2023

Material extrusion-based additive manufacturing: G-code and firmware attacks and Defense frameworks

Haris Rais
Virginia Commonwealth University

MATERIAL EXTRUSION-BASED ADDITIVE MANUFACTURING: G-CODE AND FIRMWARE ATTACKS AND DEFENSE FRAMEWORKS

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

MUHAMMAD HARIS RAIS

Master of Engineering, N.E.D. University of Engineering & Technology, Pakistan - 2016

Thesis Advisor: Irfan Ahmed,

Associate Professor, Department of Computer Science

Virginia Commonwewalth University

Richmond, Virginia

May, 2023

Acknowledgements

I am deeply grateful to God for granting me this opportunity and providing me with all the necessary help and resources. I would like to express my sincere appreciation to my advisor, Prof. Irfan Ahmed, for his unwavering guidance, sponsorship, and encouragement throughout my academic pursuit.

I am also thankful to Prof. Tamer Nadeem, Prof. Changqing Luo, Prof. Radhika Barua, and Prof. Nastassja A. Lewinski for serving on my doctoral committee and providing invaluable professional advice, guidance, and support throughout my academic journey. I would like to acknowledge Prof. Ye Li from Bradley University for his invaluable assistance with the mechanical domain challenges that arose during my study.

I am equally grateful to my wonderful colleagues at the SAFE lab and the CS department for their help and for making this journey enjoyable. I would also like to appreciate the highly cooperative staff at Virginia Commonwealth University for their dedicated support.

Moreover, I cannot forget to thank my mother, who has been my primary motivation for choosing this path, and my wife, for their support and love. Their encouragement, sacrifices, and faith in me have been the cornerstone of my success.

As I reflect on my journey, I am overwhelmed with deep gratitude and appreciation towards countless individuals who have contributed to my success with their kindness and generosity. I humbly pray for their continued well-being and prosperity.

TABLE OF CONTENTS

Chapter Page

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Abstract

MATERIAL EXTRUSION-BASED ADDITIVE MANUFACTURING: G-CODE AND FIRMWARE ATTACKS AND DEFENSE FRAMEWORKS

By Muhammad Haris Rais

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2023.

Director: Irfan Ahmed,

Associate Professor, Department of Computer Science

Additive Manufacturing (AM) refers to a group of manufacturing processes that create physical objects by sequentially depositing thin layers. AM enables highly customized production with minimal material wastage, rapid and inexpensive prototyping, and the production of complex assemblies as single parts in smaller production facilities. These features make AM an essential component of Industry 4.0 or Smart Manufacturing. It is now used to print functional components for aircraft, rocket engines, automobiles, medical implants, and more. However, the increased popularity of AM also raises concerns about cybersecurity. Researchers have demonstrated strength degradation attacks on printed objects by injecting cavities in the design file which cause premature failure and catastrophic consequences such as failure of the attacked propeller of a drone during flight.

Since a 3D printer is a cyber-physical system that connects the cyber and physical domains in a single process chain, it has a different set of vulnerabilities and security requirements compared to a conventional IT setup. My Ph.D. research focuses on the cybersecurity of one of the most popular AM processes, Material Extrusion or Fused Filament Fabrication (FFF). Although previous research has investigated attacks on printed objects by altering the design, these attacks often leave a larger footprint and are easier to detect. To address this limitation, I have focused on attacks at the intermediate stage of slicing through minimal manipulations at the individual sub-process level. By doing so, I have demonstrated that it is possible to implant subtle defects in printed parts that can evade detection schemes and bypass many quality assessment checks. In addition to exploring attacks through design files or network layer manipulations, I have also proposed firmware attacks that cause damage to the printed parts, the printer, and the printing facility.

To detect sabotage attacks on FFF process, I have developed an attack detection framework that analyzes the cyber and physical domain state of the printing process and detects anomalies using a series of estimation and comparison algorithms in time, space, and frequency domains. An implementation case study confirms that cyber-physical security frameworks are an effective solution against sophisticated sabotage attacks. The increasing use of 3D printing technology to produce functional components underscores the growing importance of compliance and regulations in ensuring their quality and safety. Currently, there are no standards or best practices to guide a user in making a critical printing setup forensically ready. Therefore, I am proposing a novel forensic readiness framework for material extrusion-based 3D printing that will guide standards organizations in formulating compliance criteria for important 3D printing setups. I am optimistic that my offensive and defensive research endeavors presented in this thesis will serve as a valuable resource for researchers and industry practitioners in creating a safer and more secure future for additive manufacturing.

CHAPTER 1

INTRODUCTION

Additive Manufacturing, also known as 3D printing, is a group of seven processes that employ various physical and chemical interactions to join materials and create parts from 3D model data. The process typically involves building each layer of the part based on a predefined direction and sequentially stacking them to match the CAD (Computer-aided Design) model. Despite the differences in building materials and techniques used in each process, the ultimate goal is to create a three-dimensional object. Among the seven categories, the material extrusion-based technique, also known as Fused Filament Fabrication (FFF), is the most commonly used process [1].

1.1 The 3D printing process chain

The 3D printing process is initiated with the creation of a 3D model using any computer-aided design software, as depicted in Figure 1. The 3D model is then transformed into a geometry file, typically in the format of stereolithography (STL). The STL file encompasses data regarding the outer surface of the 3D object as a collection of small triangles. Each triangle is represented by its three vertices and a normal vector, which distinguishes between the inner and outer surface of the object. However, not all STL files are ready for 3D printing. When necessary, an STL repair software is employed to convert the STL file into a format that is compatible with the printer.

The STL file is then fed into a slicing software, also known as toolpath generating software, which converts the geometry into a sequence of printing instructions for the

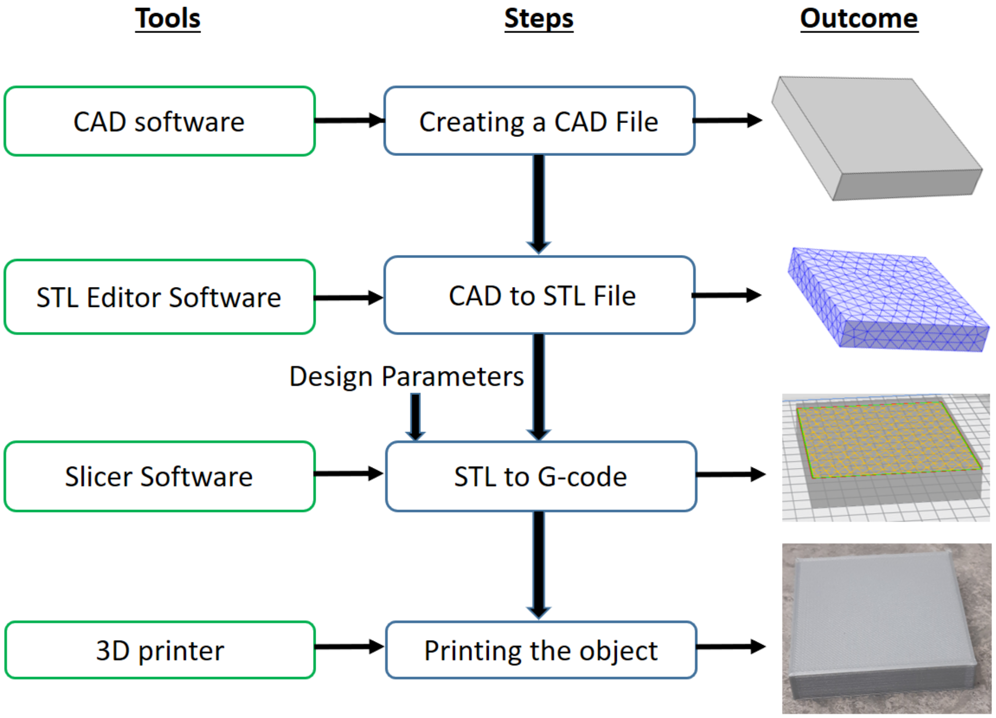


Fig. 1: 3D printing process chain

printer. These instructions are called G-codes.

1.1.1 Printing parameters

Unlike conventional subtractive manufacturing, where a block of material is trimmed from all sides to manufacture the final object, additive manufacturing (AM) offers the flexibility to decide the internal structure of the object, ranging from completely hollow to partially or completely filled. This feature provides several advantages, including saving time and materials. However, it also broadens the attack space. In addition to the infill structure, temperature profile and layer thickness profiles also play an important role in the part's properties. Some of the common printing parameters include infill pattern, infill density, anisotropy (the orientation of printing), printing speed, layer thickness, printing nozzle and bed temperature,

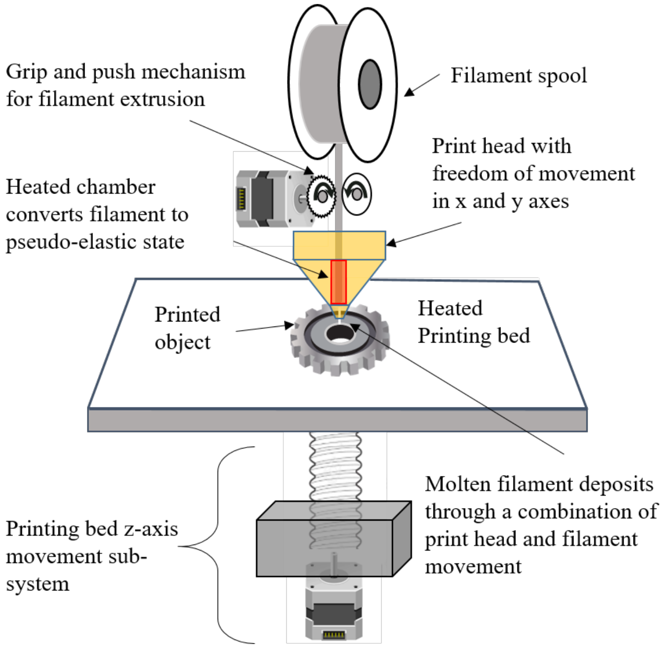

Grip and push mechanism for filament extrusion
Filament spool
Print head with freedom of movement in x and y axes
Heated chamber converts filament to pseudo-elastic state
Printed object
Heated Printing bed
Printing bed z-axis movement sub-system
Molten filament deposits through a combination of print head and filament movement

Fig. 2: A typical Fused Filament Fabrication printer

cooling fan speed, etc. Malicious manipulation of these parameters can affect the mechanical properties of the final printed object. Further discussion on this topic is included in subsequent chapters.

As previously mentioned, the STL file solely contains the outer geometry of the object. The designer incorporates these printing parameters in the G-code file, making it a more elaborate design representation than an STL file. The G-code instructions are then sent to the printer through a USB interface, SD card, or over the IP network using printer control software. The printer firmware interprets the G-code commands and physically executes them to print the object.

1.2 Fused filament fabrication

Fused filament fabrication (FFF) is a highly popular technique in the realm of additive manufacturing (AM). This technology is essentially an open-source version of fused deposition modeling (FDM), which was originally patented by Scott Crump [2]. The rise of FFF printing began after the patents expired in 2009, which prompted new startups to enter the market and offer affordable desktop printers. In the FFF process, a solid filament is heated until it reaches a viscoelastic state and is then gradually pushed out to create a thin layer of filament on the printing platform. Figure 2 provides an example of a common FFF-based 3D printer. Below, I'll elaborate on the two main components of a FFF printer.

1.2.1 Printhead

The printhead assembly typically consists of the printing nozzle and cooling fans. The head is connected to one or two stepper motors through mechanical couplings (such as belts and shafts). Another stepper motor, known as the filament motor, pushes the solid filament through the print head, where it is heated inside the nozzle's heating chamber until it reaches a viscoelastic (molten) state, and is then pushed out. The printhead and filament motor move in synchronization to create the desired geometrical shape of a specific printed layer.

1.2.2 Printing bed

The object is printed on a flat surface, typically made of glass or metal, known as the printing bed or printing platform. The printing bed is usually heated to around the glass transition temperature of the printing material to avoid warping of the object. When printing begins, the printing bed moves up to the printhead, leaving a space equivalent to the first layer's thickness. Once one layer is completed, the bed moves away from the head to create space for the next layer.

1.2.3 Main processes involved in FFF printing

FFF is a complex interplay of numerous sub-processes, including kinetics, thermodynamics, crystallization [3], and glass transition [4]. From the standpoint of a cyberattack, we identify a set of primary processes that can be influenced directly by instructions issued in the cyber domain. In FFF process, thermodynamics and kinetics are the primary processes that can be directly influenced by malicious commands initiated by a cyberattacker. The thermodynamic profile of the printed part involves the heating and cooling profiles of the nozzle and the printing bed, while kinetics include the kinetic profiles of the printhead, filament, and printing bed.

It is worth noting that monitoring the state of the part during printing is equivalent to monitoring the primary sub-processes involved in printing.

1.3 Attacks on AM process

Like any other computing system, Additive Manufacturing (AM) is susceptible to traditional attacks on confidentiality, integrity, and availability. However, being a cyber-physical system, AM is also vulnerable to a different kind of attack that involves both digital and physical components to complete an attack chain. Two common categories of cyber-physical attacks on AM processes are intellectual property (IP) theft attacks and sabotage attacks.

IP theft attacks exploit physical side channels to extract object design information, while sabotage attacks actively manipulate the printing process in the cyber-domain to influence the object's physical properties. The impact of a sabotage attack on the printed part depends on the target system for which the part is being printed, thereby increasing the incentives for cyber attackers. For instance, an attacker interested in destroying a power grid turbine may manipulate the printing process to induce inconspicuous defects in the turbine blade at the time of printing. While quality control checks are in place to detect defective parts, these tests are generally not optimized for or designed to detect malicious interventions.

This work also presents a set of attacks on the printing service availability, named denial of printing service (DoPS) attacks. The proposed DoPS attacks utilize cyber-physical interactions to result in the denial of service.

1.4 Motivation for dedicated research in securing AM process

The pace of technological advancement is accelerating rapidly, and the manufacturing industry is no exception. The first three industrial revolutions took more than two centuries, but the fourth revolution, known as Industry 4.0 or Smart Manufacturing, arrived in less than fifty years. One crucial element of this initiative is Additive Manufacturing [5]. According to Global Industry Analysts, Inc., the AM industry is projected to grow at a cumulative annual growth rate (CAGR) of 21.8% over the next five years [6]. Unlike a decade ago, when 3D printing was used mainly for prototyping, it is now being used to print important functional parts, providing higher incentives to cyber attackers. This emphasizes the need for a commensurate research effort in securing the AM process against cyberattacks.

Every major technological advancement brings new cybersecurity challenges. For instance, the adoption of IT Cloud opened up new cybersecurity attack vectors, leading to dedicated cybersecurity research, resulting in specialized security and forensic models for IT Cloud services [7]. Similarly, the AM process is fundamentally different from pure cyber systems, or from previous manufacturing technologies such as subtractive manufacturing. Therefore, focused research is required to examine the gaps in current security frameworks and create the most appropriate models for securing AM processes against cyberattacks.

Unlike conventional IT systems, Additive Manufacturing represents a cyber-physical system. While the printing process begins in the cyber domain, the later stages and the final outcome, the printed object, exist in the physical domain. Rather than relying on conventional cybersecurity solutions, incorporating knowledge of the physical domain in the cybersecurity loop provides better visibility of the process, resulting in improved security.

These factors provide a strong incentive for dedicated research efforts in cybersecurity for AM processes.

1.5 Research objectives and contributions

The main objective of this research is to improve the state of the art of cybersecurity in the FFF printing process. To achieve this objective, the research effort is focused on three aspects of cybersecurity: (1) process-level vulnerabilities and attack opportunities, (2) attack detection techniques and methods, and (3) attack analysis or forensics.

In my research, I claim the following contributions toward enhancing the security of FFF printing:

1. I propose low-magnitude and localized kinetic and thermodynamic attacks to sabotage the printed part while staying below the detection horizon. Additionally, I demonstrate that smart kinetic deviation within the printer specification tolerance zone may also impact the extrudates bonding process and the strength of the object. These attacks can significantly reduce the strength of the printed parts, as confirmed by the experimental results.

2. I introduce a novel firmware attack taxonomy and categorization tree, along with a set of firmware attacks that can be used to surveil and sabotage the FFF

printing environment and the printed part.

3. To address these security concerns, I present PrintSafe – a modular and near real-time attack detection framework. PrintSafe incorporates several techniques to detect malicious activity during the printing process, including G-code modeling, firmware function fingerprinting, and independent acquisition of the printing state. PrintSafe analyzes the printing process in the time, space, and frequency domains to identify any anomalies that may indicate a security breach.

4. Finally, I propose a novel forensic readiness framework for FFF printing that provides guidance on how to preserve and collect evidence in the event of a security incident. This framework is intended to assist users in conducting a thorough forensic investigation and identifying the root cause of any security breaches.

Together, these contributions represent a significant step towards improving the security of FFF printing, and I believe they have the potential to benefit both individual users and the wider FFF printing community.

1.6 Organization of the dissertation document

The rest of the thesis is divided into two parts, each covering different research endeavors. Part I focuses on offensive research. Chapter 2 proposes sabotage attacks that exploit thermodynamic and kinetic processes during the designing and slicing stages of printing. Our attacks are designed based on stringent criteria to be highly effective in reducing the strength of the printed object, while simultaneously staying below the detection threshold. A fundamental aspect of FFF printing is extrudates bonding. In chapter 3, we propose attacks to weaken the extrudates bonding with such low magnitude deviations that overlap with the printer's trueness specifications, making them difficult to detect. Chapter 4 presents a minimally explored area of research - firmware attacks - and introduces novel firmware attack taxonomies and nine attacks for the surveillance and sabotage of the printer, the printing facility, and the printed object.

Part II of the thesis consists of two chapters. Chapter 5 presents a modular cyber-physical framework that can detect sabotage attacks and anomalies. It also includes an implementation case study and an evaluation of the framework's effectiveness against 33 sabotage attacks. In Chapter 6, a novel forensic readiness framework is presented for FFF printing processes. This framework identifies the key information sources and their capturing methodology to attain a forensically sound repository. A case study demonstrates how the proposed framework can help a forensic investigator identify the attack and the attackers' details.

Finally, Chapter 7 presents the thesis's conclusions, summarizing the key findings and their implications for the field of additive manufacturing cybersecurity.

Part 1: Attacking Fused Filament Fabrication Process

CHAPTER 2

DYNAMIC-THERMAL AND LOCALIZED FILAMENT-KINETIC ATTACKS ON FUSED FILAMENT FABRICATION BASED 3D PRINTING PROCESS

This chapter presents four new sabotage attacks on the fused filament fabrication (FFF)-based 3D printing process: 1) cavity through filament-kinetics, 2) density variation through filament state, 3) density variation through filament speed, and 4) dynamic-thermal manipulation. These attacks produce an insignificant attack footprint on a finished printed object by targeting localized regions or using small changes in temperature profile, making them hard to detect. Specifically, the first three attacks manipulate filament-kinetics to change the print density or create a cavity in a small localized region, while the fourth attack makes slight changes to the nozzle temperature to manipulate thermal stress in a printing object without creating any visual deformation. Mechanical (tensile and three-point bending) tests carried out on the objects under attack demonstrate that these attacks with insignificant attack footprints can still change the physical properties (e.g., stress and strain) of the printed objects.[1]

2.1 Introduction

3D printing collectively refers to a group of manufacturing processes that materialize physical objects through sequentially depositing thin layers [9]. Although each process applies different physical/chemical interactions on the building materials,

[1] *This work is based on my paper which appeared in the Additive Manufacturing journal in 2021[8]*

each layer is constructed using a predefined building direction and stacked sequentially based on a CAD (Computer-aided Design) model. The layer-by-layer stacking in a 3D printing process exposes it to a different set of vulnerabilities than other manufacturing processes, such as machining. Generally, attackers target a 3D printing process with one of the two objectives: 1) intellectual property theft, and 2) sabotage attack [10]. The seminal work of Farouque *et al.* [11] demonstrates intellectual property theft using side-channel acoustic signals by printer motors. On the other hand, sabotage attacks weaken, damage or destroy a 3D-printed object by causing geometrical nonconformity and workpiece deformation [12]. This work demonstrates and measures the effectiveness of new filament-kinetics and thermal stress-based sabotage attacks on FFF printing process.

Researchers have demonstrated that the object properties can be altered by manipulating the manufacturing parameters of a printing process such as object orientation [13], fan speed [14], nozzle temperature [15], printing bed temperature [16], and fusing material patterns [17]. Changing the printing parameters affect one or more of the three processes involved in FFF-based 3D printing i.e., nozzle-kinetics, filament-kinetics and thermodynamics. Up till now, the focus of attack detection research in additive manufacturing remains on the nozzle-kinetics [14, 18, 19, 20, 21]. Although, the adverse and conspicuous effects of manipulating filament-kinetic and thermodynamic profiles have been demonstrated over the entire object earlier (discussed ahead in section 2.2.1), the practicality of achieving inconspicuous and localized attacks that degrade the mechanical properties of the printed object is not yet explored.

This chapter presents four new attacks in this direction on the FFF-based 3D printing process. The attacks are 1) cavity through filament-kinetics, 2) density variation through filament state, 3) density variation through filament speed and 4) dynamic-thermal manipulation. They are designed to produce an insignificant attack footprint (discussed in section 2.5.1) on a finished printed object, making them hard to detect. Precisely, the first three attacks manipulate filament-kinetics to change the print density or create a cavity in a small localized region. The fourth attack makes small changes to the nozzle temperature to alter the thermal stress profile of the printing object without creating any visual deformation.

We implement the attacks over PLA (Polylactic Acid) printed rectangular bars and perform the tensile and three-point bending tests to evaluate both attacked and benign specimens. The evaluation results find that the attacked bars show a noticeable deviation in physical properties such as peak load, flexure stress, and strain. Note that the attacks are generally applicable to all common FFF-based 3D printers that share same set of printing process parameters.

The contribution of the work is threefold:-

- We demonstrate new *localized filament-kinetic* attacks for cavity creation and density variation without changing the printing path sequence. Moore *et al.*'s work [22] is closest to our filament-kinetic attacks in that they modify the feed-rate parameter for the entire printing of an object. However, our localized filament-kinetic attacks target specific object regions, with minimal to no change in object weight, center of gravity, dimensions, and nozzle kinetic process to achieve concealed internal cavities or density variations.

- We demonstrate new *dynamic-thermal* attacks that do not create any visual deformation. Claud *et al.*'s work [23] is closest to our dynamic-thermal attacks in that they increase the nozzle temperature for the entire printing process. However, our dynamic-thermal attacks use planned, localized, and minor modifications in the thermodynamic profile and further ensure invisible deformation.

- A subsequent question arises about the effectiveness of such inconspicuous and minute changes targeted at specific sub-processes. In our work, we show that such attacks are effective in modifying the mechanical properties of the object. Specifically, we perform mechanical (tensile and three-point bending) tests on 3D printing objects under attack to validate the impact of localized filament-kinetic and dynamic-thermal attacks on the physical properties (e.g., stress and strain) of the object.

2.2 Background and related work

2.2.1 3D Printing sabotage attacks

Most of the existing attacks sabotage a 3D object by modifying the design files, i.e. the first stage of the process chain. The changes in the CAD file simultaneously modify the filament-kinetics and nozzle kinetics, resulting in a bigger attack footprint easier to detect by existing techniques. The changes at the advanced level in the process chain, such as the G-code or firmware can manipulate only a specific sub-process resulting in a smaller attack footprint. However, the existing work is only limited to manipulating nozzle kinetics during a printing process.

Design file modifications Sturm *et al.* [24] manipulates the STL files to create a denial of service, indents, scaling, and void attacks without modifying printing parameters during a printing process. They present a case study of void attacks by using heuristic rules to find a high-impacting location in the object and then create an enclosed void.

Belikovetsky *et al.* [12] demonstrate an attack on a 3D-printed quadcopter propeller by reducing its fatigue life, which causes it to fail prematurely during mid-flight. Specifically, they target the joint connecting the blades to the cap of the propeller and introduce gaps between them. Apparently, the gaps weaken the mechanical strength below operational conditions and thus, cause the propeller to break within seconds of normal operation.

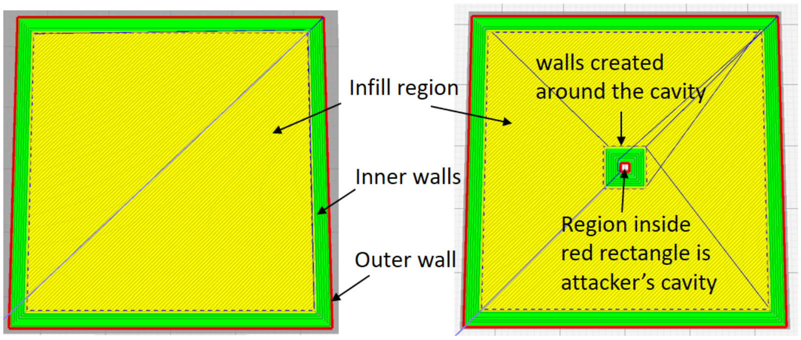
Nozzle kinetics Zeltman *et al.* [13] demonstrate a print orientation attack resulting in degradation of the mechanical properties of the printed object. Moore *et al.* [22] hijack the printer firmware and change the internal feed-rate variable by 10% to 40%, resulting in a deformed printed object. The attack impact is visible, distributed over the entire object, and also changes the object's weight proportional to the modification percentage. Similarly, Claud *et al.* [23] hijack the printer firmware and increase the temperature variable value while reporting the actual temperature.

2.3 Proposed sabotage attacks on FFF-based 3D printing

We present new *localized filament-kinetic* and *dynamic-thermal* attacks with an insignificant attack footprint (refer to Table 2) on a finished printed object. The attacks can modify the physical properties (e.g., peak load, flexure stress, and strain) of a target object.

2.3.1 Localized filament-kinetic attacks

In FFF based 3D printer, a stepper motor (called filament motor) pushes the filament through the nozzle to extrude the material from the nozzle tip. The filament motor works synchronously with the printhead motors (x and y axes motors) to ensure a homogeneous print density across the entire object. We propose three attacks that manipulate this relation by briefly tweaking filament motor kinetics to target a small localized region of a 3D printing object. The attacks target the intermediate layers and are concealed by the unmodified top and bottom layers.


Infill region
Inner walls
Outer wall
walls created around the cavity
Region inside red rectangle is attacker's cavity

(a) Actual Deign 5cm x 5cm x 1cm cube – single layer shown

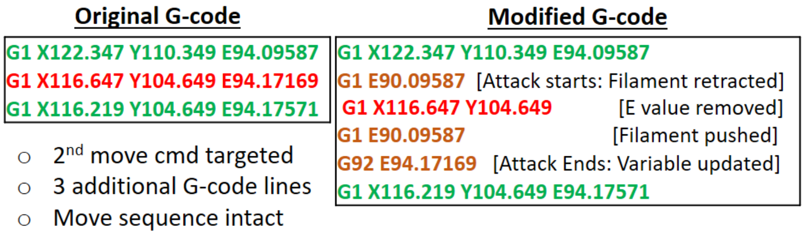
(b) 1mm x 1mm cavity introduced in design

Impact on Printing-Per Layer Stats	**(a) No Cavity**	**(b) Cavity via Design File**
No of mov commands	164	218
Time taken (sec)	57	59
Sequence	Infill > object inner-walls > object outer-wall	Infill > cavity inner-walls > cavity outer-wall > object inner-walls > object outer-wall

Fig. 3: Cavity attack through STL or CAD file modification

2.3.1.1 Cavity attack through filament-kinetics

This attack prevents the extrusion of the filament over the target area of an object to create a cavity. It can be performed at any stage of the process chain. However, if the attack modifies the design file at CAD or STL stages, it will have a much bigger attack footprint than a cavity size. To illustrate further, consider Figure 3, where a cavity attack using a design file produces a refined cavity visible as a small square enclosed within red and green concentric squares. When a slicer software finds a tiny cavity in an object's design, it creates inner and outer walls around it, similar to the object borders. In this attempt, the toolpath sequence is significantly disturbed and can be detected. For instance, the move instruction count in this example is increased from 164 to 218 per attacked layer, and the time to print is raised over 2 seconds for higher-speed upper layers and over 4 seconds for lower-speed starting layers. From the



Original G-code

```
G1 X122.347 Y110.349 E94.09587
G1 X116.647 Y104.649 E94.17169
G1 X116.219 Y104.649 E94.17571
```

- 2nd move cmd targeted
- 3 additional G-code lines
- Move sequence intact

Modified G-code

```
G1 X122.347 Y110.349 E94.09587
G1 E90.09587  [Attack starts: Filament retracted]
 G1 X116.647 Y104.649         [E value removed]
G1 E90.09587                  [Filament pushed]
G92 E94.17169  [Attack Ends: Variable updated]
G1 X116.219 Y104.649 E94.17571
```

Fig. 4: G-code snippet of filament-kinetic cavity attack

attacker's perspective, the effectiveness of a cavity protected by multiple protecting walls, is also questionable in reducing the object's physical properties.

Our proposed cavity attack is performed through filament-kinetics only, with minimal change in the move instructions (zero in most cases), no change in toolpath sequence, and minimal printing-time difference from the original design. Due to the abrupt removal of material, the cavity through filament-kinetics does not create a well-designed structure, arguably increasing the chances of higher reduction in mechanical properties. Figure 4 illustrates the modified G-code instructions as a sample for our cavity attack. There is no move instruction inclusion against the printing sequence. Three lines are added in the G-code for attacking a single move instruction (0.3 to 0.5mm thickness). Modifying the G-code file is one way of launching this attack. If the attacker controls the firmware, the attack can be performed by controlling the motor directly.

2.3.1.2 Density variation through filament state attack

This attack manipulates the filament motor state (i.e. ON or OFF) to reduce the material density in an object's target region without disturbing nozzle kinetics and toolpath. To launch the attack, the filament motor state is changed to OFF. We observed that the switching OFF event gradually affects the target object due to residual filament available at the nozzle's tip that continues to extend the filament string. However, with the filament supply cut-off, the material density in that zone is less than the actual desired value. The situation is analogous to the last sentence written from the ink pen whose cartridge is detached. As the maximum impact occurs with a delay, the attack is initiated slightly earlier than the target area. To find the exact starting point, we consider two possible cases. In case I, the attack duration is much smaller than the time to completely dry out the residual filament as shown in equation 2.1.

$$\Delta t_{attack} = |t_{ON} - t_{OFF}| < (d_{res} \ / \ v_{nozzle}) \tag{2.1}$$

where Δt_{attack} is the attack duration, $t_{ON}\ and\ t_{OFF}$ are the filament motor's switching OFF and switching ON time, d_{res} is the distance to completely consume the residual filament and v_{nozzle} is the nozzle speed at that instance. In case II, the attack duration is comparable or greater than the time to consume the residual filament, as shown in equation 2.2.

$$\Delta t_{attack} = |t_{ON} - t_{OFF}| \geq (d_{res} \ / \ v_{nozzle}) \tag{2.2}$$

To calculate the maximum impact point of the attack, we investigate the value of d_{res}, which is a function of nozzle speed, nozzle temperature, material properties, max acceleration and jerk settings in the G-code, as shown in equation 2.3. For case I, the weakest density point corresponds to the nozzle position when the filament motor is switched back ON. The material density remains higher in the neighboring regions. In case II, where attack duration is prolonged enough to dry out the residual filament completely, we attain a zero material density zone with gradual downward density slope to precede it in response to the switching OFF event, and a steeper upward slope to follow the switching ON event.

$$d_{res} = f(G_{nozzle}\ , V_{nozzle}\ , T_{nozzle}\ , max(v,a,j)\ , h) \tag{2.3}$$

where G_{nozzle}, V_{nozzle} and T_{nozzle} are the nozzle's geometry, speed and temperature, max(v, a, j) are the maximum speed, acceleration and jerk settings in the printing profile, "h" is the printing material viscosity property. Although these functions can be calculated through fluid dynamics knowledge, our experiments show that the d_{res} value remains around 25 mm to 30 mm for the settings used in our experiment.

The attack is designed by choosing the highest impact region of an object, calculating the t_{OFF} time as per the desired intensity of the attack, and then, modifying the G-code commands to mute the filament in the region of interest. We identify two variants of the attack depending upon the object design and G-code commands.

Variant I: The attack may require removing filament field "e" value in a single or multiple G-code instructions, and then updating the current filament length variable through "G92" command.

Variant II: The attack split a single command that may cause a very minimal time variation. The reason is the max acceleration and higher order peak value settings that have to be adhered when the command is split. However, the effect is very minimal (less than 20 ms as observed under most common conditions). The attacker can still avoid it by increasing the zone of attack to the complete instruction. To be effective, this attack usually continues for more than one commands, thus rounding off to next complete command is not a big change. In attacks where command splitting is not required, nozzle kinetics is not affected at all.

2.3.1.3 Density variation through filament speed attack

This attack modifies the filament motor speed to manipulate the relation between filament and nozzle kinetics without changing the nozzle speed. In FFF printers, a printing move command may trigger the movement of 3 motors (x, y and filament e). The axis undergoing the biggest move inherits the max-speed value. The printer proportionally adjusts the speed of the remaining axes to ensure that all motors start and stop at the same time, and that the filament deposition is symmetric throughout the path.

In this attack, the attacker reduces the filament speed in the target zone and compensates for the slowness by increasing the filament speed over the non-critical areas, thus ensuring that the target area receives lesser filament, while the object's net weight still remains the same. Although the weight difference in the other 2 attacks is also minimal, it is zero in this attack. There is also no impact on the toolpath sequence or the printing time.

2.3.2 Dynamic-thermal attacks

The thermodynamics of a FFF printer is a complex process and is critical for the printed object's health. Asymmetric heating and cooling profiles at different regions in an object may create residual thermal stresses [25]. If the thermodynamic profile is tweaked too far from the optimal setting, it results in noticeable deformation or warping. On the other side, tiny changes do not consistently change the material properties. The critical question in designing this attack is to find small magnitude deviation patterns that do not create any visible deformation but still change the object properties.

There are two heating elements in a FFF printer, i.e., the printing nozzle and heated bed. This attack mainly targets the nozzle's temperature and does not alter any nozzle- and filament-kinetics. There are two ways to set the nozzle temperature. First, the “M109” command pauses the printing, the printer attains the desired temperature, and resumes the printing. Creating a steep temperature variation will result in higher thermal stress. Thus, it is desirable from the attacker's standpoint; however, pausing the printing for a few seconds creates enough deviation in nozzle kinetics to generate an alert. Second, the “M104” command continues to print while simultaneously working to achieve the desired temperature. The temperature fluctuation is gradual.

This attack utilizes the “M104” command to instruct the temperature change before the nozzle passes over the target area. A small change in a single layer has a negligible impact on the object. To increase the attack impact, multiple internal layers are printed with the same modified profile. Different temperature profiles incur consistently different outcomes in physical properties. Thus, the attacker can use a temperature profile that can give the desired impact on material properties. Note that when the temperature is reduced, under-extrusion will also occur. However, in this attack, we ensure that the temperature fluctuations are within the printer's extrusion capabilities.

2.4 Attacks implementation

2.4.1 Adversary model

2.4.1.1 Assumptions

Our threat model assumes that the CAD and STL files are intact; they are protected by other security measures such as file integrity checker. However, the attacker compromises the printer firmware and G-code (e.g., Harvey [26]) and installs a rootkit that can manipulate the printing process, including printer (internal) sensors and actuator movements. This model is commonly used by the existing security research on 3D printers [21, 22, 26, 23].

2.4.1.2 Attacker's goal

The user is printing a batch of critical rectangular bars exposed to tensile and bending stresses during operation. The attacker aims to carry out inconspicuous attacks on the target object to achieve degradation in mechanical properties while evading the visual inspection and necessary quality checks (such as weight and the center of gravity).

2.4.1.3 Attack method

To achieve the goal, the attacker utilizes our proposed filament-kinetic and dynamic-thermal attacks. In control of the printer's firmware and G-code, the attacker generates the G-code files, each containing one type of attack sequence, and utilize it in the target 3D printer to attack the printing object.

2.4.2 Experimental settings

We implement the attacks on rectangular bars with dimensions (60mm length x 6mm depth x 4mm height). The bars are printed with PLA material through the Ultimaker-3 printer hosting a 0.4mm nozzle. On a control PC connected through the IP network, Ultimaker Cura 4.0.0 is used as the slicer and controlling software. The default object settings include 0.2mm layer thickness (making 20 layers in the object), 100% infill density, and ”Line” infill-pattern at 45^o angle. The default temperature for layer-1 is 210^{0}C, while 205^{o}C for the remaining layers. The printing speed is set at 50 mm/sec.

