Secure Randomized Checkpointing for Digital Microfluidic Biochips

Jack Tang, *Student Member, IEEE*, Mohamed Ibrahim, *Student Member, IEEE*, Krishnendu Chakrabarty, *Fellow, IEEE*, and Ramesh Karri, *Senior Member, IEEE*

***Abstract*—Digital microfluidic biochips (DMFBs) integrated with processors and arrays of sensors form cyberphysical systems and consequently face a variety of unique, recently described security threats. It has been noted that techniques used for error recovery can provide some assurance of integrity when a cyberphysical DMFB is under attack. This paper proposes the use of such hardware for security purposes through the randomization of checkpoints in both space and time, and provides design guidelines for designers of such systems. We define security metrics and present techniques for improving performance through static checkpoint maps, and describe performance tradeoffs associated with static and random checkpoints. We also provide detailed classification of attack models and demonstrate the feasibility of our techniques with case studies on assays implemented in typical DMFB hardware.**

***Index Terms*—Biochips, cyberphysical systems, microfluidics, security.**

I. Introduction

DIGITAL microfluidic biochip (DMFB) technology is rapidly maturing after years of research on basic physics, materials, applications and design automation. Within design automation, many fundamental topics are well developed—high-level synthesis [1]–[3], fault-tolerance [4], [5], error recovery [6], [7], chip testability [8], [9], pin-count reduction [10], and PCB escape routing [11], to name a few, have contributed to make DMFBs more practical and usable, especially on high-throughput and highly multiplexed devices. Commercial devices utilizing DMFB technology, such as the Illumina NeoPrep Library System [12], have been deployed in recent years while the market for lab-on-a-chip (LOC) technology in general reached $3.9 billion in 2014 and is projected to grow $18.4 billion in 2020 [13]. Unfortunately, these technical achievements have occurred in an era where computer security threats are rampant and increasing in sophistication. As has been shown in certain categories of devices such as the Internet-of-Things, failure to address security at the onset of the design phase can lead to disastrous results, such as with the compromise of millions of devices to form the Mirai botnet [14].

DMFBs, in contrast to continuous flow valve-based biochips [15], manipulate droplets in discrete quantities by utilizing the electrowetting-on-dielectric (EWOD) [16] effect. EWOD allows the contact angle between a droplet and its underlying electrode to be modulated through the application of a suitable control voltage. Movement of droplets can then be induced by applying a low-voltage to an electrode with a droplet and a high-voltage to an adjacent electrode. The structure of a typical DMFB consists of two electrode layers coated with a hydrophobic layer between which droplets are placed [Fig. 1(a)]. The bottom electrode layer is patterned to implement reservoirs, channels, or a general-purpose grid. Driving these electrodes with a sequence of control voltage patterns, called *actuation sequences*, can implement droplet handling operations including the dispensing of reagents and samples from reservoirs, shuttling, mixing and splitting. These basic operations in turn can be used to implement a vast array of biological assays, such as immunoassays, protein crystallization, and DNA sequencing [17]. The rate at which actuation sequences can be applied to the DMFB is limited by the physics of the droplet movement, and in typical systems is on the order of hundreds to several thousand Hertz [18].

A functional DMFB platform requires many additional components in addition to the grid of electrodes; a computer generates and sends actuation sequences to the biochip, sensors gather information on the assay execution, and actuators add extra droplet processing capabilities such as heating and cooling. These components together form a cyberphysical DMFB [Fig. 1(b)] [19]. The use of sensor feedback allows the computer to detect erroneous operation and respond appropriately, providing resilience against hardware faults and fluid handling errors such as incomplete mixing [6].

Research on error recovery techniques have utilized charge-coupled device (CCD) cameras to provide real-time sensor feedback to the biochip controller. CCD cameras are reconfigurable and able to determine the state of the biochip at arbitrary locations and times. This inspection operation is called a *checkpoint*. Software executing on the controller takes the images captured by the camera and extracts droplet presence, volume and concentration using pattern-matching algorithms with template images [20], [21]. Proposed approaches to pattern-matching include generating a correlation map for the entire biochip and determining location from electrodes with the highest correlation [20], as well as performing a single droplet correlation after cropping the image at specific locations [6]. While CCD imaging is flexible and effective,

Manuscript received February 9, 2017; revised May 7, 2017; accepted July 13, 2017. Date of publication August 31, 2017; date of current version May 18, 2018. This work was supported in part by the ARO under Grant W911NF-17-1-0320 and CCS-AD. This paper was recommended by Associate Editor S. Bhunia. *(Corresponding author: Jack Tang.)*

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

M. Ibrahim and K. Chakrabarty are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2017.2748030

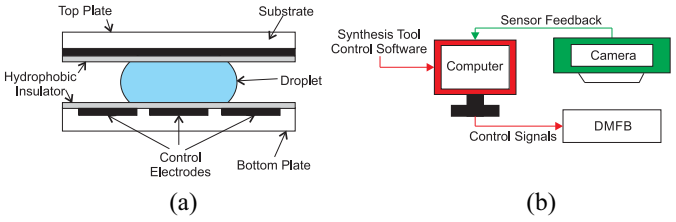


Fig. 1. (a) Structure of a DMFB array as viewed from the side. (b) Schematic of a typical cyberphysical DMFB system [19].

it may not be usable for assays that utilize light-sensitive reagents [22]. We also note that other types of sensing hardware can implement checkpoints, as was recently demonstrated for a reliability-hardening technique [23].

The fact that a DMFB is often implemented as a cyberphysical system implies unique security challenges; not only is data at risk, but now physical assets are susceptible to alteration and destruction. The DMFB controller is typically a general purpose computer or microcontroller, with network connectivity for the purposes of loading new bioassays. Such reprogrammability and connectivity provides means for an attacker to exploit the system. While research on practical threats and motivations is ongoing, preliminary work has shown that an attacker would be able to subtly alter the results of an assay as well as to cause denial-of-service (DoS) attacks [24]. Alteration of test results has serious implications for mission-critical applications such as diagnostic testing for patient care, while denial-of-service could potentially destroy expensive reagents and difficult-to-obtain samples.

The unique physical modalities of cyberphysical DMFBs give rise to new security threats, but at the same time provide an opportunity for unique defenses. In particular, the imaging systems used for the purposes of error-recovery can be utilized to provide assurances of security, since in a general sense they are designed to counteract anomalies. This paper analyzes the performance of a randomized checkpoint system, where a checkpoint is defined as the act of comparing the state of an electrode against that of the assay specification. While a system that monitors the entire biochip at every electrode for all time is guaranteed to detect all anomalies, such an implementation would impose severe processing and memory constraints on DMFB designs where the goal is to miniaturize and keep costs low. Indeed, the realization of the lab-on-a-chip depends on making any additional security features as lightweight as possible. Randomization permits fast probing of the biochip state while causing uncertainty for an attacker.

This paper addresses the need for novel security techniques in the context of digital microfluidics by describing an intrusion detection system (IDS) based on the randomization of checkpoints. We present theoretical analysis of the checkpoint system and provide guidelines for system designers. The contributions of this paper over those presented in the initial conference publication [25] are as follows.

1) We generalize the checkpoint system for use with nonuniform probability distributions, and show that the uniform distribution provides the most security benefit for most general-purpose DMFB architectures.
2) We derive a static checkpoint placement algorithm from graph-theoretic techniques that greatly increases the probability that an attack is detected.
3) Typical system constraints are analyzed to illustrate real-world performance results.
4) The techniques are applied to an application-specific commercial DMFB design to demonstrate the type of performance that can be achieved given a realistic attack scenario.

The structure of this paper is as follows. Section II briefly reviews related work. We present a detailed threat model in Section III, and then proceed to describe the proposed randomized checkpoint system in Section IV with an analysis of biased distributions in Section V. We improve upon the base design by introducing the concept of static checkpoints in Section VI. Experimental results are presented in Section VII, while a summary and general thoughts on DMFB security are presented in Section VIII.

II. Related Work

The fact that DMFBs are a relatively new technology mean that the security analysis and techniques is similarly in a nascent stage. Research has demonstrated that DMFBs are prone to a variety of stealthy attacks. Alteration of high-level assay specifications and low-level actuation sequences can lead to denial-of-service, where assays are disabled entirely, or subtler attacks where the resulting errors are undetectable [24]. Furthermore, the integrity of the DMFB supply chain is an open question [26]. Additionally, DMFB hardware is susceptible to the same intellectual property issues that have plagued the IC industry. To that end, a method to prevent piracy utilizing physical unclonable functions was proposed recently [27].

A *hardware trojan* is a modification of a circuit designed to cause unwanted behavior. Hardware trojans can be inserted at any level during the integrated circuit design flow, from high-level system design specification all the way down to the transistor level. The effects include, but are not limited to, leakage of sensitive information, denial-of-service, and alteration of the functionality of a device [28], [29]. The threat of hardware trojans is real, as there have been reported instances of trojan insertion [30], and the modern horizontal manufacturing design flow presents numerous avenues for a malicious adversary to exploit. Techniques to prevent, detect, and thwart hardware trojans include functional testing [31], delay-path fingerprinting [32], ring-oscillator characterization [33], input scrambling [34], and static verification [35]. Recent review papers provide more in-depth coverage [36]–[38].

The computer used as a controller for a typical cyberphysical DMFB implementation is susceptible to hardware trojan insertion. While this attack vector is considered part of the threat model, the emphasis of this paper is on detecting attacks that target the DMFB actuation sequences such that the biochip operates in an unintended way. This malicious objective can be achieved through the payload of a hardware trojan. We do not consider the problem of detecting the existence of hardware trojans.

Design automation technology for very large scale integration and DMFBs share some similarities, and therefore some design-for-security techniques may be directly translatable. Though more interestingly, the differences between the two technologies may lead to opportunities for divergent approaches to security. DMFBs utilize electrodes as a reconfigurable resource that can be used as either a processing element

or a routing element, whereas integrated circuits utilize transistors for processing and wires for routing. Furthermore, the resources in a DMFB may be reconfigured during the execution of an assay. Even a reconfigurable technology like an FPGA remains static during execution. This extra degree of freedom contributes to the complexity of DMFB design. Additionally, the execution of a DMFB is easily observable with a camera whereas integrated circuits are opaque to all but the most determined parties. It is precisely these differences that will drive the design of the checkpoint system proposed in Section IV.

