OFFRAMPS: An FPGA-based Intermediary for Analysis and Modification of Additive Manufacturing Control Systems

Jason Blocklove*, Md Raz*, Prithwish Basu Roy†, Hammond Pearce‡,
Prashanth Krishnamurthy*, Farshad Khorrami*, Ramesh Karri*
*New York University, New York, USA
†New York University Abu Dhabi, Abu Dhabi, UAE
‡University of New South Wales, Sydney, Australia
jason.blocklove@nyu.edu, md.raz@nyu.edu, pb2718@nyu.edu, hammond.pearce@unsw.edu.au,
prashanth.krishnamurthy@nyu.edu, khorrami@nyu.edu, rkarri@nyu.edu

***Abstract*—Cybersecurity threats in Additive Manufacturing (AM) are an increasing concern as AM adoption continues to grow. AM is now being used for parts in the aerospace, transportation, and medical domains. Threat vectors which allow for part compromise are particularly concerning, as any failure in these domains would have life-threatening consequences. A major challenge to investigation of AM part-compromises comes from the difficulty in evaluating and benchmarking both identified threat vectors as well as methods for detecting adversarial actions. In this work, we introduce a generalized platform for systematic analysis of attacks against and defenses for 3D printers. Our "OFFRAMPS" platform is based on the open-source 3D printer control board "RAMPS." OFFRAMPS allows analysis, recording, and modification of all control signals and I/O for a 3D printer. We show the efficacy of OFFRAMPS by presenting a series of case studies based on several Trojans, including ones identified in the literature, and show that OFFRAMPS can both emulate and detect these attacks, i.e., it can both change and detect arbitrary changes to the g-code print commands.**

***Index Terms*—Additive Manufacturing, Cybersecurity**

I. INTRODUCTION

Additive Manufacturing (AM), also known as 3D printing, is the process of building up a manufactured component by repeatedly adding material in specific quantities and locations. Subtractive manufacturing, instead, removes raw material until a final part is left. AM is performed by designing a part using a computer-aided design (CAD) tool such as Autodesk Fusion or Solidworks, then sending the part to a "slicer" program to separate the part into component layers and, based on the target 3D printer, exports `g-code` which encodes the print head movements used to create the part (see Figure 1).

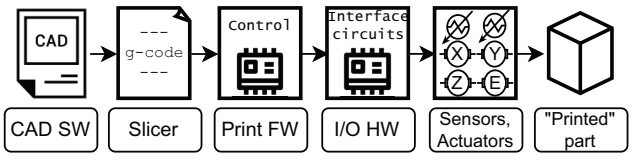


Fig. 1: Simplified Additive Manufacturing (3D printing) process. Malicious interference can occur at any step.

Declining prices and increasing quality of modern 3D printers is making them a common tool in hobbyist and professional spaces. In one form of 3D printing, fused filament fabrication (FFF)—also known as fused deposition modeling (FDM) printing—molten plastic is extruded in layers to build the part. Other forms of 3D printing include stereolithography (SLA) which is the process of shining a UV light in specific shapes per-layer through UV-curing resin to build a part up and selective laser sintering (SLS) which scans a laser a bed of reactive powder to fuse particles into solid layers.

The ubiquity and ease of use of 3D printing has grown its adoption within a variety of safety-critical industry sectors, including in the biomedical domain [1], in robotic components [2], construction [3], aerospace [4], the automotive sector [5], and others. The varying needs of each industry has prompted both innovation and proliferation of interrelated tools and products, from hobbyist to professional and commercial domains. With this growth in usage comes the growth in *threats* against additive manufacturing systems [6], [7].

The threat landscape in the 3D printing domain encompasses the hardware, software, and supply chain, and can impact every part of the process in Figure 1. Hardware attacks and defenses, which both affect and utilize the integrated circuitry and printed circuit boards used for AM systems, allow for modifications which can internally affect the quality, appearance, or structural integrity of the desired model. Software-based modifications aim to affect the generated geometry of the design, usually resulting in the export of compromised `g-code`. Finally, supply-chain attacks affecting 3D printers may include compromised slicing programs, defective components, or inherently flawed filaments and raw materials. Unfortunately, the cybersecurity threat landscape in 3D printing remains under-explored, in part due to the complexity involved in studying attacks which impact the real-world physical aspects of additive manufacturing.

Contributions: To address these shortcomings, we introduce OFFRAMPS, an FPGA-based integrated test-bed which facilitates the analysis and modification of key signals required

2158-3927/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN58291.2024.00057

to drive the functional components of modern fused filament fabrication (FFF) 3D printers. It leverages the RAMPS open-source 3D printer control platform, which is representative of commercial offerings while allowing device modification.

OFFRAMPS is the **first platform to support in-hardware analysis of both attacks and defenses**. OFFRAMPS enables in-depth exploration of novel printer attack strategies, useful for the identification of previously unexamined security blindspots, and **we provide a suite of representative Trojans** for this purpose. The analysis capabilities allow for expanding the defensive state of the art, and **we present one defense capable of identifying major Trojans from the literature**. OFFRAMPS is open-source, available here: [8].

