Tamper-Resistant Pin-Constrained Digital Microfluidic Biochips

Jack Tang
New York University
jtang@nyu.edu

Mohamed Ibrahim
Duke University
msi3@duke.edu

Krishnendu Chakrabarty
Duke University
krish@duke.edu

Ramesh Karri
New York University
rkarri@nyu.edu

ABSTRACT

Digital microfluidic biochips (DMFBs)—an emerging technology that implements bioassays through manipulation of discrete fluid droplets—are vulnerable to actuation tampering attacks, where a malicious adversary modifies control signals for the purposes of manipulating results or causing denial-of-service. Such attacks leverage the highly programmable nature of DMFBs. However, practical DMFBs often employ a technique called *pin mapping* to reduce control pin count while simultaneously reducing the degrees of freedom available for droplet manipulation. Attempts to control specific electrodes as part of an attack cannot be made without inadvertently actuating other electrodes on-chip, which makes the tampering evident. This paper explores this tamper-resistance property of pin mapping in detail. We derive relevant security metrics, evaluate the tamper-resistance of several existing pin mapping algorithms, and propose a new security-aware pin mapper with superior tamper-resistance as compared to prior work.

KEYWORDS

Digital microfluidics, electrode addressing, security, tamper-resistance

1 INTRODUCTION

Digital microfluidic biochips (DMFBs) are platforms for biochemical assays that manipulate fluids in discrete quantities [7]. DMFB technology has made significant strides over the last decade, as their reprogrammable nature is amenable to advanced design automation techniques [22]. Unfortunately, DMFBs are susceptible to actuation tampering attacks—i.e., malicious modifications of control signals—which can achieve disastrous outcomes such as Denial-of-Service (DoS) and assay result manipulation [3]. With the recent commercialization of DMFB systems such as the Baebies SEEKER [4], and high-profile incidents such as the violation of diagnostic integrity at Theranos [25], it is clear that now is a critical time to ensure the security and trustworthiness of microfluidic platforms.

Actuation tampering attacks take advantage of the simple nature of DMFB control signals; they can be easily reverse-engineered to reveal the underlying protocol [6] and then modified to perform arbitrary fluid operations. These control signals, termed *actuation sequences*, are computer-generated through a high-level synthesis flow [22]. Subsequently, a step called *pin mapping* can reduce the number of pins required to drive the DMFB while at the same time reducing the degrees of freedom available for droplet manipulation. This also reduces the types of attacks that an attacker can execute. Pin mapping may force attacks to cause inadvertent droplet movements, making them detectable. Therefore, pin-mapped DMFBs are in some sense tamper-resistant.

1.1 Contributions

This work explores the concept of pin-mapping-as-tamper-resistance in detail through the following contributions:

- We present the first security analysis of broadcast addressed, pin-constrained DMFB actuation sequences and define the tamper-resistance property with related definitions and security metrics.
- We develop a new tamper-resistant pin mapper based on a greedy heuristic graph clustering algorithm.
- We present experimental evidence on several benchmark DMFB assays to show how the proposed methods improve security with modest overhead, as compared against prior pin mapping algorithms.

1.2 Related Prior Work

Research on security and trustworthiness of microfluidic systems began in earnest with the study of cyberphysical digital microfluidic biochips. Subtle result manipulation attacks on glucose assays were described in [3], while DMFB supply chain security was evaluated in [1]. Randomized checkpoint systems were proposed to detect attacks in real time [23]. Reverse-engineering attacks were systematized in the BioChipWork framework [6]. An IP protection scheme based on the concept of a "fluidic multiplexer" was used to realize "fluidic encryption," which requires the application of a fluid-based secret key for the assay to function properly [2]. A PUF-based digital rights management scheme was proposed in [13], while a method to localize attacks on actuation sequences was proposed in [19]. Beyond digital microfluidics, transposer-based routing fabrics have been studied for their security properties under fault injection attacks [24].

2 BACKGROUND

A DMFB operates on the principle of electrowetting-on-dielectric. Fluid droplets on a hydrophobic surface change their contact angle with the substrate when an electric field is applied to the underlying electrode. By placing a patterned grid of electrodes on a substrate and carefully applying a sequence of control voltages (actuation sequences), operations such as dispensing, transporting, mixing, and spitting of droplets can be achieved (Fig. 1). These operations can then be utilized as part of a complex bioassay for applications as diverse as proteomics and DNA sample preparation [7].

