BioChipWork: Reverse Engineering of Microfluidic Biochips

Huili Chen
ECE Department
University of California, San Diego
La Jolla, CA
Email: huc044@ucsd.edu

Seetal Potluri
Xilinx Asia Pacific Pte. Ltd
5 Changi Business Park Vista
Singapore
Email: seetalp@xilinx.com

Farinaz Koushanfar
ECE Department
University of California, San Diego
La Jolla, CA
Email: farinaz@ucsd.edu

***Abstract*—Microfluidic biochip is an emerging platform that has wide applications in areas of immunoassays, DNA sequencing and point-of-care health service. This paper presents BioChipWork, the first practical framework for automatic reverse engineering and IP piracy of microfluidic biochips. Our work targets two types of presently available microfluidic biochips which are characterized based on working mechanisms: flow-based microfluidic biochip (FMFB) and droplet-based microlfuidic biochip (DMFB). More specifically, BioChipWork identifies two practical sets of reverse engineering attacks and demonstrates the attacks using our developed algorithm and an open source synthesis tool. In the first attack, the attacker extracts the hardware layout of the pertinent FMFB based on image analysis. In the second attack, the attacker reconstructs the proprietary protocol mapped onto the DMFB by analyzing the actuation sequence or the video frames recorded by the CCD camera. The proposed reverse engineering attacks are non-intrusive, scalable and easy to implement, rendering the IP of authentic owners in danger. As countermeasures to obscure the functional layout and reduce information leakage from side-channels, we suggest novel biochip camouflaging and obfuscation techniques.**

I. Introduction

With the growing need and progress in system miniaturization, Lab-on-a-Chip (LoC) technologies have been developed as miniaturized platforms to perform various experiments such as chemistry analysis, clinical diagnosis and environmental tests. Microfluidic biochip is an emerging branch in LoC that enables the automation of traditional laborious biomedical experiments, providing advantages such as low sample input, reduced human efforts, portability, and high throughput.

Microfluidic biochips are increasingly commercialized by companies such as Microfluidic Innovations LLC. [1] and Illumina [2]. However, current supply chain of biochips does not take security into account. This design hole renders the existing devices susceptible to various attacks such as result-manipulation, denial-of-service (DoS), counterfeiting, reverse engineering (RE) and intellectual property (IP) piracy. A number of recent works have highlighted the possibility of these attacks [3], [4]. Vulnerabilities of biochips may be misused to produce incorrect diagnosis outcomes and treatments, endangering patients' health.

This work was supported by ONR under grant number N00014-17-1-2500, AFOSR MURI under award number FA9550-14-1-0351, and NSF Trust-Hub under grant number CNS-1649423.

The supply chain of microfluidic biochips is analogous to the one of silicon ICs, which means the well known hardware-based attacks are mostly applicable to the existing biochips [4]. In particular, the following classic hardware-based attacks threat the security and privacy of biochips: IP piracy, hardware Trojans, side-channel-attacks, and reverse engineering [5]. The focus of this paper is on the last subject. Even though the possibility of biochip reverse engineering has been discussed [4], no practical attack or exact countermeasure construction is available in the earlier literature.

Protection of FMFBs and DMFBs is of great importance since they are already being used in critical fields related to personal health. In this paper, we present the first practical reverse engineering attacks that are applicable to both FMFBs and DMFBs. Countermeasures to thwart RE attacks are also proposed. Technical contributions of our work can be summarized as follows.

- BioChipWork demonstrates the first practical layout-level reverse engineering attack on a commercial flow-based microfluidic biochip and successfully extract the component-level netlist. The attack is non-invasive and low cost, making it attractive to malicious parties who want to pirate the design.
- BioChipWork demonstrates the first protocol-level reverse engineering attack using video or actuation sequence analysis. Performance and overhead of the proposed attack are evaluated on various benchmarks.
- We identify the increased attack interface in cyberphysical DMFBs and demonstrate that the information leakage from the imaging sensor can be misused for IP piracy. To mitigate the information leakage from the integrated sensors and the communication channel, we propose camouflaging and obfuscation as countermeasures to obscure the functional design and block direct eavesdropping on the communication channel. Security and overhead of proposed countermeasures are discussed.

This paper is organized as follows: Section II introduces background knowledge about microfluidic biochips. Section III discusses previous works on the security enhancement of biochips. Section IV presents the attack model of our framework. Section V presents the methodology of our

1063-6404/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCD.2017.11

automated reverse engineering framework. Section VI demonstrates simulation and experimental results of proposed attacks. Section VII suggests two defense mechanisms to improve the resiliency of DMFBs against reverse engineering and piracy attacks. Section VIII concludes the paper.

II. Preliminaries

Microfluidic biochip is a miniaturized biomedical platform where the experiment is carried out by controlling the transportation of fluids inside channels or manipulating discrete droplets on-chip. It has been widely used for clinical diagnosis and biomedical analysis, emancipating human from laborious experimental procedures. The first generation of microfluidic biochips is based on the manipulation of continuous flow. Microvalves and microchannels are hardware necessities to actuate and control the liquid flow [6]. The second generation of biochips is based on the manipulation of discrete droplets, providing better scalability and higher throughput. The mechanisms of FMFBs and DMFBs are discussed below.