II. Prior Work

A. 3D Printing Security Threats

Over the last decade, improvement in the quality of printing materials and printers abetted printing intricate components with ease. AM found use in diverse fields like aerospace engineering, construction engineering, bio-medical engineering, etc. With its newly gained popularity, AM has become the target of attackers with malicious intents of sabotage and espionage [9], [10]. In sabotage-motivated attacks, the attackers either aim to compromise the overall printing process or focus on compromising the quality of the printed product such that its longevity is significantly reduced, causing irreparable damage to the victim company's goodwill. In their work "dr0wned" [11], the authors demonstrated an end-to-end cyber-physical attack that was initiated by introducing malware in the victim's machine. This malware finds design files in the system, identifies spots that are vulnerable to stress, and inserts sub-millimeter holes in them. As a demonstration, they compromised the design of a quadcopter drone's propeller, which caused the drone to crash mid-flight. In another work [12], the authors have modified the Marlin firmware [13] to introduce changes ranging from minor modifications of the executing `g-code` to the execution of alternative `g-code`, leading to printing malformed or totally incorrect objects. The infrequently updated firmware's bootloader becomes an attractive target for stealthy Trojan insertion in the recent study Flaw3D [14]. Authors injected Trojans into the bootloader's flash memory, undermining the quantity of extruded material and compromising the print's quality.

In a different class of attack, the attacker aims to exfiltrate information about Intellectual Properties (IPs) being printed. A 3D printer has mechanical components, such as motors and heating elements that require specific signals to actuate. A firmware like Marlin, that resides in the controller of the printer, is responsible for parsing the `g-code` and generating these signals. The attackers have prior information about the type of motors and can analyze these signals and the power consumed [15] by the motors to gain insight into the linear movements in different axes, thus partially recovering the executing `g-code`. In the papers [16]–[18], the authors have exploited a similar correlation between the executing `g-code` and the sound emanated by the motors and the actuators to partially reverse engineer the IP. Attackers can also leverage optical side-channels to recreate `g-code` for a design being printed [19]. These findings highlight risks and security concerns in the 3D printing process.

B. 3D Printing Threat Detection Techniques

A significant number of the currently available threat detection methods are based on side-channel analysis. Side-channels are passive mediums that leak information about the printing process due to the operation of various physical components of the printing device. In the case of a 3D printer, the sound emitted by the rotation of the motors, the change in the magnetic field causing the rotation of motors, the power consumed by the motors, the temperature of the hotend, and video of the overall printing process can be considered as the source of acoustic, magnetic, power, thermal [20], and optical side-channel leakage, respectively. The assumption for this type of detection is that a good print will have a different side-channel leakage profile than a compromised print. In [17], [18], acoustic emission of the motors in a secure setup is used as the golden model and is compared to the acoustic signature of future prints to detect anomalies due to `g-code` manipulation. However, acoustic side-channels have limited accuracy in determining small and rapid moves, which reduces their efficacy. In [21] the power profiles of the four stepper motors for the X, Y, and Z axes plus the extruder are used as a golden reference model against the power profile of future prints. Power side-channel-based analysis, although effective, requires forty repetitions of each print to nullify the noise and syncing issues, making it expensive and non-scalable. In [22] video feeds (optical side-channel) of numerous printing processes are used to train an ML model and this model is used to classify the current print as actual or compromised. The optical side-channels allow visual reconstruction of the object and thus can detect layer-by-layer anomalies of the printing process as long as it has been trained to capture such errors. While quite effective, this method requires significant hardware overhead with very specific requirements for camera placement and image capture. The accuracy of the results is also dependent on the specific features being printed and the machine learning methodology used to detect the anomalies.

AM security can be thought of as a subset of cyber-physical systems security. For surveys of this area, see [23], [24].

III. OFFRAMPS Board Overview

This section details the design of the OFFRAMPS board (Figure 2), a printed circuit board (PCB) which uses a field programmable gate array (FPGA) as a machine-in-the-middle (MITM) in a popular open-source 3D printer control system. The board is designed to interface with 1) a Digilent Cmod-A7 FPGA development board, 2) an Arduino Mega running Marlin firmware, and 3) a RAMPS 1.4 3D printer control board. We add jumpers and logic level shifters to allow the signals to be rerouted as necessary for different experiments. In Figure 1, OFFRAMPS is located between the Controller (Print Firmware) and the Interface circuits (I/O Hardware),

which enables it to detect and interfere with all signals at digital level voltages (lower than 5V).

A. Open Source 3D Printing

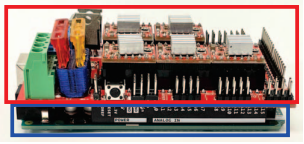
In the FFF printing space there are numerous companies which make consumer 3D printers such as Creality, Bambu, Prusa, and Ultimaker; however, open-source printers have been a mainstay in 3D printing and many manufacturers (such as Prusa) continue this tradition by open-sourcing their own designs. Many of the open-source design components for printer hardware, mechanical and electrical, fall under the RepRap project [25]. The low-cost RepRap Arduino Mega Pololu Shield (RAMPS) printer control board is one such component. In this work we use version 1.4 of this board, which is designed to interface with an Arduino Mega as a hardware-on-top (HAT) device. The Arduino must run firmware which can interface with a host computer, and for this purpose Marlin is often used as it is another fully open-source piece of the system. Most designs of RepRap FFF printers can make use of this stack of control boards, making it a prime candidate for evaluation and representative function of other boards.

The broader RepRap project is an endeavor aimed at creating self-replicating 3D printers that are open-source, and has been a major driver for the increased access of additive manufacturing. This increased access has also come at a cost, however—clones of these boards are sold by an extensive list of sellers through various online outlets. These clones, which may come with different ICs and functionalities, are of varying quality and provenance. Unaware end-users may be impacted negatively by faulty control boards, which may be caused by inferior counterfeit components with undersirable changes to the originals.

