Desieve the Attacker: Thwarting IP Theft in Sieve-Valve-based Biochips

Mohammed Shayan†, Sukanta Bhattacharjee†, Yong-Ak Song∗†, Krishnendu Chakrabarty‡, and Ramesh Karri∗

∗New York University, †New York University Abu Dhabi, ‡Duke University
Email: mos283@nyu.edu, sb6538@nyu.edu, rafael.song@nyu.edu, krish@ee.duke.edu, rkarri@nyu.edu

Abstract—Researchers develop bioassays following rigorous experimentation in the lab that involves considerable fiscal and highly-skilled-person-hour investment. Previous work shows that a bioassay implementation can be reverse engineered by using images or video and control signals of the biochip. Hence, techniques must be devised to protect the intellectual property (IP) rights of the bioassay developer. This study is the first step in this direction and it makes the following contributions: (1) it introduces use of a sieve-valve as a security primitive to obfuscate bioassay implementations; (2) it shows how sieve-valves can be used to obscure biochip building blocks such as multiplexers and mixers; (3) it presents design rules and security metrics to design and measure obfuscated biochips. We assess the cost-security trade-offs associated with this solution and demonstrate practical sieve-valve based obfuscation on real-life biochips.

I. INTRODUCTION
A biochip platform integrates complex laboratory operations into a small chip of few square centimeters in size. It has revolutionized biochemical applications such as point-of-care diagnostics [1], DNA purification [2], and biomedical research [3]. The microfluidics market is valued at $8.28 Billion in 2017 and it is expected to grow at a compound annual growth rate of 22.6% to reach $27.91 Billion by 2023 [4]. Due to rapid commercialization and deployment, intellectual property (IP) piracy has become financially rewarding [5]. Therefore, protecting bioassay IPs is of paramount importance to its developers.

Pharmaceutical companies invest large sums of money and man-hours in a slow and expensive drug development process laced with tough regulations. This process is prone to stealing of sensitive research data [6]. In 2016, two scientists at a leading pharmaceutical company were indicted for colluding with a competitor to steal promising drug research secrets [7]. For rapid and low-cost drug development, pharmaceutical companies are using various types of microfluidic biochips that minimize the assay time and reagent requirement [8].

Continuous flow-based microfluidic biochips (CFMBs) have evolved rapidly in the last decades [3], [9]. The CFMBs allow automated control of fluid flow in a network of micro-channels by suitable actuation of pressure driven micro-valves [9]. Previous work has shown that a bioassay implementation on a CFMB can be reverse engineered using biochip images and actuation sequence [10]. Therefore, there is a need to devise methods that protect the bioassay IP implementation on biochips, which in turn is critical for the successful adaptation of biochips. This work is a first step towards protecting bioassay IP based on hardware primitives on a CFMB. Our contributions are summarized as follows:

- We develop design rules, security metrics, and cost trade-offs for a sieve-valve-based obfuscated biochip.
- We demonstrate the practicality of the obfuscation method by applying it to a real-life biochip benchmark.

The rest of the paper is organized as follows. Section II provides a motivational example to describe the IP piracy threat. Section III provides the relevant background and Section IV describes the use of sieve-valves to achieve bioassay obfuscation. Section V develops the metrics and design rules associated with the sieve-valve based obfuscation. Section VI provides experimental results of obfuscation applied to real-life biochips and Section VII concludes the paper.

II. MOTIVATION
We demonstrate IP piracy through a bioassay implementation on a CFMB (Fig. 1). The platform consists of a multiplexer that selects from two input reagents and uses a rotary mixer to mix them in the desired ratio [11]. The mixing time can also be determined from the actuations. It can be inferred from the biochip snapshots that the bioassay mixes two input fluids in a 3:1 ratio [10]. The corresponding sequencing graph (IP) is shown in Fig. 1(e). The resulting fluid contains a mixture of and the peristaltic pump is activated (Fig. 1(e)). The resulting fluid contains a mixture of and 1:1 ratio (Fig. 1(c)). Next, the lower half of the mixer is filled with (Fig. 1(b)). The valves 6, 7, 8 are activated in a sequence to form a peristaltic pump that circulates the fluid in the rotary mixer, producing a mixture of and the peristaltic pump is activated (Fig. 1(e)). The resulting fluid contains and 3:1 ratio (Fig. 1(f)).

