12-1-2019

Bio-chemical assay locking to thwart bio-IP theft

Sukanta Bhattacharjee
NYU Abu Dhabi

Jack Tang
NYU Tandon School of Engineering

Sudip Poddar
Indian Statistical Institute, Kolkata

Mohamed Ibrahim
Intel Corporation

Ramesh Karri
NYU Tandon School of Engineering

See next page for additional authors

Bio-chemical Assay Locking to Thwart Bio-IP Theft

SUKANTA BHATTACHARJEE, Center for Cyber Security, New York University Abu Dhabi, UAE
JACK TANG, Tandon School of Engineering, New York University, USA
SUDIP PODDAR, Department of Electrical Engineering, National Taiwan University of Science and Technology, Taiwan
MOHAMED IBRAHIM, Centralized Design and Test Engineering Group, Intel Corp., USA
RAMESH KARRI, Tandon School of Engineering, New York University, USA
KRISHNENDU CHAKRABARTY, Duke University, USA

It is expected that as digital microfluidic biochips (DMFBs) mature, the hardware design flow will begin to resemble the current practice in the semiconductor industry: design teams send chip layouts to third-party foundries for fabrication. These foundries are untrusted and threaten to steal valuable intellectual property (IP). In a DMFB, the IP consists of not only hardware layouts but also of the biochemical assays (bioassays) that are intended to be executed on-chip. DMFB designers therefore must defend these protocols against theft. We propose to "lock" biochemical assays by inserting *dummy mix-split* operations. We experimentally evaluate the proposed locking mechanism, and show how a high level of protection can be achieved even on bioassays with low complexity. We also demonstrate a new class of attacks that exploit the side-channel information to launch sophisticated attacks on the locked bioassay.

CCS Concepts: • **Security and privacy → Embedded systems security**; • **Hardware → Bio-embedded electronics**; *Electronic design automation*;

Additional Key Words and Phrases: Digital microfluidic biochip, bioassay, IP-theft, locking

ACM Reference format:
Sukanta Bhattacharjee, Jack Tang, Sudip Poddar, Mohamed Ibrahim, Ramesh Karri, and Krishnendu Chakrabarty. 2019. Bio-chemical Assay Locking to Thwart Bio-IP Theft. *ACM Trans. Des. Autom. Electron. Syst.* 25, 1, Article 5 (November 2019), 20 pages.
https://doi.org/10.1145/3365579

5

This research is supported in part by the Army Research Office under grant number W911NF-17-1-0320, NSF Award number CNS-1833622 and CNS-1833624, NYU Center for Cyber Security (CCS), and CCS-AD. A preliminary version of this article has appeared in the proceedings of *ETS* 2018 [13].
Authors' addresses: S. Bhattacharjee (corresponding author), Center for Cyber Security, New York University Abu Dhabi, Abu Dhabi, 129188, UAE; email: sb6538@nyu.edu; J. Tang and R. Karri, Tandon School of Engineering, New York University, New York, 11201, New York, USA; emails: {jtang, rkarri}@nyu.edu; S. Poddar, Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, 700108, West Bengal, India; email: sudippoddar2006@gmail.com; M. Ibrahim, Centralized Design and Test Engineering Group, Intel Corp. San Jose, 95134, California, USA; email: Mohamed.ibrahim@intel.com; K. Chakrabarty, Duke University, Department of Electrical and Computer Engineering, Durham, North Carolina, 27708, USA; email: krish@duke.edu.

1084-4309/2019/11-ART5 $15.00
https://doi.org/10.1145/3365579

1 INTRODUCTION

Microfluidic technologies are now entering a phase of rapid commercialization and deployment. One indicator of this is the recent FDA approval of the Baebies SEEKER, a digital microfluidic platform for medical diagnostics [4]. The chemicals, materials, and biochemical protocols required to realize a modern microfluidic system are becoming increasingly sophisticated and complex, making the task of designing such a system impractical for a single organization. It is expected that the manufacture of microfluidic systems will begin to adopt a horizontal supply chain, where the holders of intellectual property (IP) that dictate a biochip's functionality send their designs to a third-party foundry for fabrication [1]. Such an approach mirrors the manufacturing model established by the semiconductor industry.

An undesirable side-effect of this manufacturing model is the potential for *untrusted* third-parties, who in the course of performing their intended duties, also steal IP or alter designs to modify the functionality of the end product. It is critical that designers of microfluidic systems prevent IP theft not only to prevent financial losses but also to preserve the trust of end users. Grey market devices fabricated with lower quality may not perform to the same standard as authentic devices, which may lead to faulty operation. Given that microfluidic systems are commonly employed in mission-critical applications, this would lead to a severe erosion in trust.

One of the most promising microfluidic technologies being deployed today is based on digital microfluidic biochips (DMFBs) [4, 16]. DMFBs operate according to a sequence of low-level control signals that are derived from the high-level biochemical assay (bioassay) specification, which forms the IP. The bioassay designer must provide this high-level specification to the foundry but will then be susceptible to IP theft. To address the need for IP protection on DMFBs, this article presents the concept of biochemical assay locking.

Our specific contributions are as follows:

(1) We propose to lock biochemical assays through the insertion of dummy mix-split primitives.
(2) We define new *bioassay-specific* security metrics to evaluate the effectiveness of the proposed scheme.
(3) We analyze the key strength, which differs fundamentally from classical encryption and logic locking in that protocols are executed in the fluidic domain.
(4) We show how the spectroscopy can be used to extract side-channel information to launch more sophisticated attacks on the locked bioassay.
(5) We validate the approach with experiments on several biochemical assays and show that negligible overhead is required to achieve satisfactory performance.

The rest of the article is organized as follows: In Section 2, we provide background information on biochemical assays and its implementation on DMFBs. An overview of the untrusted DMFB design flow and its potential vulnerabilities are also presented along with related works on biochip IP protection. In Section 3, we present our proposed locking technique. We derive security metrics in Section 4 and perform a detailed security analysis in Section 5. We then show experimental results in Section 6, and conclude in Section 7.

2 BACKGROUND

The Digital microfluidic biochip (DMFB) consists of two parallel plates. The bottom plate is patterned with addressable electrodes to actuate fluid droplets and the top plate is used as a reference electrode. A dielectric layer and a hydrophobic layer is deposited on both plate. To reduce the sample evaporation, contamination, and facilitate droplet operations, DMFB is filled with silicone

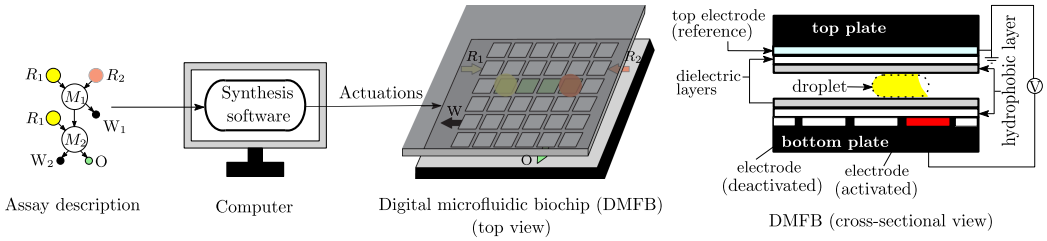

Fig. 1. Digital microfluidic biochips (DMFBs) use grids of electrodes to manipulate discrete droplets. The actuation sequence (set of control signals) is derived from a sequencing graph through high-level synthesis and is sent to the DMFB from a controller unit.

oil between the two plates, and the sample droplets are immersed in an oil medium. DMFBs operate according to the principle of electrowetting-on-dielectric (EWOD): the modulation of contact angle between a droplet and a hydrophobic surface as a function of applied electric potential [25]. EWOD can be harnessed for the precise control of droplets, sandwiched between two plates, by applying the potential difference between the two plates. Non-transparent materials can be used in both plates to hide the observability of the droplets on the DMFB. By properly sequencing voltages on adjacent elecrodes, operations such as mixing, splitting, and transport can be implemented, and these can in turn be used to construct complex biochemical assays.

The control signals used to drive a DMFB array are called actuation sequences, and are generated through a high-level synthesis flow [33]. The input to the synthesis flow is the biochemical assay to be executed on-chip, which is typically specified in the form of a directed acyclic graph (DAG). Nodes represent fluid operations and the edges represent dependencies. This forms one major component of the IP required to fabricate a functional DMFB. The output is the actuation sequence, which is a set of electrode activation patterns to be applied to the DMFB at a fixed rate (Figure 1).

Recently, the basic execution of actuation sequences on a DMFB have been extended to incorporate *conditional execution* [17, 21]. This is driven by the need for advanced biochemical protocols that alter their functionality depending on intermediate chemical reactions, and by the need for dynamic re-execution in case of run-time faults. This functionality will be leveraged in this work to unlock bioassays, which will be illustrated in Section 3.