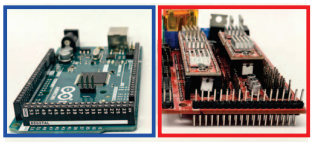
B. OFFRAMPS: Design Motivation

A common threat model for AM comes from malicious third parties modifying the controller PCBs [12], [14], [26] to feature firmware or hardware Trojans. Other attacks come from the software space, where CAD programs may be compromised [11]. It is thus desirable to support emulation of hardware, firmware, and software Trojans in a single platform, as well as provide capabilities to analyze signals passing from the firmware on the control CPU to the driving components.

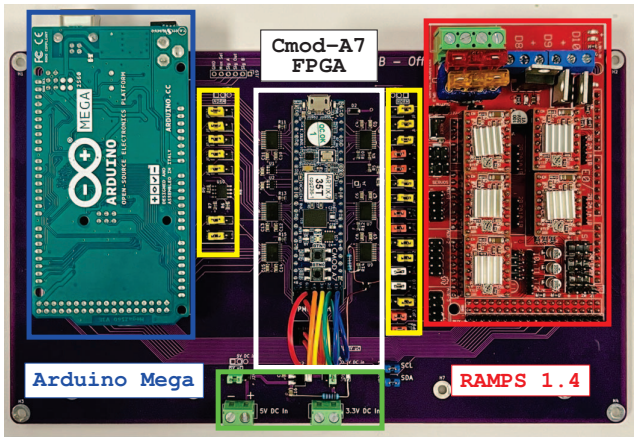
RAMPS 1.4 is an open-source control PCB for 3D printers, designed by RepRap, made to interface with an Arduino Mega by plugging into the top as a HAT device (Figures 2a, 2b). We make use of this setup for the OFFRAMPS as a representation of control boards used throughout the industry, as all FFF printers will ultimately require the same set of signals. OFFRAMPS allows the Arduino/RAMPS stack to function normally when in a standard configuration, but enables the FPGA to analyze and modify the signals passing between the original boards with simple jumper changes (the yellow highlighted banks in Figure 2). These can redirect signals from normal operation to MITM operation, changing source and destination as needed. Meanwhile, the FPGA is used as a reconfigurable platform to enable adversarial and defensive



(a) The standard stack of Arduino Mega (blue box) with RAMPS 1.4 (red box) as a HAT.

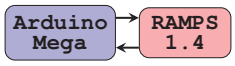


(b) The Arduino Mega (blue box) and RAMPS (red box) separated to be placed on the OFFRAMPS.



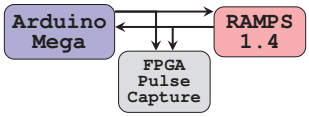
(c) OFFRAMPS board fully populated. The Arduino Mega (blue box, far left) is flipped upside down to plug into the headers and interfaces with a host computer over USB. The RAMPS 1.4 (red box, far right) receives control signals from the Arduino and sends back certain feedback information. The jumpers (two yellow boxes) determine whether each signal will be passed through the Digilent Cmod-A7 (white box, center) or come directly from the intended source. The power circuitry (green box) allows for Arduino and FPGA power to be derived from several sources as needed.

Fig. 2: The stack of Arduino Mega and RAMPS board separated and put in place on the OFFRAMPS board.



(a) Unmodified signal chain.

(b) FPGA for signal modification.



(c) FPGA for signal recording.

Fig. 3: Different signal path options on the OFFRAMPS.

techniques to be tested, analyzed, and verified in hardware. Figure 3 shows the three signal path configurations possible with the added MITM FPGA.

C. Primary OFFRAMPS Printed Circuit Board Design

OFFRAMPS has the following major components:

1) Digilent Cmod-A7: The Cmod-A7 FPGA development board [27] has a Xilinx Artix-7 35T FPGA, LEDs and two push buttons. It is used as the MITM deployment platform between the Arduino and the RAMPS boards.

To intercept all 3D printer control signals between the Arduino and RAMPS boards, all GPIO headers were used, save one pin which was broken out to assist in debugging or external signal insertion. Additionally, the Artix-7 has an onboard analog-to-digital converter (ADC) which can be used alongside an off-chip digital-to-analog converter (DAC) and opamp to read and modify analog signals like the voltage level through the thermistors on the 3D printer.

2) Arduino Mega: The Arduino Mega [28] in this system is configured to run the popular 3D-printer open-source Marlin firmware [13]. It receives `g-code` generated by a slicer such as Cura or Slic3r and sends signals to drive components on the connected 3D printer. These include (1) step and direction (`*_STEP` and `*_DIR`) signals for each of the stepper motors on the printer—`X,Y,Z` axis movement and filament extrusion; (2) fan speed control for the part cooling fan, (3) heating element control for both the heated bed and hotend, and (4) UART signals to interact with a connected display/control board. Marlin implements some safety features such as checks for thermal runaway.

3) RAMPS 1.4: The RAMPS board controls the actuator functions of the printer directly with stepper motor drivers, fan control circuity, and heating element circuitry—all driven by the aforementioned signals sent from the Arduino. In turn this board sends back signals for the endstops of the axes and the thermistors for both the heated bed and hotend of the printer. The display/control board also connects through the RAMPS using UART to allow a user to interact with the printer directly without having a connected host computer.

RAMPS has onboard configuration jumpers to micro step the stepper motor drivers, but otherwise all control for the onboard devices is managed by the Arduino. The stepper motor drivers are also modular, we opted to use the default A4988 drivers shipped with RAMPS. These are inexpensive and popular, representative of components common to commercial 3D printers.

