FlowTrojan: Insertion and Detection of Hardware Trojans on Flow-Based Microfluidic Biochips

Huili Chen
University of California, San Diego
Email: huc044@ucsd.edu

Seetal Potluri
North Carolina State University.
Email: spotlur2@ncsu.edu

Farinaz Koushanfar
University of California, San Diego
Email: farinaz@ucsd.edu

***Abstract*—We propose FlowTrojan, the first systematic framework for insertion and detection of Hardware Trojans (HTs) on Flow-based Microfluidic Biochips (FMFBs). The FMFB is an emerging platform with critical usages in the medical field due to the handling of sensitive information. We discuss the attack model where the malicious foundry aims to compromise the on-chip control circuitry. FlowTrojan is designed to automatically extract the netlist for the control circuitry from the layout and explore the internal independence between regions on FMFBs for partitioning. We demonstrate that HT triggers can feature a low activation probability while placed on the non-critical timing path to stay clandestine during functional and parametric testing. To avoid such attacks, FlowTrojan provides a parallel regime of control-value (CV) based HT detection as the countermeasure. Experimental results corroborate the effectiveness and scalability of the proposed attack and detection schemes.**

I. Introduction

Lab-on-Chip (LoC) technologies have been developed to address the growing requirement for miniaturized platforms and facilitates various on-site applications such as point-of-care testing, biomedical diagnosis, and environment monitoring [1]. Flow-based Microfluidic Biochips (FMFB) is an emerging branch in LoC that manipulates continuous flow inside microchannels using microvalves. FMFBs deliver various advantages including automation of experiments and low sample input [2], hence are increasingly commercialized by by companies such as Illumina [3], and Fluidigm [4].

While design automation and synthesis techniques have made FMFB devices available [2], [5], the lack of security configuration makes FMFBs vulnerable to potential attacks. Protecting the FMFB is of critical importance since the medical diagnosis or treatment will be compromised if the device is attacked, threatening the lives of patients. Prior works have identified the vulnerabilities of FMFBs while systematic attack frameworks are missing. [6] provides a high-level assessment of potential attacks on cyberphysical FMFBs including Denial-of-Service (DoS), design theft, and information leakage. The authors in [7] survey existing security concerns and the corresponding defense schemes on both droplet-based and flow-based microfluidic biochips. In contrast to outlining the feasibility of various attacks and defense as shown in the previous papers, we focus on identifying the susceptibility of FMFBs to Hardware Trojan attacks.

Developing a stealthy HT insertion methodology and a corresponding detection scheme for FMFBs is challenging due to two main reasons: (i) The manufacturing material and the working mechanism of FMFBs are fundamentally different from Silicon-based digital circuits [8], thus nullifying the direct adoption of existing HT techniques on Silicon ICs. (ii) The functionality and the layout of the FMFB is getting more complicated while the size of the biochip is decreasing.

KIn this paper, we present FlowTrojan framework as the first holistic solution that resolves the above two challenges. More specifically, FlowTrojan systematically characterizes the vulnerability of the on-chip control circuitry of the FMFB. We show how the netlist of the integrated control circuit can be extracted and partitioned using the intrinsic independence of signal sets on FMFBs. The partitioned sub-netlists are further analyzed to find stealthy nodes for HT insertion and identify critical timing paths for HT placement. The feasibility of the resulting HT is evaluated in simulation. FlowTrojan also provides an innovative HT detection scheme using region-based control-value analysis which flags suspicious pneumatic gates on the target FMFB. FlowTrojan bridges the gap between the valve-based control circuitry of FMFBs and the conventional Silicon-based ICs by modelling the control architecture as the logic circuit. The attack and detection schemes integrated in FlowTrojan feature lower computational complexity and runtime overhead since the *independence* between the actuation signal sets is leveraged for *parallelism*.