A. Flow-based microfluidic biochip

Similar to how MOS transistors constitute CMOS integrated circuits, the fundamental building blocks of FMFBs are microfluidic valves (microvalves). Typically, FMFBs are fabricated on an elastomeric material like PolyDiMethylSiloxane (PDMS). Figure 1 shows the working mechanism of a microvalve from Fluidigm [7].

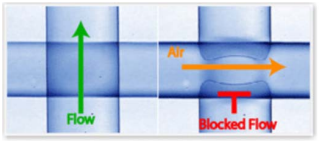


Fig. 1. Control mechanism of a microfluidic valve [7].

When the control channel is in normal condition, the fluid flows freely in the flow channel (as shown by the green arrow). When pressure is applied on the control channel, the elastomer will squeeze the lower layer and block the flow channel (as shown by the red sign). Due to this characteristic, the valve is referred to normally open microvalve. Other components such as mixers, switches and pumps can be constructed using different combinations of valves. The supply chain of FMFBs is shown in Figure II-A, consisted of application synthesis, chip layout, fabrication three main phases. A synthesis methodology of fault-tolerant architectures for FMFBs is presented in [8] which ensures successful execution of bioassay.

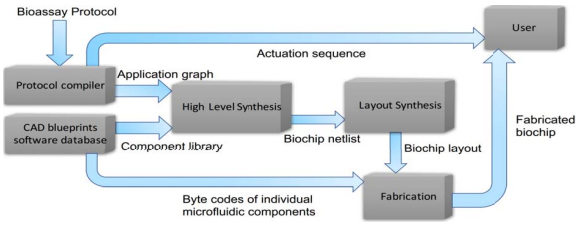


Fig. 2. Supply chain of the FMFB.

B. Droplet-based microfluidic biochip

DMFB is the second generation of microfluidic biochip that utilizes Electrowetting-on-Dielectric (EWOD) phenomena to manipulate discrete droplets on a two-dimensional electrode array. The structure of a typical two plate-DMFB is shown in Figure 3, consisting of glass substrates, dielectric layers, hydrophobic layers, continuous ground electrodes on the top plate, and discrete control electrodes on the bottom plate.

EWOD alters the surface tension of a droplet by applying an 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 grid.

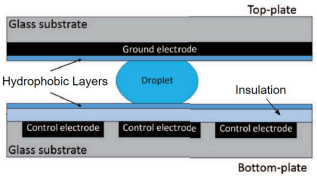


Fig. 3. Schematic diagram of a typical two-plate DMFB.

Compared to FMFBs, DMFBs are reconfigurable since the movement of droplets are determined by the voltage sequence sent to the control pins rather than the predetermined path formed by the permanent etching of substrates. Cyberphysical DMFB monitors the execution of protocols in real time and provides feedback to the control system, allowing dynamic re-synthesis and error recovery [9].

III. Related Work

Recent research has focused on improving the security of DMFBs after their susceptibilities are identified [3], [4]. In contrast, the security problems of FMFBs have not been discussed in prior literature. To the best of our knowledge, BioChipWork is the first to reveal and demonstrate attacks on FMFBs along with countermeasures proposed to mitigate the security concerns. Three relevant works on security of microfluidic biochips are discussed below.

A. Supply Chain Security

In [4], attacks that may happen in the supply chain of DMFBs are identified and categorized into three main classes: trojans, piracy attacks 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 this work.

B. Assay Protection

The authors in [10] 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 assay encryption. Without correct keys applied, malicious attackers will get the wrong assay output. FMUXs obfuscate the assays in the design flow, preventing protocol piracy and chip overbuilding. The number of inserted

FMUXs determines the length of secret keys and therefore defines the security metric of the assay encryption. However, one limitation of the proposed FMUX-based encryption is that all fabricated DMFBs with the same encrypted sequencing graph have the common secret key, which impairs the resiliency of DMFBs to IP piracy attacks.

C. Device Locking

In [11], the intrinsic manufacturing variation of DMFBs is utilized to construct a physical unclonable function (PUF) of the biochip. The absorption induced by electrodes varies uniquely from chip to chip and therefore the volume of resulting droplet which undergoes the same operations varies across chips. The comparison of relative volume can be used to generate PUF response bits. The DMFB is locked by inserting additional finite-state machine (FSM) and the volume of droplets is monitored by a CCD camera. Information leakage through the integrated CCD or on-chip sensors are not considered in this volume-based PUF.

IV. Attack Model

A. Hardware Design Attack Model

To reverse engineer the hardware layout of the FMFB and reconstruct the component-level netlist, we assume that the attacker has access to the physical biochip or its image. The attacker is assumed to know the template of microvalves, the component library and design rules used by the foundry. Furthermore, the attacker checks the valve detection result of the algorithm and manually corrects the errors.

B. Protocol Attack Model

Actuation-Sequence-based Attack. In the protocol RE scenario, our framework assumes that the attacker knows the biomedical specifications of the assays and the pin-mapping scheme deployed in the pertinent DMFB. The communication channel between the control board and the biochip is assumed to be insecure and the attacker can eavesdrop on the channel. The complete actuation sequence of the target protocol combined with the pin-mapping function allows the attacker to deduce the coordinates of activated droplets in each clock cycle from the control signal. The extracted droplet coordinates serve as the basis for operation classification and protocol reconstruction.