III. DMFB Attacks

A. Threat Model

Motivations for an attack are varied and depend on the application. In a diagnostic patient care scenario, an attacker may be interested in physically harming a person by misinforming a physician about a test result. Corporations or government entities may be interested in disrupting the progress of a scientific experiment, which famously has precedent in the Stuxnet worm [39].

We assume that a malicious adversary is able to modify the low-level actuation sequences of the DMFB hardware to an extent that she may purposefully dispense, route, and mix droplets. This could occur through the insertion of hardware trojans as described previously, or through modification of the control software. Control software may be compromised through a network connection originally integrated for software updates or convenient downloading of assay specifications [40]. Additionally, embedded systems meant to be deployed at the point-of-care are physically vulnerable to modification. The operator of the DMFB is presumed to be trustworthy, and that there is no tampering with the physical aspects of the system such as the loading of samples/reagents and the imaging system. While this paper assumes that the execution of control sequences is prone to attack, we do presume the integrity of the proposed defense mechanism.

We make no assumptions about the architecture of the DMFB. The techniques described in this paper are general and can be applied to both general-purpose programmable DMFBs as well as application-specific DMFBs. However, there may be more use for random checkpoints on a general-purpose biochip, since an attacker would have access to more resources to carry out an attack. Error recovery may or may not be implemented. If it happens to be present on the biochip under consideration, an attacker is assumed to be able to predict and evade their location since they are placed using deterministic algorithms [6].

One potential threat is illustrated in Fig. 2. This example shows the final execution cycles of a polymerase chain reaction (PCR) assay. The PCR assay is used in DNA amplification and has been studied in the DMFB literature. At clock cycle i, a malicious droplet has been dispensed from the AmpliTaq DNA polymerase port to be routed to mix module $M1$. Higher concentrations of AmpliTaq increase production of nonspecific products, lowering the quality of the assay output [41]–[43]. Either the DMFB error detection scheme will detect the wrong concentration at the output of this mix stage, or this altered droplet will be allowed to propagate through the assay if no error recovery is implemented. In either case, the result of the attack is either a denial-of-service, or output alteration/contamination.

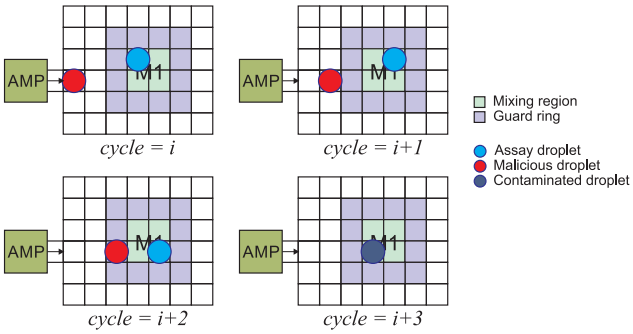


Fig. 2. Potential malicious route. Droplets can be dispensed to foul/contaminate a good droplet in a mix module. The target droplet concentrations can be altered to either cause failure of the assay (denial-of-service) or to report incorrect measurements.

The consequence of output contamination/alteration is that an assay result may be interpreted as accurately reflecting reality. This can be dangerous in cases where the assay is used to perform some measurement or test [44], for example, *in-vitro* glucose measurement. If a user's glucose measurement is inaccurate, the wrong dosage of insulin may be administered which could lead to overdose. The result of a DoS attack is that the DMFB is not able to perform its intended function, causing inconvenience while wasting samples, reagents and money. But more insidiously, a DoS attack, if not detected as a DoS attack, may trick error correction to believe that a hardware fault has occurred. Electrodes may be marked as faulty when they are still functional, causing the DMFB hardware to have reduced fault-tolerance and shorter operating lifespan.

B. DMFB Attack Modeling

All practical malicious modifications require the movement of droplets from a source to a target on the DMFB. Examples of sources include dispense ports, waste reservoirs, and backup reservoirs. Examples of targets include output ports, backup reservoirs, mix modules and droplets in transit. It is conceivable that an adversary could mount an attack that does not alter the result, such as dispensing extra reagents into unused electrodes, but the focus in this paper remains on attacks that change the assay result.

Hence, we model the class of attacks that can be formulated as a misrouting problem between a source and a target. It should be noted that not all (source, target) combinations result in a meaningful attack. For example, routing a wash droplet into an output port would be easily detected as a fault since it bears no resemblance to the desired output droplet. This observation will save some time in analyzing and evaluating the proposed defense system in Section IV. Table I enumerates the typical resources in a DMFB platform and classifies them as potential sources or targets for a malicious droplet.

C. Attack Classification

Malicious modification of a DMFB actuation sequence can be classified according to the degree of modification as follows.

1) *Bit Flip:* A single bit in the actuation sequence is modified. Such an attack can be achieved through physically

TABLE I
CLASSIFICATION OF RESOURCES

Resource	Source	Target
Dispense Port	✓	
Output Port		✓
Waste Reservoir	✓	
Backup Reservoir	✓	✓
Mix Module	✓	✓
Droplets in Transit	✓	✓

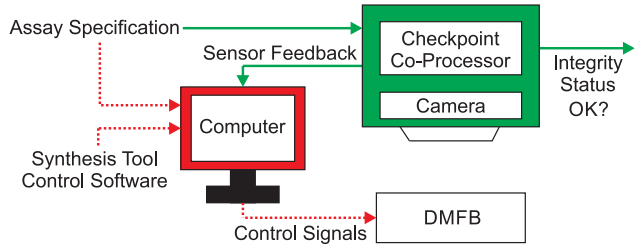


Fig. 3. DMFB secure co-processor implementation. Solid lines indicate signals assumed to be trustworthy while dotted lines are susceptible to attack.

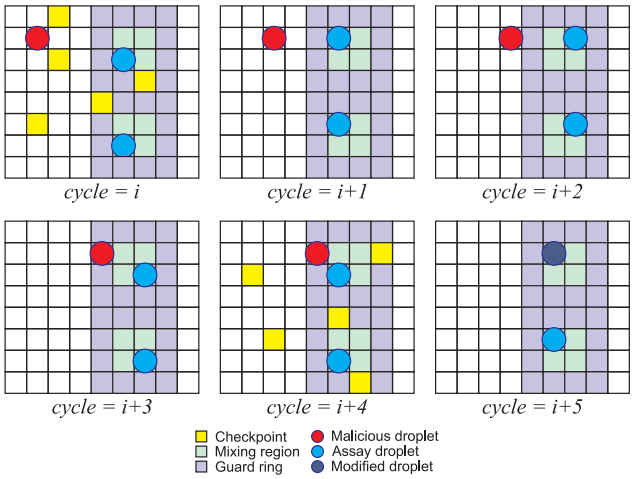


Fig. 4. Example assay execution with random checkpoints applied at cycle i and $i+4$. In this case, $s = 72$, $k = 5$, and $L = 5$. This example is a realization of the random sampling times. Assuming $c = 0.33$, then $P(E) = 0.89$.

inducing errors in the hardware, similar to reported fault injection attacks in cryptographic hardware [45].

2) *Sequence Modification/Insertion/Deletion:* N bits of the actuation sequence can either be modified, inserted or deleted. An intelligent adversary would be able to manipulate droplets in such a way that most of the assay proceeds normally.
3) *Complete Substitution:* The most extreme attack is to completely replace the correct actuation sequence with an alternate sequence. The result of such an attack is likely to be noticed, since error recovery mechanisms can detect the deviation from specification. Additionally, large deviations in processing time could be detected by the DMFB operator.

This paper models attacks as routed droplets, and thus addresses level 2 attacks. These attacks can potentially induce the most harm while being more difficult to detect.

IV. RANDOMIZED OPTICAL CHECKPOINTS

We propose the use of a security co-processor, which is physically isolated from the DMFB controller so as to increase the attack surface (Fig. 3). The co-processor is able to selectively probe the status of droplets on the biochip and compare them to the assay specification. The co-processor should have a separate physical indicator to alert the DMFB operator when an anomaly is detected.

Checkpointing, or the monitoring of an assay's progression is a technique for ensuring the integrity of an assay. Given unlimited resources, the most secure action is to record the entire assay at a high sample rate and analyze the log for anomalous behavior. This soon becomes cost prohibitive; the computing and memory requirements for analyzing the data from a high-resolution camera will quickly erode any low-cost benefits of using DMFB technology. Instead of full security, we seek a solution where we can estimate the probability that the assay has not been tampered through sampling the assay randomly in both time and space.

We propose a randomized optical checkpoint system that works as follows (Fig. 4).

1) *Determination of Which Electrode to Examine:* The system randomly chooses an electrode according to a uniform distribution over the electrodes. That is, assuming a DMFB with s number of electrodes, we assign an index $j \in \{0, 1, \ldots, s-1\}$ to each electrode according to some predefined convention (e.g., left-to-right, top-to-bottom) and define a random variable J which represents the outcome of randomly selecting an electrode. The probability mass function (PMF) is defined as $p_J(j) = 1/s \quad j \in \{0, 1, \ldots, s-1\}$.
2) *State Extraction:* The controller focuses the imaging system on the electrode and runs a correlation algorithm against a template image to extract the state of the droplet. In general the state may include volume and concentration, but in this paper we assume that it is sufficient to extract only the presence or absence of a droplet.
3) *Comparison With Specification:* The controller will then compare the state of electrode j against the specification stored in memory, and signal an error if they do not match.
4) *Repeat:* The previously chosen electrode is marked as chosen, and a new electrode is chosen from the remaining pool. The PMF is now $p_J(j) = 1/(s - |X|) \quad j \in \{0, 1, \ldots, s-1\} \setminus X$ where X is the set of electrodes already chosen. The system repeats this process up until a number k defined by the system designer. Since the DMFB is physically limited by the fundamental actuation frequency, all the inspection events within a time step can be considered to be occurring simultaneously.