II. Preliminaries and Related Work

A. Flow-based Microfluidic Biochips

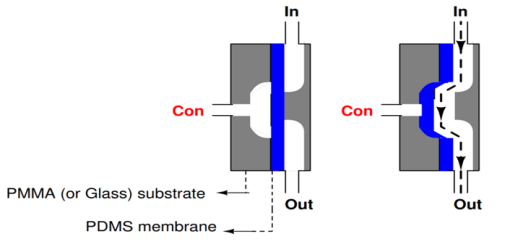


Fig. 1. Schematic diagram of a normally-closed microfluidic valve [9]. The input, output and control ports are denoted by In, Out, and Con, respectively.

FMFBs work by manipulating the transportation of fluids inside the microchannels using microvalves [2]. A typical FMFB consists of two main modules: a flow processor and an on-chip control circuit. The on-chip control circuit deploys pneumatic gates that operate on pressure and vacuum. Fig. 1 shows the schematic view of a normally-closed microfluidic valve consisting of three layers. The valve is closed by default

978-1-7281-7044-2/20/$31.00 ©2020 IEEE

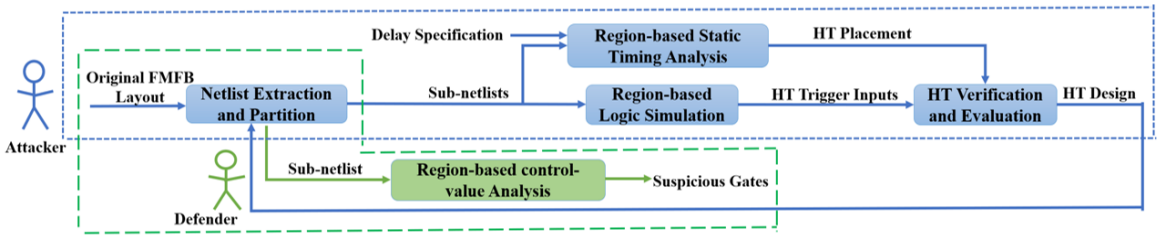
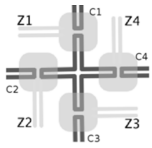


Fig. 2. Global flow of the proposed HT attack and detection.

if no vacuum is applied as shown in Figure 1(a). When vacuum is applied on the control port of the microvalve ('pneumatic gate'), the PDMS membrane is pulled towards the chamber, thus allowing the pneumatic flow to transport between the 'In' and 'Out' as shown in Fig. 1(b).

B. On-chip Pneumatic Control Circuitry of FMFBs

Fig. 3(a) shows the design of an omni-directional switch that consists of 4 microfluidic valves (Z1-Z4) and 4 microfluidic channels (C1-C4). The actuation table for this switch is shown in Fig. 3(b). Microvalves shown in Fig. 1 are used as the basic building blocks to build logic gates including NOT, NOR, NAND. The pneumatic gates are then employed to construct a complete pneumatic control logic netlist [9]. FlowTrojan explores the vulnerability of the on-chip pneumatic control circuitry for stealthy HT insertion.



(a) Omni-directional switch

Phase	Z1	Z2	Z3	Z4
C1 - C2	1	1	0	0
C1 - C3	1	0	1	0
C1 - C4	1	0	0	1
C2 - C3	0	1	1	0
C2 - C4	0	1	0	1
C3 - C4	0	0	1	1

(b) Actuation table

Fig. 3. Diagram of an omni-directional switch (a) and its corresponding actuation table (b) [9].

C. Related Work

Prior works have identified the susceptibility of droplet-based microfluidic biochips (DMFBs) to assay manipulation and piracy [10], [11]. The security concerns about FMFBs are investigated in [6] and [7] where a high-level overview of possible attacks and defense methods is outlined. [8] demonstrates the first practical layout-level reverse engineering attack on FMFBs using image analysis. Our work is distinct from [8] since unlike the reverse engineering attack, the HT attack is required to be stealthy and undetectable. Therefore, designing a covert HT is a more challenging task.

