Security Assessment of Cyberphysical Digital Microfluidic Biochips

Sk Subidh Ali, Mohamed Ibrahim, Ozgur Sinanoglu, Krishnendu Chakrabarty, and Ramesh Karri

Abstract—A digital microfluidic biochip (DMFB) is an emerging technology that enables miniaturized analysis systems for point-of-care clinical diagnostics, DNA sequencing, and environmental monitoring. A DMFB reduces the rate of sample and reagent consumption, and automates the analysis of assays. In this paper, we provide the first assessment of the security vulnerabilities of DMFBs. We identify result-manipulation attacks on a DMFB that maliciously alter the assay outcomes. Two practical result-manipulation attacks are shown on a DMFB platform performing enzymatic glucose assay on serum. In the first attack, the attacker adjusts the concentration of the glucose sample and thereby modifies the final result. In the second attack, the attacker tampers with the calibration curve of the assay operation. We then identify denial-of-service attacks, where the attacker can disrupt the assay operation by tampering either with the droplet-routing algorithm or with the actuation sequence. We demonstrate these attacks using a digital microfluidic synthesis simulator. The results show that the attacks are easy to implement and hard to detect. Therefore, this work highlights the need for effective protections against malicious modifications in DMFBs.

Index Terms—Cyber physical digital microfludic biochip, security, Trojan, denial of service attack, tampering, DMFB, in-vitro, DMFB synthesis

1 Introduction

Microfluidics technology has tremendous potential to influence subject areas from biochemical synthesis to information technology through the use of miniaturized devices for biomolecular recognition [1]. The early generation of these devices, referred to as continuous-flow microfluidic biochips [2], [3], consist of microfabricated channels, pumps, and valves which are permanently etched in a silicon or a glass substrate. Since the structure and the functionality of such devices are tightly coupled, each system is only applicable to a narrow class of applications, thus limiting the scalability and reconfigurability of this technology.

In contrast, a digital microfluidic biochip (DMFB) is a reconfigurable lab-on-a-chip technology that has achieved remarkable success in enabling miniaturized analysis systems for several biochemical applications, such as point-of-care clinical diagnostics [4], [5], DNA sequencing [6], and environmental monitoring [7]. DMFB technology allows us to manipulate multiple droplets of microliter to picoliter volumes under program control on a patterned electrode array. Benefits of miniaturization using DMFBs include reduced reagent consumption and sample requirement (which is critical in point-of-care setting), reduced analysis time due to the increased reaction speed, human-intervention-free control of droplets via design automation, and low risk of contamination. These benefits make DMFB technology an ideal alternative to conventional benchtop biochemical procedures [8].

Over the past decade, many techniques have been developed to address various aspects of automated design and optimization of DMFBs [9]. Methods for architectural-level synthesis [10], module placement [11], and droplet routing [12], [13] have been proposed and they have been combined to pave the way for Computer aided design (CAD) flows for DMFBs [14]. With the help of these CAD tools, DMFB users, including chemists, doctors, and clinicians, can adapt more easily to this emerging technology. Moreover, designers can be freed from cumbersome and labor-intensive work as they can concentrate more on meeting design targets, enhancing DMFB yield and reliability, and reducing manufacturing cost.

An equally important aspect of DMFB design is the integration of sensors, which is the cornerstone for the development of physical-aware control systems [15]. The progress of fluidic sample preparation and chemical reactions can be monitored using integrated waveguides [16], capacitive sensors [8], or CCD cameras [17]. The availability of sensor readouts in order to dynamically reconfigure DMFB operations in real-time enables more robust assay execution on an integrated microfluidic platform [18].

Despite the advantages offered by digital microfluidics for clinical diagnosis, immunoassays and DNA sequencing, there has been no study on the potential security implications of DMFBs. Recent cyberattacks have revealed the vulnerabilities of automated systems [19], [20], [21].

A fully automated DMFB is typically controlled by a computer, which applies a set of control sequences on the input pads of a DMFB. If an attacker gets control of the DMFB, he can maliciously modify the assay operation to either manipulate the assay outcome or can disrupt the assay operation. The attacker could be a person who wants to jeopardize another person's health by manipulating his/her

- *S.S. Ali and O. Sinanoglu are with New York University, Abu Dhabi, United Arab Emirates. E-mail: subidh@gmail.com, os22@nyu.edu.*
- *M. Ibrahim and K. Chakrabarty are with Duke University, Durham, NC, 27708. E-mail: {mohamed.s.ibrahim, krish}@duke.edu.*
- *R. Karri is with New York University, Brooklyn, NY 11201. E-mail: rkarri@poly.edu.*

Manuscript received 7 Sept. 2015; revised 23 Oct. 2015; accepted 9 Nov. 2015. Date of publication 17 Dec. 2015; date of current version 1 June 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2015.2509991

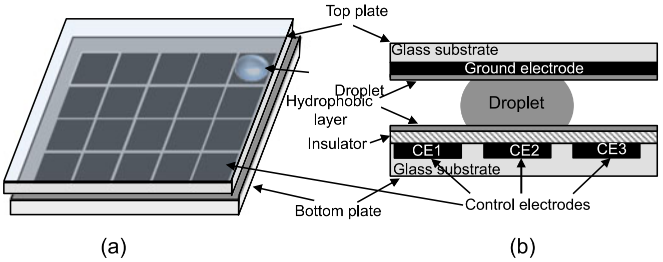


Fig. 1. Schematic view of a DMFB: (a) A DMFB with a 2D array of electrodes and (b) a side-view of the DMFB [24].

clinical diagnostic results. An organization can be adversarial and can disrupt the products from a specific vendor.

In this paper, we assess the security vulnerabilities associated with a DMFB platform. The key contributions of this paper, especially beyond what is presented in its preliminary version [22], are as follows

1) We define the first comprehensive attack framework and assess the security of a DMFB from all possible malicious aspects that relate to a potential attacker. The paper identifies strengths and weaknesses of an attacker based on his role in the design, manufacturing, and use of the DMFB.
2) We broadly identify all possible attack types, from assay-outcome manipulation to functionally disrupting the DMFB. We describe the basic security vulnerabilities for all these attacks.
3) We identify the challenges that arise from attacks in the presence of error-correction and error-recovery mechanisms in cyberphysical DMFBs. We show how to overcome these challenges and develop attacks on cyberphysical DMFBs with built-in error-recovery.
4) We demonstrate denial-of-service (DoS) attacks on a DMFB to show how an attacker can disrupt assay operations.

The remainder of the paper is organized as follows. Section 2 describes the background of the work. Section 3 assesses threats associated with DMFBs. Section 4 describes attacks on enzymatic glucose assay. Section 5 details attacks on DMFBs in the presence of built-in error-recovery mechanisms. Section 6 demonstrates denial-of-service attacks on a DMFB. Section 7 concludes the paper.

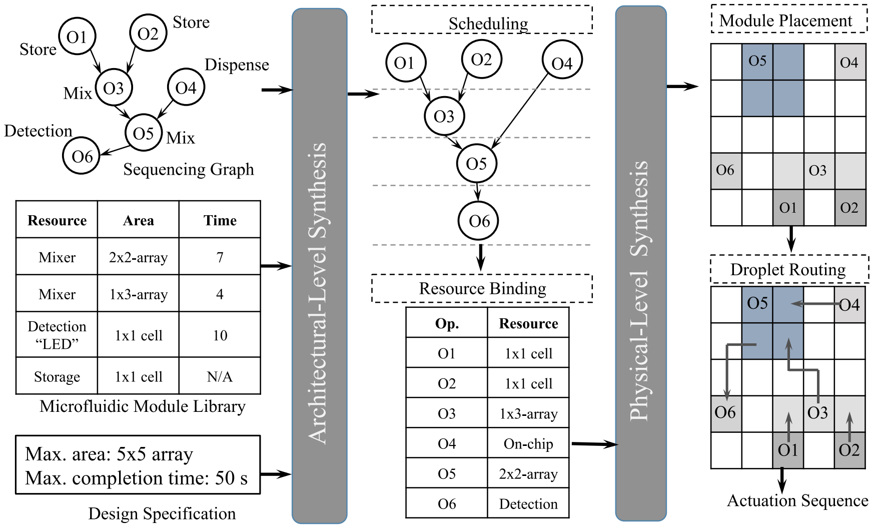
2 Background

Fig. 1 shows the schematic of a typical DMFB that consists of a two-dimensional electrode array, on-chip reservoirs, and sensors. A basic cell in a DMFB consists of two parallel plates (Fig. 1b). The electrode surface is coated with a thin layer of an insulator such as Paralyene [4]. Both plates are also coated with a thin film to provide a hydrophobic platform that is necessary for smooth droplet actuation. The gap between the top and bottom plates is usually filled with silicon oil which acts as a filler medium, preventing droplet evaporation and reducing surface contamination [8]. When an electric field V is applied between the parallel plates of a DMFB, the interfacial surface energies are modulated and an electrical double layer is created, which in turn alters the apparent contact angle $\theta(V)$ of a conductive liquid droplet that is in contact with the hydrophobic surface (Fig. 1b). The change in the contact angle, in turn, influences the wetting behavior of the droplet. This phenomenon is known as electrowetting-on-dielectric, and it can be modeled using the Lippmann-Young equation:

$$\cos\theta(V) = \cos\theta(0) + \frac{\epsilon_0 \epsilon_r V^2}{2d\gamma_{LG}}, \tag{1}$$

where γ_{LG} is the liquid-gas interfacial tension, ϵ_0 is the permittivity of vacuum, ϵ_r is the permittivity of the bottom insulator, and d is its thickness [23].