4) Logic level shifting: Both the Arduino and RAMPS 1.4 boards require a $5V$ logic level for their signals. The Cmod-A7, however, can only support voltages up to $3.3V$. To accommodate this necessary level shift for the FPGA I/O bidirectional logic level shifters and enhanced field effect transistors (FETs) are integrated on board to allow the $5V$ logic to be shifted to a usable $3.3V$ for the FPGA, and then re-converted back to $5V$ for the Arduino and RAMPS boards.

5) Board Power: The RAMPS board receives its power from the 3D printer's $24V$ power supply. The Arduino and FPGA need a separate power source, which can come from either the USB ports on each board or can be externally and separately provided. By separating the power systems and allowing their source to be selected, the OFFRAMPS can function without a host computer—a common deployment strategy for many 3D printers.

D. Test Environment

The test environment used for validating and experimenting with the OFFRAMPS consisted of a modified Prusa i3 MK3S+ 3D printer [29] and a host computer running Ubuntu 22.04. The Prusa i3 MK3S+ is an incredibly popular hobbyist-grade 3D printer compliant with the RepRap project. A small modification had to be made to the Prusa to add mechanical endstops as this is a more common method of homing than the more advanced sensorless homing that the Prusa control board supports natively. The RAMPS required small modifications to ensure compatibility with a $24V$ power supply—this was done according to the instructions for this conversion from RepRap. All prints were sliced with Ultimaker Cura and `g-code` control was done with Repetier Host.

IV. OFFRAMPS FOR TROJAN INSERTION

A. Objective and Relevance

The OFFRAMPS presents several advantages over conventional firmware and `g-code` based Trojans: the FPGA leveraged as the MITM allows for both fine-grained logic and timing level modification of all control signals at the resolution of the FPGA clock speed ($100MHz$). We present several Trojans which take advantage of these benefits and the direct access to fundamental control signals.

B. Methodology and Design Considerations

The OFFRAMPS was evaluated for its ability to implement Trojans mimicking common 3D printer issues as well as Trojans which maliciously compromise the printer hardware itself, as outlined in Table I. These Trojans are designed to modify the part, deny access to certain printer elements, or damage the part or the printer itself.

A framework for the insertion of Trojans was created using VHDL. This framework allows for both standard operation of the printer via signal bypass or malicious operation through the Trojan module. Several sub-modules were created to control the insertion of Trojans as follows:

Pulse Generation Module handles the generation of pulses for the stepper motor drivers, and allows for the customization of both frequency and pulse width, along with input parameters for micro stepping determined by the printer configuration.

Edge Detection Module implements an edge detector to identify events such as print head movements or extrusions via observation of the `STEP` and `DIR` stepper motor driver signals from the Arduino or endstop actuation for homing detection.

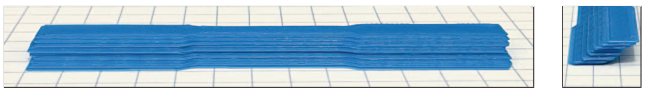
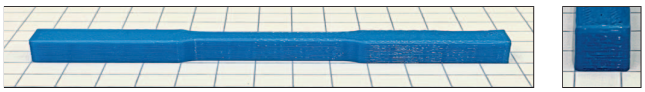
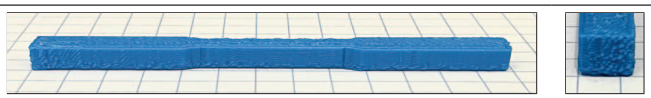
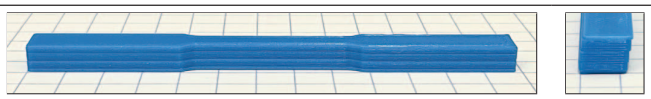
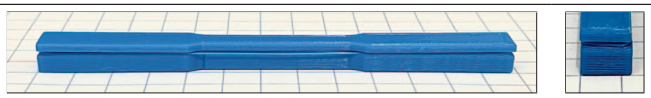
Homing Detection Module is a state machine which tracks actuation of the endstops in a defined order to determine when the print head has homed. This is the first action taken at the start of print and can determine when to activate Trojans.

Trojan Control Module has logic to enable or disable each of the Trojans (Table I), along with control units for each Trojan. The modified signals produced by this module are multiplexed with the original control signals so the Trojans can be dynamically activated or deactivated.

C. Results and Analysis of Effectiveness

Golden Print (Table I: T0), was created by setting the FPGA in 'bypass' mode. The control signals from the Arduino pass through, without modification, to the RAMPS board. The

TABLE I: Trojans evaluated using OFFRAMPS. Part modification (PM) Trojans modify the produced part, Denial of Service (DoS) Trojans disable access to a function of the printer, and Destructive (D) Trojans damage a component of the printer itself. Trojans T1 - T5 allow for completed prints which are shown placed on graph paper with line spacing of $\frac{1}{4}$ inch.

Trojan	Type	Scenario	Effect	Printed Part
T0	None	None	Golden print	
T1	PM	Loose Belt	Randomly changes steps from X or Y axis during print	
T2	PM	Incorrect Slicing	Constant over / under extrusion per print	
T3	PM	Incorrect Slicing	Increases or decreases filament retraction during Y steps	
T4	PM	Z-Wobble	Small Shift along X and Y axis on random Z layer increments	
T5	PM	Incorrect Slicing	Layer delamination via Z-layer shift	

Trojan	Type	Scenario	Effect	Trojan	Type	Scenario	Effect
T6	DoS	Hardware Failure	Denial of service via disabling D8/D10 heating element power	**T8**	DoS	Hardware Failure	Arbitrarily deactivating stepper motors via EN signals
T7	D	Hardware Failure	Forcing thermal runaway and permanently enabling heating elements	**T9**	PM	Hardware Failure	Arbitrarily reducing part fan speed mid-print

printed part shows no deformation, structural compromise, or dimensional inaccuracy.

