How Secure are Checkpoint-based Defenses in Digital Microfluidic Biochips?

Mohammed Shayan, *Student Member, IEEE*, Sukanta Bhattacharjee, *Member, IEEE*, Robert Wille, *Senior Member, IEEE* Krishnendu Chakrabarty, *Fellow, IEEE*, and Ramesh Karri *Fellow, IEEE*

***Abstract*—A digital microfluidic biochip (DMFB) is a miniaturized laboratory capable of implementing biochemical protocols. Fully integrated DMFBs consist of a hardware platform, controller, and network connectivity, making it a cyberphysical system (CPS). A DMFB CPS is being advocated for safety-critical applications such as medical diagnosis, drug development, and personalized medicine. Hence, the security of a DMFB CPS is of immense importance to their successful deployment. Recent research has made progress in devising corresponding defense mechanisms by employing so-called checkpoints. Existing solutions either rely on probabilistic security analysis that does not consider all possible actions an attacker may use to overcome an applied checkpoint mechanism or rely on exhaustive monitoring of DMFB at all time-steps during the assay execution. For devising a defense scheme that is guaranteed to be secure, an *exact* analysis of the security of a DMFB is needed. This is not available in the current state-of-the-art. In this work, we address this issue by developing an exact method, which uses the deductive power of satisfiability solvers to verify whether a checkpoint-based defense thwarts the execution of an attack. We demonstrate the usefulness of the proposed method by showcasing two applications on practical bioassays: 1) security analysis of various checkpointing strategies and 2) derivation of a counterexample-guided fool-proof secure checkpoint scheme.**

I. Introduction

Microfluidic technologies are one of the major driving forces towards the miniaturization of laboratory-based biochemical protocols. A microfluidic biochip or lab-on-a-chip (LoC) performs biochemical reactions by consuming nano-/pico-liter volume of reagents [1]. These platforms provide advantages such as minimal sample and reagent use, quicker results, automation, and reduced reliance on high-skilled personnel. Several biochip platforms have been proposed such as digital microfluidic biochip (DMFB) or continuous flow-based microfluidic biochip (CFMB). CFMBs manipulate fluid flow through a network of micro-channel by actuating pressure-driven micro-valves [2]. These are mostly custom designs and lack programmability. DMFB offers a programmable fluidic platform in which discrete fluid droplets can be manipulated through electrical actuations [3].

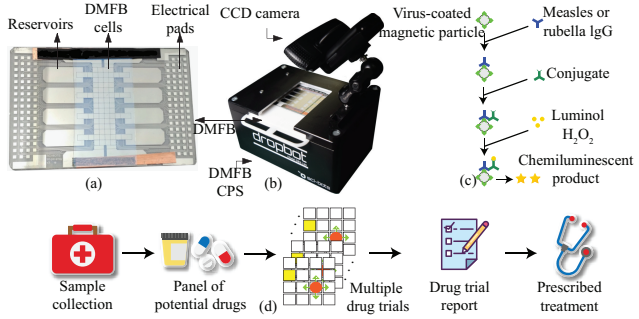


Fig. 1: An open-source DMFB system: (a) DMF biochip and (b) DropBot platform (source: https://sci-bots.com/). (c) Rubella/measles immunoassay description [11]. (d) A personalized drug development process for cancer patients [9], [10].

Biochips have brought a complete paradigm shift in several biochemical applications such as biomedical research, genomics, and environment monitoring. These chips have made a profound impact on health care by redefining point-of-care diagnostics [4], drug research and development [5], as well as personalized medicine [6]. The biochips make diagnostics affordable and accessible compared to traditional bio-labs. For example, the immunoassay platform shown in Fig. 1(a-b) performs low-cost detection of measles and rubella viruses using a single drop of blood (Fig. 1(c)). This was deployed in refugee camps where many basic life necessities were inaccessible [7], [8]. Further, biochips enable diagnostics that were not possible in traditional bio-labs. For example, microfluidics enables the development of personalized medicine needed for cancer patients by running thousands of parallel tests on patient's biosample, as shown in Fig. 1(d). Using traditional techniques, it is not possible to exactly determine which of the many clinical trials that are on offer is appropriate for a particular patient [9], [10].

The global biochip market is projected to reach to $12.3 billion by 2025 from $5.7 billion in 2018 [12]. This is corroborated by the sales [13], investment [14], and acquisitions [15] reported by microfluidic companies. Baebies' SEEKER, a DMFB-based immunoassay platform, received FDA approval

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 NYU Abu Dhabi Center for Cyber Security CCS-AD.

M. Shayan, and R. Karri are with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY, 11201 USA e-mail: (mos283@nyu.edu,rkarri@nyu.edu).

Sukanta Bhattacharjee is with Indian Institute of Technology, Guwahati, Assam, 781039, India (email: sukantab@iitg.ac.in).

Robert Wille is with Institute of Integrated Circuits, Johannes Kepler University Linz, Austria email: (robert.wille@jku.at).

K. Chakrabarty is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: krish@duke.edu).

Manuscript received December 05, 2019 and revised February 22, 2020.

in 2016 [16]. Since then, Baebies has shipped three million tests and raised $13 million in funding [13]. The SEEKER provides a high-throughput quantitative measurement of deadly diseases from the dried blood spot of newborns. 10 Genomics uses a combination of microfluidics, and bioinformatics for single-cell analysis. Since its founding in 2012, it has received $243 million in funding until 2018 [14].

A. Biochip Security: Motivation

As the biochips are penetrating the market, security and trust issues are being uncovered. Biochips have multiple usage scenarios such as in a biomedical research lab and in a remote location. Depending on the usage scenario, biochemical assay implementation faces different threats. To highlight this, we describe three real-life scenarios:

1) A disgruntled employee can tamper with the biochemical experiments to take revenge on colleagues or management [17]. Recently, a chemist at a water treatment plant was found guilty of tampering with a colleague's water test for months [18]. The usage of biochips in such labs increases the risk of such attacks due to the biochip's easy controllability.
2) An unfaithful biochip designer, who uses fraudulent or falsified claims, is a threat to the users, investors, and regulators. *Edison* microfluidic blood testing device from *Theranos* faced technical, commercial, and legal challenges over the scientific basis of its technologies [19]. Such incidents gather a lot of negative press and hamper the progress in such technologies [20].
3) Studies have flagged security flaws in medical devices such as tampering of controls, denial-of-service, data theft, and ransom attacks [21]. This has lead to a recall of a large number of the medical device and a re-evaluation of their regulations [22]. A biochip cyber-physical system (CPS) is similar to the current medical devices, which consists of hardware, software, and network connections [23]. As biochips are becoming an integral part of the health care services, these threats become more pronounced.

These threats may lead to a loss of revenue and trust or, more importantly, jeopardizes the well-being of its users [24]. They can cause denial-of-service and wrong bioassay outcomes. Addressing these threats becomes even more important as the biochips are being used in artificial-intelligence-based decision making [25] and the emergence of miniaturized versions of oneself for medical tests [26].

B. Biochip Security: Solutions

To increase the trust in such systems, a layer of security needs to be built into the biochip CPS. However, such security measures ought to be applicable in diverse usage scenarios such as in a biomedical lab, and in a remote online/offline device. Sensor-based monitoring is a common defense applied in industrial control system security [33]. Biochip researchers have adopted sensor-based run-time monitoring of certain time-steps at chosen biochip location - referred as *checkpoints*.

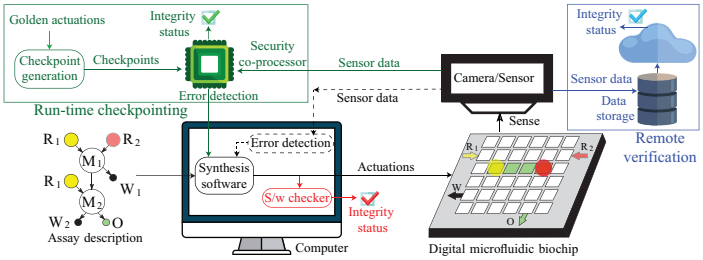


Fig. 2: A DMFB cyberphysical system with multiple defenses. The software checker defense is in red, and the remote verification is in blue. The checkpoint-based defense is in green, which replaces the dotted portion of control loop.

Such measures can complement existing software and network security measures. Table I presents a brief comparison of defense mechanisms in the context of DMFB security.

The state-of-the-art checkpoint-defense uses CCD-camera to monitor the DMFB at run time [24]. The DMFB snapshots are processed to determine the run-time state of the biochip. By comparing the run-time state against the golden state over the entire execution cycles, the bioassay execution is validated (Fig. 2). Most of the DMFB systems have minimal computing resources to minimize cost. Due to the time required for image capture and processing, continuous run-time monitoring of all DMFB cells is not possible. For example, the work in [31] shows that no more than 20 checkpoints can be examined in an execution cycle. This constraint was derived by considering an image pattern matching algorithm implementation on a mid-range ARM Cortex-M3 microcontroller. To overcome this constraint, heuristic defenses based on *checkpoints* are employed [31], [34]. Here, a spatial and temporal subset of the steps executed on the DMFB is sampled and used to compare against the golden specification. Algorithms based on randomized, weighted, and module-less choices have been proposed to derive the checkpoints [31], [34], [35].

