practical countermeasures against the proposed attacks. We evaluate the attacks and defense on various benchmarks where experimental results prove the effectiveness of our methods. Security metrics are defined to quantify the vulnerability of MFBs. The overhead and performance of the proposed attacks as well as countermeasures are also discussed.

CCS Concepts: • **Security and privacy → Embedded systems security**; **Hardware reverse engineering**; **Malicious design modifications**; *Side-channel analysis and countermeasures*;

Additional Key Words and Phrases: Microfluidic biochip, security, hardware Trojans, hardware obfuscation, camouflaging, Trojan detection

ACM Reference format:
Huili Chen, Seetal Potluri, and Farinaz Koushanfar. 2020. Security of Microfluidic Biochip: Practical Attacks and Countermeasures. *ACM Trans. Des. Autom. Electron. Syst.* 25, 3, Article 27 (April 2020), 29 pages.
https://doi.org/10.1145/3382127

1 INTRODUCTION

Recent progress in Lab-on-a-Chip technology has emancipated humans from tedious experimental work by automating procedures. The innovation of microfluidic techniques facilitates system miniaturization and makes personal health care portable. Microfluidic Biochips (MFBs) benefit biochemical experiments and provide advantages such as low sample consumption, high throughput, and reduced human effort. MFBs have been increasingly commercialized by companies, including Illumina, Microfluidic Innovations LLC, and Fluidigm. MFBs are becoming innovative platforms in various fields such as next-generation sequencing, library preparation, and disease diagnosis.

Authors' addresses: H. Chen, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA, 92093; email: huc044@ucsd.edu; S. Potluri, North Carolina State University, 2200 Hillsborough Street, Raleigh, NC, 27695, USA; email: spotlur2@ncsu.edu; F. Koushanfar, University of California, San Diego, La Jolla, 9500 Gilman Drive, La Jolla, CA, 92093, USA; email: farinaz@ucsd.edu.

1084-4309/2020/04-ART27 $15.00
https://doi.org/10.1145/3382127

The lack of security consideration in the supply chain makes MFBs vulnerable to different attacks. Protection of MFBs is of significant importance, since MFBs are already being used in critical fields related to personal health. Multiple reasons have made protection of MFBs challenging. Microfluidics is a multidisciplinary field that spans electrical, chemical, biological, and optical domains. Attacks can be performed in each domain or even the combination of multiple domains, rendering the protection of MFBs difficult. The designer who synthesizes FMFBs might not have knowledge in all involved fields and thus becomes unaware of potential attacks. The foundries who manufacture the MFBs do not know the cost of adding defense mechanisms even if they realize the vulnerabilities of the fabricated chips. Also, there is no design method available with the foundry, which provides the trade-off between security and fabrication cost. As a result, existing MFBs are susceptible to various attacks and practical defense solutions are missing.

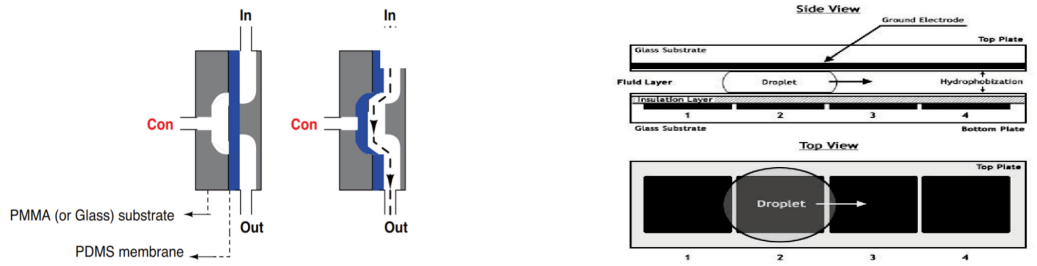
Our objective is to identify the vulnerabilities of two types of MFBs to RE attacks and HT attacks, while we also propose effective countermeasures against the attacks. Previous work on the security of MFBs only focuses on DMFBs. Security evaluation of DMFBs has been studied in References [3, 5]. The susceptibility of the DMFB supply chain is revealed in Reference [2], where proprietary protocol piracy and HT attacks are discussed. Protocol encryption and Physical Unclonable Function (PUF) have been demonstrated on DMFBs for IP protection [4, 16]. None of the previous works has discussed the security concern of FMFBs nor presented a systematic framework for the identified attacks. Securing FMFBs against possible attacks remains an open problem. In contrast to the general security discussion in the previous work, this article presents the first systematic methodology for implementing practical attacks and countermeasures on both DMFBs and FMFBs. We demonstrate two hardware attacks—RE and HT—at multiple levels. The technical contributions of our work can be summarized as follows:

- Presenting the first comprehensive vulnerability study of DMFBs and FMFBs. We identify potential threats in the supply chain of both types of MFBs and demonstrate how to perform stealthy attacks in different phases.
- Proposing systematic attack methodologies for RE and HT on MFBs and discussing the overhead of the attacks. The proposed attacks are evaluated on various benchmarks where experimental results prove the effectiveness of our attacks.
- Presenting countermeasures to protect MFBs against the attacks and assessing the performance of the defense methods on available benchmarks. The complexity and overhead of the protection schemes are analyzed.
- Developing software tools that fully automates the deployment of attacks and countermeasures. We plan to make our devised benchmarks and the software open-source so other researchers interested in the topic can test their methods.

The article is organized as follows: Section 2 introduces the background about MFBs and traditional hardware attacks. Section 3 discusses previous works related to the security and vulnerability of MFBs. Section 4 presents the attack model and a systematic framework of our RE attacks on MFBs. Section 5 describes the methodology for HT attacks on MFBs. Section 6 presents the experimental results of proposed attacks. Section 7 suggests potential countermeasures to protect MFBs against proposed attacks and discusses the overhead. Section 8 summarizes existing attacks and countermeasures for MFBs. Section 9 concludes the article.

2 PRELIMINARIES

In this section, we introduce the mechanisms of MFBs (Section 2.1), the supply chain (Section 2.2), and two hardware attacks: RE and HT (Section 2.3).



(a) Schematic of a normally-closed valve.

(b) Architecture of a typical two-plate DMFB.

Fig. 1. Schematic view of flow-based [33] and digital-based microfluidic biochips [27].

2.1 Principle of Microfluidic Biochips

Existing microfluidic biochips can be divided into two categories based on their working mechanisms. FMFBs are the first generation of MFBs that manipulate continuous flow inside microchannels to conduct experiments. In contrast, droplet-based (or digital-based) microfluidic biochips control the movement of discrete droplets to run the biomedical protocol.

2.1.1 Flow-based Microfluidic Biochip. FMFBs consist of two main components: microvalves and microchannels. The substrate of FMFBs is made of polydimethylsiloxane (PDMS) and flow paths are formed by etching the substrate. Microvalves are the building blocks of FMFBs and control the fluid in flow channels [32]. Valves are located in the intersection of the control layer and the flow layer while microchannels are located in the flow layer. Normally closed valves remain closed and restrict the fluid when no vacuum is applied as shown in Figure 1(a), where the valve is denoted by *Con* and the channel is denoted by *In*, *Out*. Valves are forced to open and allow the flow when the vacuum is applied. Complex components such as mixers, switches, pumps, and incubators can be built from valves [27]. Experiments are performed on these functional flow components.

To remove the dependence on external control, recent work has found out that pneumatic valves can be viewed as NMOS transistors, and the architecture of the FMFB can be modeled as a logic circuit [29]. Reference [32] proposes a new pneumatic valve that functions like a PMOS transistor. Reference [33] advances FMFB design for on-chip control synthesis of mVLSI biochips. We will show that the on-chip control circuitry is vulnerable to attacks in the later sections.

2.1.2 Digital Microfluidic Biochip. DMFB is the second generation of MFBs, which uses Electrowetting-on-Dielectric (EWOD) phenomena to manipulate discrete droplets on a two-dimensional electrode array [25, 39]. The structure of a typical two-plate DMFB is shown on the top part of Figure 1(b), consisting of glass substrates, dielectric layers, hydrophobic layers, and electrodes. EWOD alters the surface tension of the droplet by applying the electric field. The contact angle between the droplet and solid surface decreases when high voltage is applied on the control electrode. The resulting electrostatic force moves droplet towards the actuated electrode. The bottom part in Figure 1(b) visualizes the droplet transportation on a DMFB where the droplet is moving towards the third electrode activated at the time step. Droplets containing samples or reagents can be manipulated to carry out various operations by EWOD-based actuation.

Compared to FMFBs, DMFBs are reconfigurable, since droplets can move across the entire array instead of following the permanently etched channels. The integration of sensors facilitates the emergence of cyberphysical DMFBs that monitor the protocol execution in real-time and send feedback to the control system. Field-Programmable Pin-Constrained (FPPC) DMFB reduces

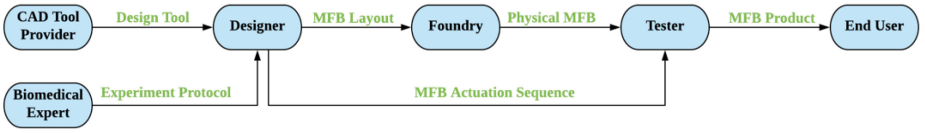


Fig. 2. Supply chain of current MFB products. The design, manufacturing, and testing of commercial MFBs are conducted by different parties.

manufacturing cost by pin-sharing and enables the mapping of various experiments to the same device [12, 14, 40]. The major advantage of FPPC-DMFBs over the early generation of DMFBs is the reconfigurability and a more flexible pin-mapping scheme. Early DMFBs are application-specific and each electrode is controlled by one external pin. On the contrary, FPPC-DMFBs are programmable and allow multiple electrodes to share control pins. Due to the flexibility and lower manufacturing cost, FPPC-DMFBs are becoming dominant in the market.

2.2 Supply Chain of MFBs

Various operations on MFBs are realized by the interaction of multiple disciplines including electricity, optics, and fluidics. The supply chain of commercial MFBs is shown in Figure 2. The multi-disciplinary nature of MFBs determines the fact that their commercialization is dependent on the collaboration of experts across various fields. More specifically, the designer uses the CAD tool from the vendor to design and synthesize the MFB based on the protocol provided by the biomedical expert. The generated MFB layout is delivered to the foundry for fabrication. The tester runs the target protocol on the manufactured MFB before it is distributed to the end customer. The supply chain of DMFBs and several potential attacks are discussed in Reference [2]. As such, one can see that the supply chain of MFBs involves multiple parties and is susceptible to malicious attacks if untrusted parties participate in the process. In this article, we exploit the vulnerabilities of the MFB supply chain and demonstrate two practical attacks: RE and HT.

2.3 Hardware Attacks

DMFBs are susceptible to conventional hardware attacks for digital circuits, and adversaries can launch attacks in different phases of the supply chain [2]. In this article, we expand the security analysis and elaborate how the attacks and defense can be deployed on the interdisciplinary MFBs, focusing on RE (Section 4) and HT (Section 5).

2.3.1 Reverse Engineering Attack. Traditional RE aims to reconstruct the desired abstraction of the circuit and involves the following tasks: identifying the underlying technique, recovering the gate-level netlist, and deducing the functionality of the chip [35, 36]. Depackaging, delayering, and image processing are standard procedures to expose the internal details of the chip and reconstruct the gate-level netlist.

Protection against RE can be achieved by obfuscation [15] and camouflaging [34]. Obfuscation hides the functionality of a design by inserting additional elements, in which case correct inputs to the added elements are required to ensure the desired functionality. The Finite State Machine (FSM) of the design can be obfuscated by inserting extra states or transitions. The state transition is valid only when the correct key is applied, otherwise the chip will be stuck at black hole states. Camouflaging is a layout-level technique that aims to hinder the image-based reconstruction of the netlist and can be carried out by making different functional elements look alike, adding dummy gates or wires, or fill in the unused space. In this article, we show that MFBs are also vulnerable to RE attacks that lead to the recovery of the chip layouts, protocols, or FSMs.

2.3.2 Hardware Trojan Attack. Hardware Trojans are defined as malicious modifications made to the circuit [35]. A Trojan usually consists of three parts: trigger, driver, and storage [43]. The driver is a malicious circuitry that leaks sensitive information or causes malfunction of the infected circuit. HT trigger activates the driver when the trigger condition is satisfied. The hostile actions to be taken are stored in the HT storage. A systematic approach for optimal Trojan creation and placement is presented in Reference [46], which aims to design a stealthy Trojan.

HT detection for conventional ICs is challenging due to multiple reasons. Process variation and measurement noise allow the attacker to hide the effect of inserted Trojans. The opaqueness and small feature size of IC internals hinder physical inspection while destructive RE is costly and inefficient. State-of-the-art HT detection techniques include functional testing, side-channel analysis (SCA), gate-level characterization, and unused circuit identification (UCI) [47].

3 RELATED WORK

Security and testing of MFBs are attracting more attention due to the growing usage of these devices. Vulnerabilities and potential attacks of DMFBs have been identified [2, 5, 9], while the security evaluation of FMFBs is still missing. In this section, we introduce the available testing methods for MFBs and security assessments for DMFBs.