A. Probability of Evasion

The security metric used to evaluate effectiveness is the probability of evasion. The complement of evasion is detection. An IDS that can monitor biochip behavior at every electrode for the entire duration of an assay should give probability of evasion equal to 0. A system without an IDS has probability of evasion equal to 1.

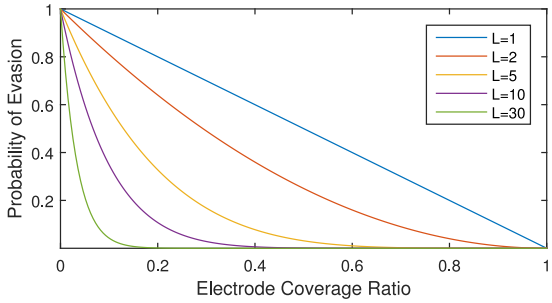


Fig. 5. Probability of evasion decreases as the checkpoint coverage ratio, k/s, increases. Here $c = 1$, meaning checkpoints are monitored on every execution cycle. Increasing L as a parameter shows exponential advantage, illustrating the intuitive notion that the longer a malicious droplet exists on an assay the more likely it is to be detected.

We model the probability of evasion in terms of a malicious droplet to be routed on the biochip. Let E be the event that a malicious droplet evades detection for the lifetime of the droplet that executes over L total cycles. L may be much less than the lifetime of the assay. E_i is the event that a malicious droplet evades detection for the ith execution cycle, and F_i is the event that the ith cycle is sampled and G_i is the event that the ith cycle's checkpoints intersect with the malicious droplet. If each cycle's checkpoints are chosen independently, and the events F_i and G_i are independent, then the evasion event is equivalent to the event that the cycle is not sampled, or the cycle is sampled and the set of checkpoints do not monitor the droplet

$$P(E_i) = P\big(\overline{F_i} \cup \big(F_i \cap \overline{G_i}\big)\big) = P\big(\overline{F_i}\big) + P(F_i) \cdot P\big(\overline{G_i}\big) \quad (1)$$

$$P(E) = P(E_1 \cap E_2 \cap \cdots \cap E_L) = \prod_{i=1}^{L} P(E_i). \quad (2)$$

The probability that a malicious droplet does not intersect with any checkpoints is the complement of the ratio of active checkpoints k at that time over the number of total electrodes s. We have

$$P\big(\overline{G_i}\big) = 1 - \frac{k}{s}. \quad (3)$$

The ratio k/s is called the *electrode coverage ratio*. Then we define the probability of sampling any execution cycle using some constant c as

$$P(F_i) = c. \quad (4)$$

This constant is a design parameter that can be adjusted in software. Therefore, the probability of evasion can be expressed as

$$P(E) = \prod_{i=1}^{L}\left((1-c) + c\left(1 - \frac{k}{s}\right)\right) = \left(1 - \frac{ck}{s}\right)^{L}. \quad (5)$$

The parameter s is a constant determined by the size of the DMFB array. The parameters c and k should be maximized in order to minimize the likelihood of evasion, subject to the computational and imaging capabilities of the DMFB platform. Note that the IDS has no control over the route an attacker may take, and that this model does not assume any particular malicious route. The only assumption on the attack is that it exists for a certain number of cycles L, and that the probability of evasion has an exponential dependence on L (Fig. 5). This key observation leads to the possibility of decreasing $P(E)$ through influencing the routability of malicious as a result of judicious placement of static checkpoints.

V. Biased Probability Distributions

The analysis presented in Section IV assumes that each electrode is chosen to be sampled independently and with uniform probability. The system can bias the distribution, since it is plausible that some electrodes are more useful for an attacker than others. However, an intelligent adversary would be able to use this information to avoid electrodes that are more likely to be detected. It is not immediately clear if there is any benefit to biasing because of the threat model. Before we show how to reconcile this conflict, we introduce some new definitions and generalizations.

A. Biased PMF

We can generalize the probability of evasion to consider nonuniform distributions over the electrodes and define our PMF as

$$p_J(j) = 1/s + b(j) \quad (6)$$

where $b(j)$ is a bias term. That is, we describe the PMF in terms of deviations from the uniform distribution. Note that $\sum p_J = 1 \Rightarrow \sum b(j) = 0$, and $b(j) \in (-1/s, 1 - 1/s)$. In the uniform case, $b(j) = 0$. For convenience, we will notate $b^0(j)$ when we know that $b(j)$ will evaluate to 0 for some j, and similarly notate $b^+(j)$ and $b^-(j)$ when we know there will be a positive or negative bias, respectively.

B. Generalized Probability of Evasion

Equation (5) is a function of several variables. We introduce the notation $P_{Ei}(j, k, b(j))$ to mean the probability of the event E_i during cycle i as a function of the currently occupied electrode indexed by j, with k number of random checkpoints per cycle and some bias function b evaluated at j. Generalizing (2) we write the product not only over the cycle i but also the attack Path, where a Path is defined as an ordered set of ordered pairs (i, j) indicating the time-step and location of a path. That is, a Path $\subset (\mathbb{N} \times \mathbb{N})^L$

$$P(E) = \prod_{i,j \in \text{Path}} P_{Ei}(j, k, b(j)). \quad (7)$$

Using b^0 simplifies to the analysis in the previous section, while $k = 1$ simplifies P_{Ei} equal to $1 - p_J(j)$. Evaluating $P_{Ei}(j, k, b(j))$ in general is difficult, since it is equivalent to the probability of selecting k combinations out of $s - 1$ electrodes not occupied by the droplet. However, we do know that an electrode with higher bias is less likely for an attacker to evade detection than an electrode with lower bias. That is, if $b(\alpha) > b(\beta)$ then $P_{Ei}(\alpha, k, b(\alpha)) < P_{Ei}(\beta, k, b(\beta))$.

C. Decomposition of Probability of Evasion

Now assume an arbitrary DMFB array with uniform distribution. If we perturb the distribution on some electrode α such that $b(\alpha) = \delta$ for $\delta \in (-1/s, 1 - 1/s)$, we must also distribute $-\delta$ among one or more of the other electrodes in order to satisfy $\sum b(j) = 0$. The probability that a malicious route

evades detection was given in (7). We can break this equation into three components according to bias. Denote some arbitrary bias $b^-(j) < b^0(j) < b^+(j)$. We rewrite (7) in terms of these three biases as

$$P(E) = \prod_{x\in X} P_{Ei}\big(x, k, b^-(x)\big) \times \prod_{y\in Y} P_{Ei}\Big(y, k, b^0(y)\Big) \times \prod_{z\in Z} P_{Ei}\big(z, k, b^+(z)\big) \tag{8}$$

where X is the set of electrodes with negative bias, Y is the set of electrodes with no bias, and Z is the set of electrodes with positive bias, and Path $= X \cup Y \cup Z$. Equation (8) can be thought of as the multiplication of $|X| + |Y| + |Z|$ number of probability terms, where each probability term in X is less than every term in Y is less than every term in Z. In certain cases, this decomposition can facilitate the relative comparison of performance between two routes.

D. Security of Biased Distributions

The security of a given bias distribution is determined by the worst-case performance. The worst-case performance in the uniform distribution is determined by the shortest attack route. We denote $P_{E\min}$ as the worst-case (shortest-path) probability of evasion for the uniform distribution. Any probability of evasion for a biased distribution P_E^* should not exceed this limit. That is

$$P_E^* \leq P_{E\min}. \tag{9}$$

It is clear that a route with one or more negatively biased electrodes has higher $P(E)$ than one that is unbiased. Similarly, a route with one or more positively biased electrodes has lower $P(E)$. Due to the fact that any positive bias has to be compensated by a negative bias on another electrode, it is conceivable that there are bias schemes that do not provide any net benefit. Since each electrode can only be negatively biased by a maximum of $-\delta = 1/s$, attempting to apply more than $+\delta = 1/s$ bias to one or more electrodes means that more than one electrode is adversely affected. Furthermore, if on a particular DMFB architecture, every electrode is part of more than one minimum length route, that means applying a compensating bias will always have negative consequences on another route.

Fig. 6(a) illustrates a uniform probability distribution over a typical general purpose DMFB source-target configuration. If a positive bias is applied in an attempt to improve the odds of catching the malicious route in red in Fig. 6(b), then a negative bias needs to be applied elsewhere. This can be done in several ways. Fig. 6(b) shows the negative bias applied to an alternate route. Fig. 6(c) shows the bias being distributed over three electrodes, forming an easier path for an attacker to take. Neither case is preferable, since the negative bias causes the condition in (9) to be violated.

In general, most electrodes on a general-purpose DMFB grid are part of more than one minimum-length route. Therefore, for security reasons and for simplicity of system design, the recommended distribution for most DMFB architectures is the uniform distribution.

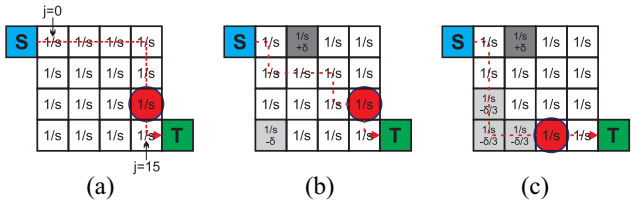


Fig. 6. (a) Representation of a uniformly distributed electrode arrangement. (b) Biasing with $+\delta$ on electrode $j = 1$ and $-\delta$ on electrode $j = 12$. (c) Biasing with compensating delta spread out over multiple electrodes. $b(j) = +\delta$ $(j = 1)$, $-\delta/3$ $(j = \{8, 12, 13\})$, $1/s$ *(otherwise)*. Electrodes in set Y are white, X in light gray, and Z in dark gray.

VI. Static Checkpoint Placement

While randomized checkpoints are the foundation of the defense, the addition of static checkpoints can increase the overall effectiveness. Static checkpoints by themselves provide weak security guarantees; it relies on their location being kept secret. Under our proposed threat model, the location of these static checkpoints are known to the attacker. An attacker with such knowledge is best served by avoiding the static checkpoints. The judicious placement of static checkpoints can influence the type of routes an attacker will take.