Video Analysis-based Attack. In a cyberphysical DMFB, the video-based protocol RE attack assumes that the adversary has access to the video sequence recorded by the integrated CCD camera. An alternative assumption is that the attacker has the physical DMFB and is able to perform the target assay on the biochip. In both cases, the attacker is assumed to know the biomedical specifications of the assay, such as the content and concentration of each input sample. BioChipWork takes the video frames of the experimental implementation as input and sequentially analyze the frames to identify operations in chronological order. The target protocol is reconstructed from the classified operations with their corresponding executing cycles and participant droplet identifiers. The protocol can be optionally visualized as a directed acyclic graph (DAG).

V. Reverse Engineering Attacks

Reverse engineering is a common attack on silicon ICs that aims to recover the desired abstraction from the design. RE of an IC consists of three levels: identifying the underlying technique, reconstructing the gate-level netlist and deducing IC's functionality [12]. Depackaging, delayering and image processing are required to expose the details of chip design and reconstruct the gate-level netlist. In this paper, we show that microfluidic biochips are also vulnerable to RE attacks. BioChipWork presents hardware RE attacks on a commercial FMFB using image analysis and protocol RE attacks on DMFBs using information leaked through the control signal or the integrated CCD sensors in simulation.

A. Reverse Engineering Hardware Design

In the design flow of FMFBs, the designer provides the foundry with the hardware layout in the format of a valve netlist. Functional components such as I/O ports, pumps and switches are built from structural combinations of valves and constitute the schematic architecture of the FMFB. BioChipWork aims to reconstruct the component-level description given an image of the target FMFB. The proposed attack is demonstrated on a commercial FMFB [1] using an image from its website. As opposed to the design piracy procedures described in [4], we prove that depackaging and delayering is not necessary. Image processing alone is enough to reverse engineer the design of FMFBs by leveraging the transparency property of substrate materials used in their fabrication.

The workflow of hardware layout RE has five stages: image pre-processing, frequency analysis, valve identification, component classification, and component connectivity identification. The details of each step are explained below:

Image Pre-processing. In our attack model, we assume the attacker has a normal image of the biochip. The image can be taken using cellphones or cameras instead of expensive microscopes required by RE attacks on silicon ICs. Denoising and non-uniform illumination correction algorithms are applied on the input image to benefit subsequent image processing.

Frequency Analysis. For bioassays that need to be executed under high temperature, heating modules such as incubators are integrated on the biochip. These heating components usually take the shape of periodic, densely-distributed line segments and therefore contribute to the high frequency part in the Fourier domain. BioChipWork exploits this characteristic of heating components and use discrete cosine transformation (DCT) to find the location of the incubator. The image of the target FMFB is first transformed to the frequency domain and thresholded, keeping high frequency only. The intermediate result is then transformed back to the spatial domain, which indicates the position of the incubator.

Valve Identification. Valves are the building blocks of functional 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.

Component Classification. In this stage, the identified valves from the previous step are clustered and labelled according to the component library of the FMFB. Figure 4 shows an example of the component library used in the foundry. The attacker is assumed to have prior knowledge about the library. The structural characteristic is exploited to recognize and label each functional component.

At the beginning of component classification, pairwise distance between all valves is computed. Proper threshold is then chosen and compared with the computed pairwise distance to decide if two valves are connected. Connected valves are considered to belong to the same cluster and each cluster corresponds to a component in the library. 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 5a.

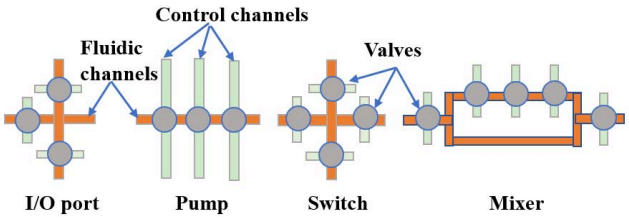
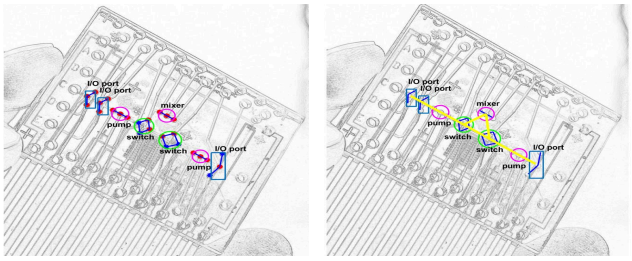


Fig. 4. Component library of an FMFB. Microvalves and microchannels are denoted by circles and rectangles, respectively.

Component Connectivity Identification. After valves and components are identified, the last step is to reconstruct the connectivity between components. We exploit the continuous property of fluids and conclude that any component cannot be independent, which means it has to be connected to another component. Our framework deploys this continuity constraint and find neighbors of each component. The pairwise distance between components are computed and compared with the threshold to determine the existence of connectivity. Figure 5b shows the intermediate output of connectivity reconstruction. The valves are grouped in clusters with their function annotated. The yellow lines denote the connection between components. It can be seen that the recovered component connectivity satisfies the continuity constraint we observe.