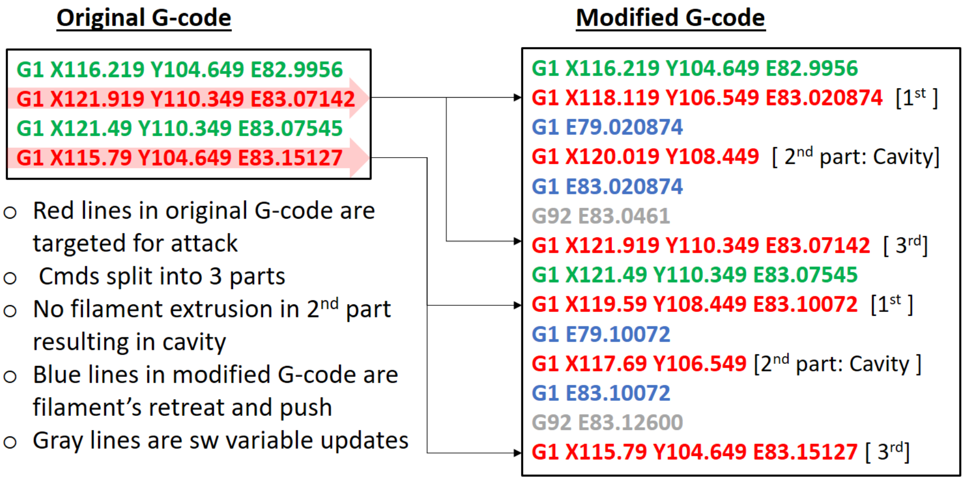
2.4.3 Localized filament-kinetic attacks

2.4.3.1 Cavity attack through filament-kinetics

Three layers from the top and bottom are not modified to attain a cavity within the internal layers. The attacker calculates the bar's central point and modifies the G-code move instructions near the center. To keep the cavity fully encapsulated from all sides, the attacker splits each line into three parts and produces a cavity only in the middle part of the line. The effective cavity dimensions per layer are around 2 mm x 0.6 mm. The filament values are adjusted using the methods described in Section 2.3. The attack starts after the 1^{st} part of the attacked command is completed. The filament is retracted by 4mm. This value is measured empirically, starting from 1mm upwards to ensure minimum retraction distance that results in zero residual filament, resulting in a clean cavity at the target spot. After retraction, the nozzle follows the toolpath for the 2^{nd} part, but without filament extrusion. When the nozzle reaches the end of the cavity, the filament is pushed back 4mm; the filament length variable is updated to avoid 'e' value modification in the subsequent commands. The 3^{rd} part is then printed normally. The phenomenon is explained in Figure 5.

2.4.3.2 Density variation through filament state attack

In this attack, the attacker mutes the two adjacent infill lines near the center of the object. One small line (0.5mm) that connects these two infill lines is also muted as a consequence. The attack is made by removing the "e" field in the required move instructions and updating the software's filament length variable. Although no visual impact is expected in this attack as the nozzle stays well within the case-1 discussed in Section 2.3.1.2, the attacker keeps the top three and bottom three layers unmodified for re-assuring no visual anomaly in the finished object. The purpose of extending



Original G-code

```
G1 X116.219 Y104.649 E82.9956
G1 X121.919 Y110.349 E83.07142
G1 X121.49 Y110.349 E83.07545
G1 X115.79 Y104.649 E83.15127
```

- Red lines in original G-code are targeted for attack
- Cmds split into 3 parts
- No filament extrusion in 2nd part resulting in cavity
- Blue lines in modified G-code are filament's retreat and push
- Gray lines are sw variable updates

Modified G-code

```
G1 X116.219 Y104.649 E82.9956
G1 X118.119 Y106.549 E83.020874  [1st ]
G1 E79.020874
G1 X120.019 Y108.449  [ 2nd part: Cavity]
G1 E83.020874
G92 E83.0461
G1 X121.919 Y110.349 E83.07142  [ 3rd]
G1 X121.49 Y110.349 E83.07545
G1 X119.59 Y108.449 E83.10072  [1st ]
G1 E79.10072
G1 X117.69 Y106.549 [2nd part: Cavity ]
G1 E83.10072
G92 E83.12600
G1 X115.79 Y104.649 E83.15127 [ 3rd]
```

Fig. 5: G-code comparison of single instance of cavity attack

this attack over more layers is to increase the attack impact on the object properties.

2.4.3.3 Density variation through filament speed attack

In this attack, the attacker selects four lines in the center of the object and reduces their material density by 20% by reducing the value of Δe (which is the difference in the filament length value for the ith and (i-1)th command). To compensate for the anticipated reduction in weight, the attacker selects four lines in the middle of the right half and four lines in the middle of the left half and distributed the lost material equally among them. As per the attacker's knowledge, these locations are less critical from the object's operational perspective. The top and bottom three layers are not attacked in this case as well. Although the deviation magnitude in this attack is smaller than the previous two attacks, a larger area of the layer is disturbed in this case.

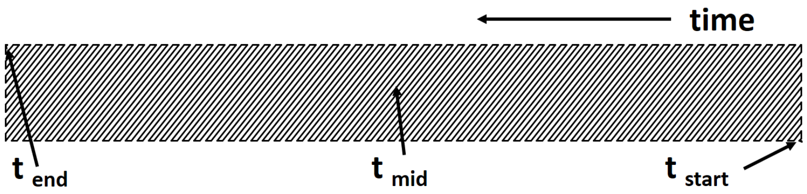


Fig. 6: Infill printing sequence

2.4.4 Dynamic-thermal attacks

The variation of temperature at different locations in an object induces thermal stresses that impact the printed object's mechanical properties. Exploiting this fact, the attacker manipulates the nozzle temperature by a small magnitude (±12^{o}C), causing residual thermal stress with no visible deformation or warping. The magnitude was selected after no particular trend was observed with a deviation of (±5^{o}C) and (±7^{o}C) for the chosen attack types. The printing of each layer starts at one end (bottom right as per the user's settings) and finishes at the other end (top left), as shown in Figure 6.

When the printing of the attacked layer starts, the temperature begins to change due to the influence of the M104 command inserted by the attacker. The printer proceeds toward the center in a line-by-line fashion. As the nozzle reaches the center, another M104 command is issued to revert the temperature to the default value. Before the nozzle reaches the other end of the object, it attains the default temperature. Sufficient time is required to ensure that the required change in temperature can be achieved as the printer reaches the center. In addition to the warping over higher temperature changes, this factor also creates an upper bound on the attack magnitude.

Figure 6 shows a sufficient time-gap between the printing of the central portion and the sides of the bar to attain the temperature difference. If the infill pattern angle is changed to 0^o, launching the dynamic-thermal attack in this manner would not be possible. However, the infill angle (or raster angle) value is an important design decision with a significant impact on the material properties [27], and its modification is not a trivial operational change.

Two different attack patterns are used in our experiment. In the first attack, the central part is printed at 12^oC higher temperature than the default value, while in the second attack, the central part is printed at 12^oC lower temperature. We observed minor variation (around $\pm 2^o C$) in the peak temperature difference induced in different samples.

2.5 Evaluation results

2.5.1 Stealthiness standpoint

The stealthiness of the attack is part of the success criteria. We observed no change in the object dimensions for the attacked samples. Table 1 shows the results of the object dimensions measured for 5 samples of each type of attacked and benign specimens. The stealthiness also includes changes in the printing time, weight differences, or modification of the toolpath. Table 2 summarizes the attacks' performance from a detection or stealthiness standpoint. We did not observe any visual indication of the attacked objects throughout the printing process. One exception is the clean-cavity attack, where a cavity can be visible to an observer during the internal layers' printing. Ultimately, the cavity is covered by the non-attacked top layers in due course. In the case of dynamic-thermal attacks, when temperature reduction was attempted over 15^oC, occasionally warping was observed in the workpiece. We restricted our attacks to 12^oC where no deformation was observed for any specimen

Attack Type	Length (mm)		Width (mm)		Height (mm)	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
No Attack	60.52	0.10	6.57	0.03	4.14	0.02
Clean Cavity Attack	60.53	0.05	6.55	0.03	4.14	0.01
Filament State Attack	60.54	0.07	6.59	0.02	4.14	0.01
Filament Speed Attack	60.54	0.04	6.61	0.03	4.14	0.01
High Temperature	60.53	0.05	6.61	0.04	4.15	0.01
Low Temperature	60.58	0.07	6.56	0.03	4.16	0.02

Table 1.: Statistics of attacked and benign samples

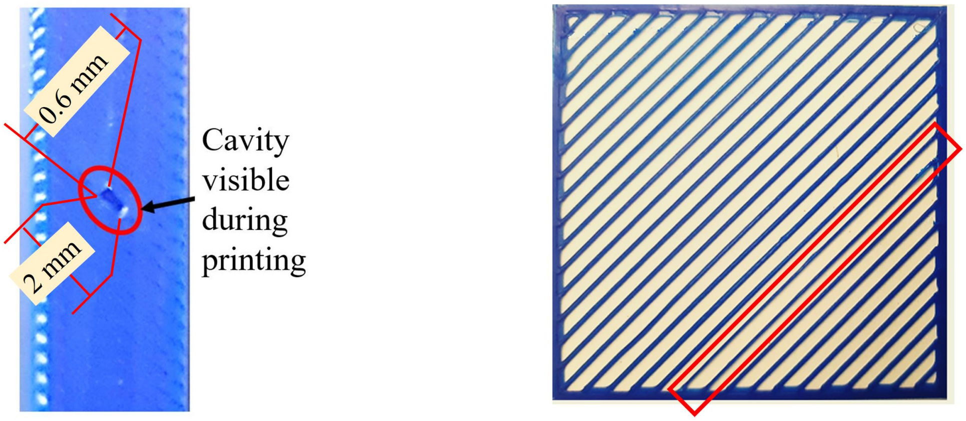
above the available 3D-scanner resolution (0.5 mm). All other attack samples were printed smoothly.

2.5.2 Confirmation of parameter changes

To examine if our attacks change the desired parameters, we carried out a few exercises. For the cavity attack, we know that the cavity (if created) will be visible during the printing of the attacked layers. As shown in Figure 7a, the cavity is created at the correct spot as intended. The image is captured during a test print by pausing the printing during one of the attacked layers. Since there is no visual indication for the other two filament-kinetic attacks, we printed a rectangular prism with less infill density and higher attack magnitude (by muting the filament motor for more time) to confirm visual deformation. Thinning of infill lines can be visually observed in the target area, as seen in Figure 7b. For dynamic-thermal attacks, we installed a thermocouple near the tip of the nozzle to examine the actual temperature during the printing. Figure 8a shows the traces of the high-temperature profile attack. The middle part is printed at the highest temperature, and both sides are printed at the default temperature. Figure 8b shows that the nozzle temperature is reduced

Attack category	**Attack name**	**Visible change**	**Change in outer dimensions**	**Toolpath sequence change**	**Printing time change**	**Weight difference**
Filament-kinetic	Clean cavity	Yes (only in internal layers during printing)	None	None	0.4 sec per attacked layer	< 1%
Filament-kinetic	Filament state	None	None	None	None	< 1%
Filament-kinetic	Filament speed	None	None	None	None	None
Dynamic-thermal	High temperature	None	None	None	None	None
Dynamic-thermal	Low temperature	None	None	None	None	None

Table 2.: Attack footprints from detection standpoint



(a) Image taken after pausing the printing during middle layers

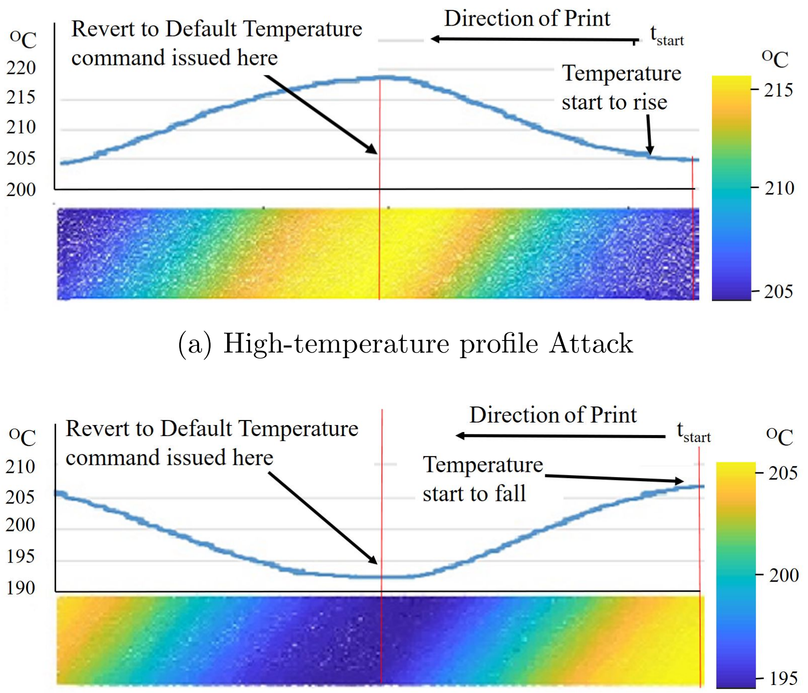
(b) Exaggerated attack magnitude results in obvious thinning of the filament

Fig. 7: Filament-kinetic attacks - Parameters deviation evidence

when the central part of the object is being printed. The traces confirm that the temperature is modified for the attacked prints. Figure 8 does not represent a thermal image taken at one instance; rather, it provides the temperature value of each pixel when it was printed.

2.6 Mechanical testing for attack impact measurement

After confirming that the attacks met the criteria of inconspicuousness and parameter modification, we carried out the mechanical tests of the original prints and the attacked prints. We performed two important destructive tests: a tensile strength test using *MTS Insight 30* and a three-point bending test using *Instron 5948* test equipment. Filament-kinetic attacks and dynamic-thermal attacks were carried out in two separate circumstances. Therefore, a separate set of default prints is used for each of them.



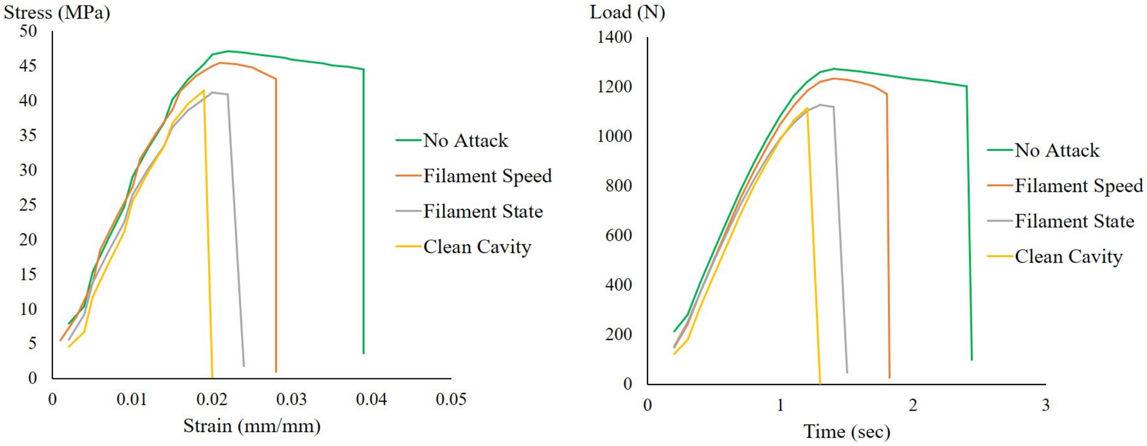
(a) High-temperature profile Attack

(b) Low-temperature profile attack

Fig. 8: Temperature profile for dynamic-thermal attacks

Attack Types	**Peak Load (N)**		**Peak Stress (MPa)**		**Peak Strain (mm/mm)**		**Modulus (MPa)**	
	Mean	**Std Dev**	**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Mean**	**Std Dev**
No Attack	1273.252	43.380	47.140	1.856	0.023	0.001	2916.865	81.932
Clean Cavity	1123.876	81.768	41.580	3.178	0.019	0.002	2880.255	48.643
Filament State	1135.753	29.283	41.940	1.060	0.020	0.000	2779.170	68.891
Filament Speed	1230.394	36.287	45.380	1.203	0.023	0.001	2814.338	81.932

Table 3.: Tensile tests summary for filament-kinetic attacks



(a) Stress vs Strain

(b) Load vs Time

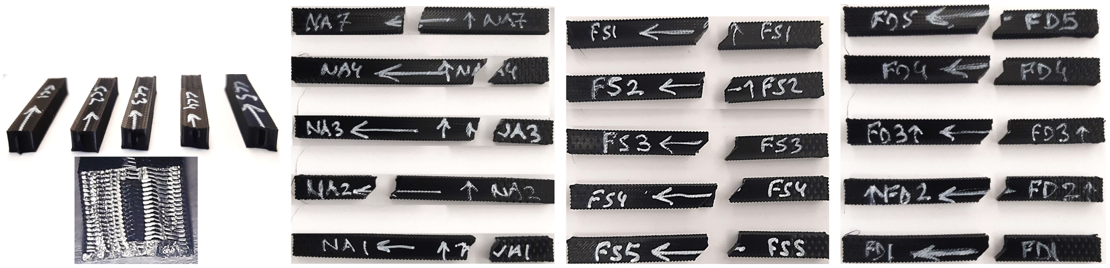
Fig. 9: Tensile test results for filament-kinetic attacks

2.6.1 Mechanical testing of filament-kinetic attacks

2.6.1.1 Tensile Test

The results of tensile tests for the filament-kinetic attacks are summarized in table 3.

All the attacks show a decrease in the peak load, peak stress and the modulus value. Clean cavity attack shows the biggest reduction in the peak load and stress values, followed by the filament state attack. However, young's modulus reduction



(a) Filament cavity attack (b) Samples with no attack (c) Filament state attack (d) Filament speed attack

Fig. 10: Filament attacks specimens after tensile tests

for the cavity attack was minimal compared to the filament state and filament speed attacks. Another interesting finding is the reduction in yield value for all types of attacks, as visible in Figure 9b.

All the attack samples broke earlier than the non-attacked samples. Another important observation is that the samples broke exactly at the point of attack, while the non-attacked samples break at random locations as visible in Figure 10b. In the case of cavity attack, the cavity got exposed after the failure, as shown in Figure 10a. During the investigation of the part failure, this evidence can point to the presence of an attack. As shown in Figure 10c and 10d, there was no obvious indication leading to the presence of any attack in the other two cases.

2.6.1.2 Three-point bending test

The three-point bending tests also show a minor reduction in the peak stress value for the attack samples. Clean cavity attack samples are the fastest to break, followed by filament state attack samples. Filament speed attack samples did not break till the maximum extension limit of the test. It is also evident from the heaviest-tailed curve of filament-speed attack in Figure 11. It indicates that the density reduction across multiple layers in the central region negatively affects the layers' bondage. The

Attack Types	Peak Flexure Stress (MPa)		Peak Flexure Strain %)		Peak Stress / Strain (MPa)	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
No attack	71.814	1.477	11.190	0.033	21.618	0.663
Clean Cavity	70.403	1.477	11.043	0.061	21.151	0.491
Filament State	65.283	1.712	11.184	0.064	19.735	0.347
Filament Density	69.361	0.484	11.152	0.062	20.881	0.112

Table 4.: Three-point bending tests summary for filament-kinetic attacks

three-point bending results indicate apparent changes in the object properties based on the attack pattern.

2.6.2 Mechanical testing of dynamic-thermal attacks

2.6.2.1 Tensile Test

Tensile tests for dynamic-thermal attacks also show a noticeable change in properties. Figure 12a shows that the benign samples break at random locations, as expected. However, a consistent interesting trend is visible with the two attack patterns. The low-temperature attack samples, shown in Figure 12b, always break at the center where the temperature deviation is maximum, while the high-temperature attack samples, shown in Figure 12c, never break at the center.

Compared with the default profile, the peak stress value was reduced by -8.1% for the low-temperature attack and -3.3% for the high-temperature attack. The most impacted property was 'Strain' with -28% difference in low-temperature profile. Table 5 presents the summary of the test results. Figure 13a and 13b presents tensile test results plots for stress versus strain, and load versus time.

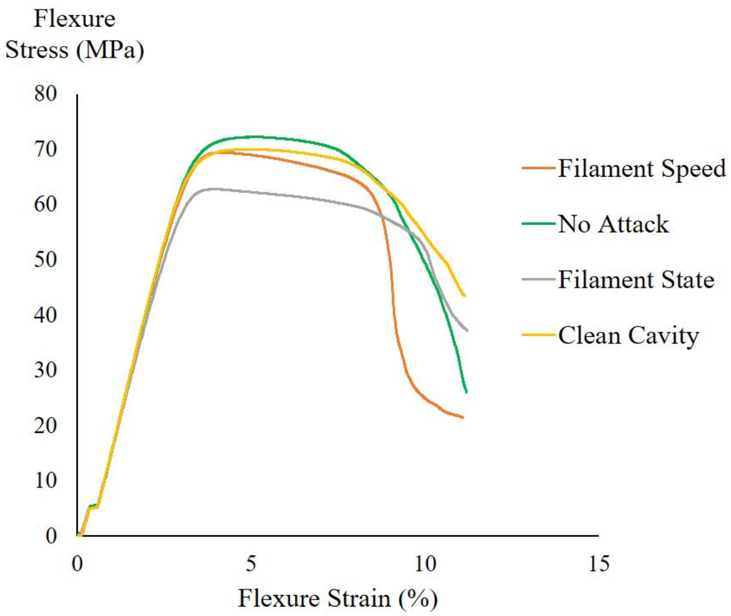
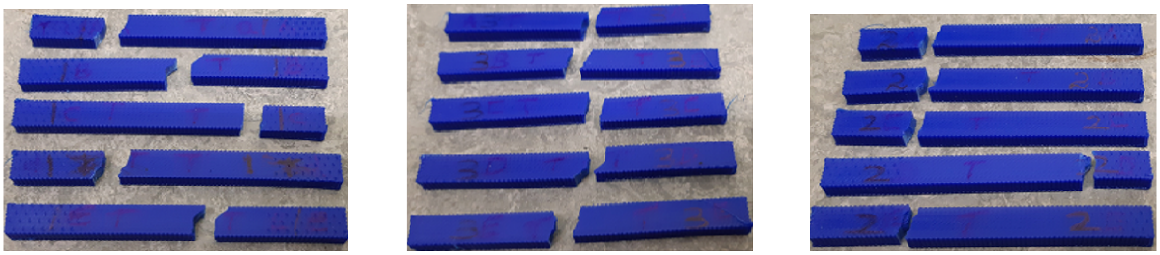


Fig. 11: Three-point bending test for filament-kinteic attacks: flexure stress vs flexure strain



(a) Default temperature profile: sample breaks at random locations

(b) Low temperature profile: specimen always break at the center

(c) High temperature profile: specimen never break at the center

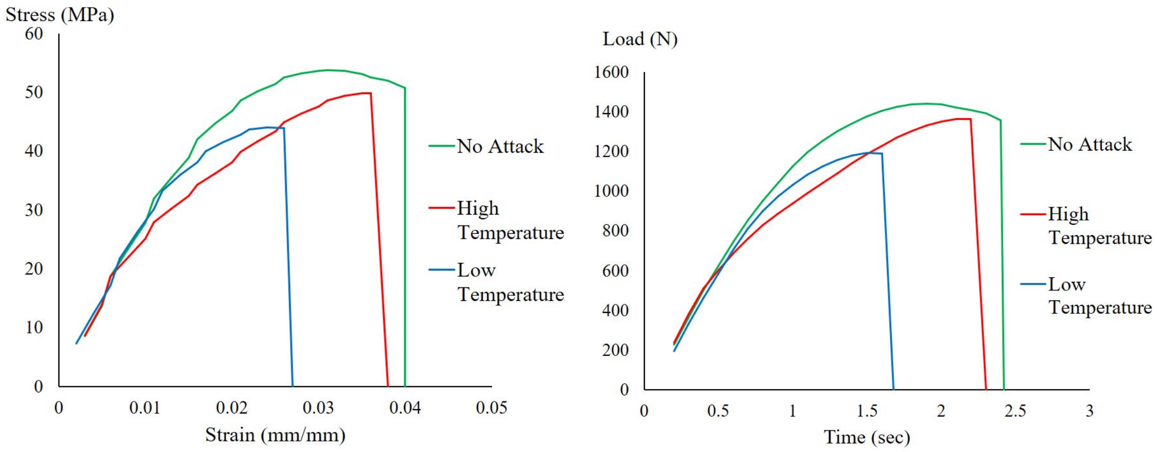
Fig. 12: Dynamic-thermal attacks specimen images highlighting the breakpoints

Attacks	Peak Load (N)		Peak Stress (MPa)		Strain at Break (mm/mm)		Modulus (MPa)	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
No Attack	1356.438	72.806	50.800	2.593	0.032	0.002	2939.383	223.311
High Temperature	1329.316	73.773	49.120	2.594	0.033	0.005	2845.408	259.791
Low Temperature	1268.334	43.458	46.820	1.617	0.023	0.001	2962.007	66.050

Table 5.: Tensile tests summary for dynamic-thermal attacks

Attacks	Peak Flexure Stress (MPa)		Peak Flexure Strain(%)		Peak Stress / Strain (MPa)	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
No Attack	80.209	4.344	10.953	0.105	7.322	0.370
High Temperature	88.288	1.885	11.003	0.036	8.025	0.191
Low Temperature	70.842	3.375	11.067	0.090	6.403	0.327

Table 6.: Three-point bending tests summary for dynamic-thermal attacks



(a) Stress vs Strain

(b) Load vs Time

Fig. 13: Tensile test results for dynamic-thermal attacks

2.6.2.2 Three-point bending test

Figure 14 represents the three-point bending test results for dynamic-thermal attacks. For this test, we set a maximum extension limit of 7.5 mm. All specimens were fractured before this value. For high-temperature attack samples, the specimen breaks abruptly, indicating strong inter-layer bondage in the central part. Though the specimen gets fractured earlier for low-temperature attacks, several layers remain intact till the max extension limit for all the low-temperature attack specimens. It shows the weakening of the inter-layer bondage caused by temperature reduction. The peak load value raised from 144 N (default) to 158.8 N for the high-temperature profile but reduced to 129.4 N for the low-temperature profile indicating 10 to 15% average deviation. The test results are presented in Table 6 and Figure 14.

2.7 Attack Countermeasures

The attack countermeasures can be categorized into groups: cyber and physical domain measures.

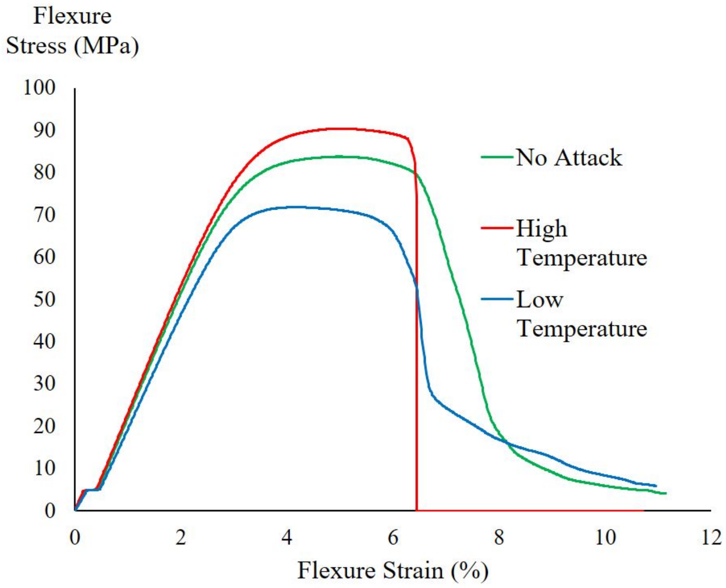


Fig. 14: Three-point bending tests for dynamic-thermal attacks: flexure stress vs flexure strain

2.7.1 Cyber-domain Countermeasures

The proposed attacks can be launched by modifying the G-code file sent to the printer, or by compromising the printer firmware. The following paragraphs discuss some countermeasures in the cyber domain to detect and block these attacks.

Network layer security McCormack et.al. [28] identify most of the surveyed printers using unencrypted communication with the control PC, increasing the chances of network layer attacks. By securing the communication channel between the control PC and the printer through advanced encryption standards and authentication techniques, the network attack vector can be controlled.

Firmware attack countermeasures Firmware of a 3D printer can be modified through an illegal upgrade activity over the network or via USB port. Network security measures and vulnerability analysis of the firmware cover the known remote exploits. To avoid an illegal USB-based firmware upgrade, physical access should be controlled, and USB drives should be regularly scanned for malware. To verify the firmware of the printers, users may utilize verification schemes proposed by researchers, such as [29] based on block-chain, [30] using instructions level abstraction, etc.

2.7.2 Physical domain countermeasures

Monitoring the physical process is an important measure to detect cyberattacks in a CPS.

Realtime and out-of-band monitoring of filament-kinetics. The test results show that the presented sabotage attacks change the physical properties in various ways and magnitudes. Cavity attacks through filament-kinetics created a clean cavity with a minimal footprint over the nozzle kinetics. The filament-state attack has zero footprint over nozzle-kinetics but a relatively bigger footprint over filament-kinetics. If the filament-kinetic state is monitored, this attack can be detected. The filament-speed attack is slow and steady. There is no footprint over nozzle-kinetics. If the magnitude of density variation is low enough, the detection could be challenging. More research can reveal new ways to launch and block these attacks. Schemes based on optical encoders [31] or electric current measurement [19] can be utilized effectively with more research. As all the attacks incurred unique effects on the object, an interesting future study can provide unique attack signatures. These signatures can help in detection when the attack magnitudes go further low into the confusion zone of the printing process' benign deviations.

Monitoring via temperature sensors or thermal cameras The two dynamic-thermal attack patterns used in our scheme successfully modified object properties confirming that these attacks are practical. Very low-temperature deviations do not affect the object, while very high changes can cause visual deformity. Within this window, different attacks can be launched to target specific properties of the object. Monitoring the nozzle temperature through thermocouple sensors or object temperature using thermal cameras [32] can help in detecting these attacks.

2.8 Conclusion

The chapter presented localized filament-kinetic and dynamic-thermal attacks on the FFF-based 3D printing process. The attacks produce no visual impairment and insignificant footprint, making them difficult to detect. With the help of tensile and three-point bending tests, we established that these attacks successfully modify the printed object's physical properties, such as peak stress and strain. The attacks that target the design or STL files usually cause big and uncontrolled printing profile changes at the slicing stage, making them obvious to simple detection schemes. On the contrary, the proposed attacks can bypass the existing detection techniques, and can still impair the normal functioning of the targeted object. To protect against these attacks, synchronized space-time analysis of the thermodynamic, nozzle-kinetic and filament-kinetic processes can be an effective way.

CHAPTER 3

SABOTAGING MATERIAL-EXTRUSION-BASED 3D-PRINTED OBJECTS THROUGH LOW-MAGNITUDE KINETIC ATTACKS IN THE SLICER PROCESS MEMORY

The increasing ubiquity of material-extrusion-based additive manufacturing is motivating cybersecurity researchers to explore its offensive and defensive landscape. Being a physical system, 3D printers have non-zero tolerance specifications for precision and trueness parameters. While a single-bit change in a digital data file is sufficient to fail its integrity and is easily detected through methods such as hashing, the printing process (and subsequently the printed object) remains compliant within the tolerance zone. Current state-of-the-art attack detection schemes do not detect malicious anomalies within the tolerance zone, offering attackers a small opportunity window to sabotage the process. This study systematically analyzes and identifies four attacks in the material extrusion process where kinetic manipulations within the printer's tolerance values degrade the part's mechanical properties. Attacks are launched through a man-in-the-middle attack scenario by hijacking the network layer communication between the 3D printer and the printer control machine. The performance of the attacks is evaluated using ASTM-compliant tensile and flexure bars. While no evident deformations are observed in the printed part dimensions and mass, the destructive tests confirm that the proposed low-magnitude attacks are effective in modifying the tensile and bending strength by up to 25% for various attack magnitudes. [1]

[1] *This work is an extension of my paper presented at the 16th IFIP WG 11.10 International Conference, ICCIP 2022, Virtual Event, March 14–15, 2022 [33]*

3.1 Introduction

Additive manufacturing methods are being commonly used in various industrial sectors including Aviation, Automobile, and Healthcare. Due to its compelling advantages such as a faster development cycle, mass customization and complex object printing capability, Industry 4.0 considers additive manufacturing as an essential component [5]. Material extrusion is the most common of the seven additive manufacturing methods defined in ISO/ASTM standard 52900 [34]. Material extrusion based fused filament fabrication (FFF) technology is anticipated to attract more attackers after the incorporation of metal-infused filaments [35]. In response to the increased incentive to attackers, cybersecurity researchers have proposed various techniques to secure the FFF-based printing process [36, 37]. In addition to conventional cybersecurity solutions, researchers have utilized physical domain knowledge to detect attacks and anomalies. Monitoring the physical process through side channels offers better coverage of the cyber-physical process as compared to conventional cyber-domain monitoring. For example, if an attacker hijacks the network communication and manipulates the G-code file sent to the printer, a physical process monitoring solution shall still detect such an attack.

Monitoring the printing process in the physical domain has its own challenges. The performance of a physical process monitoring solution depends on the quality of the sensing equipment, deployment proficiency, algorithmic errors, and environmental factors (such as changing background sound and lighting conditions). It is an active research area and the literature review shows that the detection horizon is continuously improving. For instance, Rais et al. [31] claimed to reliably detect a 1 mm deviation in the toolpath with zero false positives and false negatives in a set of objects. As the detection horizon improves, it will overlap with the printer spec-

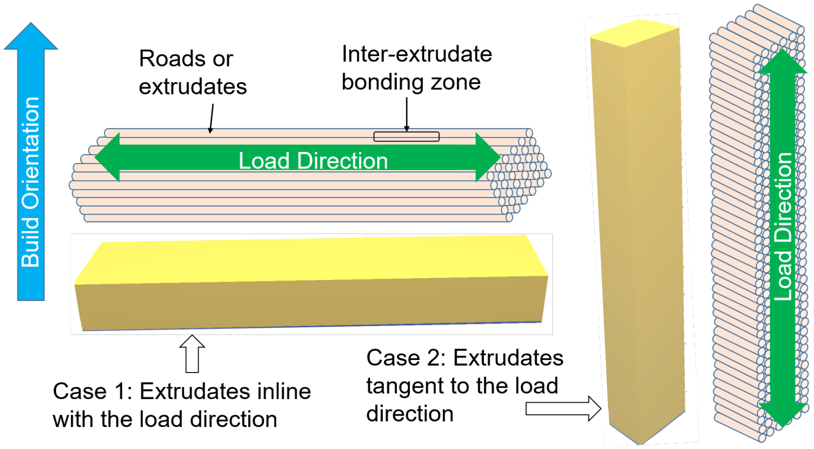

Build Orientation
Roads or extrudates
Inter-extrudate bonding zone
Load Direction
Case 1: Extrudates inline with the load direction
Case 2: Extrudates tangent to the load direction
Load Direction

Fig. 15: Impact of FFF characteristics on the mechanical strength explained through an example of changing build orientation

ifications tolerance zone. Unlike a digital artifact where a single-bit change is also not acceptable, a physical process is considered compliant within the tolerance zone. Even if a monitoring scheme is capable of detecting tiny deviations, reducing the anomaly threshold below the printer's trueness value will likely result in a significant increase in false positives.

If a smart attacker keeps the attack magnitude within the tolerance of the printing process, the attack can likely circumvent the threshold-based detection schemes. One may hypothesize that if the process is progressing within the specified green zone, there should be no reason to worry. It is not ascertained if these low-magnitude deviations can consistently and negatively influence the printed parts' mechanical properties. As the printers are not designed nor have their specifications been finalized after considering the impact of machine deviations on different printed objects, it is reasonable to doubt the above hypothesis. FFF Characteristics also play an important role in the mechanical properties of the printed object. However, all the characteristics may not offer a good opportunity for an attacker. For instance, changing the build orientation, as presented in Figure 15, significantly reduces the tensile strength but completely changes the toolpath (the printing sequence) making it a simple-to-detect attack.

In this study, the authors examine the FFF printing process to identify minimal kinetic manipulation opportunities (within the printer specifications tolerances) targeting the extrudates bonding to degrade the tensile and/or flexure strength of the printed parts. The chapter presents four attacks exploiting extrudates bonding at critical locations. The first two attacks relate to inducing bonding weakness within the infill structure. The third attack attempts to weaken the bonding between the infill and the wall structure. The fourth attack exploits the printing bed kinetics to manipulate the interlayer bond. This study uses a man-in-the-middle attack vector to inject the proposed attacks into the printing process by hijacking the G-code file in flight.