DAC '18, June 24–29, 2018, San Francisco, CA, USA

ACM ISBN 978-1-4503-5700-5/18/06…$15.00
https://doi.org/10.1145/3195970.3196125

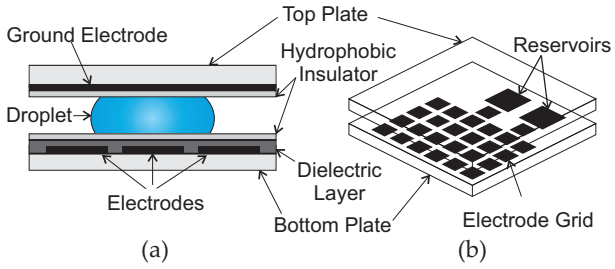

Figure 1: Structure of a DMFB. (a) Side view. Droplet contact angle is modulated by an electric field applied to the electrodes. (b) A general-purpose DMFB consists of a patterned grid of electrodes upon which droplets can be manipulated.

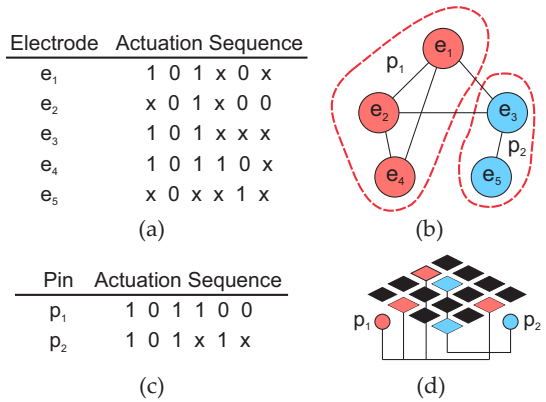

Figure 2: Broadcast addressing. (a) An example actuation sequence for five electrodes. (b) The compatibility graph, with results of the clique selection outlined in dashed lines. (c) The resulting broadcast-addressed, pin-mapped actuation sequence. (d) Electrodes are physically wired together and brought out to pins for connection to a controller.

2.1 DMFB High-Level Synthesis

High-level synthesis design flows are used for the automatic generation of DMFB control signals [22]. A specification for the bioassay to be executed on-chip is written in a high-level descriptive language and passed to the synthesis software for processing. The synthesis flow typically consists of scheduling, placement, and routing phases, although alternate flows have been proposed which tackle all phases simultaneously. Common optimizations include reliability, testability, and execution time [5, 22]. The output of the synthesis flow is the actuation sequence—a sequence of pin activations that can be applied directly to the DMFB.

One of the largest contributors to a DMFB's cost and complexity is the number of pins required to drive it [12]. The earliest general-purpose DMFBs required one IO pin from the driver circuitry for each electrode on-chip, a scheme termed *direct addressing*. This can quickly become impractical, even for modestly sized designs. For instance, a demonstrated immunosaasay DMFB targeted for point-of-care testing requires over 1, 000 pins [21]—which easily exhausts the number of pins on common MCU packages—if direct addressing is used. *Pin-constrained* DMFBs reduce the pin count with a restriction on the droplet degrees of freedom. Pin-constrained DMFBs can be generated as the final step in the high-level synthesis flow, or can considered in the overall biochip design [10, 17]. The same immunosaasay biochip described in [21], when pin-constrained, uses only 64 pins to drive over 1,000 electrodes.

2.2 Broadcast Addressing

Many post-synthesis pin mappers are based upon broadcast electrode-addressing schemes, which hardwires electrodes into pins receiving the same sequence of control signals [27]. Broadcast addressing relies on the concepts of *don't-care* values and *compatible sequences*.

Don't-cares: On a DMFB grid, the movement of a single droplet requires a single pin activation (represented in the actuation sequence as a 1) to change the contact angle and initiate movement, and the deactivation (represented as 0) of the surrounding pins to ensure that the droplet does not inadvertently split. Any other electrode not directly involved in this transfer is a don't-care (represented by *x*), and can be held either high or low. The convention is chosen by the biochip designer, though typically it is held low.

Compatible sequences: Two electrode actuation sequences are compatible with each other if each value is either identical or at least one electrode contains a don't-care. Two compatible sequences can be combined into one by replacing don't-cares with the other electrode's actuation value. This way, the two electrodes can be tied to the same pin receiving the same set of instructions. Hence, the term broadcast addressing.