A. Problem Statement

We represent a general purpose DMFB array as the integer grid $\mathbb{N} \times \mathbb{N}$, with a set of sources $S \subset \mathbb{N} \times \mathbb{N}$ of cardinality s and a set of targets $T \subset \mathbb{N} \times \mathbb{N}$ of cardinality d. The problem is to find the smallest set of static checkpoints (or obstacles) $K \subset \mathbb{N} \times \mathbb{N}$ of cardinality k such that the smallest routing length between every source and target is maximized. In other words, we seek solutions to the problem

$$\underset{K}{\text{argmax}}\, \min(\text{Routes}(S, T, K)) \tag{10}$$

where $\text{Routes}(S, T, K)$ is the set of all possible routes between every source and target that doesn't collide with an obstacle in K, and $\min(\ldots)$ returns the smallest route of this set. The function to be maximized is difficult to solve because it considers all feasible route lengths rather than just the Manhattan distance or L_1 norm. For an $m \times n$ grid, there are $\binom{m\times n}{k}$ ways to place k checkpoints. Even on a relatively small DMFB platform with 120 electrodes and 20 checkpoints, that amounts to over 2^{74} combinations to choose from. Furthermore, the computation of the set of feasible routes is nontrivial [$O(mn)$ for maze type router [46]].

B. Minimal Provably Secure Placement

The problem in (10) describes an optimal arrangement of static checkpoints. However, we can relax this requirement and still provide security guarantees by defining a *minimal provably secure* checkpoint map as a set K such that

$$\min(\text{Routes}(S, T, K)) > \min(\text{Routes}(S, T, \emptyset)). \tag{11}$$

In other words, we seek any set of checkpoints that causes the smallest possible route between the sources and targets to be greater than the smallest possible route without any static checkpoints. On a general-purpose biochip with no obstacles, a droplet can be routed with a Manhattan length route.

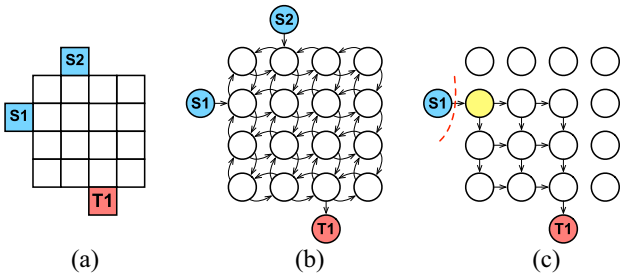


Fig. 7. (a) Sample DMFB architecture with two sources and one target. (b) Graph representation $G(V, E)$ of the DMFB. (c) Transformed graph representation $G*(V, E)$ for determination of static checkpoint placement between $S2$ and $T1$. Note that grid structures give rise to simple solutions where checkpoints should be placed close to either source or target.

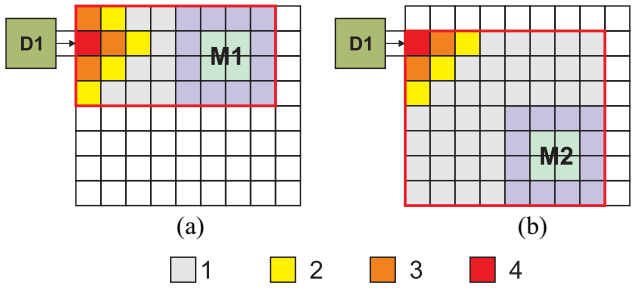


Fig. 8. (a) Rectangle used to approximate usefulness of electrodes in harboring a malicious route for dispense port 1 and mix module 1. (b) Rectangle for approximation between dispense port 1 and mix module 2.

We propose a solution set K to this problem using Hadlock's theorem [47] and concepts from graph theory.

Hadlock's Theorem: A path α from P to Q has length $d_M(P, Q) + 2 \cdot d$ where d is the detour number of α with respect to Q and $d_M(P, Q) = |x_Q - x_P| + |y_Q - y_P|$ is the Manhattan distance between P and Q.

Since the detour number d is a number greater than or equal to 0, the path length is minimized when $d = 0$. Therefore the shortest route length is equal to the Manhattan distance. This somewhat obvious statement provides a baseline by which to measure the performance of any obstruction placement algorithm. Furthermore, the fact that no detours are permitted allow us to make a simplification to the problem.

To solve for a set K that increases the route length, we can represent an array of DMFB electrodes as a directed graph $G(V, E)$, where each vertex represents an electrode and each edge represents a possible path. Associated with each vertex is an (x, y)-coordinate describing its location on the DMFB array, where the origin is taken to be the upper-leftmost electrode. On an unmodified electrode grid, a droplet is free to move in any direction, so each forward edge has a matching return edge [Fig. 7(a) and (b)].

Since no detours are permitted, we can convert the directed graph G into a modified graph $G*$ where all detour edges are removed [Fig. 7(c)]. For a target node T, a detour edge is defined as any edge connecting a two nodes A and B where $(y_T - y_A) < (y_T - y_B)$ or $(x_T - x_A) < (x_T - x_B)$. That is, $G*$ represents all possible minimal length routes. If we desire to eliminate all possible minimal length routes, the problem is to choose where to break the graph. Breaking the graph is equivalent to placing static checkpoints on the boundary of the cut, as an attacker cannot cross the cut. We seek solutions where the cut has the fewest number of edges, since the number edges translates directly into static checkpoints that must be scanned. The problem of inserting static checkpoints such that a longer route is forced is equivalent to solving the minimum s-t cut problem with each edge being assigned a uniform flow. We may compute such a solution using any of the available algorithms in [48] and [49]. This result is general and can apply to any DMFB architecture. However, the simple symmetric structure of general purpose DMFBs leads to a predictable solution for optimal placement: electrodes directly adjacent to sources and targets should be monitored [Fig. 7(c)].

To create a checkpoint map that secures the entire biochip, it suffices to enumerate all combinations of sources and targets, generate each individual checkpoint map using the s-t min-cut formulation, and superimpose them. The resulting superposition can be interpreted as a matrix where each individual cell represents the number of static checkpoints that were added. The resulting map thus presents a ranking of electrodes in terms of how many attacks it can be used to carry out. In a constrained environment, the designer may choose the highest ranked electrodes. While this checkpoint map is provably secure, it is not optimal in that another checkpoint map may exist which achieves the same performance with fewer checkpoints.

C. Heuristic Placement

Static checkpoint placement can be approximated by a heuristic algorithm, since it has been observed that symmetric general purpose DMFBs admit predictable solutions. Furthermore, we have some intuitive notion of where it might be useful to place static checkpoints; we can prevent many attacks by placing a single checkpoint at droplet sources. The idea is to rank each electrode on the DMFB array in terms of how useful it is to an attacker for routing a malicious droplet, and to select electrodes from this list up to a number defined by hardware limitations.

We give an approximate ranking of electrodes by enumerating all useful combinations of sources and sinks, and form a rectangle containing the farthest edges of these resources. A matrix representation of the DMFB constructed where the area corresponding to this bounding box is filled with ones. Such a matrix is called an electrodeWeight matrix. Electrodes closer to the source are given higher weights according to the intuition that security issues are easier to stop at the source. Fig. 8 illustrates a simplified example with one source dispense port ($D1$) and two destination mix modules ($M1, M2$). The corresponding matrices are

$$\text{electrodeWeight}_{1,1} = \begin{bmatrix} 3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 4 & 3 & 2 & 1 & 1 & 1 & 1 & 1 & 0 \\ 3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

Input: DMFB architecture A, set of sources S and targets T
Output: Matrix ranking each electrode, $arrayWeight$

```
arrayWeight ← 0
for each combination s ∈ S and t ∈ T do
    if timestep of t ≤ timestep of s then
        arrayWeight ← arrayWeight + electrodeWeight(s,t)
    end if
end for
return arrayWeight
```

Fig. 9. Electrode ranking pseudocode.

$$\text{electrodeWeight}_{1,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 2 & 1 & 1 & 1 & 1 & 1 & 0 \\ 3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \quad (13)$$

The weighting for assay as a whole, denoted as arrayWeight, is calculated by adding each combination of source and sink electrode weights as follows:

$$\begin{aligned} \text{arrayWeight} &= \sum_i \sum_j \text{electrodeWeight}(\text{source}(i), \text{target}(j)) \\ &= \text{electrodeWeight}_{1,1} + \text{electrodeWeight}_{1,2} \end{aligned} \quad (14)$$

$$= \begin{bmatrix} 3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 8 & 6 & 4 & 2 & 2 & 2 & 2 & 2 & 0 \\ 6 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 0 \\ 4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \quad (15)$$

The resultant ranking matrix satisfies the intuition; on a simple DMFB architecture, the best location to insert static checkpoints is immediately in front of a malicious droplet source. If the result of the heuristic algorithm produces identical rankings, the checkpoint placer should randomly select between them. Note that the result cannot guarantee that any routes are actually longer or more difficult than in the unaltered case, in contrast to the algorithm presented in the previous section. The electrode weighting approach is summarized in Fig. 9.

D. Temporal Randomization of Static Checkpoints

A static checkpoint can be monitored with some probability v instead of at every cycle. This will increase the probability of evasion, but there may be certain scenarios where this is an acceptable tradeoff for lower average power consumption. Let q be the number of static checkpoints, and Q be the number of static checkpoints on the malicious route. Then the probability of evasion can be modeled as

$$P(E) = (1-v)^Q \left(1 - \frac{ck}{s-q}\right)^{L-Q}. \quad (16)$$

If the static checkpoints are monitored 100% of the time, the probability of evasion is exactly zero, unless the malicious route does not cross any static checkpoints [$Q = 0$ leads to (5)]. When a malicious route crosses a static checkpoint instead of an electrode that is under random sampling, the following inequality must hold for there to be a net performance gain:

$$(1-v) \leq \left(1 - \frac{ck}{s-q}\right). \quad (17)$$

Rearranging, we find

$$v \geq c\left(\frac{k}{s-q}\right) \quad (18)$$

which can be interpreted as a tuning requirement. Recall that v and c are constants to be tuned by the system designer. Since the $k/(s-q)$ term is less than or equal to 1, v can be less than c while still lowering $P(E)$. Therefore, static checkpoints take less resources to implement than a randomized checkpoint for the same level of security while potentially increasing the difficulty of an attacker to minimize their route length.