2.1 Untrusted DMFB Design Flows

We consider the DMFB design flow in Figure 2(a). The design begins with the bioassay designer, or biocoder, who creates the biochemical assay and sends it to a third-party foundry for fabrication. The third-party foundry takes this bioassay, along with information on the fluids that the hardware must handle, cost and area constraints, and creates an integrated DMFB platform along with the synthesized actuation sequences. Such a manufacturing model offloads the burden of integrating the DMFB synthesis software with current hardware capabilities, which are subject to frequent change [10]. The completed DMFB platform is returned to the biochip designer, who can sell the platform to end users, or keep the platform for personal use. This custom design flow is in contrast to the general-purpose design flow that is often discussed in the DMFB design automation literature; in such works, it is assumed that the biochip designer can synthesize the actuation sequence and execute it on a programmable DMFB [1]. In this article, we consider the following threat model.

Threat Model: We consider the third-party supply chain for the biochip product development where a bioassay developer outsources the bioassay description (an IP) to the design house for the fabrication. The design house fabricates the biochip and develops the actuation sequence to realize

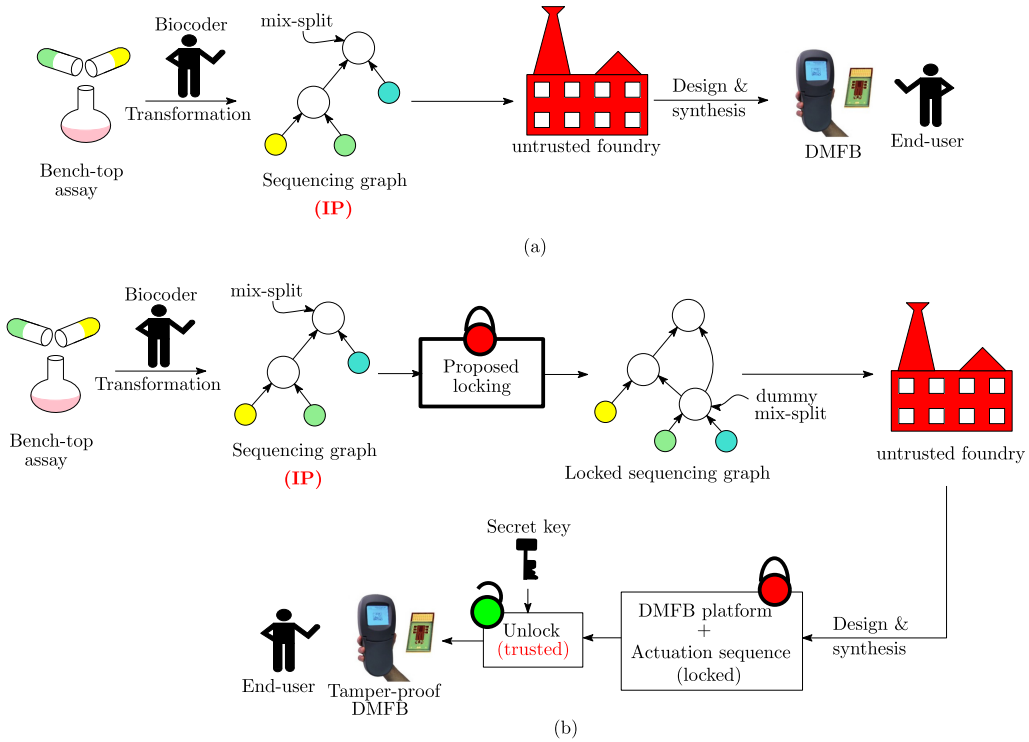

Fig. 2. (a) Untrusted DMFB platform design flow. The designer of the bioassay sends a sequencing graph to a foundry for fabrication. This forms the IP and can be stolen by untrusted foundries. (b) Proposed design flow. The bioassay designer inserts locking primitives that obscure the functionality of the design. The locked DMFB platform can be unlocked either through application of a secret key or removal of the inserted locking primitives.

the bioassay on a DMFB platform, and sends these to the bioassay developer. An attacker in the untrusted design house is motivated to steal the IP from the developer without incurring any development cost. The bioassay owner is the defender who locks the bioassay before sending it to the foundry. The bioassay owner unlocks the bioassay using the correct key and ensures the black-box usage of the DMFB platform before selling it to the market. The attacker in the untrusted foundry can use an unlocked chip to run the assay and observe the fluids in the reservoirs. However, the attacker cannot observe the droplet movements and access actuation sequence on the unlocked biochip.

2.2 Related Prior Work

Security issues specific to DMFB platforms have recently been uncovered [3, 39], many arising as a consequence of untrusted supply chains [1] and Trojans [28, 32, 37, 38]. To counter IP theft, encryption of biochemical assays has been proposed at the *fluidic* level [2]. This approach uses a "fluidic multiplexer" (FMUX) as an encryption primitive that is inserted into the original assay. The FMUX selects between two input droplets for forwarding to the output depending on the presence/absence of a control droplet. Note that there is an undesirable mix between the reference and/or control droplets (specially used for multiplexing) with the functional droplet (input droplets to the multiplexer) during the multiplexing process. Hence, after multiplexing the output droplet

is contaminated. The major shortcomings of the FMUX-based assay encryption approach are that it can be broken through attacks executed in parallel (the size of the key is small, which is determined by the number of FMUXs used in the encryption process, e.g., the FMUX-based encryption approach [2] used only 8-bit key for encryption) and that the output droplet must be mixed with an unspecified inert reference droplet during multiplexing. The design of the reference droplet is an open problem, and would lead to incorrect droplet concentrations anyway. This is due to the inherent limitation of microfluidic logic gates [44] used to realize the fluidic MUX. Furthermore, implementation of the FMUX requires large chip area. This line of research takes some cues from the hardware security literature, where techniques identified as "logic locking" and "logic encryption" are used to protect VLSI designs from IP theft and unauthorized usage [26, 43]. We note that DMFBs are only one class of biochips, and that security issues are also being discovered in other design paradigms such as flow-based biochips [29–31, 35, 36].

3 PROPOSED LOCKING

We propose to lock bioassays by hiding true mix-split operations among randomly inserted dummy mix-splits. Conditional execution capabilities of state-of-the-art DMFBs are used to select which mix-splits to activate/deactivate. We target mix-split operations, because they are abundant in nearly all bioassays, and they are critical for correct operation; if an attacker selects the incorrect mix-splits to activate/deactivate, then fluid outputs will be corrupted (Figure 2(b)).

3.1 Preliminaries: Dummy Mix-Split

A dummy mix-split is a conditional mix-split operation that is not part of the original bioassay sequencing graph. A mix-split operation takes two input droplets, mixes them, and splits them into two output droplets of equal volume. A conditional mix-split operation is a mix-split operation that either mixes or does not mix two input droplets, based on some key value. We assumed that with key value 1, the mix-split operation stalls and then forwards the two input droplets to the two outputs without mixing them. With key value 0, the mix-split operation occurs normally. Mix-split operations can be implemented on the DMFB as a "virtual module," where a pre-defined number of electrodes are reserved for mixing. The two droplets to be mixed are routed to two virtual input electrodes, merged, and then routed around the virtual module for mixing. The mixing time is declared as part of the architectural specification of the DMFB platform. When mixing is complete, the droplets are split and sent to two virtual output electrodes for routing to subsequent operations [19]. The virtual mix-split module can be of variable size, such as 1×4 or 3×4 [23].

In a standard mix-split operation, the input and output electrodes are interchangeable. In a conditional mix-split, it is important that the input droplets are forwarded to the correct output port, otherwise the bioassay will no longer be correct. We introduce new symbolic notation to represent both standard mix-split and dummy mix-split operations, as shown in Figures 3(c) and 3(d). We use two different colored circles to identify the input and output ports of a mixing node in the sequencing graph. For example, in a 1×4 array mixer, the leftmost and rightmost cells can be used for inputs and outputs. This representation of mix-split operations helps the biocoder to hide the difference between the original and dummy mix-split operations in the locked sequencing graph.

3.2 The Method

The process of locking a DAG proceeds as follows: the biocoder creates the bioassay, then replaces all mix-split operations with conditional mix-split operations with key value 0. Then dummy mix-splits are randomly inserted, which are deactivated with key value 1. The correct key values are kept secret. The locked DAG is sent to the foundry for synthesis of the actuation sequences and

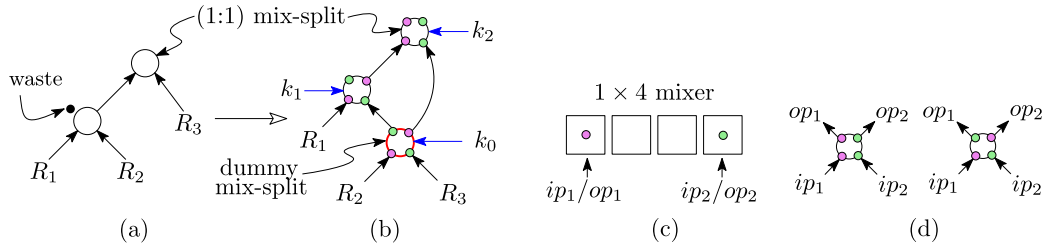

Fig. 3. (a) The bioassay specified as a sequencing graph. (b) Bioassay locking: Dummy mix-splits are added. A secret key dictates which mix-splits should be activated or deactivated. (c) A 1×4 linear mixer has two input/output (ip/op) ports, which we denote with colored circles. (d) The corresponding graph-level representation of the mix-split operation.