(a) Component classification. (b) Connectivity identification.

Fig. 5. Demonstration of component classification and component-level connectivity identification.

B. Reverse Engineering Protocols

The IP of a DMFB is the proprietary protocol mapped to the biochip. The protocol is characterized by scheduled operations combined with the biomedical library from the biocoder [4]. The protocol can be visualized as a sequencing graph $G = (N, E)$, where N is the set of nodes denoting the scheduled operations and E is the set of edges denoting the dependencies between operations [3]. Typical droplet operations are shown in Figure 6. The labels used in the operation classification step of our attack is based on the categories defined in the synthesis tool [13]. BioChipWork provides two alternatives to reverse engineer protocols, assuming the availability of actuation sequence or video frames respectively. Droplet coordinates are extracted and the protocol is reconstructed by analyzing the change of droplet locations in continuous cycles.

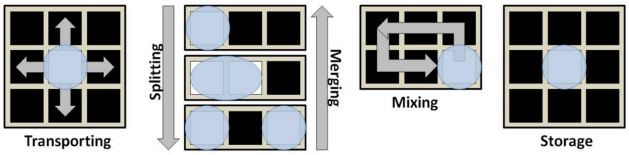


Fig. 6. Typical operational types of DMFBs. Transporting, splitting, merging, mixing and storage are shown here [11].

Actuation-Sequence-Based Protocol RE. The DMFB is controlled by a micro-controller unit (MCU) connected to a PC. The PC runs the CAD tool, synthesizes the specified bioassay and generates the actuation sequence. The actuation sequence is then sent to the MCU and transmitted to the control pins of the DMFB. The actuation sequence is a binary string where bit '1' means connected pins are activated and bit '0' means connected pins are deactivated. Assuming the availability of the actuation sequence and the pin-mapping scheme, the attacker deduces positions of actuated electrodes by analyzing the pattern of the binary string. The behavior of each droplet is identified based on its position in successive cycles. Finally, the protocol is reconstructed by analyzing the behaviors of all droplets during the execution.

The pseudocode of our framework is shown in Algorithm 1. The total number of execution cycles is T, the actuation vector in ith cycle is $\mathbf{s^{(i)}}$ and the actuation matrix is $\mathbf{S} = \left[\mathbf{s^{(1)}}; \mathbf{s^{(2)}}; ...; \mathbf{s^{(T)}}\right]$. The number of present droplets in ith cycle is $N^{(i)}$ and coordinates of droplets in ith cycle are denoted by $\mathbf{I^{(i)}} = (\mathbf{x^{(i)}}, \mathbf{y^{(i)}})$. $\mathbf{I^{(i)}}$ is a $N^{(i)}$-by-2 matrix where each row of it corresponds to the coordinate of a droplet. The target protocol $\mathbf{P}$ is represented by a set of chronological operations $\mathbf{P} = \left\{\mathbf{O^{(i)}}, i = 1, ..., T\right\}$. $\mathbf{O^{(i)}}$ is the collection of all operations that happen in ith cycle. When new droplets enter the biochip, they are labelled with unique identifiers (id). The number of droplets $N^{(i)}$ and the droplet ID list $ID^{(i)} = \{id_1, ..., id_{N^{(i)}}\}$ are updated. Parent droplets in ith cycle may move one grid or keep static during one cycle, producing child droplets in $(i+1)th$ cycle. Therefore, the droplet ID list $ID^{(i+1)}$ is obtained by finding the parent droplet in ith cycle which has one or zero distance with the child droplet in $(i+1)th$ cycle.

Algorithm 1 BioChipWork's workflow to reverse engineer the target protocol.

INPUT: **I/O port position** $\mathbf{In}, \mathbf{Out}$**; execution cycles** T**, actuation matrix** $\mathbf{S}$**; pin-mapping function** f_m**; mix duration threshold** t

OUTPUT: **Protocol set** $\mathbf{P} = \{\mathbf{O^{(1)}}, ..., \mathbf{O^{(T)}}\}$