C. Problem Statement

As checkpoints cannot monitor *all* possible behaviors of the DMFB, there is room for undetected attacks. To confirm if an attack can evade detection, exact analysis methods are needed. This is lacking in the state-of-the-art [31]. Thus far, it is unclear whether a given checkpoint-based defense strategy thwarts the targeted attacks. Consequently, significant uncertainties with respect to these defenses remain.

This problem can arguably be solved by increasing the computation capacity and incurring higher cost. Nevertheless, a biochip designer should have the tools to weigh the design choices rather than make an uninformed decision to be content with the current low-cost processor or to upgrade to a processor with higher computational ability. We offer such a tool in this work.

D. Contributions

In this work, we are proposing a solution that addresses this problem. To this end, we introduce a symbolic formulation

TABLE I: Comparison of DMFB defense mechanisms.

2*Defense	Tampering attack detection		Error recovery [29]	Application scenario		
	Software [27]	Hardware[28]		Biomedical lab	Online device	Offline device
Software checker [30]	✓			✓	✓	✓
Run-time checkpointing [31]	✓	✓	✓	✓	✓	✓
Remote verification [32]	✓	✓		✓	✓	

representing all possible executions an attacker could conduct on a presumably secured DMFB. Additionally, constraints imposed by a given checkpoint-based defense are enforced. This is used to either

verify that the given defense can thwart an attack on the bioassay implemented on the DMFB,
find an attack plan to evade the defense (if unsafe), or
devise a fool-proof checkpoint mechanism using attack plans as counterexamples.

To reason about this formulation, we use the deductive power of satisfiability solvers. Case studies on practical bioassays confirm the usefulness of the solution. Our experiments demonstrate that the exact analysis can efficiently 1) verify the security of a defense, 2) trade-off different options in the checkpoint-based defense design, and 3) help to devise an effective counter-example guided checkpoint plan. This provides an essential tool which aids biochip designers in ensuring the security of a DMFB.

E. Structure of the Paper

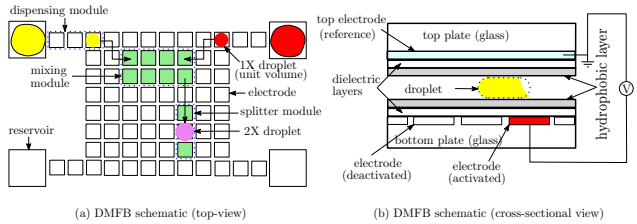
The rest of the paper is organized as follows: In Section II, we review the DMFB systems and the previous work on DMFB security. Based on that, Section III introduces the proposed solution which relies on a symbolic formulation of the considered problem describing all the attacker capabilities as satisfiability constraints. Section IV demonstrates the applications of the proposed solution. Section IV-B assesses the security risks of existing defense mechanisms by means of practically relevant bioassays; Section IV-C outlines a counterexample-guided secure checkpoint-generation mechanism and provides experimental results. We then discuss some of the highlights of our work in a Q & A format in Section V and, finally, conclude the paper in Section VI.

II. Background

In this section, we review the core concepts of DMFBs, describe the DMFB CPS, and survey the DMFB market. Further, we also describe the previous work on DMFB security. Note that, in the rest of the paper, we use an abstract framework with respect to droplet movement based on previous work on synthesis, placement, or routing for digital microfluidic biochips [36], [37], [38].

A. Digital Microfluidic Biochip

A 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 plates. To reduce the sample evaporation, contamination, and facilitate droplet operations, a DMFB is filled with silicone oil between the two plates, and the sample droplets are immersed in an oil medium. Fig. 3(a) and Fig. 3(b) show the top view and cross-sectional view of the DMFB, respectively. The DMFB manipulates fluids in discrete quantities based on the "electrowetting on dielectric" (EWOD) principle: the control of the contact angle between a droplet and substrate by applying a suitable electric potential [39]. Voltages are applied to the electrodes (also called DMFB cells) to manipulate the wetting forces on the droplet. Droplets are attracted to the neighboring electrodes with higher voltages. This results in a controlled movement of droplets in the horizontal and vertical directions on the DMFB. In the DMFB design flow, the sequence of operations corresponding to the bioassay (represented as a directed acyclic graph) is converted into a timed-sequence of microfluidic operations, as shown in the following example.



(a) DMFB schematic (top-view) (b) DMFB schematic (cross-sectional view)

Fig. 3: DMFB schematic (a) top and (b) cross-sectional view.

Example 1. *Fig. 4(a) shows a bioassay which mixes reagents R_1 and R_2 to dispense an output O with mix ratio $R_1 : R_2 = 3 : 1$. The snapshot of the execution sketched in Fig. 4(c) shows that R_1 and R_2 are mixed and split in a $1 : 1$ ratio. Afterward, the droplet W_1 is discarded while the other droplet is mixed and split again with R_1 (see Fig. 4(e)). The final droplet has a mix ratio of $3 : 1$, and is dispensed at port O (see Fig. 4(f)).*

B. DMFB Cyber-physical System

A fully integrated DMFB consists of a controller, sensor feedback, and network connection [40]. A bioassay description (represented as a sequencing graph [41]) is synthesized to an actuation sequence which realizes the bioassay through various fluidic operations on the DMFB. However, the fluidic operations are susceptible to various manufacturing imperfections, which can lead to run-time faults. To detect such faults and for error recovery in the DMFB operations [42], run-time monitoring through sensor feedback is required. CCD-camera and/or capacitive sensors are used to monitor a droplet location

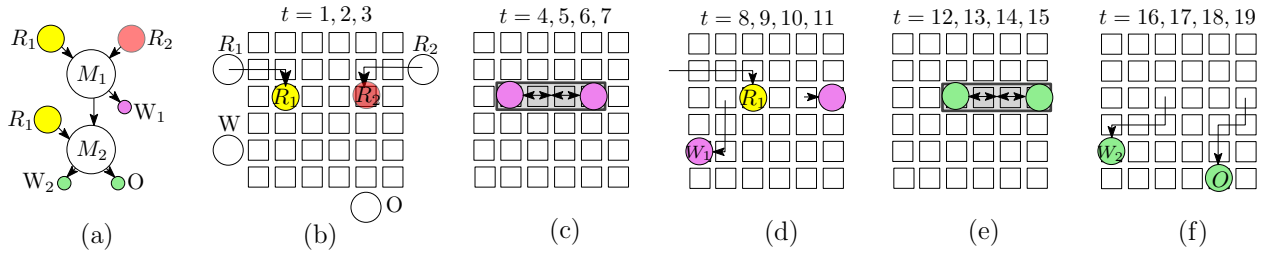


Fig. 4: (a) Directed acyclic graph (DAG) of a bioassay. (b)-(f) Implementation of the bioassay on 6 6 DMFB over t = 1, 2, , 19 cycles.

and size on the DMFB [29] CCD cameras are more popular due to their precision [29]. The image is cropped into sub-images to focus on an area-of-interest (e.g., an electrode). These sub-images are correlated with a template to monitor the droplet occupancy and size at the desired locations. The DMFB can also be connected to a network for run-time monitoring of assay operations, result analysis, and software update.

C. DMFB Market

The global biochip market is anticipated to exhibit significant growth in the near future [12]. As the market for DMFB grows, a cost-effective production process will be in high demand. Similar to the integrated circuit (IC) supply chain, outsourcing of the DMFB fabrication will be an attractive alternative. Wherein third-party IPs, CAD software, and fabrication facilities will be used. However, the horizontal supply chain is susceptible to various threats similar to the case of ICs. Prior work shows that the DMFB supply chain is susceptible to Trojan insertion [27], IP piracy [43], [44], actuation tampering [24], miscalibration [27], and counterfeiting [45].

D. Threat Models

Given that DMFBs are targeted for a variety of safety-critical applications such as diagnosis (at home) and drug development (in a lab) [46], [5], security and trust issues are of paramount importance. In previous work on DMFB security, insider threats and network-based threats have been modelled. In the following, we describe these threat models that depend on the different biochip usage scenarios.

1) Insider threat model: An attacker is a disgruntled employee, who is motivated by jealousy towards co-workers or anger against the lab management [47]. Such an attacker has access to the biochip controller and actuators. The attacker can tamper with the control software or induce faults in the controller or actuators using electrical probes or lasers [48]. The objective of the attacker is to hamper the research of individual co-workers or overall lab [18], [17]. Here, the lab management is the defender and the attacker needs to overcome access checks in place.

2) Outsider threat model: A remote attacker can be a competitor seeking to bring disrepute to the biochip designer [49]. The attacker can use malware to gain control of the network and to manipulate the control software or stored actuation sequence [50]. Alternatively, rogue elements in the biochip supply chain can tamper with the design to insert Trojans. The attacker can exploit the in-built hardware/software Trojan to access the biochip controller [31]. The attacker manipulates the results of the biochip in a stealthy and untraceable way. To do this, the attacker has to evade detection by the sensors. The defender can monitor the biochip using a CCD camera and capacitive sensors [29], [51].

E. Attacks

The attacks manifest as either addition or modification of microfluidic operations [52]. In this work, we focus on the following attacks as they are a prerequisite to the manipulation of bioassay results.

Proximity attack: droplets can be contaminated by modifying the droplet routes to perform an unspecified-merge [53].

Swap attack: two or more droplets are swapped [27], [53]. One such case is discussed later in Example 2 and Fig. 5.

Dispense attack: malicious droplets are dispensed to disturb a bioassay implementation [31].

Parameter attack: attacker tampers with mix/incubation time [27].