Generation of a broadcast addressing scheme relies on graph-based representations of electrode relationships [27]. Vertices represent electrodes while edges represent relationships between compatible electrodes. Graph cliques can then be identified and partitioned, with the partition representing a collection of pins that can be shorted to a single driving pin (Fig. 2). This problem is NP-hard, but can be solved using heuristics [27]. Extensions to this basic concept include: reliability enhancement by reducing switching frequency and consequently reducing the degradation in contact angle [26], insertion of "ground vectors" for preventing residual charge [14], power consumption reduction through elimination of "redundant actuation units (RAUs)" [15]. We refer to these as toggle-aware, GV-aware, and RAU-aware pin mappers, respectively, for consistency with the literature [11]. An optimal broadcast addressing scheme was developed in [8], achieving information-theoretical minimal pin counts. However, this scheme relies on the integration of digital logic in the biochip, which is impractical and yet to be demonstrated. Therefore we consider such DMFBs to be outside the scope of those studied in this paper.

2.3 Actuation Tampering Attacks

An actuation tampering attack is a malicious modification of the actuation sequences used to control a DMFB, and were first reported in [3]. The attack describes the mechanism by which an attacker can achieve various malicious outcomes, such as denial-of-service or result manipulation. Actuation tampering can be carried out through many different attack vectors, including alteration of data in program memory, modification of the software used to generate actuation sequences, or physical injection of hardware faults.

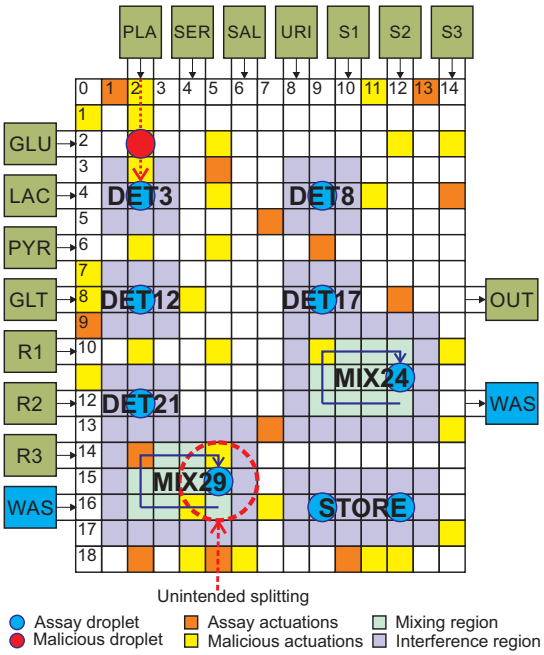

Figure 3: Tamper-resistance due to pin mapping: An attacker can modify the actuation sequence to produce the malicious droplet route in dotted lines. This attack would alter the final concentration reading. The yellow electrodes are unintentionally activated in the course of the attack due to pin mapping. Unintentional splitting may occur as a result of this, making the attack evident. This pin mapping was derived from the original broadcast clique-based strategy [27].

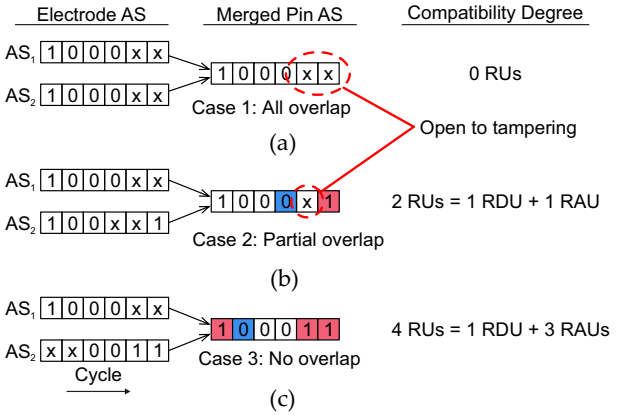

Figure 4: Actuation sequence (AS) merging. (a) Case 1: don't-cares exactly overlap. This is undesirable as an attacker can use the resulting pin-level don't-cares to insert malicious actuations. (b) Case 2: partial overlap. A redundant deactivation unit (RDU) results when a broadcast 0 drives a don't-care, while a redundant activation unit (RAU) is a 1 driving a don't-care. (c) Case 3: no overlap. This is the most desirable from a security standpoint; an attacker is forced to affect the operation of the assay since there are no don't-cares.

3 SECURITY ANALYSIS

Here, we describe the concept of tamper-resistance and how it arises as a result of pin-mapped DMFBs. We then describe the threat model and derive security metrics.

3.1 Tamper-Resistance

We claim that *the tamper-resistance of a DMFB is determined by the number and distribution of don't-cares in the pin-mapped actuation sequence.* To illustrate, consider the attack in Fig. 3. This 15×19 DMFB is executing the InVitro 4x4 multiplexed diagnostic assay, which measures glucose (GLU), lactate (LAC), pyruvate (PYR) and glutamate (GLT) in plasma (PLA), serum (SER), saliva (SAL), and urine (URI) [22]. The attacker routes an extra plasma droplet along the red dashed line in order to alter the concentration of the sample being detected in DET3.