for $1 \leq i \leq T$ **do**
  $\mathbf{I^{(i)}} \leftarrow f_m(\mathbf{s^{(i)}})$; $N^{(i)} \leftarrow \#rows(\mathbf{I^{(i)}})$
for $1 \leq i < T$ **do**
  $InputID \leftarrow HasInput(\mathbf{I^{(i)}}, \mathbf{In})$
  if $InputID \neq \varnothing$ **then**
    add $InputID$ to $ID^{(i)}$;
    add $('Input', InputID, i)$ to $\mathbf{O^{(i)}}$
  $OutputID \leftarrow HasOutput(\mathbf{I^{(i)}}, \mathbf{Out})$
  if $OutputID \neq \varnothing$ **then**
    deletes $OutputID$ from $ID^{(i)}$
    add $('Output', OutID, i)$ to $\mathbf{O^{(i)}}$
  $\mathbf{Ds} \leftarrow pdist2(\mathbf{I^{(i)}}, \mathbf{I^{(i)}})$; $\mathbf{Dx} \leftarrow pdist2(\mathbf{I^{(i)}}, \mathbf{I^{(i+1)}})$
  if $find(\mathbf{Dx} == 0) \neq \varnothing$ **then**
    $id \leftarrow find(\mathbf{Dx} == 0)$; add $('Store', id, i)$ to $\mathbf{O^{(i)}}$
  if $find(\mathbf{Dx} == 1) \neq \varnothing$ **then**
    $id \leftarrow find(\mathbf{Dx} == 1)$; add $('Mov', id, i)$ to $\mathbf{O^{(i)}}$;
  if $find(\mathbf{Ds} == 1) \neq \varnothing \& N^{(i)} > N^{(i+1)}$ **then**
    add $('Merge', id, i)$ to $\mathbf{O^{(i)}}$
  if $find(\mathbf{Ds} == 1) \neq \varnothing \& N^{(i)} < N^{(i+1)}$ **then**
    add $('Split', id, i)$ to $\mathbf{O^{(i)}}$
  for $1 \leq j \leq N^{(i)}$ **do**
    $id \leftarrow ID^{(i)}(j)$; $route \leftarrow \mathbf{I^{(i:i+t)}}(j, :)$
    if ContinuousMove($route$) == 'True' **then**
      add $('Mix', dropletID, i)$ to $\mathbf{O^{(i)}}$
  $\mathbf{I^{(i+1)}} \leftarrow UpdateIDList(\mathbf{I^{(i)}})$

Based on the operation description used in the simulation tool, BioChipWork 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 $\mathbf{P}$. For each specific operation, the data structure $\mathbf{O}$ consists of three parts: the classification label ($'Input'$, etc.), 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 field.

Video-Based Protocol RE. For a cyber-physical 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.

The video-based protocol RE first identifies the positions of existing droplets by deploying template matching algorithm on each frame. Subsequent procedures to reconstruct the protocol $\mathbf{P}$ are the same as the steps described by Algorithm 1. Since droplets are located by template matching, the pin-mapping scheme used in the synthesis phase does not affect our attack. Therefore, compared to the actuation sequence-based approach, video-based method removes the need of prior knowledge about the pin-mapping function f_m.

The execution of a PCR bioassay is visualized in Figure 7. Connected electrodes are indicated with the same number. BioChipWork proves that pin-count optimized DMFBs are also vulnerable to 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 pin-count optimized DMFB cannot mitigate the protocol piracy concern in this case.

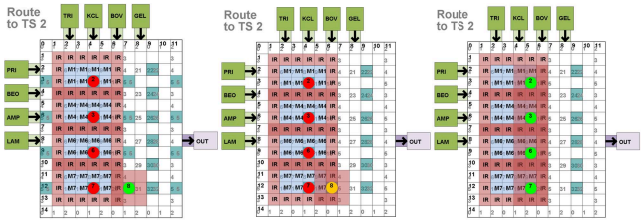


Fig. 7. Three continuous time frames of a PCR experiment visualization. The bioassay is mapped to a field programmable pin-constrained DMFB.

VI. Experimental Results

In this section, we present the results of reverse engineering attacks on hardware-design-level and protocol-level. For FMFBs, our reconstructed component-level layout is consistent with the groundtruth architecture used in the foundry. For DMFBs, we evaluate our protocol RE attacks on various benchmarks. More specifically, we define and compute the reconstruction accuracy of the attack and report the time overhead. The performance evaluation demonstrates the scalability and feasibility of our framework.

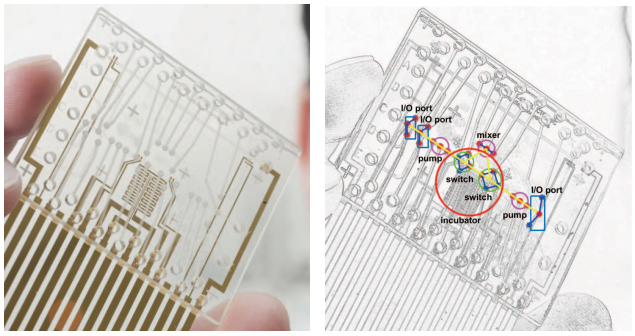
A. Hardware Design Reverse Engineering

The image of a commercial FMFB from Microfluidic Innovations LLC. [1] is shown in Figure 8a and used as the input to the image analysis algorithm. The reverse engineered layout is shown in Figure 8b. The incubator is denoted by a large, red circle. The positions of microvalves and connections are indicated by solid dots and yellow lines respectively. Each identified component is marked by a rectangle or a small circle with its functionality annotated. The valve-level connectivity inside each component is indicated by blue lines.

Comparing Figure 8a and Figure 8b, it is clear that our algorithm recovers all present valves, components and connections correctly, which constitute the correct schematic abstraction of the FMFB layout. The proposed attack can be conducted within a few seconds without the need of depackaging or delayering. BioChipWork alerts the designer to the susceptibility of the hardware design and the necessity of countermeasures.

B. Protocol Reverse Engineering

Due to the limited resources and access to commercial DMFBs, we demonstrate our protocol RE attack in simulation using the open source tool [13] instead of on the physical



(a) Image of a commercial microfluidic biochip used in our attack [1].