Fig. 1 shows the straightforward one-to-one mapping between the actuation sequence, biochip snapshots, and fluidic operations. It can be inferred from the biochip snapshots that the bioassay mixes two input fluids in a 3:1 ratio [10]. The corresponding sequencing graph (IP) is shown in Fig. 1(e). The mixing time can also be determined from the actuations. This example demonstrates the ease with which the bioassay description and its parameters can be reverse engineered [10]. To thwart the RE of bioassays, we need to obfuscate the one-to-one mapping between the actuation sequence, biochip snapshots, and fluidic operations.

III. BACKGROUND
In this section, we present the background on CMFBs and recent work on bioassay IP protection.

A. Continuous-Flow-Microfluidic Biochips
CFMB consists of two layers of permanently etched micro-channels called the flow and the control layer, as shown in Fig. 2(a). At the intersection of the two layers, a “valve” is formed. An external pressure source can control this valve.
In a normal valve, the flow channel is semi-circular shaped. When the valve is pressurized, the flexible membrane of the control layer deflects deep into the flow layer blocking the fluid flow (Fig. 2(b)). By opening/closing of the valves, complex fluid handling operations such as mixing, incubation, transportation, and storage can be performed [9]. Advancement in multi-layer soft lithography techniques enables thousands of valves to be integrated into a tiny chip [9].

In a normal valve, the flow channel is semi-circular shaped. When the valve is pressurized, it seals the flow channel (Fig. 2(b)). However, if the flow channel is rectangular, the pressurized valve membrane partially closes the flow channel, as shown in Fig. 2(d). This is the sieve valve (Fig. 2(c)) [12]. These are used in CFMBs to trap cells. Closing the sieve valve blocks the cells but allows the fluid to pass through [13].
and the sieve-valve locations a secret. The developer uses a CAD tool on a trusted offline computer to synthesize the obfuscated actuation sequence. The obfuscated sequence is loaded in the biochip controllers that are used to conduct the high-valued-experiments, as shown in Fig. 3. Minor software updates are handled in the biochip controller and major updates are performed in the trusted offline computer.

Consider the channel between port $i$ and $j$, as shown in Fig. 4(a). Let the channel $i \rightarrow j$ be open if the valve 1 is actuated, else it is closed. Such a valve is a normal valve. On the other hand, if the valve is a sieve, then the channel $i \rightarrow j$ is always open, regardless of the actuation state of valve 1. To capture the differences between a normal and a sieve valve, consider the Boolean variables defined in Table I. Using these variables, we describe the channel in Fig. 4(a) as:

$$c_{ij} = \gamma_i + g_1 \cdot v_1 = \gamma_i + v_1$$ (1)

As per the attack model, $g_1$ is secret, $v_1$ is known from the actuation sequence. Equation (1) captures the obfuscation introduced in the fluid channel characteristics due to the unknown valve type. Without the knowledge of $g_1$, an attacker does not know the channel status. The sieve valve reduces the flow-rate in the channel. However, this can be neutralized by either increasing the pressure or allowing more time for the fluid flow. For the purposes of our analysis, we ignore the change in flow-rate.

Consider an increase in the number of valves on the channel, as shown in Fig. 4(b). The characteristic of the channel is given by following Boolean equation,

$$c_{ij}^2 = (\gamma_i \cdot v_1 + v_1) \cdot (\gamma_i v_1 + v_1 b)$$

$$= \gamma_i a \cdot \gamma_i b + \gamma_i a \cdot v_1 b + \gamma_i b \cdot v_1 a + v_1 a \cdot v_1 b$$ (2)

If there are $n$ such valves on a channel $i \rightarrow j$, the characteristic of the channel can be captured as

$$c_{ij}^n = \gamma_i + v_1$$ (3)

Comparing Equation (2) and Equation (3), increasing the number of valves increases the channel obfuscation due to the corresponding increase in the number of unknown parameters ($g_1$). Using this primitive, we describe two types of biochip obfuscation: behavioral and structural.