E. Security of Checkpoint-Based Error Recovery

The concept of a checkpoint can be generalized in order to account for checkpoints inserted by error-recovery techniques. Thus, the security provided by the error-recovery system can be evaluated. We represent a general checkpoint as an ordered 7-tuple

$$\begin{aligned} C_j = <x(j), y(j), i(j), \text{vol}_{\text{low}}(j), \text{vol}_{\text{high}}(j) \\ \text{conc}_{\text{low}}(j), \text{conc}_{\text{high}}(i) > \end{aligned} \quad (19)$$

where x and y are the coordinates that the detection takes place, i is the actuation cycle that detection occurs, $\text{vol}_{\text{low}}(j)$ and $\text{vol}_{\text{high}}(j)$ define a valid interval of droplet volumes, and $\text{conc}_{\text{low}}(j)$, $\text{conc}_{\text{high}}(j)$ define a valid interval of concentrations. These interval specifications can be set to *don't-care* values simply by setting the low value to 0, and the high value to largest value perceptible by the imaging system.

We define an arrangement of checkpoints M as a set of k randomized checkpoints

$$M(i) = \{C_1, C_2, \ldots, C_k\} \quad (20)$$

where each x, y coordinate is chosen uniformly from the possible electrodes of the DMFB array, without replacement. These checkpoints are all active at the same cycle i. Testing for the presence of a droplet is specified by setting $\text{vol}_{\text{low}}(j)$ to the smallest droplet volume that can be manipulated by the DMFB hardware and $\text{vol}_{\text{high}}(j)$ to its maximum value. Testing for the absence of a droplet is specified by setting both $\text{vol}_{\text{low}}(j)$ and $\text{vol}_{\text{high}}(j)$ to zero. We do not consider the concentration in the checkpoint system, so concentration is set to *don't-care*. Finally, a randomized checkpoint system C_{rand} can be defined as a set of arrangements M, where each cycle i is selected by performing a *Bernoulli(c)* trial for each cycle of the assay execution. A checkpoint arrangement is added to the set for each success. The arrangement should be generated on-the-fly and with a high quality random number generator.

Based on this interpretation, it can be seen that the proposed security mechanism provides some level of error detection while error detection provides some measure of security. That is, they both can determine when the behavior of a DMFB system deviates from its intended operation. Correct attribution of an error is difficult in practice. For error recovery, it is

difficult to infer that a malicious adversary was the cause since faulty hardware could potentially produce the same result. On the other hand, in checkpoints used for detecting a malicious adversary, faulty hardware can lead to false positives.

Attributing a fault to either an attack or hardware failure can be done to some extent by analyzing the observed droplet behavior. For instance, if a droplet is detected at a specific point before any droplets specified by the actuation sequence have had a chance to reach it, an attack has almost certainly occurred since no failure mode is likely to have caused this behavior. Faults such as stuck droplets are more ambiguous; an attacker could easily induce this failure, but if historical reliability data for the DMFB platform indicates that stuck droplets are highly probable then the end user could reasonably conclude that the fault was caused by hardware failure. In general though, correct attribution of observed faults is nontrivial and end users may be required to consider extraneous factors such as how connected the device is, or whether poor access control policies are implemented.

VII. Experimental Results

The following case studies demonstrate the type of performance that can be expected from the described checkpoint system. Each biochip architecture and assay are drawn from the research literature, and realistic constraints are used to extract the performance metrics. The attacks are simulated using an open-source DMFB synthesis tool [50], [51] modified to incorporate our checkpoint techniques. Attacker routing is simulated using the Lee routing algorithm [46], which implements the attacker's optimal strategy of minimizing the malicious route. Monte Carlo simulation is used to validate the analysis, with the probability of evasion being given by the complement of the ratio of successful detections to number of attempted trials.

A. Realistic System Constraints

As a reminder, the motivation for a randomized checkpoint system is to lower the amount of resources required to monitor assay execution. The probability of evasion was found to decrease monotonically with the number of checkpoints monitored in a given time step. Therefore the best strategy is always to use as many checkpoints as possible. Before discussing the following case studies, it is instructive to investigate constraints on k imposed by realistic DMFB controller platforms.

The CCD camera provides sensor data in the form of an array of raw pixel values. Droplet presence, volume, and concentration can be extracted from these pixels through a pattern matching algorithm. A practical matching algorithm consists of focusing the CCD camera at specific points on the biochip, and calculating the correlation between the captured image and a template image [6]. Fig. 10 shows a *C* implementation of the correlation algorithm described in [6]. The reference template image is passed as an argument in array x while the cropped sub-image from the CCD is passed in array y. A larger correlation value means the two images are strongly related. Assuming a 25×25 template image and compiling the code to target the STM32F2x7 series of mid-range ARM Cortex-M3 microcontrollers, this code can require more than 28 500 clock cycles to execute. Operating the microcontroller at 120 MHz would thus require 238 μs to examine a single droplet. A typical operating frequency for a DMFB is 100 Hz, therefore no more than 42 droplets can be examined in a single actuation cycle.

```
#include <math.h>
#define T_SIZE 625
float cor(int x[T_SIZE], int y[T_SIZE]) {
  int num=0, den_x=0, den_y=0, sum_x=0, sum_y=0;
        int xavg, yavg;
        for(int j=0; j<T_SIZE; j++){
                  sum_x += x[j];
                  sum_y += x[j];
   }
        xavg = sum_x / T_SIZE;
        yavg = sum_y / T_SIZE;

  for(int i=0; i<T_SIZE; i++){
     num += (x[i]-xavg)*(y[i]-yavg);
     den_x += (x[i]-xavg)*(x[i]-xavg);
     den_y += (y[i]-yavg)*(y[i]-yavg);
  }
  return (float) num / sqrt((float) den_x*den_y);
}
```

Fig. 10. Droplet correlation code.

Based on this information, the following case studies will assume the checkpoint co-processor is limited to examining 20 checkpoints in each actuation cycle (i.e., $k + q = 20$) in order to leave some computing cycles for system overhead. In both cases, the probability of evasion will be evaluated with system parameters $c = 0.5$ and $v = 0.5$, which sets the probability that either static or random checkpoints are examined equal to a fair coin flip. The case where $c = 1$ and $v = 0.5$ is also investigated. The system should be capable of executing with both parameters equal to 1, but it may be desirable to reduce dynamic power consumption by throttling the amount of checking.

B. Commercial 3-Plex Immunoassay

Fig. 11 shows a commercial 3-plex immunoassay [52] DMFB with 1038 electrodes, excluding the dispense electrodes. This design is an application-specific biochip designed for acute myocardial infarction diagnosis. Despite the nonreconfigurable nature of this device, an attacker may still modify the actuation sequence to stall or introduce droplets into the reaction region. Droplets are dispensed from the routing region, processed in the reaction region and then sensed in the detection region. Error recovery is not implemented on this biochip.

The static checkpoint placement algorithms were run on this architecture, resulting in the placement maps indicated in Fig. 11. The difference between the results of the optimal versus heuristic placement algorithms are negligible. In both cases, it can be seen that they capture the intuitive notions of which electrodes are more important to monitor, which is near the dispense ports. We model an attacker who is interested in diluting the sample droplets so as to alter the final detected result. This type of attack has serious implications for quality of patient care. The malicious route is illustrated in dashed lines in the detail of Fig. 11; it attempts to dispense a reagent into one of the linear mixing regions.

1) Random Checkpoints Only: The malicious route takes nine steps to get from the reagent dispense port to the linear reaction chamber. This route length is minimal, and it

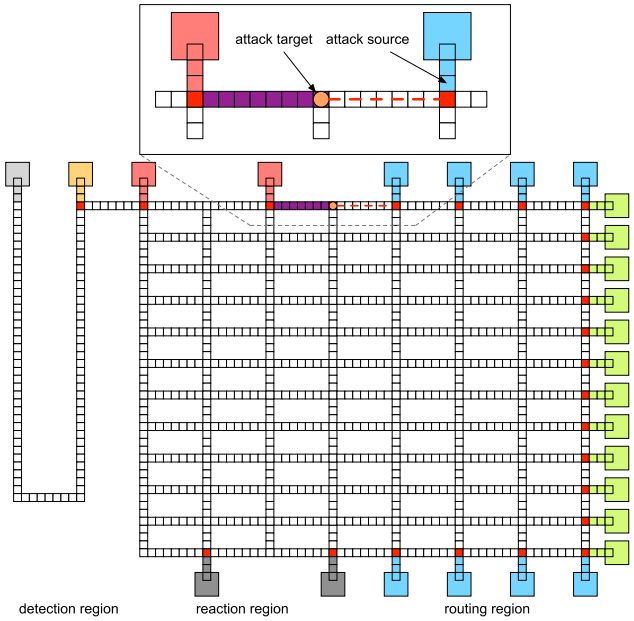


Fig. 11. Application-specific DMFB architecture for carrying out an n-plex immunoassay. The static checkpoint placer targets the outputs of the dispense ports (red electrodes). The heuristic algorithm results are identical when the top 23 electrodes are chosen.

is difficult for an attacker to choose an alternate route since this application-specific architecture provides only a single pathway for each sample to be processed. With the constraint of $k = 20$, the electrode coverage ratio is limited to 1.93%, which sets $P(E) = 0.92$ (Fig. 12). When $c = 1$ this decreases to $P(E) = 0.86$ (Fig. 13). The size of the DMFB is poorly matched with the system constraints, and results in poor performance. Either a faster controller should be selected, or static checkpoints should be used.