F. Defenses

To defend against the above attacks, security features such as software checkers, remote (off-system) verification and run-time checkpointing can be used. Here, we briefly describe these security measures and discuss there pros and cons.

1) Software checkers: An actuation tampering can also be detected by crosschecking the actuation sequence before loading it to the DMFB. This can be done using software checksum. However, software checkers cannot detect hardware fault injection attacks [31], [27], as software checkers are oblivious to changes down the control path. Other software-based defense such as encryption techniques used for software integrity can be undone as the actuation sequence has to be decrypted before applying it to the DMFB. This leaves the decrypted actuations susceptible to tampering.

2) Remote (off-system) verification: Recall that the DMFBs can be equipped with sensors that allow us to monitor its state. The sensor data of the biochip can be captured and stored, which can be validated later. This mechanism cannot support run-time error recovery as the sensor results are not processed for run-time fault detection. This makes the product susceptible to faults arising due to natural variation and hardware imperfections. In other words, run-time fault detection is indispensable in a biochip system [29] and remote verification does not make use of it.

3) Run-time checkpointing: The defender generates the checkpoints and stores it in a security co-processor for run time validation. The security co-processor is not connected to the network and is physically separated from the biochip controller. Any updates to the security co-processor are done offline by the defender by writing to the flash memory through a JTAG connection. This prevents the attacker from compromising both the bioassay execution and the defense [31].

The checkpointing defense simply leverages the sensor result processing being used for run-time fault detection. Due to its minimal overhead and full visibility of the control path, run-time checkpointing can act as a defense in diverse attack scenarios, as shown in Table I.

G. Checkpoint Verification

In the previous work, the aim was to choose the checkpoints such that they provide enough coverage to make it difficult (ideally, impossible) for an attacker to tamper with the DMFB implementation [31]. As a result, these defenses are only "probably-secure". We show through Example 2 that an attacker can evade such checkpointing.

Example 2. *The bioassay in Fig. 4 can be compromised by actuation tampering. The tampered actuation alters the route of droplets R_1 and W_1 between time steps $t = 9$ and $t = 11$ to contaminate the droplet R_1 before the mix-split operation M_2. This yields a droplet with a different mixing ratio instead of $R_1 : R_2 = 3 : 1$, as shown in Fig. 5. If the DMFB snapshots are not monitored in time-steps $t = 9$ and $t = 10$ (Fig. 5(e)-(f)), the attack evades detection.*

Existing solutions have a spectrum range from "provably-secure-defense" that overshoots the available sensing resources to "probably-secure-defense" that works with the available sensing resources. Provably-secure-defense guarantees the valid bioassay execution on the DMFB [54]. However, this requires additional resources in terms of integrated sensors to monitor all the droplets at all the time-steps. On the other hand, probably-secure-defenses include randomized checkpointing which checks random cells at random time-steps as well as static checkpointing which checks the cells in the neighborhood of the droplet paths [31], [24]. These defenses only provide probabilistic guarantees against attacks. They do not consider all possible maneuvers an attacker can perform to avoid the checkpoints [53]. This opens the door for smart manipulations by an attacker to escape monitoring.

III. Proposed Exact Analysis Method

Given the large space of bioassay designs and attacker's manipulation ability, it is not clear how secure the defenses are when applied to an arbitrary bioassay. Hence, there is a need to develop an exact methodology to analyze whether a given defense based on checkpoints indeed prevents the execution of an attack. In this work, we first propose a methodology to determine if an actuation sequence can execute an attack without being detected by any checkpoint. We do so by considering *all* possible actuation sequences by means of a *symbolic formulation*. Using the symbolic formulation, we check whether at least one sequence exists which

1) can be implemented on the DMFB,
2) matches the original bioassay at all checkpoints, and
3) attacks in the time-step not covered by the checkpoints.

If such an attack is possible, the symbolic formulation yields an attack plan explicitly showing how the defense can be compromised. If no such sequence exists, the defense has been proven to be secure. Since resolving the proposed symbolic formulation is a complex task (eventually, this requires the consideration of all possible actuation sequences), we utilize the deductive power of satisfiability solvers for this purpose. They already have been proven successfull in the design process for different biochip platforms (see, e.g., [55], [56], [57], [58]) as well as for the validation and verification of biochip designs (see, e.g., [59], [60]).

In this section, we outline the formulation and illustrate how this formulation can be solved using satisfiability solvers. Eventually, this allows assessing the strength of a checkpoint-based defense against attacks on a DMFB (Section IV-B). Moreover, this formulation can be used by the designer to devise a secure checkpointing against a given set of attack models (Section IV-C).

Note that we are addressing the security challenge in this work for an ideal biochip system - where the implementation is predictable and fault-free [61]. This simplifies our analysis and allows us to focus only on the security issues. Further, we do not consider the use of conditional loops in the bioassay [62], [63]. The extension of our solution to more complex realistic conditions of uncertain and fault-prone fluidic response is left for future work. To this end, we discuss future directions and challenges in Section VI.

A. Symbolic Formulation

Consider the bioassay DAG synthesized to the actuation sequence for the target $r \quad c$ DMFB. The bioassay requires T time-steps to finish. We decompose the input bioassay into several edge-disjoint "input to output/waste"-paths. Each path is a droplet trajectory that appears on the DMFB from an input reservoir (by dispense operation) and takes part in one or more operations (e.g., mixing/detection) and dispensed to a waste/output reservoirs. For example, we can decompose the sequencing graph from Fig. 4(a) into three edge-disjoint paths: $(R_1 \; ! \; M_1 \; ! \; M_2 \; ! \; O)$, $(R_2 \; ! \; M_1 \; ! \; W_1)$, and $(R_1 \; ! \; M_2 \; ! \; W_2)$. Finally, we assign a unique identifier to each droplet and assume that the droplets appearing on the

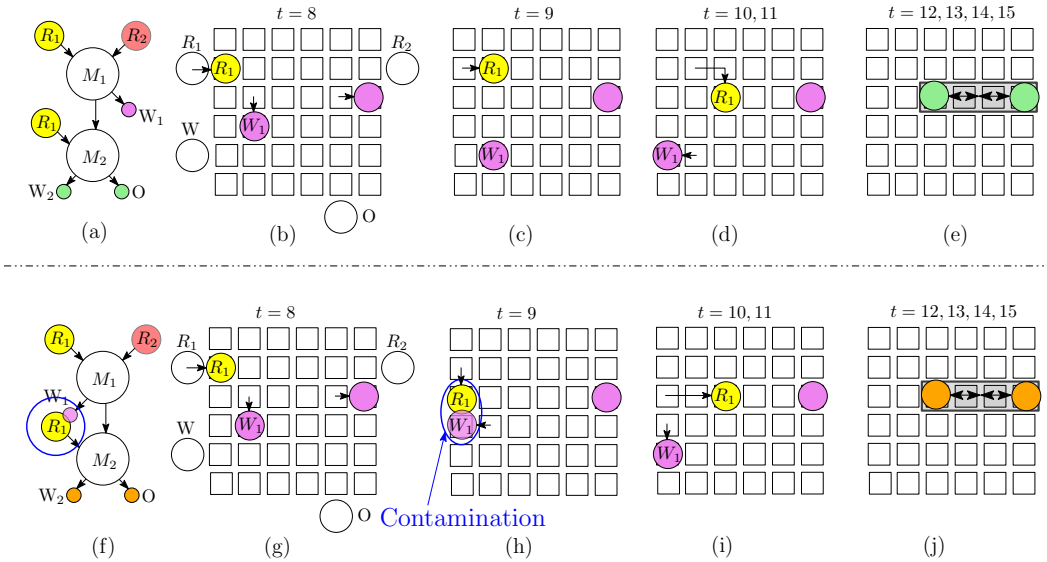


Fig. 5: (a) Directed acyclic graph (DAG) of the bioassay. (b)-(e) Snapshots of the golden execution during time step t = 8, 9, , 15. (f) Modified DAG due to the actuation tampering. (g)-(j) Proximity attack in progress where the routing paths of droplets W_1, R_1 are modified between time step t = 9 and t = 11.

DMFB (i.e., their identifiers) are stored in a set D. Then, the following variables are used to describe all possible sequences (for 1 x r, 1 y c, d 2 D, and 0 t T):

$$a^t_{x,y,d} = \begin{cases} 1, & \text{if a droplet } d \text{ appears on a DMFB cell } (x,y) \\ & \text{at time } t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Furthermore, variables are introduced for input/output operations (for 1 x r, 1 y c, and 0 t T):

$$ip^t_{x,y} = \begin{cases} 1, & \text{if a droplet is dispensed on a DMFB cell } (x,y) \\ & \text{at time } t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$op^t_{x,y} = \begin{cases} 1, & \text{if a droplet disappears from a DMFB cell } (x,y) \\ & \text{at time } t \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

B. Ensure Valid DMFB Execution

Solving this symbolic formulation (with free variables only) will admit arbitrary solutions and, hence, arbitrary actuation sequences. But sequences that violate obvious consistency constraints (e.g., a droplet suddenly appears in one time-step and disappears in the next.) have to be precluded. The following constraints are added to admit solutions that can be implemented on the DMFB.