Normally, in a direct-addressed DMFB, such an attack is trivial to implement and would have no consequences. In a pin-mapped design, all the electrodes in yellow are unintentionally activated in the course of routing the malicious droplet. This could result in the violation of design rules, such as in MIX29, where the droplet being mixed would be unintentionally split. Subsequently, the mix operation could fail, stopping the assay progression. Or, if a randomized checkpoint system is implemented, it may detect a stray droplet where none should exist [23]. An attacker can avoid this problem by targeting pins that drive a large number of don't-cares (i.e. electrodes that are not driving droplets, nor part of an interference region). Therefore, qualitatively, a tamper-resistant DMFB design will distribute don't-cares so attackers are constrained in their ability to make arbitrary droplet manipulations.

We introduce the following concepts which we will use to analyze and design tamper-resistant pin-constrained DMFBs:

Redundant Units (RUs) are don't-cares masked by either a 0 or a 1 due to the broadcast addressing scheme. This is a generalization of the redundant actuation unit (RAU)—first introduced in the power-aware pin mapper [15]—which is a don't-care that is masked by a 1. We also define a redundant deactivation unit (RDU) as a don't-care masked by a 0. For convenience, we define the function $RU(p,t)$ which returns the number of redundant units associated with pin p at timestep t, as well $RAU(p,t)$ and $RDU(p,t)$, which return number of RAU's and RDU's, respectively. These functions map pins and time-steps to integers that are pin-dependent, i.e., $RU/RAU/RDU : P \times T \rightarrow \mathbb{Z}$, where P is the set of all pins, T is the set of all assay time-steps. We also define $\boldsymbol{RU}$, $\boldsymbol{RAU}$, $\boldsymbol{RDU}$, as the set of all RU's, RAU's, and RDU's, respecitvely. A don't-care can be masked by either a 0 or a 1, but not both, so $\boldsymbol{RAU}$ and $\boldsymbol{RDU}$ are disjoint. The union of these two sets is equal to the set $\boldsymbol{RU}$. Therefore,

$$\sum_{p \in P, t \in T} RU(p,t) = \sum_{p \in P, t \in T} RAU(p,t) + \sum_{p \in P, t \in T} RDU(p,t) \quad (1)$$

Compatibility Degree (CD) measures how desirable it is to merge two electrode actuation sequences and generalizes the binary concept of compatibility. Consider two electrode actuation sequences AS_1 and AS_2, with m and n don't-cares, respectively. There are three ways in which the don't-cares can be arranged in time. In Case 1 (Fig. 4(a)), all don't-cares overlap exactly and gives an attacker an

opportunity for actuation tampering. In Case 2 (Fig. 4(b)), some subset of the don't-cares overlap, and in Case 3 (Fig. 4(c)), none overlap. Case 3 is the best-case scenario in terms of tamper-resistance; there are no exposed don't-cares, and to target a masked don't-care, an attacker will risk modifying the normal execution of the assay. Therefore, we compute the compatibility degree $CD(e_1, e_2)$ as the number of redundant units that result from the merging of two electrode actuation sequences associated with electrodes e_1 and e_2.

3.2 Threat Model

We assume that the *attacker* is a remote party who can access the DMFB platform through the network. Controllers for DMFBs typically incorporate a network interface either by default (e.g., when using off-the-shelf embedded computers), or by design for firmware updates and sensor data processing. The attacker is able to conduct stealthy actuation tampering attacks, i.e., extract the synthesized actuation sequences from memory, reverse-engineer them, and alter them. *The attacker does not want to be detected.* The extent of the alteration can range from simple augmentation or deletion of sequences, or can be as comprehensive as total replacement. Potential malicious actors and their motivations are discussed in [3]. The *defender* is the DMFB platform designer who wishes to ensure that any modifications to the actuation sequences are easily detectable.