2) Random and Static Checkpoints: The addition of static checkpoints either through the provably secure or heuristic techniques results in the monitoring of the electrodes directly adjacent to the dispense ports. However, there are more dispense ports than checkpoints allowed, so the system designer is only permitted to choose a subset of the static checkpoints. One choice could be to place static checkpoints only within the dispense ports of the reagents and samples, so as to conserve fluids which may be expensive or difficult to obtain. The attacker is thus obligated to pass through a static checkpoint and be detected with high probability. Due to the unique architecture of this chip, there is no way for the attacker to route the attack in a way that avoids these checkpoints. For this particular arrangement and attack, $P(E) = 0.49$ (Fig. 12) decreasing to $P(E) = 0.47$ (Fig. 13) when $c = 1$. Most of the benefit comes from the static checkpoint, and we get much better performance without having to increase the checkpoint capacity of the controller.

C. Polymerase Chain Reaction

The polymerase chain reaction is used for the amplification of DNA and has been demonstrated on a number of microfluidic platforms. We study the PCR protocol executing on a general-purpose DMFB architecture under DoS attack. To evaluate the proposed system, we study the PCR assay under a

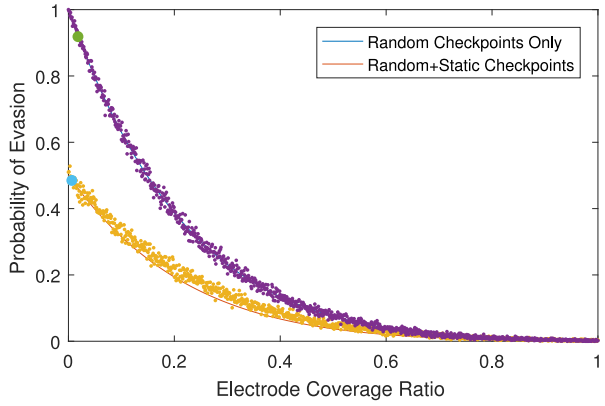


Fig. 12. Probability of evasion versus electrode coverage ratio for an attack causing dilution of the reaction chamber. The probability of sampling both random and static checkpoints is set to 50% ($c, v = 0.5$). Solid lines show analytic results, dotted lines are simulation data. Solid dots indicate operating points given the assumption of a maximum of 20 checkpoints.

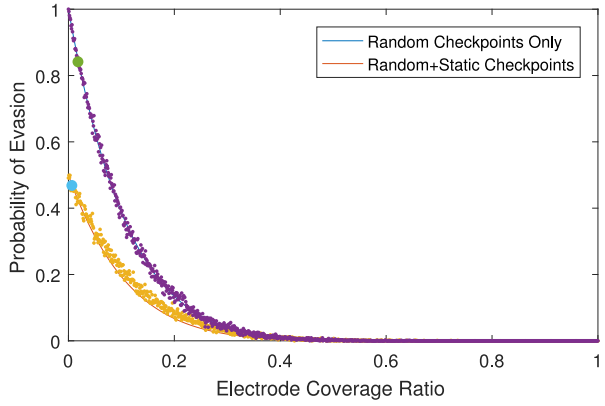


Fig. 13. When probability of monitoring random checkpoints increases to 100% ($c = 1$), performance increases only for higher electrode coverage ratios. The low electrode coverage ratio forces $P(E)$ to be somewhat high, with static checkpoints showing substantial benefit.

DoS attack. The goal of the attacker is to route a KCl droplet from the dispense port to mix module $M4$ (Fig. 14), as excess KCl concentration inhibits PCR [53]. The target module $M4$ is bound by the synthesis tool to mix KCl with Tris-HCl.

1) Random Checkpoints Only: The route taken by our malicious software router is shown in Fig. 14(a). It takes eight execution cycles to reach its destination, which is minimal since it is equal to the Manhattan distance. Note that it is possible to route the droplet through mix module 1 if the timing is chosen carefully; there is a small window of execution where the droplet being mixed in $M1$ is too far to be affected by the malicious droplet. Fig. 15 illustrates how $P(E)$ varies as a function of the electrode coverage ratio for the given route. With the given constraint of $k = 20$, the electrode coverage ratio is 10.3%. This route yields $P(E) = 0.66$.

2) Random and Static Checkpoints: We assume the adversary is able to learn about the static checkpoints, and thus is able to route around them. We fix $q = 16$ and use the heuristic algorithm in Section VI to place the static checkpoints. The result is that all the dispense ports are covered. The provably secure placer gives the same result. With q set and total checkpoints limited to 20, $k = 4$. With $v = 0.5$, the inequality in (18) is satisfied, so the optimal strategy for the adversary is

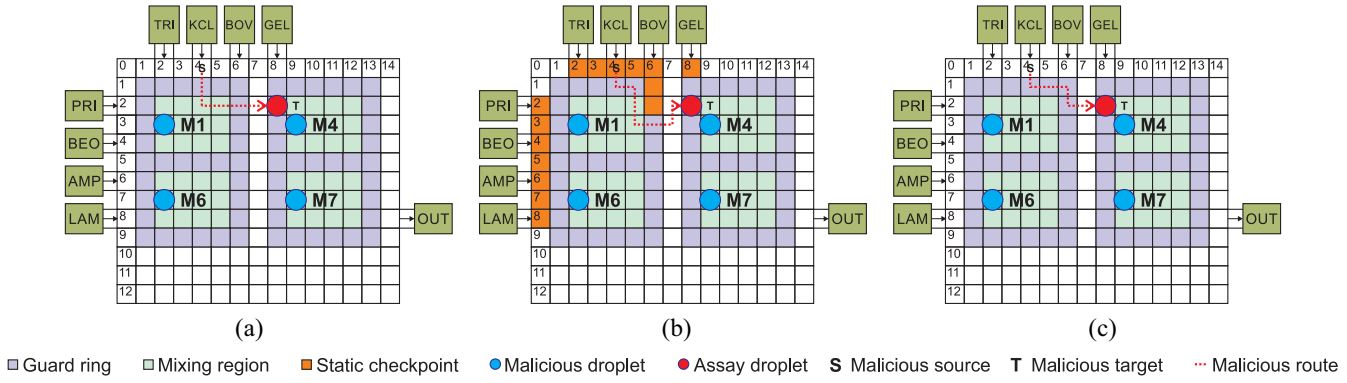


Fig. 14. (a) Random checkpoints only. Malicious route from KCl port targets $M4$ mix module. No obstacles mean that the adversary is free to route with minimal distance, minimizing probability of evasion. (b) Random and static checkpoints. The minimum Manhattan distance is no longer achievable, decreasing the probability of evasion. (c) Random checkpoints with error recovery. Malicious route is redirected to avoid mixing region $M1$ due to error recovery checkpoints. While the original route is no longer achievable, there still exists a minimal path from the source to target.

to route around the static checkpoints if possible. Fig. 14(b) illustrates the placement of the top 16 static checkpoints and the path chosen by the router. Note that the malicious route is forced to cross one static mine ($Q = 1$), and then takes nine cycles to reach the target ($L = 9$). Fig. 15 shows $P(E)$ for this longer route as being lower than the original route for all electrode coverage ratios. Blocking off the dispense port provides a tremendous advantage. In this case, the electrode coverage ratio is 2.2% and $P(E) = 0.45$. Note that if v had been set to 1, this attack would have been detected immediately.

3) Random Checkpoints With Error Recovery Checkpoints: Now we investigate how error-recovery checkpoints interact with the system. A malicious route attempting to cross through an error-recovery checkpoint would immediately trigger the error recovery process. Error-recovery checkpoints are placed at critical junctions in the protocol specification such as mix operations. Assuming the attacker's goal has not changed, the route must now move around the region defined by $M1$, which is being actively monitored [Fig. 14(c)]. The malicious droplet joins the droplet being mixed in $M4$. This is detectable depending on the how the error recovery system is setup. If the error thresholds are not set properly, changing the concentration of KCl this way is conceivable. The route length is the same as the case in Fig. 2. Thus the error recovery mechanism provides no advantage in terms of detecting malicious droplets in transit, and $P(E)$ is exactly the same as in the case without error recovery. We note that error recovery mechanisms can provide an advantage in indirectly detecting attacks, if the attackers are not careful in staying within error thresholds.

D. Discussion

The probability of evasion achieved in the immunoassay and PCR case studies provide a strong disincentive for would-be attackers. A probability of evasion equal to a fair coin flip is easy to obtain and can be achieved even if the electrode coverage ratio is less than 10% (Fig. 15), using both random and static checkpoints being monitored with 50% probability each cycle. Forcing the random checkpoints to be monitored on every cycle causes the probability of evasion curves to drop (Figs. 13 and 16). The situation only gets worse for the attacker the longer an attack takes place, as the lifetime L gives exponentially lower probability of evasion. Even if we consider other types of attacks where droplets are routed between

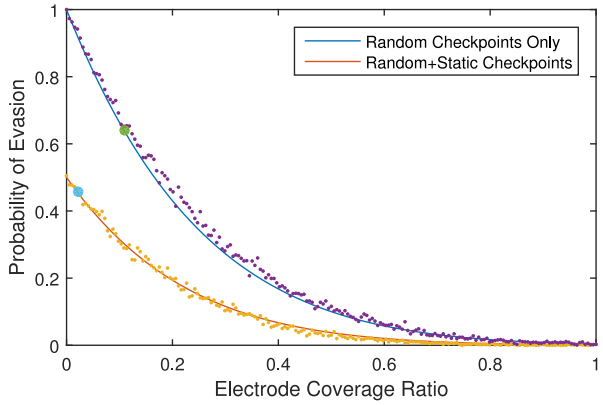


Fig. 15. Probability of evasion versus electrode coverage ratio for a minimal length route and a route with static checkpoints attempting to dispense KCl into $M4$. Probability that a given cycle is monitored was set to 50% for both random and static checkpoints. The two large dots indicate data points from the PCR case study, where the number of checkpoints is limited to 20.