(A) At time-step t, a droplet d 2 D may appear in at most one DMFB cell:

$$\bigwedge_{d2D} \bigwedge_{t=1}^{T} \left(\sum_{x,y} a^t_{x,y,d} \quad 1 \right) \tag{4}$$

(B) Each DMFB cell contains only one droplet in any time-step. The attacker can launch malicious mix operation by transporting two droplets to a single cell[1].

$$\bigwedge_{t=1}^{T} \bigwedge_{x=1}^{r} \bigwedge_{y=1}^{c} \left(\sum_{d2D} a^t_{x,y,d} \quad 2 \right) \tag{5}$$

(C) The movements of droplets on the DMFB have to satisfy the following constraints. If a droplet d 2 D is on a cell (x,y) at time-step t (i.e., $a^t_{x,y,d}$ = 1), then

(a) either d was on the same cell (x,y), or on one of its four neighbors (denoted by $N_4(x,y)$) in time-step t 1,

(b) or d is next to a dispenser creating the droplet on (x,y) at time-step t (this only needs to be described for locations (x,y), where droplets can be dispensed).

$$a^t_{x,y,d} =) \underbrace{\left(\bigvee_{\substack{(x^\theta,y^\theta)2 \\ N_4(x,y)\,[\,f(x,y)g}} a^{t\ 1}_{x^\theta,y^\theta,d} \right)}_{(a)} - \underbrace{\bigvee ip^t_{x,y}}_{(b)} \tag{6}$$

(D) Droplets may disappear when they leave the DMFB through a sink. Hence, if a cell (x,y) was occupied by d 2 D at time-step t 1, which is not present on its current location (x,y) or the neighborhood $N_4(x,y)$ at

[1]Since droplets greater than 2 cannot be moved, we ignore them [64].

time-step t, then it must go to the sink on (x, y).

$$a^t_{x,y,d} \wedge : \left(\bigvee_{\substack{(x^\theta,y^\theta) 2 \\ N_4(x,y) [f(x,y) g}} a^{t\ 1}_{x^\theta,y^\theta,d} \right) =) \quad op^t_{x,y} \quad (7)$$

C. Ensure Expected Behavior at Checkpoints

In time-steps designated as checkpoints, the original behavior has to be maintained. This is ensured by adding the following constraints:

(A) We know the time-steps when droplets appear on the grid from an input reservoir. Let the k droplets $D_{(x,y)} = f d_1, d_2, \quad , d_k g \quad D$ be dispensed on (x, y) at time-steps $t_{(x,y)} = f t_1, t_2, \quad , t_k g$, where d_i is dispensed at t_i, for $i = 1, 2, \quad , k$. The next constraint enforces correct dispensing for the input reservoir that dispenses droplets on location (x, y):

$$\bigwedge_{t 2 t_{(x,y)}} ip^t_{x,y} \wedge \bigwedge_{t 2 \not t_{(x,y)}} : ip^t_{x,y} \quad (8)$$

(B) Ensure the correct number of droplets disappear at each output reservoir. This prevents attacks due to the extra droplet [31].

(C) The designer places checkpoints to detect the presence or absence of a droplet. Let CP be the set of checkpoints. Each element in the CP is of the form (x, y, t, d_{size}), i.e., a droplet of size d_{size} 2 f0, 1, 2g must appear on (x, y) at t.

$$\bigwedge_{(x,y,t,d_{size}) 2 CP} \left(\sum_{d 2 D} a^t_{x,y,d} = d_{size} \right) \quad (9)$$

D. Enforce an Attack

In this section, we model attack behaviors reviewed in Section II-D.

Deviation attack: Let each droplet d 2 D have a defined location (x, y) for each time t 2 T. This attack is modeled as follows.

$$\bigvee_{d 2 D, t 2 T} (: a^t_{x,y,d}) \quad (10)$$

This constraint captures any deviation from the synthesized bioassay. However, not all deviations from the golden actuation sequence lead to incorrect assay execution.

In the remaining part, we describe two classes of attacks, namely swap and proximity attacks (however, several other attacks can be modeled similarly). Without loss of generality, let us assume the DMFB supports dispense, transport, balanced mixing, and splitting. The proposed exact analysis ensures correct behavior at dispense operations by enforcing constraint (8) i.e., thwarts malicious dispense operations [31]. Moreover, if an attacker splits an unit-sized droplet into two halves, the child droplets cannot be transported further [43]. This malicious behavior, i.e., undesired splitting, can be detected by checkpoints placed on the droplet trajectory. Therefore, a split operation is not meaningful without a prior mix. The proximity attack, i.e., undesired droplets coming to close to each other, models the possibility of contamination and malicious (extra) mix operations. Further, the attacker can change the droplets in a mixing operation, as modeled by the swap. By ensuring that the given checkpoint thwarts these attacks, a correct assay execution is guaranteed. More precisely:

Swap attack: After dispensing from an input reservoir, a droplet can mix with one or more droplets before going to an output/waste reservoir. The swap attack swaps one of the input droplets of a mixing operation with an undesired droplet – or it transports a droplet to the wrong output location corrupting the bioassay output. In order to verify the feasibility of a swap attack, we need to check whether there exists an implementation of the DAG that satisfies the checkpoints but alters an input of the mixing operations or dispenses a droplet into the wrong output reservoir.

Suppose droplet d 2 D is expected to be at the location (x, y) at time t. The swap attack is possible if d can be replaced by any other droplet d^0 2 D n fdg without being detected by the checkpoint-based defense. The droplet d either can be an input droplet in the mixing operations (M) or an output droplet. M is the set of all mixing operations, where each mixing operation is represented as $[(x_1, y_1, d_1, t_s), (x_2, y_2, d_2, t_s), t_{mix}, M_{type}]$ i.e., two 1 droplets d_1 and d_2 come to the locations (x_1, y_1) and (x_2, y_2), respectively, at time-step t_s and after mixing using M_{type} 2 f1 4, 4 1, 2 4, g during the next t_{mix} consecutive mixing cycles. The resulting two 1 droplets (after balanced splitting) must come to the locations (x_1, y_1) and (x_2, y_2) at time-step $t_s + t_{mix}$. Similarly, the set of output operations $O = f(x_1, y_1, d_1, t_1), (x_2, y_2, d_2, t_2), \quad , (x_k, y_k, d_k, t_k) g$ denotes the droplet dispense locations and time. The swap attack is modeled by the next clause plus the checkpoint clause from earlier.

$$\bigvee_{(x,y,d,t) 2 M [O} (: a^t_{x,y,d}) \quad (11)$$

Proximity attack: For the droplet located on cell (x, y) at time-step t, another droplet must come to any of the 8-neighboring cells of (x, y) (denoted $(N_8(x, y))$. This is formulated as follows.

$$proximity^t_{x,y} \ , \ \left(\left(\sum_{d 2 D} a^t_{x,y,d} = 1 \right) =) \bigwedge_{\substack{d 2 D, \\ (x^\theta,y^\theta) 2 \\ N_8(x,y)}} : a^t_{x^\theta,y^\theta,d} \right)$$

$$\text{proximity}^t \ , \ \left(\bigvee_{\substack{1 \ x \ r, \\ 1 \ y \ c}} : \text{proximity}^t_{x,y} \right)$$

The Boolean variable $proximity^t_{x,y}$ is true if and only if no droplet is present in any of the eight neighbors of the location (x, y) at time instant t. Analogously, the variable $proximity^t$

Algorithm 1: *IsAttackResilient*(A, T, CP)

Input: A: Actuation sequence of the bioassay; T: assay time; CP: Set of checkpoints.
Output: Yes, if CP is attack resilient, otherwise, an attack as a counter-example.

Reverse engineer A to extract the DAG (G) corresponding to the bioassay [32], [65] and other bookkeeping information such as dispense locations and mixer information.;
Decompose G into edge-disjoint paths and assign a unique identifier to each path.;
`/* Detailed modeling -- variables. */`
M := Add variables as defined in Eqns. (1)-(3).;
`/* Detailed modeling -- constraints. */`
`/* Add constraints for modeling valid DMFB operations. */`
M := Add constraints as given in Eqns. (4)-(7).;
`/* Add constraints for checkpoints. */`
M := Add constraints for enforcing input and output operations (Section III-C);
for *each checkpoint* $(x, y, t, d_{\text{size}})$ 2 CP **do**
  M := Add constraints as given in Eqn. (9).;
`/* Add constraints for attacks. */`
M := Add constraints for an attack such as proximity attack (Eqn. (12)), swap attack (Eqn. (11)), or deviation attack (Eqn. (10)).;
if M *is unsatisfiable* **then**
  return CP on A is attack resilient.;
else
  return Satisfiable assignments as an attack on A.;

is true if and only if there is a proximity attack at time-step t. Using these two Boolean variables, the following constraint enforces the proximity attack for a bioassay that requires T time-steps to finish.

$$\bigvee_{t=1,2,\quad,T} \text{proximity}^t \tag{12}$$

E. Exact Analysis of a Checkpoint

The overall flow for the exact analysis of checkpoint-based defenses is summarized in Algorithm 1. The symbolic formulation represents arbitrary behavior on the considered DMFB. Adding the constraints which ensure a valid DMFB execution and the expected behavior at the checkpoints narrows the possible solutions which represent the "expected behavior". If adding the attack constraints leaves at least one satisfying solution, then the defense is not secure. To resolve the formulation, we use satisfiability solvers (such as [66], [67], [68]). If the solver returns a satisfying assignment to all $a^t_{x,y,d}$, $ip^t_{x,y}$, and $op^t_{x,y}$-variables give the positions of all droplets, of all dispense operations, and of all disappearances of droplets, respectively explicitly describing the plan for the attack. If the solver fails to return a satisfying solution, this is proof that there is no viable attack plan and, hence, the defense is secure. The following example shows how the exact analysis detects security vulnerabilities by returning an attack pathway as a counter-example.