### Example 1
In Fig. 5(a), let $\{1b, 2b, 3b, 10b\}$ be sieve valves and the rest be normal valves, i.e., $g_{1b}, g_{2b}, g_{3b}, g_{10b} = 0000$, then the actuation set $v_{1a}, v_{1b}, v_{2a}, \cdot \cdot \cdot , v_{10a}, v_{10b}, v_{11} = 011010101111110$ pushes $A$ into the mixer (ref. Fig. 5(a)). On the other hand, if $\{1a, 2a, 3b, 10b\}$ are sieve valves and rests are normal valves, then the same actuation set will push $A$ into the mixer. Without knowing the valve type (sieve or normal), an attacker cannot determine the correct fluidic path from the Equation (3), the channel state (open/close) depends on the valve type ($g_1$), which is unknown to the attacker. The following example illustrates obfuscation on the fluidic path.

### Example 2
In Fig. 5(a), consider the valve types as in Example 1, i.e., all $g_1 = 1$ except $g_{1b}, g_{2b}, g_{3b}, g_{10b} = 0000$. Let the valve actuation be $v_{1a}, v_{1b}, v_{2a}, \cdot \cdot \cdot , v_{10a}, v_{10b}, v_{11} = 011010101111110$. This opens the mixer inlet/outlet to push out the mixer content which denotes the end of the previous mixing step. If the actuation is followed by a peristaltic...
pumping operation, then it denotes the start of a new mixing. On the other hand, if \( \{3a, 10a\} \) are sieve valves and \( \{3b, 10b\} \) are normal valves, then the given actuation set does not open the inlet/outlet of the mixer. Hence, the previous mixing step has not ended and a new mixing step has not started. This leads to obfuscation in the deduction of the mixing time and the number of mixing steps, as shown by the dotted nodes in the sequencing graph in Fig. 5(b).

D. Structural Obfuscation

The behavioral obfuscation does not change the structure of the biochip but inserts extra valves on the existing channels. Furthermore, the structure of the biochip can be obfuscated by inserting dummy channels, multiplexers and/or mixers. This is structural obfuscation. A channel can be mimicked by a dummy multiplexer with a sieve valve on the original inlet - so that it is always open and a normal valve on a dummy inlet - which is kept closed. Without the knowledge of sieve valve, the attacker cannot know which inlet is selected when both the valves are closed. Alternately, the channel can be mimicked by a dummy mixer with sieve valves forming an always open channel in the ring mixer. The valves of this module are actuated like a mixing module to mislead the attacker. To resolve this ambiguity, an attacker has to do trial-and-error by replacing each mixing operation in the actuation with a transportation operation.

Example 3. In Fig. 6, 12b, 13b are sieve valves and 12a, 13a are normal valves. For actuation set \( v_{12a}, v_{12b}, v_{13a}, v_{13b} = 0000 \), paths \( R_1^{12} \rightarrow R_1 \) and \( R_2^{12} \rightarrow R_2 \) are open. On the other hand, if 12b, 13b are normal valves and 12a, 13a are sieve valves, then for the same actuation set \( R_1^{12} \rightarrow R_1 \) and \( R_2^{12} \rightarrow R_2 \) are open. This leads to obfuscation of the fluid selected. Furthermore, a dummy mixer with \( \{15, 20, 21\} \) as sieve valves is added to path \( E \rightarrow O \). The valves of this mixer can be actuated to mimic a normal mixer, whereas in reality, it is a \( E \rightarrow O \) channel controlled by valve 10a.

V. Design for Obfuscation

We define the security metrics that capture the security-cost trade-offs and design-for-obfuscation rules.

A. Security Metrics

To RE the bioassay, the attacker has to interpret the actuation sequences that are ambiguous due to unknown valve type \( g_v \). Such actuations are referred to as ambiguous actuations. The attacker can build a biochip prototype from the snapshots without correct valve types. By trial-and-error, the attacker can replace the ambiguous actuations until the results of the bioassay on the biochip prototype become identical to the known sensor readings. The maximum number of experiments required to resolve this ambiguity is denoted as resolution effort \( E \). The biochip designer obfuscates the biochip and its actuation sequence to make RE hard-enough to deter an attacker. The design overhead for obfuscation is defined in terms of extra valves, which in turn may lead to extra pins in the biochip and extra memory for storing the corresponding actuation signals. To maximize the resolution effort, we propose the following design rules.