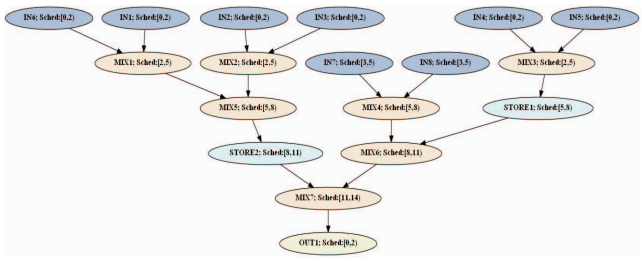
(b) Identified components and connections with corresponding functionalities annotated.

Fig. 8. Practical hardware design reverse engineering result of a FMFB.

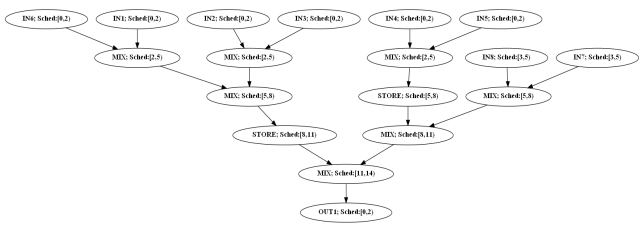
DMFBs. However, the proposed framework is general and also applicable to real DMFBs with some straightforward modification. Since the actuation sequence and recorded video frames obtained from physical world have the same representation as the output of the synthesis tool. The performance of the protocol RE attack is assessed by the portion of correctly characterized operations **O**. The protocol can also be visualized as a DAG, while it is worth noticing that one protocol can have multiple equivalent DAG descriptions.

Actuation-Sequence-Based Protocol RE.

As a case study, we choose a PCR bioassay as our target. The original and reconstructed sequencing graph are shown in Figure 9a and Figure 9b respectively. Each node denotes an operation and is annotated with corresponding properties, such as executing time, operation label. Even though the visualization is not the same, these two DAGs are characterized by the same nodes and edges, suggesting that the attack succeed.



(a) Original DAG of a PCR assay.



(b) Reconstructed DAG of the same PCR assay based on actuation sequence.

Fig. 9. Demonstration of actuation sequence-based protocol RE of a PCR assay. For simplicity, the DMFB is selected to be individually addressed.

To prove the generality and scalability of our framework, the attack is further evaluated on other bioassays. The performance and overhead of the attack is summarized in Table I. The specification of the DMFB platform is given by the chip dimension ($height \times width$). The protocol running time and the number of involved operations is given by execution cycles and nodes number respectively. The actuation frequency of the DMFB is set to 100Hz, therefore the actual execution time in second can be computed as executed cycles divided by 100. The reconstruction accuracy is defined as the percentage of correctly identified operations. Our algorithm is implemented in Matlab 2017 a on a 64bit PC with Inter Core i7, 3.5GHz, 32G RAM. The reconstruction time is reported as the total running time of the algorithm.

As can be seen from Table. I, the time overhead is dependent on both the number of nodes and the execution cycles of the protocol. For large benchmarks with complex operations and long execution time, BioChipWork is still able to finish the attack within a reasonable time. 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, meanning that BioChipWork cannot distinguish 'Detect' from 'Store'.

Our framework can characterize all operations correctly except for 'Detect'. Hence the reconstruction accuracy is dominated by the percentage of 'Detect' operations in the groundtruth protocol as reported in Table I. For example, the assay InVitro has 16 'Detect' operations and 80 operations in total, our attack recognizes all operations except for 'Detect', therefore the reconstruct accuracy is $(80 - 16)/80 = 80\%$.

Video-Based Protocol RE. We demonstrate the video-based protocol RE on a pin-count optimized DMFB. An example of video-based protocol RE is shown in Figure. 10b, where the protocol is presented as a DAG. Compared to Figure. 10a, the reconstructed DAG has exactly the same structure as the original DAG, indicating the feasibility and high accuracy of video-based protocol RE.

VII. Potential Countermeasures

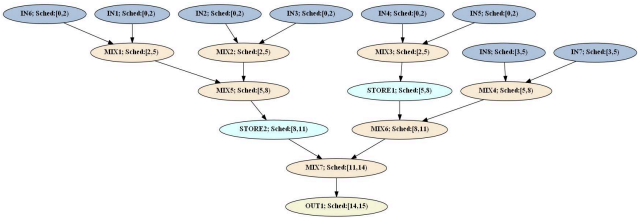
As discussed in the section I, previous papers [3], [4] has already pointed out that DMFBs are susceptible to various attacks and suggested potential defenses. On the contrary, this paper demonstrates the first practical reverse engineering attacks on FMFBs and DMFBs that compromises the IP of designers and protocol owners. To mitigate the security concern , we suggest camouflaging and obfuscation as countermeasures to improve the resiliency of DMFB against RE attacks.