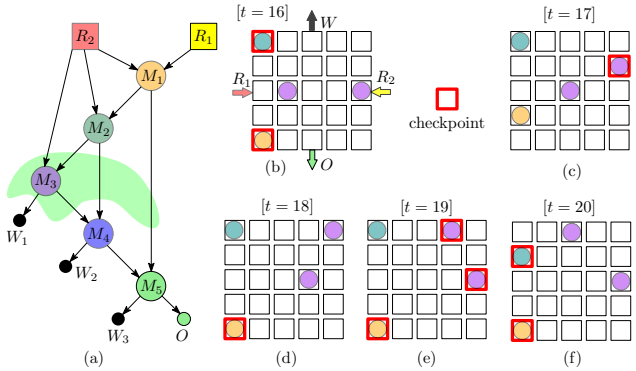


Fig. 6: (a) DAG for mixing reagents R_1 and R_2 using *REMIA* [69]. (b-e) DMFB snapshots that implement the highlighted portion of the DAG: (b) Snapshot after the mixing M_3 at time-step t = 16, (c)-(f) droplet routing operations for subsequent assay operations. The checkpoints for monitoring are shown.

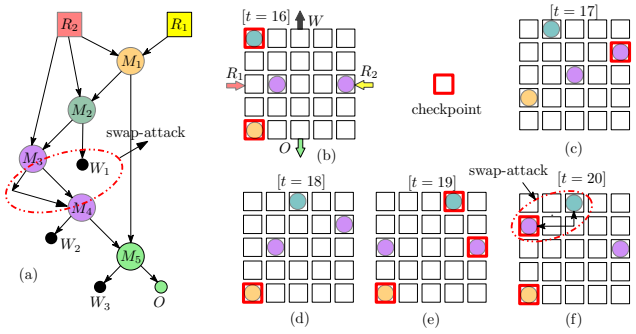


Fig. 7: (a) Modified DAG due to a swap attack. (b)-(f) Swap-attack between the violet and green droplet (t = 20) is returned as a counter-example to prove that the checkpoints are incorrectly placed.

Example 3. *Let us consider the DAG for mixing two input reagents using* REMIA *[69] as shown in Fig. 6(a). This bioassay implementation on a* 5 5 *DMFB takes* T = 40 *time-steps. Figs. 6(b)-(f) show the snapshots of the DMFB for time-steps* t = 16, 17, , 20. *Checkpoints are incorporated by the designer as a defense and are highlighted on the DMFB grid in each time-step in Fig. 6. For these set of checkpoints, the proposed exact analysis shows that two droplets can be swapped from time-step* t = 16 *to* t = 20, *as shown in Figs. 7(b)-(f). The resultant modified bioassay is shown in Fig. 7(a).*

Suppose the designer incorporates a different set of checkpoints, as highlighted in Figs. 8(a)-(e). Here, the cell (5, 1) *is not monitored at* t = 19 *compared to the checkpoint in Fig. 6. Now, the exact analysis shows that a proximity attack can be performed. It provides a pathway to the proximity attack that contaminates two droplets at time-step* t = 19 *as a counter-example (Fig. 8(d)).*

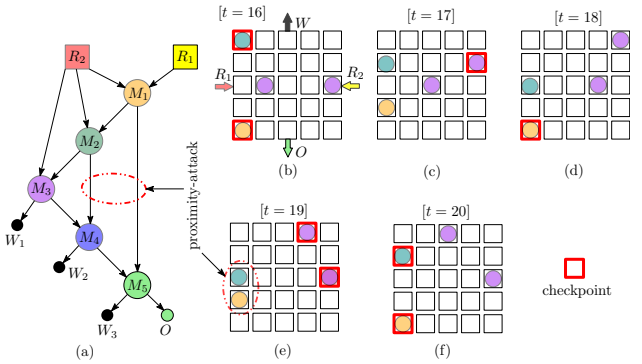


Fig. 8: (a) DAG showing the contamination attack. (b)-(f) Contamination-attack (at $t = 19$) is returned as a counter-example to prove that the checkpoints are incorrectly placed.

TABLE II: Considered bioassays from [41].

Assay	DMFB size	#Droplets	#I/O-ports	Assay time (T)	#Mix-Split
REMIA	5 5	4	4	40	5
Linear gradient	9 8	6	5	40	6
PCR mix	8 15	8	10	70	7
PCR stream	8 15	13	11	73	15

IV. Application of Exact Analysis

The exact analysis has been implemented in Python 2.7 on an Intel Core-i7 machine. We use Z3 [68] to solve the resulting instance. In this section, we apply the exact method to practical bioassays and show that it can verify the checkpoint-based defenses. Moreover, we also show how the proposed exact analysis can be used to derive a counter-example guided secure checkpointing determination method.

A. Considered Bioassays and Defense Strategies

We use the following set of DMFB-implementations of bioassays (taken from [41] and representing practical use cases) to illustrate the importance of exact analysis in checkpointing:

REMIA: a sample preparation scheme that minimizes reagent usage for a given target concentration.
Linear gradient: a sample preparation scheme used to optimally dilute a sample in a linear gradient while minimizing wastage.
PCR mix: a polymerase chain reaction (PCR), which is used for DNA amplification and involves mixing of seven fluids in a desired ratio.
PCR stream: another PCR which is optimized to mix for multiple droplet generation as demanded by the application.

For each bioassay, a corresponding DMFB-implementation is derived and summarized in Table II.

Next, we applied a checkpoint (CP) defense, wherein all the droplets and input ports are checked at each time-step. This is referred to as *baseline defense*. The designer can derive multiple variants of baseline defenses by exercising the following options:

Reduce the number of cells checked at each time-step.
Increase the time interval between the checkpoints.

For the experiments considered here, several plausible decisions have been taken to this effect. These decisions are shown in the second and third column of Table III and Table IV.

B. Exact Analysis of Considered Defenses

In the following, we used the proposed exact analysis to evaluate the resulting CP strategies and analyze their security against swap, proximity, and deviation attack as described in Section II-E. The current state of the art does not yet provide an exact analysis to verify whether those CP decisions indeed yield a secure chip. Through the method proposed in this paper, we can now conduct such an analysis.

Tables III and IV summarize the results obtained when exploring two variants of baseline defenses. More precisely, for each attack (Swap, Proximity, and Deviation) as well as for each considered number of monitored cells and checkpoint interval, it is listed whether the proposed analysis determines the defense as secure (denoted by P for pass) or insecure (denoted by F for fail). The first row for each bioassay (highlighted yellow) lists the results obtained for the defense where all the droplets are checked at each time-steps. However, the following rows list the results for the variants of the baseline defenses. Finally, the total run time for conducting the checks is provided in the final column.

The results show that, for the first time, the proposed solution can be used as an oracle to evaluate the various options for the checkpoint-based defense design. This does not only allow to verify whether a particular defense is secure but also to trade-off, e.g., between "lowering the number of cells per checkpoint" versus "increasing the interval between checkpoints". The results suggest that the former is better compared to the latter. Further, the proposed method can be used by the designer for red-teaming against the defense, i.e., to learn from the counter-example of a defense failure. The results show that exact security could be achieved by checking a lesser number of cells than the baseline solution, as shown by the *linear gradient* and *PCR mix* assays in Table III.

One key observation from the results (Table III and Table IV) is that - *it is possible to guarantee the functional correctness of bioassay execution without monitoring all the electrodes at all time-step.* In other words, it is possible to defend against swap and contamination attacks, thereby safeguarding the integrity of bioassay implementation. Note that without swap or contamination attack, the deviation attack does not violate the correctness of a bioassay.

C. Counterexample-guided Determination of Checkpoints

Motivated by the results in Section IV-B, we additionally derived a secure CP methodology which 1) does not need to monitor all the electrodes at all time-steps and 2) is proved to

TABLE III: Verify defense by varying the #cells checked at CP.

2*Assay	CP interval	Cells /cycle	Secure? [Pass (P) / Fail (F)] Swap	Proximity	Deviation	Run time
3*REMIA	3*1	4	P	P	P	0.21 s
		3	P	F	F	0.23 s
		2	F	F	F	0.3 s
4*Linear gradient	4*1	6	P	P	P	1.50 s
		5	P	P	F	1.3 s
		4	P	F	F	1.34 s
		3	F	F	F	1.02 s
4*PCR mix	4*1	8	P	P	P	7.78 s
		7	P	P	F	7.03 s
		6	P	F	F	7.46 s
		5	F	F	F	5.5 s
4*PCR stream*	4*1	11	P	P	P	15.95 s
		10	P	F	F	15.93 s
		9	P	F	F	12.87 s
		8	F	F	F	12.2 s

In PCR stream the maximum number of droplets at a given cycle is 11.

TABLE IV: Verify defense by varying the interval between CPs.