B. Design-for-Obfuscation Rules

In a crude attack, the attacker will try all combinations of \( g_v \). However, a smart attacker will leverage functional properties to prune the search space. To achieve a robust obfuscated design, we frame four design rules.

1) Channel: To push a fluid in a CFMB, an input-output channel needs to be opened. The attacker tries to identify which input-output path is opened in a given cycle. If there exists an input-output path without any unactuated valve, then the actuation is unambiguous to the attacker. Else, the attacker has to guess if any of the unactuated valves on the input to output paths is a sieve valve. This leads to the first design rule.

Rule #1: In an ambiguous actuation, every input to output channel path must have at least one closed sieve valve.

Consider a channel that has \( h_{chl} \) unactuated valves in an ambiguous actuation cycle. Without knowing the valve type \( g_v \), i.e., sieve or normal, the de-obfuscation effort \( E_{chl} \) involves trials that map each closed valve to two possibilities - closed and open. Hence, \( E_{chl} \leq 2^h_{chl} \). The effort increases with the number of distinct input-output paths with closed valves in a cycle.

2) Multiplexer: An attacker can use the following properties of a multiplexer to resolve the obfuscation. P1: At most one path of the multiplexer can be open at any time. P2: It is likely that each inlet fluid is selected at least once in a bioassay. An attacker can collect all the unique actuations applied to the multiplexer and along with the properties P1 and P2 the attacker can de-obfuscate the multiplexer actuations as discussed in the following example.

Example 4. Consider a 3-inlet multiplexer with two valves \( +a \) and \( +b \) on each inlet. For each inlet, the set of actuations \( v_{a1}, v_{a2} = \{11, 00\} \) is unambiguous and \( v_{b1}, v_{b2} = \{10, 01\} \) is ambiguous. Between any pair of inlets, there are four possible combinations of these ambiguous actuations. In Fig. 7, 3-out-of-4 combinations are used for actuating the valves \( v_{1a}, v_{1b}, v_{2a}, v_{2b} \). The unused ambiguous actuation combination on the inlet \( In1 \) and \( In2 \) is \( v_{1a}, v_{1b}, v_{2a}, v_{2b} = 1010 \). The attacker can decipher that this actuation opens both inlets \( In1 \) and \( In2 \) and hence is not used due to property P1. Alternately, an attacker can guess that the least used actuation on an inlet is used to open the respective inlet. In Fig. 7, actuation...
The number of ambiguous mixer actuations is dependent on the number of valves on the mixer inlet and outlet, provided rule #4 is satisfied. However, to minimize the cost we use two valves on each inlet. Therefore, the RE effort for a mixer is $E_{mix} = 2^{s_{mix}} = 2^4$.

4) Dummy Structures: The same rules apply to dummy structures such as multiplexers and mixers (Fig. 6). To resolve the ambiguity about $n_{dum}$ dummy structures, ($E_{dum} = 2^n_{dum}$) trial experiments must be performed. However, the cost of introducing dummy structures include not only extra valves but also extra channels and extra input/output ports.

The number of ambiguous mixer actuations is dependent on the number of valves on each inlet. However, this increases the cost. To avoid cost escalation, we use two valves per inlet with design rules #2 and #3.

Rule #2: Apply ambiguous actuations to no more than two inlets at a time. One inlet is the fluid being pushed and one from the other $m-1$ inlets of the multiplexer.

Rule #3: Apply unambiguous actuations when no fluid is pushed through the multiplexer.

Rule #4: The gap between an ambiguous and other mix operations must be more than the minimum mix time, $t_{min}$.

The number of ambiguous mixer actuations is dependent on the number of valves on each inlet. However, this increases the cost. To avoid cost escalation, we use two valves per inlet with design rules #2 and #3.