III. FlowTrojan Methodology

Threat Model. We assume that the adversary is the untrusted foundry which obtains the FMFB architecture description and the delay specification from the FMFB vendor. The attacker intends to insert a stealthy HT into the FMFB design that is manufactured and delivered to the end user. As the result of the malicious modification, the execution of desired bioassay will be disturbed, producing an incorrect outcome.

A. Attack Methodology

The work flow of FlowTrojan is shown in Fig. 2. Firstly, the the on-chip pneumatic control circuit of the FMFB is reverse engineered to a gate-level netlist. The recovered netlist is then partitioned into smaller sub-netlists using the independence between the signal sets. Logic simulation and standard STA are performed on each region in parallel. The trigger is carefully designed and placed in order to minimize the activation probability and delay impact. The compatibility of the trigger inputs is ensured by the SAT solver. We show that the HT as simple as a 2-AND gate can be effective and stealthy. The details of each step are explained below.

- **Netlist extraction and partition:** FlowTrojan first performs netlist extraction and partition via reverse engineering the hardware description of the FMFB layout. Control components and control tubes are analogous to logic gates and wires in digital circuits, respectively. Since we aim to insert the HT into the control circuitry, FlowTrojan ignores the flow architecture and only extracts the netlist of the control architecture denoted by the set $\mathbf{N}$. Dependent signals in $\mathbf{N}$ are grouped into clusters and denoted by $\mathbf{N_1}, ..., \mathbf{N_K}$.
- **Region-based logic simulation:** Logic simulation generates random test vectors and applies the vectors on the netlists obtained from the first step. The evaluation can be performed on the netlist $\mathbf{N}$ once (referred as the baseline), or on the partitioned sub-netlists $\mathbf{N_1}, ..., \mathbf{N_K}$ simultaneously (referred as the parallel implementation). The later approach explores the isolation between signal sets present on the netlist. FlowTrojan deploys the region-based method since it brings memory saving and speedup. We define the memory saving R_m as the ratio of the size of all possible test vectors in the baseline regime and its size in the parallel regime:

$$R_m = \frac{\#N \cdot 2^{\#N}}{\sum_{r=1}^{r=K} \#N_r \cdot 2^{\#N_r}} \quad (1)$$

A number of rarely activated signals which satisfy $P_1 \leq \epsilon$ are identified and stored in the set $\mathbf{R}$.

- **Region-based Static Timing Analysis:** To prevent side-channel based HT detection, STA is performed to compute early slacks and late slacks of each sub-netlist. Critical points (nodes with negative slacks) in $\mathbf{R}$ are removed by region-based STA and the remaining ones are stored in the set $\mathbf{S}$. Finally, timing-critical paths with the maximum path delay are identified and avoided during HT placement.
- **HT verification and evaluation:** The SAT solver is deployed to check if a pair of two nodes (S_i, S_j) chosen from the set $\mathbf{S}$ can be simultaneously activated to 1. If two nodes are verified to be compatible, the joint probability $P_{Trigger} = P_1(S_i \cdot S_j)$ is computed. The verification continues until all compatible pairs are identified and the corresponding $P_{trigger}$ is calculated. The node pair with the minimal $P_{trigger}$

is selected as the trigger input. Statistical simulation is applied to assess the performance of the resulting HT.

B. Control-value based HT Detection

Besides the HT attack, FlowTrojan also presents the first systematic methodology for identifying suspicious gates on FMFBs using region-based control-value analysis. To the best of our knowledge, there is no work on detecting potential HT on FMFBs. Alg. 1 outlines our HT detection scheme. The input sub-netlists $\mathbf{N_1}, ..., \mathbf{N_K}$ are obtained by netlist extraction and partition as described above.

Algorithm 1 Framework of HT Detection for FMFB

INPUT: Partitioned netlist $\mathbf{N_1}, ..., \mathbf{N_K}$, Boolean decomposition function f, user defined threshold ϵ and the number of selected rows M.

OUTPUT: Suspicious gates in sub-netlists $\mathbf{N_1}, ..., \mathbf{N_K}$.