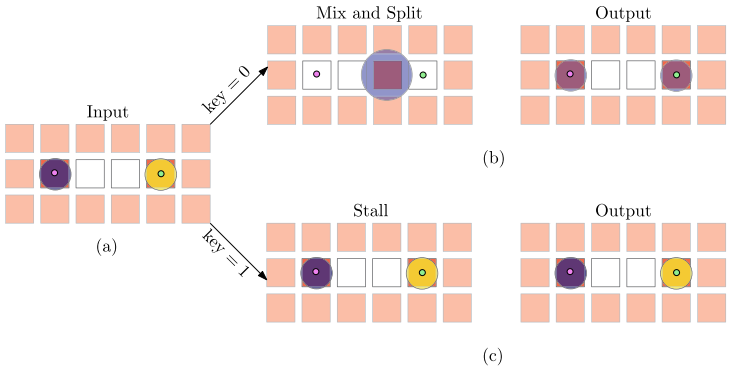

Fig. 4. (a) Input droplets for a 1×4 mix-split. (b) Conditional execution with key = 0 results in a standard mix-split operation. (c) Key = 1 results in a stall with no mixing.

incorporation into a hardware platform. The dummy mix-splits are indistinguishable from real mix-splits, thus hiding the true functionality of the bioassay and preventing its unauthorized use. When the fabricated DMFB platform is returned, the end user must unlock the device by providing the correct key values for each mix-split operation. Alternately, they may remove the dummy mix-split operations. Note that this method does not pose any restrictions on existing synthesis algorithms, so the proposed modifications can be easily incorporated.

Example 1. Consider the input sequencing graph shown in Figure 3(a), where two droplets of input reagents R_1 and R_2 are mixed together. After mixing, one of the two resultant droplets is mixed with a droplet of input reagent R_3. Before sending it to an untrusted design house, the input sequencing graph is locked by adding a dummy mix-split operation between R_2 and R_3. Mixing operations are realized depending on the 0/1 value of the particular key bits. Without loss of generality, we assume that if the key value associated with a mixing operation is zero, two input droplets are mixed (Figure 4(b)). Otherwise, two input droplets stall without mixing (Figure 4(c)). The correct key for the example in Figure 3 is $k_2k_1k_0 = 001$. After applying the correct key, the unlocked actuation sequence transports two droplets of R_2 and R_3 to the input ports of the desired mix-split modules. The mixing between two droplets of R_2 and R_3 is not performed as the key bit k_0 is set to one. However, the remaining two mixing operations are executed, as desired. Hence, the unlocked actuation sequence preserves the correctness of the bioassay.

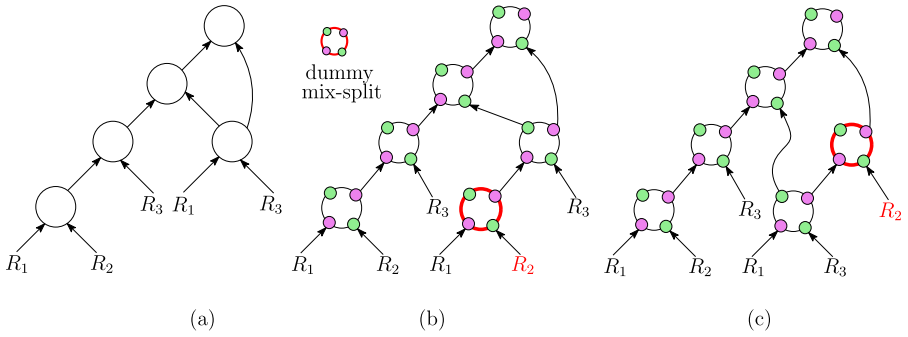

Fig. 5. (a) Input sequencing graph, (b, c) sequencing graph after inserting dummy nodes with extra input reagents.

3.3 Placement of the Dummy Mix-Splits

We can add a dummy mix-split operation into a sequencing graph in several ways as described below.

Add Extra Droplets

The simplest way is to add extra fluid droplets into a sequencing graph. Consider the sequencing graphs shown in Figures 5(b) and 5(c), in which an extra input reagent droplet is added as a leaf node using a dummy mix-split operation. We have highlighted dummy mix-split nodes with a separate color and two input/output ports of each mix-split operations are distinguished with two small circles of different colors. In Figure 5(b), we have to use a different input reagent (R_2 or R_3) from the other one (R_1) used in the dummy mix-split operation. However in case of Figure 5(c), we can use any input reagents. If a wrong key is used to unlock the fabricated DMFB, then undesired mixing with input reagent may take place.

Reuse Waste Droplets

A waste droplet available in the sequencing graph can be mixed with other intermediate droplets in a dummy mix-split operation. However, we cannot choose any arbitrary droplet, because it may create a cycle in the locked sequencing graph. This is a design rule violation, as sequencing graphs with cycles are not synthesizable. For each waste droplet, we associate a subgraph of the sequencing graph on which the waste droplet is generated on the root node of that subgraph. We denote it as the "waste-subgraph" corresponding to the waste droplet. Figure 6(a) shows the *waste-subgraph* for the waste droplet w_2. We use an intermediate droplet from subgraph, that is disjoint from the *waste-subgraph*, to participate in a dummy mixing node. In Figure 6(b) the waste droplet w_2 and an intermediate droplet are used in a dummy mix-split operation.

We may also combine a waste droplet with an intermediate droplet in a dummy mix-split operation that lies on the forward path starting from the root node of *waste-subgraph* associated with the waste droplet. Figure 6(c) shows the sequencing graph after inserting a dummy mix-split operation that combines waste droplet w_2 with an intermediate droplet.

Combine Two Subgraphs

Finally, we may combine two droplets from two disjoint subgraphs of the input sequencing graph in a dummy mix-split operation. If a wrong key is used, then undesirable mixing between the fluids represented by the two disjoint subgraphs is carried out, corrupting the assay outcome. We have adopted a graph traversal technique for selecting the candidate subgraphs. We start from

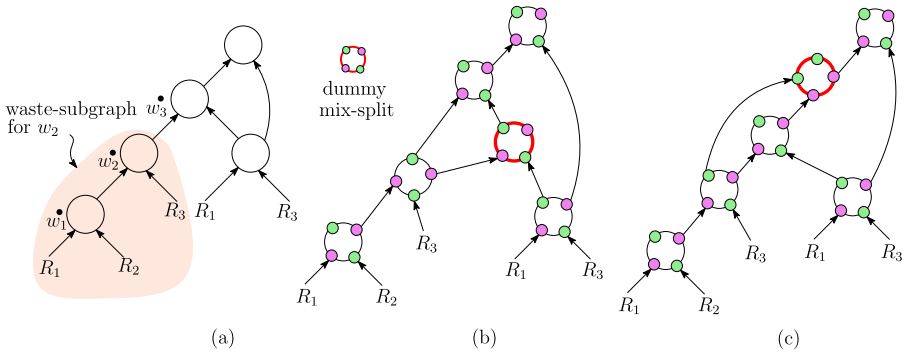

Fig. 6. (a) Input sequencing graph, (b, c) sequencing graphs after inserting a dummy node with a waste droplet as one of the input.

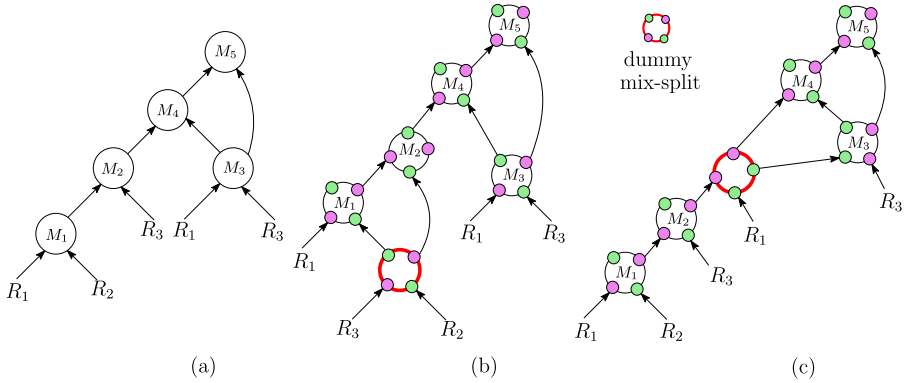

Fig. 7. (a) Input sequencing graph, (b, c) sequencing graphs after inserting a dummy node that use droplets from two different and disjoint subgraphs.

an arbitrary leaf node (i.e., input reagent) and traverse in the forward direction. If two disjoint subgraphs are found, then we use them in a dummy mix-split node. Otherwise, we re-start traversing from another leaf node in the sequencing graph.