3.1 Testing Methods and Security Evaluation for MFBs

The existence of physical defects has been a concern since the innovation of FMFBs. Experiments running on a defective biochip might lead to wrong outcomes or abnormal termination. Repeating experiments consumes more samples and induces extra cost. To address the problem, Reference [20] proposes to model the flow architecture of MFBs as a logic circuit and deploy standard ATPG tools to generate testing patterns. Results of the ATPG tool are then mapped back to the biochip where the states of primary inputs correspond to the activation or deactivation of pumps, and the responses correspond to the expected feedback from the pressure sensors. Reference [31] identifies the influence of physical defects on MFBs and uses graph-theory for maximal fault coverage via test point insertion.

A comprehensive study of recent advances in DMFB testing techniques is presented in Reference [38]. Existing testing methods such as structural testing, functional testing, Built-in-Self-Test (BIST), Design For Testability (DFT), error recovery, and defect-aware synthesis are revisited.

There is no work discussing the security problems of FMFBs, whereas possible attacks on DMFBs have been identified. Reference [2] summarizes hardware attacks that threaten the security and privacy of DMFBs. Attacks that may happen in the different phases of the supply chain are identified and categorized into three classes: Trojans, IP piracy, and counterfeiting. Potential countermeasures such as watermarking, metering, locking, and obfuscation are suggested. Nevertheless, side-channel attacks and vulnerabilities induced by cyberphysical components are not considered in their work. We show how to exploit the security hole for applying attacks on MFBs and provide corresponding defense against the attacks.

3.2 Security Enhancement of DMFBs

Along with the development of testing technologies, progress has been made on the protection of DMFBs. The authors in Reference [4] present a method to encrypt biomedical protocols by inserting Fluidic Multiplexers (FMUX) into the original sequencing graph. The control inputs to FMUXs serve as the secret keys of the assay encryption. Without correct keys, synthesizing the FMUX-inserted bioassay generates an incorrect sequencing graph that leads to wrong outputs. FMUXs obfuscate the assays in the design phase, preventing protocol piracy and chip overbuilding. The number of inserted FMUXs determines the length of secret keys and, therefore, the overhead

as well as the security level of assay encryption. However, the proposed encryption scheme has the disadvantage that all fabricated DMFBs with the same encrypted sequencing graph share the same secret keys, which impairs the resiliency of DMFBs. Also, the encrypted assay remains vulnerable to protocol piracy attacks launched by manufacturers and authenticated-but-curious end-users.

The intrinsic manufacturing variation of the electrodes on the DMFB can be utilized to construct an on-chip PUF [16] that prevents RE and IP piracy. The absorption induced by electrodes varies uniquely from chip to chip and therefore the volume of resulting droplet after undergoing the same operations can be used as the "fingerprint" of the biochip. The PUF response is generated by the comparison of droplet volumes that are estimated from the images taken by CCD cameras. The DMFB is locked by inserting additional states to the original FSM. Only the authentic device can generate the correct PUF response, which is used to unlock the FSM. The device is required to pass the verification test before the desired protocol is performed.

4 REVERSE ENGINEERING ATTACKS ON BIOCHIPS

Reverse engineering is a common attack on silicon circuits that facilitates IP piracy and Trojan insertion. In this article, we propose the first systematic approach to RE two types of biochips from multiple perspectives. Our RE attack spans layout-level, protocol-level, and FSM-level. The attack model and attack methodology for FMFBs and DMFBs are presented in Section 4.1 and Section 4.2, respectively.

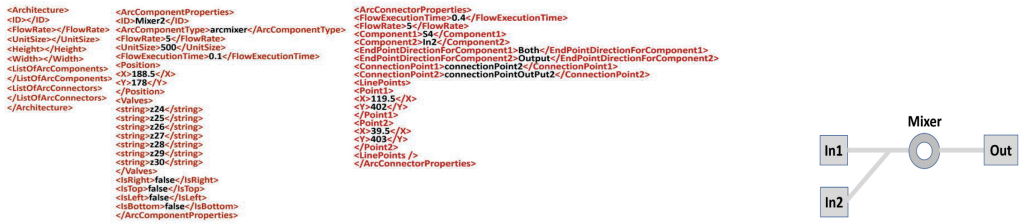
4.1 RE Attacks on FMFBs

Although the threats of physical defects have been identified and addressed, the methodology and prevention of malware attacks on FMFBs have not been well studied. In this section, we propose a systematic RE attack framework on FMFBs at multiple levels and discuss the required overhead.

4.1.1 Attack Model. We use the application-specific FMFB as the attack platform where only one specific protocol can be performed. RE attacks at three abstraction levels are presented. First, we reconstruct the layout of the FMFB assuming the adversary knows the control specification owned by the designer or has the image of the FMFB (Section 4.1.2). Combining the recovered layout with the valves control table generated by control synthesis, the protocol can be further reconstructed (Section 4.1.3). Finally, assuming the attacker has the physical FMFB device, we RE the FSM by signal tracing (Section 4.1.4).

Note that for a reprogrammable FMFB, layout RE attacks do not apply, since the layout of the target FMFB can be changed by synthesis after fabrication [45]. As for reverse engineering the protocol or FSM of a programmable FMFB, the adversary uses the proposed attacks in Section 4.1.3 and 4.1.4 assuming he knows the valves assignment scheme of the target FMFB in addition to the valve control tables. Our RE attack methodology is applicable to programmable FMFBs, since the only difference between RE and application-specific FMFB and a general-purpose FMFB is how to decide the states of valves in each component from the valve control tables, which can be determined using the knowledge of the valve assignments.

4.1.2 Layout Level RE. As explained in Section 2.1, the flow channels of FMFBs are formed by etching the substrate. The fabrication process is deterministic and non-invertible, making the design of FMFB vulnerable to RE. The malicious foundry or the end-user in the supply chain might perform layout RE attacks on the FMFB to recover and sell the design of the FMFB. In this section, we present two methods to reconstruct the layout of the FMFB under different assumptions about the knowledge of the adversary. The complexity and overhead of both methods grow linearly with the number of components in the design.



(a) Hardware description of a sample FMFB architecture file provided by the simulation tool [29].

(b) Simple FMFB layout schematic.

Fig. 3. Example of FMFB layout description and visualization.

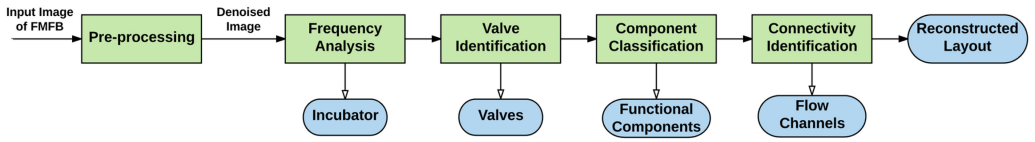


Fig. 4. Workflow of reverse engineering the layout of the FMFB using image analysis. Flow components and their connections are recovered and combined to reconstruct the layout.

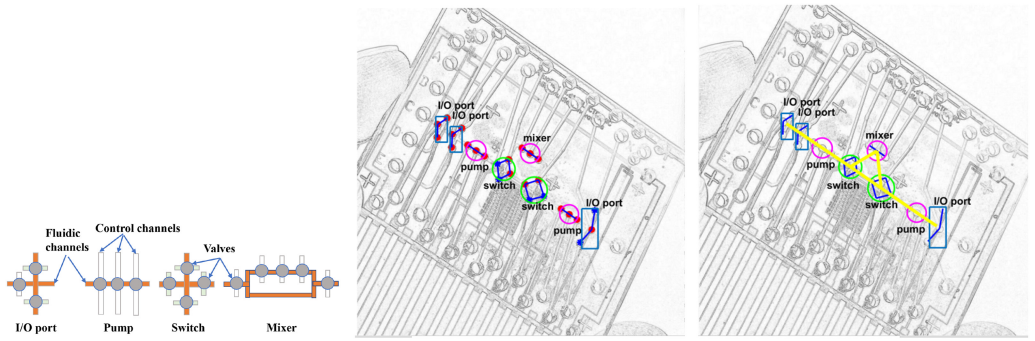
Layout RE based on Hardware Description: Layout RE may occur in the design phase assuming the adversary knows the architecture description of the FMFB. Figure 3(a) shows an example description of the FMFB following the design process in Reference [37]. In this case, the malicious foundry in the supply chain can be the adversary and is able to perform layout RE attacks, since the foundry receives the hardware description from the designer to manufacture the FMFB.

The first layout RE approach assumes that the attacker has the architecture file of the target FMFB. The netlist extraction module in our software takes the architecture description as input and generates a Directed Acyclic Graph (DAG) visualization of the FMFB's layout. The node set and edge set represent the components and the connector, respectively. These two sets are combined to recover the DAG that has the same structure as the layout shown in Figure 3(b).

Layout RE based on Image Analysis: Assuming the adversary has an image of the target FMFB and knows the component library deployed in the FMFB, an alternative approach for layout RE is image analysis. In practice, a malicious end-user in the supply chain with knowledge of components can launch such an attack and reconstruct the component-level architecture of the target FMFB. Image analysis is an essential procedure in the conventional RE of digital circuits. In contrast to the IP piracy procedures described in Reference [2], we prove that depackaging and delayering is unnecessary for FMFBs. Image processing is sufficient to recover the layout of FMFBs by leveraging the transparency property of the substrate materials used in manufacturing.

Our image-based RE attack is general and applicable to various FMFBs where different component libraries might be used. A specific component may have different numbers of microvalves, microchannel connectivity, and layouts in different designs. However, the template for any component on the target FMFB is deterministic and can be identified from the corresponding component library. Therefore, the template-matching method used in our image-based RE attack can handle such structural variability across designs assuming the attacker knows the component library.

The workflow of image-based layout RE is shown in Figure 4. The attack involves five stages: image pre-processing, frequency analysis, valve identification, component classification, and component connectivity identification. We explain the details of each step as follows:



(a) (b) (c)

Fig. 5. Example of component library (a), component classification (b), and connectivity identification (c).

(1) Image Pre-processing: We assume that the attacker has a regular image of the FMFB. The image can be taken using cellphones or digital cameras instead of expensive microscopes that are usually required by silicon RE. Denoising and non-uniform illumination correction algorithms are applied to the input image to facilitate subsequent image processing.

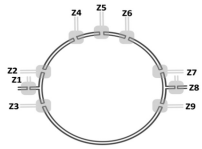
(2) Frequency Analysis: For bioassays that require high temperature in specific processes, heating modules such as incubators are integrated on the biochip. These heating components typically take the shape of periodic, densely distributed line segments, thus contributing to the high-frequency part in the Fourier domain. Our layout RE tool takes advantage of this observation and uses Discrete Cosine Transformation (DCT) to locate the incubator. The image of the target FMFB is first transformed into the frequency domain and thresholded by a predefined value. Only the high-frequency component is kept and transformed back to the spatial domain, indicating the area spanned by the incubator.

(3) Valve Identification: Valves are the building blocks of components and can be identified using template matching. Given the template of a valve and the image of the FMFB, a typical template-matching algorithm is deployed to identify the center positions of all valves.

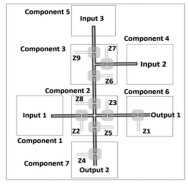
(4) Component Classification: The identified valves in Step 1 are clustered and labeled using the pertinent component library of the FMFB. Figure 5(a) shows an example of the component library assumed to be known to the attacker. Structural characteristic is exploited to classify and label each functional component. To identify components on the FMFB, the pairwise distance between all valves is computed and compared with the proper threshold to determine whether two valves belong to the same component. The functionality of each cluster is automatically annotated by matching the pattern of the cluster to the ones in the library. An example of component classification is shown in Figure 5(b), where valves are grouped in clusters with their functions annotated.

(5) Component Connectivity Identification: After valves and components are identified, the last step is to reconstruct the connectivity between components. Our attack framework explores the continuous property of fluids and finds neighbors of each component. The pairwise distance between components is computed and compared with the threshold to determine the connectivity. Figure 5(c) shows the intermediate output of connectivity reconstruction. The yellow lines denote the connection between components, satisfying the continuity constraint we explore.

4.1.3 Protocol Level RE. The protocol mapped to the FMFB can be represented by an application graph. Our protocol RE attack aims to reconstruct the scheduled operations in the graph by



State	Input			Output						
	Q_3	Q_2	Q_1	Z2	Z3	Z4	Z5	Z6	Z7	Z9
Loading 1	0	0	0	1	0	1	1	1	1	0
Loading 2	0	0	1	0	1	X	X	X	0	1
Mixing 1	0	1	0	1	1	1	0	0	1	1
Mixing 2	0	1	1	1	1	0	1	0	1	1
Mixing 3	1	0	0	1	1	0	0	1	1	1
Unloading 1	1	0	1	1	0	1	1	1	1	0
Unloading 2	1	1	0	0	1	X	X	X	0	1
Unused	1	1	1	X	X	X	X	X	X	X



State	Input			Output								
	Q_3	Q_2	Q_1	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9
Closed	0	0	0	0	X	X	0	X	X	X	X	X
C1 → C6	0	0	1	1	1	1	0	0	X	X	0	X
C1 → C7	0	1	0	0	1	0	1	1	X	X	0	X
C4 → C6	0	1	1	1	0	1	0	0	1	1	1	0
C4 → C7	1	0	0	0	0	0	1	1	1	1	1	0
C5 → C6	1	0	1	1	0	1	0	0	1	0	1	1
C5 → C7	1	1	0	0	0	0	1	1	1	0	1	1
Unused	1	1	1	X	X	X	X	X	X	X	X	X

Fig. 6. Schematics and the corresponding valve control table for a mixer and a simple flow architecture.

analyzing the valve control table. As discussed in Section 2.1.1, valves are the building blocks of FMFBs and are responsible for controlling the flow inside the components or channels. Valve assignments specify which valves belong to which components or flow paths and thus are essential for the attacker to infer the fluid activity from valve states. The attacker who aims to perform protocol RE attacks on FMFBs can be the malicious foundry or user in the supply chain. The FMFB layout, valve assignments, and the valve control tables are assumed to be known to the Attacker.