3.3 Attack Constraints

While the threat model grants an attacker tremendous capabilities, in practice, several factors will cause attacks to become evident to the end user. Therefore, arbitrary actuation sequence modifications may not be feasible due to the following constraints:

(1) *Completion time.* Assays may have completion times that are known to the end user. Relatively simple assays, e.g., sample preparation, can be assumed to execute in constant time. Such assays are commonly used as benchmarks in the DMFB literature and are studied in this work. More complex assays with multiple branching points depending on intermediate results have variable execution times [16]. Still, an end user may suspect incorrect execution if an assay completes much faster or slower than their experience suggests is normal.
(2) *Error recovery.* DMFBs are known to be prone to several hardware faults. Cyberphysical integration has been proposed to detect and recover from errors [18]. The design of these mechanisms require fine tuning on the error tolerance, which may be exploitable for carrying out an attack. Furthermore, since placement of error recovery inspection points (i.e. *checkpoints*) is deterministic, a resourceful attacker could simply avoid making changes directly in critical paths.
(3) *Intrusion detection.* Intrusion detectors can monitor parts of the biochip that are not actively sensed by error recovery systems. Deterministic detection can in theory provide 100% security, but in practice, a low overhead scheme (e.g. randomized checkpoints [23]) must be implemented.
(4) *Attack surface.* We consider network-based attacks where the actuation sequence can be recovered and modified at-will. Physical fault injection attacks are possible on DMFBs, but these typically present poor localization and would be unlikely to result in a stealthy attack.
(5) *Reverse engineering.* Many state-of-the-art designs for DMFBs store the actuation sequence in a format that has a one-to-one mapping between encoded bits and the biochip. Reverse engineering is thus straightforward [6]. If some mechanism were introduced to obfuscate the mapping, the attacker would not be able to make controlled changes to the assay.

3.4 Threat Model Refinement

We now refine the threat model according to the attack constraints:
Increasing or decreasing the number of time-steps in the actuation sequence is prohibited. This is to satisfy Constraint 1 for the non-conditional assays studied in this work. Even slight variations in the actuation sequence length can result in noticeable execution time differences, as DMFB actuation periods are often on the order of milliseconds (which is coarse enough to be detected by a stopwatch). Therefore, the attack can only consist of *modifications* of the actuation sequence.
The number of modifications to the actuation sequence must be minimized. This is to avoid detection by either the end user, or detection by a checkpoint system (Constraints 2 & 3). In some cases, the effect of making an incremental change in the actuation sequence can be quantified; if a randomized checkpoint system is implemented, each additional change exponentially increases the probability of being detected [23].
Modifications to the actuation sequence will preferentially target don't-cares. To do otherwise would be to modify actuations (1s) inserted to control droplets or deactivations (0s) inserted as part of an interference region. On pin-constrained designs, modifying a pin-level actuation will change several electrode states. Therefore, if an attacker's goal is to control a single electrode, attacking a pin may cause unintentional changes to other electrodes, potentially causing a detectable change in assay execution.

3.5 Security Metric: Coverage

Based on the previous discussion, it is clear that it is desirable to mask as many don't-cares as possible. Thus, we define:
Redundant Unit Coverage (RUC): It is defined as the proportion of electrode-level don't-cares that are masked by pin-level actuations (i.e. redundant units) over all pins and all assay time-steps. Therefore, it can be calculated as

$$\text{RUC} = \frac{\text{\# redundant units}}{\text{\# total don't-cares}} \tag{2}$$

RUC should be maximized. In the ideal case, coverage is equal to 100%, meaning that there are absolutely no exposed don't-cares for an attacker to leverage.
Proximity Coverage Class (PCC): A variation of RUC such that only electrodes within the vicinity of a droplet are counted.

$$\text{PCC} = \frac{\text{\# redundant units near any droplet}}{\text{\# total don't-cares near any droplet}} \tag{3}$$

Here, "near any droplet" means adjacent to the interference region along the x or y axis of a droplet. This coverage metric narrows the scope to attacks targeting assay droplets. That is, we exclude electrodes far from any assay droplets since they are unlikely to be used for manipulation attacks.

4 DESIGN OF TAMPER-RESISTANT DMFBS

We propose to increase tamper-resistance by designing a pin mapper that is optimized to maximize the coverage of redundant units. This will prevent an attacker from making stealthy modifications to the actuation sequence.

4.1 Problem Statement

The formal problem statement is described as follows:
Input: A DMFB architecture $\mathcal{A}$ consisting of a set of electrodes $\mathcal{B}$ and a set of electrode actuation sequences $\mathcal{AS}$.
Output: A pin-constrained DMFB design assigning each electrode to a set of pins $\mathcal{P}$, where $|\mathcal{P}| < |\mathcal{B}|$, and a set of pin-mapped actuation sequences $\mathcal{AS}_{PM}$.
Objective: Maximize the tamper-resistance by maximizing the redundant unit coverage (RUC).