To evaluate the effectiveness of the proposed attacks, an experiment is designed with attack magnitudes ranging from 0.015 mm to 0.2 mm. Tensile and three-point bending tests are conducted for the attacked and non-attacked samples to measure the attacks' impact on the tensile and flexure strengths. The results confirm that planned malicious deviations within the above-mentioned range are sufficient to compromise the mechanical strength of the printed parts.

3.2 Related Work

The researchers have demonstrated sabotage attacks by inducing defects at the designing stage of the printing process [12, 38, 13]. A few other researchers have explored attacks on the G-code instructions (post-designing stage), either through MiTM between the control machine and the printer [8] or by manipulating the slicer memory [39]. The literature also shows limited effort in manipulating the printer's firmware or bootloader to inject defects in the printed parts [40]. Rais et al. [33]

proposed two low-magnitude attacks targeting infill structure to reduce the part's strength. Instead of restricting to infill structure, this work systematically examines all sub-structures and inter-substructure bonds to identify low-magnitude attack opportunities. This work does not relate to cavities or thermodynamic attacks, it only includes low-magnitude kinetic attacks targeting the bonding between adjacent extrudates.

The aim of this study is to find process deviations that can fly below the detection horizon, we discuss the existing attack detection schemes. Chhetri et al. [21] proposed the use of audio emissions to detect kinetic anomalies in the print object. Belikovetsky et al. [20] used fingerprinting method to authenticate the printed part by generating a master audio profile and using it for the next printed parts. A similar technique was adopted by Gatlin et al. [19] wherein instead of using audio, electric current signals were used to generate a master profile and compared with in-process signals. A deviation beyond the threshold was categorized as anomalous behavior. Gao et al. [14] acquired data through Inertial Measurement Unit (IMU) sensors and cameras. Using mathematical modeling and image processing they were able to detect significant geometry distortions due to cooling process anomalies. Wu et al. [41, 42] employed static and moving cameras technique to capture and train images on the machine learning algorithm to detect infill deviations in the print geometry. Rais et al. [31] adopted a multi-sensing technique and utilized optical encoders and thermal sensors to accurately estimate the printing state. The proposed framework, *Sophos*, transforms G-code instructions through spatiotemporal modeling and compares it with the sensor values. Instead of waiting for the printing to complete, the framework continuously examines the printing process and effectively detects attacks after each layer is printed. We use these studies to estimate attack detection's state of the art.

3.3 Methodology and the proposed attacks

This section first presents the criteria formalized for a successful attack, followed by the existing attack detection horizon identified through literature. Then the precision and trueness values of common FFF printers are reviewed to identify the limits of attack magnitude. Once these constraints are established, compliant attack opportunities in the FFF process are highlighted. Finally, the four proposed attacks are described in the section.

3.3.1 Defining success criteria for the proposed attacks

This study hypothesizes that malicious low-magnitude kinetic deviations can noticeably and consistently modify a printed part's mechanical properties. In this context, low-magnitude deviations are considered as the ones that are (1) below the existing attack-detection horizon, and (2) within the printer specifications tolerances. The success of the proposed attacks is therefore based on the validity of the hypothesis, examined using the following criteria.

1. Attacked parts should statistically maintain the shape, dimensions, and weight
2. Attacks should be able to evade the existing state-of-the-art detection schemes discussed in Section 3.2
3. Mechanical strength of the attacked parts should be consistently reduced
4. Resultant deviations in the printed parts should remain within the tolerance of a typical FFF printer's specifications

This study only focuses on attacks causing kinetic process deviations in order to influence the bond between adjacent extrudates. All thermodynamic attacks and kinetic attacks that do not target extrudates bonding, such as introducing a big cavity within the infill structure, are excluded from the scope. An important reason for not exploring these attacks is their non-compliance to criterion 4 enumerated above.

3.3.2 Existing attack detection horizon

The best results in detecting process deviation reported in the literature presented in Section 3.2 are 1 second per layer for timing profile, 0.05 mm for layer thickness, 1 mm^2 single area mismatch with at least 0.3 mm length per axis, and 5^oC variation in the nozzle and printing bed thermal profile. These values constitute the current detection horizon for attacks on the FFF process.

3.3.3 Precision and trueness values of common FFF printers

A few vendors have reported the precision and trueness values of their printers. Stratasys conducted a study on its printers and reported 130 μm tolerance in the true value observed for 95% parts printed through Fortus 360mc/400mc printers [43]. A few independent researchers have studied the dimensional accuracy of 3D printers. Kim et al. conducted a study on the precision and trueness of dental models printed through some of the available printing methods [44]. For the FFF printers used in the study, they observed the precision and trueness as $99 \pm 14\mu m$ and $188 \pm 14\mu m$, respectively. Msallem et al. measured the precision and trueness of an Ultimaker 3 Ext FFF-based printer as $160 \pm 9\mu m$ and $50 \pm 5\mu m$, respectively [45]. These studies indicate a zone of confusion that can be exploited by an attacker. Low-magnitude variations in the order of the above-reported values are likely to evade the attack detection systems, labeling them as the expected behavior of the printer.

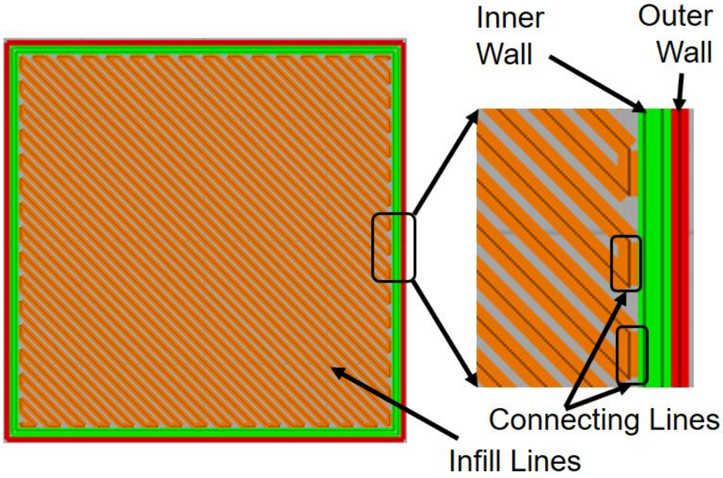


Fig. 16: Components of an internal layer in FFF printing

3.3.4 Examining FFF process for available attack opportunities

Figure 16 presents a single internal layer of a cube sliced for a FFF printer. Two main components in an internal layer are walls and an infill structure. The density and the type of infill pattern influence the strength of a printed part [46]. For a solid load-bearing part, a common choice of infill pattern is 'lines' or 'rectilinear'. As presented in Figure 16, two infill lines are connected through connecting segments. These connecting segments play an important role in the context of extrudates bonding. The length of the connecting segments for solid parts is proportional to the nozzle diameter (typically a fraction of a millimeter). Manipulating the placement and the size of these connecting segments offers opportunities for low-magnitude attacks.

As the molten filament is extruded from a printer's nozzle, it either interacts with the printing bed or with the already extruded filament. Heat energy from the latest extrusion is used in melting (wetting) a small part of the existing filament in its proximity. The process is referred to as interdiffusion.

Let $P_{x,y,z}$ be the current location of the nozzle in 3D space during printing, where (x,y) represent its location in the raster plane from a reference vertex on the printing bed, and 'z' represents the relative distance between the nozzle and the printing bed. The interdiffusion zone $D.Z_{P_{x,y,z}}$ around $P_{x,y,z}$ can be represented as a set of pixels in 3D space around $P_{x,y,z}$ approximated by Equation 3.1

$$\begin{aligned} D.Z_{P_{x,y,z}} = \{P_{i,j,k} \mid \mathbf{f}(P_{i,j,k}, P_{x,y,z}) \geq \mathbf{e}(T_{P_{i,j,k}}) \\ and\ t_{P_{i,j,k}} \leq t_{P_{x,y,z}} \quad \} \end{aligned} \tag{3.1}$$

where $\mathbf{f}(P_B, P_A)$ estimates the influence of heat energy of currently extruded pixel P_A over any pixel P_B in 3D printing space, $\mathbf{e}(T_a)$ is the energy required to raise the temperature of a unit volume at temperature T_a to the glass transition temperature, and t_{P_A} represents the time at which the filament is extruded at pixel P_A. Analytical derivation of these functions is complicated and influenced by multiple factors including temperature of the nozzle and printing bed, speed of cooling fans, environmental conditions, printing speed, shape of the geometry, material thermal properties, etc. As our aim is to identify kinetic attack opportunities with attack magnitudes within the order of the printer's specifications, our zone of interest is practically restricted to the adjacent extrudates only. To ascertain the impact, the authors measure the strength of the attacked parts through destructive mechanical tests, including tension and flexure strength tests.

3.3.5 Proposed attacks

Analysis of the printing operation at any instance in time shows that the most recently extruded filament interacts with the existing extrudates belonging to the wall or infill structure of the same layer and the adjacent layers. The following subsections present 4 feasible attack opportunities to successfully sabotage the printed parts.

3.3.5.1 Attack 1: Infill lines spacing attack

Bonding between two spatially adjacent infill lines influences the overall strength of a solid part. In this attack, two consecutive extrudates from infill lines are separated by increasing the length of the connecting segment by a small fraction. Figure 17 presents one instance of this attack. The attacked connecting segment length d_a is increased by $\Delta\, d_a$, which is a fraction of the original segment length d_o. The length of two adjacent connecting segments d_{c_1} and d_{c_2} is reduced by Δd_{c_1} and Δd_{c_2}, respectively. Equation 3.2 and 3.3 presents the relationship and constraints of the attack variables. K_s ranging from 0 to 1, is the stealth factor against any visible deformation.

$$0 < \left\{ \begin{array}{l} \Delta\, d_{c_1} = d_o - d_{c_1} \\ \Delta\, d_{c_2} = d_o - d_{c_2} \end{array} \right\} < \; (1 - K_s) \, * \, d_o / 2 \tag{3.2}$$

$$\Delta\, d_a = \Delta\, d_{c_1} + \Delta\, d_{c2} \; = \; d_a - d_o \tag{3.3}$$

3.3.5.2 Attack 2: Infill vertices spacing attack

This attack also targets the bonding between consecutive infill extrudates within a layer. Instead of reducing the overlap across the two consecutive infill lines, the attack manipulates only one edge of the targeted part as presented in Figure 18. By reducing the length of two consecutive connecting segments at one edge, an inverse wedge is produced. Depending upon the attack magnitude, the attack may result only in overlap reduction (for smaller magnitudes) and a visible inverse wedge (for higher magnitudes). As the attack targets internal layers only, it is concealed in the final part for all magnitudes. The attack only manipulates the vertices of the connecting segments at one edge. This attack causes a minimal deviation in the local raster angle and the length of the two consecutive infill lines involved in the attack. Equation 3.4

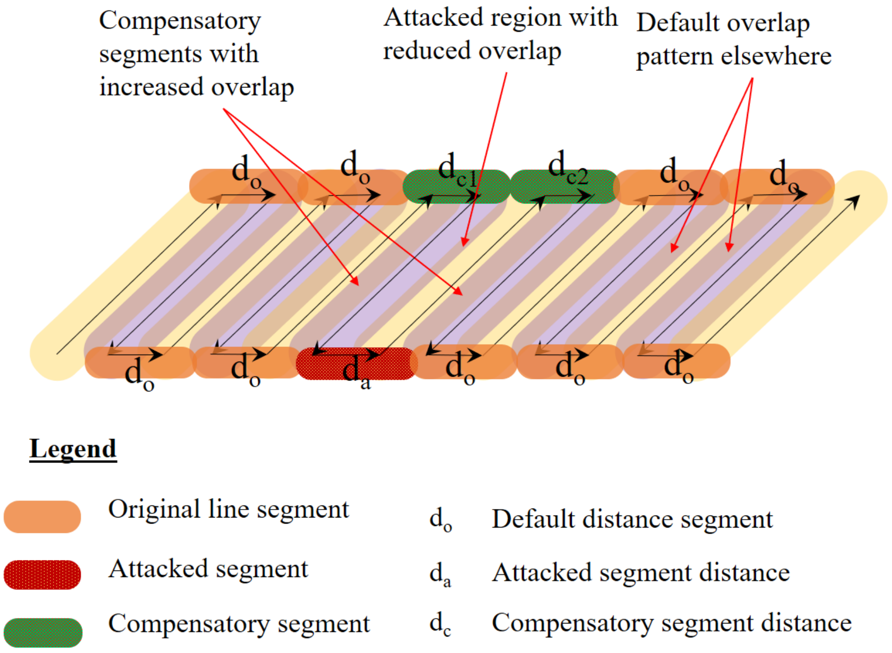

Compensatory segments with increased overlap
Attacked region with reduced overlap
Default overlap pattern elsewhere
d_o
d_o
d_c1
d_c2
d_o
d_o
d_o
d_o
d_a
d_o
d_o
d_o
Legend
Original line segment
Attacked segment
Compensatory segment
d_o Default distance segment
d_a Attacked segment distance
d_c Compensatory segment distance

Fig. 17: Infill lines spacing attack representation

represents the change in the length of the infill-lines, and Equation 3.5 represents the change in the raster angle,

$$d_{IF_a} = \sqrt{d_{IF_o}^2 - 2 * \Delta d_s * sin(\theta_o) * d_{IF_o} + \Delta d_s^2} \tag{3.4}$$

$$\theta_{IF_a} = tan^{-1}\frac{(d_{IF_o} * sin(\theta_o) + \Delta d_s)}{d_{IF_o} * cos(\theta_o)} \tag{3.5}$$

where d_{IF_a} and d_{IF_o} are the modified and the original length of the infill lines, Δd_s is the difference between the original connecting segment length d_o and the modified length d_a, and $\theta_o\ and\ \theta_a$ represent the default and the modified raster angles. Considering a 15 mm infill line configured at a raster angle of 45^o, a 0.1 mm decrease in the connecting segment length will result in Δd_{IF} (change in infill line length) of around 0.07 mm, and the change in raster angle of 0.2 ^{o}. These minimal changes are within the printer tolerances and beyond the capability of the existing attack detection schemes. The attack can be accomplished using different values of d_{a_1} and

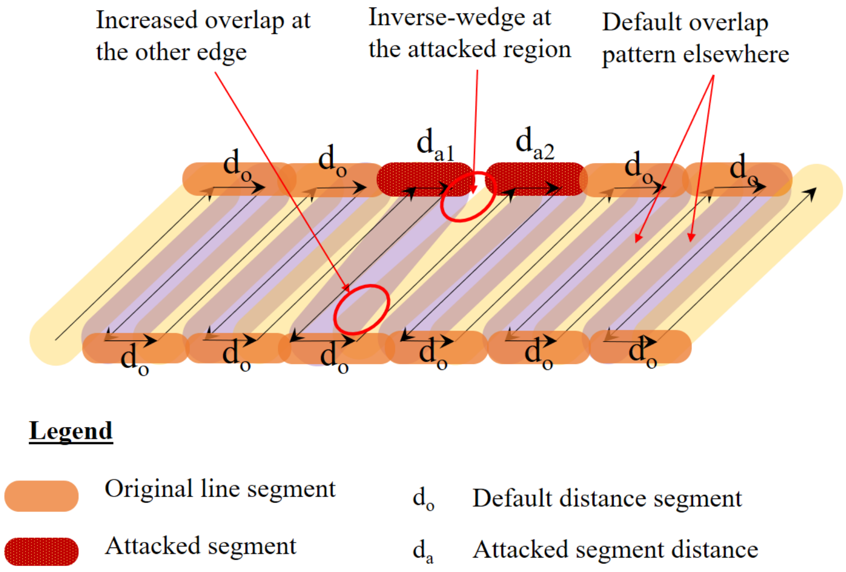

Increased overlap at the other edge
Inverse-wedge at the attacked region
Default overlap pattern elsewhere
d_o
d_a1
d_a2
Legend
Original line segment
Attacked segment
d_o Default distance segment
d_a Attacked segment distance

Fig. 18: Infill vertices spacing attack representation

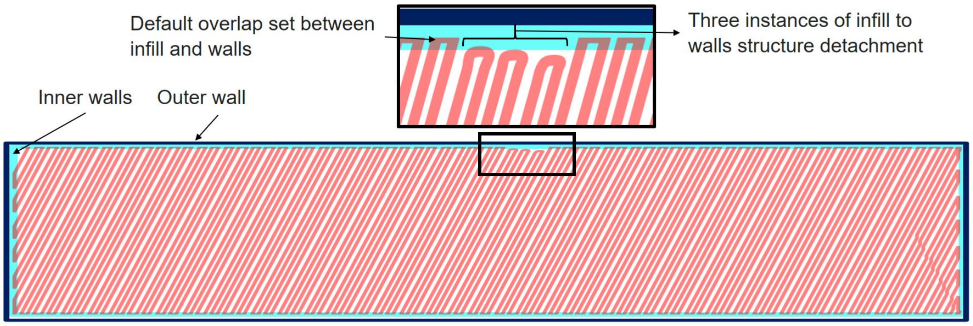

Default overlap set between infill and walls
Three instances of infill to walls structure detachment
Inner walls
Outer wall

Fig. 19: Infill structure to walls bonding attack

d_{a_2}. Interestingly, the changes in IF_1 and IF_2 have opposite polarity. If one decreases, the other increases, and vice versa. As the attack instances are launched over multiple layers, this polarity reversal helps in canceling out (instead of accumulating) the difference in the original and the attacked printing profile, making it more challenging for the attack detection schemes. Algorithm 1 outlines the attack process for both attacks.

3.3.5.3 Attack 3: Infill and wall structure bonding attack

Unlike the previous two attacks, this attack targets fusion between the infill structure and the walls. Slicer software offers a choice to print the infill before or after the internal walls. In either case, these two constituents of internal layers are temporally displaced. If the infill is printed first, the later printed extrudate of the internal wall will interact with the infill structure creating a bond by interdiffusion. This attack manipulates the bonding strength between the infill and wall structure at the point of attack by reducing the overlap between the two regions. Figure 19 presents a typical attack with three instances of varying magnitudes increasing from left to right. Each attack instance is executed by modifying the end vertices of two consecutive toolpath instructions; the first instruction prints the preceding infill line and the second one prints the targeted connecting segment. The length of the connecting segment is not changed, while the infill line segment length is decreased by the magnitude of the attack (typically a small fraction of a millimeter).

3.3.5.4 Attack 4: Inter-layer bonding attack

As the filament is extruded out of the nozzle, it also interacts directly with the material from the previous layer. The impact of interlayer bonding on object strength is a well-researched topic [47]. This attack induces interlayer bonding weaknesses in the printed part. After printing a layer if the bed is lowered more than the designed value without increasing the filament flow rate, the transferred heat and the pressure exerted by the new extrusion on the existing layer are reduced. For instance, if the printing bed is lowered by 0.2 mm for the n^{th} layer against the designed layer thickness of 0.1 mm, the bonding between nth and (n-1)th layer will be poor. However, it creates an obvious mark on the sides of the part. To conceal the poor bonding mark, this

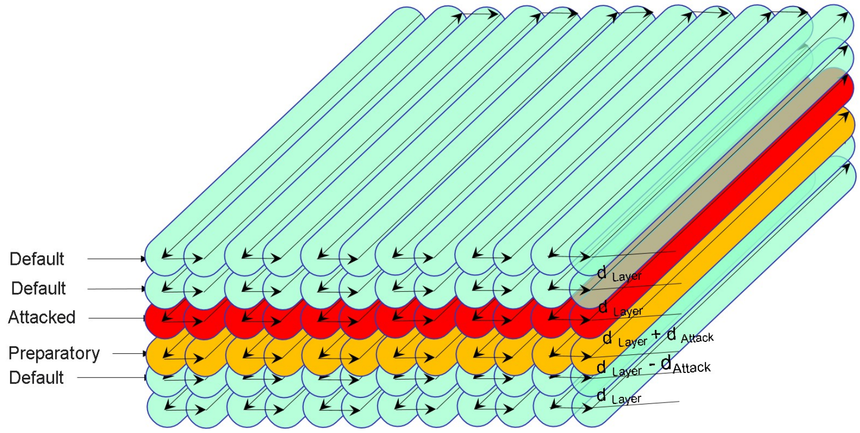


Fig. 20: Manipulation of inter-layer distance for the infill structure

attack exploits the interlayer bonding for the infill structure only. The walls structure is still printed using the default profile. Figure 20 elaborates an attack scenario showing the infill structure. The layer thickness of the attacked layer (say nth) layer is modified to $d_{layer} + d_{attack}$ where $0 < d_{attack} < d_{layer}$. The increase in layer thickness for the nth layer is pre-compensated at (n-1)th layer by reducing the layer height for the infill structure to $d_{layer} - d_{attack}$. One may argue that the attack may incur some deformation in the layers other than the attacked ones. While it is plausible, it is not a concern for the attacker. If the compensatory moves are within the attack detection thresholds and do not cause obvious deformation, they do not conflict with the attack success criteria. The attack can be more effective against parts with walls printed prior to the infill. The inner edge of the wall structure will partially hold the elevated infill lines resulting in less pressure and transfer of energy to the lower layer.

Algorithm 1 Low-magnitude extrudates bonding kinetic attacks

```
**Input:** Network traffic b/w printer and controller
**Output:** G-code$_{Attacked}$
**Attack$_{\textbf{param}}$** : { A$_{No}$ , A$_{Mag}$ , A$_{Loc}$ , A$_{Layers}$ }
Launch ARP Poisoning Attack
Sniff printer - controller communication
**if** Controller sends G-code to printer **:**
    Extract G-code file → G-code$_{Original}$
    G-code$_{Attacked}$ ← **Attack[A$_{No}$]-function**(Attack$_{param}$ , G-code$_{original}$)
Send G-code$_{Attacked}$ to printer via MiTM
Manage communication
**Attack-1-function**(Attack$_{param}$ , G-code$_{original}$) **:**
*(Infill lines spacing attack*)
while A$_{Loc}$ $\notin$ Infill-structure:   shift A$_{Loc}$
$\forall$ i $\in$ A$_{Layers}$ :
    s$_1$← Search nearest connecting segment to A$_{Loc}$
    Calculate new x and y coords, such that :
        No change in the slope for any infill or segment
        $|d_{s_1}|$ ← $|d_{s_1}|$ - $|A_{Mag}|$ ;        $|d_{s_2}|$ ← $|d_{s_2}|$ + $|A_{Mag}|$
        $|d_{s_3}|$ ← $|d_{s_3}|$ - $|A_{Mag}|$ ;        No change in $|Infill_1|$ & $|Infill_2|$
     $\forall$ j $\in$  Attacked commands :
        Compute new G-code(j)
        Update G-code(j) in G-code$_{Attacked}$
**return** G-code$_{Attacked}$
```

```
**Attack-2-function**(Attack$_{param}$ , G-code$_{original}$) **:**
*(Infill vertices spacing attack)*
while A$_{Loc}$ $\notin$ Infill-structure:   shift A$_{Loc}$
$\forall$ i $\in$ A$_{Layers}$ :
    Calculate new x and y coords, such that:
        No change in the slope of segments (slight change for infill lines)
        $|d_{s_1}| \leftarrow |d_{s_1}|$ - $|A_{Mag}|$ ;        $|d_{s_2}|$ not modified
        $|d_{s_3}| \leftarrow |d_{s_3}|$ - $|A_{Mag}|$ ;  (Infill-lines magnitude will slightly change)
    $\forall$ j $\in$  Attacked commands :
        Compute new G-code(j)
        Update G-code(j) in G-code$_{Attacked}$
**return** G-code$_{Attacked}$
**Attack-3-function**(Attack$_{param}$ , G-code$_{original}$) **:**
*(Infill to wall structure bonding attack)*
while A$_{Loc}$ $\notin$ Infill-structure:   shift A$_{Loc}$
$\forall$ i $\in$ A$_{Layers}$ :
    $\forall$ j $\in$ A$_{instances}$ **:**
        Calculate new x and y coords, such that
        No change in slope for infill lines or connecting segments
        $|d_{IF_1}| \leftarrow |d_{IF_1}|$ - $|A_{Mag}|$
        $|d_{IF_2}| \leftarrow |d_{IF_s}|$ - $|A_{Mag}|$
        $|d_{s_1}|$ or $|d_{s_2}|$ not modified
        Add $IF_1$ and $S_1$ in Attacked-commands list
    $\forall$ k $\in$  Attacked-commands :
        Compute new G-code(k)
        Update G-code(k) in G-code$_{Attacked}$
**return** G-code$_{Attacked}$
```

```
**Attack-4-function**(Attack$_{param}$ , G-code$_{original}$) :
*(Interlayer bonding attack)*
$\forall$ i $\in$ A$_{Layers}$ :
    Identify limits of Infill structure - [IF$_0$, IF$_n$]
    Identify IF$_{st}$ and IF$_{end}$  $\ni$  $0 \leq$ st  $<$ end  $\leq$ n
    if index(i) is even :
        Z$_{attk}$ $\leftarrow$ Z$_{current}$ - A$_{mag}$
    else:
        Z$_{attk}$ $\leftarrow$ Z$_{current}$ + A$_{mag}$
    Append Z = Z$_{attk}$ to G-code(i,st)
    Append Z = Z$_{current}$ to G-code(i,end)
    Update G-code(i,st) & G-code(i,end) in G-code$_{Attacked}$
return G-code$_{Attacked}$
```

3.4 Experimental details

3.4.1 Attack vector

Man-in-the-Middle (MiTM) is one of the most effective adversarial tools to intercept and manipulate legitimate communication between two clients. An attacker having access to the local network where the printing environment is set up could position themselves in-between printer and user (control PC) and can eavesdrop on the network traffic amongst these parties. The traffic between these two entities is not encrypted making it vulnerable to such attacks. The attacker uses ARP poisoning to initiate a MiTM attack. Inhere the attacker associates own MAC address with the IP address of the 3D printer and start proxying the network traffic intended for the

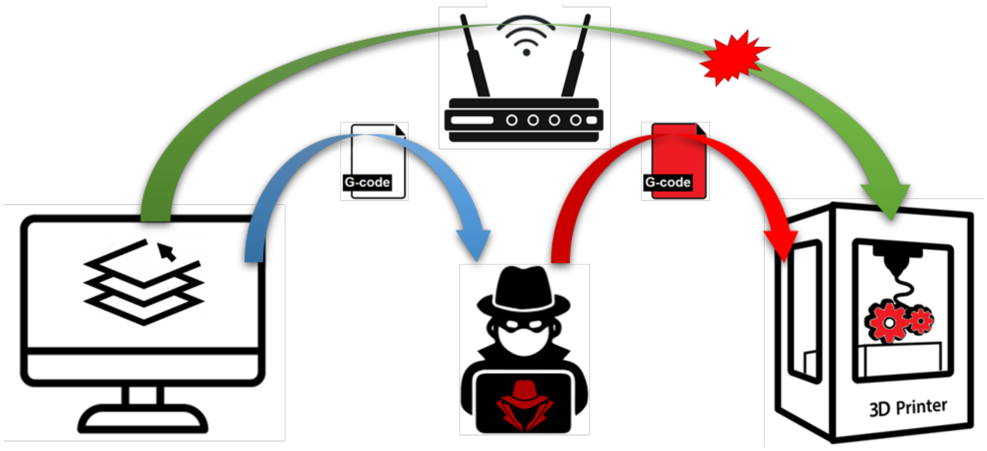


Fig. 21: MiTM attack to manipulate G-code file

printer, as shown in Figure 21.

Slicer software running on the control PC creates the G-code that is being communicated over the network to the 3D printer. Having control of the communication, the adversary can now intercept the G-code commands and maliciously manipulate them without giving an indication of the communication integrity being compromised to any of the legitimate entities.

3.4.2 Experimental settings

This section details the specification of the FFF printer, printing parameters, and the specimens used for the experiment. The effect of the proposed attacks on tension and flexure strength is examined on ASTM D638 Type IV standard tensile bars and ASTM D790 compliant flexure bars, respectively. The overall dimensions of the tensile bar are $115 \times 19 \times 4.07$ mm $(length \times width \times thickness)$, with the central part 6.5 mm in width. Due to the test equipment having a maximum span of 41 mm, a smaller thickness specimen is used for the flexure test with dimensions $76.8 \times 12.7 \times 2.4$ mm. The dimensions, however, comply with the standard's requirement to maintain a span-to-thickness ratio of 16. All the parts were printed with Polylactic Acid (PLA) polymer using an FFF-based printer - Ultimaker 3. The printer is connected over the local area network (LAN) to the control machine hosting Windows 10 operating system and running Cura version 4.10 slicer application. The printing profile used by slicer software during the experiment is presented in Table 7. As Attack 3 targets the bonding between the infill structure and the walls structure, the number of walls is increased from 2 to 4 in an attempt to make the walls' structure strength comparable to the infill structure. As Attack 4 deals with layer thickness manipulation of the infill structure, the interlayer bond degrades more if the walls are available to hold the infill raster.

Table 7.: Printing parameters selected for the experiment

S/No	Printing parameter	Selected value
1	Layer thickness	0.2 mm
2	Nozzle diameter	0.4 mm
3	Build plate temperature	60^{o}C
4	Nozzle temperature - Layer 1	210^{o}C
5	Nozzle temperature -Layer 2 onwards	205^{o}C
6	Infill pattern	LINE at 45^o
7	Infill percentage	100%
8	Infill overlap with walls	20%
9	Number of layers for tensile specimens	20
10	Number of layers for flexure specimens	12
11	Bottom layers	2
12	Top layers	Nil
13	Printing speed for initial layer	20 mm/sec
14	Printing speed for top/bottom layers	45 mm/sec
15	Infill printing speed	70 mm/sec
16	Walls printing speed (outer/inner)	50/55 mm/sec
17	Number of walls in Attack 1,2,4	2
18	Number of walls in Attack 3	4
19	Printing sequence for Attack 1, 2 & 3	Infill first
20	Printing sequence for Attack 4	Walls first

3.4.3 Design of experiment

Each of the attacks is implemented for a range of magnitudes to identify the strength reduction trend using statistical parameters including mean and standard deviation. Table 8 outlines the design of the experiment. Kinetic manipulations in the first three attacks involve either x and y axes, whereas Attack 4 only involves z-axis manipulation. Attacks 1 and 2 are performed on the middle infill line of the internal layers. Attack 3 is implemented on consecutive infill lines in the central infill region. Attack 4 involves z-axis manipulation only, and implants the attack instances in the internal layers. Initially, the attack magnitudes selected for all attacks are 0.05 mm, 0.1 mm, and 0.2 mm. Where needed to further examine the trend, additional steps are inserted at appropriate places. Attack 1 is examined up to 0.015 mm magnitude, whereas an extra step of 0.15 mm is inserted for Attack 3 and 4, and they are not examined below 0.05 mm. Five samples are printed for each attack instance.

Table 8.: Design of experiment for the proposed attacks

	Proposed Attacks			
	Attack 1	**Attack 2**	**Attack 3**	**Attack 4**
Attack target : Inter-extrudates bonding between	2 consecutive infill lines across the span	2 consecutive infill lines at one edge	Infill and wall structure	Infill across 2 consecutive layers
Kinetic manipulation axes	x , y	x , y	x , y	z
Attack Location	Internal layers, middle infill	Internal layers, middle infill	Internal layers, central infill zone	3 instances/attack in internal layer
Attack Instances	**Attack Magnitudes (mm)**			
1	0.015	0.025	0.05	0.05
2	0.025	0.05	0.10	0.10
3	0.05	0.10	0.15	0.15
4	0.10	0.20	0.20	0.20
5	0.20	-	-	-

3.5 Experiment results

This section presents the performance of the attacked specimens in accordance with the success criteria outlined in Section 3.3.1.

3.5.1 Attack stealthiness against part inspection

As the attacks are performed on the internal layers without manipulating the walls structure, no visual impairment or modification is observed in any of the printed parts. The measurements of the printed parts confirm that the statistical difference between the thickness and width of the attacked versus non-attacked samples remains less than 0.1 mm for all cases. Similarly, the deviation in the mass of the printed parts is less than 0.03 grams. Table 9 presents the mean, standard deviation, and the difference between dimensions and mass of the attacked specimens compared to their corresponding non-attacked specimens values.

Table 9.: Stealthiness performance: impact on dimensions and mass of the attacked parts

Attack	**Attack**	**Width (mm)**			**Thickness (mm)**			**Mass (gram)**		
type	**mag**	**Mean**	**Std dev**	**Diff.**	**Mean**	**Std dev**	**Diff.**	**Mean**	**Std dev**	**Diff.**
Infill lines spacing (A1)	0	12.877	0.023	0.000	2.555	0.021	0.000	2.876	0.014	0.000
	0.015	12.885	0.028	0.008	2.518	0.012	-0.037	2.881	0.110	0.005
	0.025	12.853	0.009	-0.023	2.510	0.010	-0.045	2.883	0.012	0.007
	0.05	12.843	0.012	-0.033	2.537	0.014	-0.018	2.868	0.016	-0.008
	0.10	12.853	0.021	-0.023	2.543	0.019	-0.012	2.859	0.011	-0.014
	0.20	12.868	0.020	-0.008	2.558	0.011	0.003	2.853	0.018	-0.023
Infill vertices spacing (A2)	0	12.877	0.023	0.00	2.555	0.020	0.000	2.876	0.014	0.000
	0.025	12.885	0.042	0.008	2.590	0.023	0.035	2.869	0.011	-0.007
	0.05	12.907	0.038	0.030	2.588	0.018	0.033	2.874	0.019	-0.002
	0.10	12.898	0.045	0.021	2.580	0.021	0.025	2.886	0.017	0.010
	0.20	12.892	0.041	0.015	2.583	0.012	0.028	2.865	0.012	-0.011
Infill to walls bonding (A3)	0	12.873	0.017	0.000	2.563	0.017	0.000	2.870	0.022	0.000
	0.05	12.902	0.016	0.029	2.563	0.012	0.000	2.895	0.032	0.025
	0.10	12.891	0.017	0.018	2.547	0.005	-0.017	2.896	0.006	0.026
	0.15	12.887	0.028	0.014	2.567	0.012	0.003	2.881	0.021	0.012
	0.20	12.901	0.008	0.028	2.557	0.005	-0.007	2.870	0.022	0.000
Interlayer bonding (A4)	0	12.724	0.043	0.000	2.563	0.017	0.000	2.897	0.023	0.000
	0.05	12.721	0.022	-0.003	2.587	0.025	0.024	2.890	0.019	-0.006
	0.10	12.743	0.017	0.019	2.578	0.007	0.014	2.881	0.023	-0.015
	0.15	12.750	0.027	0.026	2.590	0.022	0.027	2.878	0.016	-0.019
	0.20	12.726	0.030	0.002	2.574	0.016	0.011	2.871	0.022	-0.026

Table 10.: Stealthiness performance: impact on the commonly monitored parameters

Attack Type	**Attack magnitude**	**Launch time delay (sec)**	**Max spatial deviation Linear (mm)**	**Angular (degree)**	**Per attack command time difference (sec)**	**Per command filament length difference (mm)**
Infill lines spacing attack (A1)	0.015	<0.25	0.015	0	<0.005	None
	0.025	<0.25	0.025	0	<0.005	None
	0.05	<0.25	0.05	0	<0.005	None
	0.1	<0.25	0.1	0	<0.005	None
	0.2	<0.25	0.2	0	<0.005	None
Infill vertices spacing attack (A2)	0.025	<0.25	0.025	0.154	<0.005	None
	0.05	<0.25	0.05	0.308	<0.005	None
	0.1	<0.25	0.1	0.612	<0.005	None
	0.2	<0.25	0.2	1.211	<0.005	None
Infill to wall bonding attack (A3)	0.05	<0.25	0.05	0	<0.005	None
	0.1	<0.25	0.1	0	<0.005	None
	0.15	<0.25	0.15	0	<0.005	None
	0.2	<0.25	0.2	0	<0.005	None
Interlayer bonding attack (A4)	0.05	<0.25	0.05	0	None	None
	0.1	<0.25	0.1	0	None	None
	0.15	<0.25	0.15	0	None	None
	0.2	<0.25	0.2	0	None	None

3.5.2 Attack stealthiness against process monitoring schemes

The attacks are launched after the user sends the printing instruction. The overall time in finding, recalculating and modifying the G-codes is less than 200 ms on a Core i7-8700, 16 GB RAM machine. The induced delay is low enough to avoid any suspicion or alarm. Maximum deviation from the original toolpath is equivalent to the attack magnitude. In the existing literature, the lowest detectable toolpath deviation attack with acceptable false-positives [31] is 0.3 mm in the xy axes, making our highest magnitude attack to be still below the detection horizon. For layer thickness, the detectable threshold in the literature is 0.05 to 0.1 mm if repeated over multiple layers [31, 19]. In the proposed Attack 4, the deviation is not accumulated; thus making it more challenging for the detection tools. The angular displacement only occurs in Attack 2 by a small value ranging from 0.154^o to 1.211^o at different attack magnitudes. There is no work in the literature claiming to detect this magnitude of angular deviation in a high-velocity FFF printing setup. The attacks effectively maintain the timing profile integrity on a per-instruction basis. When sampled at 5 ms, no statistical difference is observed in the execution time for Attacks 1 to 3 at the selected printing settings. The printing bed movement in Attack 4 takes from 50 to 150 ms in our printed specimen. Malicious bed movement is incorporated within the infill move command ensuring no extra time for Attack 4. No attack modifies the filament consumption for any G-code instruction.