To facilitate operation identification, we divide operations into two categories: functional operations that are running on the flow components, and transportation operations that move the flow between different components. The top and bottom images in Figure 6 show the schematic design and valve control table for a mixer (functional) and a flow architecture (transportation), respectively. The "Output" columns specify the valves used in the component where the elements can take three values: 0, 1, and X, corresponding to "closed," "open," and "don't care," respectively. The values of valves are encoded by the signals in "Input" columns.

The workflow of our protocol RE attack on FMFB is outlined in Algorithm 1. The attacker obtains sub-tables from the original valve control table by selecting the columns that are used by the component or the flow path. The sub-tables determine the activities of each component and flow path and are used to infer the scheduled functional operations and flow transportation.

ALGORITHM 1: Framework of Reverse Engineering Protocols Mapped to the FMFB.

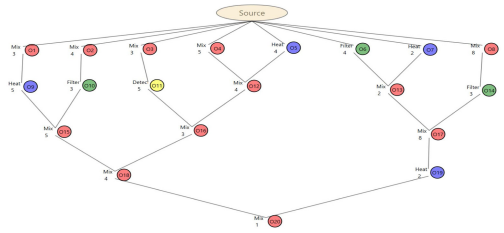
Input: FMFB layout L, valve assignment f_V, valves control table $\mathbf{T}$ for the protocol.
Output: DAG representation of the protocol.
Extract component set $\mathbf{C} = \{C_1, ..., C_M\}$ and flow path set $P = \{P_1, ..., P_N\}$ from the layout.
Each element C_i and P_i is a set of valves $\{z_1, ...z_j\}$ obtained from the valve assignment f_V;
Operation counter $cnt \leftarrow 0$;
for $1 \le i \le M$ **do**
  Determine the operation type of the component: $type \leftarrow L(C_i)$;
  Select columns from $\mathbf{T}$ and construct the valves control sub-table for the component C_i:
  $\mathbf{T_i} \leftarrow \mathbf{T}(f_V(C_i))$;
  Identify the scheduled time intervals for the operation:
  $\mathbf{O_{time}} \leftarrow IdentifyOperationTime(\mathbf{T_i})$;
  Recover the functional operations and add to the node set $\mathbf{N} \leftarrow \{type, \mathbf{O_{time}}\}$;
for $1 \le i \le N$ **do**
  Select columns from $\mathbf{T}$ and construct valves control sub-table for the flow path P_i:
  $T_i \leftarrow \mathbf{T}(f_V(P_i))$;
  Identify the time intervals when the flow path P_i allows transportation:
  $\mathbf{P_{time}} \leftarrow IdentifyPathTime(\mathbf{T_i})$;
  Add to the edge set $\mathbf{E} \leftarrow \{P_i, \mathbf{P_{time}}\}$;
DAG representation of the protocol: $DAG \leftarrow \{\mathbf{N}, \mathbf{E}\}$

Valves Control Table

Information: 1 : Closed X : Free 0 : Open

Time	z0	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12	z13	z14	z15
00:00:00:000	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
00:00:00:100	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
00:00:00:200	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
00:00:00:300	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
00:00:00:400	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1
00:00:00:500	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1
00:00:00:600	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1
00:00:00:700	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1
00:00:00:800	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	0
00:00:00:900	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	0
00:00:01:000	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	0
00:00:01:100	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	0

(a) Valves control table for the bioassay in (b)



(b) Application graph of a test protocol.

Fig. 7. Valve control table and DAG representation of a test protocol from the FMFB simulator [23].

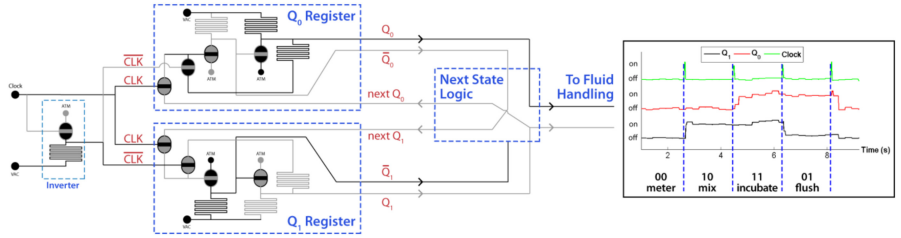
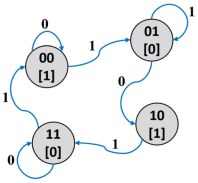


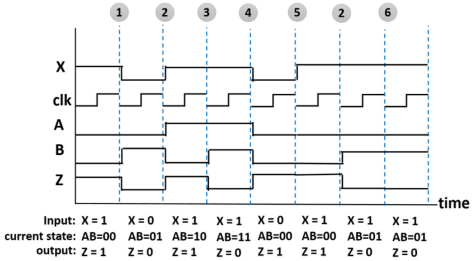
Fig. 8. Schematic diagram of a two-bit FSM and its corresponding timing waveform [28]. The two-bit states Q_0, Q_1 from the FSM are sent to the fluid handling system.



(a) Example of a two-bit FSM.

A	B	X	A*	B*	Z	
0	0	0	0	0	1	5
0	0	1	0	1	0	1
0	1	0	1	0	1	2
0	1	1	0	1	0	6
1	0	0	0	0	1	
1	0	1	1	1	0	3
1	1	0	1	1	0	
1	1	1	0	0	1	4

(b) Truth table of state transitions and output of the example FSM.



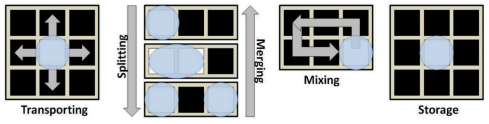
(c) Timing waveform of the under-test FSM.

Fig. 9. Demonstration of FSM RE using signal tracing method.

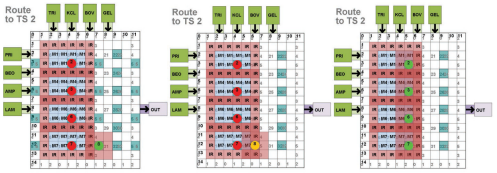
The application graph is essentially a DAG where the nodes and edges represent the functional operations and transportation operations, respectively. Figure 7 shows an example of the valve control table and the DAG of a test protocol mapped to the FMFB. The states of valves in each time step are specified in the control table. The overhead of protocol RE increases linearly with the number of valves in the valve control table and the execution cycles of the bioassay.

4.1.4 FSM Level RE Based on Exhaustive Signal Tracing. The FSM controller and asynchronous counter design for FMFBs is presented in Reference [28], which aims to achieve autonomous fluid handling. Their work takes advantage of pneumatic valves to construct a digital circuit controlling the fluid handling components. Figure 8 shows a two-bit FSM that provides four individual operations.

We present the first FSM RE attack on FMFBs based on an exhaustive signal tracing method. Signal tracing applies a set of comprehensive input sequences to stimulate all transitions present in the target FSM. Figure 9 demonstrates how the signal tracing method is used to RE FSM. The



(a) Typical operations that can be executed on the DMFB. Transporting, splitting, merging, mixing and storage are shown here [16].



(b) Visualization of three continuous frames of a PCR experiment. The bioassay is mapped to a field programmable pin-constrained DMFB.

Fig. 10. Demonstration of typical DMFB operations and simulation of a bioassay.

example FSM has four states encoded by two bits A, B, the next state is denoted by A^*, B^*, and the state transition is controlled by the input X. The output of the FSM Z is optional and dependent on the application. Without loss of generality, we assume that the FSM is reset to state $AB = 00$. An input sequence $\{1, 0, 1, 1, 0, 1, 1\}$ is applied on the unknown FSM and stimulates the state transitions. From the timing waveform, it can be seen that only six transitions marked by the rectangular boxes in Figure 9(b) are discovered by the input sequence and two other transitions are missing. To recover the full truth table, the attacker can apply an additional input sequence $\{1, 0, 1, 0\}$, tracing from state 00 to 01 to 10 to 11, for capturing the remaining transitions.

The attacker starts by applying the comprehensive collection of input sequences to the FMFB control circuit, observing the operations, and measuring the feedback from the sensor. The set of executing operations represents the next state of the FSM and the sensor feedback corresponds to the output under current state and input stimulus. The timing waveform (Figure 9(c)) is reconstructed from the continuous monitor of the challenge-response pairs ($currentState$, $input$; $nextState$, $output$). Then the truth table of the next state and output (Figure 9(b)) is obtained from the timing waveform. Finally, the state transition diagram of the FSM (Figure 9(a)) describing the control mechanism of the protocol is recovered from the truth table.

4.2 RE Attacks on DMFBs

In this section, we focus on protocol level RE of DMFBs and propose a systematic attack methodology leveraging the actuation sequence or the video frames for the target experiment.

4.2.1 Attack Model. The proposed RE attack aims to reconstruct the protocols mapped to a general-purpose DMFB (can be either pin-constrained or direct-addressed). The proprietary bioassay is represented by a DAG, which is the desired output of our protocol RE attack. The layout RE attack on DMFBs is not considered here, since general-purpose DMFBs are reprogrammable, which means their layouts can be modified by synthesis. To launch protocol RE attacks, we demonstrate two approaches leveraging either the video frames or the actuation sequences for the target protocol. The video-based protocol RE attack assumes the attacker has access to the videos of the bioassay taken by the camera in a cyberphysical DMFB (Section 4.2.2), while the actuation sequence-based method assumes the adversary can eavesdrop on the communication channel between the control signal generation module and the external pins of the DMFB (Section 4.2.2).

4.2.2 Protocol Level RE on DMFBs. The proprietary protocol constitutes the IP of the DMFB and can be characterized by the scheduled operations combined with pertinent biomedical libraries [2]. The protocol can be visualized as a sequencing graph $G = (N, E)$, where N is the node set denoting the scheduled operations and E is the edge set denoting the dependencies between operations [3]. Typical droplet operations are shown in Figure 10(a), where different operations can be distinguished by the movement patterns of droplets. Our attack framework provides two alternatives

to reverse engineer the protocol, assuming the availability of actuation sequence or video frames, respectively. The possible adversary might be the malicious tester or the end-user in the supply chain of the DMFB. Droplet coordinates are extracted and the protocol is reconstructed by analyzing the change of droplet locations in continuous cycles. The workflow of two attack methods is discussed below.

ALGORITHM 2: Framework of Actuation Sequence-based Protocol Reverse Engineering.

Input: I/O port position **In**, **Out**; execution cycles T, actuation matrix **S**; pin-mapping function f_m; mix duration threshold t.
Output: Protocol set $\mathbf{P} = \{\mathbf{O^{(1)}}, ..., \mathbf{O^{(T)}}\}$.

```
for 1 ≤ i ≤ T do
    I^(i) ← f_m(s^(i));   N^(i) ← #rows(I^(i));
for 1 ≤ i < T do
    InputID ← HasInput(I^(i), In);
    if InputID ≠ ∅ then
        add InputID to ID^(i);   add ('Input', InputID, i) to O^(i);
    OutputID ← HasOutput(I^(i), Out);
    if OutputID ≠ ∅ then
        deletes OutputID from ID^(i);   add ('Output', OutID, i) to O^(i);
    Ds ← pdist2(I^(i), I^(i));   Dx ← pdist2(I^(i), I^(i+1));
    if find(Dx == 0) ≠ ∅ then
        id ← find(Dx == 0);   add ('Store', id, i) to O^(i);
    if find(Dx == 1) ≠ ∅ then
        id ← find(Dx == 1);   add ('Mov', id, i) to O^(i);
    if find(Ds == 1) ≠ ∅&N^(i) > N^(i+1) then
        add ('Merge', id, i) to O^(i);
    if find(Ds == 1) ≠ ∅&N^(i) < N^(i+1) then
        add ('Split', id, i) to O^(i);
    for 1 ≤ j ≤ N^(i) do
        id ← ID^(i)(j); route ← I^(i:i+1)(j, :);
        if ContinuousMove(route) == 'True' then
            add ('Mix', dropletID, i) to O^(i);
    I^(i+1) ← UpdateIDList(I^(i));
```

Actuation Sequence–based Protocol RE: Bioassay execution is determined by the actuation sequence sent from the controller to the external pins of DMFBs. The actuation sequence is a binary string where bits "1" mean the connected pins (and all electrodes connected to those pins) are activated, while bits "0" mean connected pins are deactivated. The actuation sequence-based protocol RE involves three steps: **(1)** deducing the positions of droplets in each time step; **(2)** identifying droplet operations from continuous actuation sequences; and **(3)** recovering the protocol (DAG) from all operations identified in Step2. The workflow of the actuation sequence-based attack is outlined in Algorithm 2.