Using DMFB CAD tools, a high-level assay specification is converted into an actuation sequence that runs the DMFB. Fig. 2 highlights the overall CAD flow for DMFBs. First the high-level assay specification is converted into a *sequencing graph* $G = (V, E)$, where a node $v \in V$ corresponds to a fluid-handling operation (e.g., dispensing, mixing, dilution, and detection) and an edge $e \in E$ between two nodes $(v1, v2)$ represents the dependency between them. Besides this sequencing graph, the design specifications and the module library of the DMFB are inputs to the CAD tools [9]. The design specification specifies the maximum size of the

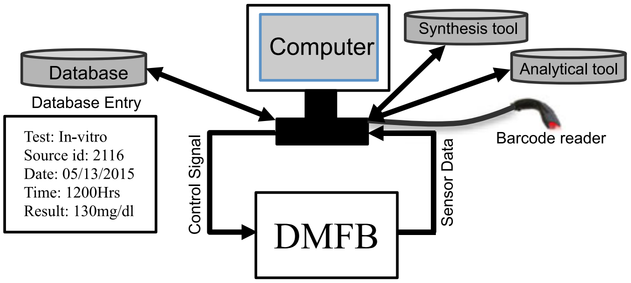


Resource	Area	Time
Mixer	2x2-array	7
Mixer	1x3-array	4
Detection "LED"	1x1 cell	10
Storage	1x1 cell	N/A

Max. area: 5x5 array
Max. completion time: 50 s

Op.	Resource
O1	1x1 cell
O2	1x1 cell
O3	1x3-array
O4	On-chip
O5	2x2-array
O6	Detection

Fig. 2. CAD flow for automated design of a DMFB.



Test: In-vitro
Source id: 2116
Date: 05/13/2015
Time: 1200Hrs
Result: 130mg/dl

Fig. 3. A fully automated DMFB platform.

microfluidic array and the upper limit on the assay completion time. The module library includes different microfluidic functional modules, such as mixer, storage unit along with their parameters, such as width, length, and operation duration. This module library is analogous to the standard/custom cell library used in cell-based VLSI design.

During architectural-level synthesis [10], resource binding and scheduling operations are performed. In resource binding, the assay operations are mapped onto the available functional resources. Once the resource binding is carried out, one can determine the start time and the end time for all the operations of the assay. Then the scheduler schedules the operations according to the precedence constraints imposed by the sequencing graph.

In physical-level synthesis, placement determines the location of the microfluidic modules, such as integrated optical detectors and reservoirs/dispensing ports, in a two-dimensional microfluidic array [11]. The two-dimensional array at the top-right in Fig. 2 shows the placed modules in the array. Once the placement is carried out, the routing algorithm determines the optimal routes of individual droplets of the assay operation subject to the scheduling constraints. It also accounts for fluidic constraints, such as the minimum distance between droplets, to prevent accidental mixing of droplets. The output of the droplet routing step is the actuation sequence, which stores the droplet movement control information at each time step. The status of an individual control signal in a given time step is “1” (actuated), “0” (not actuated), or “X” (don’t-care) [25].

According to a recent announcement by Illumina, a market leader company in DNA sequencing, an automated digital microfluidic platform has been transitioned to the marketplace for next-generation sample preparation [26]. A fully automated digital microfluidic platform consists of a CPU that controls the DMFB (Fig. 3). The CPU runs a biosystem software consisting of four modules: 1) CAD tool that takes an input sequencing graph, performs the scheduling, binding, module placement, routing and produces the actuation sequence; 2) An analytic tool that takes the sensor data, performs analysis and generates the final results of the assay; 3) A barcode reader through which the sample and the reagent details are fed to the system; 4) A database system that stores the details of every individual test, such as the source id and the test results.

Due to its recent introduction to the marketplace, the current commercial production of digital microfluidic systems follows a custom design flow. In this application-specific flow, all stages of the design flow are performed in-house, i.e., vertically integrated. However, due to the inherent reconfigurability in DMFBs, it is anticipated that the current use of these devices will shift from an application-specific setting to a general-purpose approach [27], [28]. Therefore, opportunities will be created for third party companies to be involved in the design flow. Discussion about general-purpose and custom DMFB design flows, and their potential vulnerabilities are introduced in the next section.

3 Threat Assessment of DMFBs

Advances in microfluidic technology offer tremendous benefits for enzymatic analysis, DNA analysis, immunoassays, toxicity monitoring, clinical diagnostics, point-of-care diagnosis of diseases. It has been also considered as a means to counter bio-terrorism [29].

On the other hand, since standard CMOS is an attractive technology option for DMFBs [30], they may be a target of attacks demonstrated on CMOS ASICs [31] and FPGAs [32]. However, unlike CMOS chips (ASIC or FPGA) used in security applications, a DMFB does not process secret information. Therefore, attacks related to stealing secret information from CMOS chips used in security applications are not relevant to DMFBs. Attacks such as stealing of hardware intellectual property (IP), chip and IP reverse engineering [33] and chip counterfeiting [34] are applicable to DMFBs. Attacks on DMFBs are similar to those on CMOS chips used in mission-critical applications, where the attacker takes control of the application to either manipulate results or disrupt the system [21]. In this paper, we assess the security of DMFBs against two types attacks that are unique to DMFBs: the manipulation attacks manipulate the results of a DMFB, while the denial-of-service attacks disrupt the functioning of the DMFB.

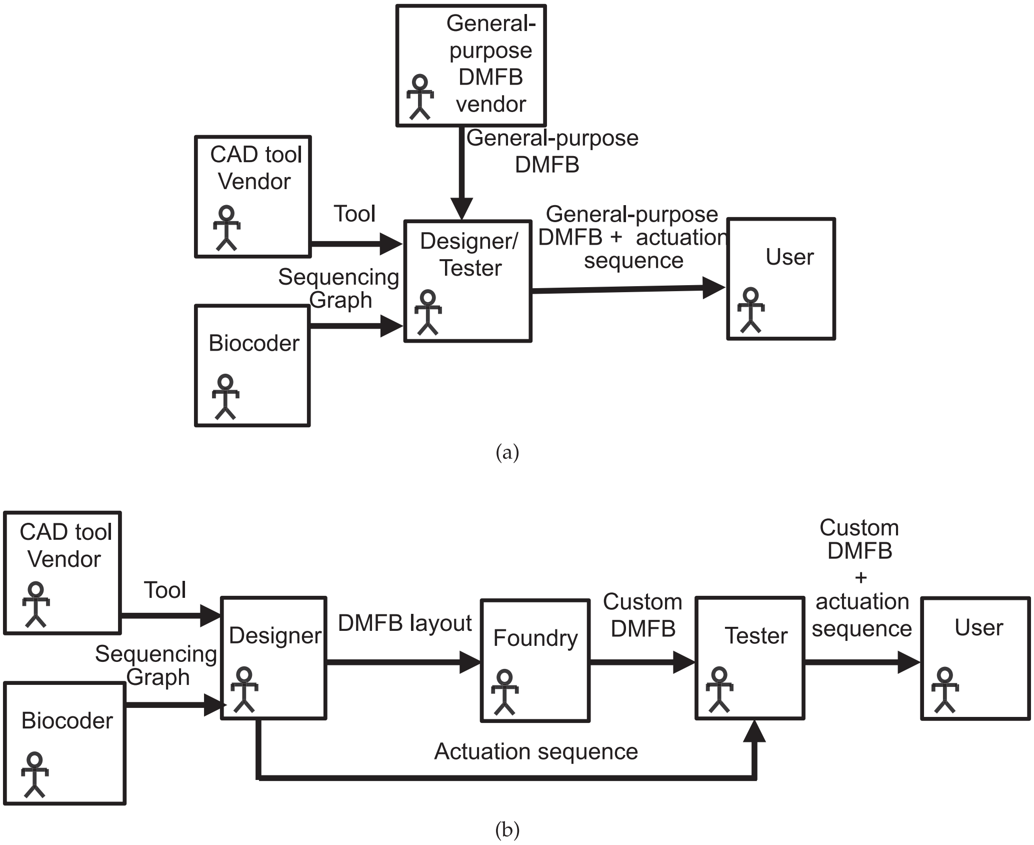
3.1 Motivations for Attacking DMFBs

There are multiple motivations for attacking a DMFB. An attacker may want to jeopardize a patient’s health by manipulating his/her clinical diagnosis results. An organization may hire an attacker to disrupt the products from a competitor. A terrorist organization that has spread deadly biological agents may bypass the detection methods by attacking the error-recovery methods in DMFBs. A pollution control authority may have introduced DMFBs in the field for real-time monitoring of toxins in the environment. The detection capability of these DMFBs may be compromised. An attacker may bypass the food quality control check ability of DMFBs used for this purpose.

Another motivation for studying the security aspects of DMFBs lies in the anticipated surge in interest in executing experimental protocols and lab routines on remote robotic systems. Considerable interest has been generated in recent years in remote-access laboratories that can implement robotics-based automated procedures for running biochemistry protocols [35], [36]. Since these protocols are downloaded on remote servers, cybersecurity remains a major concern. Over the next few years, it is anticipated that these robotic laboratories will be miniaturized to lab-on-chip DMFBs. Therefore, security challenges for such lab automation systems are also relevant to emerging DMFBs.

3.2 Who is the Attacker?

The attacker could be a user of the DMFB or anyone associated with the design and manufacturing flow of a DMFB.



(a)

(b)

Fig. 4. Participants in (a) a general-purpose DMFB design flow, and (b) custom DMFB design flow.

For a remotely accessible lab automation system, the attacker can be anybody with internet access who can compromise the service provider's cybersecurity system. Let us consider the following two DMFB design flows.

3.2.1 General-Purpose DMFB Design Flow

This is an FPGA-like design flow, where a general-purpose DMFB is procured, i.e., it can run any bioassay [28]. The sequencing graph of a bioassay is synthesized onto the DMFB, i.e., the corresponding actuation sequence is generated. Fig. 4a shows the different participants in this design flow. This design flow is also applicable when designing cyberphysical DMFB systems where the synthesis step is repeated and a new actuation sequence is generated based on feedback from the sensors on the DMFB. In this design flow, it is reasonable to assume that the biocoder—who converts an assay protocol into a sequencing graph and provides it to the designer—the DMFB designer, the tester, and the user are the same individual. Hence, one needs to consider two adversaries namely, the biocoder/designer and the CAD tool vendor.