Trojan T1 implements an arbitrary shift along the X and Y axes every ten seconds. This print shows extensive shift along both axes, affecting dimensional accuracy and part practicality. The FPGA on the OFFRAMPS allows to injection stepper motor pulses in between the original control pulses, causing longer travel motions of the print head. This effect is used by the Trojan to add extra steps without adding extra print time.

Trojan T2 modifies the amount of material extruded during printing, similar to a 'flow' parameter used when slicing a STL model. The Trojaned part, was printed while masking half of extruder stepper motor pulses sent to the RAMPS board, reducing the flow and amount of material extruded by 50%. This implements reduction Trojans from Flaw3D.

Trojan T3 mimics a type of problem which can occur from improper settings when slicing a model into `g-code`. Retraction refers to the amount of filament that is pulled back during certain movements. By affecting extruder steps during some movements we can cause over or under extrusion in a way that could appear to a user as if part settings were incorrect when sliced. This Trojan is shown with over extrusion in Table I: T3.

Trojan T4 implements a Z-wobble Trojan. Z-wobble is common build issue with 3D printers, where the frame holding the Z-axis is not rigid; thus, the print head can shift during printing. Trojan T4 emulates this error by adding steps on one axis during printing causing layer shifts.

Trojan T5 causes an arbitrarily sized shift on the Z-axis, causing poor layer adhesion or, in severe cases, layer delamination. This mimics improper slicing settings if the layer spacing is modified throughout the print, and poor hardware setup if a shift is done at the start of print, causing the part to fail to adhere to build plate.

Trojan T6: 3D printers have multiple heating elements including a heated bed, to assist with part adhesion to the build plate, and a hotend for extruding molten material. Should these heating elements be unable to reach the necessary temperature, the 3D printer may be unable to begin a print or if the temperature should suddenly drop mid-print the part quality could be severely negatively impacted. This can also cause the firmware to throw a thermal runaway error and halt all operation. This Trojan was observed to successfully turn off the PID controlled MOSFETs employed in providing power to the heating elements, causing the Marlin firmware to enter an error state and end the print prematurely.

Trojan T7: The inverse of Trojan T6, Trojan T7 forces the heated elements to continue heating regardless of the firmware temperature control. By implementing this Trojan in hardware we are not only able to force overheating, but also able to ignore the firmware's thermal runaway panic and continue heating the elements. This is a purely destructive Trojan which can not only severely degrade part quality but can damage or destroy components of the printer itself. This Trojan was observed to successfully enable power MOSFETs for the heating elements permanently, bypassing all thermal control and fail-safes from the firmware, heating the element past the working specification. Furthermore, since the MOSFETs are fully turned on at a 100% duty cycle, the temperature of the hot-end was observed to rise extremely fast, passing the intended temperature within a few seconds of activation.

Trojan T8: Each stepper motor driver has an input signal `*_EN` which determines if the motor is engaged and able to be moved. By actuating this signal throughout the print we can disable stepper motor movements strategically to fail a print. This emulates issues with stepper motor drivers or the motors as they can be made to arbitrarily cease functioning.

Trojan T9 affects the part-cooling fan on the printer and causes either over- or under-cooling during printing. Depending on the print material and the stage of printing, the cooling fan runs at various speeds, determined by the `g-code`. Control signals for this fan are passed through the FPGA for full control. Print quality can be degraded by either over- or under-cooling. It can fail if excessively cooled at the first layer causing it to pull off the build plate.

Trojans **T1 - T5** produce prints with visible or structural anomalies and were printed using the OFFRAMPS with a Prusa i3 MK3S+ printer, shown in Table I. Trojans **T6 - T9** affect aspects of the 3D printer which either prevent printing a part or would be destructive to the printer hardware; they were validated on our printer but did not produce parts we could show.

D. Key Takeaways

Low-level access to all control signals afforded by the OFFRAMPS board is a powerful capability to implement many different types of Trojans (the listed Trojans are not exhaustive of all possibilities). The FPGA also allows for the insertion of arbitrary combinations of Trojans along with their triggers. Accessibility and granular control for interception and modification of signals makes the OFFRAMPS board a powerful tool in inserting and testing 3D printer Trojans.

V. OFFRAMPS FOR PRINT MONITORING

In addition to signal modification in hardware, OFFRAMPS also allows for signal extraction, enabling tests for print verification and Trojan detection. In a sense, the FPGA can act as a rudimentary 'digital logic analyzer' for the control signals passing between the Arduino and RAMPS boards.

A. Objective and Relevance

Most 3D printer Trojans rely on maliciously altering a design at a stage prior to the signal transfer between the firmware and the control circuitry. Flaw3D [14], for example, modifies a design using a malicious bootloader to edit the `g-code` as it is sent to the firmware, resulting in prints with reduced structural integrity. Dr0wned [11] creates modified models prior to their slicing, which ultimately will result in improper `g-code` being sent to the firmware. By intercepting the signals after they are decoded by the firmware, we are able to record the real movements of the stepper motors and verify them against a known-good model.

This would be useful, for instance, in safety-critical parts in industrial manufacturing. Typically, randomly-selected parts may undergo destructive testing to validate performance. However, if a small subset of parts are intentionally and maliciously defective, then random testing may not identify them. OFFRAMPS addresses this through continuous monitoring of prints—all parts will be checked, not just a random subset. Further, parts are checked during production, meaning that large malicious divergences can be detected and aborted early to save machine time and material cost.