The proposed protocol RE attack is applicable to both direct-addressed and pin-constrained DMFBs. Each pin is only connected to one electrode in the direct-addressing scheme, while multiple electrodes may share one external pin in pin-constrained scheme. Note that the pin-mapping scheme of an FPPC-DMFB is fixed after manufacturing. Using the continuity constraint of droplet transportation (a droplet can only move to its adjacent electrode), the attacker can deduce the position of the droplets on a direct-addressed DMFB or FPPC-DMFB from the states of pins assuming he knows the pin-mapping scheme and has access to the external pins.

The movement pattern of each droplet is identified based on its positions in successive cycles. Furthermore, the protocol is reconstructed by combining the activities of all droplets during the execution. The total number of execution cycles is T, the actuation vector in ith cycle is $\mathbf{s^{(i)}}$ and the

actuation matrix is $\mathbf{S} = [\mathbf{s}^{(1)}; \mathbf{s}^{(2)}; \ldots; \mathbf{s}^{(\mathbf{T})}]$. The number of present droplets in ith cycle is $N^{(i)}$ and coordinates of droplets in ith cycle are denoted by $\mathbf{I}^{(\mathbf{i})} = (\mathbf{x}^{(\mathbf{i})}, \mathbf{y}^{(\mathbf{i})})$. $\mathbf{I}^{(\mathbf{i})}$ is an $N^{(i)}$-by-2 matrix where each row of it corresponds to the coordinate of a droplet. The positions of peripheral input/output ports **In**/**Out** are assumed to be known by the attacker. The target protocol **P** is represented by a set of chronological operations $\mathbf{P} = \{\mathbf{O}^{(\mathbf{i})}, i = 1, \ldots, T\}$, where $\mathbf{O}^{(\mathbf{i})}$ is the collection of operations that happen in ith cycle. When new droplets enter the DMFB, they are labelled with unique identifiers (id) and the droplet ID list $ID^{(i)} = \{id_1, \ldots, id_{N^{(i)}}\}$ is updated. Parent droplets in ith cycle may move one grid or keep static during one cycle, producing child droplets in $(i + 1)th$ cycle.

Using the operation types defined in the simulation tool [13], our software classifies operations into seven categories: "Input," "Output," "Merge," "Mix," "Split," "Move," and "Store." The output of our protocol RE attack is the protocol description set **P**. For each specific operation, the data structure **O** consists of three parts: the classification label, participant droplet identifier (id), and the execution clock cycle (i). The principle of operation classification is intuitive, since the problem of operations classification is equivalent to activity recognition, a popular branch in computer vision. To evaluate the performance of the proposed attack, we compare the recovered operations in the set **P** with the groundtruth bioassay description from the simulation tool. The accuracy of protocol RE is defined as the percentage of correctly identified operations.

Video Analysis–based Protocol RE: Prior works have developed design and synthesis frameworks that incorporate cyberphysical components (such as CCD cameras and optical sensors) into DMFBs for error recovery, fault detection, and real-time quantitative analysis [11, 18, 21, 22]. However, none of these papers consider the vulnerability and the enlarged attack surface induced by the cyberphysical components. Existing cyberphysical DMFBs are not protected against malicious attacks, and the adversary may obtain illegal access to the integrated CCD cameras. In the following, we demonstrate how video analysis can be deployed to recover the protocol mapped to a cyberphysical DMFB (with/without pin-constrained schemes).

For a cyberphysical DMFB equipped with a CCD camera, the execution of the bioassay is monitored in real time and used as the feedback to the control system. Information leaked though CCD cameras can be misused by the attacker to pirate the IP. As an alternative approach to recover the protocol, video-based RE first identifies the positions of existing droplets by applying template matching on each frame. Subsequent procedures to reconstruct the protocol are the same as the steps described in Algorithm 2. Since droplets are located by template matching, the pin-mapping scheme used in the synthesis phase does not affect the result of matching. Due to the irrelevance of pin-mapping, video-based method removes the requirement of prior knowledge about the pin-mapping function f_m required by the sequence-based approach. As such, the proposed video-based protocol RE attack is applicable to both FPPC-DMFBs and direct-addressed DMFBs.

The execution of a PCR bioassay on an FPPC-DMFB is visualized in Figure 10(b). Connected electrodes share the control pins and are indicated with the same number. Our developed tool proves that pin-count optimized DMFBs are also vulnerable to the protocol RE attack if the video record of the target assay is available to the malicious adversary. Re-synthesizing the bioassay or re-configuring the FPPC-DMFB cannot mitigate the concern of protocol piracy in this case.

5 HARDWARE TROJAN ATTACKS ON BIOCHIPS

In this section, we present HT attacks on MFBs that may result in the leakage of sensitive information or malfunction of the biochips. In Section 5.2, we focus on the Trojan attack on the control circuitry of FMFBs, while plausible Trojan insertions in other locations are also discussed. In Section 5.3, we demonstrate how to insert Trojans into the integrated logic circuit of DMFBs and reveal the vulnerabilities of DMFBs to Trojan attacks in FSM level.

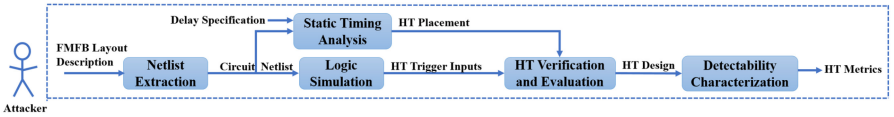


Fig. 11. Global flow of HT creation, verification, and characterization on the control circuitry of FMFBs.

5.1 Motivation and Challenges

In this section, we discuss the difference of HT attacks and detection techniques between digital circuits and MFBs. There are two main differences between conventional circuits and MFBs: (i) silicon-based digital circuits are fabricated using nontransparent materials while MFBs are fabricated with transparent ones; (ii) digital circuits have longer life cycles compared to MFBs, since typical biochips are consumable. On the one hand, it is not practical to extract internal details of the MFB by querying it with different inputs to guide HT insertion as the attacks on digital circuits. On the other hand, the transparent nature of MFBs materials makes RE attacks on biochips easier and feasible compared to the ones on digital circuits, thus facilitating HT insertion in MFBs.

As for HT detection, traditional techniques for digital circuits such as logic testing and side channel analysis [7] cannot be directly applied to MFBs. Logic testing for MFBs [17] can be bypassed by a stealthy Trojan that remains dormant during the testing phase. HT detection techniques including logic testing and side-channel analysis require executing protocols on the target biochip with specific inputs to collect output/side-channel information, which shortens the lifetime of the queried FMFB. To tackle the above challenges, we present a systematic methodology to perform HT attacks on MFBs that is effective and stealthy. Moreover, we propose an HT detection scheme that does not impair the lifetime/quality of the examined MFB and yields no false negatives (Section 7.2) as a practical countermeasure.

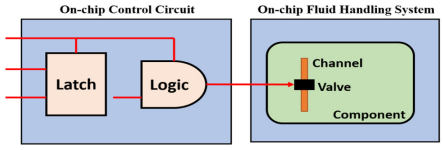
5.2 HT Attacks on FMFB

The security concerns of FMFBs have been ignored since the innovation of the devices, although several techniques have been proposed to enhance the reliability of the platforms [20]. Here, we present a systematic framework to embed Trojans into FMFBs in various locations and analyze the overhead of the proposed attacks.

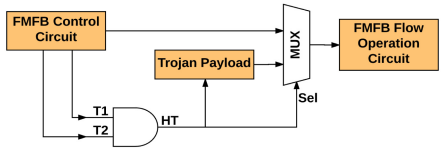
5.2.1 Attack Model. Our attack model assumes the FMFB is application-specific and equipped with on-chip control circuitry that can be implemented using the control synthesis method proposed in Reference [33]. We assume the attacker has knowledge of the architecture specification as well as the delay specification of FMFBs for attack on the control circuitry. The biomedical protocol and the CAD tool are assumed to be available when the attacker targets at the application graph or the valve control table, respectively. Note that our control circuit HT attack also applies to programmable FMFBs, since the reconfigurability does not affect the auxiliary control circuit.

5.2.2 HT Insertion into On-chip Control Circuitry. In this section, we present a systematic framework for HT design and insertion into the control circuitry of the FMFB. The objective of the adversary is to insert a stealthy HT that can evade functional testing with high probability while inducing negligible effects on timing and power side channels. The global flow and the procedures of the proposed attack are explained below.

Global Flow: The global flow of our proposed Trojan attack on the control circuit of the FMFB is shown in Figure 11. With the known routed layout of the FMFB, the attacker first reverse engineers the on-chip control circuitry and extracts the netlist. Logic simulation (LS) is performed to find rarely activated signals in the netlist. Delay specification is used in static timing analysis (STA)



(a) Schematic of on-chip microfluidic control and fluid handling scheme.



(b) Schematic diagram of the proposed Hardware Trojan attack on the control circuitry.

Fig. 12. Demonstration of FMFB on-chip control scheme and proposed Trojan insertion on the control circuit.

to locate timing-critical paths and critical nodes, which facilitates the placement of the Trojan. The feasibility of the trigger is verified using a SAT solver. The resulting Trojan is evaluated in simulation and characterized by detectability metrics.

Figure 12 shows the on-chip control scheme of FMFBs and the schematic of the proposed malware attack. Trigger inputs $T1$ and $T2$ can be generated from the control circuit. The trigger signal HT actuates the Trojan payload circuitry, which diverts the output sequence sent to the fluid handling system running biomedical protocols. The main steps for HT FMFBs are outlined in Algorithm 3 and discussed below.

ALGORITHM 3: Framework of Hardware Trojan Attack on the FMFB's Control Circuitry.

Input: Layout and delay specification of the FMFB, predefined probability threshold ϵ.
Output: Extracted netlist, inputs to a 2-AND HT gate.
Netlist Extraction: Recover the netlist from the routed layout description, denoted as set **Net** = $\{O_1, ..., O_T\}$;
Logic Simulation: generate random test vectors and apply them on the netlist **Net**;
Initialization: $\mathbf{O} = \emptyset, \mathbf{O}' = \emptyset, k = 0, Pair = \emptyset$;
for $1 \le i \le T$ **do**
  $P_1(O_i) = \frac{\#(O_i=1)}{\#TestVectors}$;
  if $P_1(O_i) \le \epsilon$ **then**
    add node O_i to new set **O**;
Perform STA: identify critical nodes and critical paths
for $1 \le i \le length(\mathbf{O})$ **do**
  if slack($\mathbf{O_i}$) ≥ 0 **then**
    add node O_i to set $\mathbf{O}'$;
for $1 \le i \le length(\mathbf{O}') - 1$ **do**
  for $i + 1 \le j \le length(\mathbf{O}')$ **do**
    if $SATsolver(O_i', O_j')$ == *'Satisfiable'* **then**
      $Pair(++k) = (O_i', O_j')$;
      $P_{Trigger}(++k) = Prob(O_i' = 1, O_j' = 1)$;
if $Pair == \emptyset$ **then**
  Increase ϵ and restart from Line 3;
else
  $k^* = argmin P_{Trigger}(k)$;
  $TriggerInput \leftarrow Pair(k^*)$;
  Place the 2-AND HT gate on a non-critical timing path;
  Compute the detectability of the designed HT;

Netlist Extraction: The first phase of the HT attack is netlist extraction, achieved by reverse engineering the hardware description of the FMFB layout. We use the architecture specification from Reference [29] to demonstrate the attack. The architecture file specifies the placement of four types of units: flow components, flow tubes, control components, and control tubes. Flow components such as mixers and incubators are actual functional units that perform the protocol, while control components are logic gates made of valves. Flow tubes and control tubes connect flow components and control components, respectively. An example of control layout specification and control component library is shown in Figure 13. The attacker is only interested in recovering the control circuit composed of control components and control tubes. Combining the description

```
<ControlComponent >
<Id >5 adb1c0d -f05e -4757 -8acb -72 d903dd4334C3 </Id >
<Type >2-And </ Type >
<Orientation >450 </ Orientation >
<Inverted >True </ Inverted >
<Position >
<X >58 </X>
<Y >39 </Y>
</ Position >
</ ControlComponent >

<ConnectedPorts>
    <Port ComponentId="16" ConnectionIdentifier="VNCONTROL" />
    <Port ComponentId="6f8fbcea-818f-4f63-a949-5cccebe3aabd" ConnectionIdentifier="OUT" />
    <Port ComponentId="f0695f6d-8ac4-43ca-a7a9-d53aa609b577" ConnectionIdentifier="B" />
    <Port ComponentId="52df627a-7b4d-4744-9c90-c9ac213b6afc" ConnectionIdentifier="A" />
</ConnectedPorts>
```

(a) Example of a control architecture description contained in the FMFB layout file.