3.2.2 Custom DMFB Design Flow

This is an ASIC-like design flow (Fig. 4b), where the biocoder who controls the DMFB platform (Fig. 3) sends the biochemical protocol to the design house as a sequencing graph. He gets the actuation sequence and the fabricated application-specific DMFB from the design house and programs the DMFB with it. The DMFB platform runs the assay on the DMFB according to the actuation sequence. In this design flow, the CAD tool vendor, the biocoder, the designer, the foundry, or the tester could be a potential adversary.

3.3 Attacks on General-Purpose DMFBs

Let us assess the security implications of a general-purpose DMFB when the attacker is one of the following two individuals.

3.3.1 Malicious Biocoder/Designer

A malicious biocoder/designer can launch the strongest attack as he can tamper with the assay or other system software. The attack steps that a malicious designer can follow are: 1) tamper with the assay operation by modifying the assay or by altering the CAD steps; 2) generate the actuation sequences for the golden assay and the malicious assay; 3) deploy the golden actuation sequence and opportunistically replace it with the malicious actuation sequence. Since the designer is also the user of the DMFB, he can manually substitute the golden actuation sequence with the malicious actuation sequence. He can also do the same by using a DMFB control software controlled trigger. Therefore, he can launch manipulation and denial-of-service attacks.

3.3.2 Malicious CAD Tool Vendor

A compromised CAD tool can add malicious operations into the assay. For example, it can add operations that corrupt the assay outcome. This attack is similar to the always-active hardware trojan attack [31]. The assay outcome will always be wrong and can be easily detected by a user.

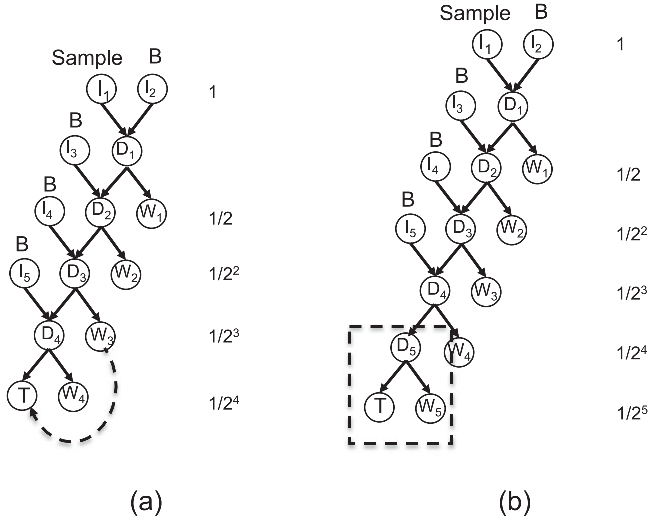


Fig. 5. Malicious modification of the sequencing graph for sample preparation: (a) Replacing the final droplet with the waste droplet. (b) Inserting additional mix-and-split cycle. B, W, and T are the buffer, the waste, and the target droplets, respectively. The nodes I and D represent the dispensing and the dilution operations, respectively.

3.4 Attacks on Custom DMFBs

In a custom DMFB design flow, there are numerous parties that could be malicious. Only a malicious biocoder can incorporate different types of triggers with the actuation sequence as he has access to the DMFB platform. Other parties, even if malicious, can not incorporate different types of triggers as they do not have access to the DMFB platform. In the absence of triggers, malicious operations will be always active during the assay execution and can be easily detected by a user. Therefore, only the malicious biocoder can launch stealthy attacks.

3.5 Result-Manipulation Attacks on DMFBs

In this section, we highlight result-manipulation attacks on DMFBs wherein a malicious biocoder/designer manipulates an assay outcome by maliciously altering different parameters of the assay, such as sample concentration, incubation time, and mixing time.

3.5.1 Compromising the Sample Preparation

In DMFBs, one of the major tasks is to prepare samples and reagents of desired concentrations, which are then mixed together to perform the assay operation. A typical DMFB uses LED-photodiode sensors. In an assay, the rate of reaction is proportional to the concentration of a specific element in the sample. The rate of reaction is equivalent to the rate of change of the optical absorbance [5]. Hence, if the concentration of the sample is altered, the optical absorbance measured by the photodiodes will be different.

A malicious biocoder/designer can tamper with the sequencing graph of the assay in order to manipulate the assay outcome. We show two such attacks that alter the sample concentration by tampering with the sequencing graph.

Consider the case, where the required concentration of the sample is $\frac{1}{2^m}$, where m is an integer. The required concentration is achieved during the sample preparation phase of the assay by performing a series of m mix-and-split (dilution) operations using the original sample and the buffer.[1] In each mix-and-split operation, the sample concentration is reduced by half. Let us assume that the test result will be considered to be positive only when the concentration of the sample is greater than or equal to CF_{high}. In another case, the test result will be negative when the sample concentration is less than CF_{low}.

Suppose, the malicious biocoder/designer wants to manipulate the result to change it to positive. In this case, he saves one of the discarded sample droplets, e.g., the waste droplet of the i-th mix-and-split operation, such that the concentration ($\frac{1}{2^i}$) of the discarded waste droplet is greater than CF_{high}. At the end of the sample preparation, the saved intermediate waste droplet is replaced by the target droplet. The malicious biocoder/designer knows about the assay operation corresponding to the sample preparation, and hence, he can simulate the sequencing graph of the assay and figure out the value of i. On the other hand, to produce a negative test outcome, the malicious biocoder/designer can perform additional mix-and-split operations on the target sample. Therefore, instead of the m-th mix-and-split operation, the target droplet is generated at the n-th mix-and-split operation ($n > m$), such that the final concentration $\frac{1}{2^n}$ is less than CF_{low}.

Fig. 5a shows the sequencing graph for sample preparation. The concentration of the sample droplet is diluted to $\frac{1}{2^4}$ by performing four mix-and-split operations. In each mix-and-split operation, the sample droplet is mixed with a buffer droplet and then split into two droplets of half the concentration. One of the two droplets (W) is discarded and the other one (I) is used for the next mix-and-split operation.

In this example, $CF_{high} = \frac{1}{2^4}$ and $CF_{low} = \frac{1}{2^5}$. The dotted line in Fig. 5a shows how the target droplet can be replaced by the waste droplet of the third mix-and-split operation, enforcing positive test result ($\frac{1}{2^3} > CF_{high}$). On the other hand, Fig. 5b introduces an additional mix-and-split operation (highlighted by the dotted rectangle) to further reduce the concentration to $\frac{1}{2^5}$. Therefore, one can manipulate the assay outcome by adding only a few edges or nodes in the original sequencing graph.

The sample concentration can also be tampered with during DMFB synthesis. A malicious designer can alter the concentration either during the architectural-level or during the physical-level synthesis. During the architectural-level synthesis, the malicious designer can modify the timing of scheduled operations. In the physical-level synthesis, malicious modifications can be done in two different ways: 1) by changing the placement of mixing/dilution operations, such that the operations are assigned to mixers with unsuitable sizes; and 2) by tampering with the droplet routing to extend a droplet route, increasing the rate of liquid evaporation in DMFBs [8]. Such tampering can manipulate the results when a general-purpose DMFB is used. Malicious modification of results is not possible in custom DMFB design flow, due to lack of trigger mechanism.

[1] A buffer solution, such as 1 M NaOH, is mixed with a sample droplet in order to dilute the constituent sample.

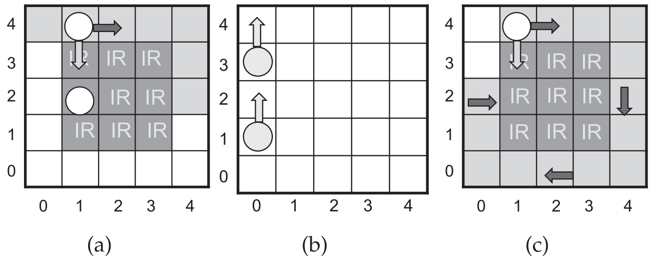


Fig. 6. Illustration of contamination: (a) A droplet enters the isolation region. (b) A droplet comes too close to another droplet. (c) The route of the droplet is deliberately prolonged. Instead of directly transporting the droplet to cell (2,1), it is transported by following a circuitous route.

3.5.2 Compromising the Incubation Time and Mixing Time

The incubation time and the mixing time of an assay have a major impact on the sample preparation. Incubation is widely used for cell culture, lysis, immunoassays, etc. For example, immunoassays are based on reactions that perform antigen-antibody bindings. One can target a specific antibody by designing a sample with the corresponding antigen attached to magnetic beads [4]. Therefore, the sample droplet is mixed with one droplet containing magnetic beads with primary capture antibodies and another droplet containing reporter antibodies to report the binding progress. This mix is then incubated for a specific duration depending on the target percentage of antigen capture [4]. If the incubation time is not sufficient, the required percentage of binding may not be reached [4], [37]. Mixing time also has a similar impact on sample preparation, i.e., an insufficient mixing time might lead to incorrect results. A malicious biocoder can tamper with the incubation/mixing time while providing the assay specifications to the designer. He can develop a result-manipulation attack by getting the golden and the malicious actuation sequences from the designer, and then by opportunistically replacing the golden actuation sequence with the malicious one. This attack is applicable to general-purpose and custom DMFB design flows.

Instead of directly changing the mixing time, a malicious biocoder can change the mixer geometry during resource binding. Mixing via a 4×2 two-dimensional array takes less time than that via a 4×1 linear array. The malicious biocoder can also manipulate the incubation/mixing time by altering the timing of scheduled operations (architectural-level synthesis) or by tampering with droplet routing (physical-level synthesis).