3.5.3 Impact of attacks on tension and flexure strength

In this subsection, the results of tensile and bending tests are presented for the proposed attacks. As the study was performed over a span of a few months using different PLA spools, a set of non-attacked parts is printed for each category except for the tensile specimens for Attack 1 and 2 (being printed through the same spool and settings).

3.5.3.1 Attack 1 mechanical tests results

The tensile test results and the stress vs strain curves for Attack 1 are presented in Table 11 and Figure 22, respectively. This attack shows up to a 33% reduction in peak stress value at 0.1 mm or higher attack magnitude. The attacked specimens always broke from the point of attack and at a lower strain value. Table 12 presents three-point bending test results showing a 28% reduction in the peak flexure stress for the highest attack magnitude. Flexure stress vs flexure strain curves for Attack 1 are presented in Figure 23.

Table 11.: Tensile test results for attack 1: Infill lines spacing attack

	Peak load (N)			**Peak stress (MPa)**			**Strain at break (mm/mm)**		
Attack magnitude	**Average**	**Std dev**	**%age diff**	**Average**	**Std dev**	**%age diff**	**Average**	**Std dev**	**%age diff**
0	936.9	98.8	0.0	35.5	3.4	0.0	0.035	0.003	0.00
0.015	938.1	40.9	0.1	35.5	1.7	0.1	0.034	0.004	-2.86
0.025	919.9	35.7	-1.8	34.4	1.6	-3.1	0.03	0.002	-14.29
0.05	694.7	18.0	-25.8	25.9	0.7	-26.9	0.031	0.004	-11.43
0.1	622.6	34.7	-33.6	23.2	1.4	-34.6	0.026	0.002	-25.71
0.2	624.3	32.6	-33.4	23.3	1.3	-34.3	0.024	0.004	-31.43

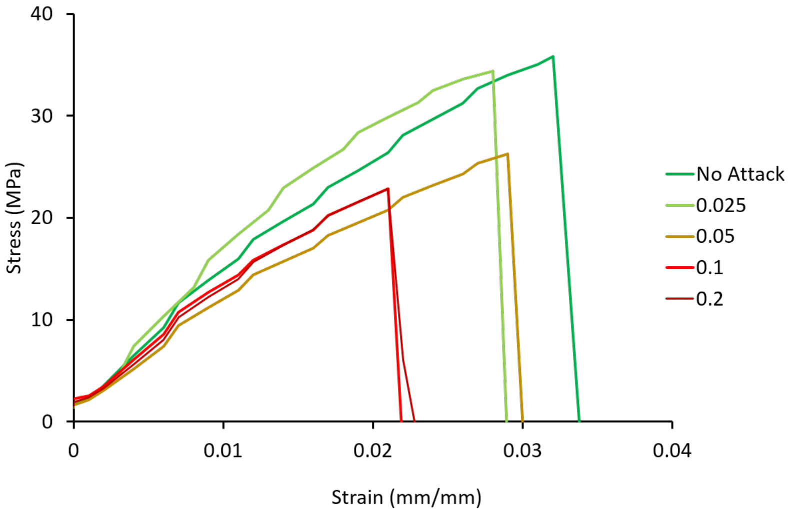


Fig. 22: Attack 1 strain-strain curves for tensile tests

Table 12.: Three point bending test results for attack 1: Infill lines spacing attack

	Peak load (N)			Peak flexure stress (MPa)		
Attack magnitude	**Mean**	**Std dev**	**%age diff**	**Mean**	**Std dev**	**%age diff**
0	103.09	4.44	0.00	74.54	3.12	0.00
0.025	101.98	4.64	-1.08	71.82	3.02	-3.64
0.05	99.84	1.44	-3.16	69.61	1.93	-6.61
0.1	77.51	4.96	-24.82	53.28	3.60	-28.52
0.2	74.18	13.08	-28.05	53.23	10.64	-28.58

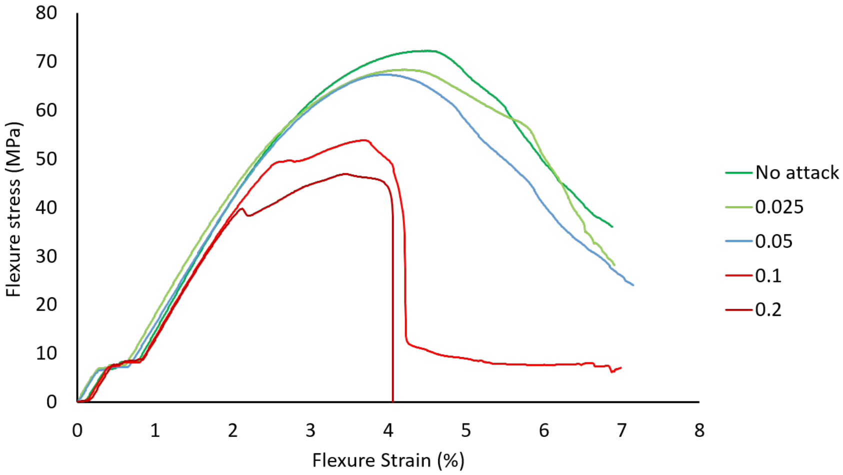


Fig. 23: Attack 1 flexure stress vs strain curves for three-point bending tests

3.5.3.2 Attack 2 mechanical tests results

Table 13 and Figure 24 presents the tensile tests results, whereas Table 14 and Figure 25 presents the three-point bending tests results for Attack 2. The maximum tensile stress reduction of 12.4% is observed at 0.2 mm attack magnitude. At all magnitudes, the attacked specimens broke at a lower strain value. The bending tests show a reduction of 25% in the peak flexure stress value.

Table 13.: Tensile test results for attack 2: Infill vertices spacing attack

	Peak load (N)			**Peak stress (MPa)**			**Strain at break (mm/mm)**		
Attack magnitude	**Average**	**Std dev**	**%ag diff**	**Average**	**Std dev**	**%age diff**	**Average**	**Std dev**	**%age diff**
0	1036.1	42.5	0.0	38.6	1.5	0.0	0.035	0.003	0.0
0.025	1041.9	59.8	0.6	38.7	2.4	-0.5	0.03	0.003	-13.5
0.05	1008.7	39.1	-2.6	37.9	1.3	-2.5	0.027	0.001	-21.5
0.1	953.4	44.4	-8.0	35.5	1.8	-8.7	0.026	0.008	-25.0
0.2	916.3	36.5	-11.6	34.1	1.2	-12.4	0.025	0.006	-27.8

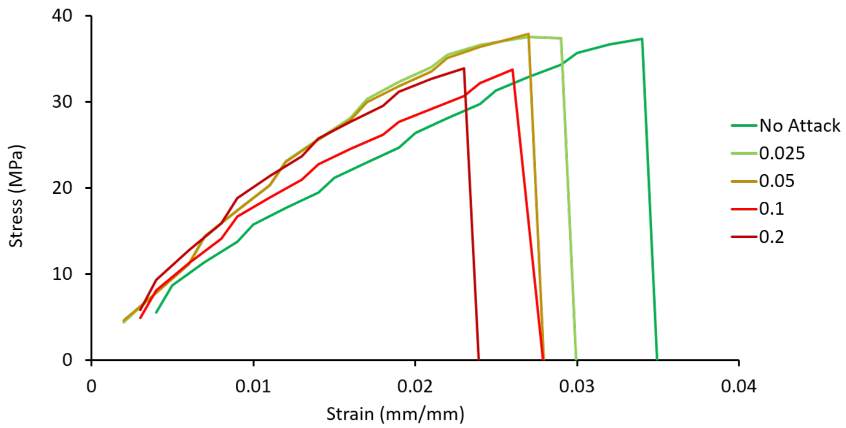


Fig. 24: Attack 2 strain-strain curves for the tensile tests

Table 14.: Three point bending test results for attack 2: Infill vertices spacing attack

	Peak load (N)			Peak flexure stress (MPa)		
Attack magnitude	**Mean**	**Std dev**	**%age diff**	**Mean**	**Std dev**	**%age diff**
0	103.09	4.44	0.00	74.54	3.12	0.00
0.025	102.12	3.95	-0.94	73.43	1.92	-1.49
0.05	98.06	1.89	-4.88	70.85	1.17	-4.94
0.1	93.27	2.46	-9.52	67.67	1.06	-9.22
0.2	74.94	2.10	-27.31	55.52	2.52	-25.51

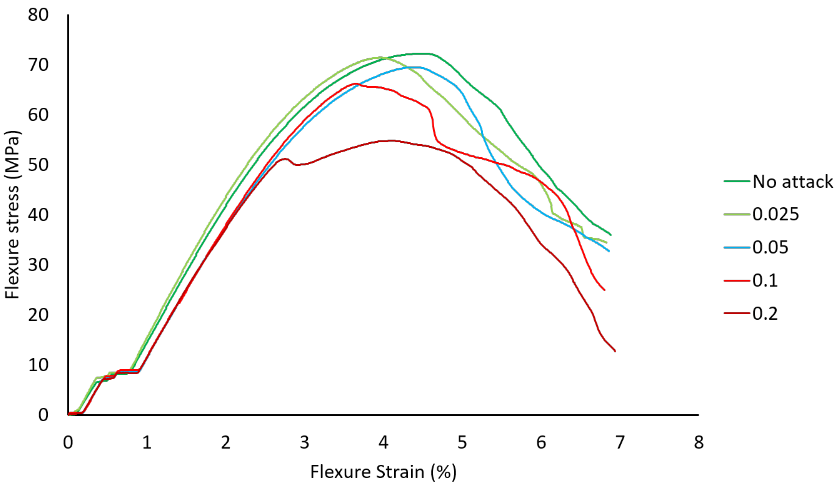


Fig. 25: Attack 2 flexure stress vs strain curves for three-point bending tests

3.5.3.3 Attack 3 mechanical tests results

Table 15 and Figure 26 presents the tensile tests results, whereas Table 16 and Figure 27 presents the three-point bending tests results for Attack 3. Although the attacks show a consistent reduction in peak tensile load and stress values, the maximum reduction is only 5.8%, which is not as pronounced as in Attacks 1 and 2.

All attacked specimens still broke at a lower strain value. Similarly, the maximum reduction in bending strength is 6.17%. The reason for this low impact is discussed ahead in Section 3.6.

Table 15.: Tensile test results for attack 3: Infill to wall structure bonding attack

	Peak load (N)			Peak stress (MPa)			Strain at break (mm/mm)		
Attack magnitude	**Average**	**Std dev**	**%ag diff**	**Average**	**Std dev**	**%age diff**	**Average**	**Std dev**	**%age diff**
0	1195.9	14.5	0.0	42.5	0.8	0.0	0.0305	0.006	0.000
0.025	1180.2	15.5	-1.3	42.1	0.5	-0.9	0.0293	0.004	-3.780
0.05	1147.5	21.9	-4.0	41.2	0.9	-3.1	0.0223	0.001	-26.776
0.1	1130.1	15.2	-5.5	40.5	0.6	-4.6	0.0237	0.002	-22.404
0.2	1121.9	8.0	-6.2	40.0	0.5	-5.8	0.0254	0.004	-16.721

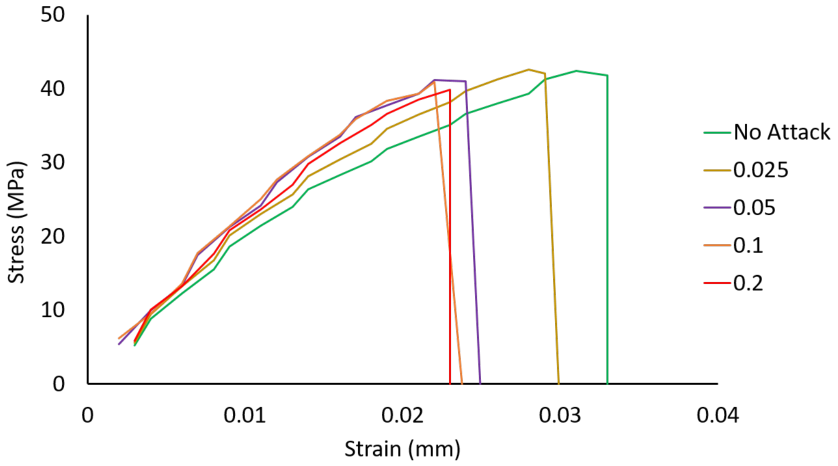


Fig. 26: Attack 3 strain-strain curves for the tensile tests

Table 16.: Three point bending test results for attack 3: Infill to wall spacing attack

	Peak load (N)			**Peak flexure stress (MPa)**		
Attack magnitude	**Mean**	**Std dev**	**%age diff**	**Mean**	**Std dev**	**%age diff**
0	107.78	0.62	0.00	78.35	1.67	0.00
0.05	105.26	0.26	-2.72	75.75	2.83	-3.31
0.1	103.03	0.44	-5.12	73.51	1.23	-6.17
0.15	103.48	1.89	-4.63	73.92	1.29	-5.64
0.2	103.24	0.54	-4.91	73.71	0.72	-5.91

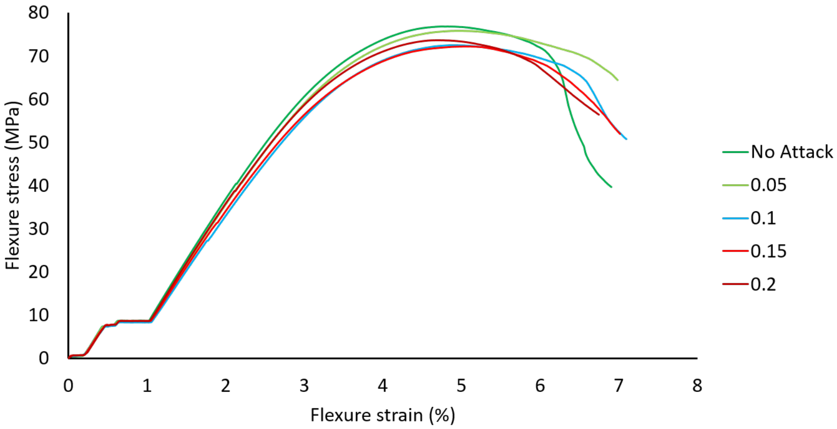


Fig. 27: Attack 3 flexure stress vs strain curves for three-point bending tests

3.5.3.4 Attack 4 mechanical tests results

Table 17 and Figure 28 presents the tensile tests results, whereas Table 18 and Figure 29 presents the three-point bending tests results for Attack 4. Peak tensile stress reduction observed at the highest attack magnitude is 23%. Unlike the other three attacks, these attacked specimens did not break earlier except for the ones attacked with the highest magnitude (0.2 mm). Maximum reduction in the peak flexure stress is recorded as 16.56%.

Table 17.: Tensile test results for attack 4: Interlayer bonding attack

	Peak load (N)			**Peak stress (MPa)**			**Strain at break (mm/mm)**		
Attack magnitude	**Average**	**Std dev**	**%ag diff**	**Average**	**Std dev**	**%age diff**	**Average**	**Std dev**	**%age diff**
0	1345.1	13.49	0.00	50.60	0.163	0.00	0.023	0.000	0.00
0.05	1269.8	9.88	-5.59	48.53	0.822	-4.08	0.024	0.001	2.86
0.10	1244.1	21.46	-7.51	46.50	0.852	-8.10	0.022	0.001	-7.14
0.15	1139.6	4.13	-15.28	43.07	0.665	-14.89	0.022	0.001	-7.14
0.20	1043.3	8.79	-22.44	38.93	0.471	-23.06	0.020	0.000	-14.29

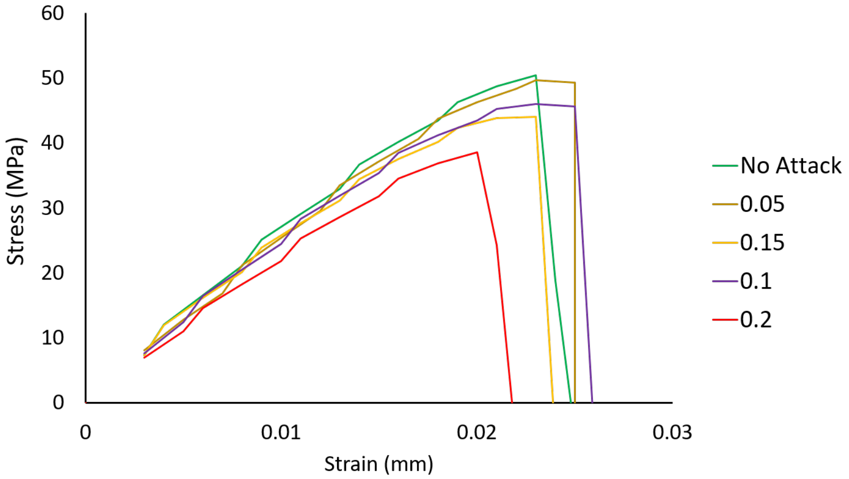


Fig. 28: Attack 4 strain-strain curves for the tensile tests

Table 18.: Three point bending test results for attack 4: Interlayer bonding attack

	Peak load (N)			**Peak flexure stress (MPa)**		
Attack magnitude	**Mean**	**Std dev**	**%age diff**	**Mean**	**Std dev**	**%age diff**
0	103.81	0.00	0.00	74.99	0.00	0.00
0.05	103.14	0.16	-0.65	73.74	0.60	-1.67
0.10	102.08	2.10	-1.67	73.84	1.33	-1.56
0.15	98.36	2.87	-5.25	69.46	1.18	-7.49
0.2	86.98	1.15	-16.22	63.49	0.75	-16.56

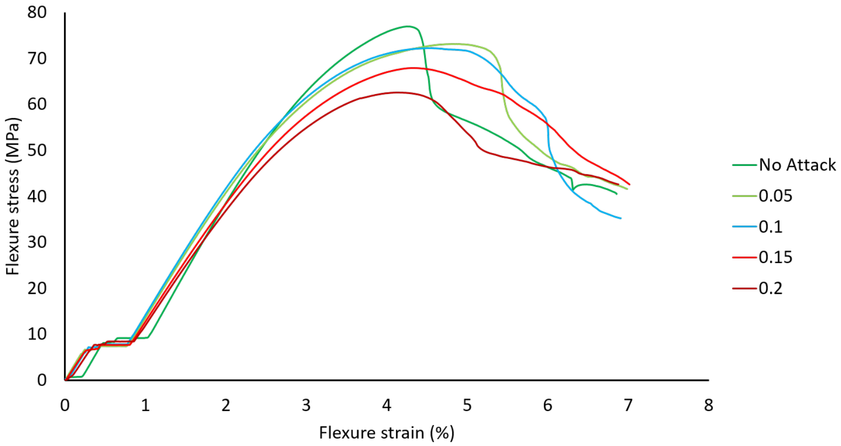


Fig. 29: Attack 4 flexure stress vs strain curves for three-point bending tests

3.6 Analysis

The results presented in the previous section confirm that the manipulation of extrudates bonding through kinetic variations can negatively impact the tensile and bending strength profiles of the printed part. These proposed low-magnitude attacks produce a very small footprint and can evade most of the existing attack or anomaly detection techniques. Attack 1 and 2 fully comply with the four-point success criteria proposed in Section 3.3.1. In Attack 4, although the attack magnitude remains within 0.2 mm, some techniques [31] claim to reliably detect layer thickness deviations of 0.05 mm or more. The performance of these detection techniques is not measured after enabling the auto-leveling feature available in the latest printers. The continuous movement of the printing bed for automatic leveling can confuse the detection scheme and creates an opportunity to camouflage the attack in excessive false positives.

Figure 30 presents the peak tensile and flexure strength values plotted against the attack magnitude. Attacks 1 and 2 caused the highest reduction in tensile and bending strength, but they are suitable for solid parts only. Attacks 3 and 4 caused less reduction in tensile and bending strength, but they are not limited to solid geometries only. As the attacks create imperfections along different axes, the direction and type of load will impact the choice of attack. A higher reduction in tensile strength in the first two attacks is attributed to the suitable direction of imperfection with respect to the applied load.

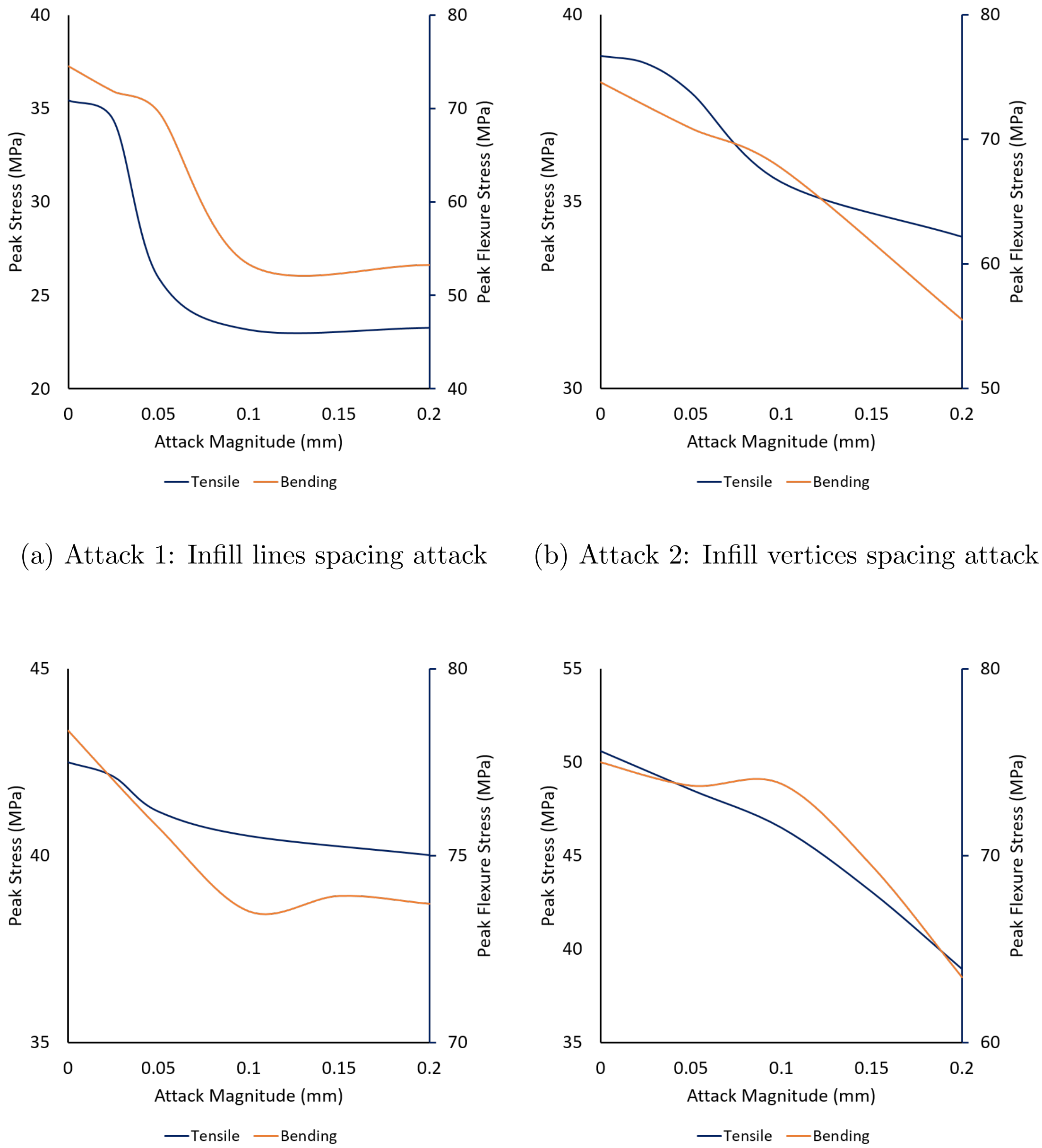
For attack magnitudes greater than 0.1 mm in Attack 1, the attacked layers did not contribute to the tensile strength of the specimens. As presented in Figure 30a, the peak stress value becomes nearly constant after a magnitude of 0.1 mm. The imperfection in Attack 2 is only introduced at one end of the infill lines pair (refer to Figure 18), resulting in a lesser impact on tensile strength as presented in Figure 14. Unlike Attack 1, the reduction in Attack 2 continues after 0.1 mm but at a slower rate.

Attack 3 causes a minimal reduction in tensile strength. A cross-sectional view of the attacked specimens after destructive tests show a tiny crack across multiple layers at 0.2 mm magnitude (see Figure 31). The defect introduced in Attack 3 is aligned with the tensile load direction; thus not causing any significant impact. This kind of attack is effective for parts under compression or shear stress. In Attack

4 which targets interlayer bonding, the tensile strength reduction continues till 0.2 mm and beyond at a nearly linear rate. Figure 32 presenting cross-sectional views of 5 instances of Attack 4 highlights that the outer wall structures on both edges (in green color) do not show notable signs of attack. The weak interlayer bond in the infill structure gets obvious as the attack magnitude increases from zero to 0.2 mm (from left to right in the figure).

A similar trend with slight differences is observed in the bending test results. As Attacks 1 and 2 are planted at the center of the part, the three-point bending results show considerable strength reduction proportional to the attack magnitude. The zone of steep reduction for Attacks 1 and 2 was shifted by approximately 0.05 mm in comparison to the tensile test results. In Attack 3, a small reduction in bending stress is observed at a lower attack magnitude as visible in Figure 30c, but the trend did not continue as the attack magnitude increased to 0.2 mm. As this experiment was restricted to a maximum deviation of 0.2 mm, the study did not investigate the effect at higher magnitudes. In Attack 4, a considerable impact on the bending stress is observed after the attack magnitude is raised from 0.1 towards 0.2 (see Figure 29).

Fig. 30: Peak tensile and bending stresses at various attack magnitudes

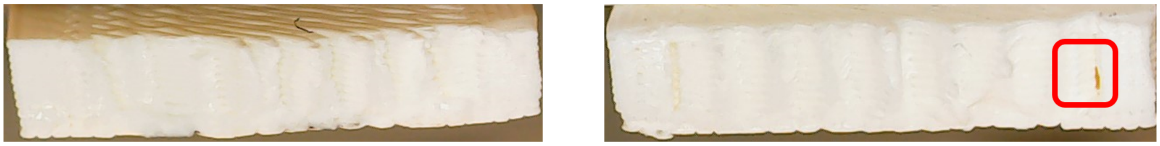


(a) Attack 1: Infill lines spacing attack

(b) Attack 2: Infill vertices spacing attack

(c) Attack 3: Infill to wall structure bonding attack

(d) Attack 4: Interlayer bonding attack



Non-attacked specimen for Attack-3

Attack-3: 0.2 mm

Fig. 31: Attack 3 specimens after bending tests

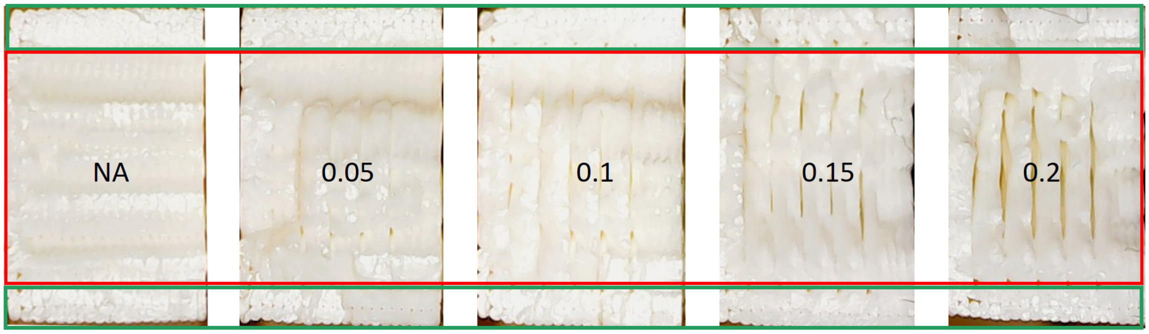
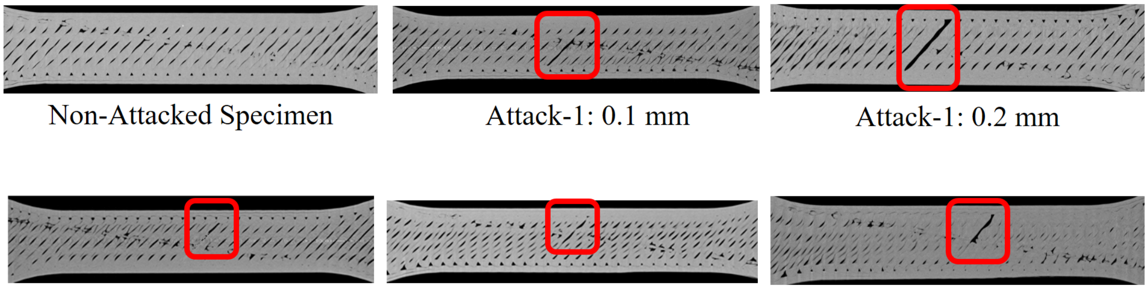


Fig. 32: Interlayer bonding attack (Attack 4) specimens after performing tensile tests



Non-Attacked Specimen

Attack-1: 0.1 mm

Attack-1: 0.2 mm

Attack-2 : 0.05 mm

Attack-2: 0.1 mm

Attack-2: 0.2 mm

Fig. 33: Micro CT scan results for a few selected specimens

3.7 Attack Countermeasures

This section discusses possible attack avoidance and detection measures against the proposed attacks. As the attacks are launched by hijacking the network connection, cybersecurity measures including access control and encrypted communication, DHCP snooping, and dynamic ARP inspection are effective in avoiding MiTM attacks. These attacks can also be performed by compromising the printer's firmware or by using a kernel module to manipulate the G-code file in the slicer process memory. Tools such as Tracee [48] that detect and report the loading of any kernel module are helpful in detection and investigation. However, it is still challenging to automatically detect these attacks without any ground truth. To ascertain firmware integrity, periodic firmware verification through out-of-band methods may be employed [49].

If the attacker succeeds in launching the attacks, some non-destructive tests may still reveal the abnormalities in the internal structure. To examine the feasibility of such a possibility, the authors conducted x-ray micro-computed tomography tests on a few selected specimens from Attack 1 and 2. The scans were performed using a Skyscan 1173 machine at 1.8 seconds exposure, 0.5 rotational steps, and 20μm pixel size. The attack with magnitudes of 0.1 mm and higher can be easily spotted in Figure 33. However, attacks with lower magnitudes do not create obvious defects. Another challenge with micro CT scans is the manual examination that requires time and expertise. Yoginath et al. have proposed the use of trained probabilistic models to highlight the low-magnitude anomalies using phase-angle deviation in noisy side-channel data [50]. A real-time detection solution based on the approach can help in discriminating malicious low-magnitude deviations from the noise.

3.8 Limitations and future work

The two main physical processes in fused filament fabrication-based printing that impact the extrudates bonding and can be influenced by cyberattacks are kinetics and thermodynamics. This study was limited to kinetic manipulations. A future direction is to examine low-magnitude thermodynamic manipulations followed by hybrid attacks that leverage the full potential of modifiable parameters by keeping below the attack detection horizon. An important future direction is to develop cybersecurity solutions that are capable of detecting malicious deviations and simultaneously estimating their impact on the printed part to take an informed verdict about continuing, aborting, or mitigating the attack. One such security scheme could base its decision on the mechanical properties of the printed part estimated in real-time.

3.9 Conclusion

With all the benefits and the promise, the material extrusion printing process also brings up some unique challenges. To use material extrusion for printing critical parts, it is important to establish the integrity of the printing operation. Material extrusion printer's trueness and precision specifications offer an opportunity window to launch attacks that do not deviate the process beyond the tolerance window and still impact the mechanical strength. It is challenging for process monitoring-based techniques to declare such small within-tolerance deviations as anomalies. This study proposes four attacks on fused filament fabrication printing by manipulating the bonding between two neighboring extrudates from the same and adjacent layers. The attacks were demonstrated by modifying the G-code file through a man-in-the-middle attack on the network segment between the control machine and the printer. An experiment was designed to evaluate the impact of the proposed attacks on ASTM D638 Type IV standard tensile bars and ASTM D790 compliant flexure bars using attack deviation magnitudes ranging from 0.015 mm to 0.2 mm. The destructive tests conducted on the attacked specimens confirm that the attacks are capable of reducing tensile and bending strength by up to 28% and 25%, respectively. Manual analysis of Micro CT scans of the attacked specimen shows that higher magnitude attacks can be spotted. However, manual scanning and analysis is not a scalable solution for commercial manufacturing. In addition to the standard cybersecurity tools and methods to avoid MiTM attacks, the authors recommend that researchers may investigate the option of real-time CT scanning and automated analysis to ascertain the printing process integrity. Another potential solution against such attacks is to monitor the printing process and apply a faster version of finite element analysis to predict the printing properties of the part. If FFF has to be successful in printing functional parts, a dedicated research effort is required to safeguard it against low-magnitude sabotage attacks on extrudates bonding.

CHAPTER 4

WEAPONIZING 3D PRINTERS: HOW FIRMWARE ATTACKS ENABLE ADVERSARIAL OBJECTIVES

Fused filament fabrication (FFF) based 3D printing has become increasingly ubiquitous in modern manufacturing, whether for functional part production or aesthetic purposes. As concerns about international supply chain integrity grow in the geopolitical scenario, the risk of firmware attacks is also increasing. In a hypothetical but realistic scenario, a user receives a 3D printer at a classified outfit and performs a firmware upgrade to ensure that the printer runs on trusted and the latest firmware. However, the printer deceives the user by displaying the new firmware version while retaining the previous malicious copy. The starting motivation for this study is to find out what goals an attacker can achieve with malicious 3D printer firmware. We use Nickerson's approach to propose a firmware attack taxonomy based on attack goals and firmware interactions. We systematically expand the attack goals in the proposed taxonomy by increasing the levels of the categorization tree to separate the printed object, printer, and printing environment. In order to fill the research gap on firmware attacks in the existing literature, we propose nine innovative attacks on Marlin - the most widely used printer firmware that is utilized by more than 25 3D printer vendors. These attacks are tested on a standard 3D printer and evaluated through relevant destructive and non-destructive tests. To ensure the persistence of the attacks, we design and implement a malicious bootloader that can circumvent firmware upgrade activities. Attack analysis reveals that some trivial attack actions at pre-firmware stages are computationally complex or infeasible at the firmware stage. This finding moti-

vates us to examine 46 attacks (the proposed and the existing ones) and analyze their feasibility status at five different stages of the printing process. To summarize our results, we developed an Attack Feasibility Index (AFI) that scores the feasibility of each attack category for each of the five cyber artifacts corresponding to the printing process stages. While the AFI score for the firmware (or printing stage) is highest for printing facility surveillance and sabotage attacks, it is not the leading stage for carrying out sophisticated attacks that involve intelligently manipulating a printed part to cause damage to the system where the part will be installed. Our presented attacks and analysis will encourage researchers to investigate cybersecurity solutions that are process-stage specific and optimized for feasible attacks.

4.1 Introduction

The popularity of additive manufacturing (AM) is on the rise [51] with critical industrial sectors such as aerospace [52], automobile and healthcare [53] using 3D-printed functional parts. Malicious actors can now earn a higher reward for attacking an AM setup and sabotaging the printed part. Concurrently, the current industry trend of fully connected and converged IT and industrial networks [54] potentially extends the reach of cyber attackers to manufacturing units. For the last few years, the research community has been actively working on the offensive and defensive side of AM security.