Name	Implementation	Structural Complexity
NOT		4
AND		5
OR		6
NAND		6
NOR		7

(b) Logic gate library used in the control circuit.

Fig. 13. Demonstration of layout-level RE of FMFBs.

of Boolean gates and wires, the logic circuit can be extracted as a counterpart to the on-chip control circuit. The nodes present in the reconstructed netlist are denoted by a set $\mathbf{Net} = \{O_1, \ldots, O_T\}$, where O_i is the signal and T is the total number of nodes in the netlist.

Logic Simulation: After recovering the netlist from the layout description, the netlist is fed to the logic simulation module that computes states of all signals when the input vector is given. To analyze the activity of each signal, enormous random test vectors are generated and applied on the netlist. The activation (state = 1) and transition (state changes from 1 to 0 or from 0 to 1) frequency is calculated from the statistical test. The signal with small activation probability is a good candidate to trigger the Trojan, since we are using a 2-AND gate as the HT trigger. We denote the set of rarely activated signals as $\mathbf{O} = \{O_1, \ldots, O_N\}, N \leq T$. Switching activity can be used to evaluate the effects of Trojans on the power side channel.

Static Timing Analysis: To complicate side-channel-based Trojan detection, STA is performed to compute early slacks and late slacks of the control circuitry. Critical nodes are defined as nodes with negative slacks, which means these nodes violate timing constraints and thus are undesirable as trigger inputs. Timing-critical paths with maximum path delay are identified and avoided during Trojan placement. The set of suitable trigger inputs is updated as $\mathbf{O}' = \{O'_1, \ldots, O'_M\}, M \leq N$, which is obtained by deleting critical nodes from the previous set $\mathbf{O}$.

HT Verification: The SAT solver is deployed to check if a pair of two nodes (O'_i, O'_j) selected from $\mathbf{O}'$ can be simultaneously actuated to 1. If two nodes are verified to be compatible, then the trigger probability $P_{Trigger} = P_1(O'_i \cdot O'_j)$ will be computed. Since the correlation between two signals cannot be neglected, $P_{Trigger}$ should not be estimated as the product of $P_1(O'_i)$ and $P_1(O'_j)$. Our software includes a module that derives the Boolean expression of any primary output or intermediate output in terms of primary inputs (PIs), which helps to simplify the product $O'_i \cdot O'_j$ and compute $P_{Trigger}$. The verification and computation procedures are repeated until all compatible pairs are identified and the corresponding trigger probabilities are calculated. The node pair with minimal $P_{Trigger}$ is selected as inputs to the HT trigger. The trigger gate is placed on a non-critical path found by STA to complete the Trojan design.

Detectability Characterization: Security metrics are defined to evaluate the difficulty of HT detection. We profile the HT from three aspects: activation probability $P_{Trigger}$, transition ratio R_t, and structural observability R_c. The detectability is defined as the modulus of the triplet $d = \{P_{Trigger}, R_t, R_c\}$. The activation probability $P_{Trigger}$ can be approximated by random simulation:

$$P_{Trigger} = \#(HT = 1)/\#RandomTests. \tag{1}$$

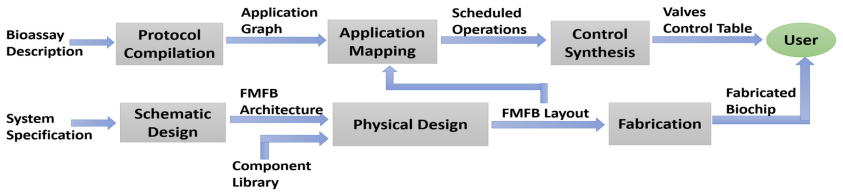


Fig. 14. Design flow of an FMFB. HT attacks may occur at various phases of the design flow.

The switching activity of the netlist is quantified by the transition ratio defined as below:

$$R_t = \frac{\#Transitions_{HT}}{\#Valves_{HT}} / \frac{\#Transitions_{original}}{\#Valves_{original}}. \tag{2}$$

The total number of valves and transitions in the original netlist and the HT-inserted netlist are considered by R_t. Structural observability is determined by the average components complexity $\overline{S}$ and the average placement density $\overline{C}$. The complexity value of each control component S_i is shown in the last column of Figure 13(b). The pairwise distance between two nodes is computed using the positions specified in the layout description. A predefined distance threshold is used to find the number of neighboring nodes C_i for each node. The structural observability is defined as:

$$R_c = (\overline{S}_{original} \times \overline{C}_{original})/(\overline{S}_{HT} \times \overline{C}_{HT}), \tag{3}$$

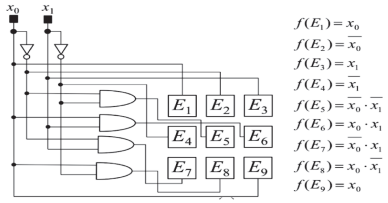
where $\overline{S}_{original} = \sum_{i=1}^{T} S_i/T$, $\overline{C}_{original} = \sum_{i=1}^{T} C_i/T$. For a K-input HT gate, the average value is computed for the single gate: $\overline{S}_{HT} = \sum_{i=1}^{K} S_i/K$, $\overline{C}_{HT} = \sum_{i=1}^{K} C_i/K$.

As can be seen from the definition of the metrics, smaller detectability indicates that it is more difficult to detect the inserted Trojan. More specifically, low activation probability ensures that the HT is rarely triggered; low transition ratio suggests that the malicious HT has negligible effect on the power side channel; and low structural observability means the inserted HT is hard to identify using visual inspection and structural analysis.

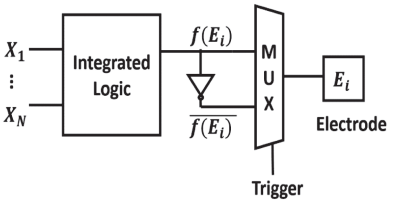
5.2.3 HT Attacks on FMFBs in Other Locations. Apart from the vulnerability of the integrated control circuitry, Trojans might also be inserted in other locations of the FMFB, including but not limited to sensors, biomedical libraries, mechanical valves, application graphs, and valve control tables. Figure 14 shows the design flow of FMFBs, which mainly consists of schematic design, physical design, application mapping, and control synthesis.

Trojan Insertion in Application Graph: Application graph is the visualization of bioassay and is vulnerable to attacks. The protocol compilation that converts the bioassay description to an application graph can be affected by Trojans. For example, the attacker can add undesired mixing operations to contaminate the reagent or add extra heating operations to invalidate the sample, resulting in the incorrect experimental outcome. Original operations might also be deleted, which makes the protocol incomplete and produces incorrect outcomes.

Trojan Insertion in Valve Control Table: The valve control table is generated by control synthesis and determines the states of valves in each time step. The attacker can insert Trojans into the CAD tool and corrupt control synthesis, resulting in the wrong valve control table. Specific operations can be disturbed if the adversary reverse engineers the control table to scheduled operations using the method described in Section 4.1.3. In this way, the rows in the valve control table that correspond to the target operation are identified and maliciously manipulated. An alternative way to apply the Trojan attack is to replace the genuine control sequence in the field.



(a) An example of integrating a logic circuit to control the electrodes on the DMFB.



(b) Schematic of the proposed Trojan insertion in the on-chip logic circuit.

Fig. 15. Example of integrated logic circuit for pin-count reduction and the proposed Trojan insertion.

5.3 HT Attacks on DMFB

The research of HT attacks on DMFBs is in its infancy, although HT attacks and the countermeasures have been well explored on traditional ICs [47]. The vulnerabilities of DMFBs to Trojans are discussed in prior works [2, 3, 5] while no practical attacks are demonstrated. Performing the HT attack on a DMFB is different from attacking a conventional digital ICs in terms of the procedures and the complexity. To design a rarely activated trigger for the stealthy HT, the adversary needs to obtain certain knowledge of the target chip, which holds for both traditional ICs and DMFBs. Current DMFBs typically consist of two transparent Polydimethylsiloxane (PDMS) layers, whereas traditional ICs are composed of more opaque layers that are mainly made of semiconductor. The structural and material properties of DMFBs reduce the complexity and the cost of acquiring information about the internal details of the target DMFB compared to conventional digital ICs.

As for the impact on the fluidic system, HT can be inserted on DMFBs to disturb the actuation sequence, resulting in undesired sample contamination and incorrect droplet transportation. This might further lead to the increasing consumption of samples or even wrong diagnosis results from the bioassay [3, 5]. In the following, we present an innovative HT attack that exploits the vulnerabilities of the integrated logic circuit and the FSM of DMFBs.

5.3.1 Attack Model. We assume that the design description of the original DMFB is known to the attacker for Trojan attacks on the logic circuit and the FSM. The adversary can be a malicious designer in the supply chain who analyzes and tampers with the design of the DMFB before fabrication or synthesis stage.

5.3.2 HT Attack on the Integrated Logic Circuit. Figure 15 shows the principle of logic circuitry integration and the proposed Trojan insertion. Integrating logic circuit with the DMFB has been presented in Reference [10] to reduce the pin counts of DMFBs and therefore cut down the fabrication cost. Signals from the control pins (x_0, x_1) are sent to the logic circuitry whose outputs are connected to the electrodes (E_i) of the DMFB. As shown in Figure 15(b), Trojans as simple as a multiplexer together with a few Boolean gates can effectively divert the control of the DMFB. The trigger controls the output of MUX and alters the droplet movement. The trigger signal can be derived from the on-chip sensors, droplets, underlying electrodes, or the integrated logic circuit by leveraging the interdisciplinary nature of DMFBs.

Note that HT attacks on the integrated logic circuit of a DMFB is analogous to the attacks on the control circuitry of an FMFB discussed in Section 5.2.2 and can be performed following the same workflow. This similarity is due to the fact that both the on-chip control circuit for FMFB and the logic circuit for the DMFB are deployed to generate electric signals sent to the external pins, which is irrelevant of the mechanism of the underlying fluidic system. As for the impact, HT attacks on the integrated logic circuit of an FPPC-DMFB disturb the correct pin-sharing scheme and invalidate the actuation sequence for the experiment and result in the failure of the bioassay execution.

Fig. 16. Framework of pre-synthesis Trojan insertion on DMFBs by FSM manipulation.

(a) FMFB layout from BioChip Simulator [23].

(b) Reverse engineered layout of the sample.

Fig. 17. Demonstration of layout-level RE of FMFBs.

5.3.3 HT Attack on FSM. Pre-synthesis Trojan insertion on traditional digital circuits has been discussed in Reference [43]. Figure 16 shows how we adapt the FSM-level Trojan attack to DMFBs. Assuming the attacker knows the design description of the original DMFB, he first reverse engineers the design to FSM model. The attacker then describes the functionality of the Trojan using FSM and merges it with the original FSM. To make the Trojan inseparable from the original design, it is beneficial to construct the Trojan using the elements that belong to the genuine design. The Trojan FSM can be inserted directly into the control signal generation module on the target DMFB [19] and disturbs the execution of the bioassay. Alternatively, the HT can be embedded in the error dictionary FSM to invalidate error recovery of the DMFB [24]. Control-level FSM manipulation attacks have been shown to incur low area and power overhead [43].

6 EXPERIMENTAL RESULTS

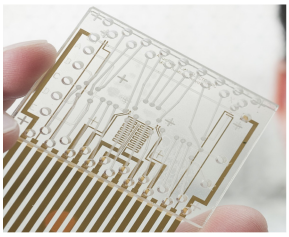
In this section, we present the attack results of RE and HT on MFBs in various levels, proving the feasibility and the effectiveness of the proposed attacks.

6.1 RE Attack on FMFB

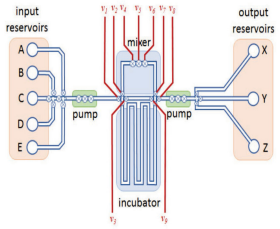
We evaluate the hardware description-based layout RE using the test cases in BioChip Simulator [23] and the image-based layout RE on a commercial FMFB using an image from the website [26]. Since the current version of the BioChip Simulator does not assign valves to control modules such as detector, heater, and filter, experimental results of FMFB protocol RE are not shown here.

6.1.1 Hardware Description-based Layout RE. Figure 17 illustrates the results of the description-based layout RE. Hardware description similar to the the example shown in Figure 3(a) is analyzed to identify the components and connections in the layout. Comparing the reverse-engineered DAG representation of the architecture shown in Figure 17(b) with the schematic layout shown in Figure 17(a), one can see that the recovered DAG has the same structural information as the original design, indicating the effectiveness of our layout RE.

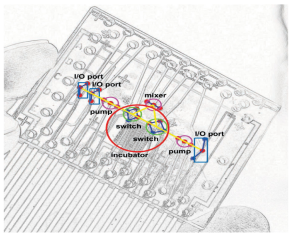
6.1.2 Image Analysis-based Layout RE. An alternative approach to reconstruct the layout of the FMFB is image analysis. As opposed to the attack flow of traditional circuits, depackaging and delayering is not performed on the target FMFB, since its transparent nature allows pure image analysis for RE. Given the image of a commercial (Figure 18(a)), the labeled components and



(a) An image of a commercial FMFB from the website [26].



(b) Schematic architecture of the FMFB.



(c) Reverse Engineered layout using image analysis.