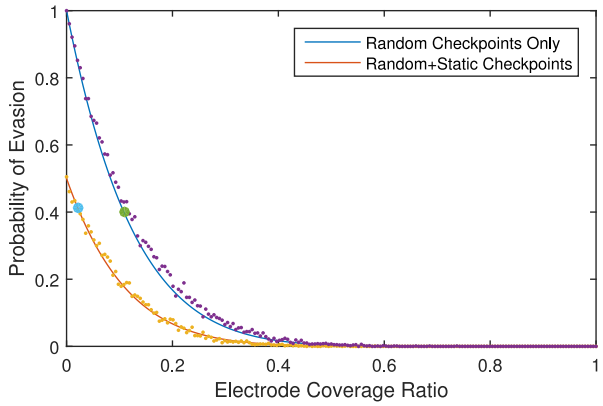


Fig. 16. Performance improves when the probability of monitoring random checkpoints at each cycle is increased to 100%. Probability of monitoring static checkpoints is 50%. The two large dots indicate data points from the PCR case study, where the number of checkpoints is limited to 20. $P(E)$ drops to 0.40 at 10.3% coverage ratio for random checkpoints only.

modules where L is lower, and consequently, $P(E)$ is higher, evaluation of probabilistic models likely underestimates the real-world effect of implementing such a system. The fact that a randomized checkpoint system exists is a deterrent to any would-be attackers.

VIII. Conclusion

This paper presented the analysis and design of an intrusion detection system targeting malicious modification of DMFB cyberphysical systems. We analyzed a randomized checkpoint system that utilizes CCD camera technology to monitor the real-time execution of an assay. We showed that the probability of evasion is largely determined by the length of the malicious route, and that the uniform distribution is secure for general-purpose DMFB arrays. Static checkpoints were introduced in order to influence malicious droplets to take a circuitous path while enforcing critical electrodes, and both provably secure and heuristic placement algorithms were presented. We demonstrated how existing error recovery schemes contribute to the security of the DMFB system. The concept was implemented using an open-source DMFB simulation tool and evaluated using a software malicious router on a real-world architecture designed for immunoassays and on a general-purpose biochip running a PCR assay. The simulation results evaluated the probability of an attack evading detection, and the results supported the design intuition and analysis presented.

Interpretation of the probability of evasion is not straightforward, and depends on several factors outside the control of the system designer. Certain applications can be expected to have a higher tolerance for risk than others. For instance, assays used for the determination of medical decisions likely has a much lower threshold for probability of evasion than for a lone researcher working on a single experiment. Looking from the perspective of an attacker, the probability of evasion must be sufficiently high since the consequences of being detected may be devastating; a company conducting corporate sabotage needs to be near absolutely certain than an attack cannot be detected. Furthermore, a finite probability of detection would detract from the cost-benefit analysis of developing sophisticated hacking techniques—the Stuxnet worm is believed to have been developed with the resources of a nation-state [54]. If the attackers had to target PLC controllers with security hardware in place, they may have focused on other, perhaps nontechnical, means of thwarting Iran's nuclear program. Thus an intrusion detection system provides a great disincentive from would-be attackers.

Future work in cyberphysical DMFB security could proceed along several lines.

1) *Usage Model:* The fact that the security co-processor requires its own separate assay specification (Fig. 3) complicates the set-up phase for the end user and manufacturer. It is conceivable that the security co-processor could execute using only knowledge of typical usage patterns instead of comparing directly to the assay specification, but this would lead to false positives and make the design of the system more similar to traditional intrusion detection systems.
2) *Poisoning Attacks:* Poisoning attacks refer to the compromise of data used to build machine learning models [55], [56]. While our randomized checkpoint system is an example of an expert system, a similar concept could apply here in that the co-processor's assay specification could be compromised. Addressing this expanded threat model would provide better assurances of security in the real world.
3) *Attribution:* Determination of whether an observed fault was caused by an attack or hardware failure, as mentioned in Section VI-E, could be systematized. There are likely observed phenomenon that can be attributed with certainty, while others may only be attributable up to some probability.
4) *Attack Tolerance:* Hardware should be able to continue to function correctly despite the presence of an attack. Attack-tolerance design techniques will likely overlap with research on reliable and fault-tolerant hardware design.

The authors hope that this paper will provide a stepping stone into further research on DMFB security, which is needed at this critical time in the maturation of DMFB technology—it is still considered an emerging technology but is on a cusp of becoming mainstream. In Internet security, it is often asked what could have been done differently in the Internet's early technical design that may have prevented some of the complex security issues we face today. DMFB designers can perform a similar thought experiment with the luxury of being able to act upon it. The long term viability of DMFB technology depends on such security-minded thinking.

References

[1] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 4, 2008, Art. no. 1.
[2] V. Agarwal *et al.*, "Reservoir and mixer constrained scheduling for sample preparation on digital microfluidic biochips," in *Proc. Asia South Pac. Design Autom. Conf.*, Chiba, Japan, 2017, pp. 702–707.
[3] S. Windh, C. Phung, D. T. Grissom, P. Pop, and P. Brisk, "Performance improvements and congestion reduction for routing-based synthesis for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 1, pp. 41–54, Jan. 2017.
[4] F. Su and K. Chakrabarty, "Module placement for fault-tolerant microfluidics-based biochips," *ACM Trans. Design Autom. Electron. Syst.*, vol. 11, no. 3, pp. 682–710, 2006.
[5] P. Pop, M. Alistar, E. Stuart, and J. Madsen, "Design methodology for digital microfluidic biochips," in *Fault-Tolerant Digital Microfluidic Biochips: Compilation and Synthesis*. Cham, Switzerland: Springer, 2016, pp. 13–28.
[6] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 59–72, Jan. 2013.
[7] K. Hu *et al.*, "Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Atlanta, GA, USA, 2015, pp. 1–4.
[8] T. Xu and K. Chakrabarty, "Parallel scan-like test and multiple-defect diagnosis for digital microfluidic biochips," *IEEE Trans. Biomed. Circuits Syst.*, vol. 1, no. 2, pp. 148–158, Jun. 2007.
[9] T. A. Dinh, S. Yamashita, T.-Y. Ho, and K. Chakrabarty, "A general testing method for digital microfluidic biochips under physical constraints," in *Proc. IEEE Int. Test Conf.*, Anaheim, CA, USA, 2015, pp. 1–8.
[10] C. C.-Y. Lin and Y.-W. Chang, "ILP-based pin-count aware design methodology for microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 9, pp. 1315–1327, Sep. 2010.
[11] J. McDaniel, Z. Zimmerman, D. Grissom, and P. Brisk, "PCB escape routing and layer minimization for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 1, pp. 69–82, Jan. 2017.
[12] *Illumina Neoprep Library Prep System*, Illumina, San Diego, CA, USA, 2016. [Online]. Available: http://www.illumina.com/systems/neoprep-library-system.html/
[13] J. Evans, "Global biochip markets: Microarrays and lab-on-a-chip," BCC Res., Wellesley, MA, USA, Tech. Rep. BIO049F, Apr. 2016.
[14] B. Krebs. *Hacked Cameras, DVRs Powered Today's Massive Internet Outage*. Accessed: Aug. 2, 2017. [Online]. Available: https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/
[15] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002.