The existing offensive research focuses on either stealing the IP information through side channels [55, 56] or inducing defects in the printed part [12]. Being fundamentally different from its predecessor technologies, AM offers many unique attacks to sabotage the physical properties of the printed parts. Some of these attacks degrade the object's mechanical strength without modifying the dimensions, mass, center of mass, and other measurable attributes [33]. Although the researchers acknowledge the possibility of firmware attacks [8], most of the sophisticated attacks are demonstrated at the pre-firmware stages. Moreover, no taxonomy of firmware attacks exists in AM security literature.

This study proposes a two-dimensional taxonomy of firmware attacks in the additive manufacturing (AM) domain. The taxonomy is developed based on attack goals and firmware interactions, following Nickerson's approach [57]. The goal of the taxonomy is to understand the capabilities of malicious firmware. The taxonomy identifies five elements that malicious firmware interacts with, namely network, human-machine-interface (HMI), actuators, sensors, and print objects, to achieve four top-level attack goals, including surveillance, denial of service, integrity breach, and unauthorized printing. We systematically expand the attack goals in the proposed taxonomy by increasing the levels of the categorization tree to separate printed object, printer, and printing environment.

To evaluate our taxonomy, we implemented nine cyber and cyber-physical attacks using Marlin, which is the most widely used open-source firmware for 3D printers, utilized by more than 25 vendors with minor modifications [58]. The attacks cover the proposed categorization tree (except for the obvious nodes). We install the malicious firmware in the printer by assuming a supply chain attack vector. As a 3D printer firmware can be easily updated through the printer control software, all the attack efforts may go in vain. To handle this issue, our attacks ensure persistence by preventing the firmware from being updated while maintaining an unaltered user experience. The source code for the malicious bootloader (MalBoot) and malicious firmware for all attacks, along with relevant test G-code files, is publicly available to facilitate future research. One of the attacks we implemented, named *Print your own grave*, prints a tool and uses it to physically damage the printer's components. Another attack, named *Incurable*, deceives the user by mimicking common printing faults, leading to prolonged and ineffective troubleshooting of the printing environment. By highlighting the challenges associated with the proposed attacks, we dispel the perception that all attacks are simple at the firmware level. In fact, all attacks are not feasible at any stage of the printing process, including firmware.

The 3D printing process consists of three distinct stages: designing, slicing, and printing. The difficulty level of implementing an attack at a specific stage may vary depending on the attack goal. For example, denying printing services by ignoring the printing instructions is a straightforward task for malicious firmware, whereas the computation and space complexity for scaling an object at the firmware is very high. Assessing the complexity of attacks is crucial for understanding the risks involved and prioritizing defensive measures. However, we found no existing research on the complexity analysis of additive manufacturing (AM) attacks.

To fill this gap, we conduct an in-depth analysis of 46 different attacks, including our proposed attacks, to evaluate their implementation complexity and the feasibility of detecting compliance with attacks at different stages of the printing process. We assume that an attacker could manipulate any of the five cyber artifacts associated with the three process stages. Each attack is classified as either Infeasible, High-difficulty, Medium-difficulty, or Low-difficulty attack. To summarize our findings, we introduce an Attack Feasibility Index (AFI), which represents the feasibility of implementing a specific category of attacks at a particular stage of the printing process. An AFI value of zero for Stage s indicates that a particular category of attacks is not feasible to launch at Stage s, while an AFI value of one indicates that the attack objective category is relatively simple to implement at Stage s.

The source code for both the malicious bootloader (MalBoot) and malicious firmware used in all proposed attacks, along with the relevant test G-code files, have been made publicly available to encourage and support future research in the field.

4.1.1 Motivation

Despite the existence of firmware integrity verification techniques for IoT and CPS devices, no solution is infallible enough to guarantee 100% protection against future attacks. As a result, malicious firmware installation and concealment in 3D printers are possible, as exemplified by the use of supply chain vulnerabilities. By understanding the potential attack objectives and the level of complexity involved in implementing them through malicious firmware, cybersecurity researchers can gain valuable insight into the attack execution path, its likelihood of success, its impact, and associated risks. The motivation for conducting this research lies in the absence of comprehensive answers in the existing literature to the following questions: (1) What are the potential attack goals that can be achieved through malicious firmware in AM? (2) How can attacks on AM firmware be analyzed and understood? (3) What is the level of complexity and feasibility associated with implementing an AM attack through malicious firmware? By answering these questions, this study provides valuable insights into the execution path, likelihood of success, and potential impact of such attacks, which can help cybersecurity researchers in developing effective defense strategies.

4.1.2 Contributions

This study offers the following contributions:

1. A novel taxonomy for AM firmware attacks based on attack goals and firmware interactions, including an attack goals categorization tree.

2. Implementation of nine novel AM firmware attacks on a FFF 3D printer, made persistent by developing a bootloader malware (MalBoot), and evaluation through destructive and non-destructive tests.

3. An analysis of the feasibility of 46 AM attacks at various stages of the AM process, providing insights into the complexity of implementing AM attacks through malicious firmware.

4.2 Background & related work

4.2.1 Related work

This section provides a concise summary of the ongoing research endeavors concerning attack taxonomies and firmware attacks on AM systems.

Attack Taxonomies Several research studies have proposed taxonomies for cyber-physical system (CPS) attacks in additive manufacturing (AM) systems. Yampolskiy et al. [59] developed an attack taxonomy that focuses on semantically identical manipulations introduced by different compromised elements. Their taxonomy includes a subset of targeted properties known as 'attack targets,' but does not elaborate on the attacker's goals, nor does it consider denial of service attacks.

Pan et al. [60] proposed a taxonomy that comprises vulnerability, attack vector, attack target, and attack impact. However, it is not focused on attack goals. Similarly, Berger et al. [61] developed a multi-layer taxonomy for Industrial IoT attacks, using Nickerson's iterative approach [57], including a single-layer classification for attack consequences.

Mahesh et al. [62] presented a four-level attack taxonomy for AM systems, starting with attack goals, methods, targets, and countermeasures. However, their taxonomy does not cover an in-depth categorization of attack goals, and they include service denial and IP theft as methods rather than attack goals.

Finally, Wu et al. [63] developed a taxonomy for AM attacks that includes two parallel streams of cyber and physical attacks. However, it only enumerates a few attack outcomes under cyber and physical attack consequences.

Firmware Attacks on 3D Printers While many attacks on 3D printers have been demonstrated at the pre-firmware stages, such as IP theft and printed part sabotage, there has been limited work done on firmware attacks. Xiao [23] demonstrated the feasibility of firmware attacks on 3D printers by modifying the open-source RepRap firmware through a USB-based serial connection. Moore et al. [22] studied the impact of compromised firmware on print quality, showing that they were able to manipulate extruder feedrate or print alternate geometries using an attacked firmware. However, this study did not provide a comprehensive analysis of attacks achievable through firmware compromise. Recently, Pearce et al. [40] presented ”FLAW3D,” a bootloader trojan capable of attacking AVR-based Marlin-supported 3D printers. They demonstrated two simple, low-footprint attacks that could reduce the strength of printed parts.

Our work is distinguished from these studies in two aspects. First, we focus on firmware attacks and consider firmware interactions, which are not covered in the existing taxonomies. Second, our attack taxonomy includes a comprehensive categorization of attack goals that is not available in other taxonomies. By focusing on firmware attacks, our work sheds light on a critical but relatively under-explored aspect of 3D printer security. Through our taxonomy, we aim to provide a comprehensive framework that can guide future research on firmware attacks and inform the development of more effective security measures for 3D printers.

4.3 Proposed two-dimensional attack taxonomy

This section presents a two-dimensional firmware attack taxonomy based on the attacker’s goals and the firmware interactions required to achieve those goals.

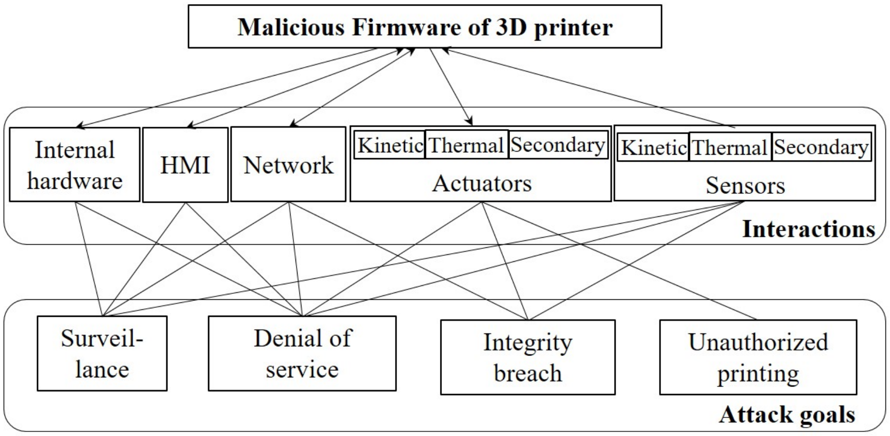


Fig. 34: Two-layers taxonomy of firmware attacks

4.3.1 Assumptions

The taxonomy assumes that malicious firmware is already installed on the printer by compromising the supply chain or through any other mechanism. As a consequence, the taxonomy does not focus on how' but rather on what' after the firmware is compromised. The malicious firmware deceives the employed integrity verification scheme [64, 65] during the pre-printing phase and successfully reaches the printing phase to exhibit malicious behavior. The taxonomy also assumes that only the firmware is compromised by the attacker, and other independent interacting entities (such as slicing software) are not initially compromised.

4.3.2 Methodology

The taxonomy is developed using Nickerson's approach incorporating five characteristics: concise, robust, comprehensive, extendible, and explanatory [57]. After analyzing the taxonomy's purpose, we adopt a two-dimensional taxonomy with 'attack goal' as the meta-characteristic, and 'firmware interactions' with the outside entities as the other dimension. We adopt the conceptual approach to create the taxonomy followed by an empirical evaluation and a final revision. The five characteristics mentioned above serve as the subjective ending conditions. To qualify for inclusion in the taxonomy, at least one attack must be classified under each characteristic and dimension. An unchanged iteration culminates the taxonomy development exercise.

4.3.3 Taxonomy development

The taxonomy is used to examine the capabilities of malicious firmware and the firmware interactions required to achieve attacker goals.

4.3.3.1 Firmware interactions

A 3D printer firmware interacts with both cyber and physical entities. In the cyber domain, malicious firmware may interact with various hardware components and software applications, including the embedded system processor, RAM, EEPROM, storage elements, slicing software, and network nodes. In the physical domain, firmware interacts with the printing bed, printhead, filament, and printing environment through various sensors and actuators.

We classify cyber domain interactions into four categories: internal hardware, bootloader, human-machine-interface (HMI), and network. The internal hardware' category refers to the hardware that is not removed during normal operation, excluding peripherals such as LCD panel and SD-card. The 'HMI' category includes actors involving human interactions for printing purposes, such as the printer control application, LCD panel, and SD card. The 'network' category includes all network and application layer interactions that malicious firmware can perform, excluding core printing functions covered in the 'HMI' category.

Physical interactions involve various sensors and actuators that use kinetic and thermodynamic processes to interact with the physical process of printing. For comprehensiveness, we also use ‘secondary actuators and sensors’ to describe all other sub-processes, including environmental sensors and actuators.

4.3.3.2 Attack goals

This section covers the top-level goal selection, whereas the detailed tree is explained in the next section. We define ‘attack goal’ as a technological direct outcome of a set of planned malicious actions taken by an attacker. This definition excludes three of the four objectives (thrill, political or financial gain) defined by Howard et. al [66] or socio-political-economic-cultural (SPEC) factors presented by Gandhi [67]. Howard also includes ’damage’ as an objective but as it complies with our definition, it is taken as an attack goal. We start the exercise with the conventional CIA triad, except for rewording them as surveillance, denial of service, and integrity breach to reflect the attack goals. The literature review highlights multiple AM attacks in each category. Although these attacks are not performed at the firmware, a conceptual analysis suggests their feasibility through malicious firmware. A narrow dimension of illegal printing’ is reported in the literature [10] referring to printing weapons or other illegal objects. We consider any object printed without considering the printing process owner approval is considered as ‘unauthorized printing’ and is a feasible firmware attack goal.

4.4 Categorization tree of firmware attack goals

Continuing with the same methodology and set of assumptions mentioned in Section 4.3, we develop and present in Figure 35 a detailed tree of goals an attacker can achieve through malicious firmware. As we scroll down the tree, the attack goals and the subsequent firmware interactions get more specific. To facilitate conciseness

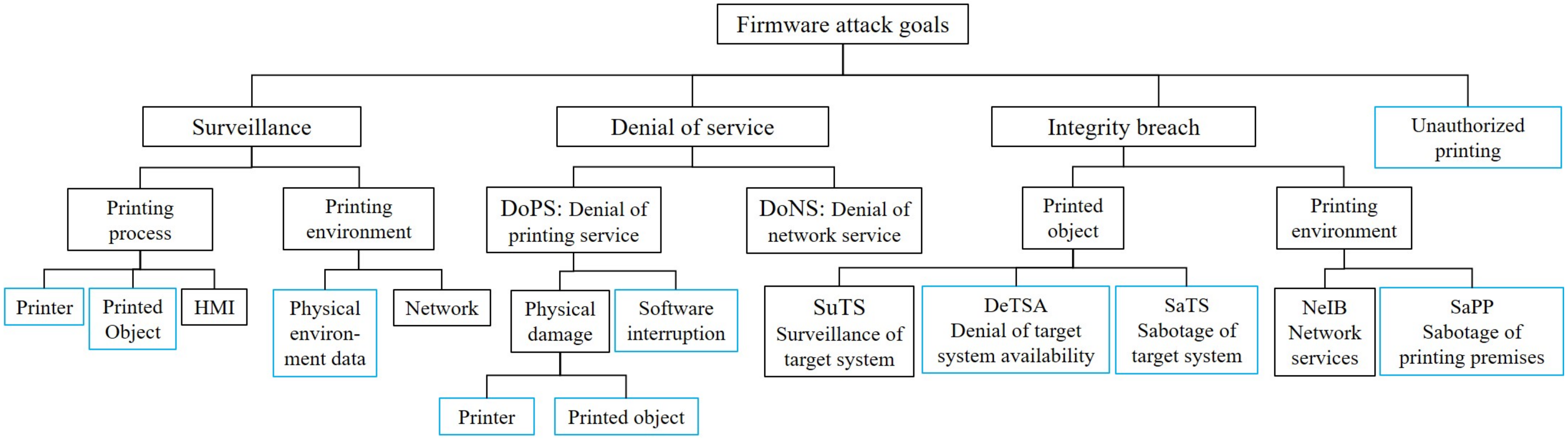


Fig. 35: Categorization of firmware attacks (categories in blue are covered through proposed attacks)

and comprehensiveness, the depth of the tree is not explored after the printed object, the printing process, and the printing environment are segregated under a common parent node. For instance, physically damaging the printing bed or the printing nozzle is considered as one attack goal, ‘physical damage to printer’.

4.4.1 Surveillance

Surveillance attacks do not modify the printing process but aim to steal information about the printing facility or the printing process. The printing process surveillance includes spying on the printer, the printed object, and the printer controller. In Section 4.6.1, we present an attack to steal the sketch of the object’s shell for a few selected layers in only 256 bytes to conveniently fit in the non-volatile memory element for retrieval at a suitable time.

A business owner is interested in knowing the types of prototypes being printed in the research lab of the competitor, resulting in the disclosure of business secrets and/or intellectual property (IP). Surveillance can also help in future attack planning wherein the information pertaining to the control software and the printer can help in fingerprinting the system or generating an attack pertinent to print geometry. Hacktivists may target classified premises for surveillance purposes. 3D printers installed with cameras, temperature gauges, and other sensors can be used to steal physical surveillance data from the printing facility.

The printing environment surveillance also includes conventional confidentiality attacks on networked devices. For example, sensors like the camera and the temperature gauges installed on many printers can reveal the printing facility details.

4.4.2 Denial of Service-DoS

Malicious firmware is capable of achieving DoS goals through cyber or physical domain corruptions to the printing process. The two categories based on these domains are denial of printing service-DoPS and denial of network services-DoNS.

4.4.2.1 Denial of Printing Service-DoPS

DoPS is achieved by causing physical or software interruptions in the core printing process. These interruptions can be achieved through physical damage to the printer or printed object, or through software-based attacks that pause or terminate the printing process.

Physical Damage for DoPS Attacks that cause physical damage to the printer or printed object fall under this category. These attacks may involve damaging an important part of the printer or printing obviously defective or completely different parts to ensure rejection. Two interesting attacks in this category, ‘Print your own grave’ and ‘Incurable’, are proposed and demonstrated in Section 4.6. ‘Print your own grave’ is used to break the printing bed glass by throwing it out of the printer. A third type of physical damage to printed objects includes specialized defects that lead the user to troubleshoot the printing environment. Malicious firmware can cause common printing problems like stringing, poor bridging, z-wobble, warping, etc., which mislead the user and result in a waste of time and resources in lengthy and futile troubleshooting sessions.

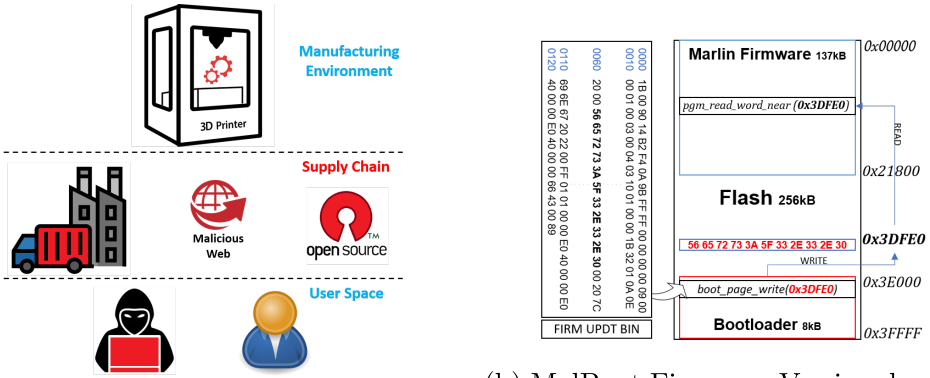
Software Interruptions If an attacker does not intend to cause physical damage, they may achieve DoPS by pausing or terminating the printing process through various software-based attacks at the firmware level.

4.4.2.2 Denial of network services-DoNS

In this category, malicious firmware targets the networked device or service of the printer to achieve a conventional DoS attack. The firmware can launch a network or application-layer flooding attack to flood the network with traffic and cause a denial of service. Alternatively, a planned attack may exploit a vulnerability to crash a high-priority target on the printer's network, disrupting the overall network communication.

4.4.3 Integrity breach

This category comprises unconventional attack goals that offer high dividends to the attacker. It is further divided into two sub-categories: 'integrity breach of the printed object' and 'integrity breach of printing environment'.



```
0000  1B 00 90 14 B2 F4 0A 9B FF FF 00 00 00 00 09 00
0010  00 01 00 03 00 04 03 10 01 00 00 1B 32 01 0A 0E
0060  20 00 56 65 72 73 3A 5F 33 2E 33 2E 30 00 20 7C
0110  69 6E 67 20 22 00 FF 01 01 00 00 E0 40 00 00 E0
0120  40 00 00 E0 40 00 00 66 43 00 89
```

(a) MalBoot AM Supply Chain Attack Surface

(b) MalBoot Firmware Version doctoring

Fig. 36: MalBoot Attack Surface and false version update in Flash Memory

4.4.3.1 Integrity breach of the printed object

Manipulation of the printed object can serve two distinct goals. If the objective is to generate obvious defects beyond the acceptance window, it is considered a DoS attack. If the attacks create defects that go undetected during the printing and post-printing process and ultimately impact the target system, they qualify under this category. The magnitude of loss incurred by these attacks depends on the nature of the target system where these objects are employed. For instance, introducing hidden defects in a power-plant turbine propeller can abruptly fail it during operation, causing physical damage. This category is further divided into three main types as defined below:

Surveillance of target system - SuTS Although no attack has been demonstrated in this category yet, the literature suggests it is feasible to include some spying capability in a 3D-printed object. For instance, printing an RFID tag may reveal the precise location of the printed part [68]. As 3D printing incorporates a variety of raw materials, we expect to see more research in this domain in the near future.

Denial of target system availability - DeTSA The aim of DeTSA attacks is to cause DoS at the target system. For instance, ”fitment attacks” induce a scaling or alignment error in specific parts that may disallow their assembly in the overall system. We demonstrate one DeTSA attack in this study. Attacks that cause early failures may also be driven by the DeTSA goal.

Sabotage of target system - SaTS These attacks may look similar in attack actions to DeTSA, but they have distinct goals. Although service denial is also an added consequence, SaTS attacks are designed to cause damage to the target system, not just the part’s failure. The research community has proposed and discussed these attacks at the pre-firmware stages [12, 69], and we demonstrate their feasibility at the firmware level.

4.4.3.2 Integrity breach of printing environment

The printing environment includes the physical components in the printing facility and the cyber components on the network. Malicious firmware acts as a rogue network element, and depending on the network features supported by the attacked printer, an attacker can launch a variety of integrity attacks on the networked devices. In the physical space, integrity attacks include physical damage to the printing facility. The 3D printing process typically involves high temperatures up to a few hundred Celsius. Through malicious firmware, an attacker can bypass the safety limits and the control mechanism to indefinitely operate the heating elements causing a fire hazard. As researchers are working to improve the flame retardancy characteristic of the filaments [70], an attacker can exploit the material's flammability to cause a fire incident. Another health hazard relates to the tiny particles and volatile organic compounds (VOC) released in the air during printing, potentially increasing the risk of respiratory, cardiovascular, and other disorders [71, 72]. Malicious firmware can increase the emission levels in the printing facility for prolonged periods, increasing the health risk for the exposed workforce.

4.4.4 Unauthorized printing

If an attacker wants to utilize the printing facility to print something without the competent authority's approval, they can accomplish it through malicious firmware. Although it seems like a straightforward task to feed a few static Gcode files, our set of assumptions (refer to Section 4.3.1) may pose a challenge in finding storage for the Gcode files. However, there can be interesting scenarios in this category, for instance, firmware printing an offensive or threatening message for the printer owner.

4.5 Attack Execution

Our firmware attacks follow a three-step approach: updating the firmware to meet malicious intent, installing it on the targeted printing device, and making the attack persistent. We use supply chain attacks to install the malicious firmware and add a malicious bootloader, MalBoot, to the printer to ensure persistence. More details on this approach are provided in the following subsections.

To install the malicious firmware, we consider an AM supply chain. There are multiple factors that can lead to the compromise of the system as shown in Figure 36a. Malicious actors in the distribution or manufacturing facility with physical access, or a naive or disgruntled employee downloading firmware from a malicious site and installing it on the printer can compromise the system. Performing firmware updates is an elementary task to maintain and update system functionality. However, it could attract adversarial attention, making it a potential target to exploit. Moreover, open-source firmware while attracting more contributors can leverage the attacker to find and exploit critical bugs. Therefore, updating such firmware from a malicious source can potentially lead to damage or compromise of the device, including the print or printer itself.

4.5.1 MalBoot

In order to achieve attack persistence, a slightly modified version of the bootloader program called MalBoot has been designed and implemented. USB-connected printing devices supported by Marlin typically use a small bootloader program for firmware updates. MalBoot has two main objectives: first, it manipulates critical bootloader functionality to perform new firmware updates that maintain the attack persistence. Second, it maintains normal communication with the user, including interactive and networking features, to avoid raising any alerts that could potentially fail the attack. Additionally, MalBoot provides the adversary with the tool to perform malicious firmware updates.

The bootloader uses stk500v2 [73] protocol implemented over serial communication. The protocol defines a set of commands communicated between the device and the host application. Through careful examination and understanding of these commands, we were able to make key observations, leading to the development of MalBoot. Below, we provide a brief overview of the sequence of commands that are exchanged during the firmware update process.

1. The host sends a restart signal to the device. Wherein, the device upon receiving the signal waits for the incoming commands and goes to the program execution state if no command is received.

2. The host sends in $CMD\_PROGRAM\_FLASH\_ISP$, transitioning the device to programming mode to write the flash. The device sends $STATUS\_CMD\_OK$ along with the calculated CRC.

3. The host sends a $CMD\_LOAD\_ADDRESS$ to move the pointer to the next address in memory.

4. The above two steps are repeated until the complete firmware code is copied onto the flash memory.

5. The host sends in $CMD\_LEAVE\_PROGMODE\_ISP$, indicating the device to go into the program execution state.

Following observations were made that could help attacker achieve their adversarial goals. First, the host system does not authenticate the program (firmware) being updated in the memory. Secondly, the host system only validates the CRC value to confirm payload integrity. The bootloader updates the new firmware binary in the flash memory being transferred in pages of fixed sizes (256 bytes). While in programming mode the bootloader first erases the flash and copies the new binary page by page onto it. The Malboot changes that functionality where it keeps the previous version of the firmware intact and the new firmware update is not pushed to the flash memory. The Malboot, however, accurately calculates the CRC value and responds to the host system as normal.

Additionally, during each firmware upgrade, Malboot searches for the newest version of firmware and passes it along to the running program, ensuring that the user is always presented with the latest version on the screen. The firmware version information is stored in the flash memory above the bootloader code at the offset address ($0x3DFE0$). When the firmware is executed, it searches for the firmware version at this offset address read the content using $'pgm\_read\_word\_near'$ function, and updates the relevant variable to display the new version to the user. Figure 36b presents the complete scenario of how Malboot performs the firmware version doctoring. Another associated challenge was to enable adversaries to push malicious firmware updates, but the Malboot functionality denies it. Also, downloading the bootloader erases the flash content needing a fresh firmware update. To cater to that Malboot adds an external trigger to the program whereby if set to high it enables the adversary to update malicious firmware at will.

It is pertinent to note that implementing authentication and strong validity checks during firmware updates would prevent this attack, but the current printer firmware lacks such mechanisms, leaving them vulnerable to these types of attacks.

4.6 Proposed firmware attacks and implementation

This section proposes nine novel attacks from the attack tree highlighted in blue in Figure 35. Conventional network-related attacks, such as MiTM were skipped in this case study. All the attacks are implemented on the open-source Marlin firmware, widely used in commonly available printers in industrial settings.

4.6.1 Stealing printed object geometry

Primary attack tree: Surveillance → P. Process→ Object

Secondary category: (1) Surveillance → P. Process→ Printer

(2) Surveillance → P. Env.→ Env. data

Outcome: Capture and transfer the printed object geometry

Method: The Marlin firmware running on an embedded system has limited storage, making it difficult to save the large G-code files that represent printed objects. However, malicious firmware can record the potentially useful instructions by developing a small engine that efficiently identifies and captures the sketch of the printed object using three approximations: (1) ignoring the complete infill structure, (2) truncating the sub-millimeter part of x,y coordinates, and (3) activating once per mm of z-axis movement.

To address the challenge of identifying the outer shell of the printed object, a circular buffer with sufficient length to accommodate the vertices of the object is introduced. For each layer, the shell is printed at the start or end, and a shell identification algorithm is employed on the ring buffer at the layer-change event.

In a case study of a printer with a printing-bed dimension of less than 255 x 255 mm, one byte is sufficient for each axis position data after converting to binary format. The engine captures the approximate shape of an object in only 256 bytes and saves it

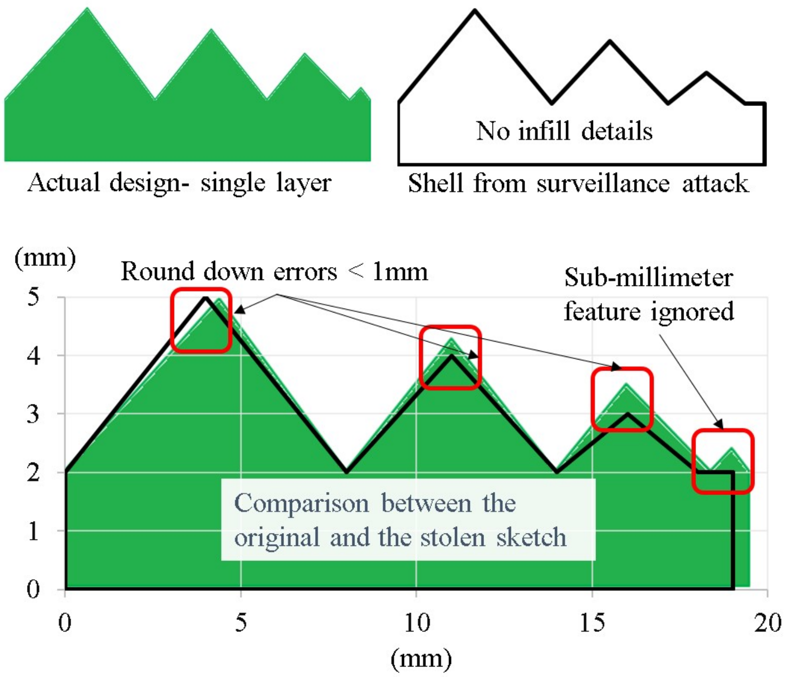


Fig. 37: Printed object outline stolen through surveillance attack

in EEPROM. Once an attacker inserts an SD card in the printer, the firmware verifies it and downloads the stolen information within 5 seconds. The complete attack flow is outlined in Algorithm 2 in the Appendix.

In addition to preserving the object geometry, two variants of this attack note printer hardware configuration information and the environment data (such as the environmental temperature) through physical sensors. The algorithm for these attacks is presented in Appendix 2.

The results and evaluation of this approach are presented in Figure 37, which shows an actual design (green), the stolen outline sketch, and a superimposed image highlighting approximation errors. Although the sub-millimeter features are ignored, the sketch provides valuable information about the object's shape and size to the adversary.

4.6.2 Print your own grave - PYOG

Primary attack tree: DoS → DoPS → Physical damage→ Printer

Secondary category: (1) DoS → DoPS → Software interruption

(2) Unauthorized printing

Outcome: Break the printing bed glass sheet

Method: PYOG attacks utilize the printing function to damage the printing setup. This attack breaks the printing bed's glass sheet by throwing it out of the printer. Exploiting the nonexistence of a hardware protection layer between the printing bed and the nozzle, we attempted to break the glass by overriding the firmware checks and hitting the bed against the nozzle. The approach does not provide enough impulsive force to break the glass that resides securely over the metal bed. To overcome this bottleneck, the firmware prints a destruction tool, grabs it with the nozzle, waits for it to cool down, and then conducts an intelligent scan on the edges of the printing bed to compromise the retaining clips installed (usually) at the corners. Finally, the malicious code pushes the glass from the rear edge to throw it out of the printer. The attack can be triggered by a specific instruction or an inactivity period and covers two additional categories during execution. The first category is software interruption, which is achieved by introducing a planned pause and not accepting any printing commands during that time to allow the destruction tool to cool enough for detachment from the printing bed. The second category is unauthorized printing, which is achieved by printing the destruction tool. The algorithm for this attack is presented in Appendix 3.

Results and evaluation: Figure 38 presents a pictorial view of the attack sequence from A to E. The attack utilizes only 20 lines of code to print the tool, which would normally require over 27,000 G-code instructions and more than 500 KB of space.

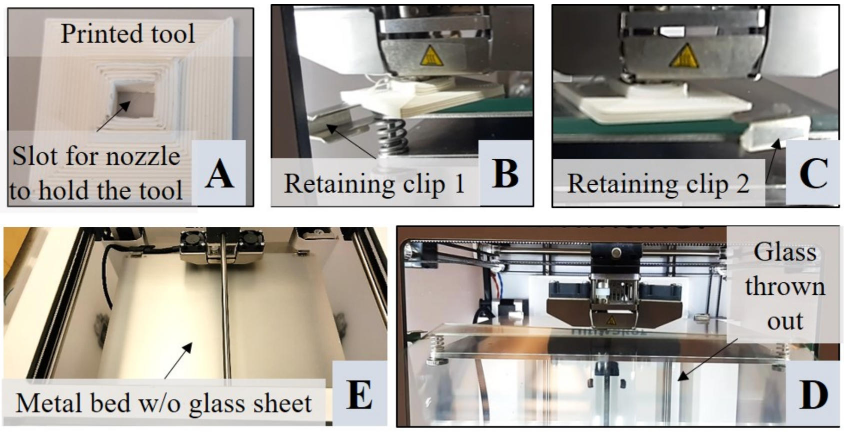


Fig. 38: Glass-breaking attack stages:

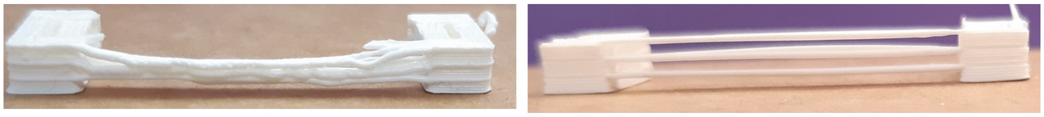
The entire attack code fits well within the available flash memory by only increasing the firmware size from 130 KB to 134 KB.

4.6.3 Incurable - Printing faults imitation attack

Attack tree: DoS → DoPS → Physical damage → Printed part

Outcome: Imitate poor bridging problem

Method: By imitating common 3D printing problems, the 'Incurable' attack deceives users into spending significant time and effort on futile troubleshooting exercises. This attack focuses on the poor bridging problem, which tests a printer's ability to extrude filament between two raised points without sagging. An extrusion instruction from $A_{x,y}$ to $B_{x,y}$ in i^{th} layer will belong to a bridge if there is no extrusion between $A_{x,y}$ and $B_{x,y}$ in $(i\text{-}1)^{th}$ layer. To identify a bridge, the attacker needs to maintain spatial information of the current and the previous layer. To ensure uninterrupted printing, the attacker cannot analyze and map detailed printing instructions on a compute-constraint system. Hence, the attacker uses a coarse approximated representation of a 100 mm x 100 mm targeted zone by only a 5 x 5 elements array (named as layer-map), where each element represents a square of 20 mm x 20 mm.



Imitating poor bridging error

Normal bridging performance

Fig. 39: Bridging error imitation attack

The bridging performance is typically evaluated over 20 mm and beyond [74]. When a move instruction is received, the layer-map is updated, and once a layer is completely printed, it is saved to identify any bridges in the next layer. For each move instruction, the attacker checks if there was any extrusion at the corresponding location in the previous layer. If there was no extrusion, the move instruction is categorized as part of a bridge. To create poor bridging performance, the attacker modifies several parameters, including slowing down the cooling fan, increasing the extrusion amount, and reducing the printing speed. See Appendix 4 for the complete algorithm

Results and evaluation: We conducted an evaluation of the attack by printing a shape with 3 bridges across 25 mm apart pillars, with each bridge being 5 layers apart from the previous one. As shown in Figure 39, the attack successfully imitated poor bridging performance. The sag visible on the 25 mm gap between the pillars could lead the user to believe that there are printing setting problems related to poor bridging.

4.6.4 Two-dimensional scaling attack

Attack tree: Integrity → Printed object → DeTSA

Outcome: Modify dimensions to disallow fitment

Method: DeTSA aims to disrupt the availability of the system for which a part is being printed by slightly modifying the object dimensions. This task can be easily accomplished at the designing or slicing stage using the 'scaling' switch. However, since the firmware receives printing instructions in temporal sequence, it cannot plan for perfect scaling in advance. Therefore, achieving perfect 3D scaling at the firmware level is not feasible, unless it learns from one object and attacks another. Additionally, 2D scaling is also not trivial at the firmware level. As shown in Figure 40, scaling of printing instructions can also lead to the scaling of connecting segment instructions, creating gaps between consecutive extrudates that expose the attack.

To execute a scaling attack through firmware, we exploit the conventional printing format. The cross-sectional view of the internal layers of an object includes the infill pattern and walls structure. The outer walls mark the edges of the object and create a directed cycle where the destination coordinates for a move instruction are repeated after 'k' instructions. The walls structure is printed adjacent to the layer change event.

The attacker identifies the directed cycle to find a geometrical feature and modifies the object dimensions by adding an extra wall around it. Under generic printing settings, wall thickness is proportional to the nozzle diameter, which implies a 0.8 mm to 1.2 mm difference in dimensions across the two opposite walls for 0.4 mm and 0.6 mm nozzles. Due to memory constraints, it is not possible to track the destination coordinates of all move instructions. To overcome this problem, the attacker creates a circular buffer that can contain one extra entry than the maximum number of edges in the anticipated polygon. To manage the limited computation power, the attacker uses the change-of-layer instruction to trigger the polygon identification routine. Once the polygon is identified, the attacker selects an appropriate point outside the object and prints a new polygon by adjusting the sequence of coordinates in the identified cycle. The algorithm is presented in Appendix 5.