Fig. 18. Demonstration of the image-based hardware layout RE.

connections are reconstructed. The image-based layout RE is successful, since the layout recovered in Figure 18(c) is consistent with the schematic architecture shown in Figure 18(b).

6.1.3 Discussion of Protocol and FSM RE Attacks. The experimental results of protocol RE and FSM RE attacks on FMFBs are not provided in this article, since the only open-sourced FMFB simulator [23] does not output the valve actuation table files nor the visualization frames for the target protocol. The successful recovery of the FMFB protocol or FSM requires the knowledge of both the FMFB architecture and the valve actuation tables. It is intuitive to see that protocol RE attacks on an FMFB is analogous to the one on a DMFB, both involving three main steps: recovering the movement of flows/droplets in each time step; deducing the functional operations; and assembling the executed operations to reconstruct the target protocol. As such, the proposed FMFB protocol RE attack is effective assuming the availability of the valve control table.

6.2 RE Attack on DMFB

Due to the limited resources and access to commercial DMFBs, we demonstrate our protocol RE attack in the synthesis tool. Modification to apply the attack on real DMFBs should be straightforward, since the actuation sequence and recorded video frames obtained from the physical world have the same representation as the output of the synthesis tool. The performance of the protocol RE attack is assessed by the portion of correctly characterized operations **O**. The protocol is visualized as a DAG, while it is worth noticing that one protocol can have multiple equivalent DAG descriptions. Our software is implemented in Matlab 2017a on a 64 bit PC with Inter Core i7, 3.5 GHz, 32 G RAM.

6.2.1 Actuation Sequence-based Protocol RE. Table 1 summarizes the evaluation results of the protocol RE attack on DMFBs, suggesting the generality and scalability of our methodology. The DMFB specification is given by the chip dimension ($height \times width$). The protocol runtime and the number of involved operations are given by execution cycles and nodes number, respectively. As can be seen from Table 1, the time overhead is dependent on both the number of nodes and the execution cycles of the protocol. Due to the lack of knowledge about the sensors position on the DMFB platform, the label "Detect" produced by the simulation tool is not supported by our current framework. The reason is that the behavior of the droplet during detection is the same as the one in static phase, meaning that the software cannot distinguish "Detect" from "Store."

6.2.2 Video Analysis–based Protocol RE. We demonstrate the video-based protocol RE on an FPPC-DMFB running a PCR bioassay. The original and the recovered sequencing graph are shown in Figure 19(a) and Figure 19(b), respectively. Each node denotes an operation annotated with corresponding properties. Although the visualization is not the same, these two DAGs are characterized by the same nodes and edges, suggesting that the attack succeeds.

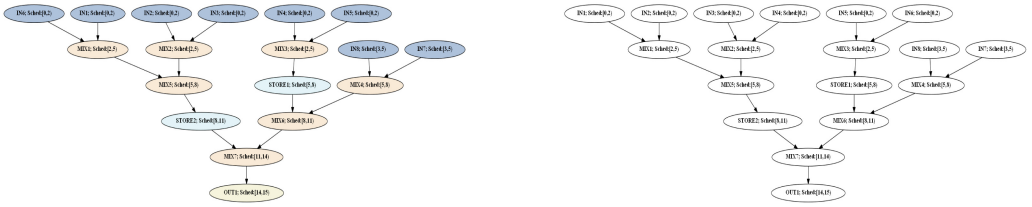
Table 1. Experimental Results of Actuation Sequence-based Protocol RE on DMFBs

Protocol	*Chip Dimension*	*Execution Cycles*	*Nodes Number*	*RE Accuracy*	*RE Time(s)*
Two Dilution	8*8	554	8	100%	1.099
PCR	15*12	1,458	16	100%	1.392
Protein Mix	8*121	8,141	58	93.1%	34.758
InVitro	19*15	3,705	80	80%	12.233

Benchmarks are evaluated in the open source synthesis tool [13]. Attack time is reported as the total time of the algorithm.

Table 2. Experimental Results of Proposed HT Attack on Various Benchmarks

Benchmark	*Area*	*# Components*	*# Valves*	*# Gates*	$P_{Trigger}$	R_t	R_c	$\|d\|$	*HT Time(s)*
AquaFlux	17,500	17	26	28	0.0647	0.9253	0.5831	1.0956	3.1522
Urbanski	19,600	11	48	31	0.0611	0.9313	1.0261	1.3871	3.0155
PCR1s	16,100	11	57	37	0.0667	0.9472	0.7954	1.2386	3.2519
PCR2s	18,900	16	77	95	0.0038	0.9719	0.9421	1.3536	4.9499
PCR3s	20,800	23	96	146	0.008	0.9765	0.8121	1.2701	6.1217
EA1s	30,000	20	92	297	0.0042	0.9746	1.0505	1.4331	10.2675



(a) Original DAG of a PCR assay. (b) Reconstructed DAG of the same assay

Fig. 19. Demonstration of video-based protocol RE of a PCR assay.

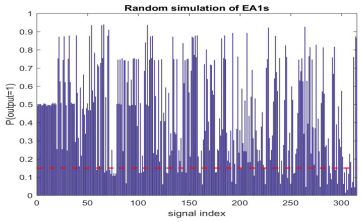
6.3 HT Attack on Control Circuit of FMFB

We demonstrate the feasibility of the malware attack by inserting Trojans that can be as simple as a single gate. We evaluate the Trojan attack on all benchmarks in Reference [29] following the steps in Algorithm 3 and summarize the results in Table 2. The implementation of the 2-AND trigger requires only two microvalves and three microchannels, which incurs negligible cost for the highly integrated FMFB. The results show that the attack is effective and has small timing overhead (all attacks finish within one minute). The time consumption is dominated by netlist extraction and HT verification, which scale linearly and quadratically with the number of gates in the control circuit, respectively. A case study of the benchmark EA1s is discussed below.

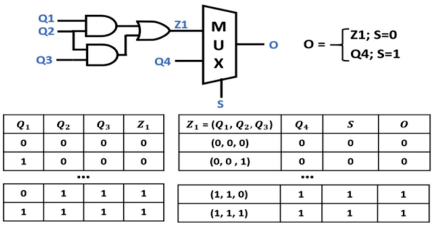
Netlist Extraction: The adversary applies the method in Section 5.2.2 to reverse engineer the on-chip control circuit. Our software tool takes the routed layout description as input and generates a standard Verilog file as output. The extracted netlist of EA1s has 18 primary inputs (PI), 60 primary outputs (PO), and 237 wires, which are distinguished by prefixes Q, Z, and N.

Logic Simulation: 10K random testing vectors are generated and applied on the netlist. The activation probability P_1 is estimated from the statistical test and visualized in Figure 20(a). The threshold is set at $\epsilon = 0.5$ to ensure that compatible nodes can be found in one round. The obtained set $\mathbf{O} = \{O_1, \ldots, O_N\}$ includes $N = 163$ nodes that satisfy $P_1 \leq \epsilon$.

Static Timing Analysis: STA is performed on the netlist to compute slacks using the delay specification. The updated set $\mathbf{O}' = \{O'_1, \ldots, O'_M\}$ is obtained by deleting critical nodes from the set



(a) Activation frequency of the EA1s.



Q_1	Q_2	Q_3	Z_1
0	0	0	0
1	0	0	0
...			
0	1	1	1
1	1	1	1

$Z_1 = (Q_1, Q_2, Q_3)$	Q_4	S	O
(0, 0, 0)	0	0	0
(0, 0 , 1)	0	0	0
...			
(1, 1, 0)	1	1	1
(1, 1, 1)	1	1	1

(b) Example of control-value computation.

Fig. 20. (a) Activation probability analysis of EA1s benchmark using 10K random testing vectors; (b) Example of control values computation for a multiplexer. Truth tables of $Z1$ and MUX are shown.

O. Under our experiment setting, only $M = 50$ nodes are left after checking the timing violation. Timing-critical paths are identified and avoided when the HT gate is placed.

HT Verification: In our experiment, a single 2-AND gate is used as the Trojan trigger and requires two compatible signals as inputs. MiniSat solver [1] is used to determine if a pair of two signals can be concurrently activated to 1. If the trigger condition ($O'_i = 1, O'_j = 1$) is satisfiable, then the pair is recorded in the set **Pair** and the corresponding $P_{Trigger}$ is computed. For EA1s benchmark, three pairs are found to be satisfiable and the one with minimal $P_{Trigger}$ is selected as trigger inputs. The selected pair (N_{226}, Z_{12}) are rewritten as Boolean functions of PIs using the extracted netlist:

$$N_{226} = Q_{13} \cdot \overline{Q_{14}} \cdot Q_{15} \cdot \overline{Q_{16}} \cdot \overline{Q_{17}} \cdot Q_{18}; \quad Z_{12} = \overline{Q_7} \cdot \overline{Q_9}. \tag{4}$$

Therefore, the trigger condition can be simplified as: $HT = \overline{Q_7} \cdot \overline{Q_9} \cdot Q_{13} \cdot \overline{Q_{14}} \cdot Q_{15} \cdot \overline{Q_{16}} \cdot \overline{Q17} \cdot Q_{18} = 1$. The resulting Trojan is active only when the primary inputs of the control circuit satisfy the trigger condition $(Q_7, Q_9, Q_{13}, Q_{14}, Q_{15}, Q_{16}, Q_{17}, Q_{18}) = (0, 0, 1, 0, 1, 0, 0, 1)$. Consequently, the theoretical trigger probability can be computed as $P_{Trigger} = \frac{1}{2^8} = 0.0039$.

Detectability Characterization: The malicious benchmark is devised by placing the Trojan on a non-critical timing path of the netlist. The activation probability of HT is approximated by random simulation $P(HT = 1) = 0.0042$, which is close to the theoretical value. The change of switching activity is quantified by the transition ratio $R_t = 0.9746$, which indicates that the inserted Trojan does not increase the power consumption. The small structural observability $R_c = 1.0505$ suggests that the Trojan is difficult to detect by visual inspection.

Discussion: We focus on HT attacks in the control circuitry of the FMFB instead of the flow components, since HT attack in the control circuit is more generic and flexible. By inserting an HT in specific gates on the control circuit, the adversary can indirectly disturb the functionality of the corresponding flow components whose microvalves are controlled by the attacked gates. As such, HT attacks on the component microvalves of the FMFB can be achieved by performing HT attack on the control circuitry, suggesting the flexibility of our proposed attack methodology.

7 COUNTERMEASURES

In this section, we present systematic countermeasures to prevent or detect the proposed RE and HT attacks on MFBs. We show how to adapt traditional defense techniques such as camouflaging and obfuscation to protect MFBs and analyze the corresponding overhead.

7.1 Defense Against RE

Camouflaging: Camouflaging is a common defense mechanism in silicon ICs that aims to hinder image-based RE of the gate-level netlist. To find the design of the FMFB, dummy valves and

Type	I/O	Switch	Mixer	Pump
Pattern				
Number of Valves	3	4	5	3

(a) Component library.

ALGORITHM 4: FMFB Layout Camouflaging Work-flow.

Input: Original FMFB layout T; Component library L.
Output: Camouflaged FMFB layout T^c
$T_{src} \leftarrow Select\_Source\_Components(T)$
for *each component* $e \in T$ **do**
  if $e \in T_{src}$ **then**
    $e^* \leftarrow Find\_Feasible\_Camouflaging\_Conversions(e, L)$
    $e^t \leftarrow Select\_Target\_Component(e^*)$
    $T^c \leftarrow Replace\_One\_Component(T, e, e^t)$
Return: Camouflaged FMFB layout T^c

(b) Workflow of layout camouflaging.

Fig. 21. Illustration of an FMFB component library (a) and workflow of FMFB layout camouflaging (b).

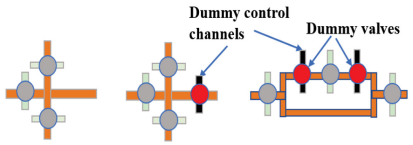
dummy channels can be inserted into the original layout. In this circumstance, the component library obtained by the attacker is useless and even misleads him to an incorrect component-level abstraction. Correct pressure signals needs to be applied on dummy valves for ensuring the original functionality of camouflaged component. Camouflaging decouples the relationship between the appearance of the component and its functionality, misleading the attacker to extract the incorrect component-level layout.

The workflow of FMFB layout camouflaging is outlined in Algorithm 4. The designer first selects a subset of components (referred to as "source components") to be camouflaged. Camouflaging one source component involves three steps: **(1)** identifying feasible camouflaging conversions by comparing the structural composition of the source component and other components in the library. If the source component can be converted to another type of component (referred to as the "target component") by adding microvalves in specific locations, the conversion is feasible; **(2)** selecting one target component from all feasible camouflaging options based on the designer's requirement; **(3)** converting the source component to the target component by adding dummy valves and dummy channels. The positions of the required dummy elements are determined from a straightforward comparison between the microvalves composition of the source component and selected target component. These three steps are repeated for all source components selected by the designer.