Figure 7(b) is generated by combining two droplets of R_2 and R_3 in a dummy mix-split node, and these two disjoint subgraphs (consist of a reagent node only) are found by following the path $(R_1, M_1, M_2, M_4, M_5)$ in the sequencing graph shown in Figure 7(a). Similarly, Figure 7(c) combines droplets corresponding to two left subgraphs of M_3 and M_4 that lie on the path (R_3, M_3, M_4, M_5) of the sequencing graph shown in Figure 7(a).

3.4 Critical Path Length Aware Placement of the Dummy Mix-splits

So far, we inserted the dummy mix-split operations randomly into the input sequencing graph. Random insertion may increase in the length of the critical path in a locked sequencing graph. We propose a critical path length aware insertion of dummy mix-splits operations. We define the length of a root to leaf path in the sequencing graph as the summation of the mixing time of all nodes on that path. The length of the critical path, which determines the minimum assay completion time, is the maximum among all such paths. Figure 8 shows the input sequencing graph where highlighted vertices and edges lie on the critical path. The length of the critical path (L_c) is $4 \cdot t_{mix}$, where t_{mix} denotes the mixing time of each mixing node. The length of the critical path L_c

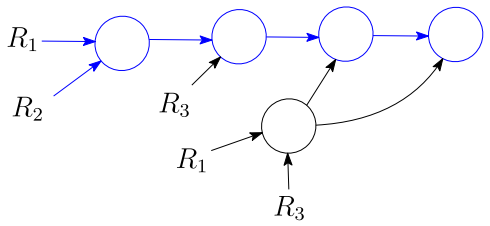

Fig. 8. Input sequencing graph where edges lying on the critical path are highlighted.

is passed as an input to the locking routine, which inserts dummy mix-split operations randomly. If it is not possible to insert a dummy mix-split satisfying the maximum length of critical path, then the routine increases the critical path length. Experimental results (Section 6) reveal that the assay execution time overhead in the critical path length aware dummy insertion technique is on average 20% and 33% less for PCR mixture preparation and PCR mixture droplet streaming, respectively, than when not considering it. We present an outline of the critical path length aware dummy mix-split node insertion technique in the following paragraph.

When adding an extra fluid droplet into a sequencing graph, we select an edge that is not on the critical path to insert dummy mix-split operation. For example, in Figure 8, we choose a black edge to insert a dummy mix-split operation. When reusing a waste droplet, a *waste-subgraph* can be combined with one of the disjoint subgraphs. We select a disjoint subgraph to combine it with the *waste-sugraph* using the dummy mix-split operation if the length of the critical path in the locked sequencing graph remains the same. If no such disjoint sub-graph exists for the *waste-subgraph*, then we repeat the process with another waste droplet, i.e., a new *waste-subgraph*. Similarly, in case of combining two subgraphs, we randomly choose two disjoint sugraphs to combine using a dummy mix-split operation that does not increase the critical path length in the locked sequencing graph. If there exists no such pair of disjoint subgraphs, then we select two subgraphs that increase the critical path length in the locked sequencing graph minimally.

3.5 Implementation Details

The proposed dummy mix-split-based unlocking method is fast and simple. The weaknesses is that the dummy mix-splits introduce stalls into the assay. Stalls may be unacceptable for assays with strict completion time requirements, but slows down brute-force attacks. The ease of use makes this method well-suited for applications where one can sell the secret keys to end users.

If stalls are not acceptable, then one may alter the synthesized actuation sequence to trim out the dummy mix-split operations. Existing Methods can be used for the automated removal of the dummy mix-split operations [10]. It is computationally efficient to change a synthesized actuation sequence than to generate it from scratch, thus preserving one motivation for outsourcing DMFB fabrication.

Stall removal requires that the end user process the synthesized actuation sequence. A trusted third party should provide the software. The actuation sequence provided by the foundry needs to be accessible and modifiable. This is a complex usage scenario, but has the advantage of recovering the original assay, which can be executed with zero overhead. However, an attacker who gains physical possession of the DMFB platform can extract the unlocked actuation sequence and thus reverse engineer the bioassay. So, this method is better suited to private users who will not relinquish physical control of the unlocked DMFB, such as researchers developing novel bioassays. When the end user is not trustworthy, the bioassay owner unlocks the bioassay using the correct key and ensures the black-box usage of the DMFB platform before selling it to the

market. For a successful attack, an attacker (e.g., foundry) has to overcome the tamper-proof feature of the unlocked DMFB. Note that the DMFB technology is emerging and the security and trust issues during manufacturing are yet to be considered in practice. We follow the reasoning that the semiconductor industry adopted an untrusted third-party fabrication business model, which is also vulnerable to similar attacks by the foundry. Several solutions have been proposed based on physical functions (PUFs) to thwart invasive attacks on ICs. For example, a protective coating that ensures some degree of randomness can be used to fingerprint the device [42]. Tampering with the protective coating changes the fingerprint, which will raise an alarm. We can use a similar coating PUF in the tamper-proof DMFB.

4 SECURITY METRICS

A locked design, upon application of an incorrect key, must produce an output that is dissimilar from the correct output. In classical encryption, this requirement is captured through the notion of indistinguishability of the ciphertext. In logic locking, the difference between outputs due to correct and incorrect keys is quantified through the Hamming distance metric, with 50% Hamming distance being ideal [26]. A bioassay experiment can either be qualitative or quantitative, direct or indirect. In a quantitative measurement of the bioassay outcome, the response of a stimulus is transformed to a value. For example, in an immunoassay, the specificity of the antigen-antibody reaction is quantitatively measured in terms of fluorescence signal intensity after the reaction. Because of the unavoidable variation in a biological response, the bioassay outcome is also measured statistically (e.g., DNA analysis). The correctness criteria (e.g., selectivity, stability, accuracy, and precision) also varies with the bioassay. Therefore, we believe that *security metrics for bioassay locking are bioassay-dependent.* This is a significant departure from VLSI logic locking, where security metrics are circuit agnostic. We expect that a multitude of security metrics will be discovered for related families of bioassays.

Sample preparation is an essential step in almost all biochemical protocols for mixing two or more biochemical reagents in a given volumetric ratio. In molecular analysis, 90% of cost and 95% of time is associated with sample collection, transportation, and preparation [9]. DMFB offers a promising fluid handling platform for on-chip sample preparation that reduces the overall assay completion time and cost. As sample preparation is ubiquitous in the bioassay, we focus on the sequencing graph of sample preparation in the bioassay locking and therefore define the bioassay security metric in terms of output ratios.

4.1 Preliminaries: Sample Preparation

In many bioassays, a desired ratio of input reagents must be generated as a pre-processing step. This process is called sample preparation, and in a DMFB it is typically implemented by repeatedly mixing two droplets of equal volume and splitting the resultant droplet into two equal size droplets (i.e., using a 1:1 mix-split operation [9]). Several algorithms have been proposed in the literature for optimizing this task [7–9, 12, 14, 24, 27].

We consider sample preparation assays that take k reagents $R_1, R_2, \ldots, R_i, \ldots, R_k$ and generates a mixture with ratios of $O = \{c_1 : c_2 : \cdots : c_i : \cdots : c_k\}$, where c_i $(0 \leq c_i \leq 1)$ denotes the corresponding concentration factor (CF) of reagent R_i. A pure reagent has $CF = 1$ while neutral buffers have $CF = 0$. Since O is a mixing ratio, it must satisfy $\sum_{i=1}^{k} c_i = 1$ [41]. Existing DMFB sample preparation algorithms transform the input ratio depending on the underlying mixing model (e.g., (1:1) for DMFBs) and user-defined error tolerance limit ϵ, where $0 < \epsilon < 1$. Formally, in case of (1:1) mixing model, the target ratio O is transformed as $O' = \{\frac{x_1}{2^d} : \frac{x_2}{2^d} : \cdots : \frac{x_i}{2^d} : \cdots : \frac{x_k}{2^d}\}$ by choosing $d \in N$ such that $\max_i\{|c_i - \frac{x_i}{2^d}|\} \leq \epsilon$. The transformed ratio O' is used in the DMFB

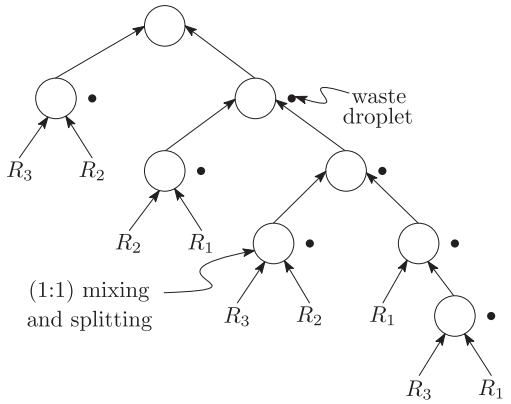

Fig. 9. Sequencing graph generated with MinMix [41] for target ratio $\{R_1 : R_2 : R_3 = 7 : 14 : 11\}$.

sample preparation algorithms for generating a sequencing graph (also known as mixing graph) to represent the sequence of mix-split steps to achieve the mixing ratio O'. The following example illustrates the sample preparation on the DMFB.