4.2 Proposed Solution

The problem of grouping electrodes into pins can be modeled as a graph partitioning problem [27]. Here, we are also concerned with grouping electrodes into pins but now have imposed an additional constraint due to the desire to maximize tamper-resistance. Based on our definition of compatibility degree, merging of highly compatible electrodes results in a tamper-resistant design. Therefore, the DMFB design is modeled by a graph $G = (V, E)$ where each vertex $v \in V$ represents an electrode on the DMFB array, and the set of edges E represent relationships between two compatible electrodes, similar to the original broadcast strategy. However, we now include an edge weighting function $w : E \rightarrow \mathbb{Z}$ that evaluates the compatibility degree between the two electrode actuation sequences, and a "color" function $c : V \rightarrow \mathbb{Z}$, which represents the pin assignment.

By grouping electrodes that are highly compatible, we promote solutions that increase the number of redundant units, and therefore result in a more tamper-resistant design. The grouping of high dimensional data represented by graphs is known as the graph clustering problem [20]. In particular, our specific problem of forming k number of pins out of the graph vertices associated with a distance function (i.e. "compatibility degree") is known as the minimum k-clustering problem, which is known to be NP-hard [9]. We therefore propose a greedy heuristic graph-based algorithm to solve the tamper-resistance optimization problem (Fig. 5), which proceeds as follows:

(1) Initialize the graph and weights (G, w).
(2) Form an initial guess p_{init} for the number of pins to form.
(3) Starting from the highest-weighted edges, assign their corresponding two vertices (electrodes) to a pin by setting their color to a unique value, and skipping vertices that have already been assigned.
(4) Repeat forming new pins until p_{init} pins have been formed.
(5) Expand the pins by greedy iteration. For each pin, examine all neighbor vertices that are not-yet assigned, and clique-compatible with the pin, add the vertex with highest compatibility, then move on to the next pin.
(6) Repeat until either all electrodes are assigned or no more valid neighbors can be added.
(7) Repeat the overall procedure on the remaining unassigned electrodes, and failing that, assign each an individual pin.

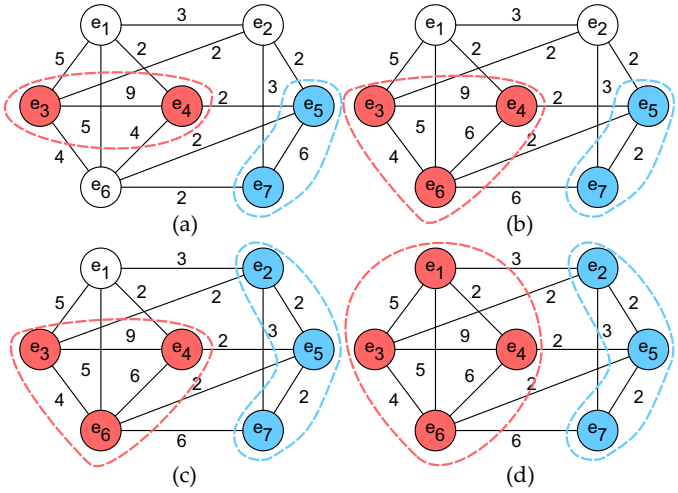

Figure 5: Heuristic tamper-resistant pin mapping. (a) Initial phase of the heuristic algorithm. Vertices represent electrodes, weighted edges represent the compatibility degree between compatibile electrodes. The highest weighted edges are selected for the initial set. (b) First iteration expands the red pin by selecting a clique-compatible vertex with highest compatibility. (c) Second iteration expands the blue pin. (d) Completion of the procedure.

This is a fast heuristic algorithm that attempts to group together pins that are most compatible, with complexity $O(|V|)$ since an electrode is greedily assigned at each step. The initial guess for the pin count can be established through trial-and-error, or using knowledge of typical broadcast-addressed pin counts.

5 EXPERIMENTAL RESULTS

We evaluated our tamper-resistant pin mapper against the broadcast addressing [27], RAU-aware [15], ground-vector-aware (GV-aware) [14], and toggle-aware [26] pin mappers using four benchmark assays: PCR, InVitro 4x4, Protein, and Protein Split 5 (which we refer to as the A, B, C, and D assays, respectively). The benchmark simulation data was generated with the open-source MFStaticSim tool [11] using a 15×19 DMFB array, virtual topology placer and Roy maze router. We imported this data for analysis and implemented our tamper-resistant pin mapper in MATLAB. We summarize the tamper-resistance performance in Table 1, where RUC and PCC are measured in percentages. We also include pin-count ($|P|$) and number of switching toggles (SW) measured in thousands, as an inversely related indicator of power and reliability.