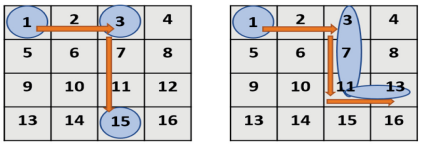
A metric to evaluate the effectiveness of a camouflaging approach is the Hamming distance between the original component netlist and the one reconstructed from the camouflaged layout [35]. The area overhead of camouflaging is induced by the addition of dummy elements and is determined by the camouflaging decision made by the designer. Since the addition of dummy valves is incremental, the area overhead increases approximately linearly with the number of source components selected by the designer. It is intuitive that a more complex camouflaged design has higher security against attacks while incurring larger area overhead. The trade-off between security and area overhead can be utilized by the designer based on the specific application and concerns.

Figure 21(a) shows an example of the component library used in FMFB layout design. The overhead of any feasible camouflaging conversion from a source component to a target component can be quantified using the structure information about each component in the library. Figure 22(a) demonstrates how to camouflage an I/O port as a switch or a mixer by inserting dummy valves and channels in different locations. The overhead of these two camouflaging conversions in terms of the number of additional microvalves is 1 and 2, respectively. Our proposed camouflaging technique is able to profile the overhead of a layout camouflaging option in terms of the number of dummy valves based on the component library. Such characterization, in turn, helps the designer to leverage the trade-off between camouflaging overhead and security level as discussed above.

Obfuscation: Protocol RE attacks on DMFBs are feasible and have been demonstrated in Section 4.2.2, threatening the IP of the designer. We show how to adapt obfuscation to prevent the



(a) Camouflaging component library.



(b) Effect of actuation sequence obfuscation.

Fig. 22. (a) Camouflage the layout of FMFB's component. The original structure of I/O port, camouflaging the I/O port as a switch or a mixer are shown from left to right, respectively; (b) RE results from obfuscated actuation sequence. The attacker will extract incorrect operations from the encrypted sequence.

actuation-sequence-based protocol RE. More specifically, obfuscation requires the control signals to be encrypted before the transmission in the untrusted communication channel. The secret key for encryption can be obtained from the license issued by the foundry or the unique physical properties of a manufactured DMFB [4, 16]. Let us assume the assay lasts T clock cycles and the length of actuation sequence in each cycle is L. The total actuation sequence $\mathbf{S}$ is a T-by-L matrix with the element $s_{i,j}$ indicating the actuation status of jth electrode in clock cycle i. The designer can encrypt $\mathbf{S}$ using XOR operation $\mathbf{S_e} = \mathbf{S_o} \oplus \mathbf{e_k}$, where $\mathbf{e_k}$, $\mathbf{S_o}$, $\mathbf{S_e}$ denotes the symmetric key, original sequence, and encrypted one, respectively. The control signal is decrypted using the same secret key $\mathbf{S_d} = \mathbf{S_e} \oplus \mathbf{e_k}$ before being connected to the external pins.

Integrating XOR gates to DMFBs is feasible and has low area overhead, since the manufacturing process of DMFB is compatible with CMOS techniques and the size of a XOR gate is much smaller than the electrode cell. The complexity of both encryption and decryption is $O(TL)$, suggesting the scalability of our obfuscation scheme. Figure 22(b) demonstrates the effect of actuation sequence obfuscation on a 4-by-4 direct-addressed DMFB. In this example, we set the parameters to $T = 6, L = 16$ and $\mathbf{e_k} = 0010011001110110$. Each row in $\mathbf{S}$ is XORed with $\mathbf{e_k}$ before transmission. The authentic trajectory and recovered one using the RE attack in Section 4.2.1 are shown in the left and right image, respectively, suggesting the effectiveness of obfuscation.

Security of obfuscation: The security of our actuation sequence obfuscation scheme against brute-force attacks is determined by the key length. More specifically, the attack time has an exponential relation with respect to the key length. Therefore, employing a longer encryption key enhances the security of our obfuscation scheme against brute-force attacks with theoretical guarantee. Advanced attacks on digital obfuscation that aim to find the correct key typically require querying the chip with specific inputs and observing the corresponding outputs. These types of satisfiability (SAT)-based attacks are not feasible for consumable DMFBs due to the high cost. The attacker needs to purchase samples and reagents for his desired biomedical protocol to run a sufficient number of experiments on the DMFB, which may incur excessive cost.

SAT-based attacks on our activation encrypted DMFBs are expensive due to the following reasons: (i) Existing SAT-based attacks require the knowledge of the gate-level netlist of the encrypted circuit and oracle access to the corresponding active chip for identifying the correct key bits. Our obfuscation scheme employs the combination of Physical Unclonable Function (PUF) response and the additional key bits to encrypt the FSM that represents DMFB behaviors instead of encrypting the gate-level netlist of the DMFB. Therefore, the adversary cannot directly apply conventional SAT attacks on the locked DMFB. The complexity of unlocking our encrypted DMFB is exponential with respect to the key length. (ii) Both the target DMFB and the samples/reagents are consumable. Furthermore, the biochemical samples and the biochip consist of vulnerable components such as proteins and enzymes. As such, they might be contaminated or deactivated due to the environmental variation during the transportation process, resulting in incorrect

Table 3. Experimental Results of the Control Value–based HT Detection on the Devised Malicious Benchmarks

Benchmark	*AquaFlux*	*Urbanski*	*PCR1s*	*PCR2s*	*PCR3s*	*EA1s*
False Positive Rate	0	3.125%	0.53%	0	0	0.34%
Baseline Runtime (s)	3.5077	3.7792	4.1538	11.022	17.2205	283.8529
Parallel Runtime (s)	2.405	2.284	2.609	8.563	11.536	198.05
Speedup	1.458	1.654	1.592	1.287	1.493	1.433

The runtime of the baseline implementation and the sub-circuits parallel implementation is measured.

measurements of the experiment performed by the attacker. Let us consider the Polymerase Chain Reaction (PCR) experiment as an example of real-world applications. A consumable DMFB and a single PCR sequencing kit (containing the required reagents) from the Oxford Nanopore company cost 1000 and 600 dollars, respectively. To find the correct key bits used by our digital obfuscation, the attacker might need to spend millions of dollars, which is prohibitively expensive.

7.2 Defense Against HT

Framework of Control-value based Trojan Detection for FMFBs: We present an innovative systematic framework for identifying suspicious gates on FMFBs. Control value (CV) of a primary input is defined as the probability that the change of the input results in the change of the gate output. For each input Q_i, all other input columns in the truth table are held fixed and $CV(Q_i)$ is represented by the fraction of rows whose outputs are influenced by Q_i [44]. A gate g with N input wires has a vector **CV** with N elements, and its control value $CV(g)$ is defined as the median of the vector. For an intermediate output wire w, we derive the Boolean expression of w in terms of PIs $w = f(Q_1, \ldots, Q_K)$ and define $CV(w)$ as the weighted sum of control values of PIs: $CV(w) = \sum_{i=1}^{K} t_i \times CV'(Q_i)$. The weight t_i is the control value of Q_i computed from the truth table of w. $CV'(Q_i)$ is obtained by replacing w with the string $(Q_1, \ldots, Q_K)$ in the truth table of g and holding all inputs fixed except for Q_i. If multiple wires are replaced by strings, then we treat each string as an independent set of PIs.

Our approach leverages the fact that the trigger has small effect on the outputs to evade functional testing, which means the control value of the trigger should be small. The proposed HT detection scheme computes CVs of all gates and flags the gates whose CVs are smaller than the predefined threshold. Algorithm 5 outlines the procedure of the detection method. The function $EstCV(T, w, M)$ is used to estimate the control values of the wires while restricting the computational overhead. M pairs of rows are randomly selected from the truth table T for approximating $CV(w)$ instead of using the exponentially large table.

The main difference between our work and Reference [44] is that we define the CV of intermediate wire as the weighted sum of $CV(PI)$ while Reference [44] treats the multiple-bit wire as a set of independent wires. Figure 20(b) shows how to compute the control value of a multiplexer. Control values of independent wires are computed by the definition: $CV(S) = CV(Q_4) = \frac{8}{2^4} = 0.5$. For intermediate wire $Z_1 = Q_1 \cdot Q_2 + Q_2 \cdot Q_3$, we replace Z_1 with the string (Q_1, Q_2, Q_3) in the truth table of MUX. The truth table of Z_1 is used to compute the weight of each PI component: $t_1 = CV(Q_1) = \frac{1}{2^2} = 0.25, t_2 = CV(Q_2) = \frac{2}{2^2} = 0.5, t_3 = CV(Q_3) = \frac{1}{2^2} = 0.25$. The control value of each PI on the MUX output is computed: $CV'(Q_1) = CV'(Q_3) = \frac{2}{2^4} = 0.125, CV'(Q_2) = \frac{6}{2^4} = 0.75$. Therefore, the weighted sum $CV(Z_1) = \sum_{i=1}^{3} t_i \times CV'(Q_i) = 0.4375$. The control value of MUX is the median of the vector $CV(MUX) = median\{CV(Z_1), CV(Q_4), CV(S)\} = 0.5$.

Experimental Results of Proposed Trojan Detection: We evaluate the proposed CV-based Trojan detection technique on the malicious FMFB benchmarks devised in Section 5.2. Experimental results of time consumption and false positive rates are summarized in Table 3. In contrast

to UCI, our detection scheme is able to operate without false negative, since we flag gates with low influence on the outputs instead of identifying completely unused gates. In addition, our defense method has small false positive rates: Less than 1% of gates are reported as suspicious in most cases. The proposed countermeasure is scalable due to the usage of random sampling in control value approximation.

ALGORITHM 5: Framework of Trojan Detection for FMFB

Input: Extracted netlist **Net** with N gates, user defined threshold ϵ and number of selected pairs M
Output: Suspicious gates that trigger the Trojan
for $0 < i \leq N$ **do**
  gate $g \leftarrow \textbf{Net}(i), T_g \leftarrow TruthTable(g)$;
  for $0 < j < \#Inputs(g)$ **do**
    Pick jth input $w \leftarrow Inputs(g)[j]$;
    if w *is intermediate wire* **then**
      $w \leftarrow f(Q_i, ...Q_K)$; $T_w \leftarrow TruthTable(w)$;
      Replace w with $(Q_i, ...Q_K)$ in T_g ;
      for $0 < m \leq K$ **do**
        $t_i = EstCV(T_w, Q_i, M)$; $CV'(Q_i) = EstCV(T_g, Q_i, M)$;
      $CV(w) = \sum_{i=1}^{K} t_i \times CV'(Q_i)$;
    else
      $CV(w) = EstCV(T_g, w, M)$;
  $CV(g) = median\{CV(w_1), ..., CV(w_{max}\}$;
  if $CV(g) < \epsilon$ **then**
    Flag gate g as suspicious;

Complexity Analysis of Control Value–based HT Detection: The computation complexity of our proposed HT detection scheme has an approximately linear relation with respect to the number of gates in the control circuitry, since the control value is computed for each gate. Furthermore, the time overhead can be reduced by setting the parameter M in the function $EstCV(T_g, Q_i, M)$. Leveraging the fact that the control circuit can be partitioned into independent sub-circuits whose signals are uncorrelated, our HT detection scheme can be accelerated and runs in parallel in the identified sub-circuits. The results of the parallel implementation of the control value–based HT detection are summarized in Table 3, and an average speedup of 1.49 is achieved compared to the baseline implementation. As such, the proposed HT detection scheme is scalable to large MFBs.

Countermeasures against Hardware Trojans on DMFBs: Obfuscation and locking can be adopted to prevent HT attacks on DMFBs [2]. The architecture and the actuation sequence of a DMFB can be obfuscated such that the adversary cannot extract any useful information, and designing a stealthy Trojan might be prohibitive.

8 DISCUSSION

8.1 Summary of MFB Security

Table 4 provides the taxonomy of the state-of-the-art attacks and countermeasures proposed for DMFBs and FMFBs. As can be seen from the table, our work is the first attempt that identifies the vulnerabilities of both types of MFBs to HT and RE attacks.