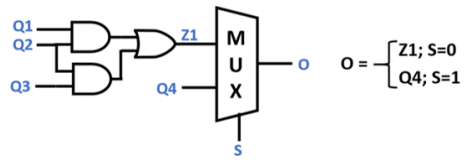
▷ Paralleled region-based CV computation.

for $1 \leq r \leq K$ **do**
  for $1 \leq i \leq length(\mathbf{N_r})$ **do**
    gate $g \leftarrow \mathbf{N_r}(i), T_g \leftarrow TruthTable(g)$;
    for $1 \leq j \leq \#Inputs(g)$ **do**
      Pick jth input: $w \leftarrow Inputs(g)[j]$;
      if w is an intermediate wire **then**
        $w \leftarrow f(Q_i, ...Q_L)$
        $T_w \leftarrow TruthTable(w)$
        Replace w with $(Q_i, ...Q_L)$ in T_g
        for $1 \leq m \leq L$ **do**
          $t_i = EstCV(T_w, Q_i, M)$
          $CV^{'}(Q_i) = EstCV(T_g, Q_i, M)$
        $CV(w) = \sum_{i=1}^{L} t_i \times CV^{'}(Q_i)$
      else
        $CV(w) = EstCV(T_g, w, M)$
    $CV(g) = median\{CV(w_1), ..., CV(w_{max}\}$
    if $CV(g) < \epsilon$ **then**
      Flag gate g as suspicious

Control value (CV) of a primary input is defined as the probability that the change of the input results in the change of the gate output. For each input Q_i, all other input columns in the truth table are held fixed and $CV(Q_i)$ is represented by the fraction of rows whose outputs are influenced by Q_i [12]. A gate g with n input wires has a vector $\mathbf{CV}$ with n elements and its control value $CV(g)$ is defined as the median of the vector. For an intermediate output wire w, we derive the Boolean expression of w in terms of PIs $w = f(Q_1, ..., Q_L)$ and define $CV(w)$ as the weighted sum of control values of PIs: $CV(w) = \sum_{i=1}^{L} t_i \times CV^{'}(Q_i)$. The weight t_i is the control value of Q_i computed from the truth table of w. $CV^{'}(Q_i)$ is obtained by replacing w with the string $(Q_1, ..., Q_L)$ in the truth table of g and holding all inputs fixed except for Q_i. If multiple wires are replaced by strings, we treat each string as an independent set of PIs.

Fig. 4 shows how to compute the control values of a multiplexer. Control values of independent wires are computed by definition: $CV(S) = CV(Q_4) = \frac{8}{2^4} = 0.5$. For intermediate wire $Z_1 = Q_1 \cdot Q_2 + Q_2 \cdot Q_3$, we replace Z_1 with the string (Q_1, Q_2, Q_3) in the truth table of MUX. The truth table of Z_1 is used to compute the weight of each PI component: $t_1 = CV(Q_1) = \frac{1}{2^2} = 0.25, t_2 = CV(Q_2) = \frac{2}{2^2} = 0.5, t_3 = CV(Q_3) = \frac{1}{2^2} = 0.25$. The control value of each PI on the MUX output is computed: $CV^{'}(Q_1) = CV^{'}(Q_3) = \frac{2}{2^4} = 0.125, CV^{'}(Q_2) = \frac{6}{2^4} = 0.75$. Therefore, the weighted sum $CV(Z_1) = \sum_{i=1}^{3} t_i \times CV^{'}(Q_i) = 0.4375$. The control value of MUX is the median of the vector $CV(MUX) = median\{CV(Z_1), CV(Q_4), CV(S)\} = 0.5$.

The CV-based detection scheme uses the fact that the trigger has little effect on the outputs in order to evade functional testing, which means the control value of the trigger should be small. FlowTrojan computes CVs of the present gates in the partitioned sub-netlists and flags the gates whose CVs are smaller than the predefined threshold. The marked gates can be transformed back to the FMFB layout for locating the suspicious microvalves. The function $EstCV(T, w, M)$ is used to estimate the control values of the wires while restricting the computational overhead. M rows are randomly selected from the truth table T to approximate $CV(w)$ instead of computing the fraction from the exponentially large table.