Example 2. In the preparation of Plasmid DNA by Alkaline Lysis with Sodium Dodecyl Sulfate (SDS), a mixture of three input reagents are required [20] in the ratio $\{R_1 : R_2 : R_3 = 0.22 : 0.44 : 0.34\}$. In case of DMFB supporting (1:1) mixing model and user-defined error tolerance limit $\epsilon = 0.001$, the ratio can be transformed as $\{R_1 : R_2 : R_3 = \frac{7}{2^5} : \frac{14}{2^5} : \frac{11}{2^5}\}$ by choosing $d = 5$. Note that, $\max\{|0.22 - \frac{7}{32}|, |0.44 - \frac{14}{32}|, |0.34 - \frac{11}{32}|\} \leq \epsilon (= 0.001)$. The sequencing graph for the transformed ratio is shown in Figure 9.

A sample preparation assay may generate m different outputs, which we denote as a set $T = \{O_1, O_2, \ldots, O_j, \ldots, O_m\}$, where each O_j specifies a different mixture of reagents.

4.2 Output Ratio Corruption

When an incorrect key is applied to the locked bioassay, we desire a large number of outputs to be corrupted beyond the error tolerance ϵ. We therefore define our security metric as *the proportion of outputs whose ratios exceed the error tolerance*. We call this *output ratio corruption (ORC)*. We can determine whether an individual output is corrupted by measuring the uniform norm of the difference between an output and its specification. If this norm exceeds ϵ, then it is corrupted. We define a helper function to indicate corruption as

$$\Phi(j) = \begin{cases} 1 & ||\hat{O_j} - O_j||_\infty > \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where we use the uniform norm defined as

$$||\hat{O_j} - O_j||_\infty = \max\{|\hat{c_1} - c_1|, \ldots, |\hat{c_k} - c_k|\}, \tag{2}$$

where (O_j, c_k) indicates the correct ratio values and the hat $(\hat{O_j}, \hat{c_k})$ indicates the actual ratios for some incorrect key. Then, the output ratio corruption is measured as

$$ORC = \frac{100}{m} \sum_{j=1}^{m} \Phi(j). \tag{3}$$

In other words, this is the percentage of outputs that are corrupted. Ideally, it is 100% for all possible incorrect keys. This is different than logic locking, which ideally has 50% corruption as measured

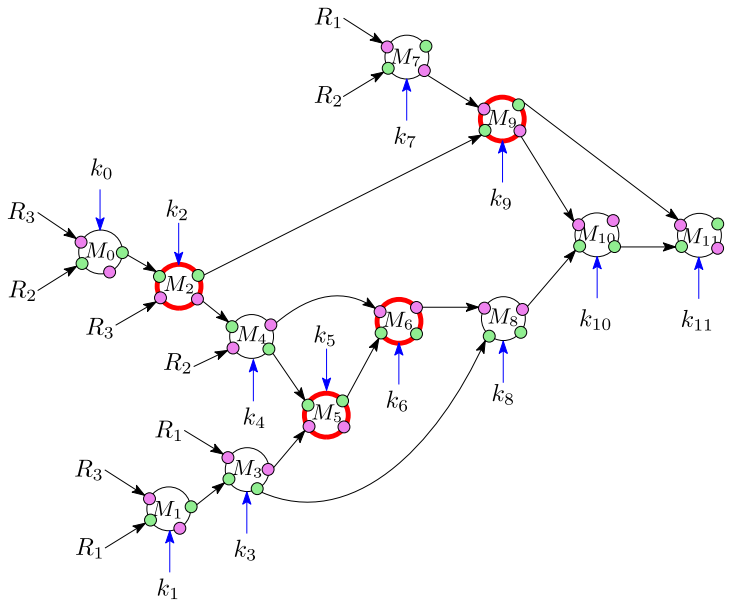

Fig. 10. Locked sequencing graph.

using the Hamming distance metric [26]. Digital outputs that are 100% corrupted are simply the complement of the correct output. No such notion exists for fluid concentrations.

Example 3. Figure 10 shows the sequencing graph after locking the input sequencing graph (Figure 9) by adding four dummy mix-split operation. Note that, the correct key is $k_{11}k_{10}k_9k_8k_7k_6k_5k_4k_3k_2k_1k_0 = 001001100100$. If an attacker uses 00000000100 as a key, then the generated mixing ratio is $\{R_1 : R_2 : R_3 = 10 : 13 : 9\}$. The uniform norm between the correct and incorrect ratio is $\max\{|\frac{10}{32} - \frac{7}{32}|, |\frac{13}{32} - \frac{14}{32}|, |\frac{9}{32} - \frac{11}{32}|\} = 0.09375 > \epsilon(= 0.001)$. Hence $ORC = 100\%$.

5 SECURITY ANALYSIS

In this section, we evaluate the strength of the proposed locking scheme under brute-force attack. We also consider a side-channel attack in which an attacker exploits modern spectroscopy techniques for qualitative and quantitative analysis of the output fluids.

5.1 Brute-force Attacks

A brute-force attack attempts to recover the original bioassay by trying all key combinations. This is equivalent to the problem of identifying which mix-split operations are dummy mixers. Assume the original assay contains n mixing operations. If we insert d dummy mixing operations, then the locked assay will contain $n + d$ mixing operations. If the attacker knows d, then they must consider $\binom{n+d}{d}$ different combinations of dummy mixers to remove. If the attacker has no knowledge of d, then they must consider every possible combination of dummy mixers to remove, up to $d = n$. This is because $\binom{n+d}{d}$ is maximized for $d = n$. Therefore the total number of subsets to consider is 2^{n+d-1}. Only the empty set is invalid, so finally the total number is $2^{n+d-1} - 1$.

Key Length Selection

The key length determines the strength of the locking and is dictated by the number of parallel experiments an attacker can execute p, the bioassay execution time m, and the required minimum

lifetime of the protection λ. If the attacker knows d, then the number of dummy mixers required must satisfy the inequality

$$\binom{n+d}{d} \geq \left(\frac{\lambda \cdot p}{m}\right), \tag{4}$$

where the left-hand side represents the number of brute-force attacks required to break the locking scheme, and the right-hand side is the number of bioassays that can be executed in a given time period. If the attacker does not know d, then the quantity on the left side becomes 2^{n+d-1}, which leads to the inequality

$$n + d \geq \log_2\left(\frac{\lambda \cdot p}{m} + 1\right) + 1. \tag{5}$$

If we assume $\lambda = 20$ years (the lifetime of a U.S. patent), $p = 1{,}000$, and $m = 1$ minute, then the right-hand side quantity is equal to 34.3. That is, as long as the total number of mix-split operations is greater than 34, enough security can be achieved for patent protection. This implies that small bioassays can be secured by adding a large number of dummy mix-split operations, while large bioassays require less. However, in practice, these parameters may vary: the bioassay execution time is variable and doesn't include the time required to prime the DMFB platform and interpret results, while the cost to manufacture a hardware platform will deter parallel attacks.

5.2 Side-channel Attack

Raman spectroscopy

Spectroscopy leverages the interaction of electromagnetic radiation with vibrational modes associated with chemical bonds of the sample molecules for qualitative measurement of its composition [15]. A beam of light (infra-red, visible, or ultra-violet) is passed though the sample and recorded for spectrum analysis where peaks correspond to specific vibrational and rotational modes within the molecule. Each chemical has its own unique fingerprint, which can be cross-referenced with a library of known spectra for identification [15, 22].

Raman spectroscopy is widely used for quantitative analysis of mixture constituents for rapid noninvasive measurements of the concentration of important analytes, e.g., glucose, lactic acid, cholesterol, and triglyceride [5, 6]. A quantitative analysis framework based on Raman spectroscopy is used to estimate the concentration of ethanol in hand sanitizers [22]. In quantitative Raman spectroscopy, the Raman signature is acquired by exciting the target sample with monochromatic laser. The signature, which is the convolution of all molecular signatures present in the sample, is pre-processed for removing unwanted background noises. Multivariate data analysis techniques are used for estimating the concentration of desired analyte by using one or more features extracted from the pre-processed Raman spectra. In the estimation process, a calibration curve is generated beforehand from the Raman spectras of several known concentrations of the target analyte. Figure 11 shows the essential steps in quantitative Raman for estimating the concentration of an analyte present in a complex mixture. The following example illustrates the estimation accuracy of a target analyte's concentration in a mixture.