The attacker can decipher with high probability that these ambiguous actuations on the mixer inlet (outlet) is two. This implies that the maximum number of ambiguous actuations applied to the multiplexer and mixer are $s_{mix} = \left(\frac{m}{2}\right)$ and $s_{mix} = 4$, respectively. An attacker’s effort in resolving the behavioral obfuscation is $E_{behav} = E_{mix} \cdot E_{struct} = 2\left(\frac{m}{2}\right) \cdot 2^4 = 2^{14}$.

The four ring mixers (A-D) are connected to four fluid inlets that are used to wash the contents of the respective mixer. The inlet channel is replaced by a dummy multiplexer to select between the original wash fluid and a wash fluid corresponding to other mixers, as shown in Fig. 9. In eight more valves and four more channels. The effort to resolve the structural obfuscation of $n_{dum} = 4$ mixers is $E_{struct} = 2^{n_{dum}} = 2^4$. The obfuscated sequencing graph is shown in Fig. 10(a). The effort to resolve the behavioral + structural obfuscation is $E = E_{behav} \cdot E_{struct} = 2^{14} \cdot 2^4 = 2^{18}$. Each ChIP trial takes 3.5 h. The time for all trials is over a thousand years. Also, each trial consumes reagents, samples, and biochips.
TABLE II: Obfuscation of real-life biochips.

<table>
<thead>
<tr>
<th>Biochip</th>
<th># Max (m:1)</th>
<th># Ring mixers</th>
<th># Other channels</th>
<th># Valves</th>
<th>Behavioral obfuscation</th>
<th>Structural obfuscation</th>
<th>Total effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChIP [13]</td>
<td>1 (5:1)</td>
<td>4</td>
<td>1</td>
<td>50</td>
<td>2(^{14})</td>
<td>4</td>
<td>2(^{14})</td>
</tr>
<tr>
<td>Kinase act [18]</td>
<td>2 (3:1)</td>
<td>2</td>
<td>2</td>
<td>44</td>
<td>2(^{2})</td>
<td>3</td>
<td>2(^{2})</td>
</tr>
<tr>
<td>mRNA iso. [19]</td>
<td>4 (2:1)</td>
<td>4</td>
<td>1</td>
<td>56</td>
<td>2(^{16})</td>
<td>4</td>
<td>2(^{16})</td>
</tr>
<tr>
<td>Nucleic-Acid proc. [2]</td>
<td>3 (5:1)</td>
<td>3</td>
<td>3</td>
<td>54</td>
<td>2(^{15})</td>
<td>3</td>
<td>2(^{15})</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

Microfluidic platforms have immense potential in paving the way for rapid and low-cost biochemical analysis. However, the cyberphysical system that enables biochip automation is susceptible to IP theft. This is a major hurdle in the large-scale adaptation of microfluidic technologies in industries that are prone to stealing of sensitive research data. Our work addresses this pressing problem with a practical obfuscation methodology that can be easily integrated with the current biochip design flow. We developed sieve-valve-based obfuscation design rules and showcased their application to the real-life biochips. The results show that the de-obfuscation effort is daunting enough to act as a deterrent to an attacker.

The strength of our proposal can be demonstrated in comparison with two IP protection techniques. First, firmware encryption has been used to protect firmware IPs. However, this doesn’t apply to the biochips because the biochip actuators are electrical signals applied to either the valves or to the pneumatic actuators. Even if the actuation sequence is encrypted, it has to be decrypted before it is applied to the biochip control ports. Further, the actuations can be extracted by image and video-based RE. Proposed obfuscation complements encryption to thwart RE of the electrical signals. Second, logic locking is used to prevent IP piracy in VLSI designs. The number of trials needed to de-obfuscate a logic-locked VLSI design is of the order of 2\(^{15}\) [20]. These trials can be done on high-speed computers. On the other hand, the bioassay trials take several hours to complete. Also, unlike VLSI, the bioassay recovery trials require perishable reagents and biochips. The cost and time spent on these trials go against an attacker’s economic objective of stealing a bioassay IP.

ACKNOWLEDGMENT

This research is supported in part by the Army Research Office under grant number W911NF-17-1-0320, NSF Award numbers CNS-1833622 and CNS-1833624, NYU Center for Cyber Security (CCS), and CCS-AD.

REFERENCES