3.6 Denial-of-Service Attack

The best choice for a malicious biocoder to launch a DoS attack on a DMFB is by forcing/causing contamination. During transportation, a sample/reagent droplet leaves behind residues along its route. If another droplet follows the same route or intersects with the route, it may be contaminated by this residue. A droplet can also be contaminated if it unexpectedly mixes with another droplet or enters into an isolation region (IR). In order to isolate an assay operation, an IR is wrapped around the functional region of a microfluidic module (an IR is shown in Fig. 6) [38]. Three examples of contamination are as follows:

1) The route of a sample is changed in such a way that it either crosses the route of another sample or enters into an IR. This is done either by changing the routing algorithm or by maliciously modifying the actuation sequence (e.g., by deliberately flipping certain bits of the actuation sequence). In Fig. 6a, the droplet was originally programmed to move to cell (2, 4), but due to a malicious route change, it moves to cell (1, 3), which is inside an IR.
2) The routing algorithm can be maliciously modified to violate microfluidic transportation rules. For example, if the minimum spacing between adjacent droplets is below a threshold, these droplets will merge. Technology constraints in microfluidics mandate that the distance between two droplets be more than two cells. Fig. 6b shows an example, where the spacing between the two droplets is only one cell. Hence, the two droplets are likely to merge.
3) The length of the route that a droplet traverses can be increased or decreased by altering the routing algorithm. This way additional electrodes are contaminated with the residues of the droplet. So, when another droplet passes through this electrode later, it might be contaminated. Fig. 6c shows one such example, where instead of directly transporting the droplet to its destination cell (2,1), the droplet is transported around the perimeter of the operation region.

These contamination attacks assume that the malicious designer can opportunistically control the routing algorithm, which is only possible if the general-purpose DMFB design flow is used. In a custom DMFB design flow, a malicious biocoder can launch such an attack as follows. He sends the golden and the malicious assays to the design house. Both assays are synthesized onto the same DMFB. Once he receives the two actuation sequences and the custom DMFB, he can then opportunistically replace the golden actuation sequence with the malicious one.

Any malicious designer or untrusted foundry can tamper with on-chip sensors in several ways. The sensor results can be altered by tampering with the signal conditioners and the analog-to-digital converters. Even if the sensors report accurate results, the controller that receives this signal can be made to interpret them in a wrong way. Alternatively, the sensors themselves may be tampered with so that they report wrong results. However, physical tampering of sensors is an irreversible DoS attack.

3.7 Relation to Prior Work on Hardware Trojans

Hardware trojans can be inserted into ASICs and FPGAs. This is possible because the design flow is horizontal and distributed among different actors. Hardware trojans inserted into ASICs and FPGAs can be classified along five different dimensions: insertion phase, abstraction level, activation mechanism, effects and location [31], as shown in Fig. 7. We will explain the attacks on DMFBs described in this paper according to this taxonomy [31]. The hardware trojan taxonomy was proposed assuming a horizintally distributed design flow.

The result-manipulation attacks launched during the general-purpose DMFB design flow can be classified as hardware trojans inserted during the design phase and at the system level. Further, these trojans are externally triggered by

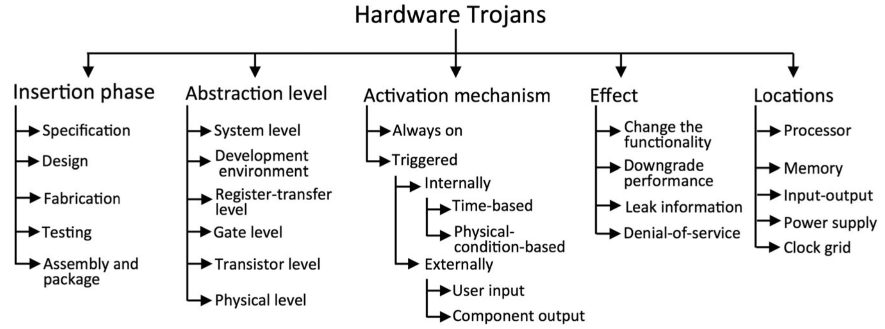


Fig. 7. A taxonomy of all possible hardware trojans along five different attributes has been proposed in [31]. The attacks on DMFBs describe as yet unexplored classes of attacks within this taxonomy. These are malicious specifications introduced at the system level and triggered manually.

a malicious biocoder/designer and the trigger mechanism is manual. Finally, these trojans change functionality and is inserted into the (DMFB) processor. The only difference for a custom DMFB design flow is the insertion phase. Unlike the general-purpose DMFB design flow, the trojan is inserted in the specification phase when the malicious biocoder provides the assay sequencing graph to the design house. The DoS attacks on DMFBs are identical to the result-manipulation attacks in four-of-the-five attributes. The only difference is that the effect is one of denial-of-service.

Our attacks on DMFBs explored trojans inserted during the specification phase and that are manually triggered. As the vertically-integrated DMFB design flows evolve to a horizontally-distributed design flow, additional trojan attacks demonstrated on ASICs and FPGAs [31] become relevant.

4 Case Study: Manipulation Attacks on Glucose Test Results

We use in-vitro measurement of glucose, which is a widely used clinical-diagnosis method for diabetes mellitus (hyperglycemia), as a case study. According to data from the Centers for Disease Prevention and Control (CDC), in 2011 alone, 22.9 million people in the US were diagnosed with diabetes [39]. A diabetic patient has to undergo regular glucose test for proper monitoring. Based on the blood glucose level, the amount of insulin to be injected into the patient is determined.

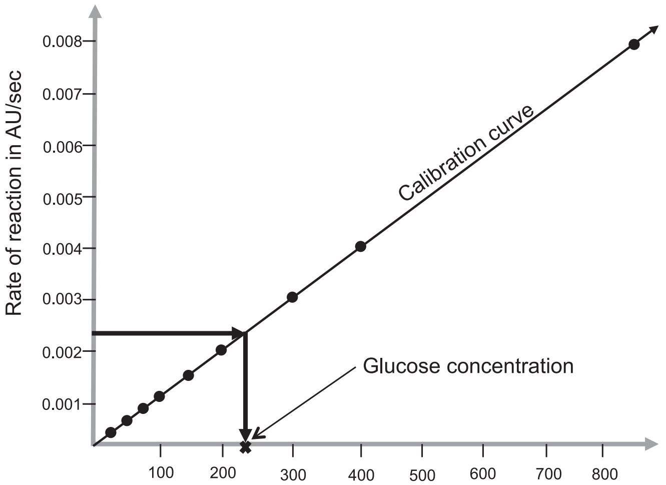


Fig. 8. Glucose calibration curve.

We demonstrate two attacks on the in-vitro measurement of glucose in serum and show that a malicious biocoder/designer can manipulate the assay outcome. We consider a general-purpose DMFB design flow. The malicious biocoder/designer can generate either positive or negative test results irrespective of the original glucose concentration in the serum. An erroneous positive test result (high blood glucose) could trigger a high dose of insulin injection into the patient's body, which may lead to a life threatening hypoglycemia. Similarly, an erroneous negative test result (low blood glucose) may further worsen the hyperglycemia (the patient remains untreated). In both cases, the patient's life is endangered. Moreover, the clinical laboratory under the attack may be subjected to litigation and lawsuits due to inaccurate test results.

4.1 In-Vitro Glucose Test

In-vitro glucose test is used to determine the concentration of glucose in human physiological fluids, such as serum. An obvious application of this economical test is the determination of the blood sugar level.[2] The bench-top sequence for this test, known as *glucose assay*, is realized on a DMFB as a colorimetric assay, in which the color change is detected using an absorbance measurement system consisting of a light emitting diode and a photodiode [40].

This assay measures the glucose concentration level in a blood sample by constructing the glucose *calibration curve* (Fig. 8) via serial dilutions of the standard glucose solution. The X-axis represents the different concentrations formed by these dilutions (in mg/dL) and the Y-axis represents the rate of reaction quantified by the change in absorbance degree reported as AU/sec (absorbance unit per second). This curve helps interpolate the concentration of the glucose sample under test. As shown in Fig. 8, the reaction rate of the sample is a point on the Y-axis and the corresponding point on the X-axis is the sample concentration.

4.2 DMFB Attack Trigger Mechanism

Suppose the malicious biocoder/designer wants to manipulate test samples of a specific patient. In this case, the malicious biocoder/designer can trigger it using the source id. He can mark the targeted source ids in the database. The trigger program is activated when the barcode scan matches

[2] Normal/safe levels of blood glucose (measured in mg/dL) depend on several factors, such as age, food activity, interference with other diseases, etc.

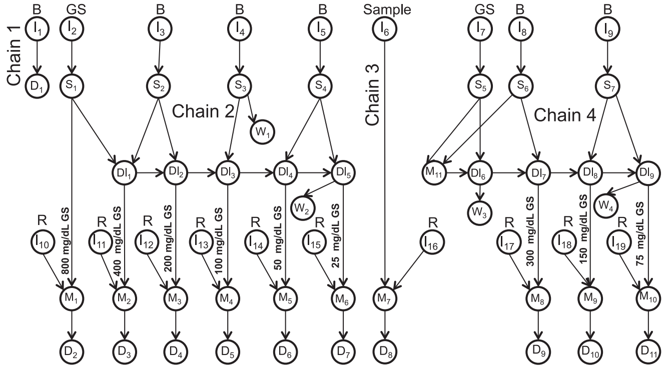


Fig. 9. Golden execution: B is the $1.4\ \mu\text{L}$ buffer droplet, Sample is the $0.7\ \mu\text{L}$ glucose sample droplet, R is the $0.7\ \mu\text{L}$ reagent droplet, GS is the $1.4\ \mu\text{L}$ 800 mg/dL glucose solution droplet, and W_i is the waste droplet. D_i, DI_i, S_i, M_i, and I_i are the detection, dilution, splitting, mixing, and dispensing operations, respectively.

the target source id. Once activated, the malicious actuation sequence is executed instead of the original one.

4.3 Attack Model

The malicious biocoder/designer can manipulate the assay by tampering with the calibration curve shown in Fig. 8. The calibration curve can be tampered with when the DMFB is calibrated using standard glucose solutions. The malicious biocoder/designer alters the concentrations for one or more of the standard glucose dilutions and produces a wrong calibration curve. He can also manipulate the assay by changing the concentration of the sample itself. Note that the LED-photodiode sensor in this assay is not designed to identify malicious modifications. Its sole purpose is to report the rate change in absorbance in the enzyme-kinetic reaction [40].