Example 4. Let us consider an analytical model for estimating the concentration of a reagent (say, R_1) in a mixture of R_1, R_2, and R_3 using Raman spectroscopy. Also assume that the estimation accuracy of the analytical model is 10%. If the original ratio of three reagents in a mixture is $R_1 : R_2 : R_3 = c_1 : c_2 : c_3$, ($c_i$ is the concentration of R_i for $i = 1, 2, 3$), then the estimated value of the concentration of R_1 (say, c_1') can take any value between $(c_1 - 0.1 \times c_1) < c_1' < (c_1 + 0.1 \times c_1)$.

Though Raman spectroscopy has high molecular specificity in qualitative analysis, the performance of Raman for quantitative analysis depends on several factors such as technological advancement of highly efficient laser sources, low-noise detectors, effective Rayleigh filters, sample

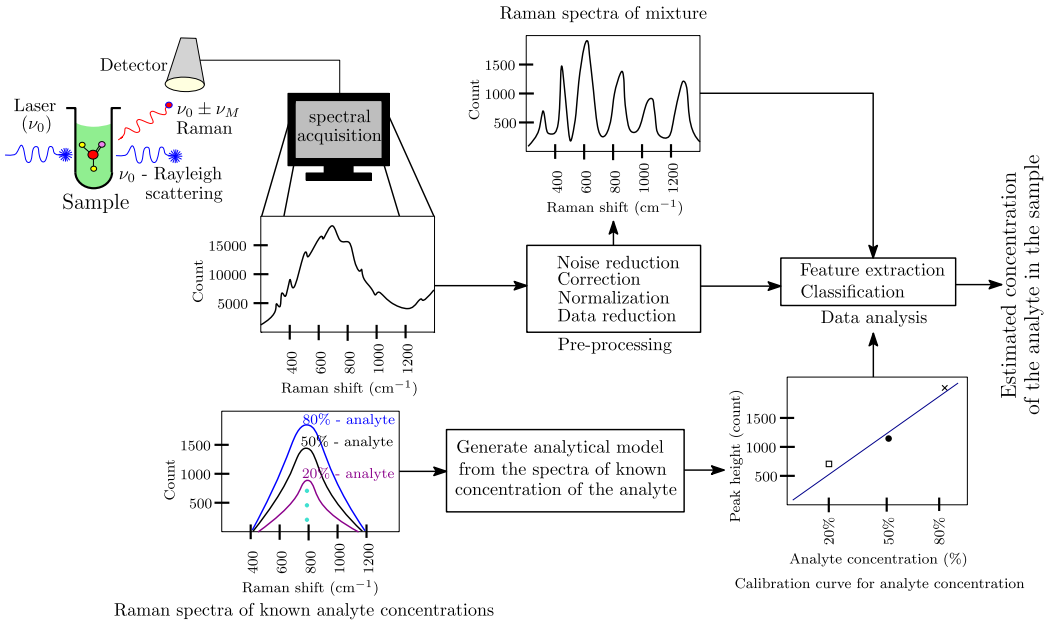

Fig. 11. Quantitative analysis framework to estimate the concentration of an analyte in a mixture using Raman spectroscopy.

processing for data calibration, and analytical model used in estimation process. Moreover, Raman spectra becomes more convoluted when dealing with a complex mixture.

Side-channel attacks exploit side information (such as timing, power, delay, electromagnetic radiation, optical, and acoustic) emanating from the implementation of the system to reveal secrets [40]. Side-channel attacks based on power analysis of cryptographic hardware can be used to leak information about the secret key [45]. In microfluidic biochips, fluids in the on-chip reservoirs could be used as a side channel. An attacker can exploit spectral techniques such as infra-red, ultraviolet or Raman spectroscopy on the outputs of the functional biochip to gain additional information, such as chemical constituents and mixing ratios of the fluids at the input/output reservoirs, which can be leveraged to reveal the secret key used to lock the bioassay. In this section, we discuss for the first time how an attacker can exploit the fluidic side channel on sample preparation.

Can an attacker launch a SAT-attack?

Note that that there are only two possibilities for each mixing operation, which can motivate an attacker to launch a SAT-attack [34] on a sample-preparation bioassay. For a SAT-attack to be meaningful, we need an oracle that can verify the correctness of the output. In VLSI chips, the attacker can use an unlocked chip as an oracle. In the scenario of microfluidic biochip, an attacker can use spectroscopy to estimate the concentration of an analyte within a mixture. The attacker also needs to generate an analytical model from the known concentrations of the target analyte within a mixture. An analytical model derived in this manner is not useful when the mixture is changed. Moreover, all analytes within a mixture cannot be estimated, because the signal of the desired analyte becomes convoluted with the signals of other analytes. Therefore, the precise measurement of an output droplet using spectroscopy often leads to inconclusive outcomes. This characteristic restricts the usage of spectroscopy for verifying the correctness of the sample preparation bioassay.

Table 1. Attacker's Effort in Different Locking Schemes with Respect to the Accuracy of Estimation of an Analyte Using Spectroscopy

Locking scheme	Estimation accuracy of an analyte*	#dummy mix-split (d) in the key				
		$d = 3$ (165)†	$d = 4$ (330)†	$d = 5$ (462)†	$d = 6$ (462)†	$d = 7$ (330)†
Subgraph sharing: 2 Waste sharing: 2	5%	37	50	44	32	17
	10%	67	119	163	166	115
	15%	86	158	216	204	127
	20%	116	231	321	320	235

† Total number of keys considering d dummy mix-split operations.
* Spectroscopy is used to estimate the concentration of R_2.

Spectroscopy as a Side-channel Attack Vector Model

It is evident form the previous discussion that the attacker can estimate the concentration of an analyte present in the fluid of an output reservoirs using Raman or any other spectroscopy techniques. In the attack model, we assume that the attacker can estimate the concentration range of an analyte in the output mixing ratio. The estimation accuracy depends on the spectroscopy technique, target analyte, underlying analytical model, and sample processing for generating a calibration curve from known concentration of the analyte. From the perspective of microfluidics technology, Raman is a sophisticated and costly scenario, even if it can be performed offline. The following example shows the side-channel attack where the attacker exploits spectral information to launch a more sophisticated attack compared to the brute force.

Example 5. We lock the input sequencing graph (Figure 9) by adding four dummy mix-split operations (Figure 10). The locking scheme is given in the Table 1. Note that the locked sequencing graph (Figure 10) generates the mixing ratio $\{R_1 : R_2 : R_3 = 7 : 14 : 11\}$ when it will be unlocked with the correct key.

The attacker does not know the number of dummy mix-split operations used in the locking process. Therefore, the attacker needs to exercise all possible values of d for finding the correct key. Table 1 reports the attacker's effort for different estimation accuracy of the analyte in the mixture. For each estimation accuracy, Table 1 reports the number of sequencing graph that the attacker needs to check, by varying the attacker's choice in the number of dummy mix-split operations (d) used to lock the input sequencing graph. It is evident that the attacker's effort in identifying the correct key is dependent on the estimation accuracy of the analyte in the mixture.

6 EXPERIMENTAL RESULTS

We implemented the proposed locking scheme in Python 2.7 and evaluated it on several benchmark DMFB sample preparation assays: Polymerase chain reaction (PCR) mixture preparation [27], PCR mixture droplet streaming [10], multi-target dilution [7], and two other mixing ratios used in real-life bioassays. Locked assays were synthesized using MFStaticSim [18] with list scheduler, left-edge placer, and modified maze router. We assume mix-splits take four seconds on a 2Hz DMFB.

6.1 Output Ratio Corruption

We inserted five dummy mix-splits into the assays to show that acceptable corruption can be achieved with low overhead. We randomly selected a large number (1,000) of input keys and compared the corresponding output ratios against the correct outputs with the output ratio corruption metric. For the PCR mixture droplet streaming assay, we set $\epsilon = 1/32$. We plot the histogram of ORC values in Figure 12(a), and observe the values are concentrated at 100%. The multi-target

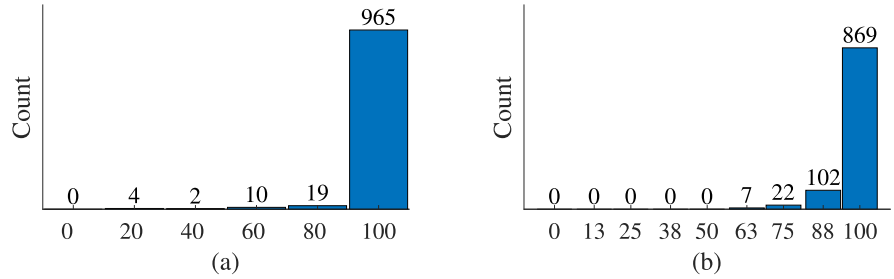

Fig. 12. Histograms of Output Ratio Corruption [%]. One thousand random keys were used to unlock the assays. (a) PCR mixture droplet streaming. (b) Multi-target dilution.