8.2 Satisfiability-based Attack

Despite the RE attacks demonstrated in Section 4, satisfiability (SAT) can be to used to perform RE with less knowledge of the target MFB. SAT has been well explored to attack traditional digital ICs [8, 30], and it is intuitive that SAT is also helpful for RE attacks on MFBs. For instance, SAT can be used to reverse engineer the pin-mapping scheme of an FPPC-DMFB. Given a physical DMFB, the attacker can apply various actuation sequences to the target DMFB and observe the

Table 4. Comparison of Existing Attacks and Countermeasures for DMFBs and FMFBs

Types	Attack	Defense	Description
DMFB	IP Piracy	Encryption	[4]: Insert fluidic multiplexers (FMUXs) in the original sequence graph. The execution of the protocol is correct only when the true key is applied on FMUXs
	IP Piracy	Locking	[6]: Insert dummy mix-split operations in the original sequence graph to hide its functionality. Correct secret key shall be applied on the dummy operations to unlock the DMFB.
	IP Piracy	PUF	[16]: Lock the DMFB by adding additional FSM. Leverages the intrinsic manufacturing variation of electrode on DMFB to generate the secret key used for device unlocking.
	Assay Manipulation	–	[3]: Demonstrate two result-manipulation attacks that change the sample concentration or modify the golden calibration curve.
	DoS	Checkpointing	[42]: Propose a randomized checkpointing method to detect malicious modification of droplets routes
	Hardware Trojans	Obfuscation	This work. Demonstrate HT attacks on DMFB in the integrated logic IC level and the FSM level. Suggest obfuscation to prevent HT attacks.
	Reverse Engineering	Obfuscation / Camouflaging	This work. Propose a systematic approach to recover the protocol mapped to a general-purpose DMFB. Suggest encrypting the actuation sequence before the transmission to prevent RE attacks.
FMFB	DoS	–	[41]: Explore the vulnerabilities of microfluidic crossbars and show how fluids can be directed to incorrect locations by fault-injection attacks.
	Hardware Trojans	Trojan Detection	This work. Propose a systematic method for HT insertion in the control circuitry of an FMFB to divert the bioassay execution. Present a control value-based HT detection scheme as the countermeasure.
	Reverse Engineering	Camouflaging / Obfuscation	This work. Present a methodological approach to reconstruct the layout and protocol of an FMFB. Demonstrate how camouflaging and obfuscation can be adopted to prevent RE attacks.

corresponding droplet movements. Using SAT constraints and the input-output observations, the attacker can deduce the connectivity between the external pins and the electrodes. However, the SAT-based RE attack on MFBs might be prohibitive due to the high cost, since most current MFBs are consumable, meaning that they can only be used a limited number of times. The cost of samples and reagents to conduct the attack is also non-negligible.

9 CONCLUSION

We present the first systematic hardware attacks and propose corresponding countermeasures for two popular microfluidic biochips: flow-based and droplet-based microfluidic biochips. We demonstrate attack methodologies for reverse engineering as well as Hardware Trojans and evaluate the attacks on various benchmarks. The vulnerabilities of MFBs are exploited in different phases of the design flow and the supply chain. Furthermore, we show how to adapt traditional defense techniques for applicability on MFBs. Component camouflaging, actuation obfuscation, and control-value-based Trojan detection are presented with their overheads discussed. We want to emphasize

the importance and necessity of taking security as well as privacy as metrics in the design flow of MFBs. The security of MFBs is worth more attention from researchers and companies.

REFERENCES

[1] MiniSAT. 2015. In *Proc. SAT-05: 8th Int. Conf. on Theory and Applications of Satisfiability Testing*. http://minisat.se/MiniSat.html.

[2] Sk Subidh Ali, Mohamed Ibrahim, Jeyavijayan Rajendran, Ozgur Sinanoglu, and Krishnendu Chakrabarty. 2016. Supply-chain security of digital microfluidic biochips.*IEEE Comput.* 49, 8 (2016), 36–43.

[3] Sk Subidh Ali, Mohamed Ibrahim, Ozgur Sinanoglu, Krishnendu Chakrabarty, and Ramesh Karri. 2015. Security implications of cyberphysical digital microfluidic biochips. In *Proceedings of the International Conference on Computer Design (ICCD'15)*. IEEE Computer Society, 483–486.

[4] Sk Subidh Ali, Mohamed Ibrahim, Ozgur Sinanoglu, Krishnendu Chakrabarty, and Ramesh Karri. 2016. Microfluidic encryption of on-chip biochemical assays. In *Proceedings of the Biomedical Circuits and Systems Conference (BioCAS'16)*. IEEE, 152–155.

[5] Sk Subidh Ali, Mohamed Ibrahim, Ozgur Sinanoglu, Krishnendu Chakrabarty, and Ramesh Karri. 2016. Security assessment of cyberphysical digital microfluidic biochips. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 13, 3 (2016), 445–458.

[6] Sukanta Bhattacharjee, Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2018. Locking of biochemical assays for digital microfluidic biochips. In *Proceedings of the IEEE 23rd European Test Symposium (ETS'18)*. IEEE.

[7] Swarup Bhunia, Michael S. Hsiao, Mainak Banga, and Seetharam Narasimhan. 2014. Hardware trojan attacks: Threat analysis and countermeasures. *Proc. IEEE* 102, 8 (2014), 1229–1247.

[8] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019. GenUnlock: An automated genetic algorithm framework for unlocking logic encryption. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'19)*. IEEE, 1–8.

[9] Huili Chen, Seetal Potluri, and Farinaz Koushanfar. 2017. BioChipWork: Reverse engineering of microfluidic biochips. In *Proceedings of the IEEE International Conference on Computer Design (ICCD'17)*. IEEE, 9–16.

[10] Trung Anh Dinh, Shigeru Yamashita, and Tsung-Yi Ho. 2015. An optimal pin-count design with logic optimization for digital microfluidic biochips. *IEEE Trans. Comput.-aided Des. Integ. Circ. Syst.* 34, 4 (2015), 629–641.

[11] Daniel Grissom. 2014. *Design of Topologies for Interpreting Assays on Digital Microfluidic Biochips*. Ph.D. Dissertation. UC Riverside.

[12] Daniel Grissom and Philip Brisk. 2013. A field-programmable pin-constrained digital microfluidic biochip. In *Proceedings of the 50th Design Automation Conference*. ACM, 46.

[13] Daniel Grissom, Kenneth O'Neal, Benjamin Preciado, Hiral Patel, Robert Doherty, Nick Liao, and Philip Brisk. 2012. A digital microfluidic biochip synthesis framework. In *Proceedings of the IEEE/IFIP International Conference on Very Large Scale Integration of System-on-Chip (VLSI-SOC'12)*. IEEE, 177–182.

[14] Daniel T. Grissom, Jeffrey McDaniel, and Philip Brisk. 2014. A low-cost field-programmable pin-constrained digital microfluidic biochip. *IEEE Trans. Comput.-aided Des. Integ. Circ. Syst.* 33, 11 (2014), 1657–1670.

[15] Zimu Guo, Jia Di, Mark M. Tehranipoor, and Domenic Forte. 2017. Obfuscation-based protection framework against printed circuit boards unauthorized operation and reverse engineering. *ACM Trans. Des. Automat. Electron. Syst.* 22, 3 (2017), 54.

[16] Ching-Wei Hsieh, Zipeng Li, and Tsung-Yi Ho. 2017. Piracy prevention of digital microfluidic biochips. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'17)*. IEEE, 512–517.

[17] Kai Hu. 2015. *Optimization, Testing and Design-for-Testability of Flow-Based Microfluidic Biochips*. Ph.D. Dissertation. Duke University.

[18] Kai Hu, Bang-Ning Hsu, Andrew Madison, Krishnendu Chakrabarty, and Richard Fair. 2013. Fault detection, real-time error recovery, and experimental demonstration for digital microfluidic biochips. In *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 559–564.

[19] Kai Hu, Mohamed Ibrahim, Liji Chen, Zipeng Li, Krishnendu Chakrabarty, and Richard Fair. 2015. Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform. In *Proceedings of the Biomedical Circuits and Systems Conference (BioCAS'15)*. IEEE, 1–4.

[20] Kai Hu, Feiqiao Yu, Tsung-Yi Ho, and Krishnendu Chakrabarty. 2014. Testing of flow-based microfluidic biochips: Fault modeling, test generation, and experimental demonstration. *IEEE Trans. Comput.-aided Des. Integ. Circ. Syst.* 33, 10 (2014), 1463–1475.

[21] Mohamed Ibrahim and Krishnendu Chakrabarty. 2015. Efficient error recovery in cyberphysical digital-microfluidic biochips. *IEEE Trans. Multi-scale Comput. Syst.* 1, 1 (2015), 46–58.

[22] Mohamed Ibrahim, Krishnendu Chakrabarty, and Kristin Scott. 2017. Synthesis of cyberphysical digital-microfluidic biochips for real-time quantitative analysis. *IEEE Trans. Comput.-aided Des. Integ. Circ. Syst.* 36, 5 (2017), 733–746.

[23] Illumina. 2017. *BioChip Simulator*. Retrieved from https://sites.google.com/site/biochipsimulator/.
[24] Yan Luo, Krishnendu Chakrabarty, and Tsung-Yi Ho. 2013. Error recovery in cyberphysical digital microfluidic biochips. *IEEE Trans. Comput.-aided Des. Integ. Circ. Syst.* 32, 1 (2013), 59–72.
[25] Lidija Malic, Teodor Veres, and Maryam Tabrizian. 2009. Biochip functionalization using electrowetting-on-dielectric digital microfluidics for surface plasmon resonance imaging detection of DNA hybridization. *Biosens. Bioelect.* 24, 7 (2009), 2218–2224.
[26] Microfluidic Innovations LLC. 2017. Thin film based microfluidic devices. Microfluidic Innovation LLC. https://www.sbir.gov/sbirsearch/detail/13282.
[27] Wajid Hassan Minhass, Paul Pop, and Jan Madsen. 2011. System-level modeling and synthesis of flow-based microfluidic biochips. In *Proceedings of the 14th International Conference on Compilers, Architectures and Synthesis for Embedded Systems*. ACM, 225–234.
[28] T. Nguyen, S. Ahrar, P. Duncan, and E. Hui. 2011. Microfluidic finite state machine for autonomous control of integrated fluid networks. *Proc. Micro. Total Anal. Syst.* 2 (2011), 741–743.
[29] P. Pop, W. H. Minhass, and J. Madsen. 2016. *Microfluidic Very Large Scale Integration (VLSI): Modeling, Simulation, Testing, Compilation and Physical Synthesis*. Springer International Publishing.
[30] Seetal Potluri, Aydin Aysu, and Akash Kumar. 2020. SeqL: Secure scan-locking for IP-protection. In *Proceedings of the International Symposium on Quality Electronic Design (ISQED'20)*. IEEE.
[31] Seetal Potluri, Paul Pop, and Jan Madsen. 2018. Design-for-testability of on-chip control in mVLSI biochips. *IEEE Des. Test* 36, 1 (2018), 48–56.
[32] Seetal Potluri, Paul Pop, Jan Madsen, and Alexander Rudiger Schneider. 2018. *Microfluidic Valve*.U.S. Patent WO 2018 104 516.
[33] Seetal Potluri, Alexander Schneider, Paul Pop, Jan Madsen et al. 2017. Synthesis of on-chip control circuits for mVLSI biochips. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE'17)*. IEEE, 1799–1804.
[34] Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. 2013. Security analysis of integrated circuit camouflaging. In *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*. ACM, 709–720.
[35] Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri. 2014. A primer on hardware security: Models, methods, and metrics. *Proc. IEEE* 102, 8 (2014), 1283–1295.
[36] Masoud Rostami, Farinaz Koushanfar, Jeyavijayan Rajendran, and Ramesh Karri. 2013. Hardware security: Threat models and metrics.. In *Proceedings of the International Conference on Computer Aided Design (ICCAD'13)*. IEEE, 819–823.
[37] M. F. Schmidt. 2012. *Biochip Simulator—Flow-based Microfluidic Biochip Simulation*. Master's thesis. Technical University of Denmark, DTU Informatics.
[38] Vineeta Shukla, Fawnizu Azmadi Hussin, Nor Hisham Hamid, and Noohul Basheer Zain Ali. 2017. Advances in testing techniques for digital microfluidic biochips. *Sensors* 17, 8 (2017), 1719.
[39] Fei Su and Krishnendu Chakrabarty. 2008. High-level synthesis of digital microfluidic biochips. *ACM J. Emerg. Technol. Comput. Syst.* 3, 4 (2008), 1.
[40] Fei Su, William Hwang, and Krishnendu Chakrabarty. 2006. Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'06)*, Vol. 1. IEEE, 1–6.
[41] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2017. Security trade-offs in microfluidic routing fabrics. In *Proceedings of the IEEE 35th International Conference on Computer Design (ICCD'17)*. IEEE, 25–32.
[42] Jack Tang, Ramesh Karri, Mohamed Ibrahim, and Krishnendu Chakrabarty. 2016. Securing digital microfluidic biochips by randomizing checkpoints. In *Proceedings of the IEEE International Test Conference (ITC'16)*. IEEE, 1–8.
[43] Mohammad Tehranipoor and Farinaz Koushanfar. 2010. A survey of hardware trojan taxonomy and detection. *IEEE Des. Test Comput.* 27, 1 (2010).
[44] Adam Waksman, Matthew Suozzo, and Simha Sethumadhavan. 2013. FANCI: Identification of stealthy malicious logic using Boolean functional analysis. In *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*. ACM, 697–708.
[45] Qin Wang, Shiliang Zuo, Hailong Yao, Tsung-Yi Ho, Bing Li, Ulf Schlichtmann, and Yici Cai. 2017. Hamming-distance-based valve-switching optimization for control-layer multiplexing in flow-based microfluidic biochips. In *Proceedings of the 22nd Asia and South Pacific Design Automation Conference (ASP-DAC'17)*. IEEE, 524–529.
[46] Sheng Wei, Kai Li, Farinaz Koushanfar, and Miodrag Potkonjak. 2012. Hardware trojan horse benchmark via optimal creation and placement of malicious circuitry. In *Proceedings of the 49th Design Automation Conference*. ACM.
[47] Jie Zhang, Feng Yuan, and Qiang Xu. 2014. Detrust: Defeating hardware trust verification with stealthy implicitly triggered hardware trojans. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 153–166.

Received October 2017; revised April 2019; accepted January 2020