Q_1	Q_2	Q_3	Z_1
0	0	0	0
1	0	0	0
...			
0	1	1	1
1	1	1	1

$Z_1 = (Q_1, Q_2, Q_3)$	Q_4	S	O
(0, 0, 0)	0	0	0
(0, 0 , 1)	0	0	0
...			
(1, 1, 0)	1	1	1
(1, 1, 1)	1	1	1

Fig. 4. Demonstration of computing the control values of a multiplexer. Truth tables of $Z1$ and MUX are shown.

IV. Experimental Results

We use the benchmarks in Biochip Simulator [13] to evaluate FlowTrojan in terms of effectiveness and runtime overhead. We run FlowTrojan framework on a PC with Intel Core i7-6700HQ 8-core CPU at 2.60GHz and 32GB of RAM. The overall and the breakdown runtime of FlowTrojan's parallel implementation is shown in Fig. 5. FlowTrojan is scalable since its runtime complexity has a linear relation with the # pneumatic gates on the FMFB.

A. Attack Results

We demonstrate the feasibility of the HT by inserting a trigger using a single 2-AND gate (which only needs two microvalves and three microchannels). Experimental results of baseline and parallel routines are compared in Table I. Our region-based processing achieves up to three orders test pattern compression and up to sixfold overall speedup.

B. FlowTrojan Detection Results

We evaluate the proposed control-value based HT detection scheme on the malicious FMFB benchmarks devised in Sec. III-A and summarize the results in Table II. In contrast to UCI [14], FlowTrojan's detection scheme is able to operate without false negatives (HT detection rate is 100% across

TABLE I
EXPERIMENTAL RESULTS OF FLOWTROJAN ATTACK ON VARIOUS BENCHMARKS.

Benchmark	*Area*	*# Flow Valves*	*# Pneumatic Gates*	$P_{trigger}$	R_m	*Net Extract Time (s)*	*Baseline LS Time (s)*	*Parallel LS Time (s)*	*Baseline STA Time (s)*	*Parallel STA Time (s)*	*Overall Speedup*
AquaFlux	17500	26	28	0.0647	433.23	1.092	1.356	0.018	0.733	0.362	2.205
Urbanski	19600	48	31	0.0611	409.6	1.189	2.879	0.0194	0.706	0.527	3.365
PCR1s	16100	57	37	0.0667	1250	1.378	7.269	0.0187	0.655	0.238	6.19
PCR2s	18900	77	95	0.0038	892.9	2.626	10.359	0.0269	0.984	0.746	4.894
PCR3s	20800	96	146	0.008	108.5	3.965	12.354	0.234	1.31	0.294	3.886
EA1s	30000	92	297	0.0042	375	6.722	17.348	0.122	1.023	0.236	3.517

TABLE II
EXPERIMENTAL RESULTS OF PROPOSED CONTROL-VALUE BASED HT DETECTION ON THE DEVISED MALICIOUS BENCHMARKS.

Benchmark	*AquaFlux*	*Urbanski*	*PCR1s*	*PCR2s*	*PCR3s*	*EA1s*
Detection Rate	100%	100%	100%	100%	100%	100%
False Alarm Rate	0	3.125%	0.53%	0	0	0.34%
Baseline Runtime (s)	3.5077	3.7792	4.1538	11.022	17.2205	283.8529
Parallel Runtime (s)	2.405	2.284	2.609	8.563	11.536	198.05
Speedup	1.458	1.654	1.592	1.287	1.493	1.433

all benchmarks as shown in the second row of Table II). This is due to the fact that we flag pneumatic gates with low influence on the outputs instead of completely unused gates. A false negative means a trigger gate that we do not detect In addition, our detection has small false positive rates: less than 1% gates are reported as suspicious in most cases. A false positive means an authentic gate is flagged as suspicious. FlowTrojan detection is fast and scalable due to the region-based processing. The trade-off between the detection overhead (in terms of memory consumption as well as runtime) and the detection accuracy can be leveraged by the defender via changing the number of rows M in Alg.1.