Results and evaluation: For evaluating this attack, we printed a rectangular prism with dual sizes. Figure 40 displays the original and attacked samples. Although

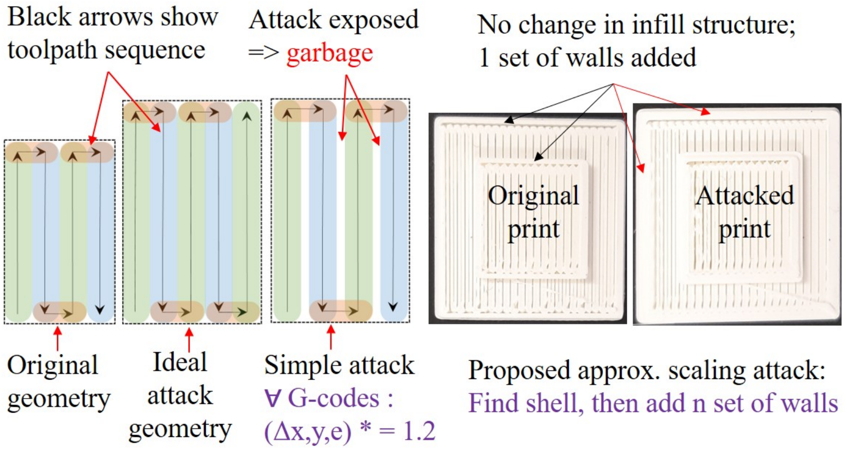

Black arrows show toolpath sequence
Attack exposed => garbage
No change in infill structure; 1 set of walls added
Original print
Attacked print
Original geometry
Ideal attack geometry
Simple attack
∀ G-codes : (Δx,y,e) * = 1.2
Proposed approx. scaling attack:
Find shell, then add n set of walls

Fig. 40: Fitment attack through approximate scaling

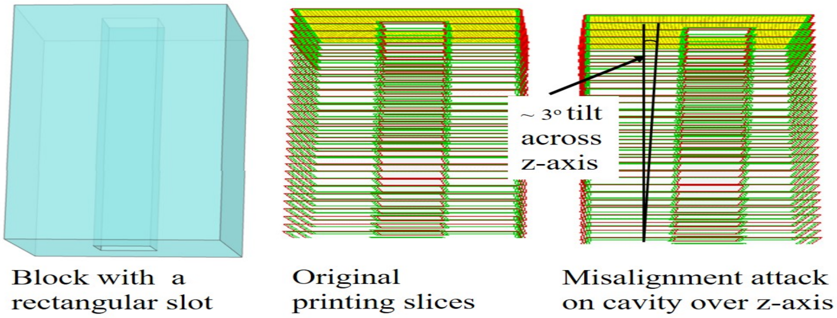

~ 3° tilt across z-axis
Block with a rectangular slot
Original printing slices
Misalignment attack on cavity over z-axis

Fig. 41: Geometric feature misalignment attack

there are no noticeable changes in the infill structure, an additional set of walls can be observed in the attacked sample. We measured the distance between opposite edges at five different locations and found that there was an average increase of 0.96 mm ± 0.25 mm for each dimension.

4.6.5 Axial misalignment of interfacing feature

Attack tree: Integrity breach → Printed object → DeTSA

Outcome: Minute axial misalignment to disallow fitment

Method: A fitment error can also be achieved by maliciously modifying an important geometrical feature across the build orientation (z-axis). In this attack, the G-code execution pipeline is delayed by k_{max} printing instructions to ensure that the attack circular buffer is filled before execution. The circular buffer described in Section 4.6.4 is used by the attacker. Once a change-of-layer event occurs, either the x or y coordinates of all vertices of the directed cycle are slightly modified proportionally to the current z-axis value to achieve a continuous drift. Objects with precise fitment requirements, such as driving shafts, driven assemblies, nuts, and bolts, are likely targets of this attack.

Results and Evaluation: This attack was implemented on a rectangular female square-fitting slot that interfaces with a male driving shaft. As presented in Figure 41, the attack introduces a 3^o axial shift that prevents interfacing with the male shaft. Unlike Section 4.6.4, this attack achieves the goal without adding to the number of printing instructions.

4.6.6 Internal cavity

Attack tree: Integrity breach→ Printed object → SaTS

Outcome: Introducing an internal cavity to cause early failure

Assumption: This attack assumes that the geometry of the targeted object is symmetric over the z-axis. This assumption holds true for ASTM tensile and flexure models, and for other objects it is valid over blocks of layers.

Method: Introducing cavities inside a printed object is a well-known attack that can be carried out at the designing and slicing stages [13, 75]. If a new geometric feature is introduced at the design stage, the overall toolpath sequence is redesigned by the slicer, resulting in a bigger attack footprint. On the other hand, if a cavity is introduced at a post-slicing stage, it is not very precise but the attack footprint is negligible. In this study, we implement a filament-kinetic cavity attack to achieve the SaTS goal. As the attacks are launched in intermediate layers, the attacker uses the initial layers to learn about the object dimensions and decide on the cavity size and location.

The attacker counts the total number of G-code instructions in the (i-1)th layer and attacks the central instructions. To ensure that the cavity remains hidden, the attacker splits the instruction into three parts and mutes the filament motor for the central part only. The attack continues for ten layers to implant a cavity that is consistent and effective in reducing the printed object's performance. If the attack code determines that the instructions are part of the skin rather than the infill layers, the attack may cease before reaching 10 layers to ensure that the cavity remains concealed.

Results and evaluation: To evaluate the attack, we printed two ASTM-compliant tensile bars. Figure 42 illustrates the cavity in the top image, taken after pausing the printing process. In the bottom image, we can see that the cavity is concealed in the final print by the top and bottom layers.

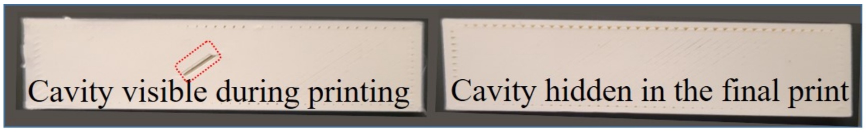

Cavity visible during printing
Cavity hidden in the final print

Fig. 42: Cavity attack specimen during and after printing

4.6.7 Object density variation

Attack tree: Integrity breach → Printed object → SaTS

Outcome: Reducing object density

Assumption Same as in Section 4.6.6

Reducing the object density was previously explored at a pre-firmware stage [8]. This attack implements a localized filament density variation attack through malicious firmware. Unlike cavity attacks, it does not create any visual deformation during or after printing. The attack preparation steps are the same as those described in Section 4.6.6, with two slight changes. Firstly, the zone of interest is increased, and secondly, retract instructions are not required. Instead of changing the filament state, the attack manipulates the filament motor speed to reduce the amount of material in the targeted zone.

Results and evaluation: We printed six ASTM-compliant tensile bars using both the original and attacked firmware and observed no visual or dimensional differences between the two sets of prints. Tensile tests were then conducted using the MTS Insight 30 machine, and the results are presented in Table 19. The data shows a significant reduction of 15.84% and 17.66% in the peak tensile load and stress values, respectively, when comparing the tensile strength of the attacked prints to those printed using the original firmware.

Sample type	**Peak load (N)**		**Peak stress (N)**	
	Avg of 6 samples	Std. dev	Avg of 6 samples	Std. dev
Original	498.44	39.65	15.38	1.17
Attacked	419.49	24.54	12.67	0.75
Difference	**78.96**		**2.72**	
%age reduction	**15.84**		**17.66**	

Table 19.: Tensile test results for filament density attacks

4.6.8 Filament-erosion attack

Attack tree: Integrity breach→ Printed object→ SaTS

Outcome: Reducing object density

Method: While the desired outcome is similar to that in Section 4.6.7, this attack employs a different and indirect method. In most fused filament fabrication (FFF) printers, sharp teeth on the extruder motor shaft grip the filament with the support of a free-rotating roller (see Figure 43). As the motor rotates, the teeth push the filament axially towards and subsequently out of the heated nozzle in a molten state.

In this attack, a portion of the filament is eroded away as it passes through the feeding chamber, which reduces the quantity of filament at the point of attack. The defected (eroded) filament portion then passes through the nozzle, creating low-density zones in the printed object. A carefully planned filament erosion attack can thus lower the material density at a critical region, reducing the strength of the printed object. Figure 43 illustrates an example of a filament erosion attack.

There are two ways to conduct this attack. The first method involves forcing cold extrusion through the nozzle, which erodes the filament. However, this method requires bypassing the firmware check for the minimum temperature required for extrusion. Alternatively, a burst of high jerk oscillatory movements can be applied

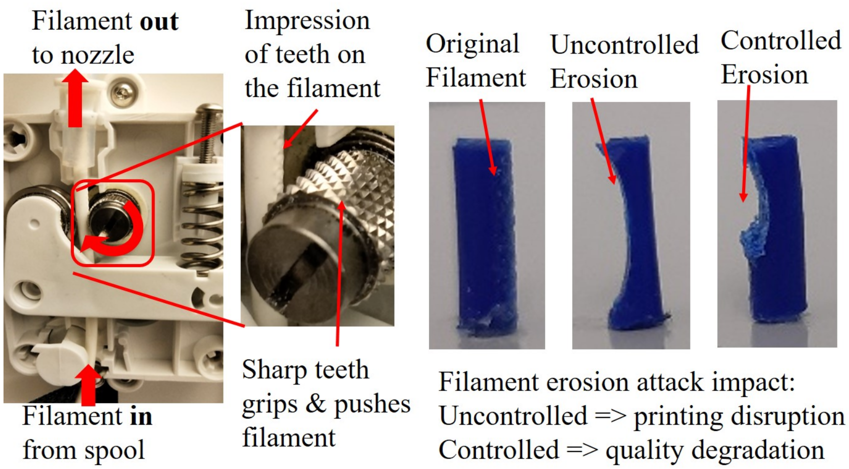

Filament **out** to nozzle
Impression of teeth on the filament
Original Filament
Uncontrolled Erosion
Controlled Erosion
Sharp teeth grips & pushes filament
Filament **in** from spool
Filament erosion attack impact:
Uncontrolled => printing disruption
Controlled => quality degradation

Fig. 43: Filament erosion attack

to break the grooves formed due to gear pressure, which erodes the filament without requiring the nozzle to cool down. This method causes less erosion but still achieves the defective printing goal.

Regardless of the chosen method, it is important to note that there exists an upper limit for filament erosion beyond which the extrusion system can no longer push the filament forward. As a result, the attack may transition from an integrity breach to a denial of service attack.

Results and evaluation: The effectiveness of the filament erosion attack was evaluated by conducting two attack instances. Using PLA filament of 2.85 $pm0.1mm$ diameter, it was observed that excessive attack caused a reduction in weight from 0.077 grams to 0.049 grams, resulting in a 36% reduction. However, the filament erosion was substantial enough to cause a failure in further pushing the filament, thereby exposing the attack. On the other hand, a fast-to-execute and low-magnitude attack resulted in a 15% reduction, with the equivalent length of filament weighing 0.065 grams.

4.6.9 Printing facility air quality degradation

Attack tree: Integrity breach → Printed object → SaTS

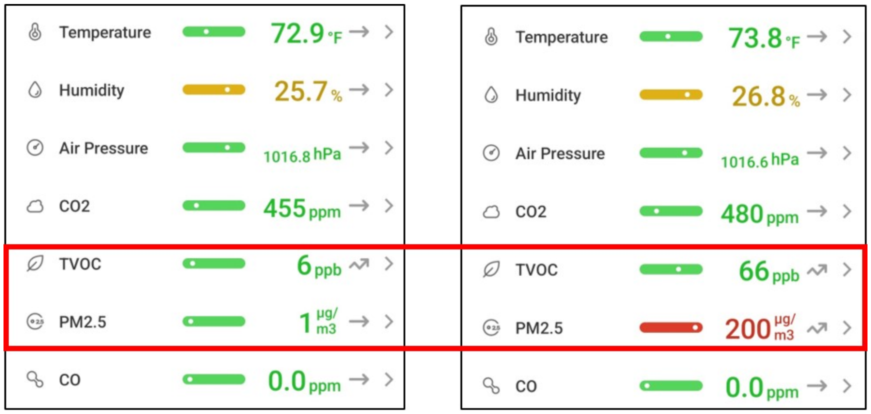
Outcome: Inducing health hazard

Method: The objective of this attack is to intentionally degrade the air quality within a printing facility by increasing the emission of microparticles and VOCs from the printer. To achieve this goal, the attacker employs malicious firmware that performs two main actions. Firstly, it configures and monitors a no-activity timer to initiate low-temperature scrubbing activity, which increases the emission of microparticles into the air. Secondly, it switches the nozzle heater ON after disabling the temperature feedback control circuit, leading to the emission of excessive fumes or VOCs. As a result of the high filament temperature, the low-density fluid drops down from the nozzle, leaving suspicious droplets on the printing bed.

To avoid fluid dropping and to increase vaporization, the attacker retreats the filament, leaving a small quantity in the chamber before raising the temperature to the extreme zone. The reduced filament quantity also decreases the attack time. It should be noted that this attack poses a severe threat to the printing facility's environmental health and safety, as it can cause harmful effects on the workers' health and the facility's equipment.

Result and evalution In order to evaluate the impact of the attack, a series of experiments were conducted to measure the particles and VOCs count before and after the attack. Specifically, measurements were taken 5 minutes before the attack, as well as 5 minutes, 30 minutes, and 1 hour after a single instance of the attack. This experiment was repeated 5 times, with the effects of the previous attack dismissed prior to each subsequent trial.

The results of these experiments are presented in Figure 44. The data indicate


Temperature 72.9 °F
Humidity 25.7 %
Air Pressure 1016.8 hPa
CO2 455 ppm
TVOC 6 ppb
PM2.5 1 µg/m3
CO 0.0 ppm

Normal printing

Temperature 73.8 °F
Humidity 26.8 %
Air Pressure 1016.6 hPa
CO2 480 ppm
TVOC 66 ppb
PM2.5 200 µg/m3
CO 0.0 ppm

Under attack

Fig. 44: Air quality stats for facility contamination attack

that the VOCs count increased from 6 ppb to 66 ppb, while the $PM_{2.5}$ value increased from 1 $\mu g/m^3$ to 200 $\mu g/m^3$ against the safe limit of 10 $\mu g/m^3$ [76].

4.7 Attacks feasibility and complexity analysis

4.7.1 Motivation

The attacks presented in Section V exhibit a range of distinct workload requirements, depending on the specific attack goal in question. Notably, certain attack goals may be accomplished prior to the firmware stage, thus warranting a comprehensive feasibility analysis of all attack goals across various stages of the additive manufacturing process. To this end, we conducted a meticulous examination of the feasibility of each attack goal at different stages of the process, in order to provide a detailed and comprehensive assessment.

4.7.2 Methodology

To begin our analysis, we undertook a thorough examination of the additive manufacturing process to identify the various independent stages that could be targeted

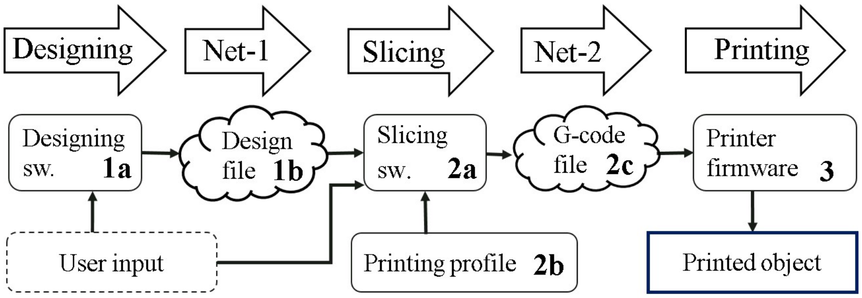


Fig. 45: Stages in AM process highlighting cyber artifacts compromisable through a cyberattack

by potential attacks. We then proceeded to develop a comprehensive set of feasibility and complexity criteria, which we used to assign feasibility scores for each stage, as outlined in Table 20.

In order to accurately evaluate the feasibility of attacks at the firmware stage, we relied on the attack data previously presented in Section 4.6. For other stages, we drew upon relevant methodologies and findings presented in the literature [12, 31, 8, 77, 38], which allowed us to develop a comprehensive understanding of the various factors that contribute to the feasibility of attacks at each stage of the additive manufacturing process.

4.7.3 Printing process stages

We examine the printing process to identify independent attackable stages. Figure 45 shows the core stages. We consider 1a' and 1b' as a single stage as an attacker who captures the 3D model file can perform the same set of attack actions available by compromising the designing software. Intruding into the slicer software provides a different set of capabilities to the attacker than attaining the G-code file through the network Net-2,' so we retained both stages. The printing profile selected by the user is an important input at the slicing stage, and an attacker can hack and modify it to perform a different set of attack actions. Therefore, 2a, 2b, and 2c are analyzed as separate stages. Finally, 'printing' is the last stage that a cyberattacker can directly compromise by modifying the firmware, without affecting any previous stage.

4.7.4 Feasibility and complexity criteria

The study considers two factors in determining the feasibility and complexity of an attack at a particular stage. The first factor is the interpretability of the attack criteria, including but not limited to the spatiotemporal location and the thermal profile. The second factor is the implementation complexity associated with launching the attack. At any particular stage, an attack is considered:

- an infeasible attack if there is no mechanism available to execute the attack actions, or there is no way to confirm compliance with the attack criteria.
- a low-difficulty attack if the information needed to identify the attack compliance and the mechanisms to launch the attack are readily available
- a medium-difficulty attack if the stage offers ready compliance to the attack criteria or the implementation mechanism but not both
- a high difficulty attack if both the factors are not readily available and require additional workload to estimate or calculate them.

For instance, the spatial location information required in dynamic-thermal attacks [8] is readily available in the G-code file and there is a G-code instruction to manipulate the nozzle temperature. Hence, the dynamic-thermal attacks are considered 'low difficulty' attacks at the G-code or slicing stage. Assuming f_1 and f_2 are the attack compliance and the implementation feasibility factors with possible values of 0,1,2 for 'Not available', 'Not readily available', and 'Readily available' respectively, then the feasibility of attack n at stage m is defined as:

$$FS_{n,m} = \begin{cases} Infeasible & \text{if } f_{1_{n,m}} \cdot f_{2_{n,m}} = 0 \\ High\ difficulty & \text{if } f_{1_{n,m}} \cdot f_{2_{n,m}} = 1 \\ Medium\ difficulty & \text{if } f_{1_{n,m}} \cdot f_{2_{n,m}} = 2 \\ Low\ difficulty & \text{if } f_{1_{n,m}} \cdot f_{2_{n,m}} = 4 \end{cases} \tag{4.1}$$

The following subsections discuss the important findings presented in Table 20 pertaining to the attack complexity and firmware interactions. The detailed assessment for 46 attacks and 5 stages is presented in Table 22 in the Appendix, and summarized in Table 21.

4.7.5 Designing stage

The designing stage focuses on the geometry of the desired object. It is the best stage to attack if the goal requires precise manipulation of geometric features. The fitment attacks, anisotropy attacks (for DeTSA), and geometric feature insertion or removal are the simplest at the designing stage. If the attacker has access to the designing software process, surveillance attacks on the printed object and the connected network are also feasible. For the rest of the attack goals, the designing stage is not the best bid for an attacker.

4.7.6 Slicing and control software

With an STL file and printing profile as the input and a G-code file as the output, the stage offers a vast spectrum of attack opportunities. It outperforms all other stages in achieving DeTSA and SaTS goals by compromising printed part integrity. However, the stage is less effective for 'fitment' and 'physical damage to the printer

Sr. No	**Attack action**	**Firmware interactions**	**Attack category**	**Attack complexity**
1	Printed object sketch recovery	I, H	Part surveillance	High
2	Print your own grave - PYOG	A, S	Physical damage for DoPS	Medium
3	Imitating poor bridging error	A, S	Garbage print for DoPS	Medium
4	2D scaling attack to fail fitment	H, A	DeTSA	Medium
5	Geometry feature misalignment	H, A	DeTSA	Medium
6	Internal cavity attack	H, A	SaTS	Low
7	Printed object density attack	H, A	SaTS	Low
8	Filament erosion attack	A, S	SaTS	Medium
9	Air quality degradation	A, S	SaPP	Low

H: HMI I: Internal hardware N:Network A:Actuators S:Sensors

Table 20.: Categorization of proposed attacks and their implementation complexity

for DoPS‘ attacks.

4.7.7 Printing profile

The printing profile comprises parameters used by the slicer software to attain a set of printing instructions, including layer thickness, nozzle temperature, printing speed, infill pattern, walls count, and top and bottom layers. With a few exceptions, these parameters globally apply to the entire printing process. One example of this exception is the flexibility to set a different temperature profile for the initial layers. By manipulating these values an attacker can launch an object-warping attack to achieve DoPS. For attacks requiring spatiotemporal precision, the printing profile is not a suitable stage.

4.7.8 G-code file through Net-2

The third artifact of the slicing stage is the G-code file. Researchers have demonstrated attacks by capturing and manipulating the G-code file through the MiTM attack on the network between the printer and the controller machine [39]. A G-code file does not offer all manipulation possibilities available through the slicing software. The chronological structure of a G-code file suits the introduction of localized defects to achieve DeTSA, SaTS, and DoPS. However, global modifications require an extra computational load for reverse-engineering the G-code file to a 3D model for applying global changes and then re-slicing it to get a modified G-code file.

4.7.9 Firmware

For incurring ‘physical damage to the printer’, and ‘printing interruptions’ for DoPS, the firmware stage leads all other stages. The firmware can also compromise the printing environment to degrade the quality of the printed parts. Malicious firmware can contaminate or damage the printing facility through micro-particles and VOCs emission, or through electric circuit abuse. However, firmware is not the optimal stage to launch the DeTSA and SaTS attacks discussed in Table 20 as the precision to achieve the required stealthiness is difficult to ensure at firmware due to its limited temporal view. The limited computational resources complicate launching integrity attacks on networked devices. In spite of different values for each category, coincidentally, they both have the same cumulative attack goal feasibility index of 0.66.

4.7.10 Attack goal feasibility index - AFI

To assess the feasibility of attacks at different stages of the printing process, we introduce the term "Attack Goal Feasibility Index" (AFI), ranging from 0 to 1. An AFI value of 0 indicates that an attack goal is not feasible at a particular stage, while an AFI value of 1 indicates low difficulty as defined in Section 4.7.4. The index incorporates the cumulative effect of all attacks in a particular category, and is calculated as follows:

$$AFI_{g,s} = \frac{1}{n \ * \ F_{max}} \sum_{i=1}^{n} \sum_{k=1}^{m} f_{k_{i,s}} \tag{4.2}$$

where $AFI_{g,s}$ represents the AFI value for the attack goal category g at stage s. n is the total number of attacks in the category g, F_{max} is the numeric value assigned to the 'low-difficulty' level, and $f_{k_{i,s}}$ represents the value for the k^{th} factor for the i^{th} attack at stage s.

Table 21 displays the Attack Feasibility Index (AFI) for the examined attacks, broken down by the stages presented in Figure 45. The AFI scores have been calculated using Equation 4.1 and then normalized to account for variations in attack counts across categories, as shown in Table 20. Notably, only the slicing software

Attack goal category	**Normalized attack feasibility index for process stages**				
	1.	**2a.**	**2b.**	**2c.**	**3.**
Printing process surveillance	0.5	0.8	0.2	0.5	0.45
Printing facility surveillance	0.25	0.25	0	0.25	0.5
Damage to the printer for DoPS	0	0.04	0	0.04	0.57
Garbage printing for DoPS	0.5	1	1	1	1
Printing interruption for DoPS	0	0.88	0	0.88	1
Fitment for DeTSA	1	0.25	0	0.25	0.25
Sabotage of target system - SaTS	0.25	0.89	0.25	0.61	0.56
Poor printing environment for DoPS	0	0.43	0.18	0.43	0.79
Sabotage of printing premises - SaPP	0	0.38	0	0	1
Unauthorized printing	0.25	1	0	1	0.25
Networked devices integrity - NeTB	1	1	0	1	0.5
Networked devices availability - DoNS	1	1	0	1	1
Cumulative attack feasibility index per process stage	**0.40**	**0.66**	**0.14**	**0.58**	**0.66**

1 : Designing 2a : Slicing software 2b : Printing profile 2c : G-code file 3: Firmware

Table 21.: Stage-wise feasibility summary for attack goals

(Stage 2a) and the firmware (Stage 3) demonstrate non-zero AFI values across all attack goals.

4.8 Firmware attack countermeasures

4.8.1 Cyber-physical security frameworks

Firmware attacks present a wide range of attack options that are difficult to detect compared to pre-firmware attacks. One way to detect most of the attacks discussed in Table 20 is by adding an appropriate validation function for the output of a stage. However, as the firmware output is a physical printed object, a cyber domain post-firmware validation option is not available. Researchers have proposed attack detection solutions based on monitoring the side channels emitted by the printer, such as acoustic signals, electric current, and magnetic fields [18, 19, 78]. However, these approaches can only handle sabotage attacks on printed objects.

4.8.2 Hardware design protection layer

As 3D printers are not designed with security in mind, some attacks exploit weaknesses in the hardware design to cause physical damage to the printer. An effective way to address this issue is to confine the firmware within the protected boundaries of well-designed hardware. For example, we observed in the case study that end-stops are only available towards one end of each axis, whereas the other end is vulnerable to malicious interventions and errors.

4.8.3 G-code queue monitoring and profiling

Most printer firmware, Marlin and its derivatives, runs on embedded systems with low computational and memory resources. The firmware maintains a small queue to receive G-codes from the control software. When a malicious firmware tries to calculate an attack, it temporally affects the G-code queuing profile. Attack detection solutions based on fingerprinting the benign G-code queuing profile can identify anomalous behavior of malicious firmware.

4.8.4 Acquire and verify the binary file

An intuitive mechanism to avoid firmware attacks is to verify the firmware code. We observed that a user can upgrade but cannot retrieve the firmware through the control software. An effective countermeasure is to include firmware uploading and verification feature in the control software. However, a sophisticated attacker can still modify the firmware and deceive the user by sending a good copy on inquiry while hiding the malicious code. A more reliable method to acquire the firmware copy is through hardware-based approaches, such as Joint Test Action Group (JTAG) acquisition [49], which can provide greater assurance of firmware code integrity.

4.9 Conclusion

This study presented a new approach to understanding and classifying firmware attacks in additive manufacturing. Specifically, we proposed a two-dimensional taxonomy that maps attack goals to firmware interactions and elaborated on these attacks by proposing an attack categorization tree. To demonstrate the feasibility of our approach, we conducted nine attacks on the Marlin firmware installed on a reputable 3D printer. Through a series of destructive and non-destructive tests, including tensile and air-quality testing, we confirmed the effectiveness of these attacks. Our study also revealed that not all attack goals can be achieved at every stage of the printing process, and we assigned a feasibility score to each goal for each stage. This analysis can guide researchers in developing defense solutions that are optimized for specific stages of the printing process and the set of feasible attacks at each stage. We believe that the proposed taxonomy and the attack feasibility analysis will motivate further research in this area, and help to improve the security of additive manufacturing.

4.10 Algorithms for Firmware Attacks

Algorithm 2 Printed object surveillance attack

```
Output: Object sketch file
Phase-1: Sketch compilation
On restarts: $*eeprom_{Attk_{end}}$ → spyFile
if G-code == G0 or G1 then
    if !spyFile then
        if L.Change() && $Z_{dst} \geq (E_{no} * Z_a + 1)$ then
            Shell ← FindShell()
            *eeprom$loc$ ← $L.Header$; loc++
            *eeprom$loc$ ← $Z_{dst}$; loc++
            for P ∊ Shell do
                *eeprom$loc$ ← $Px$; loc++
                *eeprom$loc$ ← $Py$; loc++
            end for
        else
            if $Z_{dst} == Z_{current}$ then
                Queue ← Queue ∪ P$x, y$
            else
                if printingDone() then
                    Queue.reset()
                    spyFile = 1
                    *eeprom$Atk_{end-1}$ ← $(loc - loc_o)$
                    *eeprom$Atkend$ ← $0x01$
                    Reset$Queue$, $loc$
                end if
            end if
        end if
    end if
end if
Continue-execution
Phase-2: File transfer
if (SDinserted) && (SDstateChange) then
    if (spyFile) then
        if (SDauthenticate()) then
            SD.openFile(”spidy.txt”, ‘w’)
            for i: 0 to *eeprom$Atk_{end-1}$ do
                SD.write(*eeprom($loc_o +\ i$))
            end for
            SD.closeFile()
            spyFile = 0
            *eeprom$Atk_{end-1}$ ← $0x00$
            *eeprom$Atkend$ ← $0x00$
        end if
    end if
end if
```

Algorithm 3 Print your own grave: Breaking the printing glass

```
Output:Breaking the printing glass
Trigger:An unused G-code G98
Preheat the printing bed and nozzle
$\forall$ layer $\in$ [1,n] where 'n' is desired no of layers
    [To oscillate the starting point]
    x = 112.5 +osc*0.1
    y=112.5+osc*0.1
    osc*=-1
    if layer > 8: line-count ← small-square
    else line-count ← big-square
    $\forall$ line $\in$ [1,m]   where 'm' is the no. of lines
        if line < 4 , speed = slow
        else if layer > 4, speed = moderate
            else speed = fast
        x=x+dir*lenx , y+dir*leny , e=e+dir*lene
        Move to (x,y,e)
        lenx+=0.8 , leny+=0.8 , lene = 0.058 * lenx
While(nozzle and printing bed cools down) wait!
Grip(printed-tool): Nozzle tip in the central cavity and holding with printing bed
Unlock(retaining-clips)
Manipulate-current-position-variable(y,z)
Move nozzle tip beyond and below glass sheet
Guide the glass out through the walls and throw it
```

Algorithm 4 Bridging errors simulation attack

```
Output:Poor bridging performance along x-axis
Attack resides within Move instruction code region
G-code instruction: Move from A to B
if  $B_z$ < $Layer_{width}$, Initialize layer-number
else if  $B_z$ > $A_z$ :
    Increment layer-number
    Copy $LMAP_{current}$ to $LMAP_{prev}$
else if $\Delta e > 0$ && $\Delta x$ ! = 0
    direction = $B_x$ > $A_x$ ? = +1 : -1
     xvar ← $A_x$ rounded to pixel width (20 mm)
    yvar ← $A_y$ rounded to pixel width (20 mm)
     while xvar < $B_x$
        if xvar within Attack-Zone
            i,j ← $LMAP_{ref}$ index for (xvar,yvar)
            if $LMAP_{prev_{i,j}}$ == 0
                attack-the-command=true
             $LMAP_{current_{i,j}}$ = 1
          Move xvar one pixel (20 mm) towards $B_x$
    if (attack-the-command)
             Increase extruder temperature by $5^oC$
            Reduce feedrate & fan speed by 50%
             Increase extrusion length by 25%
            Execute the move command
             Revert the changes
            attack-the-command=false
```

Algorithm 5 Outer geometry dimensions attack

```
**Output:** Bigger geometry over x & y axes
Initialize new object
G-code instruction rx: Move from A to B
if $B_z > A_z$:
    while Queue-size ≥ 3:
        Update Tail position in queue
        while Head not reached:
            Traverse the queue
            if Tail coordinates found:
                Polygon-found = True ; break
        if(Polygon-found):
            break
        Queue-size=-1
    if (Polygon-found):
        Find $P_{tail'}$ outside polygon adjacent to $P_{tail'}$
        Move to $P_{tail'}$
        for $P_i \in$ (Tail , Head)
            Find $P'_i$ corresponding to $P_i$
            Inherit $P_{i'_e}$ (extruded length) from $P_{i_e}$
            Move to $P'_i$
        Adjust $P_{Tail_e}$ for extra filament used
        Attack accomplished for current layer
        Reset Queue for the next layer
else if $B_z = A_z$:
    Add B at the Tail
    Update Head and Tail positions
else:
    Reset Queue
```

S.No	Attack action	Attack goal category	Designing (1a/1b)	Slicing software (2a)	Printing profile (2b)	Net-2 (gcode file) (2c)	Firmware (3)
1	Fingerprinting the designing software	Process surveillance	L	M	N	N	N
2	Fingerprinting the slicing and printer control sw	Process surveillance	M	L	N	M	M
3	Fingperprinting firmware behavior and responses	Process surveillance	N	M	N	M	L
4	Stealing printed object geometry information	Part Surveillance	L	L	N	L	H
5	Extracting printing profile (thermal/infill pattern /density)	Part Surveillance	N	L	L	M	M
6	Extract network devices data	Network Surveillance	M	M	N	M	M
7	Printing facility physical details (audio/visual/thermal/others)	Env. Surveillance	N	N	N	N	M
8	Print your own grave: Beak the printing glass	Physical damage for DoPS	N	H	N	H	M
9	Breaking the endstops and limit switches	Physical damage for DoPS	N	N	N	N	L
10	Deshaping the nozzle orifice	Physical damage for DoPS	N	N	N	N	M
11	Fracturing the extruder assembly	Physical damage for DoPS	N	N	N	N	M
12	Burning the nozzle heater	Physical damage for DoPS	N	N	N	N	M
13	Outer geometry modification	Garbage prints for DoPS	L	L	L	L	L
14	Object deformation through thermodynamic manipulation	Garbage prints for DoPS	N	L	L	L	L
15	Impersonating low quality bridging	Garbage prints for DoPS	N	L	L	M	M
16	Impersonating layer-shifting error	Garbage prints for DoPS	N	L	L	L	L
17	Impersonating stringing errors	Garbage prints for DoPS	N	L	L	L	L
18	Object warping	Garbage prints for DoPS	N	L	L	L	L
19	Pause and print to degrade timing and thermal profile	Interruption for DoPS	N	L	N	L	L
20	Extrusionless printing to evade nozzle kinetic detectors	Interruption for DoPS	N	L	N	L	L
21	Cold extrusion	Interruption for DoPS	N	N	N	N	L
22	Denying printer access through mac table & ARP corruption	Interruption for DoPS	N	L	N	L	L
23	Selected vertex relocation	DeTSA	L	M	N	M	M
24	Object specific feature's scaling	DeTSA	L	N	N	N	N
25	Axial misalignment of feature to cause misfitment	DeTSA	L	N	N	M	M
26	Geometric feature insertion or removal	Object Integrity	L	N	N	N	N
27	Cavity through filament-kinetics w/o modifying toolpath	SaTS	N	L	N	L	L
28	Localized density variation by filament status/speed change	SaTS	N	L	N	L	L
29	Infill pattern change	SaTS	N	L	L	H	N
30	Infill density variation (1% or more)	SaTS	N	L	L	H	N
31	Localized layer thickness attack	Object Integrity	N	L	N	L	L
32	Localized toolpath sequence modification attack	Object Integrity	N	L	N	L	L
33	Anisotropy attack	Object Integrity	L	L	N	H	N
34	Pritning instructions injection/deletion/modification	Object Integrity	N	L	N	L	L
35	Unsynchronized nozzle trajectory for x,y,e axes	DoPS	N	H	H	H	M
36	Filament erosion based density attack	Object Integrity	N	H	N	H	M
37	Swapping nozzles in multinozzle printer	Object Integrity	N	L	N	L	L
38	Modifying printing speed for localized zones	SaTS	N	M	N	M	M
39	Trapezoidal profile manipulation to cause excessive jerks	DoPS	N	N	N	N	L
40	Degrading power transfer system - belt grooves abrasion	DoPS	N	H	H	H	M
41	Nozzle partial clogging	Object Integrity	N	M	H	M	M
42	Microparticles and VOC flooding to degrade air quality	SaPP	N	M	N	N	L
43	Thermal circuit abuse to cause fire	SaPP	N	H	N	N	L
44	Printing the jobs without authorization	Unauthorized printing	H	L	N	L	H
45	Traffic manipulation to breach network integrity	NeIB	L	L	N	L	M
46	Mac table & ARP corruption to denyNetwork availability	DoNS	L	L	N	L	L

N: Not feasible H: High difficulty M: Medium difficulty L:Low difficulty * not for all objects

Table 22.: Categorization of attacks with feasibility at various stages

Part 2: Defending Fused Filament Fabrication Process

CHAPTER 5

PRINTSAFE- A NEAR REAL-TIME ANOMALY DETECTION FRAMEWORK FOR FUSED FILAMENT FABRICATION PRINTING USING PRINTING ENVIRONMENT ESTIMATION

Additive Manufacturing (AM) is a method of creating physical objects by stacking layers based on computer-aided design (CAD) information. Unlike subtractive manufacturing, this fundamentally different approach offers new attack opportunities to sabotage printed objects by introducing inconspicuous defects in internal layers that are hidden in the final print. To detect attacks on 3D printing, researchers have proposed methods to generate a master profile of every object in a protected environment and use it to examine the integrity of subsequent prints. However, this approach is unsuitable for mass customization, which is prevalent in the fourth industrial revolution. This chapter presents `PrintSafe`*, a framework that models printer firmware and printing instructions set (G-code file) to generate a real-time reference corpus of a 3D model. During the printing process,* `PrintSafe` *continuously monitors the process through physical sensors and utilizes a series of detection algorithms to identify any attempts to sabotage the printed parts. The modular and scalable framework of* `PrintSafe` *can easily meet different printing requirements. The project implements the proposed framework on a common printer in industry settings and evaluates it against eighty instances of twenty attacks.* `PrintSafe` *successfully detects the attacks and improves the existing detection horizon.* [1]

[1] *This chapter is an extension of my work presented at the ICCPS Conference 2021 [31]*

5.1 Introduction

The emerging field of 3D printing encompasses a range of manufacturing processes that create physical objects through the deposition of thin layers, using a predetermined building direction and a set of printing instructions [9]. While each process utilizes unique physical and chemical interactions with the building materials, the layer-by-layer construction exposes 3D printing to vulnerabilities that differ from those of traditional manufacturing methods, such as machining. As 3D printing gains popularity for producing critical and functional parts, securing the AM process chain becomes increasingly important. The cyber-physical nature of this process requires a different approach to security, as attacks in cyberspace can have physical consequences and vice versa.