B. Methodology and Design Considerations

To monitor the 3D printer in real-time, the FPGA is programmed to both record and export the relevant control information. We designed a module to track the number of steps sent to each stepper driver after homing. This translates into absolute positions within the build volume of the printer and a definite amount of extruded material. The detection algorithm works as follows: (1) a "golden" model is captured by verifying a set of `g-code`. This is done by first performing a print then completing extensive verification through both non-destructive and destructive testing to ensure the part meets the physical demands and constraints of the product. (2) Once assured, the pulse profile can be used as a point of comparison for future prints.

Axis Tracking: This module analyzes the stepper motor control signals, `STEP` and `DIR`, for each of the axes and the extruder to determine their positions. This consists of a set of rising edge detectors and counters, which increment for each `STEP` rising edge when `DIR` dictated that the motors were moving in the positive direction and decrement when they moved negatively. By correlating the steps to the movement along each axis—information available with printer setups—we are able to track the absolute position of the print head within the build volume and the amount of filament extruded.

To ensure the step counting always began from 0 in a known location, we leveraged the homing detection module created for Trojan insertion. When the printer is homed at the beginning of each print, the step counts and UART transaction counter are initialized. As the number of steps to home is determined by the arbitrary position of the print head at the start of the print, capturing this data was deemed unnecessary for evaluating the Trojans.

UART: For accurate pulse counts between all tests, the counter to determine the frequency of the UART transactions starts after the print head is homed and the first `STEP` edge is found. This synchronization significantly increased accuracy

over initial tests which did not wait for the first step before beginning the counter. With the analysis started, the UART control unit sends a 16-byte transaction containing step counts for all of the motors each 0.1 seconds.

Overhead: A detection method which significantly impacts the speed or quality of a print is counterproductive to the goal of verifying that a print is of good quality. We estimated that the maximum propagation delay of any signal captured in the detection design is $12.923ns$ on the `Y_DIR` signal. The ordinary signals between the Arduino and RAMPS boards were measured to have maximum frequencies less than $20kHz$ with a minimum pulse width of $1\mu s$. Given these parameters, a $12.923ns$ delay is negligible and we found no effect on print quality while running our detection hardware.

C. Detecting Trojan-Induced Edits

Our Trojan detection strategy compares the captured pulse counts of a given print against a known-good capture, either derived from a print that was captured and then separately validated or from a simulation of the firmware. In a print without Trojan manipulation the Arduino will always send the same quantity of `STEP` and `DIR` control signals with approximately the same timing as the known-good print, but where the print commands have been interfered with these counts will change. Mismatches outside of a reasonable margin of error suggest this kind of interference, and in our study, this translates to the presence of a Trojan. Here, the margin of error is due to the challenge of synchronizing the step counting with the UART transactions. Additive manufacturing systems are asynchronous, so an instruction can take a slightly different amount of time when executed multiple times or across multiple prints. This variation, referred to as "time noise" [30], means that some drift in the step counts will occur over the course of even known-good test prints. This drift was, however, always less than a 5% difference in our testing, so for our evaluations we used this 5% margin of error against our ideal profile. This 5% margin of error can be made significantly smaller with a faster communication protocol, as fewer steps possible per transaction would lower the potential drift in counts. The concern of having too large of an error margin is also mitigated with a final check with a 0% margin of error, ensuring that the correct number of steps was counted on each axis at the conclusion of the print.

A Python script compares a newly captured print against a "golden" model. Should a mismatch outside of the 5% margin of error occur the transaction number and mismatching values are printed. At the termination of the capture file the script then gives a report stating the total number of mismatches, the greatest error found, and the total number of captured transactions—based on these a determination of whether or not a Trojan is suspected is made. This analysis can also be done in real-time while printing, enabling a user to halt a print as soon as a Trojan is suspected.

TABLE II: Flaw3D Trojans. Modification value for reduction is a factor by which extrusion amount is reduced. For relocation it is the number of movements before filament is relocated.

Test Case	Type	Modification Value	Detected
1	Reduction	0.5	✓
2	Reduction	0.85	✓
3	Reduction	0.9	✓
4	Reduction	0.98	✓
5	Relocation	5	✓
6	Relocation	10	✓
7	Relocation	20	✓
8	Relocation	100	✓

D. Analysis of Flaw3D Trojans

To evaluate the Trojan detection methodology we emulated Trojans from Flaw3D [14]. In the original work a modified bootloader was used to change `g-code` on the fly to implement one of two types of Trojan: reduction of extruded filament or occasional relocation of filament during the print. We recreate these Trojans using a Python script which modifies given `g-code` in the same way the malicious bootloader does. This yielded eight Trojans from two categories, each with varying levels of severity as enumerated in Table II.

Each of these Trojans was printed and their pulse profiles were captured using the OFFRAMPS. Those captures were then compared against the known-good reference and the detection program was able to identify all of the Trojans. A selection of the captures and tool output is given in Figure 4 showing mismatches outside of the margin of error and the detection tool output identifying them. Here, the Trojan used was Test Case 7 (Table II), which is not a stealthy Trojan.

The stealthiest Trojans tested are Test cases 4 and 8 which reduce extrusion by only 2% and relocate material every 100 moves, respectively. In both, the Trojan is minimal enough that structural integrity was not noticeably impacted. However, the detection strategy was still able to identify their presence.

We did not use this detection method to evaluate our own Trojans, as both the attacks and defense would be co-located in the same FPGA and we do not believe this would demonstrate any meaningful capabilities.