2*Assay	Cells / cycle	CP interval	Secure? [Pass (P) / Fail (F)] Swap	Proximity	Deviation	Run time
3*REMIA	3*4	1	P	P	P	0.21 s
		2	P	F	F	0.15 s
		3	F	F	F	0.16 s
3*Linear gradient	3*6	1	P	P	P	1.50 s
		2	P	F	F	1.23 s
		3	F	F	F	1.37 s
2*PCR mix	2*8	1	P	P	P	7.78 s
		2	F	F	F	5.07 s
3*PCR stream*	3*11	1	P	P	P	15.95 s
		2	P	F	F	16.16 s
		3	P	F	F	16.16 s

In PCR stream the maximum number of droplets at a given cycle is 11.

be secure. Recall that at each checkpoint, the CCD camera is used to capture a snapshot of the DMFB at run time. Then, the image is cropped into sub-images to extract the DMFB cell-of-interest. This sub-image is then correlated with a template of expected images which has predetermined information about droplet occupancy and size. Note that the number of time-steps covered by the CP strategy determines the number of times the CCD camera needs to capture the DMFB snapshots. Further, the number of image processing operations (sub-image extraction and correlation) is determined by the number of cells being in each of the CP time-step.

In the proposed checkpointing, our objective is to determine a set of time-steps such that if all the droplet locations are monitored in these time-steps, then it ensures secure execution of the bioassay. We initialize the CP with the time-steps where mixing operations start. After that, we invoke exact analysis to check the security of the bioassay execution against different attacks for the given set of checkpoints CP. If the CP cannot secure the execution, the oracle returns a counterexample. We use this counterexample to analyze the point of defense

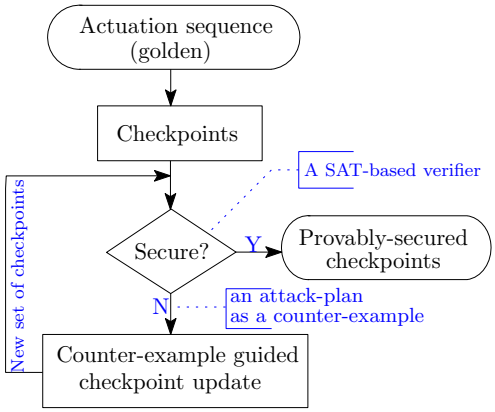


Fig. 9: Overview of the proposed method.

failure and update the CP accordingly. The query and CP update continues until the CP guarantees the security of the bioassay execution. Fig. 9 shows the overall scheme of a counterexample-guided secure checkpointing.

Algorithm 2: *CP-Time-Steps*(A, T)

Input: A: Actuation sequence of the bioassay; T: assay time;
Output: CP: Set of checkpoint time-steps.
/* Initializing CP */
CP = start time t of all mixing operations in the assay;
/* loop until bioassay is secure. */
while *true* **do**
  /* Generate counterexample CE */
  CE = *IsAttackResilient*(A, T, CP);
  if $CE == \phi$ **then**
    return CP;
  else
    /* Find droplets that cause verification failure */
    bad_droplet = [];
    for *each droplet* d **do**
      for *each assay operation* op **do**
        if d *does not participate in* op *or* d *is contaminated* **then**
          Add (d, op, t) to bad_droplet
    /* Update CP list */
    for (d, op, t) *in bad_droplet* **do**
      Backtrace first deviation point t_1 for d leading to op at t.;
      Add t_1 to CP;
return CP ;

1) Checkpoint Time-step Derivation: Recall that in a bioassay, a droplet lifetime starts with a dispense operation, then it participates in assay operations such as mixing, incubation, and finally dispensed to the output/waste reservoir. We use the exact analysis oracle to verify if each droplet meets the expected behavior in each checkpoint (CP), i.e., each droplet participates in the desired assay operations and does not contaminate other droplets. The verification fails either due to a droplet d not participating in an operation op at time-step t or droplet d contaminating another droplet d^0 at time-step t. In such a case, the exact analysis oracle returns a counter-example. Then, we back-trace the droplet d's route from time t

and place a new checkpoint at time-step t_1 where the droplet d begins to deviate from its specified route, leading to the verification failure at time-step t.

After updating the CP list, the oracle is run again to verify the security. Here, we use the incremental solving ability of the Z3 solver to reduce the run time overhead between successive calls of the oracle. Wherein the constraints for modified checkpoints are added as new constraints. This helps Z3 solver to decide the truth value of the instances quickly. This updating of CP and incremental solving is repeated until the coverage of the CP list is large enough to secure the bioassay execution. The pseudo-code for CP methodology guided by the counterexample is shown in Algorithm 2. In the following, respective details are described:

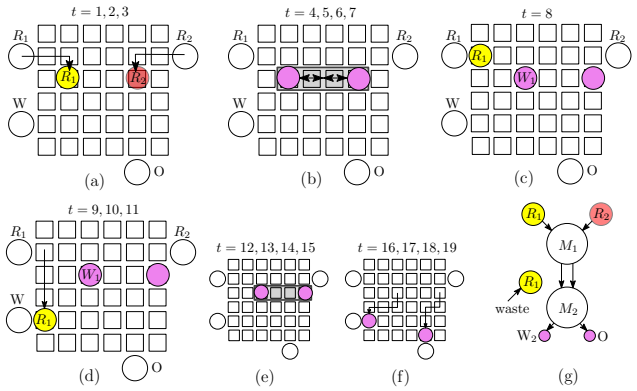


Fig. 10: Actuation tampering. (a)-(f) Swap attack in progress where droplets W_1, R_1 are swapped between $t = 8$ and $t = 10$. (g) Modified DAG due to the attack.

Example 4. *Consider the bioassay shown in Fig. 4. The CP is initialized with the start time-steps ($t = 4, 12$) of two mix-split operations. With this set of CP, the exact analysis returns a counterexample shown in Fig. 10. Here, droplet R_1 and W_1 are swapped, as shown in Fig. 10(d). Algorithm 2 detects these are bad droplets and backtracks their paths. The first deviation of droplet R_1 (W_1) droplets is at time-step $t = 9$ ($t = 10$). Time-steps $t = 9$ and $t = 10$ are added to the CP and the exact analysis continued. The updating of CP is continued until the exact analysis returns the status of CP as safe.*

2) Local Minimization: We next minimize the number of droplets monitored in a CP time-step by doing a local search. Let a droplet monitored in a CP time-step is in a sparse area of the biochip. Such droplet can be safely dropped from the monitoring (CP) list; if the time interval between current CP time-step (t) and previous CP time-step (t^0) is smaller than the Manhattan distance between itself at t and its nearest neighbor at t^0. We use this observation to minimize the number of droplets monitored in a CP time-step. We drop a droplet d in CP time-step t 2 CP, if the Manhattan distance between droplet d at time-step t and its nearest neighbour d^0 in the previous CP time-step t^0 is greater than the time difference between the time-step ($t \quad t^0 + c$). Where $c > 0$ is a constant, that provides a guardband. This way the droplets that are far from the other droplets are safely dropped from the CP list.

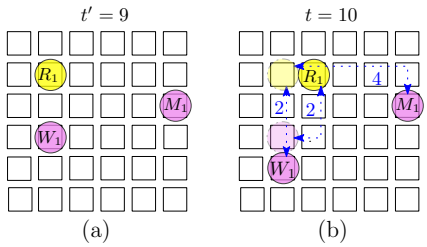


Fig. 11: Snapshots of DMFB at (a) previous CP time-step $t^0 = 9$ and (b) current CP time-step $t = 10$. Manhattan distance between each droplet at $t = 10$ from its nearest neighbor at $t^0 = 9$ is shown.

This minimizes the number of cropping of sub-images and correlation performed in a CP time-step.

Example 5. *Consider the case of Example 4, wherein time-step $t = 9, 10$ are added to CP by the counterexample-guided checkpoint update routine. Here, M_1 droplet at $t = 10$ is farther from the other two droplets at previous CP time-step ($t^0 = 9$) by at least three steps (Manhattan distance), as shown in Fig. 11(c). The time difference between the time-steps is $t \quad t^0 = 1$. We choose $c = 1$, to avoid the possibility of droplets coming in the neighboring cell. This means that the droplet M_1 at $t = 10$ satisfies our sparsity condition. Therefore, the droplet M_1 at $t = 10$ is dropped from CP. This avoids the computation of the image correlation of one electrode. However, the droplet R_1 at $t = 10$ cannot be dropped as the Manhattan distance between droplets at $t = 10$ from droplets at $t^0 = 9$ is smaller than the time difference of time-steps. Therefore, all the droplets at $t^{00} = 12$ are retained.*

3) Experimental Results: We applied the proposed CP generation scheme to real-life benchmark assays described in Section IV-A. The results of the CP generation are tabulated in Table V. The proposed CP time-step derivation scheme monitors between 40% (REMIA) to 66% (PCR stream) cells to provide foolproof security compared to 100% cell coverage requirement in baseline strategy. The total run-time varies from 1 min (REMIA) to 60 min (PCR stream), which is a one-time cost incurred offline during design. The number of cells to be monitored is further reduced to 34% (Lin. gradient) to 63% (PCR stream) by using local minimization. We used $c = 1$ for our experiments and verified the security through the verify routine. This shows that the utility of proposed verification oracle in developing secure defense against a given attack model with practical resource requirement.

The total run-time for CP time-step derivation is determined by the number of invocations of the checkpoint verify routine. The run-time of the verify routine depends on the size of the bioassay and the DMFB size. We can reduce the run time overhead of the exact verification by decomposing the input bioassay into multiple sub-bioassays and verify each sub-bioassay separately. This will ensure the security of the entire bioassay and yet make the verification process scalable.