The American Society for Testing and Materials (ASTM) 52900:2015 defines seven categories of additive manufacturing processes. Of these, 'Material Extrusion' is the most widely used, with a market share of approximately 50% [79]. Fused filament fabrication (FFF) is widely recognized as the most prevalent material extrusion technique. In FFF process, a hot nozzle extrudes molten filament in viscoelastic state, which is then precisely deposited in a 2-dimensional pattern to create a thin layer. The subsequent layers are then stacked over the previous ones to produce a complete 3D object. By introducing metal-infused filaments into the Fused Filament Fabrication (FFF) method, it is expected that FFF printing of functional parts will become increasingly common. This is because the use of metal-infused filaments enables the production of parts with improved mechanical properties, such as strength and durability, that are suitable for a wider range of applications [35]. As a result, FFF technology will likely be adopted more widely for industrial and commercial purposes, contributing to the growth of the 3D printing industry.

In the realm of 3D printing, attacks can be categorized into two broad types: sabotage attacks and intellectual property (IP) theft attacks, as noted in a survey conducted by Yampolskiy [10]. IP theft attacks may involve passive tactics, such as stealing design details from a competitor's outfit, while sabotage attacks are typically more active and involve deliberate manipulation of the printing process to cause hidden or visible damage to the printed object. Such attacks could compromise the integrity of the finished product, causing safety concerns or financial losses.

Researchers have investigated various methods to detect sabotage attacks on 3D printing processes using side-channel information emitted during the printing. While the performance of these techniques has improved over time, the current state-of-the-art is still not adequate to fully prevent attackers from causing damage to the printed object without being detected. The development of more robust and sophisticated detection techniques, coupled with effective prevention measures, will be critical to safeguarding the integrity and security of 3D printing processes. The current detection schemes, as described in the related work section, typically involve creating a master profile of every unique object by printing it in a secure environment and continuously monitoring and logging the printing state through various side channels. However, this approach is not in line with the Industry 4.0 vision, which emphasizes mass customization over bulk production.

Creating a master profile for each object can be time-consuming and impractical for production setups that require high throughput. Additionally, many existing detection methods require the printing of the entire object before conducting the analysis, which can be problematic if an attack occurs on a specific layer. This means that users would have to wait until the entire object is printed (potentially consisting of hundreds of layers) before conducting any analysis against the master profile. To address these challenges, new detection methods should be developed that can analyze the printing process in real time and detect anomalies as they occur, without requiring the creation of a master profile for each unique object. Such techniques should be scalable for mass production and provide quick feedback to users, enabling them to stop the printing process as soon as an anomaly is detected.

This chapter introduces a novel framework called `PrintSafe`, which leverages physics, printing process knowledge, and automated fingerprinting of firmware behavior to create a fine-grained anomaly detection framework. Rather than profiling each unique object, `PrintSafe` models the printing environment and creates a synthesized ground truth for every unique object based on its input G-code file. `PrintSafe` employs out-of-band sensors to measure the printing state in the physical domain and utilize a series of algorithms to identify any anomalous deviations in the printing process. The fundamental assumption behind `PrintSafe` is that any cyberattack to sabotage the printed object would involve manipulating the primary physical processes, such as the kinetics and thermodynamics processes in the case of FFF. By authentically monitoring the state of these processes and comparing them with ground truth, we can verify the printing integrity or detect any anomalies. Another key feature of `PrintSafe` is its ability to remain independent of the cyber path of the printing process chain, ensuring that if an attacker successfully breaches the printing stage, they cannot evade `PrintSafe` by providing false information.

The study implements `PrintSafe` on a common FFF printer in industry settings - Ultimaker 3 and evaluates its performance on eighty different attack instances of twenty sabotage attacks. `PrintSafe` claims superior detection thresholds than the existing state of the art, and the evaluation confirms successful detection of all attack instances above the claimed detection thresholds with zero false negatives and false positives.

5.1.1 Motivation

In a hypothetical scenario, politically motivated attackers could exploit a zero-day vulnerability in an application running on the 3D printer controller to gain root access. The attackers could then modify the printing process to introduce defects in the final printed part and delete logs to cover their tracks. Such an attack could lead to compromised functionality and create potential risks for safety-critical applications, such as medical implants or aerospace components. Although this attack could evade conventional cybersecurity solutions, an anomaly detection system based on physical process data could effectively detect such cyberattacks in 3D printing facilities. The detection system is independent of the printing process chain and attains printing state information only through the physical domain. It is fair to assume that an attacker will not compromise two independent systems simultaneously. Therefore, leveraging the physical domain knowledge to secure a cyber-physical system significantly enhances the security against sabotage attacks. Failure to use such a system would be analogous to guarding a fence with one eye closed.

5.1.2 Research Challenges

A G-code file contains explicit instructions for the printer to print a specific object. While a legacy printer would follow these instructions exactly, a newer printer may attempt to "optimize" the printing process in various ways, such as accommodating tilt in the printing bed by auto-generating Z-axis movements. This means that it is difficult to distinguish between legitimate printer-induced deviations and malicious attacker-generated deviations, as the G-code file does not contain information about these potential optimizations.

Another hurdle in effectively detecting anomalies in 3D printing is that it is a cyber-physical process that has certain non-zero printing specification tolerance values. This implies that even with flawless monitoring of the system state, which is practically unattainable, labeling the slightest deviation as anomalous would lead to a notable number of false positives. Additionally, the significance of the process deviation magnitude varies greatly with the attack context. As a result, any detection approach must take into account the random physical process deviations, measurement inaccuracies, modeling errors, and the attack context to issue a reliable verdict regarding the presence of an anomaly.

5.1.3 Contributions

The overall contributions of this work are as follows:

- Introduces `PrintSafe`, an innovative anomaly detection solution for FFF-based 3D printing processes that models the printing environment using physics, AM process knowledge, and firmware functions fingerprinting.
- Documents a comprehensive implementation study of a common printer that can be easily replicated by interested readers, providing a clear roadmap for implementing `PrintSafe` in different 3D printing environments.
- Presents a thorough evaluation of `PrintSafe`'s performance against existing AM attacks documented in the literature, demonstrating its effectiveness in detecting and mitigating a wide range of AM attacks, making it a valuable addition to the 3D printing security toolkit.

5.2 Related Work

The focus of this study is detecting sabotage attacks in 3D printing security, which has been extensively explored by researchers. Their primary efforts have been directed toward addressing print integrity attacks on the FFF process. Researchers have proposed several techniques to acquire the printer state through a variety of independently deployed sensors, rather than relying on the system's feedback.

Chhetri *et al.* (2016) were the first to utilize audio sensors for 3D printer security by detecting the acoustic signals generated by stepper motors, resulting in an overall detection accuracy of 77% for path and speed modifications. However, for small deviations of 3 mm or less, the true positive rate was 71% and the false positive rate exceeded 30%.

Belikovetsky et al. (2019) used a different algorithm in combination with an acoustic sensor on a mobile phone, achieving higher accuracy and resolution. The method was able to detect modifications sustained for at least 1 second, while a minimum detectable threshold of 2.6 seconds was required for command reordering.

Gao *et al.* (2018) utilized Inertial Measurement Unit (IMU) sensors and cameras, applying sensor fusion techniques and random forest to detect attacks on kinetic properties. The selected test cases were significantly different from the original print, including around 20% changes in infill, a reduction of fan speed from 100% to 25%, and an increase in printing speed from 30 mm/s to 120 mm/s. However, the accuracy of the solution under small deviations was not ascertained.

Wu *et al.* (2017) used static and moving cameras to detect infill pattern attacks. The injected attack patterns were large enough to cover around 10-20% of the infill area. Similarly, Bayens *et al.*(2017) used a microphone, IMU sensors, and a camera to verify infill patterns, and doped filament to verify material integrity via CT scan. The infill patterns selected for testing the technique were Honeycomb and Rectilinear with 20%, 40%, and 60% density, which presented a substantial difference to detect.

Gatlin *et al.* (2019) measured the electric current drawn by the stepper motors during printing to create a power signature profile of an object, which required multiple prints to create a master profile. However, this method could not detect individual layer thickness variations or filament extrusion attacks.

Yu *et al.* (2020) extended Chhetri *et al.*'s (2016) work by studying acoustic, magnetic, and visual information, resulting in the successful detection of a 4 mm deviation. However, the authors acknowledged the limitations of identifying smaller changes.

Furthermore, existing techniques are limited in their ability to detect filament extrusion rate variations. If an attacker stops or alters the speed of the filament extrusion motor, the resulting object specifications will be changed. While camera-based inspection techniques proposed by Wu *et al.* [41] may work in some cases, such as when the attacker completely halts the extrusion motor, detecting changes in filament density by reducing the extrusion motor's speed seems unlikely using visual cameras. In addition, the printhead on top of the object can obstruct the camera's view, and it is not always feasible to pause the printing process to capture a clean image after each layer.

The minimum detectable change in existing techniques is not small enough to effectively prevent feasible attacks. An attacker can damage the object by making 1 mm changes in dimensions or by injecting or removing commands of 1 second, or simply by reducing the extrusion speed. Previous studies have already demonstrated the impact of such attacks and their practicality [12, 22, 8, 33, 39].

Another significant limitation of most existing work is the learning phase, which requires one or more training prints of each unique object in a protected environment to create its master profile. This requirement limits the use cases to repetitive production setups and is not aligned with the Industry 4.0 vision of mass customization. Therefore, there is a need for new approaches that can address these limitations and provide more robust security for 3D printing systems.

5.3 PrintSafe - An FFF attack detection framework

The concept of `PrintSafe` is based on the collaboration of cyber and physical domain information to enhance the security of cyber-physical systems. The final output of a CPS is typically in the physical domain. In Additive Manufacturing, the output is the printed part. Even if an attack goes unnoticed in the cyber domain, it can be detected by using cyber-physical security solutions, such as `PrintSafe`. While one hashing algorithm suffices to verify the integrity of cyber-domain artifacts belonging to various industries, cyber-physical security frameworks must be more process-oriented. This section details a cyber-physical interactions-based anomaly detection framework `PrintSafe` presented in Figure 46.

5.3.1 Data Acquisition and Accumulation

FFF (Fused Filament Fabrication) printing is a highly complex process that involves multiple sub-processes, all of which can impact the final outcome. However, our focus is on identifying the sub-processes that can be direct targets of cyberattacks. Equation 5.1 formalizes the search for a minimal subset of independent processes, denoted as P_x, which are involved in printing and can be directly influenced by at least one command in set C through a function f that maps each command to the set of processes it can influence.

$$\begin{aligned} P_x = \{P_x \subseteq P \mid \exists\, C_i \in C : P \subseteq f(C_i) \land \\ \forall p_1, p_2 \in P : p_1 \neq p_2 \implies p_1 \perp p_2\} \end{aligned} \tag{5.1}$$

In FFF printing, there are two primary processes that can be influenced through cyber-domain manipulations: kinetics and thermodynamics. After analyzing the FFF printing process, we have identified four key components that are part of the printer's kinetics, including two stepper motors for the printhead, one motor for the printing

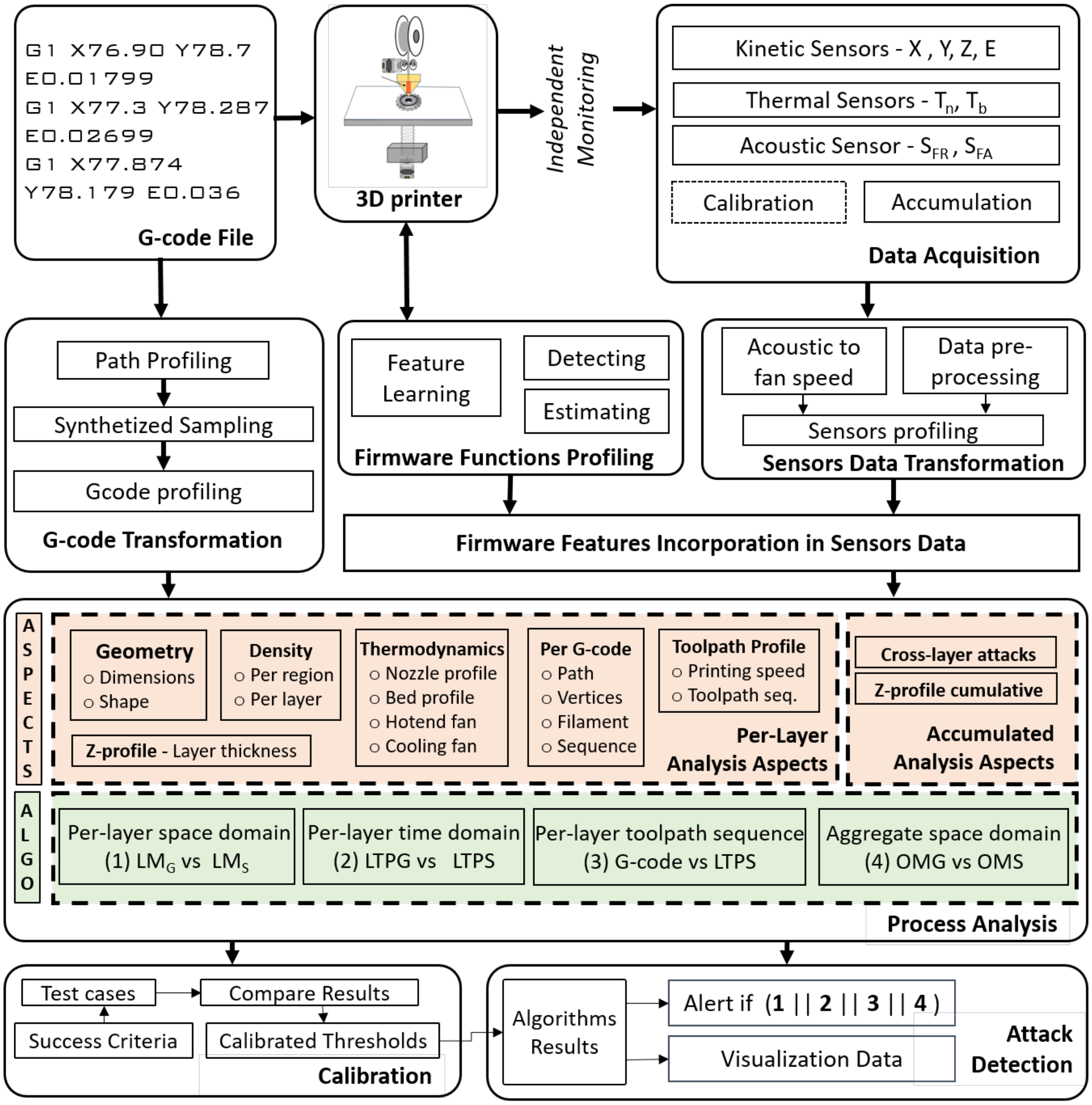


Fig. 46: PrintSafe - A cyber-physical anomaly detection framework for FFF-based 3D printing

bed, and one for the filament extrusion. These components control the movement and deposition of the printing material, making them critical to the printing process. Similarly, we have identified four critical components that impact the printer's thermodynamic profile, including the nozzle heater, printing bed heater, hotend fan, and part cooling fans. These components regulate the temperature of the printing material, ensuring proper adhesion and cooling.

To ensure the authenticity and integrity of the data collected during the printing process, it is recommended to use an independent set of sensors for data acquisition. The choice of sensors may vary depending on the requirements and installation provisions, but they should comply with the non-intrusiveness, accuracy, noise resilience, ease of deployment, and budget requirements of the solution. Encoders are recommended for kinetic, thermocouples for thermodynamic, and acoustic sensors for fan speed measurements based on the literature. `PrintSafe` uses these sensors to model a specific component or functionality of the printer. Once the model is available, `PrintSafe` can detect anomalies in near real-time without learning the object's master profile.

Acoustic sensor for fan speed measurement Six out of the seven sensors are directly measuring the individual component's desired property. Optical encoders independently measure the movement of each axis, and thermocouples measure the temperature of the probe location. However, direct measurement of fan speed is not possible in compliance with the set criteria of non-intrusiveness and deployment feasibility. Fortunately, the fans emit specific frequency patterns and harmonics. The spectrum shifts slightly with the fan speed and the frequency zone is insignificantly impacted by other acoustic noises such as the motors and motion system sounds. The spectrogram presented in Figure 47 displays an audio file that records the sound of

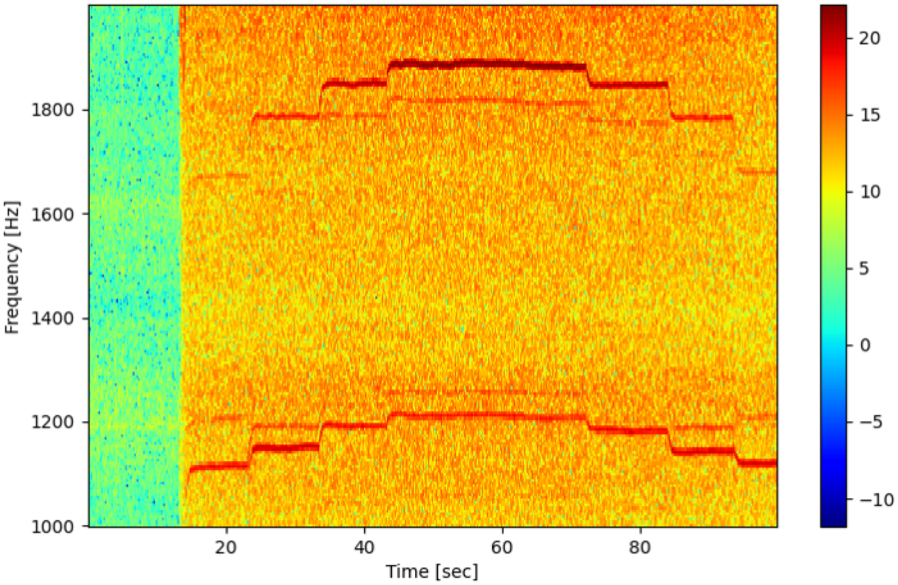


Fig. 47: Spectrogram of the acoustic profile of the cooling fans

cooling fans being adjusted to different levels, starting from the lowest bracket, going up to the highest, and then back down to the lowest.

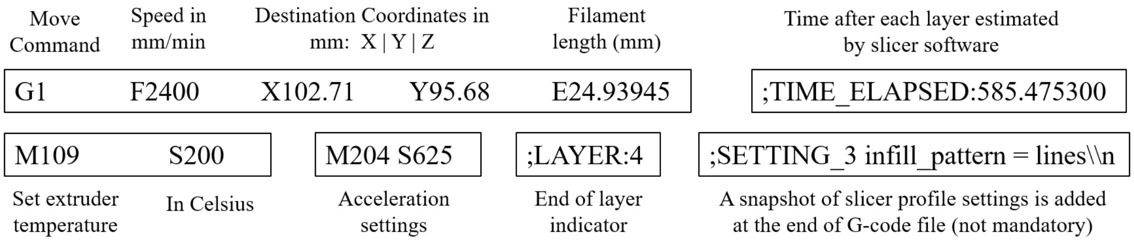
Sensors data accumulation The data from various sensors is accumulated using a microcontroller board, such as Arduino, or a single-board computer, such as Raspberry Pi. The acoustic sensors are commonly available with USB interface; hence, it is feasible to bypass the intermediate data accumulation stage. For other sensors, interrupt routines are used for capturing their data. External interrupts are used for tracking the fast-moving printhead, while internal interrupts are used for the slow-varying kinetic and thermodynamic data.

Sampling rate selection: Choosing an appropriate sampling rate is crucial for accurate and efficient data transfer from the microcontroller board to the data repository. Ubiquitous acoustic sensors mostly operate up to 20 to 20 KHz frequency range.

However, the acoustic sampling rate should be selected in line with the fans frequency spectrum including the significant harmonics. PrintSafe distinguishes between the printhead's printing moves (involving filament extrusion) and alignment moves (without filament extrusion) based on data sample values. As the filament moves slower than the printhead, it is possible that consecutive samples may show the same reading for the filament state, even when the filament is moving. For a fine-grained detection framework, this situation can lead to false positives. To address this, we use two separate data structures with distinct sampling rates for the fast and slow varying data. Capturing slow-moving filament data through higher-resolution sensors also helps in mitigating the incorrect state deduction problem.

5.3.2 Transformation of the object design

When it comes to selecting a ground truth for printing objects, there are a few options available, including the 3D model or CAD file, the STL file, and the G-code file. However, the G-code file is the most comprehensive option, as it includes the necessary details from the CAD and STL files, as well as critical parameters for printing such as the number of slices, temperature and speed settings, and infill pattern and density. That's why, for PrintSafe, we utilize the G-code file as the ground truth for printing objects.



```
G1 F2400 X102.71 Y95.68 E24.93945
;TIME_ELAPSED:585.475300
M109 S200
M204 S625
;LAYER:4
;SETTING_3 infill_pattern = lines\\n
```

Move Command
Speed in mm/min
Destination Coordinates in mm: X | Y | Z
Filament length (mm)
Time after each layer estimated by slicer software
Set extruder temperature
In Celsius
Acceleration settings
End of layer indicator
A snapshot of slicer profile settings is added at the end of G-code file (not mandatory)

Fig. 48: Samples of G-code commands

G-code: In Figure 48, several G-code specimens are depicted. The instructions beginning with the letter 'G' are primarily utilized to control kinetics. For instance, instructions such as G0' or 'G1' are move instructions that may have up to five parameters, including X/Y/Z coordinates, the printing speed, and the length of the filament to be extruded until the move is complete. If an argument is not provided, the previous value is used, implying that there will be no motion along that axis. The second category of instructions that commences with the letter M' governs thermodynamics and other printing parameters, such as positioning mode, acceleration, and jerk settings. G-code lines beginning with a semicolon are merely comments. Typically, the slicing profile is appended to the bottom of the G-code file as comments by the slicer software. To synthesize space and time-domain samples, `PrintSafe` parses the G-code file and converts each layer instruction set into a sequence of samples.

The multi-feature bitmap representation from the G-code file can be achieved by following the sequence of tasks outlined in Algorithm 6. For an object with 'n' layers, Equation 5.2 defines Object-map(OM) as an 'n' elements array of layer-thickness profile (ZP) and two-dimensional Layer-map (LM) that represents each layer through a matrix of pixels with print time (t), instantaneous bed height (z), nozzle temperature (T_N), bed temperature (T_B), filament length (e), and the extrusion state (c) attributes.

$$OM = \{LM_k, ZP_k \ \mid k = 1, 2, ..., n\} \ where \ LM_{k_{i,j}} = (t, z, T_N, T_B, e, c) \tag{5.2}$$

The print time for each pixel in a move instruction is computed using motion equations that satisfy the constraints of maximum speed and acceleration given in the G-code file. The value of extruded filament is distributed linearly throughout the move, which is the intended and standard behavior of the printer. The nozzle-temperature attribute of a pixel indicates the temperature of the filament during extrusion at that location. Since thermal variations change much slower than kinetic attributes, we use a rough estimate of the PID circuit used in the printer to approximate the temperature values. Examining the object's time-domain profile (OTP) highlights certain aspects that are not conspicuous in OM. For instance, the printing speed can be measured with higher precision with the timing profile data. Equation 5.3 defines OTP as an array of pairs of each layer's timing-profile LTP and the layer height for that layer. From the sensors data perspective, OTP is the primary representation that is transformed into OM through the same algorithmic logic as mentioned in Algorithm 6. f_h and f_p in Equation 5.3 represents the hotend fan state (running or stopped) and the part cooling fan speed.

$$OTP = \{LTP_k, ZP_k \ \mid k = 1, 2, ..., n\} \ where \ LTP_{k_t} = (x, y, z, T_N, T_B, e, c, f_h, f_p) \quad (5.3)$$

5.3.3 Estimating the firmware induced functions

The G-code move commands serve as explicit instructions that the firmware adheres to in both form and function. As printers advance and automation becomes more prevalent, they become increasingly intelligent and capable of offloading certain user tasks. For example, printing bed leveling, once a manual task, can now be accomplished automatically using a proximity sensor and a sequence of instructions to detect the tilt in the printing bed and adjust it continuously during the printing operation. While this allows the printer to produce nice prints on an unlevel bed, it also implies that a single G-code file printed at different times on the same printer may result in a different sensor data set. A security solution considering G-code as the ground truth may raise false alerts over the legitimate compensation for tilt by the firmware. To overcome this challenge, `PrintSafe` proposes a semi-automated

Algorithm 6 Transforming G-code to Space Domain Data

```
Require: G-code file
Ensure: ObjectMap (OM)
Initialize OM
Initialize LM    # For each layer
while not end of G-code file do
    I = pickNextInstruction()
    if I $\in$ layerEndMarker then
        optimize(LM$_i$)
        OM $\leftarrow OM \cup$ LM   # Append the layerMap to ObjectMap
        Initialize LM    # Initialize new layer
    else if I $\in$ PrintingParameters then
          Update PM # current parameter set
      else if I $\in$ MoveCommands then
          Calculate MA and S$_{AB}$ ,   # MA is the major axis, S is the slope
          for $m \in$ MA do
              Find $n \leftarrow \lfloor P_{res} \cdot (|mA| \cdot S_{AB}) \rfloor$  # n $\in$ minor axis of AB; P$_{res}$ is pixel resolution
              Assign following attributes to pixel P in LM marked by m & n
              Calculate PixelTime$_P$ :
              Let $\overline{D_1 D_2}$ b/w A and B is constant speed zone
              if $|AP| \leq |AD_1$— then
                  PixelTime$_P$ = $\sqrt{2 * |AP| / ACC_m}$
              else if $|AD_1| > |AP| \geq |AD_2|$ then
                  PixelTime$_P$ = $|D_1 P| / S_m + \sqrt{2 * |AD_1| / Acc_m}$
              else
                  PixelTime$_P$ = $2 * \sqrt{2 * |AD_1| / ACC_{max}} + (|AB| - 2 * |AD_1|) / S_m - \sqrt{(2 * |BP|) / Acc_m}$
              end if
              N.T$_P$ $\leftarrow$approxPID($N.T_{cur}, N.T_{dest}, PrintTime_A$)    # N.T is the Nozzle temperature
              B.T$_P$ $\leftarrow$approxPID($B.T_{cur}, B.T_{dest} - B.T_{prev}$)   # B.T is the bed temperature
              F.L$_P$ $\leftarrow e_A + (m/$—MA—$) \cdot \Delta e$    # F.L is the filament length
              if $\Delta e > 0$ then
                  $F.S_P = 1$   # Indicates material presence at the pixel. Default is zero
              end if
              LM$_{x2,y2}$ $\leftarrow$ $(PixelTime_P, Z_{cur}, F.L_P, N.T_P, B.T_P)$, x,y=m,n or n,m as per major Axis
              coverFilamentThickness(LM$_{x2,y2}$, majorAxis, f$_w$ ) # f$_w$ is filament thickness
          end for
          $pointA, e_A, B.T_{prev} \leftarrow pointB, e_B, B.T_{cur}$
      end if
  end while
  return OM
```

function profiling module that can incorporate new features into the overall security framework without compromising automation. The function profiling module consists of three phases. In the learning phase, a user creates a fingerprint of the function by generating test cases, measuring the sensor data for each case, and extracting unique features to arrive at a robust fingerprint. Algorithm 7 presents the method to extract the fingerprints of firmware-induced functions. Once the printing starts, the detection engine continuously searches for the functions signatures in the incoming sensors data. If a signature is matched, `PrintSafe` calculates the impact of the function's call and incorporates it in the future sensors data before any further analysis. In this way, the core analysis modules can be reused to compare the modified sensors' data set against the original G-code synthesized data set.

Algorithm 7 Generating fingerprint for firmware induced functions

```
Require: Firmware function semantic knowledge
Ensure: Pattern Fingerprint F = $(p_1, t_1, sn_1), (p_2, t_2, sn_2), ..., (p_m, t_m, sn_m)$
Generate test cases incorporating functional knowledge
Run test cases and attain data samples set S = $S_1, S_2, ..., S_n$
for window $w$ in range $[W_{min}, W_{max}]$ do
    for time $t$ from $t_{(max-w)} - t_o$ do
        Identify a pattern $p_i$ (value / variation / event / correlation)
        if pattern $p_i$ not in $F$ and $p_i$ exists in all samples in $S$ then
            Add $p_i$ to $F$ (add $p_i$, t, seqNo);
            seqNo++
        end if
    end for
end for
return Fingerprint F
```

Axes homing function The homing command'G28' does not specify the magnitude of movement, as other G-code move instructions do. Instead, it relies on the limit switches of the printer's axes to stop the movement. When an axis reaches its limit switch, it goes through a slower final homing sequence to avoid overshoots.

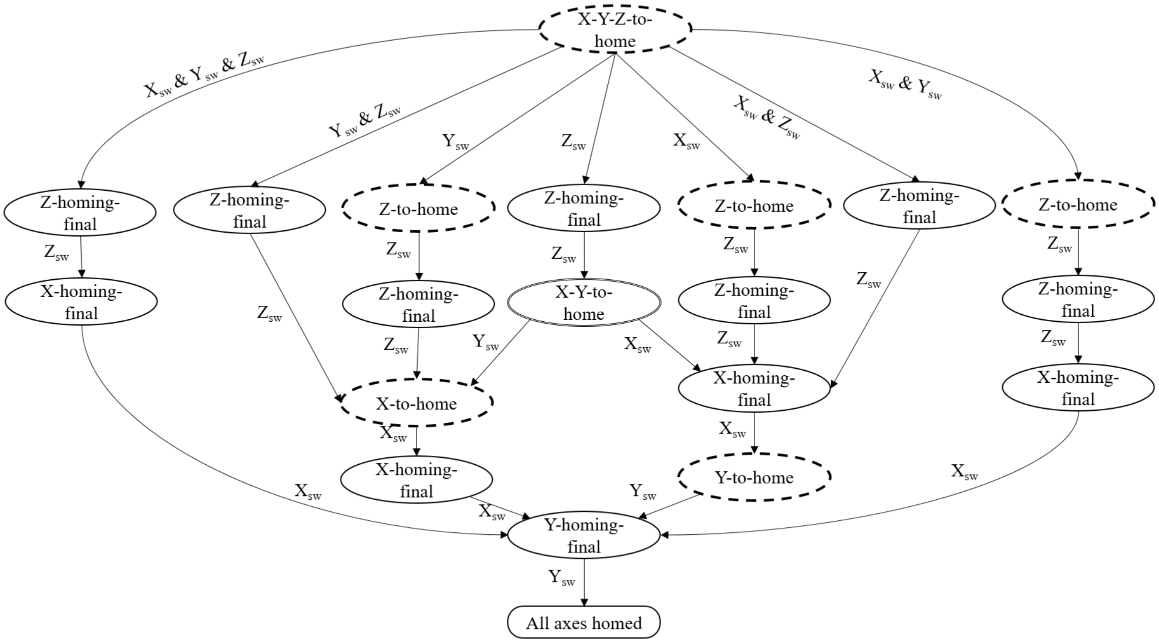


Fig. 49: State transition diagram for the printing axes homing feature

Homing is essential to synchronize the firmware and G-code coordinates system, and the printer firmware uses this calibration at various instances, such as starting a new printing job or performing automatic bed leveling. When a homing command is initiated, a printer could be in one of the eight different states starting from all axes already homed to none of them is homed. Figure 49 presents the homing command's state diagram, where the engagement of the limit switches is considered an event, and the moves are considered states. Our homing function learning module extracts the fixed pattern for the final homing of each individual axis, the relative temporal positioning, and the maximum and minimum time window for the activity completion to create the homing function fingerprint. Section 5.4 presents a practical example showing the sensors' data in response to the generated test cases.

5.3.4 Process Analysis

As previously mentioned, the PrintSafe framework involves iteratively examining the printing process after each layer is completed. The process analysis can be divided into two major categories: Assessment of the latest layer, and the accumulated assessment up to the latest layer. A layer is a fundamental building unit in additive manufacturing. Two layers are connected through a simple and short-lived layer-transition event, which is the downward movement of the printing bed to create space for the next layer. The layer-by-layer analysis effectively covers most of the attackable aspects of the printing. Considering Z-axis as the direction of the printing bed (a typical case in FFF printers), a single-layer analysis can reveal defects within the XY plane. However, if the attacker implants the defects in the XZ, YZ, or in any other plane, such as XZ or YZ or any curved surface, and also makes sure that its reflection on a single layer stays within the detection horizon, then such attacks can evade the layer-monitoring solution. Therefore, PrintSafe also analyzes important aspects of the accumulated process after every layer.

5.3.4.1 Per-layer analysis

Once i^{th} layer is printed, PrintSafe extracts the raw sensors data between the i^{th} and $(i-1)^{th}$ layer-change markers and apply transformation algorithms to attain the LM_i, LTP_i, ZP_i, and update the OM, OTP, and ZP data structure. These data structures along with the ground truth are fed to the process analysis stage. The aspects of the printing process and the printed object covered in the per-layer analysis include the following aspects.

Geometry: The layer geometry is examined to verify the outer dimensions, identify malicious voids, detect changes in infill pattern, and verify infill density. It is worth noting that PrintSafe considers each infill line as a separate geometric feature and verifies its integrity. This detailed verification process eliminates many potential attacks from the list of possible threats.

Filament density: The object density or filament density variations may occur if an attacker intentionally disrupts the ratio between the distance covered by the printhead and the length of the extruded filament. PrintSafe conducts a thorough analysis of the filament consumption of the entire layer and its sub-regions with lower or higher densities. 'Smart voids' are an extreme case of low-density zones.

Thermodynamic profile: The PrintSafe framework monitors four aspects of the thermodynamic profile of the printed object. The nozzle's thermodynamic profile in the space domain constitutes the temperature of the nozzle when the pixel on the layer-map LM was printed. LTP covers the temporal profile of the nozzle temperature and the printing bed temperature. LTP also constitutes the state of the hotend fan and the speed of the part cooling fans.

Parameters per G-code command: Another aspect PrintSafe validates for each layer relates to the G-code commands. It verifies if the extruder path measured by the sensors is aligned with the instructions within the allowed tolerances. It also verifies the vertices or the endpoints of the instructions and the filament consumed during each instruction.

Toolpath profile: The toolpath profile specifically covers the toolpath sequence of the actually printed layer and the printing speed profile.

Z-profile: The z-axis movement is used as the layer-change marker to split the data set into layers. In addition, the z-movement preceding the layer's printing dictates the thickness of the layer. This z-axis movement should correlate with the filament density of the layer. An attacker can implement layer-bonding reduction attacks by manipulating the z-profile. Hence, `PrintSafe` considers z-axis movement as an important aspect of the per-layer analysis.

5.3.4.2 Accumulated analysis aspects

After printing the i^{th} layer, LM_i, and LTP_i are appended to the object-map OM and OTP, respectively. The transition from i^{th} to $(i+1)^{th}$ layer is a distinct event in the space domain. Most of the adjacent layers have a significant overlap in the layer-map layout; hence, the space-domain aspects can highlight attackers' attempts to amplify a non-detectable attack on i^{th} layer by extending it at the same location in the adjacent layers. In the accumulated aspects, `PrintSafe` considers space-domain defects across layers and the z-profile accumulated error effects that may go unnoticed at the individual layers.