VI. Discussion

Evaluation: The OFFRAMPS was successfully used to both insert and detect 3D printer Trojans and, with the reconfigurability of the MITM FPGA, could implement more novel Trojans, requiring fine-grained manipulation and analysis of the firmware-produced control signals. This platform provides a basis for considerable future experimentation, with expansion of both the kinds of attacks which may be undertaken as well as new golden-free methods for detection and even reverse-engineering printed parts from their control signals.

Given the increasing adoption of Additive Manufacturing for safety-critical components and commercial applications, facilitating the analysis of security vulnerabilities in printers and real-time validation of part printing enables designs to be produced safely by providing methods to detect interference.

Limitations: While OFFRAMPS was able to detect the tested firmware Trojans from literature, the study still has some design limitations. Firstly, the platform is limited in its ability to relay high-speed information (i.e. a high-frequency data capture) to a host PC due to a lack of circuitry to support a high speed communication interface such as Ethernet or USB, preventing complex analysis strategies. Though it can *emulate* them, OFFRAMPS is currently unable to *detect* any Trojans which affect the heating elements, whether implemented in firmware or hardware. In addition, though the platform is designed with power isolation between the major components and can also support undervolting and brown-out attacks, this study did not explore this area, nor is there suitable circuitry for detecting such an attack. Many 3D printers are intended to be run while not actively connected to a host computer, which the OFFRAMPS cannot currently support for its Trojan detection functionality.

Related platforms: Other methods of attacks and defenses exist but are predominantly based on lossy side-channels such as acoustic, power, electromagnetic emission, or optical analysis. The OFFRAMPS, by connecting directly to control signals, is uniquely able to modify or analyze prints with no loss of data. Including support for some of these other side-channel techniques is being considered for future revisions, but we have found no examples of other hardware platforms which can be used for 3D printer attacking or modifying in the same manner as the OFFRAMPS.

Responsible disclosure: As all studied attacks required modification to the underlying components, no responsible disclosure is necessary for this work.

VII. CONCLUSION

In this work we present OFFRAMPS, a new hardware tool for emulation, evaluation, and detection of 3D printer Trojans. By using an FPGA in a machine-in-the-middle configuration we are able to dynamically modify 3D printer control signals post-firmware as well as detect Trojans implemented at or before the firmware level. This enables investigation of both attack and defense scenarios, a task otherwise complicated by the relationship between digital and real-world components.

Using OFFRAMPS we are able to emulate existing Trojans from the literature in hardware, as well as implement new Trojans which cannot easily be done in firmware. In total we implemented 9 such attacks from simple denial-of-service to subtle part modifications and thermal runaway, the largest suite ever supported by a single platform. On the defensive side, we present a simple yet effective strategy which can be implemented in OFFRAMPS to count the number of pulses over a series of time windows and compare this against a golden series of data (which can come from simulation), this strategy could detect all Trojans from the literature.

Index,	X,	Y,	Z,	E
5113,	6060,	8266,	960,	52843
5114,	6304,	8095,	960,	52856
5115,	7218,	8285,	960,	52856
5116,	8166,	8483,	960,	52856
5117,	8671,	8620,	960,	52859
5118,	8384,	8733,	960,	52875

(a) Selection of transactions from the golden reference.

Index,	X,	Y,	Z,	E
5113,	6027,	8499,	960,	52832
5114,	6113,	8213,	960,	52846
5115,	6489,	8133,	960,	52856
5116,	7437,	8331,	960,	52856
5117,	8384,	8528,	960,	52856
5118,	8601,	8644,	960,	52863

(b) Selection of transactions from Flaw3D Trojan print.

```
...
Index: 5115, Column: X, Values: 7218, 6489
Index: 5116, Column: X, Values: 8166, 7437
...
Largest percent difference found: 93.19%
Number of transactions compared: 12416
Number of mismatches: 952
Trojan likely!
```

(c) Selected output of the Trojan detection tool identifying a mismatch of transactions on the X axis with index 5115 and 5116. These values fall outside of the 5% margin of error, so the tool reports a Trojan is likely alongside other metadata.

Fig. 4: Detection of an emulated Flaw3D Trojan which relocates material every 20 movements.

ACKNOWLEDGEMENTS

This work was supported in part by DoE Kansas City. Honeywell Federal Manufacturing & Technologies, LLC operates the Kansas City National Security Campus for the United States Department of Energy / National Nuclear Security Administration under Contract Number DE-NA0002839.

REFERENCES

[1] J. K. Placone and A. J. Engler, "Recent Advances in Extrusion-Based 3D Printing for Biomedical Applications," *Advanced Healthcare Materials*, vol. 7, no. 8, p. 1701161, 2018.

[2] C. Tawk, H. Zhou, E. Sariyildiz, M. in het Panhuis, G. M. Spinks, and G. Alici, "Design, Modeling, and Control of a 3D Printed Monolithic Soft Robotic Finger With Embedded Pneumatic Sensing Chambers," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 876–887, Apr. 2021.

[3] O. Lakhal, T. Chettibi, A. Belarouci, G. Dherbomez, and R. Merzouki, "Robotized Additive Manufacturing of Funicular Architectural Geometries Based on Building Materials," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2387–2397, Oct. 2020.