Table 2. Assay Execution Varies Time with Respect to d, the Number of Inserted Dummy Mix-split Operations

Assay	#mix-splits n	Execution time [s]	DMFB size	Locking scheme	Locked assay execution time [s]						
					$d = 4$	$d = 5$	$d = 6$	$d = 7$	$d = 8$	$d = 9$	$d = 10$
PCR mixture droplet streaming [10]	15	35 (5)	15 × 20	Random	41 (6)	47 (7)	49 (7)	69 (9)	57 (8)	NA	NA
				Critical-path aware	33 (5)	33 (5)	35 (5)	37 (5)	39 (6)		
PCR mixture preparation [27]	19	45 (8)	15 × 20	Random	57 (9)	65 (9)	65 (10)	63 (10)	69 (11)	NA	NA
				Critical-path aware	45 (8)	53 (9)	53 (9)	51 (9)	53 (9)		
One-step Miniprep method [11]	8	51 (8)	15 × 20	Random	69 (11)	69 (11)	77 (12)	81 (13)	87 (14)	87 (14)	93 (15)
				Critical-path aware	51 (8)	51 (8)	51 (8)	51 (8)	51 (8)	57 (9)	57 (9)
Splinkerette PCR method [11]	8	41 (6)	15 × 20	Random	53 (8)	59 (9)	57 (9)	63 (10)	69 (11)	81 (13)	93 (15)
				Critical-path aware	41 (6)	41 (6)	41 (6)	41 (6)	47 (7)	47 (7)	69 (9)
Multi-target dilution [7]	28	45 (8)	15 × 20	Random	NA	57 (8)	63 (10)	75 (8)	75 (11)	69 (8)	81 (12)
				Critical-path aware		45 (8)	51 (8)	51 (9)	53 (9)	51 (9)	63 (9)

The length of a critical path in the locked sequencing graph is reported within the parenthesis.

dilution assay was evaluated with $\epsilon = 1/128$. We also observe the same concentration of values near 100% (Figure 12(b)). For both of these assays, no outputs with 0% corruption were observed. The PCR mixture preparation assay has only one output, which we observed as always being corrupted when tested with $\epsilon = 1/256$. Note that the error limit ϵ was used in generating the source sequencing graphs. While this evidence suggests that bioassay locking can ensure that no random guess gives a correct output, we have not provided theoretical guarantees that this is true.

6.2 Overhead

We quantify the overhead in terms of number of dummy mix-splits and assay execution time for random and critical-path length aware dummy mix-split node insertion strategies. The number of inserted dummy mix-splits correlates linearly with chip area overhead, as a mixing module must be reserved on-chip. As seen in the evaluation of ORC, the outputs are sufficiently corrupted even with a small number of dummy mix-splits, so the overhead is small. The overhead in terms of assay execution time is shown as a function of number of dummy mix-split operations in Table 2. In some cases the locked assay time can increase by over 50% if we use random insertion strategy. Increase in the assay execution time is due to unnecessary increase of the critical path length of the locked sequencing graph. As expected, the run-time overhead in critical path length aware locking scheme is significantly less than that of random insertion strategy. If it is desired to lock

Table 3. Attacker's Effort in the Presence of Locking Schemes Relative to the Accuracy of Estimation of an Analyte Using Spectroscopy

Assay	#mix-split (n)	Locking scheme	Estimation accuracy of an analyte	#dummy mix-split (d) in the key				
				$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 7$
#valid ratios to check in brute force for an attacker's guess in number of dummy mix-split steps (d):				165	330	462	462	330
Synthetic case: Example 5 $R_1 : R_2 : R_3 = 7 : 14 : 11$	8	Subgraph sharing: 2 Waste sharing: 1 Extra reagent addition: 1	5%	61	80	58	22	3
			10%	105	170	197	178	128
			15%	118	204	241	213	144
			20%	127	237	322	326	239
		Subgraph sharing: 1 Waste sharing: 1 Extra reagent addition: 2	5%	45	62	54	33	14
			10%	88	154	204	216	176
			15%	121	229	305	291	204
			20%	150	296	407	400	281
#valid ratios to check in brute force for an attacker's guess in number of dummy mix-split steps (d):				165	330	462	462	330
One-step Miniprep method [11]: $R_1 : R_2 : R_3 = 128 : 123 : 5$	8	Subgraph sharing: 1 Waste sharing: 2 Extra reagent addition: 1	5%	27	58	95	110	88
			10%	79	133	152	128	89
			15%	129	233	295	270	181
			20%	157	285	355	314	197
		Subgraph sharing: 1 Waste sharing: 1 Extra reagent addition: 2	5%	80	182	278	297	225
			10%	132	243	319	312	227
			15%	163	315	422	408	287
			20%	164	323	440	424	292
#valid ratios to check in brute force for an attacker's guess in number of dummy mix-split steps (d):				165	330	462	462	330
Splinkerette PCR method [11]: $R_1 : R_2 : R_3 : R_4 : R_5 = 9 : 17 : 26 : 9 : 195$ Transformed ratio: 2 : 4 : 7 : 2 : 49	8	Subgraph sharing: 2 Waste sharing: 2	5%	5	19	47	72	70
			10%	21	42	69	87	76
			15%	44	112	182	198	145
			20%	45	117	189	201	145
		Subgraph sharing: 1 Waste sharing: 1 Extra reagent addition: 2	5%	24	84	168	210	168
			10%	24	85	171	213	169
			15%	52	140	238	266	196
			20%	56	146	242	267	196
#valid ratios to check in brute force for an attacker's guess in number of dummy mix-split steps (d):				220	495	792	924	792
$R_1 : R_2 : R_3 : R_4 = 17\% : 40\% : 9\% : 34\%$ Transformed ratio: 5:13:3:11	9	Subgraph sharing: 2 Waste sharing: 2	5%	44	101	156	123	111
			10%	84	186	264	262	189
			15%	125	238	325	319	219
			20%	155	309	443	469	368
		Extra reagent addition: 4	5%	84	168	210	168	84
			10%	100	228	348	382	317
			15%	121	258	379	402	324
			20%	175	375	582	672	582

an assay with strict scheduling requirements, then the DMFB designer can remove the stalls as described in Section 3.5.

6.3 Spectroscopy Side Channel

We perform experiments to evaluate the attacker's effort when the attacker exploits the spectral techniques (IR or Raman spectroscopy) to estimate the concentration of an analyte in the output

mixture. In our experiments, we have locked input sequencing graphs[1] for the desired target ratio by inserting four dummy mix-split operations. We use two different locking scheme for each input sequencing graph and evaluate the attacker's effort by varying the estimation accuracy of an analyte. Without knowing the number of dummy mix-split operations in the locked sequencing graph, the attacker has to guess the number of dummy mix-split operations. For each locking scheme and estimation accuracy, Table 3 reports the number of sequencing graphs that an attacker needs to check for a particular guess in the number of dummy mix-split operations.

The experimental results reveal that the spectroscopy-based side-channel information can be leveraged to launch more sophisticated attacks on the bioassay locking compared to the brute force. However, the estimation accuracy of the spectral techniques are very specific to the target analyte and the underlying analytical model. We also note that, the success of the side-channel attack is dependent on the locking scheme. The experimental data shows that if we increase the number of extra reagents as opposed to subgraph and waste sharing, the attacker's effort increases. In this article, we have not investigated the best possible option to lock an input sequence graph. We have posed it as a future research problem in Section 7.

7 CONCLUSION AND FUTURE DIRECTION

We have presented a *practical* biochemical assay locking scheme for DMFBs. We leverage dummy mix-split operations and conditional execution as a locking primitive. This approach is easy to implement and with some overhead in chip area and execution time. Compared to previously reported fluidic locking techniques, this work provides strong key strength without any dependencies on inherent DMFB failure modes, and avoids the severe chip area penalty required to implement a fluidic multiplexer. We defined biochemical assay security metrics in terms of output ratio corruption. We have also shown that spectroscopy-based side-channel information can be exploited by an attacker to launch more sophisticated attacks. It will be a fruitful area of future research to devise a spectroscopy resilient locking scheme for bioassays. Moreover, the protection against a backdoor inserted by the DMFB manufacturer could be an interesting future research direction.

REFERENCES

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

[2] 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 Systems Conference*.152–155.

[3] 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. Bioinform.* 13, 3 (2016), 445–458.

[4] Baebies, Inc. 2017. Baebies SEEKER. Retrieved from http://baebies.com/products/seeker/.

[5] Andrew J. Berger, Tae-Woong Koo, Irving Itzkan, Gary Horowitz, and Michael S. Feld. 1999. Multicomponent blood analysis by near-infrared Raman spectroscopy. *Appl. Opt.* 38, 13 (1999), 2916–2926.

[6] Andrew J. Berger, Yang Wang, and Michael S. Feld. 1996. Rapid, noninvasive concentration measurements of aqueous biological analytes by near-infrared Raman spectroscopy. *Appl. Opt.* 35, 1 (1996), 209–212.