As described in Section 4.1, known concentrations of glucose solution are used to plot the calibration curve. On this curve, the rate of reaction measured by a fluorescence detector is plotted against the glucose concentration. After drawing the curve, the rate of reaction for a glucose sample is used to determine the sample concentration. To demonstrate the attacks, we consider the following three scenarios:

1) *Golden execution*: No attack is carried out.
2) *Attack 1*: The concentration of the glucose sample is modified via a malicious dilution operation.
3) *Attack 2*: The calibration curve is manipulated by tampering with the concentrations of the glucose solution during calibration.

4.3.1 Golden Execution

The sequencing graph shown in Fig. 9 describes the golden execution for the glucose assay. The sequencing graph consists of four independent reaction chains 1, 2, 3, and 4, measuring the rate of reaction for a blank/buffer droplet (chain 1), glucose solution concentrations 800, 400, 200, 100, 50, 25 mg/dL (chain 2), glucose solution concentrations 300, 150, 75 mg/dL (chain 4), and the glucose sample (chain 3). The calibration curve is generated using the reaction chains 1, 2, and 4, while the reaction chain 3 is used to determine the glucose concentration of the sample.

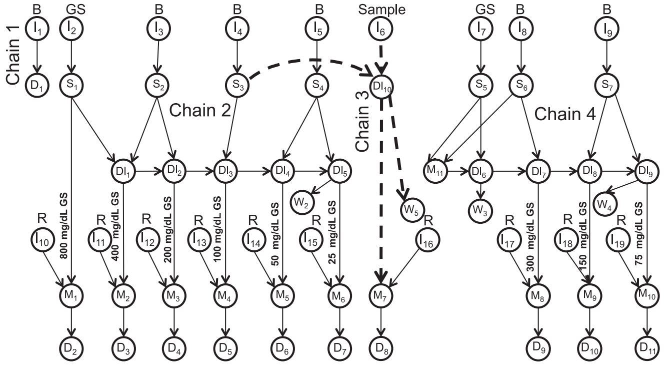


Fig. 10. In Attack 1, the waste buffer droplet generated by the splitting operation S_3 is used to dilute the sample droplet in DI_{10}. The thick dotted lines show the changes with respect to the golden sequencing graph.

4.3.2 Attack 1

The malicious biocoder/designer tampers with the assay result by changing the concentration of the glucose sample as shown in Fig. 10. The thick dotted lines show the changes in the sequencing graph compared to the golden sequencing graph. The waste buffer droplet W_1 generated from S_3 is mixed with the glucose sample droplet of I_6 and then diluted in DI_{10}. Since the concentration of the glucose sample is halved, the result of the assay execution will be wrong. The user is unaware that a waste buffer droplet is used for tampering with the sample concentration. Using the golden calibration curve shown in Fig 11, the user will interpret the result as follows. In the golden calibration curve, the dots are the standard sample points corresponding to glucose solution concentrations $(75, 150, \ldots, 800$ mg/dL). The user will interpret the sample concentration as 110 mg/dL instead of the original concentration of 220 mg/dL. Hence, the patient may not be treated with the medication for high blood sugar, which could be life threatening.

If the patient or the medical practitioner wants to verify the result then there are two possible options: either to repeat the same test on the same DMFB, or to test it on a different DMFB by a different lab. It is highly likely that the test, when repeated by a different lab, will yield the correct result.

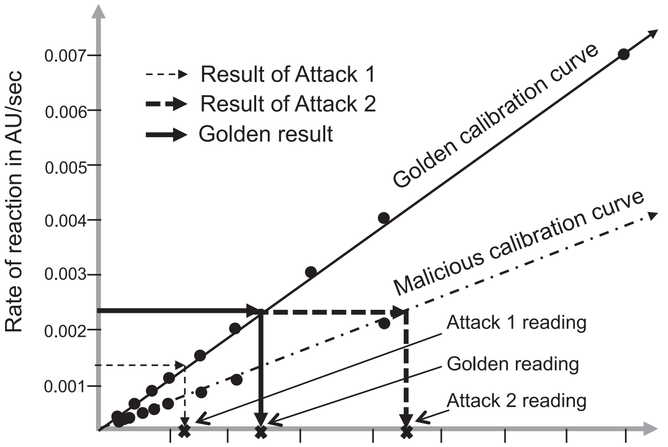


Fig. 11. Malicious glucose concentrations: The thin dotted lines represent the modified glucose concentration measured on the golden curve. The thick dotted lines represent the glucose concentration measured on the malicious curve.

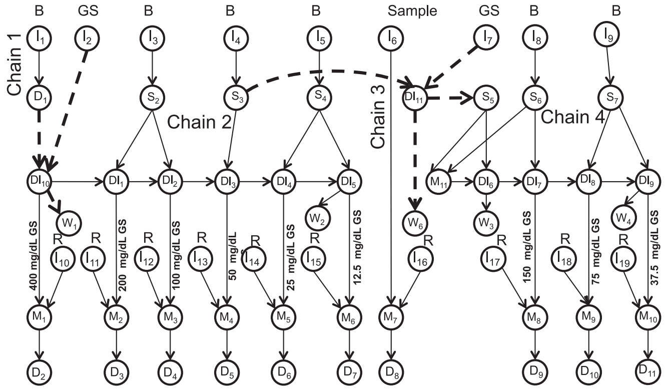


Fig. 12. In Attack 2, the discarded buffer droplets of D_1 and S_3 are mixed with the droplets of I_2 and I_7, respectively, diluting the reaction chains 2 and 4, respectively.

4.3.3 Attack 2

The malicious biocoder/designer tampers with the golden calibration curve to have the resulting reported concentration of the glucose sample different from (either higher or lower than) the golden value. We will show how the reported concentration of the glucose sample can be made higher than the golden value. The attack is performed by tampering with the sequencing graphs for reaction chains 2 and 4 to generate a malicious calibration curve. The two waste buffer droplets generated from D_1 and S_3 in the golden sequencing graph are used for this purpose. The malicious sequencing graph for such an attack is shown in Fig. 12.

The thick dotted lines show the changes with respect to the golden sequencing graph. The waste buffer droplet (after D_1) in the reaction chain 1 is merged with the glucose solution (the droplet generated from I_2) in the reaction chain 2, thus diluting the entire reaction chain 2. The glucose solution concentrations in the reaction chain 2 are reduced to (400, 200, 100, 50, 25, 12.5 mg/dL) half of their golden values. Similar effect can also be seen in the reaction chain 4, where the waste buffer droplet generated from S_3 is mixed with the glucose solution droplet generated from I_7. The dotted curve in Fig. 11 shows the malicious calibration curve generated by Attack 2.

The DMFB user is unaware that the calibration curve is malicious. The user will interpret the result using the malicious calibration curve (the dotted curve in Fig. 11). The result will show a higher concentration of glucose compared to the golden result. As the figure shows, the original concentration is 220 mg/dL when the golden calibration curve is used. Following Attack 2, the measured concentration is 440 mg/dL since the malicious calibration curve is used. Hence, the patient will be falsely alarmed and may receive a high dose of insulin, if this is the only test that he relies on.

A practical step-by-step scenario for the attack described above is as follows.

- *Step 1*: The patient visits the pathology department and his blood sample is collected, labelled with the patient's barcode, and forwarded to the diagnosis lab.
- *Step 2*: In the lab, the pathologist scans the sample using the barcode reader (connected to the target system) and then selects the in-vitro diagnostic test (actuation sequence) in the system.
- *Step 3*: The trigger program is activated upon the scan of the barcode and alerted that the sample belongs to the target patient. The trigger program then selects the Attack 2 actuation sequence instead of the original actuation sequence.
- *Step 4*: When the malicious actuation sequence is executed, a high glucose concentration in the sample will be detected by the assay operation.
- *Step 5*: The patient will be falsely treated with high dose of insulin.

4.4 Experimental Results

The golden, Attack 1, and Attack 2 sequencing graphs are executed using an open-source DMFB tool [28] on a 17 × 31 electrode-array DMFB with 7 input reservoirs.[3] A 100 Hz clock was considered for actuating the electrodes. The DMFB design times for the golden, Attack 1, and Attack 2 assays are 35, 39, and 62 milliseconds, respectively, while the assay execution times are 8.5, 9.26, and 10.46 seconds, respectively. Attack 1 is difficult to detect, since the difference in the DMFB synthesis time (35 ms versus 39 ms) and assay execution time (8.5 s versus 9.26 s) are negligible. This is because the attack alters the glucose sample using one additional dilution operation. Attack 2 impacts a large portion of the glucose assay, since it alters the concentrations of the glucose solution. The difference between the golden assay execution time and Attack 2 assay execution time is 1.9 seconds (8.5 s versus 10.46 s). The difference in the DMFB synthesis time is only 23 ms (39 ms versus 62 ms). It is unlikely that the user can notice such a small change in the DMFB assay execution times.

5 Case Study: Attack in Presence of Error-Recovery Mechanism

In cyberphysical DMFBs, error-recovery is added to the DMFB control software [15], [41]. There are two basic approaches to detect runtime operational errors related to droplet mobility, size, or sample concentration, and to facilitate the appropriate rollback to ensure reliable execution. The first approach is based on integrated optical detectors and the second approach is based on the use of a charge-coupled device (CCD) camera.

5.1 Optical Detector Based Error-Recovery

This technique is based on introducing intermediate checkpoints into the assay sequencing graph. Fig. 13a shows two checkpoints inserted into the original assay sequencing graph in Fig. 5a. At a checkpoint C_i, the droplet is transported to an on-chip optical detector. After each detection, the control software compares the intermediate result with the pre-determined value. If the comparison fails, then the corresponding operation in the assay is re-executed. For example, if checkpoint C_1 reports a failure, the dilution operation D_1 is repeated.