[16] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab Chip*, vol. 2, no. 2, pp. 96–101, 2002.
[17] H.-H. Shen, S.-K. Fan, C.-J. Kim, and D.-J. Yao, "EWOD microfluidic systems for biomedical applications," *Microfluidics Nanofluidics*, vol. 16, no. 5, pp. 965–987, 2014.
[18] K. Choi, A. H. C. Ng, R. Fobel, and A. R. Wheeler, "Digital microfluidics," *Annu. Rev. Anal. Chem.*, vol. 5, no. 1, pp. 413–440, 2012.
[19] Y. Luo, K. Chakrabarty, and T.-Y. Ho, *Hardware/Software Co-Design and Optimization for Cyberphysical Integration in Digital Microfluidic Biochips*. New York, NY, USA: Springer, 2014.
[20] Y.-J. Shin and J. B. Lee, "Machine vision for digital microfluidics," *Rev. Sci. Instrum.*, vol. 81, no. 1, 2010, Art. no. 014302.
[21] C.-L. Sotiropoulou *et al.*, "Real-time machine vision FPGA implementation for microfluidic monitoring on lab-on-chips," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 2, pp. 268–277, Apr. 2014.
[22] D. Witters, K. Knez, F. Ceyssens, R. Puers, and J. Lammertyn, "Digital microfluidics-enabled single-molecule detection by printing and sealing single magnetic beads in femtoliter droplets," *Lab Chip*, vol. 13, no. 11, pp. 2047–2054, 2013.
[23] G.-R. Lu *et al.*, "On reliability hardening in cyber-physical digital-microfluidic biochips," in *Proc. Asia South Pac. Design Autom. Conf.*, Chiba, Japan, 2017, pp. 518–523.
[24] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security assessment of cyberphysical digital microfluidic biochips," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 3, pp. 445–458, May/Jun. 2016.
[25] J. Tang, R. Karri, M. Ibrahim, and K. Chakrabarty, "Securing digital microfluidic biochips by randomizing checkpoints," in *Proc. IEEE Int. Test Conf.*, Fort Worth, TX, USA, 2016, pp. 1–8.
[26] S. S. Ali, M. Ibrahim, J. Rajendran, O. Sinanoglu, and K. Chakrabarty, "Supply-chain security of digital microfluidic biochips," *Computer*, vol. 49, no. 8, pp. 36–43, 2016.
[27] C.-W. Hsieh, Z. Li, and T.-Y. Ho, "Piracy prevention of digital microfluidic biochips," in *Proc. Asia South Pac. Design Autom. Conf.*, Chiba, Japan, 2017, pp. 512–517.
[28] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.
[29] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Des. Test. Comput.*, vol. 27, no. 1, pp. 10–25, Jan./Feb. 2010.
[30] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Leuven, Belgium, 2012, pp. 23–40.
[31] F. Koushanfar and A. Mirhoseini, "A unified framework for multimodal submodular integrated circuits trojan detection," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 162–174, Mar. 2011.
[32] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *Proc. IEEE Int. Workshop Hardw. Orient. Security Trust*, Anaheim, CA, USA, Jun. 2008, pp. 51–57.
[33] J. Rajendran, V. Jyothi, O. Sinanoglu, and R. Karri, "Design and analysis of ring oscillator based design-for-trust technique," in *Proc. VLSI Test Symp.*, Dana Point, CA, USA, May 2011, pp. 105–110.
[34] A. Waksman and S. Sethumadhavan, "Silencing hardware backdoors," in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, May 2011, pp. 49–63.
[35] M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith, "Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically," in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, May 2010, pp. 159–172.
[36] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proc. IEEE*, vol. 102, no. 8, pp. 1283–1295, Aug. 2014.
[37] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proc. IEEE*, vol. 102, no. 8, pp. 1229–1247, Aug. 2014.
[38] J. Rajendran, O. Sinanoglu, and R. Karri, "Regaining trust in VLSI design: Design-for-trust techniques," *Proc. IEEE*, vol. 102, no. 8, pp. 1266–1282, Aug. 2014.
[39] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.
[40] P. A. Williams and A. J. Woodward, "Cybersecurity vulnerabilities in medical devices: A complex environment and multifaceted problem," *Med. Devices Evidence Res.*, vol. 8, pp. 305–316, Jul. 2015.
[41] Z. Hua *et al.*, "Multiplexed real-time polymerase chain reaction on a digital microfluidic platform," *Anal. Chem.*, vol. 82, no. 6, pp. 2310–2316, 2010.
[42] V. Srinivasan, "A digital microfluidic lab-on-a-chip for clinical diagnostic applications," Ph.D. dissertation, Dept. Elect. Comput. Eng., Duke Univ., Durham, NC, USA, 2005.
[43] S. Kennedy and N. Oswald, *PCR Troubleshooting and Optimization: The Essential Guide*. Norfolk, U.K.: Caister Acad. Press, 2011.
[44] The Wall Street Journal. (Mar. 2016). *Theranos Results Could Throw Off Medical Decisions, Study Finds*. [Online]. Available: http://www.wsj.com/articles/theranos-results-could-throw-off-medical-decisions-study-finds-1459196177?mod=e2fb
[45] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proc. IEEE*, vol. 100, no. 11, pp. 3056–3076, Nov. 2012.
[46] C. Y. Lee, "An algorithm for path connections and its applications," *IEEE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 346–365, Sep. 1961.
[47] F. O. Hadlock, "A shortest path algorithm for grid graphs," *Networks*, vol. 7, no. 4, pp. 323–334, 1977.
[48] J. Hao and J. B. Orlin, "A faster algorithm for finding the minimum cut in a graph," in *Proc. ACM SIAM Symp. Discrete Algorithms*, Orlando, FL, USA, 1992, pp. 165–174.
[49] M. Stoer and F. Wagner, "A simple min cut algorithm," in *Proc. Eur. Symp. Algorithms*, Utrecht, The Netherlands, 1994, pp. 141–147.
[50] D. Grissom *et al.*, "A digital microfluidic biochip synthesis framework," in *Proc. IEEE/IFIP Int. Conf. VLSI Syst. Chip*, Santa Cruz, CA, USA, Oct. 2012, pp. 177–182.
[51] D. Grissom and P. Brisk, "A field-programmable pin-constrained digital microfluidic biochip," in *Proc. IEEE/ACM Design Autom. Conf.*, Austin, TX, USA, 2013, pp. 1–9.
[52] R. Sista *et al.*, "Development of a digital microfluidic platform for point of care testing," *Lab Chip*, vol. 8, no. 12, pp. 2091–2104, 2008.
[53] R. Higuchi, C. Fockler, G. Dollinger, and R. Watson, "Kinetic PCR analysis: Real-time monitoring of DNA amplification reactions," *Biotechnology*, vol. 11, no. 9, pp. 1026–1030, 1993.
[54] N. Anderson. (Jun. 2012). *Confirmed: U.S. and Israel Created Stuxnet, Lost Control of It*. [Online]. Available: https://arstechnica.com/tech-policy/2012/06/confirmed-us-israel-created-stuxnet-lost-control-of-it/
[55] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, Apr. 2014.
[56] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 6, pp. 1893–1905, Nov. 2015.

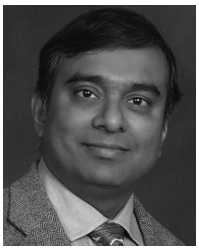
Jack Tang (S'14) received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, CA, USA, in 2006 and the M.S. degree in electrical engineering from San José State University, San Jose, CA, USA, in 2012. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY, USA.

His current research interests include cyberphysical microfluidic systems, MEMS, analog circuit design, and their application toward secure and trustworthy hardware.

Mohamed Ibrahim (S'13) received the B.Sc. (Hons.) degree in electrical engineering and the M.Sc. degree from Ain Shams University, Cairo, Egypt, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA.

He was a design-for-test intern with Intel Corporation, Santa Clara, CA, USA, and Intel Corporation, Austin, TX, USA. He was also a Visiting Scholar with the Technical University of Munich, Munich, Germany and the University of Bremen, Bremen, Germany. His current research interests include design automation of microfluidic systems for biomolecular analysis, security of microfluidic biochips, and design-for-test of emerging technologies.

Mr. Ibrahim was a recipient of the Best Paper Award at the 2017 IEEE/ACM Design, Automation, and Test in Europe (DATE) Conference. He is a student member of ACM.



Krishnendu Chakrabarty (F'08) received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, Kharagpur, India, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1992 and 1995, respectively.

He is currently the William H. Younger Distinguished Professor and Chair of the Department of Electrical and Computer Engineering and a Professor of Computer Science with Duke University, Durham, NC, USA. He holds 10 U.S. patents, with several patents pending. His current research interests include testing and design-for-testability of integrated circuits and systems, digital microfluidics, biochips, and cyberphysical systems, data analytics for fault diagnosis, failure prediction, anomaly detection, and hardware security, smart manufacturing.

Prof. Chakrabarty was a recipient of the National Science Foundation CAREER Award, the Office of Naval Research Young Investigator Award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS Donald O. Pederson Best Paper Award in 2015, the *ACM Transactions on Design Automation of Electronic Systems* Best Paper Award in 2016, the 2008 Duke University Graduate School Dean's Award for excellence in mentoring and the 2010 Capers and Marion McDonald Award for Excellence in Mentoring and Advising, Pratt School of Engineering, Duke University, and over a dozen best paper awards at major conferences. He was also a recipient of the IEEE Computer Society Technical Achievement Award in 2015, the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award in 2017, and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur in 2014. He is a Research Ambassador of the University of Bremen, Germany and a Hans Fischer Senior Fellow (named after Nobel Laureate Prof. Hans Fischer) with the Institute for Advanced Study, Technical University of Munich, Germany. He served as the Editor-in-Chief of IEEE DESIGN & TEST OF COMPUTERS from 2010 to 2012 and the *ACM Journal on Emerging Technologies in Computing Systems* from 2010 to 2015. He currently serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is also an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON MULTISCALE COMPUTING SYSTEMS, and *ACM Transactions on Design Automation of Electronic Systems*. He is a fellow of ACM and a Golden Core Member of the IEEE Computer Society. He was a 2009 Invitational Fellow of the Japan Society for the Promotion of Science. He has served as a Distinguished Visitor of the IEEE Computer Society from 2005 to 2007 and from 2010 to 2012), a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2006 to 2007 and from 2012 to 2013, and an ACM Distinguished Speaker from 2008 to 2016.

Ramesh Karri (SM'11) received the Ph.D. degree in computer science and engineering from the University of California, San Diego, CA, USA. He is currently a Professor of Electrical and Computer Engineering with New York University, Brooklyn, NY, USA.

He is the Area Director of Cybersecurity with the New York State Center for Advanced Technology in Communications, the Hardware Security Leader of the NYU Center for Interdisciplinary Studies in Security and Privacy, and the Co-Founder of the NYU Center for Cybersecurity and Trust-Hub. His research and education activities span hardware cybersecurity including trustworthy ICs, processors and cyberphysical systems, security-aware computer-aided design, test, verification, validation and reliability, nanotechnology security, metrics, benchmarks, hardware cybersecurity competitions, and additive manufacturing security. He has authored over 200 journal and conference publications, including tutorials on Trustworthy Hardware in IEEE Computer and the Proceedings of the IEEE. His group's work on hardware cybersecurity was nominated for best paper awards at ICCD 2015 and DFTS 2015, and received awards at ITC 2014, CCS 2013, DFTS 2013, and VLSI Design 2012. His group has also received awards at ACM Student Research Competitions at DAC 2012, ICCAD 2013, DAC 2014, ACM Grand Finals 2013, Kaspersky Challenge, and Embedded Security Challenge.

Prof. Karri is a recipient of the Humboldt Fellowship and the National Science Foundation CAREER Award. He organizes the annual Embedded Security Challenge, a red-team/blue-team hardware security competition at NYU. He co-founded and served as the Chair of the IEEE Computer Society Technical Committee on Nanoscale Architectures (NANOARCH). He co-founded and serves on the Steering Committee of the IEEE/Association of Computing Machinery (ACM) Symposium on NANOARCH. He served as the Program Chair and General Chair of several conferences, including the IEEE International Conference on Computer Design (ICCD), the IEEE Symposium on Hardware Oriented Security and Trust (HOST), the IEEE Symposium on Defect and Fault-Tolerant Nano VLSI Systems (DFTS), IEEE/ACM NANOARCH, RFIDsec, and ACM WiSec. He serves on several program committees, including DAC, ICCAD, HOST, ITC, VTS, ETS, ICCD, DTIS, and WIFS. He is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, *ACM Journal of Emerging Technologies in Computing*, *ACM Transactions on Design Automation of Electronic Systems*, the IEEE ACCESS, the IEEE TRANSACTIONS ON EMERGING TECHNOLOGIES IN COMPUTING, IEEE DESIGN AND TEST, and IEEE EMBEDDED SYSTEMS LETTERS. He also served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2010 to 2014. He was an IEEE Computer Society Distinguished Visitor from 2013 to 2015. He is on the Executive Committee of IEEE/ACM Design Automation Conference, initiating and leading the Security@DAC initiative. He has delivered invited keynotes, talks, and tutorials on Hardware Security and Trust at venues such as ESRF, DAC, DATE, VTS, ITC, ICCD, NATW, LATW, and CROSSING.