A. Camouflaging

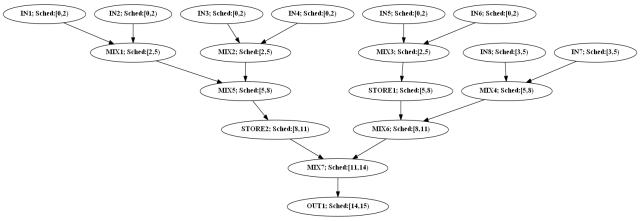
Camouflaging is a common defense mechanism in silicon ICs that aims to hinder image-based reverse engineering of the gate-level netlist. To hind the design of the FMFB, dummy valves and dummy channels can be inserted into the original layout. In this circumstance, the component library obtained

Protocol	*Chip Dimension*	*Execution Cycles*	*Nodes Number*	*Reconstruction Accuracy*	*Reconstruction Time(s)*
Two Dilution	8 * 8	554	8	100%	1.099
PCR	15*12	1458	16	100%	1.392
Protein Mix	8*121	8141	58	93.1%	34.758
InVitro	19*15	3705	80	80%	12.233

TABLE I
PROTOCOL-LEVEL REVERSE ENGINEERING. BENCHMARKS ARE EVALUATED IN THE OPEN SOURCE SYNTHESIS TOOL [13].



(a) Original DAG of another PCR assay.



(b) Reverse engineered DAG using video analysis.

Fig. 10. Video-based protocol reverse engineering of the PCR assay running to a pin-count optimized DMFB.

by the attacker is useless and even misleads him to an incorrect component-level abstraction.

Figure 11 demonstrates how to camouflage a FMFB by inserting dummy valves and channels. The original structure of an I/O port component is given in (a). Adding an additional valve results in the structure in (b), where the camouflaged I/O port has the same appearance as a switch in the component library. An alternative option is to camouflage the I/O port as a mixer by adding two valves as shown in (c). Proper pressure signals are required to be applied on dummy valves for ensuring the correct functioning 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.

Inserting dummy valves and dummy channels during fabrication increases the manufacturing cost, control complexity as well as communication overhead. However, in recent microfluidic Very Large Scale Integration (mVLSI) fabrication process, the size of valves is very small ($100 \times 100\ um^2$) and a single FMFB can accommodate thousands of valves. This suggests that adding proper number of dummy valves and channels will not induce large cost or overhead. A metric to evaluate the effectiveness of a camouflaging approach can be defined as the Hamming distance between the original component netlist and the reconstructed netlist from the camouflaged layout [12].

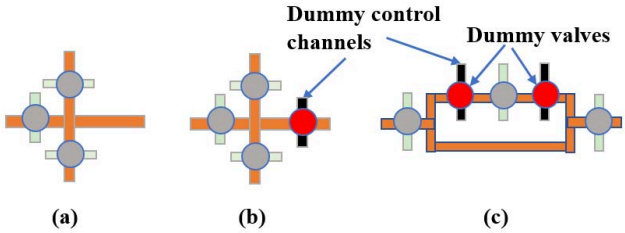


Fig. 11. Camouflage the layout of FMFB's component. (a) is the original structure of I/O port; (b),(c) shows the methods to camouflage the I/O port as a switch or a mixer, respectively. Dummy valves are indicated with red circles and dummy control channels are indicated with dark lines. The component classification is based on the component library shown in Figure. 4.

The security offered by camouflaging is determined by the number and the positions of the inserted dummy valves. The trade-off between the security level and the additional cost can be assessed by the manufacturer in the design phase.

B. Obfuscation

In standard IC, obfuscation can be implemented by obscuring the functionality or the the finite state machine (FSM) [12]. FSM obfuscation on DMFBs has already been demonstrated in [11]. The author uses the combination of PUF response bits and license issued by the foundry as the key to unlock the DMFB. However, the PUF-based scheme is not secure against authorized-but-curious users. Our framework shows that information about the proprietary protocol may be leaked through actuation sequence or data from the CCD sensor. In the following sections, we propose two advanced obfuscation methods to mitigate the information leakage.

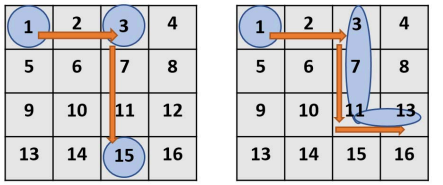
Actuation Sequence Obfuscation.

Actuation sequence contains information about the assay and may be misused for protocol piracy. This attack is less of a threat to field-programmable DMFBs which can be reconfigured after manufacturing. To alleviate the piracy concern, control signals needs to be obfuscated before the transmission in the communication channel. Actuation sequence can be encrypted using license issued by the foundry or secret keys obtained from PUFs [10], [11].

Assuming the assay lasts T clock cycles and the length of actuation sequence in each cycle is L. The L bit symmetric key is denoted by $\mathbf{e_k}$, the original, encrypted and decrypted actuation sequence are denoted by $\mathbf{S_o}, \mathbf{S_e}, \mathbf{S_d}$ respectively. $\mathbf{S}$ is a T-by-L matrix with the element $s_{i,j}$ indicating the actuation status of jth electrode in clock cycle i. The protocol designer encrypts the actuation sequence using XOR operation $\mathbf{S_e} =$

$\mathbf{S_o} \oplus \mathbf{e_k}$. The control signal is decrypted using the same secret key $\mathbf{S_d} = \mathbf{S_e} \oplus \mathbf{e_k}$ before sending them to control pins.