5.1 Comparison with Prior Work

We see that tamper-resistant pin mapper achieves, on average, 66.3% higher RUC than the next-best prior work, which is the broadcast clique pin mapper. When we restrict attacks to the proximity coverage class (PCC), coverage can be as high as 65.6% for some assays using the tamper-resistant pin mapper. Most of the pin mappers achieve coverage rates of less than 40% across all assays. At the same time, this work's pin counts are comparable with all other

Table 1: Pin Mapper Performance Comparison. |P| = pin count, SW = thousands of switching toggles, RUC = redundant unit coverage, PCC = proximity coverage class. A = PCR, B = InVitro 4x4, C = Protein, D = Protein Split 5.

	Broadcast [27]				RAU-Aware [15]				GV-Aware [14]				Toggle-Aware [26]				This Work			
Assay	|P|	SW	*RUC*	*PCC*	|P|	SW	*RUC*	*PCC*	|P|	SW	*RUC*	*PCC*	|P|	SW	*RUC*	*PCC*	|P|	SW	*RUC*	*PCC*
A	19	16.7	42.3	34.4	30	9.5	26.6	18.7	28	10.1	28.6	20.6	21	15.3	37.7	34.9	28	21.7	61.3	65.6
B	65	74.5	37.9	33.7	69	44.5	28.7	22.2	66	46.3	29.3	25.3	70	53.8	32.5	29.3	96	70.2	51.1	40.3
C	52	133.6	24.2	24.4	55	74.5	23.5	22.5	55	89.1	24.2	22.1	55	133.6	25.3	25.0	72	175.8	41.4	40.9
D	90	343.4	18.4	17.0	94	265.6	16.3	13.0	94	266.1	16.3	12.8	95	314.7	18.1	15.2	121	376.9	31.7	24.8

pin mappers, while switching activity is increased but remains on the same order. Therefore, we conclude that the proposed algorithm achieves its goal and is able to produce a quantifiably more tamper-resistant pin-constrained DMFB design. We also note that there exists trade-off between performance-related optimizations and security. This has been a recurring theme in the security and electronic design literature.

6 CONCLUSION

We presented the first study of DMFB pin mappers as a tamper-resistance mechanism. The restriction on droplet movements imposed by pin mappers simultaneously lowers an attacker's ability to arbitrarily route droplets, and causes undesirable side-effects on other droplets existing on the chip. We introduced the redundant unit coverage security metric to describe the masking of don't-cares. Experimental results show that existing pin mapping algorithms, while optimizing for reliability and power consumption, lead to poor tamper-resistance. A new pin mapping algorithm was proposed to increase masking effects. Comparison with prior work shows marked improvement in tamper-resistance, with modest pin count and switching overhead. Some pin count overhead is acceptable since the PCB layer count can form a substantial portion of overall system cost anyway [12].

Incorporating security measures in the pin mapping phase of the DMFB design flow is highly advantageous; no extra circuitry or control hardware is required. While other countermeasures such as encryption could be used to secure DMFBs against actuation tampering, these would incur more hardware and processing overhead while complicating the usage scenario by requiring secret keys. Furthermore, this is a hardware-based technique that is not susceptible to attacks that take advantage of network-enabled controllers.

ACKNOWLEDGMENTS

This research was supported in part by the Army Research Office under grant number ARO W911NF-17-1-0320. Ramesh Karri and Jack Tang are associated with the NYU Center for Cyber Security (cyber.nyu.edu). Ramesh Karri is also associated with the NYU-AD Center for Cyber Security (sites.nyuad.nyu.edu/ccs-ad/).