TABLE V: Results of secure CP generation.

2*Assay	Baseline CP		CP post-time-step derivation					Local pruning CP	
	τ_b time-steps	c_b cells	τ_s time-steps	c_s cells	#iters	Run time	$\frac{\tau_s}{\tau_b}$ 100	c_l cells	$\frac{c_l}{c_b}$ 100
REMIA	40	112	16	55	15	66.6 s	40%	48	42%
Lin. gradient	40	204	16	105	11	73.7 s	40%	69	34%
PCR mix	70	368	35	229	30	1226 s	50%	182	49%
PCR stream	74	604	50	454	40	3557 s	66%	384	63%

V. Discussion

In this section, we highlight some key points of our work in a Q & A format:

Q1. *Is the proposed exact analysis method applicable to different sensing schemes and different DMFB architectures?*
A: We have considered in our CCD-camera-based sensing as it is more popular. Also, we have considered individual pin addressable DMFB architecture as it is a generic version [70]. However, our method works independently of the sensing mechanism or the DMFB architecture. A designer would always be interested in determining defense that does not rely on full coverage. We believe this to be true irrespective of the sensing scheme and DMFB architecture, its associated constraints, and costs.
Q2. *Who could use the proposed exact analysis framework?*
A: On one hand, the exact analysis framework can be used by an attacker to find an attack plan, if the attacker knows the CP strategy. On the other hand, the designer can use it to red-team against the defense and check if the defense is fool-proof.
Q3. *How does the proposed method compare against the security verification method in other fields?*
A: Digital circuits can be logic-locked to prevent unauthorized access [71]. The satisfiability solvers have been used to break the logic-locked circuits [71]. Similarly, we used the satisfiability solvers to deduce the pathway to break the defense of DMFB. However, in case of DMFBs, careful modeling of DMFB droplet behavior is required, unlike the simple logical behavior of gates in circuit design. We have explained this modeling in detail in Section III.
Q4. *How can the checkpointing defend against a bad designer?*
A: The regulatory authorities use passive measures such as voluntary malfunction reporting to initiate action against the bad designers [72]. This can be very slow process, especially where biomedical systems are being constantly updated. In fact, Theranos fraudulent technology was exposed in October 2015 [73], but it voided *two years* of results only in May 2016 [19]. The regulatory authorities can use checkpointing-based active monitoring to regulate the DMFB products. This can also ensure that all the future software updates adhere to safety standards.
Q5. *Is the counterexample returned by the exact analysis always physically realizable on a DMFB?*
A: Yes, the counterexample returned by our tool is always realizable on a DMFB. The constraints presented in Section III-B ensure that the counterexample is realizable.
Q6. *Are all deviations in the DMFB implementation harmful? Can the exact analysis result be a false positive?*
A: Not all the deviations lead to meaningful attacks. For example, if a waste droplet being discarded from the DMFB is tampered with, then it does not lead to any tampering of final results. The defender can analyze the counterexample and its effect to determine the false positives.

VI. Conclusion & Future Work

Recent progress in DMFB security research has shown that it is possible to detect an attack with high probability [74]. This sets up the need to substantiate the same through exact analysis. Our work addresses this by developing a precise analysis, which proves if a DMFB implementation is secure. We came up with a symbolic formulation which surmounts the shortcomings of probabilistic security analysis by modeling all possible attack plans and applying satisfiability solvers to resolve them. We illustrate its application in checking various defense strategies for practical bioassays. This enables the designer to trade-off and compare the performance of various defense options. Next, we show that a fool-proof checkpoint can be derived using our verification oracle. Further, designers can use the verification oracle to design/verify optimal algorithm suited for different DMFB architectures and resource constraints.

We expect our contribution will advance the DMFB defense research to explore solutions that are not "probably secure" but "provably secure" boosting the commercial deployment of DMFBs. This can enhance the much-needed trust of medical practitioners, regulators, investors, users of the DMFBs, which in turn can boost the advancement of the technology. Future work can address the following open problems:

1) Uncertain and fault-prone response: Our current work does not distinguish between a hardware fault and an attack. We know from previous work that the range of uncertainty of fluidic operation timing can be determined experimentally [40]. Using this information, the SAT formulation can be modified to pose a different question: If the droplet sensor readings satisfy a given set of locations at a range of time intervals, then is it possible to execute a bioassay other than the one specified? This condition might increase the number of checkpoints required to verify the checkpoint defense scheme. However, the feasibility of such an analysis is still an open question.

2) Run-time defined control flow: A bioassay can include conditional loops such as if-then-else statements or loops with a non-constant number of iterations. In such a case, the actual execution path is resolved at run-time based on sensor readings. The proposed verification method does not consider such cases. However, a possible solution can be to save a

set of predetermined checkpoints depending on the execution path and use the SAT solver offline to verify the execution. Our current solution can be used to derive such predetermined checkpoints for each possible execution path. During run-time, the checkpoints corresponding to the chosen path are captured and saved for verification.

REFERENCES

[1] G. M. Whitesides, "The origins and the future of microfluidics," *Nature*, vol. 442, no. 7101, pp. 368–373, 2006.

[2] P. Pop, I. E. Araci, and K. Chakrabarty, "Continuous-flow biochips: Technology, physical-design methods, and testing," *IEEE Design & Test*, vol. 32, no. 6, pp. 8–19, 2015.

[3] N. Vergauwe *et al.*, "A versatile electrowetting-based digital microfluidic platform for quantitative homogeneous and heterogeneous bioassays," *J. Micromechanics Microengineering*, vol. 21, no. 5, p. 054026, 2011.

[4] C. D. Chin, V. Linder, and S. K. Sia, "Commercialization of microfluidic point-of-care diagnostic devices," *Lab Chip*, vol. 12, pp. 2118–2134, 2012.

[5] N. Khalid, I. Kobayashi, and M. Nakajima, "Recent lab-on-chip developments for novel drug discovery," *System Biology and Medicine*, vol. 9, no. 4, p. e1381, 2017.

[6] M. Turetta *et al.*, "Emerging technologies for cancer research: Towards personalized medicine with microfluidic platforms and 3d tumor models," *Current Medicinal Chemistry*, vol. 25, pp. 4616–4637, 2018.

[7] R. Fobel, C. Fobel, and A. R. Wheeler, "DropBot: An open-source digital microfluidic control system with precise control of electrostatic driving force and instantaneous drop velocity measurement," *Applied Physics Letters*, vol. 102, no. 19, p. 193513, 2013.

[8] (2016) Wheeler group at kakuma refugee camp in kenya. [Online]. Available: http://www.chem.utoronto.ca/edistillations/fall2016/kakuma.html

[9] (2016) A high throughput screening system to identify actionable treatments for cancer patients. [Online]. Available: https://biosero.com/wp-content/uploads/2016/10/a_high_throughput_screening_system_to_identify_actionable_treatments_etc_npm.pdf

[10] (2019) SEngine - Precision medicine. [Online]. Available: https://senginemedicine.com/

[11] A. H. C. Ng *et al.*, "A digital microfluidic system for serological immunoassays in remote settings," *Science Translational Medicine*, vol. 10, no. 438, 2018.

[12] (2019) Zion market research. [Online]. Available: https://www.globenewswire.com/news-release/2019/04/17/1805498/0/en/Global-Microfluidics-Market-Will-Surpass-USD-12-380-Million-By-2025-Zion-Market-Research.html

[13] (2019) Shipment of 3,000,000th test. [Online]. Available: https://baebies.com/celebrating-shipment-of-3000000th-test/

[14] (2019) 10x genomics funding. [Online]. Available: https://www.10xgenomics.com/news/10x-genomics-lands-new-financing/

[15] (2018) Illumina-press release. [Online]. Available: https://www.illumina.com/company/news-center/press-releases/press-release-details.html?newsid=1840193

[16] "FDA advisors back approval of Baebies' SEEKER analyzer for newborns," 2016. [Online]. Available: http://baebies.com/fda-advisors-back-approval-baebies-seeker-analyzer-newborns

[17] (1986) 5 cases of AIDS-study sabotage reported. [Online]. Available: https://www.chicagotribune.com/news/ct-xpm-1986-09-18-8603090884-story.html

[18] (2019) Jealousy led montana chemist to taint colleagues water tests. [Online]. Available: https://www.nytimes.com/2019/08/08/us/montana-chemist-water.html

[19] (2016) Theranos voids two years of edison blood-test results. [Online]. Available: https://www.wsj.com/articles/theranos-voids-two-years-of-edison-blood-test-results-1463616976

[20] (2019) Theranos effect. [Online]. Available: https://www.bloomberg.com/news/articles/2019-03-23/elizabeth-holmes-theranos-still-haunt-blood-testing-startups

[21] (2014) It's insanely easy to hack hospital equipment. [Online]. Available: https://www.wired.com/2014/04/hospital-equipment-vulnerable/

[22] (2018) When medical devices get hacked, hospitals often don't know it. [Online]. Available: https://www.healthcareitnews.com/news/when-medical-devices-get-hacked-hospitals-often-dont-know-it

[23] N. Dey, A. S. Ashour, F. Shi, S. J. Fong, and J. M. R. S. Tavares, "Medical cyber-physical systems: A survey," *Journal of Medical Systems*, vol. 42, no. 4, p. 74, 2018.

[24] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Towards secure and trustworthy cyberphysical microfluidic biochips," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 4, pp. 589–603, 2019.