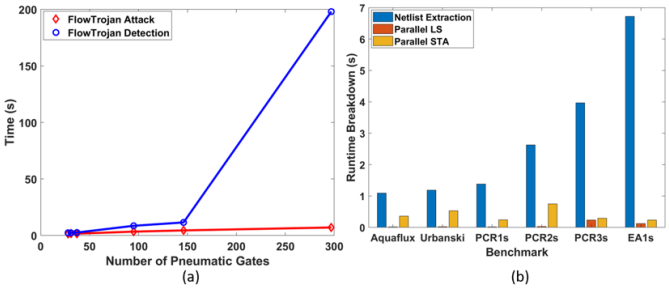


Fig. 5. (a) Overall runtime overhead of FlowTrojan attack and detection schemes with different # of pneumatic gates. (b) Runtime breakdown of parallelized FlowTrojan attack.

V. CONCLUSION

This work proposes FlowTrojan, the first systematic framework for Hardware Trojan insertion and detection on Flow-based Microfluidic Biochips. The on-chip pneumatic control circuit of the FMFB is extracted and partitioned into sub-netlists for parallel processing. FlowTrojan attack finds satisfiable trigger with minimum activation probability and places the HT on a non-critical timing path. We demonstrate that HTs as simple as a single gate can be effective to divert the control sequence of the FMFB. FlowTrojan detection presents a novel region-based control-value analysis technique as a countermeasure and evaluates the performance on the malicious benchmarks. Experimental results prove the effectiveness of FlowTrojan (incurs no false negatives and negligible false positives) and indicate that our region-based approach contributes to testing pattern compression as well as runtime reduction.

REFERENCES

[1] K. Jain, "Applications of biochips: from diagnostics to personalized medicine." *Current opinion in drug discovery & development*, vol. 7, no. 3, pp. 285–289, 2004.

[2] W. H. Minhass, P. Pop, and J. Madsen, "System-level modeling and synthesis techniques for flow-based microfluidic very large scale integration biochips," Ph.D. dissertation, Technical University of Denmark, Department of Informatics and Mathematical Modeling, 2012.

[3] Illumina, "Illumina," Jul. 2017. [Online]. Available: https://www.illumina.com/

[4] Fluidigm, "The Fluidigm corporation," Jul. 2017. [Online]. Available: http://www.fluidigm.com

[5] M. C. Eskesen, P. Pop, and S. Potluri, "Architecture synthesis for cost-constrained fault-tolerant flow-based biochips," in *Design, Automation and Test in Europe*. IEEE, 2016, pp. 618–623.

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

[7] J. Tang, M. Ibrahim, K. Chakrabarty, and R. Karri, "Toward secure and trustworthy cyberphysical microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 4, pp. 589–603, 2018.

[8] H. Chen, S. Potluri, and F. Koushanfar, "Biochipwork: Reverse engineering of microfluidic biochips," in *IEEE International Conference on Computer Design*, 2017, pp. 9–16.

[9] S. Potluri, P. Pop, and J. Madsen, "Design-for-testability of on-chip control in mvlsi biochips," *IEEE Design & Test*, vol. 36, no. 1, pp. 48–56, 2018.

[10] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security implications of cyberphysical digital microfluidic biochips," in *IEEE International Conference on Computer Design*, 2015.

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

[12] A. Waksman, M. Suozzo, and S. Sethumadhavan, "Fanci: identification of stealthy malicious logic using boolean functional analysis," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 697–708.

[13] M. F. Schmidt, "Biochip simulator," Jul. 2012. [Online]. Available: https://sites.google.com/site/biochipsimulator/

[14] M. Hicks, M. Finnicum, S. T. King, M. M. Martin, and J. M. Smith, "Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically," in *IEEE Symposium on Security and Privacy*, 2010, pp. 159–172.