Integrating logic circuits on DMFB for pin-count reduction has been discussed in [14]. Inspired by the work, we propose to integrate XOR gates on DMFBs for on-chip decryption of the actuation sequence. Since the manufacturing process of DMFB is compatible with CMOS techniques and the size of a XOR gate is much smaller than an electrode cell, the area overhead of adding XOR gates is negligible. The computational complexity of both encryption and decryption is $O(TL)$. This means the complexity increases linearly with total number of clock cycles T and the number of independent control pins L, making the obfuscation scheme scalable. The security of encryption depends on the length of the key e_k. Therefore, individually addressed DMFBs have stronger security with the cost of higher decryption overhead.



(a) Original path (b) Reverse engineered path

Fig. 12. Reverse engineering results from obfuscated actuation sequence. The attacker will extract incorrect operations from the encrypted sequence.

Figure 12 demonstrates the effect of obfuscating the sequence. Figure 12a shows the groundtruth trajectory of a droplet running on a 4-by-4 individually addressed DMFB. In this case, $T = 6, L = 16$. Assuming the Lbit-length secret key is chosen as $\mathbf{e_k} = 0010011001110110$, each row in $\mathbf{S}$ is XORed with $\mathbf{e_k}$ before being transmitted to control pins. By eavesdropping on the communication channel and analyzing the encrypted control signal, the attacker will reconstruct the incorrect trajectory as shown in Figure 12b. The comparison proves that obfuscating actuation sequence can prevent attackers from extracting useful information by directly observing of signals in the communication channel.

Sensor Feedback Obfuscation. Data collected by the integrated cameras or other sensors may leak information about the executing assay. Even if the DMFB is locked by inserting additional FSM as described in [11], manufacturers and authorized end-users can still reverse engineer the protocol by leveraging sensor data. BioChipWork demonstrates that it is feasible to pirate the protocol from the droplet positions during the complete execution. The coordinates of present droplets can be determined either from video frames or on-chip capacitive sensors. These additional hardware components in cyberphysical DMFBs increase the attack interface and allow cyber attacks. To the best of our knowledge, our framework is the first to exploit the vulnerabilities in cyberphysical biochips and demonstrate attack simulation results. One potential solution to protect sensor data is to encrypt it with keys extracted from PUF [11] or FMUX control inputs [10].

VIII. Conclusion

We develop BioChipWork, the first automatic and scalable framework to reverse engineer the hardware layout and biomedical protocols of microfluidic biochips. Our image processing algorithm takes advantage of the intrinsic transparent properties of materials used in the fabrication process and extracts the component-level netlist of the pertinent biochip without invasive procedures of depackaging and delayering. The attack is proven successful on a commercial FMFB. We also demonstrate simulation results of protocol reverse engineering based on actuation sequence analysis or video analysis. Accuracy and overhead of the attack are evaluated on various benchmarks. To the best of our knowledge, BioChipWork is the first to reveal the cyber vulnerabilities in cyberphysical DMFBs and exploit information leakage from the communication channel or CCD sensors to pirate the IP. To prevent reverse engineering and IP piracy attacks on biochips, we propose camouflaging and obfuscation as two countermeasures. Security metric and overhead of these two defense are discussed.

Acknowledgment

The authors would like to thank Siam Hussain and Mohammad Ghasemzadeh for their valuable comments on the paper.

References

[1] *Microfluidic Innovations LLC.*, http://cfpub.epa.gov/npdes/.
[2] *illumina*, https://www.illumina.com/.
[3] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security implications of cyberphysical digital microfluidic biochips." in *ICCD*. IEEE Computer Society, 2015, pp. 483–486.
[4] S. S. Ali, M. Ibrahim, J. Rajendran, O. Sinanoglu, and K. Chakrabarty, "Supply-chain security of digital microfluidic biochips." *IEEE Computer*, vol. 49, no. 8, pp. 36–43, 2016.
[5] R. Torrance and D. James, "The state-of-the-art in ic reverse engineering," in *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer, 2009, pp. 363–381.
[6] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002.
[7] *The Fluidigm Corporation*, http://www.fluidigm.com.
[8] M. C. Eskesen, P. Pop, and S. Potluri, "Architecture synthesis for cost-constrained fault-tolerant flow-based biochips," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*. IEEE, 2016, pp. 618–623.
[9] M. Ibrahim and K. Chakrabarty, "Efficient error recovery in cyberphysical digital-microfluidic biochips." *IEEE Trans. Multi-Scale Computing Systems*, vol. 1, no. 1, pp. 46–58, 2015.
[10] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Microfluidic encryption of on-chip biochemical assays," in *Biomedical Circuits and Systems Conference (BioCAS), 2016 IEEE*. IEEE, 2016, pp. 152–155.
[11] C.-W. Hsieh, Z. Li, and T.-Y. Ho, "Piracy prevention of digital microfluidic biochips." in *ASP-DAC*. IEEE, 2017, pp. 512–517.
[12] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics." *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014.
[13] D. Grissom, K. O'Neal, B. Preciado, H. Patel, R. Doherty, N. Liao, and P. Brisk, "A digital microfluidic biochip synthesis framework." in *VLSI-SoC*. IEEE, 2012, pp. 177–182.
[14] T. A. Dinh, S. Yamashita, and T.-Y. Ho, "A logic integrated optimal pin-count design for digital microfluidic biochips," in *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 75.