5.2 CCD Camera based Error-Recovery

This technique uses a CCD camera to monitor if the droplets are at the correct locations on the DMFB at selected

[3] We used list scheduler, left-edge placer, and maze router for the DMFB design.

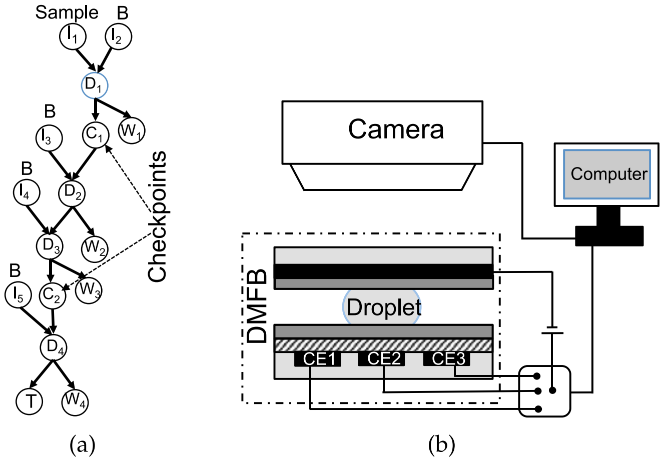


Fig. 13. Runtime operational error-recovery techniques: (a) Two checkpoints C_1 and C_2 are inserted into the sequencing graph to detect the intermediate products. (b) A CCD camera monitors droplets along their transportation route.

instances [15]. Fig. 13b shows the components of a CCD-based microfluidic platform. Error-recovery software in the platform monitors the intermediate fluidic operations of the DMFB. The CCD camera captures the images of fluidic operations and sends it to the error-recovery software. The software locates droplets in the DMFB from these captured images. In order to locate a droplet in the image, a template image of the droplet is moved to a possible position in the captured image and the corresponding part of the image is cropped. The cropped image is then correlated with the template image. The correlation index is then compared with a predetermined threshold to determine if the droplet is at specific sites at specific time instances as expected. For example, suppose as per the assay operation, at time instance t, a drop d is expected at the microfluidic electrode $c_{i,j}$, where i and j represent the index of the electrode in the two-dimensional array of microfluidic electrodes. At run time, to validate whether d was present at $c_{i,j}$ at t, an image of the array is captured and the corresponding template image of the droplet is searched in the image at $c_{i,j}$. If a match is found, then the droplet operation at time t is considered to be error-free; otherwise, the droplet route is considered faulty.

5.3 Attacks on a Cyberphysical DMFB System

From a biocoder's perspective, the outlined error-recovery techniques might expose the malicious modifications in the assay. In order to make the attack stealthy, the biocoder not only needs to manipulate the sample as shown in Attack 1 and 2 in Section 4.3, but also needs to bypass the error-recovery techniques. However, from a designer's perspective, the outlined error-recovery techniques are relatively easy to bypass.

5.3.1 Attacks on a General-Purpose Cyberphysical DMFB

The malicious designer needs to bypass the checkpoints and he can do so by removing them from the malicious sequencing graph. The checkpoints are meant for the control software and not for the users. When a fault-free DMFB is used, the removal of checkpoints will not impact the assay outcome. However, if a faulty DMFB is used and the user knows the expected assay outcome *a priori*, only then he will infer that the DMFB is faulty and the deployed error-recovery technique will fail to detect the fault. It should be noted that the attack does not permanently bypass the error-recovery mechanism. Rather, it is bypassed only for the specific samples from the targeted individual as mentioned in the five steps of the last paragraph of Section 4.3.3. We highlight that this is a novel attack within the taxonomy shown in Fig. 7 and it has not been reported or explored in the context of ICs. Therefore, the user will not have prior knowledge of the attack.

Alternately, the malicious designer can tamper with the thresholds for intermediate results so that the error-recovery software does not report an error. This latter approach can bypass both error-recovery techniques.

5.3.2 Attacks on a Custom Cyberphysical DMFB

In this design flow, the malicious biocoder provides a malicious sequencing graph to the design house. The design house incorporates checkpoints into the malicious sequencing graph by following their standard design process for the cyberphysical DMFB with error recovery [15], [41]. Therefore, the malicious biocoder will receive an actuation sequence with support for error recovery for the malicious assay. Now, although the malicious biocoder can replace the golden actuation sequence with the malicious actuation sequence. the checkpoints remain in the malicious assay and the error-recovery component remains in the control software. Therefore, while executing the malicious assay, the error-recovery software will compare the intermediate results generated at the checkpoints with the golden values. Since the intermediate results for the malicious assay will be different from those for the golden assay, an error is generated, triggering re-synthesis of the golden assay. This in turn overwrites the malicious actuation sequence with the golden actuation sequence. Thus the attack will fail.

5.4 Attacks on the Control Software

Error-recovery techniques in a cyberphysical DMFB pose new challenges to an attacker (malicious biocoder/designer), specially in a custom DMFB design flow. The only option for the malicious biocoder is to tamper with the error-recovery software. In this case the attack will be the same for both design flows. A malicious biocoder can tamper with the error-recovery portion of the control software to bypass the error-recovery mechanism. This requires reverse engineering the control software binary [42], [43].

6 Case Study: Denial-of-Service Attacks on In-Vitro Glucose Test

In this section, we demonstrate DoS attack case studies on a DMFB executing multiplexed in-vitro diagnostics of human physiological fluids [12]. For the assay, we use two types of human physiological fluids: serum and plasma. Three measurement operations, namely, glucose, lactate and pyruvate, are performed on each fluid. Fig. 14 shows the sequencing graph of the assay, where three droplets of plasma I_1, I_2, and I_3 are mixed with the three reagent droplets R_1, R_2, and R_3, respectively. Similarly, three droplets of serum are mixed

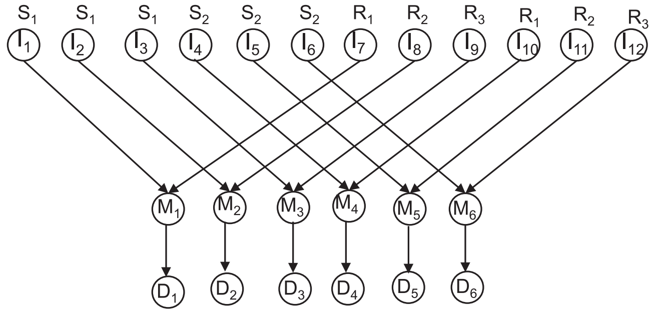


Fig. 14. In-vitro measurement of glucose, lactate, and pyruvate in human physiological fluids serum (S_1) and plasma (S_2) samples. R_1, R_2, and R_3 are the glucose, lactate, and pyruvate reagents, respectively.

with the corresponding droplets of the reagents. After each mixing operation, the corresponding glucose, lactate, or pyruvate measurement is done using an on-chip optical detector.

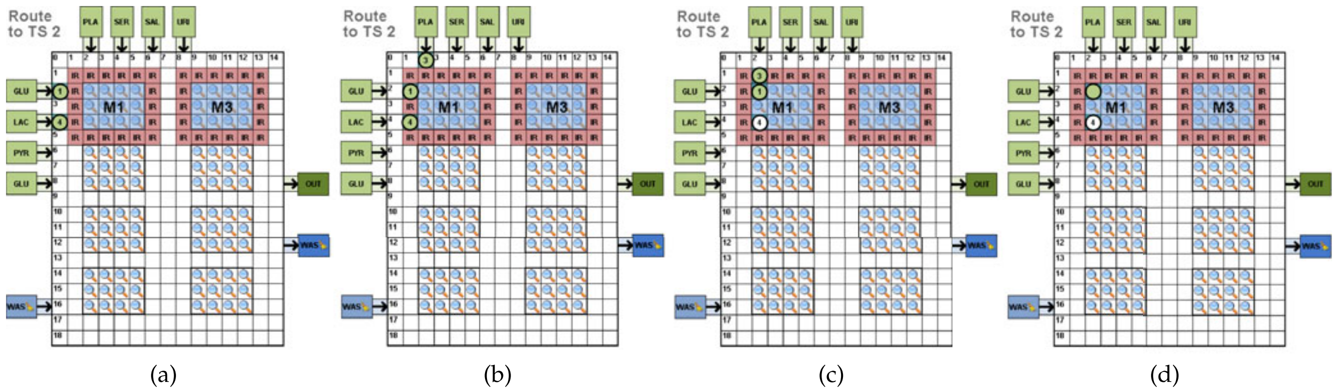
Suppose, a malicious designer has access to the CAD tool binary (in a general-purpose DMFB design flow). He can reverse engineer the binary as described in Section 5.4. The aim of the malicious designer is to corrupt the assay. This can be done by corrupting only a few droplets of the assay. We show two DoS attacks; in one the malicious designer violates the minimum spacing between two droplets and in the other, the droplet route is modified to intentionally merge with another droplet in a mixer. We simulated these attack scenarios using an open-source DMFB synthesis tool [28]. We used maze droplet router to demonstrate the DoS attacks.

6.1 Violating the Minimum Spacing Between Droplets

This can be done by targeting a specific droplet pair and then violating the minimum spacing condition for this target pair of droplets. In the attack, a plasma droplet is forced to merge with a glucose droplet. Fig. 15a shows two reagent droplets, a glucose droplet 1 and a lactate droplet 4 being dispensed. Droplets 1 and 4 are routed to mixers M_3 and M_1, respectively. In the next cycle (Fig. 15b), a plasma droplet 3 is dispensed and routed towards droplet 4 at mixer M_1. However, in the third clock cycle (Fig. 15c), droplet 3 comes next to droplet 1 and they merge in the next cycle.