[25] J. Riordon, D. Sovilj, S. Sanner, D. Sinton, and E. W. Young, "Deep learning with microfluidics for biotechnology," *Trends in Biotechnology*, vol. 37, no. 3, pp. 310–324, 2019.

[26] Y. S. Zhang, "A medical mini-me: one day your doctor could prescribe drugs based on now a biochip version of you reacts," *IEEE Spectrum*, vol. 56, no. 4, pp. 44–49, April 2019.

[27] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security assessment of cyberphysical digital microfluidic biochips," *IEEE/ACM Trans. on Compututational Biology and Bioinformatics*, vol. 13, no. 3, pp. 445–458, 2016.

[28] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Security trade-offs in microfluidic routing fabrics," in *Proc. ICCD*, 2017, pp. 25–32.

[29] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 59–72, 2013.

[30] J. Zambreno, A. Choudhary, R. Simha, B. Narahari, N. Memon, and N. Memon, "Safe-ops: An approach to embedded software security," *ACM Trans. Embed. Comput. Syst.*, vol. 4, no. 1, pp. 189–210, Feb. 2005.

[31] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Secure randomized checkpointing for digital microfluidic biochips," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 6, pp. 1119–1132, 2018.

[32] S. Bhattacharjee, A. Banerjee, K. Chakrabarty, and B. B. Bhattacharya, "Correctness checking of bio-chemical protocol realizations on a digital microfluidic biochip," in *Proc. VLSID*, 2014, pp. 504–509.

[33] S. McLaughlin *et al.*, "The cybersecurity landscape in industrial control systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, May 2016.

[34] C. Lin, J.-D. Huang, H. Yao, and T.-Y. Ho, "A comprehensive security system for digital microfluidic biochips," in *Proc ITC Asia*, 2018, pp. 151–156.

[35] S. Chakraborty, C. Das, and S. Chakraborty, "Securing module-less synthesis on cyberphysical digital microfluidic biochips from malicious intrusions," in *Proc. VLSID*, 2018, pp. 467–468.

[36] D. Grissom *et al.*, "An open-source compiler and PCB synthesis tool for digital microfluidic biochips," *INTEGRATION, the VLSI journal*, vol. 51, pp. 169–193, 2015.

[37] Z. Li, T.-Y. Ho, K. Lai, K. Chakrabarty, P. Yu, and C. Lee, "High-level synthesis for micro-electrode-dot-array digital microfluidic biochips," in *Proc. DAC*, 2016, pp. 1–6.

[38] O. Keszocze, R. Wille, T.-Y. Ho, and R. Drechsler, "Exact one-pass synthesis of digital microfluidic biochips," in *Proc. DAC*, 2014, pp. 1–6.

[39] R. B. Fair *et al.*, "Chemical and biological applications of digital-microfluidic devices," *IEEE Design & Test of Computers*, vol. 24, no. 1, pp. 10 – 24, 2007.

[40] Y. Zhao, T. Xu, and K. Chakrabarty, "Integrated control-path design and error recovery in the synthesis of digital microfluidic lab-on-chip," *ACM J. on Emerg. Technol. in Comput. Syst.*, vol. 6, no. 3, pp. 11:1–11:28, 2010.

[41] S. Bhattacharjee, S. Chatterjee, A. Banerjee, T. Ho, K. Chakrabarty, and B. B. Bhattacharya, "Adaptation of biochemical protocols to handle technology-change for digital microfluidics," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 36, no. 3, pp. 370–383, 2017.

[42] K. Hu, M. Ibrahim, L. Chen, Z. Li, K. Chakrabarty, and R. Fair, "Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform," in *Proc. IEEE BioCAS.*, 2015, pp. 1–4.

[43] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri,

"Microfluidic encryption of on-chip biochemical assays," in *Proc. BioCAS*, 2016, pp. 152–155.

[44] S. Bhattacharjee, J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Locking of biochemical assays for digital microfluidic biochips," in *Proc. ETS*, 2018, pp. 1–6.

[45] 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.

[46] H. C. Alphonsus, K. Choi, R. P. Luoma, J. M. Robinson, and A. R. Wheeler, "Digital microfluidic magnetic separation for particle-based immunoassays," *Analytical Chemistry*, vol. 84, no. 20, pp. 8805–8812, 2012.

[47] U. S. Service, CERT, C. Magazine, and Deloitte, "2011 cybersecurity watch survey: How bad is the insider threat?" 2011. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a589979.pdf

[48] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Security implications of cyberphysical flow-based microfluidic biochips," in *IEEE ATS.*, 2017, pp. 115–120.

[49] U.S.Government., "Increase in insider threat cases highlight significant risks to business networks and proprietary information," 2014. [Online]. Available: https://www.ic3.gov/media/2014/140923.aspx

[50] R. Langner, "Stuxnet: dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[51] M. A. Murran and H. Najjaran, "Capacitance-based droplet position estimator for digital microfluidic devices," *Lab on a Chip*, vol. 12, no. 11, pp. 2053–2059, 2012.

[52] M. Shayan, J. Tang, K. Chakrabarty, and R. Karri, "Security assessment of micro-electrode-dot-array biochips," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1831–1843, 2019.

[53] S. Mohammed, S. Bhattacharjee, T.-C. Liang, J. Tang, K. Chakrabarty, and R. Karri, "Shadow attacks on MEDA biochips," in *Proc. ICCAD*, 2018, pp. 73:1–73:8.

[54] T.-C. Liang, M. Shayan, K. Chakrabarty, and R. Karri, "Execution of provably secure assays on meda biochips to thwart attacks," in *Proc. ASPDAC*, 2019, pp. 51–57.

[55] O. Keszocze, R. Wille, K. Chakrabarty, and R. Drechsler, "A general and exact routing methodology for digital microfluidic biochips," in *Proc. ICCAD*, 2015, pp. 874–881.

[56] O. Keszocze, Z. Li, A. Grimmer, R. Wille, K. Chakrabarty, and R. Drechsler, "Exact routing for micro-electrode-dot-array digital microfluidic biochips," in *Proc. ASPDAC*, 2017, pp. 708–713.

[57] A. Grimmer, W. Haselmayr, A. Springer, and R. Wille, "Design of application-specific architectures for networked labs-on-chips," *IEEE Trans. on CAD of Integr. Circuits and Syst.*, vol. 37, no. 1, pp. 193–202, 2018.

[58] A. Grimmer, Q. Wang, H. Yao, T. Ho, and R. Wille, "Close-to-optimal placement and routing for continuous-flow microfluidic biochips," in *Proc. ASPDAC*, 2017, pp. 530–535.

[59] A. Grimmer, W. Haselmayr, A. Springer, and R. Wille, "Verification of networked labs-on-chip architectures," in *Proc. DATE*, 2017, pp. 1679–1684.

[60] A. Grimmer, B. Klepic, T. Ho, and R. Wille, "Sound valve-control for programmable microfluidic devices," in *Proc. ASPDAC*, 2018, pp. 40–45.

[61] Y. Luo, K. Chakrabarty, and T.-Y. Ho, *Biochemistry Synthesis Under Completion-Time Uncertainties in Fluidic Operations*. Springer International Publishing, 2015, pp. 95–116.

[62] J. Ott, T. Loveless, C. Curtis, M. Lesani, and P. Brisk, "Bioscript: Programming safe chemistry on laboratories-on-a-chip," *Proc. ACM Program. Lang.*, vol. 2, no. OOPSLA.

[63] M. Willsey *et al.*, "Puddle: A dynamic, error-correcting, full-stack microfluidics platform," in *ASPLOS '19*, April 2019.

[64] R. B. Fair, "Digital microfluidics: is a true lab-on-a-chip possible?" *Microfluid Nanofluid*, vol. 3, no. 3, pp. 245–281, 2007.

[65] H. Chen, S. Potluri, and F. Koushanfar, "Biochipwork: Reverse engineering of microfluidic biochips," in *Proc. ICCD*, 2017, pp. 9–16.

[66] N. Eén and N. Sörensson, "An extensible SAT-solver," in *Proc. of SAT*, ser. Lecture Notes in Computer Science, vol. 2919, 2003, pp. 502–518.

[67] R. Wille, G. Fey, D. Große, S. Eggersglüß, and R. Drechsler, "SWORD: A SAT like prover using word level information," in *Proc. of VLSI-SoC*, 2007, pp. 88–93.

[68] L. M. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Proc. TACAS*, 2008, pp. 337–340, [Z3 is available at https://github.com/Z3Prover/z3].

[69] J.-D. Huang, C. Liu, and T. Chiang, "Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees," in *Proc. ICCAD*, 2012, pp. 377–383.

[70] D. Grissom and P. Brisk, "A field-programmable pin-constrained digital microfluidic biochip," in *Proc. DAC*, 2013, p. 46.

[71] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. HOST*, 2015, pp. 137–143.

[72] (2019) Medical Device Reporting: How to Report Medical Device Problems. [Online]. Available: https://www.fda.gov/medical-devices/medical-device-safety/medical-device-reporting-mdr-how-report-medical-device-problems

[73] (2015) Hot startup theranos has struggled with its blood-test technology. [Online]. Available: https://www.wsj.com/articles/theranos-has-struggled-with-blood-tests-1444881901

[74] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Tamper-resistant pin-constrained digital microfluidic biochips," in *Proc. DAC*, 2018, pp. 1–6.