[4] B. Blakey-Milner, P. Gradl, G. Snedden, M. Brooks, J. Pitot, E. Lopez, M. Leary, F. Berto, and A. du Plessis, "Metal additive manufacturing in aerospace: A review," *Materials & Design*, vol. 209, p. 110008, Nov. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127521005633

[5] J. C. Vasco, "Chapter 16 - Additive manufacturing for the automotive industry," in *Additive Manufacturing*, ser. Handbooks in Advanced Manufacturing, J. Pou, A. Riveiro, and J. P. Davim, Eds. Elsevier, Jan. 2021, pp. 505–530. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128184110000100

[6] S. E. Zeltmann, N. Gupta, N. G. Tsoutsos, M. Maniatakos, J. Rajendran, and R. Karri, "Manufacturing and Security Challenges in 3D Printing," *JOM*, vol. 68, no. 7, pp. 1872–1881, Jul. 2016. [Online]. Available: https://doi.org/10.1007/s11837-016-1937-7

[7] P. Mahesh, A. Tiwari, C. Jin, P. R. Kumar, A. L. N. Reddy, S. T. S. Bukkapatanam, N. Gupta, and R. Karri, "A Survey of Cybersecurity of Digital Manufacturing," *Proceedings of the IEEE*, pp. 1–22, 2020.

[8] Anonymous, "Offramps repository." [Online]. Available: https://doi.org/10.5281/zenodo.10279360

[9] N. Gupta, A. Tiwari, S. T. S. Bukkapatnam, and R. Karri, "Additive Manufacturing Cyber-Physical System: Supply Chain Cybersecurity and Risks," *IEEE Access*, vol. 8, pp. 47 322–47 333, 2020.

[10] M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici, "Security of additive manufacturing: Attack taxonomy and survey," *Additive Manufacturing*, vol. 21, pp. 431–457, May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221486041730502X

[11] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici, "dr0wned – {Cyber-Physical} Attack with Additive Manufacturing," 2017. [Online]. Available: https://www.usenix.org/conference/woot17/workshop-program/presentation/belikovetsky
[12] S. B. Moore, W. B. Glisson, and M. Yampolskiy, "Implications of malicious 3d printer firmware." Proceedings of the 50th Hawaii International Conference on System Sciences, 2017.
[13] MarlinFirmware, "Marlin Firmware," Dec. 2023. [Online]. Available: https://marlinfw.org/
[14] H. Pearce, K. Yanamandra, N. Gupta, and R. Karri, "Flaw3d: A trojan-based cyber attack on the physical outcomes of additive manufacturing," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5361–5370, 2022.
[15] J. Gatlin, S. Belikovetsky, Y. Elovici, A. Skjellum, J. Lubell, P. Witherell, and M. Yampolskiy, "Encryption is futile: Reconstructing 3d-printed models using the power side-channel," in *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*, 2021, pp. 135–147.
[16] M. A. Al Faruque, S. R. Chhetri, A. Canedo, and J. Wan, "Acoustic side-channel attacks on additive manufacturing systems," in *2016 ACM/IEEE 7th international conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2016, pp. 1–10.
[17] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, "Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2016, pp. 1–8.
[18] S. Belikovetsky, Y. A. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici, "Digital audio signature for 3d printing integrity," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1127–1141, 2018.
[19] S. Liang, S. Zonouz, and R. Beyah, "Hiding my real self! protecting intellectual property in additive manufacturing systems against optical side-channel attacks," in *Proceedings 2022 Network and Distributed System Security Symposium. Internet Society*, 2022.
[20] M. A. Al Faruque, S. R. Chhetri, A. Canedo, and J. Wan, "Forensics of thermal side-channel in additive manufacturing systems," *University of California, Irvine*, vol. 12, no. 13, p. 176, 2016.
[21] J. Gatlin, S. Belikovetsky, S. B. Moore, Y. Solewicz, Y. Elovici, and M. Yampolskiy, "Detecting sabotage attacks in additive manufacturing using actuator power signatures," *IEEE Access*, vol. 7, pp. 133 421–133 432, 2019.
[22] M. Wu, Z. Song, and Y. B. Moon, "Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1111–1123, Mar. 2019. [Online]. Available: https://doi.org/10.1007/s10845-017-1315-5
[23] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," *Annual Reviews in Control*, vol. 47, pp. 394–411, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578819300185
[24] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-Physical Systems Security—A Survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017, conference Name: IEEE Internet of Things Journal. [Online]. Available: https://ieeexplore.ieee.org/document/7924372
[25] RepRap, "RAMPS 1.4 - RepRap," Dec. 2022. [Online]. Available: https://reprap.org/wiki/RAMPS_1.4
[26] C. Beckwith, H. S. Naicker, S. Mehta, V. R. Udupa, N. T. Nim, V. Gadre, H. Pearce, G. Mac, and N. Gupta, "Needle in a Haystack: Detecting Subtle Malicious Edits to Additive Manufacturing G-Code Files," *IEEE Embedded Systems Letters*, vol. 14, no. 3, pp. 111–114, Sep. 2022, conference Name: IEEE Embedded Systems Letters. [Online]. Available: https://ieeexplore.ieee.org/document/9619477
[27] Digilent, "Cmod A7 Reference Manual - Digilent Reference," Oct. 2019. [Online]. Available: https://digilent.com/reference/programmable-logic/cmod-a7/reference-manual
[28] Arduino, "Mega 2560 Rev3 | Arduino Documentation," Nov. 2023. [Online]. Available: https://docs.arduino.cc/hardware/mega-2560
[29] J. Prusa, "Original Prusa i3 MK3S+ 3D Printer kit | Original Prusa 3D printers directly from Josef Prusa," Dec. 2023. [Online]. Available: https://www.prusa3d.com/product/original-prusa-i3-mk3s-3d-printer-kit/
[30] S. Liang, X. Peng, H. J. Qi, S. Zonouz, and R. Beyah, "A Practical Side-Channel Based Intrusion Detection System for Additive Manufacturing Systems," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2021, pp. 1075–1087, iSSN: 2575-8411. [Online]. Available: https://ieeexplore.ieee.org/document/9546490