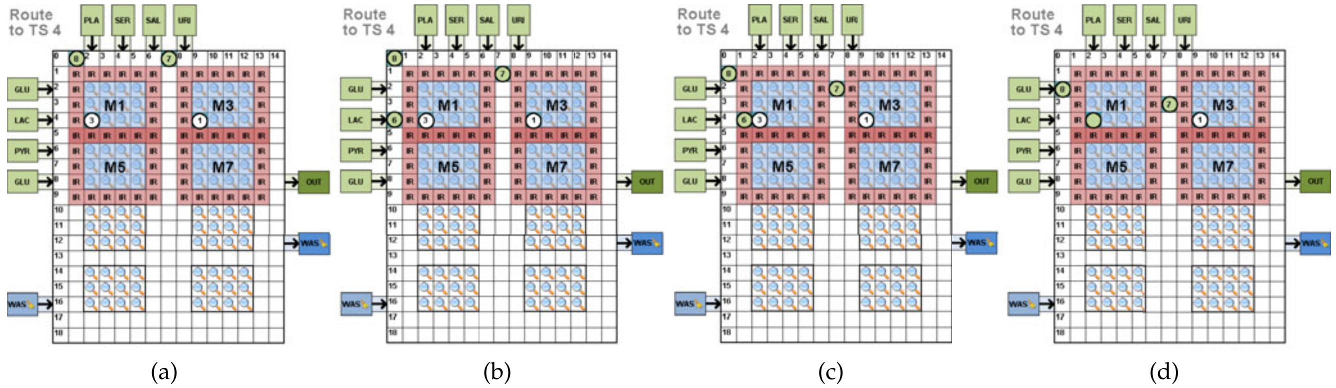
6.2 Malicious Modification of the Droplet Route

In this attack scenario, the route of a droplet is intentionally modified in such a way that the droplet moves to the isolation region and merges with a droplet in the mixer. Fig. 16a shows that droplet 3 and droplet 1 are waiting to merge with droplet 8 and droplet 7, respectively. However, in the next cycle (Fig. 16b), droplet 6 is dispensed and move towards the mixer M_5, while droplet 7 is in its normal route towards mixer M_3. The route of droplet 6 is intentionally modified to move it towards droplet 3 (Fig. 16c). In the fourth cycle, it merges with droplet 3 (Fig. 16d). This attack can be launched by a malicious biocoder by tampering with the actuation sequence of the assay. In this case, the malicious biocoder only needs to flip certain bits of the golden actuation sequence to create a malicious actuation sequence.



(a) (b) (c) (d)

Fig. 15. Droplet routing: (a) Droplets 1 and 4 are dispensed. (b) Droplet 3 is dispensed while droplet 4 is waiting in M_1 to be merged with droplet 3. Droplet 1 is on its way to mixer M_3. (c) Droplet 3 is next to droplet 1. (d) Droplets 1 and 3 merge.



(a) (b) (c) (d)

Fig. 16. Droplet routing: (a) Droplets 1 and 3 are waiting to be merged with droplet 7 and droplet 8, respectively. (b) Droplet 6 is dispensed to move towards M_5. (c) Droplet 6 comes next to droplet 3. (d) Droplets 6 and 3 merge.

TABLE 1
Access Control of DMFBs

	Assay synthesis	Assay execution	Database Update
Administrator	Yes	Yes	Yes
Designer	Yes	Yes	No
Pathologist	No	Yes	No

7 Conclusions

We have reported the first ever assessment of the security of current and emerging DMFBs. We have described multiple attacks that can have a catastrophic effect on the integrity of the DMFB assay outcomes. For example, we have demonstrated attacks on a state-of-the-art DMFB-based, in-vitro glucose measurement system that manipulates the assay outcomes. We have demonstrated attacks that change the DMFB design steps or maliciously alter the actuation sequence. These attacks are practically feasible and stealthy. The attacks require small and easy to implement changes to the sequencing graph and or the actuation sequence.

One can easily obviate these attacks by controlling access to the DMFB system [44]. For example, consider three users of the DMFB system: 1) the administrator, who maintains the security and the integrity of the system, 2) the designer, and 3) the user of the DMFB such as a pathologist. If we assume that the administrator is trusted, then Table 1 summarizes the access control policies that need to be enforced for the system. Only the administrator has the right to modify or update the system (e.g., the system software). The designer can synthesize a design and execute assays, whereas a pathologist can only execute assays. Although similar access control policies can be developed, they are difficult to enforce and are susceptible to insider attacks.

The proposed security assessment has direct applications to not only the automation of point-of-care (e.g., Lab-on-a-Chip), but also environmental monitoring (e.g., the deployment of Internet-of-Things for environmental monitoring), and 3D bio-print.

An important next step in our research is to develop hardware- and cyberphysical-enabled defenses against attacks on basic and cyberphysical DMFBs.

Acknowledgments

This work was supported in part by the Center for Interdisciplinary Studies in Security and Privacy (CRISSP/CRISSP-AD). A preliminary version of this work was published in the *Proceeding of the IEEE International Conference on Computer Design* (*ICCD*), October 2015.

References

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

[2] T. Thorsen, S. J. Maerkl, and S. R. Quake, “Microfluidic large-scale integration,” *Science*, vol. 298, no. 5593, pp. 580–584, 2002.

[3] E. Verpoorte and N. F. De Rooij, “Microfluidics meets MEMS,” *Proc. IEEE*, vol. 91, no. 6, pp. 930–953, 2003.

[4] R. Sista, Z. Hua, P. Thwar, A. Sudarsan, V. Srinivasan, A. Eckhardt, M. Pollack, and V. Pamula, “Development of a digital microfluidic platform for point of care testing,” *Lab Chip*, vol. 8, no. 12, pp. 2091–2104, 2008.

[5] V. Srinivasan, V. K. Pamula, and R. B. Fair, “An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids,” *Lab Chip*, vol. 4, no. 4, pp. 310–315, 2004.

[6] D. J. Boles, J. L. Benton, G. J. Siew, M. H. Levy, P. K. Thwar, M. A. Sandahl, J. L. Rouse, L. C. Perkins, A. P. Sudarsan, R. Jalili et al., “Droplet-based pyrosequencing using digital microfluidics,” *Analytical Chemistry*, vol. 83, no. 22, pp. 8439–8447, 2011.

[7] Y. Zhao, S. K. Chung, U.-C. Yi, and S. K. Cho, “Droplet manipulation and microparticle sampling on perforated microfilter membranes,” *J. Micromechanics Microeng.*, vol. 18, no. 2, p. 025030, 2008.

[8] R. B. Fair, “Digital microfluidics: Is a true lab-on-a-chip possible?” *Microfluidics Nanofluidics*, vol. 3, no. 3, pp. 245–281, 2007.

[9] K. Chakrabarty, “Design automation and test solutions for digital microfluidic biochips,” *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 57, no. 1, pp. 4–17, Jan. 2010.

[10] F. Su and K. Chakrabarty, “Architectural-level synthesis of digital microfluidics-based biochips,” in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2004, pp. 223–228.

[11] F. Su and K. Chakrabarty, “Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips,” in *Proc. IEEE/ACM Des. Autom. Conf.*, 2005, pp. 825–830.

[12] F. Su, W. Hwang, and K. Chakrabarty, “Droplet routing in the synthesis of digital microfluidic biochips,” in *Proc. Des. Autom. Test Eur.*, 2006, pp. 1–6.

[13] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, “BioRoute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips,” *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 27, no. 11, pp. 1928–1941, Nov. 2008.

[14] T.-W. Huang, J.-W. Chang, and T.-Y. Ho, “Integrated fluidic-chip co-design methodology for digital microfluidic biochips,” in *Proc. ACM Int. Symp. Physical Des.*, 2012, pp. 49–56.

[15] Y. Luo, K. Chakrabarty, and T.-Y. Ho, “Error recovery in cyberphysical digital microfluidic biochips,” *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 32, no. 1, pp. 59–72, Jan. 2013.

[16] N. M. Jokerst, L. Lin, S. Palit, M. Royal, S. Dhar, M. Brooke, and T. Tyler, “Progress in chip-scale photonic sensing,” *IEEE Trans. Biomedical Circuits Syst.*, vol. 3, no. 4, pp. 202–211, Aug. 2009.

[17] Y.-J. Shin and J.-B. Lee, “Machine vision for digital microfluidics,” *Rev. Sci. Instrum.*, vol. 81, no. 1, p. 014302, 2010.

[18] Y. Luo, K. Chakrabarty, and T.-Y. Ho, “Real-time error recovery in cyberphysical digital microfluidic biochips using a compact dictionary,” *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 32, no. 12, pp. 1839–1852, Dec. 2013.

[19] W. Houser, “Could what happened to sony happen to us?” *IT Prof.*, vol. 17, no. 2, pp. 54–57, 2015.

[20] C. Li, A. Raghunathan, and N. K. Jha, “Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system,” in *Proc. IEEE Int. Conf. e-Health Netw. Appl. Serv.*, 2011, pp. 150–156.

[21] R. Langner. (2013). To kill a centrifuge: A technical analysis of what Stuxnet’s creators tried to achieve [Online]. Available: http://www.langner.com/en/wpcontent/uploads/2013/11/To-kill-a-centrifuge.pdf

[22] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, “Security implications of cyberphysical digital microfluidic biochips,” in *Proc. IEEE Int. Conf. Comput. Des.*, 2015, pp. 483–486.

[23] F. Mugele and J.-C. Baret, “Electrowetting: From basics to applications,” *J. Phys.: Condensed Matter*, vol. 17, no. 28, p. R705, 2005.

[24] M. Pollack, A. Shenderov, and R. Fair, “Electrowetting-based actuation of droplets for integrated microfluidics,” *Lab Chip*, vol. 2, no. 2, pp. 96–101, 2002.

[25] T. Xu and K. Chakrabarty, “Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips,” in *Proc. IEEE/ACM Des. Autom. Conf.*, 2008, pp. 173–178.

[26] (2016). Illumina. Illumina neoprep library prep system [Online]. Available: http://www.illumina.com/systems/neoprep-library-system.html/

[27] Y. Luo and K. Chakrabarty, “Design of pin-constrained general-purpose digital microfluidic biochips,” *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 32, no. 9, pp. 1307–1320, Sep. 2013.

[28] D. Grissom and P. Brisk, “A field-programmable pin-constrained digital microfluidic biochip,” in *Proc. IEEE/ACM Des. Autom. Conf.*, 2013, pp. 1–9.