[7] Sukanta Bhattacharjee, Ansuman Banerjee, and Bhargab B. Bhattacharya. 2014. Sample preparation with multiple dilutions on digital microfluidic biochips. *IET Comput. Digit. Tech.* 8, 1 (2014), 49–58.

[8] Sukanta Bhattacharjee, Ansuman Banerjee, Tsung-Yi Ho, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. 2019. Efficient generation of dilution gradients with digital microfluidic biochips. *IEEE Trans. CAD Integr. Circ. Syst.* 38, 5 (2019), 874–887.

[9] Sukanta Bhattacharjee, Bhargab B. Bhattacharya, and Krishnendu Chakrabarty. 2019. *Algorithms for Sample Preparation with Microfluidic Lab-on-Chip*. River Publisher.

[1] We used MinMix [41] algorithm.

[10] Sukanta Bhattacharjee, Sharbatanu Chatterjee, Ansuman Banerjee, Tsung-Yi Ho, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. 2017. Adaptation of biochemical protocols to handle technology-change for digital microfluidics. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 36, 3 (2017), 370–383.

[11] Sukanta Bhattacharjee, Yi-Ling Chen, Juinn-Dar Huang, and Bhargab B. Bhattacharya. 2018. Concentration-resilient mixture preparation with digital microfluidic lab-on-chip. *ACM Trans. Embed. Comput. Syst.* 17, 2 (2018), 49:1–49:12.

[12] Sukanta Bhattacharjee, Sudip Poddar, Sudip Roy, Junin-Dar Huang, and Bhargab B. Bhattacharya. 2017. Dilution and mixing algorithms for flow-based microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 36, 4 (2017), 614–627.

[13] Sukanta Bhattacharjee, Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2018. Locking of biochemical assays for digital microfluidic biochips. In *Proceedings of the European Test Symposium*. 1–6.

[14] Sukanta Bhattacharjee, Robert Wille, Juinn-Dar Huang, and Bhargab. B. Bhattacharya. 2019. Storage-aware algorithms for dilution and mixture preparation with flow-based lab-on-chip. *IEEE Trans. CAD Integr. Circ. Syst.* (2019), early access.

[15] Holly J. Butler, Lorna Ashton, Benjamin Bird, Gianfelice Cinque, Kelly Curtis, Jennifer Dorney, Karen Esmonde-White, Nigel J. Fullwood, Benjamin Gardner, Pierre L. Martin-Hirsch, Michael J. Walsh, Martin R. McAinsh, Nicholas Stone, and Francis L. Martin. 2016. Raman spectroscopy as a quantitative tool for industry. *Nature Protocols* 11, 4 (2016), 664–687.

[16] GenMark Diagnostics, Inc. 2018. ePlex. Retrieved from https://www.genmarkdx.com/.

[17] Daniel Grissom, Christopher Curtis, and Philip Brisk. 2014. Interpreting assays with control flow on digital microfluidic biochips. *ACM J. Emerg. Technol. Comput. Syst.* 10, 3 (2014), 24.

[18] Daniel Grissom, Christopher Curtis, Skyler Windh, Calvin Phung, Navin Kumar, Zachary Zimmerman, O'Neal Kenneth, Jeffrey McDaniel, Nick Liao, and Philip Brisk. 2015. An open-source compiler and PCB synthesis tool for digital microfluidic biochips. *Integr. VLSI J.* 51 (2015), 169–193.

[19] Daniel T. Grissom and Philip Brisk. 2014. Fast online synthesis of digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 33, 3 (2014), 356–369.

[20] Yi-Ling Hsieh, Tsung-Yi Ho, and Krishnendu Chakrabarty. 2012. A reagent-saving mixing algorithm for preparing multiple-target biochemical samples using digital microfluidics. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 31, 11 (2012), 1656–1669.

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

[22] Lilian Norris, Cicely Rathmell, and Yvette Mattley. 2012. Raman spectroscopy as a quantitative tool for industry. *Spectroscopy* 27, 6 (2012), 2916–2926.

[23] Phil Paik, Vamsee K. Pamula, and Richard B. Fair. 2003. Rapid droplet mixers for digital microfluidic systems. *Lab. Chip* 3, 4 (2003), 253–259.

[24] Sudip Poddar, Sukanta Bhattacharjee, Subhas C. Nandy, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. 2019. Optimization of multi-target sample preparation on-demand with digital microfluidic biochips. *IEEE Trans. CAD Integr. Circ. Syst.* 38, 2 (2019), 253–266.

[25] M. G. Pollack, A. D. Shenderov, and R. B. Fair. 2002. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab. Chip* 2, 2 (2002), 96–101.

[26] Jeyavijayan Rajendran, Huan Zhang, Chi Zhang, Garrett S. Rose, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. 2015. Fault analysis-based logic encryption. *IEEE Trans. Comput.* 64, 2 (2015), 410–424.

[27] Sudip Roy, Partha Pratim Chakrabarti, Srijan Kumar, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. 2015. Layout-aware mixture preparation of biochemical fluids on application-specific digital microfluidic biochips. *ACM Trans. Design Autom. Electr. Syst.* 20, 3 (2015), 45:1–45:34.

[28] Mohammed Shayan, Sukanta Bhattacharjee, Tung-Che Liang, Jack Tang, Krishnendu Chakrabarty, and Ramesh Karri. 2018. Shadow attacks on MEDA biochips. In *Proceedings of the ACM International Conference on Computer-Aided Design*. 73:1–73:8.

[29] Mohammed Shayan, Sukanta Bhattacharjee, Yong Rafael Song, Krishnendu Chakrabarty, and Ramesh Karri. 2019. Desieve the attacker: Thwarting IP Theft in Sieve-Valve-based biochips. In *Proceedings of the IEEE/ACM Design, Automation and Test in Europe*. 210–215.

[30] Mohammed Shayan, Sukanta Bhattacharjee, Yong Rafael Song, Krishnendu Chakrabarty, and Ramesh Karri. 2019. Security assessment of microfluidic Fully-Programmable-Valve-Array biochips. In *Proceedings of the IEEE International Conference in VLSI Design*. 197–202.

[31] Mohammed Shayan, Sukanta Bhattacharjee, Jack Tang, Krishnendu Chakrabarty, and Ramesh Karri. 2019. Bioprotocol watermarking on digital microfluidic biochips. *IEEE Trans. Inform. Forensics Secur.* 14, 11 (2019), 2901–2915.

[32] Mohammed Shayan, Jack Tang, Krishnendu Chakrabarty, and Ramesh Karri. 2018. Security assessment of microelectrode-dot-array biochips. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* (2018), early access.

[33] Fei Su and Krishnendu Chakrabarty. 2008. High-level synthesis of digital microfluidic biochips. *ACM J. Emerg. Technol. Comput. Syst.* 3, 4 (2008), 1:1–1:32.
[34] Pramod Subramanyan, Sayak Ray, and Sharad Malik. 2015. Evaluating the security of logic encryption algorithms. In *Proceedings of the International Symposium on Hardware Oriented Security and Trust.* 137–143.
[35] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2017. Security implications of cyberphysical flow-based microfluidic biochips. In *Proceedings of the IEEE Asian Test Symposium.* 110–115.
[36] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2017. Security trade-offs in microfluidic routing fabrics. In *Proceedings of the IEEE International Conference on Computer Design.* 25–32.
[37] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2018. Secure randomized checkpointing for digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 37, 6 (2018), 1119–1132.
[38] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2018. Tamper-resistant pin-constrained digital microfluidic biochips. In *Proceedings of the ACM Design Automation Conference.* 1–6.
[39] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2019. Towards secure and trustworthy cyberphysical microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 38, 4 (2019), 589–603.
[40] Mohammad Tehranipoor and Cliff Wang. 2011. *Introduction to Hardware Security and Trust.* Springer.
[41] William Thies, John Paul Urbanski, Todd Thorsen, and Saman P. Amarasinghe. 2008. Abstraction layers for scalable microfluidic biocomputing. *Natur. Comput.* 7, 2 (2008), 255–275.
[42] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. 2006. Read-proof hardware from protective coatings. In *Proceedings of the Conference on Cryptographic Hardware and Embedded Systems.* 369–383.
[43] Muhammad Yasin, Jeyavijayan Rajendran, Ozgur Sinanoglu, and Ramesh Karri. 2016. On improving the security of logic locking. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 35, 9 (2016), 1411–1424.
[44] Yang Zhao and Krishnendu Chakrabarty. 2010. Digital microfluidic logic gates and their application to built-in self-test of lab-on-chip. *IEEE Trans. Biomed. Circ. Syst.* 4, 4 (2010), 250–262.
[45] YongBin Zhou and DengGuo Feng. 2005. Side-channel Attacks: Ten Years after Its Publication and the Impacts on Cryptographic Module Security Testing. Retrieved from http://eprint.iacr.org/2005/388.

Received January 2019; revised September 2019; accepted September 2019