REFERENCES

[1] Sk Subidh Ali, Mohamed Ibrahim, Jeyavijayan Rajendran, Ozgur Sinanoglu, and Krishnendu Chakrabarty. 2016. Supply-chain security of digital microfluidic biochips. *Computer* 49, 8 (2016), 36–43.
[2] Sk Subidh Ali, Mohamed Ibrahim, Ozgur Sinanoglu, Krishnendu Chakrabarty, and Ramesh Karri. 2016. Microfluidic encryption of on-chip biochemical assays. In *Proc. IEEE Biomed. Circuits Syst. Conf.* 152–155.
[3] Sk Subidh Ali, Mohamed Ibrahim, Ozgur Sinanoglu, Krishnendu Chakrabarty, and Ramesh Karri. 2016. Security assessment of cyberphysical digital microfluidic biochips. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 13, 3 (2016), 445–458.
[4] Baebies, Inc. 2017. Baebies SEEKER. (2017). http://baebies.com/products/seeker/
[5] Krishnendu Chakrabarty. 2010. Design automation and test solutions for digital microfluidic biochips. *IEEE Trans. Circuits Syst. I* 57, 1 (2010), 4–17.
[6] Huili Chen, Seetal Potluri, and Farinaz Koushanfar. 2017. BioChipWork: Reverse Engineering of Microfluidic Biochips. In *Proc. IEEE Int. Conf. Comput. Des.* 9–16.
[7] Kihwan Choi, Alphonsus HC Ng, Ryan Fobel, and Aaron R Wheeler. 2012. Digital microfluidics. *Annu. Rev. Anal. Chem.* 5, 1 (2012), 413–440.
[8] Trung Anh Dinh, Shigeru Yamashita, and Tsung-Yi Ho. 2015. An optimal pin-count design with logic optimization for digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 34, 4 (2015), 629–641.
[9] Teofilo F Gonzalez. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38 (1985), 293–306.
[10] Daniel Grissom and Philip Brisk. 2013. A field-programmable pin-constrained digital microfluidic biochip. In *Proc. IEEE/ACM Des. Autom. Conf.* 46.
[11] Daniel Grissom, Christopher Curtis, Skyler Windh, Calvin Phung, Navin Kumar, Zachary Zimmerman, O'Neal Kenneth, Jeffrey McDaniel, Nick Liao, and Philip Brisk. 2015. An open-source compiler and PCB synthesis tool for digital microfluidic biochips. *INTEGRATION, the VLSI journal* 51 (2015), 169–193.
[12] Daniel T Grissom, Jeffrey McDaniel, and Philip Brisk. 2014. A low-cost field-programmable pin-constrained digital microfluidic biochip. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 33, 11 (2014), 1657–1670.
[13] Ching-Wei Hsieh, Zipeng Li, and Tsung-Yi Ho. 2017. Piracy prevention of digital microfluidic biochips. In *Proc. Asia South Pacific Des. Autom. Conf.* 512–517.
[14] Tsung-Wei Huang, Tsung-Yi Ho, and Krishnendu Chakrabarty. 2011. Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips. In *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.* 448–455.
[15] Tsung-Wei Huang, Hong-Yan Su, and Tsung-Yi Ho. 2011. Progressive network-flow based power-aware broadcast addressing for pin-constrained digital microfluidic biochips. In *Proc. IEEE/ACM Des. Autom. Conf.* 741–746.
[16] Mohamed Ibrahim, Krishnendu Chakrabarty, and Kristin Scott. 2017. Synthesis of cyberphysical digital-microfluidic biochips for real-time quantitative analysis. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 36, 5 (2017), 733–746.
[17] Yan Luo and Krishnendu Chakrabarty. 2013. Design of pin-constrained general-purpose digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 32, 9 (2013), 1307–1320.
[18] Yan Luo, Krishnendu Chakrabarty, and Tsung-Yi Ho. 2013. Error recovery in cyberphysical digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 32, 1 (2013), 59–72.
[19] Pushpita Roy and Ansuman Banerjee. 2016. A new approach for root-causing attacks on digital microfluidic devices. In *Proc. IEEE Asian Hardware-Oriented Security Trust Symp.* 1–6.
[20] Satu Elisa Schaeffer. 2007. Graph clustering. *Comput. Sci. Rev.* 1, 1 (2007), 27–64.
[21] Ramakrishna Sista, Zhishan Hua, Prasanna Thwar, Arjun Sudarsan, Vijay Srinivasan, Allen Eckhardt, Michael Pollack, and Vamsee Pamula. 2008. Development of a digital microfluidic platform for point of care testing. *Lab. Chip* 8, 12 (2008), 2091–2104.
[22] Fei Su and Krishnendu Chakrabarty. 2008. High-level synthesis of digital microfluidic biochips. *ACM J. Emerg. Technol. Comput. Syst.* 3, 4 (2008), 1.
[23] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2017. Secure Randomized Checkpointing for Digital Microfluidic Biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* (2017).
[24] Jack Tang, Mohamed Ibrahim, Krishnendu Chakrabarty, and Ramesh Karri. 2017. Security Trade-offs in Microfluidic Routing Fabrics. In *Proc. IEEE Int. Conf. Comput. Des.* 25–32.
[25] The Wall Street Journal. 2016. Theranos Results Could Throw Off Medical Decisions, Study Finds. (March 2016). http://www.wsj.com/articles/theranos-results-could-throw-off-medical-decisions-study-finds-1459196177
[26] Shang-Tsung Yu, Sheng-Han Yeh, and Tsung-Yi Ho. 2015. Reliability-driven chip-level design for high-frequency digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 34, 4 (2015), 529–539.
[27] Yang Zhao, Tao Xu, and Krishnendu Chakrabarty. 2011. Broadcast electrode-addressing and scheduling methods for pin-constrained digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 30, 7 (2011), 986–999.