[29] F. Su and K. Chakrabarty, “High-level synthesis of digital microfluidic biochips,” *ACM J. Emerging Technol. Comput. Syst.*, vol. 3, no. 4, pp. 16–32, 2008.

[30] Y.-Y. Lin, R. D. Evans, E. Welch, B.-N. Hsu, A. C. Madison, and R. B. Fair, "Low voltage electrowetting-on-dielectric platform using multi-layer insulators," *Sens. Actuators B: Chemical*, vol. 150, no. 1, pp. 465–470, 2010.

[31] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware Trojans," *IEEE Comput.*, vol. 43, no. 10, pp. 39–46, 2010.

[32] S. M. Trimberger and J. J. Moore, "FPGA security: Motivations, features, and applications," *Proc. IEEE*, vol. 102, no. 8, pp. 1248–1265, 2014.

[33] J. Rajendran, A. Ali, O. Sinanoglu, and R. Karri, "Belling the CAD: Toward security-centric electronic system design," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 34, no. 11, pp. 1756–1769, Nov. 2015.

[34] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges Ahead," *J. Electron. Test.*, vol. 30, no. 1, pp. 9–23, 2014.

[35] (2016). Transcriptic. Automated cell and molecular biology laboratory [Online]. Available: https://www.transcriptic.com

[36] (2016). Emeraldcloudlab. The emerald cloud laboratory [Online]. Available: http://www.emeraldcloudlab.com

[37] J. W. Karaszkiewicz. (2010). Critical factors in immunoassay optimization [Online]. Available: https://www.kpl.com/docs/techdocs/BENCH2.PDF

[38] Y. Zhao and K. Chakrabarty, "Cross-contamination avoidance for droplet routing in digital microfluidic biochips," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 31, no. 6, pp. 817–830, Jun. 2012.

[39] (2011). Centers for disease control and prevention: Diabetes public health resource [Online] http://www.cdc.gov/diabetes/statistics/prev/national/ figpersons.htm

[40] V. Srinivasan, V. K. Pamula, and R. B. Fair, "Droplet-based microfluidic lab-on-a-chip for glucose detection," *Analytica Chimica Acta*, vol. 507, no. 1, pp. 145–150, 2004.

[41] 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. Emerg. Technol. Comput. Syst.*, vol. 6, no. 3, p. 11, 2010.

[42] (2003). Sality: An entry-point obscuring polymorphic file infector [Online]. Available: http://www.symantec.com/security_response/writeup.jsp/?docid=2006-011714-3948-99

[43] D. Regalado, S. Harris, A. Harper, C. Eagle, J. Ness, B. Spasojevic, R. Linn, and S. Sims, *Gray Hat Hacking the Ethical Hacker's Handbook*, 4th ed. New York, NY, USA: McGraw-Hill, 2015.

[44] R. Sandhu and P. Samarati, "Access control: Principle and practice," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 40–48, 1994.

Sk Subidh Ali received the BE degree in computer engineering from the Burdwan University and the ME degree from the West Bengal University of Technology, India. He received the PhD degree in computer science and engineering from the Indian Institute of Technology Kharagpur, in 2012. He is currently a postdoc research associate in the New York University, Abu Dhabi. His research interest is in hardware security: side-channel analysis, fault analysis of cryptochip, secure design for testability, and security aspects of digital microfluidic biochips. He has contributed to hardware security with 30 publications.

Mohamed Ibrahim received the BSc (Hons.) degree in electrical engineering from the Ain Shams University, Cairo, Egypt, in 2010, and the MSc degree from the same university, in 2013. He is currently working toward the PhD degree with the Department of Electrical and Computer Engineering at the Duke University, Durham, NC. He was appointed as a research and teaching assistant by the Faculty of Engineering, Ain Shams University, since his graduation. In 2014, he joined Prof. Chakrabarty's lab to work on the design automation and test of next-generation cyberphysical digital-microfluidic biochips. His research interests also include mixed-signal very-large scale integration design, microelectromechanical-system modeling, and simulation.

Ozgur Sinanoglu received the BS degrees, one in electrical and electronics engineering and the other in computer engineering, both from the Bogazici University, Turkey, in 1999. He received the MS and PhD degrees in computer science and engineering from the University of California San Diego, in 2001 and 2004, respectively. He is an associate professor of electrical and computer engineering at the New York University, Abu Dhabi. He has industry experience at the TI, IBM, and Qualcomm, and has been with the NYU Abu Dhabi since 2010. During his PhD, he won the IBM PhD fellowship award twice. He also received the best paper awards at the IEEE VLSI Test Symposium 2011 and ACM Conference on Computer and Communication Security 2013. His research interests include design-for-test, design-for-security, and design-for-trust for VLSI circuits, where he has more than 140 conference and journal papers, and 15 issued and pending US Patents. He has given more than a dozen tutorials on hardware security and trust in leading CAD and test conferences, such as DAC, DATE, ITC, VTS, ETS, ICCD, ISQED, etc. He is serving as track/topic chair or technical program committee member in about 15 conferences, and as (guest) associate editor for *IEEE TCAD, ACM JETC, Elsevier MEJ, JETTA*, and *IET CDT* journals. He is the director of the Design-for-Excellence Lab at the NYU Abu Dhabi. His recent research in hardware security and trust is being funded by US National Science Foundation, US Department of Defense, Semiconductor Research Corporation, and Mubadala Technology.

Krishnendu Chakrabarty received the BTech degree from the Indian Institute of Technology, Kharagpur, in 1990, and the MSE and PhD degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively. He is now the William H. Younger Distinguished Professor of Engineering in the Department of Electrical and Computer Engineering and Professor of Computer Science, Duke University. He also serves as director of the Graduate Studies for Electrical and Computer Engineering. He received the US National Science Foundation Early Faculty (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 CAD Donald O. Pederson Best Paper award (2015), and 11 best paper awards at major IEEE conferences. He also received the IEEE Computer Society Technical Achievement Award (2015) and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur (2014). He is a research ambassador of the University of Bremen (Germany). His current research projects include: testing and design-for-testability of integrated circuits; digital microfluidics, biochips, and cyberphysical systems; optimization of enterprise systems and smart manufacturing. He has authored 17 books on these topics, published over 550 papers in journals and refereed conference proceedings, and given over 250 invited, keynote, and plenary talks. He has also presented 40 tutorials at major international conferences. He holds six US patents, with several patents pending. He was a 2009 Invitational Fellow of the Japan Society for the Promotion of Science (JSPS). He received 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. He served as a distinguished visitor of the IEEE Computer Society during 2005-2007 and 2010-2012, and as a distinguished lecturer of the IEEE Circuits and Systems Society during 2006-2007 and 2012-2013. He currently serves as an ACM Distinguished Speaker. He served as the editor-in-chief of the *IEEE Design & Test of Computers* during 2010-2012. He currently serves as the editor-in-chief of the *ACM Journal on Emerging Technologies in Computing Systems* and *IEEE Transactions on VLSI Systems*. He is also an associate editor of the *IEEE Transactions on Computers, IEEE Transactions on Biomedical Circuits and Systems, IEEE Transactions on Multiscale Computing Systems, and ACM Transactions on Design Automation of Electronic Systems*. He serves as an editor of the *Journal of Electronic Testing: Theory and Applications (JETTA)*. In the recent past, he has served as associate editor of the *IEEE Transactions on VLSI Systems* (2005-2009), *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2001-2013), *IEEE Transactions on Circuits and Systems I* (2005-2006), and *IEEE Transactions on Circuits and Systems II* (2010-2013). He is a fellow of the ACM and IEEE, and a Golden Core Member of the IEEE Computer Society.

Ramesh Karri received the PhD degree in computer science and engineering from the University of California, San Diego. He is a professor of electrical and computer engineering at the Tandon School of Engineering, New York University. His research interests include trustworthy ICs and processors; High assurance nanoscale IC architectures and systems; VLSI Design and Test; Interaction between security and reliability. He has over 200 journal and conference publications in these areas. These include tutorial articles in IEEE Computer and Proceedings of IEEE on Trustworthy Hardware. His groups work on hardware cybersecurity was nominated for best paper awards (ICCD 2015, DFTS 2015) and received awards at ACM Computer and Communication Security (CCS 2013, DFTS 2013, VLSI Design 2012) and at other competitions (ACM Student Research Competition at DAC 2012, ICCAD 2013, DAC 2014, and the Grand Finals 2013). He was the recipient of the Humboldt Fellowship and the US National Science Foundation CAREER Award. He is the area director for cyber security of the NY State Center for Advanced Telecommunications Technologies, NYU-Poly; co-founded the Center for research in interdisciplinary studies in security and privacy -CRISSP (http://crissp.poly.edu/), co-founded the Trust-Hub (http://trust-hub.org/), and organizes the annual red team blue team event at the NYU, the Embedded Security Challenge (http://www.poly.edu/csaw2014/csaw-embedded). He co-founded the IEEE/ACM Symposium on Nanoscale Architectures (NANOARCH). He served as program and general chair of several conferences including IEEE International Conference on Computer Design (ICCD), IEEE Symposium on Hardware Oriented Security and Trust (HOST), IEEE Symposium on Defect and Fault Tolerant Nano VLSI Systems (DFTS), NANOARCH, RFIDSEC 2015, and WISEC 2015. He serves on several program committees. He was the associate editor of the *IEEE Transactions on Information Forensics and Security* (2010-2014), *IEEE Transactions on CAD* (2014-present), *ACM Journal of Emerging Computing Technologies*(2007-present), *ACM Transactions on Design Automation of Electronic Systems* (2014-present), *IEEE Access* (2015-present), *IEEE Transactions on Emerging Technologies in Computing* (2015-present), and *IEEE Design and Test* (2015-present). He is an IEEE Computer Society Distinguished Visitor (2013-present). He is on the Executive Committee of IEEE/ACM Design Automation Conference leading the Security@DAC initiative (2014).