5.3.4.3 Process verification algorithms

The core of `PrintSafe` process analysis module are the four algorithms that verify the per-layer and accumulated aspects of the printing described in the preceding subsection.

Per-layer space domain comparison algorithm The first algorithm pertains to space domain integrity testing. The algorithm compares the most recently obtained layer-map LM_S, which is generated from sensors data, with the corresponding layer-map, LM_G, synthesized from the G-code file. The first step of Algorithm 8 is to synchronize the layer-maps by shifting LMS by ±$syncth$ pixels in both dimensions to minimize the raw error in the pixel's binary attribute, 'color', which represents the presence or absence of material the pixel location. The synchronized layer-maps are then compared using the 'noMatch' function, which checks if the attributes of corresponding pixels in LMS and LMG are within the allowed thresholds. The tolerance threshold values vary depending on the attribute type. For instance, there is no tolerance for the 'color' attribute. The highest tolerance is assigned to the 'printing time' attribute to compensate for the estimation and random errors in the process. The difference array ($diffArray$) passes through an optimization function that compensates for the instantaneous spatial drifts in the printhead location. The optimization step reduces the number of false positives but also sets a minimum detectable spatial deviation. Finally, a list of contiguous mismatching areas is computed and sent to the attack detection phase.

Per-layer time-domain comparison algorithm The second comparison algorithm uses the reference timing profile for a layer, LTP_G to find corresponding samples in LTP_S from the printing sensors. Unlike space domain layer-maps, a timing profile is a one-dimensional sequence of samples. This algorithm compares two time-domain samples LTP_{G_i} and LTP_{S_i} to identify if all the attributes are within the acceptable thresholds. The comparison sequence is presented in Algorithm 9. The first step is to synchronize the two sequences by taking a small window of samples starting from the first sample in LTP_{G_i} and identifying corresponding samples in LTP_{S_i}. Once synchronization is achieved, it is practically difficult to identify fine-grained pixel-level manipulations through time-domain analysis due to less accurate estimation, this algorithm outperforms the space-domain analysis in detecting the printing speed attacks and toolpath re-sequence attacks. Moreover, the slow-varying fan speed attacks

Algorithm 8 Space domain comparison algorithm: Non-conforming areas list

```
Require: $LM_G$ and $LM_S$
Ensure: Non-conforming areas list
# synchronizing the layermaps: ($LM_G$,$LM_S$)
$syncErr \leftarrow \infty$ ; $di, dj \leftarrow 0$
for $i$ ranging from $-LM_{th-sync}$  $to +$ $LM_{th-sync}$ do
    for $j$ ranging from $-LM_{th-sync}$  $to$ $+LM_{th-sync}$ do
        if $syncErr$ ¿ cumuRawErr($LM_G$,$LM_{S_{dft\ by\ i,j}}$) then
            $syncErr \leftarrow$ cumuRawErr($LM_G$,$LM_{S_{dft\ by\ di,dj}}$)
            $di \leftarrow i; dj \leftarrow j$
        end if
    end for
end for
$LM_S \leftarrow LM_{S_{dft\ by\ di,dj}}$
initialize $diffArray$
for $i$ in rowsIn($LM_G$) do
    for $j$ in colsIn($LM_S$) do
        for attr in attrList do
            if noMatch($LM_{G_{i,j}}$, $LM_{S_{i,j}}$,att$_{th}$) then
                diffArray$_{i,j}$.attr = 1
            end if
        end for
    end for
end for
diffArry ← optimizeLM(diffArray)
ncAreaList ← getContiguousAreas(diffArray)
ncAreaList.insert(0, rowsIn($LM_G$) * colsIn($LM_G$))
return $ncAreaList$ # Non-conforming areas list
```

and layer-thickness attacks are also examined efficiently by comparing LTPs.

Algorithm 9 Time domain comparison algorithm: Mismatched samples computation

```
Require: $LTP_G$ and $LTP_S$
Ensure: Mismatched LTP samples list
# synchronizing the time profiles: ($LTP_G$,$LTP_S$)
for $i$ ranging from 0 to $G_{th}$ do
    for $j$ ranging from 0 to $S_{th}$ do
        if $LTP_{G_i} \cong LTP_{S_j}$ then
            # Confirmation window
            if $ginit \leftarrow$ confirmSync($LTP_G$,$LTP_S$,i,j,winSize)==True then
                syncAchieved=True
                break
            end if
        end if
    end for
end for
if syncAchieved==False then
    return Synchronization failed
end if
for i ranging from $ginit$ to (len ($LTP_G$) − $th_{windw}$) do
    for att in attrLIst do
        if noMatchInWindow ($LTP_G$,$LTP_S$,$i$,$th_w$) then
            ncSamplesList.append($LTP_{G_i}$)
        end if
    end for
end for
return ncSamplesList
```

Per-layer toolpath sequence verification algorithm The G-code move instructions are directly responsible to create the object. Unlike many thermodynamic or parameter setting instructions, the move instructions are executed in a strict sequence. Let's consider a move from $A_{x1,y1}$ to $B_{x2,y2}$ and let LTP_{S_i} be the sample in the sensors data set that corresponds to $A_{x1,y1}$. In a perfect scenario, every sample ahead of LTP_{S_i} should monotonically get closer to B. Before finding a sample LTP_{S_f} corresponding to B, if we do not find any sample LTP_{S_k} such that the distance between LTP_{S_k} and B is more than the distance between $LTP_{S_{k-1}}$ and B, then the samples sequence has correctly followed the toolpath.

As outlined in Algorithm 10, the first step is to extract the move instructions from the G-code file corresponding to the recent layer. After synchronizing the sample series with the first G-code move command in the *movSet*, the algorithm iteratively validates each qualified move command with the subsequent samples. In addition to getting monotonically closer to vertex B, a sample must also adhere to the maximum allowed distance, $th_{pt2line}$, between the sample and line AB. Vertex B is found if there exists and LTP_{S_i} such that $\|LTP_{S_I} - B\| < th_{vertex}$. Once vertex B is found, refineVertex function is run to select the best candidate sample for B. Once vertex B is refined, the algorithm compares the filament consumption mentioned in the G-code move command with the one attained by measuring the difference between the filament length for the sample $LTP_{index.B}$ and $LTP_{index.A}$.

The vertex vicinity threshold, point-to-line distance threshold, and filament consumption threshold parameters can be used to adjust the algorithm's detection resolution and false positive rate. One of the key benefits of toolpath sequence verification, as opposed to Algorithm 9, is the ability to take advantage of frequent synchronization opportunities presented by G-code command vertices. Since adjacent G-code move commands follow different slopes, this provides an additional useful feature for refining the task of finding vertices. As we are not considering the timestamp, we do not detect printing speed attacks through this algorithm.

Aggregate Space-Domain Comparison Algorithm The three algorithms discussed so far carry out the integrity analysis of a single printed layer, which is typically the most recent one. While this approach is effective against the majority of attacks in the literature, it is not sufficient to detect cross-layer attacks. Even a tiny void in a single layer may go unnoticed, but if it is repeated across multiple layers at the same

Algorithm 10 Per-layer G-code commands verification

```
Ensure: G-code validation status
Require: GcodeFile, $LTP_i$   #   $i^{th}$ layer timing profile
movSet ←extractMoves(GcodeFile,i)
for cmd in $movSet$ do
    if $d_{cmd}$ ¿ $d_{th}$ then     # mov distance big enough to be verified
        if  syncSampleSeries(cmd, SeriesSyncTh) ==True then
            break
        end if
    end if
end for
if syncSampleSeries == False then
    return ”InitialSyncFailure”
end if
for  $C$ in $movSet$ do    # Each C represents move from A to B
    # Finding sample $s_i$ in $LTP_i$ : $s_i \overset{\sim}{=}$ A
    if  findVertex($C_A$ , LTP$_i[cur:]$) ≠ Null then
        ind$_A$ ← findVertexIndex(A , LTP$_i[cur:]$)
        ind$_A$ ← refineVertexIndex(A, LTP$_i$[ind$_A$ : ])
    else
        return ” Cmd $C$ in Layer$_i$ not verified”
    end if
    ind$_B$ ← ind$_A$
    while $C_B \ncong$ LTP$_i$[ind$_B$] ) && ind$_B$ ¡ len(LTP)) do
        ind$_B$+=1
        if getCloser(LTP[ind$_B$], C$_B$) then   # Samples monotonically closing to C$_B$
            if dist(LTP[ind$_B$] , $\overline{AB}$) ≤ $th_{pt2line}$ then  # Sample in proximity to line $\overline{AB}$
                if findVertex(B)==True then
                    ind$_B$ ← refineVertexIndex(B)
                else
                    return Cmd k findVertex(B) failed
                end if
                if (LTP$_i$[ind$_B$].e − LTP$_i$[ind$_A$].e) − ($C_B$.e - $C_A$.e ) ≤ th$_{flmt}$ then
                    Command k Verified
                else
                    return Cmd k filament consumption test failed
                end if
            end if
            return Cmd k samples proximity to $\overline{AB}$ failed
        end if
        return Cmd k gettingCloser(B) failed
    end while
end for
return Cmd verification passed
```

location, it can have severe consequences. Figure 50 illustrates this issue, showing examples of cavities that create an amplified effect across multiple layers.

To address this vulnerability, `PrintSafe` conducts an aggregate analysis of the space domain up to the latest layer in three dimensions. One approach to handling this vulnerability is to reuse the per-layer space-domain comparison algorithm designed for the XY plane, in the XZ and YZ planes. However, this approach is computationally expensive since a rectangular prism of 50 mm x 50 mm x 50 mm will have unique 500 XZ and YZ planes each, resulting in 1000 independent 2D analysis runs. Furthermore, this approach does not account for the curved surfaces and planes involving all three axes.

To overcome these challenges, we modify the non-compliant contiguous areas finding problem to a non-compliant contiguous volume finding problem. Regardless of the shape and axes of the cavity, the algorithm searches for contiguous pixels that do not comply with the criteria. We use the single-layer space-domain comparison algorithm for each layer to arrive at a 'diff3DArray' comprising an array of 'diff2DArray' corresponding to each layer. after optimizing 'diff2DArray', it is appended to the 'diff3DArray'. A contiguous volume finding algorithm then identifies all the zones that have non-compliant contiguous pixels. To calculate the volume of each contiguous zone, Equation 5.4 is used:

$$zoneVol_i = \sum_{l=1}^{n} ht_l \times P_{res_X} \times P_{res_Y} \times PZ_{i_l} \tag{5.4}$$

In this equation, $zoneVol_i$ represents the volume of the i^{th} zone in cubic millimeters (mm^3). The height of the l^{th} layer in millimeters is denoted as ht_l. P_{res_X} and P_{res_Y} represent the pixel dimensions in the x and y-axis in millimeters, respectively. The number of contiguous pixels in zone i that are part of the l^{th} layer is represented

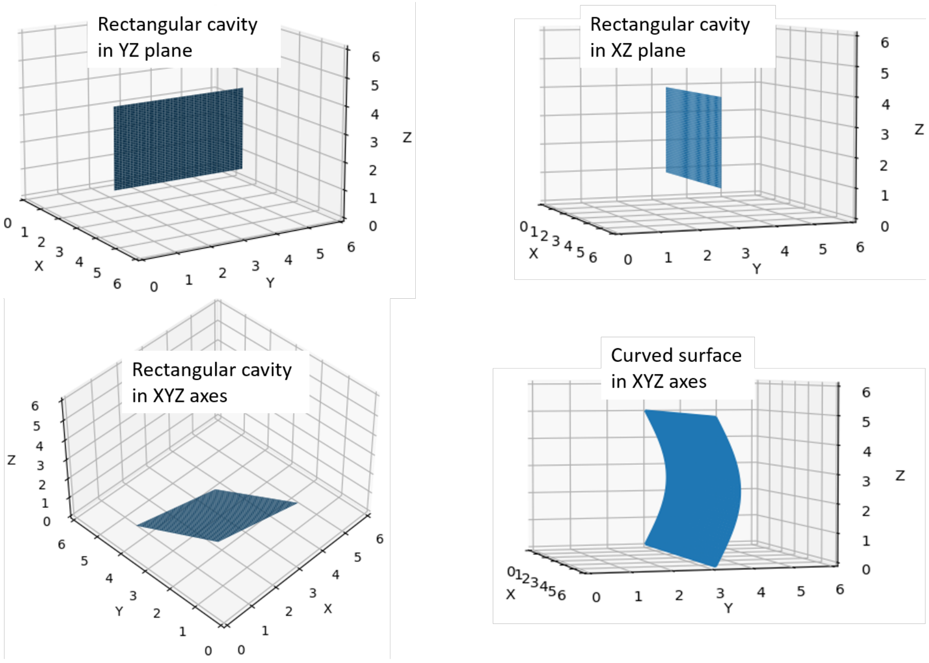


Fig. 50: Examples of cross-layer attacks: The footprint remains low on a single layer as PZ_{i_l}.

5.4 PrintSafe implementation case study

5.4.1 Testbed setup

The generic `PrintSafe` framework can work on any typical fused filament fabrication printer that permits the attachment of sensors to measure the movement of the monitored axes, whether it be from the stepper motors or the targeted moving part directly. For instance, our analysis of the product information shows `PrintSafe` implementation suitability for Ultimaker2/3/5/S5/S7, Prusa i3 MK3/MK4, Lulzbot TAZ6 / TAZ Pro S, and Creality 3D Ender-5 S1. For this particular experiment, we used an Ultimaker-3 printer controlled to open source Cura 5.0 software running on Windows 10 operating system hosted on a machine with a core i7-8700 processor and 16GB of RAM. The nozzles' diameter in this dual-nozzle printer is 0.4mm. We used

Algorithm 11 Aggregate space domain comparison: Non-conforming volumes list

```
Require: $LM_{G_n}$, $LM_{S_n}$ , $OM_G$ and $OM_S$
Ensure: Non-conforming volumes list
$OM_G \leftarrow OM_G$.append($LM_{G_n}$)   # Appending $n^{th}$ layer-map
$OM_S \leftarrow OM_S$.append(syncLM($LM_{G_n}$,$LM_{S_n}$))
for $i$ ranging from 1 to recentLayerCount do
    diff2DArray ← compare2Dspace($OM_{G_i}$,$OM_{S_i}$)
    diff2Doptimal ← optimizeLM(diff2DArray)
    diff3DArray ← diff3Darray.append(diff2Doptimal)
end for
# Finding contiguous mismatched zones
for each pixel $(k, i, j)$ in $diff3DArray$ do
    if diff3DArray[$k$][$i$][$j$] == 1 then
        zoneVol ← ($Pix_{res_x} \times Pix_{res_y} \times ht_k$)
        initiate(oVertexArray[$k$][$i$][$j$] )
        pushToStack(oVertexArray[$k$][$i$][$j$], $objStack$)
        while $objStack \neq \varnothing$ do
            pop($v$, $objStack$)
            for each neighbor pixel $(k', i', j')$ of $v$ do
                if $(k', i', j')$ is a valid pixel and $diff3DArray$[k'][i'][j']== 1 then
                    zoneVol ← $zoneVol + (Pix_{res_x} \times Pix_{res_y} \times ht_{k'})$
                    initiate($oVertexArray[k'][i'][j']$) # create a new node
                    objStack.push(oVertexArray[$k'$][$i'$][$j'$])
                end if
            end for
        end while
        if $zoneVol \neq 0$ then
            ncVolList.add($zoneVol$)
        end if
    end if
end for
return ncVolList # Non-conforming volume zones list
```

PLA 2.85 mm diameter filament for printing throughout the experiment.

The remainder of this section presents the implementation details for `PrintSafe` modules including physical deployment, fingerprinting, detection algorithms, and calibration.

5.4.2 Data Acquisition

The data acquisition module consists of a set of sensors and an accumulation network. The sensors are utilized to continuously acquire the direct-mainpulable sub-processes (kinetics and thermodynamics).

5.4.2.1 Independent Sensors

Table 23 presents three categories of sensors used to extract eight independent features of FFF printing. Three types of kinetics including nozzle, printing bed, and filament kinetics are monitored through ubiquitous optical encoders. The X,Y, and E (filament) axes are monitored through rotary encoders, and the Z axis (the printing bed) is monitored through a linear optical encoding strip.

Kinetic sensors: The printer offers more than one suitable place for kinetic monitoring. For X and Y axes monitoring, we installed the sensors on the power transfer system between the stepper motors and the extruder assembly. Both the sensors are installed on the outside of the printer by extending the driving shaft through a PLA-printed cylinder of 8 mm diameter and 15 mm length. The combined weight of the shaft and the optical disc is less than 2.5 grams, adding no noticeable burden on the system confirmed through a series of measurements before and after the deployment.

The printing bed kinetics is directly monitored through a linear encoding strip with a resolution of 500 cycles per inch of bed movement. The slow-moving filament

Process	**Monitored parameter**	**Sensor type**	**Manufacturer**	**Part number**	**Specs**	**Resolution as per system deployment**
Kinteic	X-axis	Optical-Rotary	US Digital	E2-512-315-NE-H-D-B	512 cycles/rev	0.1 mm
	Y-axis	Optical-Rotary	US Digital	E2-512-315-NE-H-D-B	512 cycles/rev	0.1 mm
	Z-axis	Optical-Linear Strip	US Digital	LIN-500-9.5-N	500 cycles/inch	0.012 mm
	Filament	Optical-Rotary	US Digital	E2-2000-315-IE-E-D-3	2000 cycles/rev	0.0035 mm
Thermodynamic	Nozzle Temperature	Thermocouple	Adafruit	Type-k & MAX31855	Upto 500^{o}C	0.25^{o}C
	Bed Temperature	Thermistor	Omega	SA1-TH-44006-40-T	Upto 120^{o}C	0.2^{o}C
	Fan speed	Acoustic	Movo	Lavalier USB-M1	Omni-dir., sensitivity: -30dB, upto 18KHz	5% variation

Table 23.: Sensors specifications for the case study

motor is captured through a higher resolution encoder (2000 cycles/rev) to ensure that consecutive printing state samples belonging to an extrusion move should be distinguishable. During the deployment of sensors, we suspect to exceed the tolerances for axial and radial play for the optical encoders (± 0.01 in and ± 0.04 in, respectively). However, due to the repetitive nature of this error, it is adjustable during the conversion from rotation to linear motion. These errors are handled and compensated during the calibration phase.

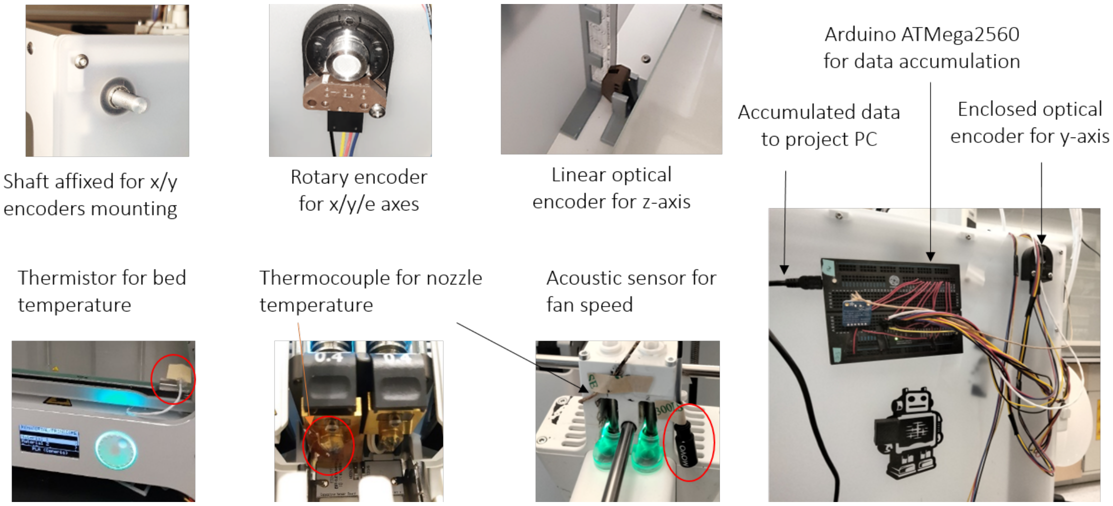

Shaft affixed for x/y encoders mounting
Rotary encoder for x/y/e axes
Linear optical encoder for z-axis
Arduino ATMega2560 for data accumulation
Accumulated data to project PC
Enclosed optical encoder for y-axis
Thermistor for bed temperature
Thermocouple for nozzle temperature
Acoustic sensor for fan speed

Fig. 51: Deployment of sensors on the case study printer

Thermodynamic sensors: There are two categories of actors that influence thermodynamics: heaters and coolers. The case study printer includes a nozzle heater and a printing bed heater. A thermistor is installed on a corner of the printing bed to measure the bed temperature. For the nozzle temperature monitoring, a type-K thermocouple is used that is conveniently routed to the tip of the nozzle within the extruder assembly.

Acoustic sensor: To estimate the fan speed of the part cooling fans, and the fan state of the hotend cooling fan, we utilize an affordable omnidirectional acoustic sensor called the 'Movo Lavalier USB-M1'. Although the sensor is installed on the extruder near the fans, it has many acoustic sources in the vicinity and requires some pre-processing and filtering to ascertain the state of both types of fans.

5.4.2.2 Data Accumulation

The acoustic sensor for fan speed estimation uses a standard USB serial interface and is connected to the project machine running the `PrintSafe` detection engine. All other sensors, including kinetic and thermodynamic sensors, provide raw binary or analog data. To aggregate this data, we use an Arduino ATMega 2560 board developed by Stemtera in a breadboard form factor. Interrupt service routines capture the fast-moving kinetic data from the sensors, while the thermodynamic data is periodically polled.

To send data from the accumulation stage to the project repository, we utilize two sampling rates. We sample the faster samples, S_{fast}, every 5 ms, and the slower samples, S_{steady}, which contain a complete snapshot of the sensors, are sent every 50 ms. We chose these values to align with the printing speed limits, the sensor resolution, and the capacity of the data accumulation system. Equation 5.5 outlines the components of both samples. In this equation, S' and T' represent the starting and terminating characters, t represents the time, and x, y, z, and e represent the X, Y, Z, and filament axes, respectively. T_n and T_b represent the nozzle and printing bed temperature, respectively.

$$S_{fast_i} = \{`S', t_i, x_i, y_i, `T'\}\,;\;\; S_{steady_i} = \{`S', t_i, x_i, y_i, z_i, e_i, T_{n_i}, T_{b_i}, `T'\} \tag{5.5}$$

5.4.3 Sensor data transformation

5.4.3.1 Conversion of acoustic data to fan speed

While kinetic sources produce acoustic profiles that undergo continuous variation, the acoustic profiles of the heated nozzle (hotend) fan and the part cooling fans during normal printing operations remain relatively stable. To study the acoustic profile in the frequency domain and its relation to fan speed, the acoustic profile was observed by generating a set of audio files covering the fan states and speeds under a random system and environmental noise. The microphone was placed near the extruder fans, minimizing the effect of environmental noise. The process for generating these audio files is presented in Algorithm 12.

Once the files are created, they are transformed to the frequency domain using Fast Fourier Transform and the power spectrum. To identify the fan acoustic pattern, localized frequency peaks are extracted throughout the spectrum. Although the exercise was started across the complete range of the acoustic sensor, the interesting frequencies, including noticeable harmonics related to the fan acoustics, are covered within 3000 Hz. Algorithm 13 presents the calculation of local peak frequencies and their magnitudes from the audio sample. Figures 52 and 53 present the power spectrum graphs for the training audio samples for two specific zones. These frequency ranges offer the best discriminating features in the presence of system and environmental acoustic noise. As the hotend cooling fan speed is not manipulated through G-codes in standard Marlin firmware, its state is only identified as ON or OFF. For the part cooling fan, the speed is estimated using two interpolation functions and then classified into six classes. The finalized rules for the hotend fan state estimation is mentioned in Equation 5.6, where $State_{HEF}$ represents the state of the hotend fan and 'peak_freqs' is the array of the local frequency peaks extracted through Algorithm

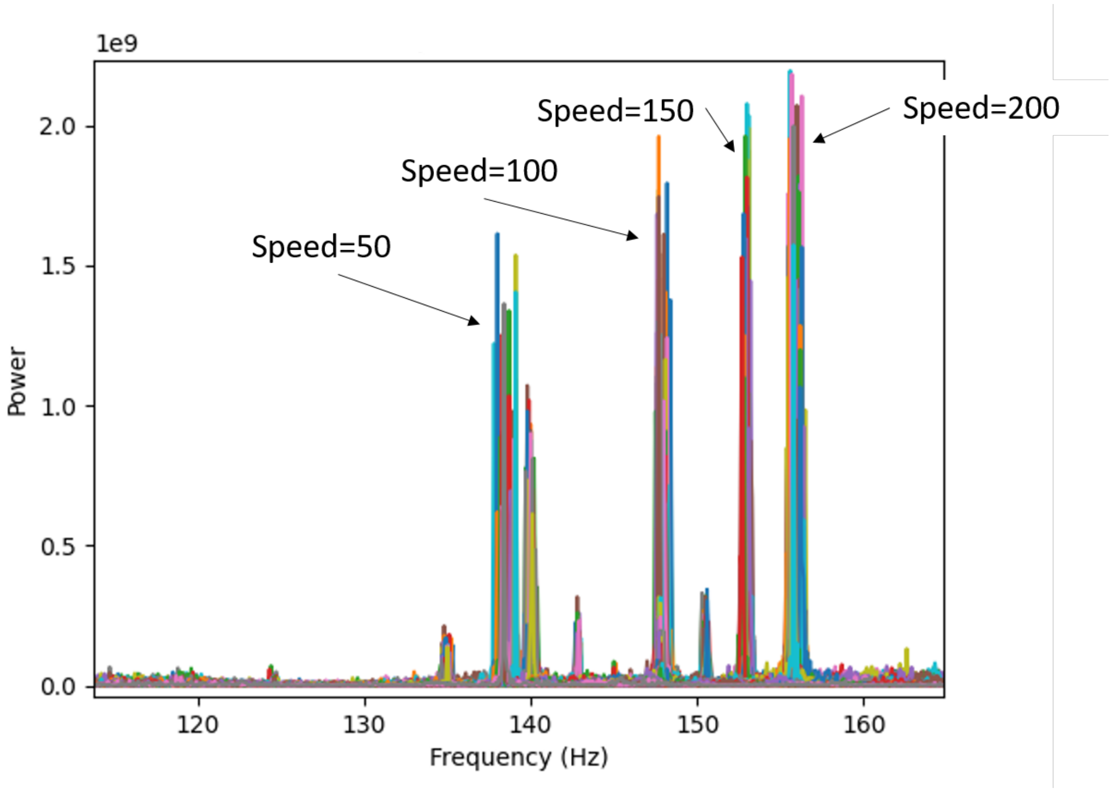


Fig. 52: Power spectrum at different speeds of part cooling fans in frequency zone 1 between 100 Hz and 200 Hz

13. The speed of the part cooling fans, $Speed_{PCF}$, is estimated using Equation 5.7 where $interp1$ and $interp2$ are the polynomial interpolation functions learned from the training data in two distinct frequency zones. $SpeedCat$ function picks the speed category nearest to the estimated speed.

$$\text{State}_{HEF} = \begin{cases} ON & \text{if } (\exists i \in [0,1,2] : 1220 \leq \text{peakFreq}_i \leq 1250) \\ & \quad \wedge (\exists f \in \{\text{peakFreqs}\}, 610 \leq f \leq 625) \\ & \quad \wedge (\exists f \in \{\text{peakFreqs}\}, 2440 \leq f \leq 2500) \\ OFF & \text{otherwise} \end{cases} \tag{5.6}$$

Algorithm 12 Audio files generation for fan speed estimation

```
AxialFanState ← $0, 1$
RadialFanState ← $0, 50, 100, 150, 200, 255$    ▷ Part cooling fan speed categories
timeDuration ← 10 seconds
fileCount ← 20    ▷ sample files per pattern
$t_{\text{min}}$ = AxialFanONtemperature
sendGcodeCommand(coolExtruder)
while $t_{\text{ext}} > t_{\text{min}}$ do
    wait()
end while
generateRandomMoves()    ▷ Initiate a dedicated thread to continuously send move instructions
connectAudioStream(samplingRate, serialPort, channel)
for $r$ in RadialFanState do
    setFanSpeed($r$)
    for $f$ in range($0$, fileCount) do
        for $t$ in range($0, t_{\text{audioFile}}$) do
            data = audioStream.read(samplingRate)
            frames.append(data)
        end for
        saveAudioFile($f, a = 0, r$, frames)
    end for
end for
setNozzleTemperature($2 * t_{\text{min}}$)
for $r$ in RadialFanState do
    setFanSpeed($r$)
    for $f$ in range($0$, fileCount) do
        for $t$ in range($0, t_{\text{audioFile}}$) do
            data = audioStream.read(samplingRate)
            frames.append(data)
        end for
        saveAudioFile($f, a = 1, r$, frames)    ▷ 10 seconds audio file saved; name represents the parameters
    end for
end for
```

Algorithm 13 Extracting power spectrum peak frequencies and magnitudes

```
Require: audio_file, Parameters: no_of_peaks, ratio_clutter, one_peak_zone,
Ensure: filtered_peak_frequencies, filtered_peak_powers
audio_signal, sample_rate ← load(audio_file)
fft ← fft(audio_signal)
power_spectrum ← $|fft|^2$
freqs ← fftfreq(len(power_spectrum), d = 1/sample_rate)
freqs ← freqs[: len(freqs)//2]
power_spectrum ← power_spectrum[: len(power_spectrum)//2]
peak_power ← max(power_spectrum)
for i in range(len(power_spectrum)) do
    if power_spectrum[i] < (1/ratio_clutter) * peak_power then
        power_spectrum[i] ← 0
    end if
end for
peak_indices ← argsort(power_spectrum)[:: −1][: no_of_peaks]
peak_frequencies ← freqs[peak_indices]
peak_powers ← power_spectrum[peak_indices]
filtered_peak_indices ← []
for i, (f1, p1) in enumerate(zip(peak_frequencies, peak_powers)) do
    retain_peak ← True
    for j in filtered_peak_indices do
        f2, p2 ← peak_frequencies[j], peak_powers[j]
        if $|f1 - f2| \le$ one_peak_zone then
            if $p1 > p2$ then
                filtered_peak_indices.remove(j)
            else
                retain_peak ← False
                break
            end if
        end if
    end for
    if retain_peak then
        filtered_peak_indices.append(i)
    end if
end for
filtered_peak_frequencies ← peak_frequencies[filtered_peak_indices]
filtered_peak_powers ← peak_powers[filtered_peak_indices]
return filtered_peak_frequencies, filtered_peak_powers
```

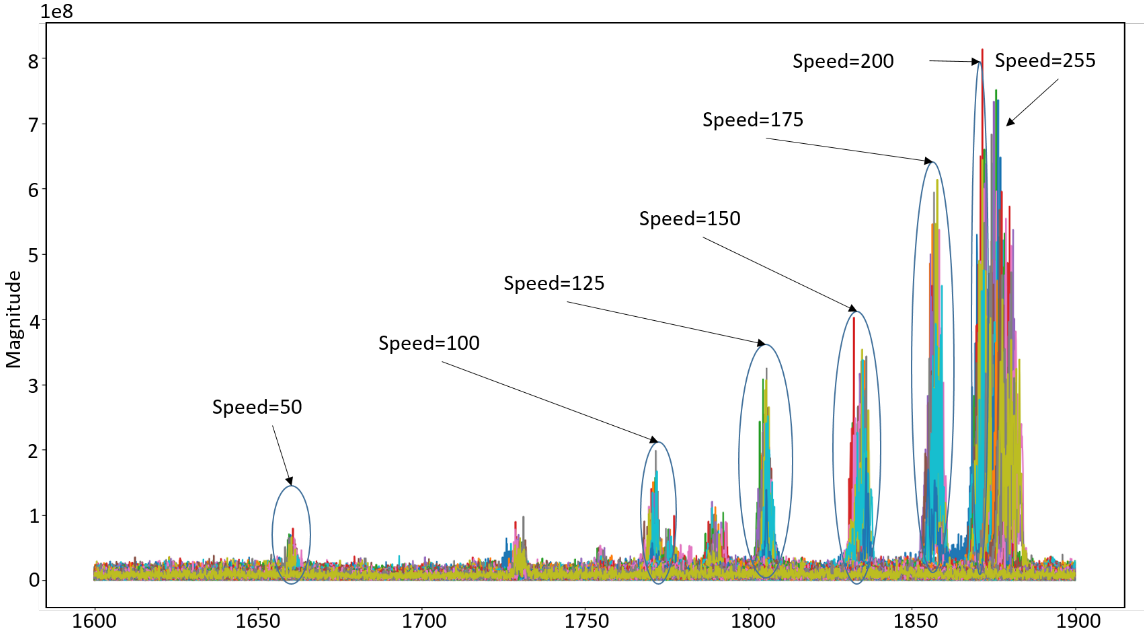


Fig. 53: Power spectrum at different speeds of part cooling fans in frequency zone 2

$$\text{Speed}_{PCF} = \begin{cases} \text{speedCat}(\text{interp}_1(f_1)) & \text{if } \exists f_1 \in \{\text{peakFreqs}_i\}_{i=0}^{2}, 135 \leq f_1 < 155) \\ \text{speedCat}(\text{interp}_2(f_2)) & \text{if } (\exists f_1 \in \{\text{peakFreqs}_i\}_{i=0}^{2}, 155 \leq f1 \leq 159) \\ & \wedge(\exists f_2 \in \{\text{peakFreqs}\}, 1860 \leq f2 \leq 1880) \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

5.4.3.2 Transformation of sensors data to space-domain

The sensors data is typically a time-sampled series of fast and steady samples. By utilizing the logic presented in algorithm 6, we transform the sensors data in the space domain. In addition to the object-map OM_S, we generate LM_{S_n} for each layer. Each entry in LM represents the attributes of each pixel of 0.1 mm x 0.1mm dimensions.

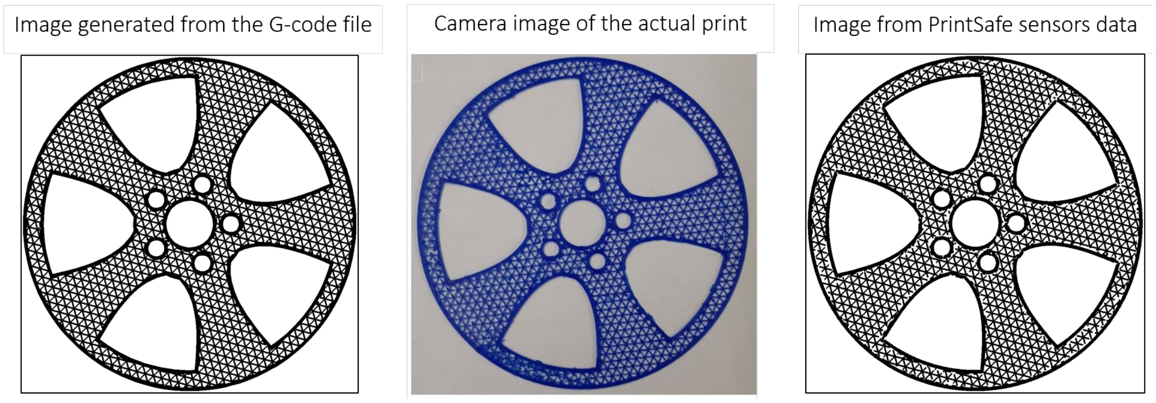


Image generated from the G-code file
Camera image of the actual print
Image from PrintSafe sensors data

Fig. 54: Space domain representation: G-code synthesized data (left), actual image of the printed layer (center), and the image from the sensors data (right)

Figure 54 shows a cross-sectional view of a 5 cm radius car wheel model, with the outer dimensions and infill lines accurately captured with sub-millimeter precision. This visualization provides valuable insights into the internal structure and potential malicious manipulations.

5.4.4 Firmware functions profiling

In this case study, we have profiled two firmware functions where the kinetic and thermodynamic footprints are not rigidly coupled to the G-code instructions.

5.4.4.1 Axes homing function

The fingerprint of axes homing function is generated using Algorithm 7. The three stages, namely Y-homing-final’, X-homing-final’, and ‘Z-homing-final’, have a distinct signature and are strictly linked in reverse order. These stages involve re-bumping at a slower speed to re-engage the limit switch after it is initially hit. Depending upon the starting state, there could be variable length moves instructions filled within the final homing patterns. Figure 55 presents the kinetic sensors

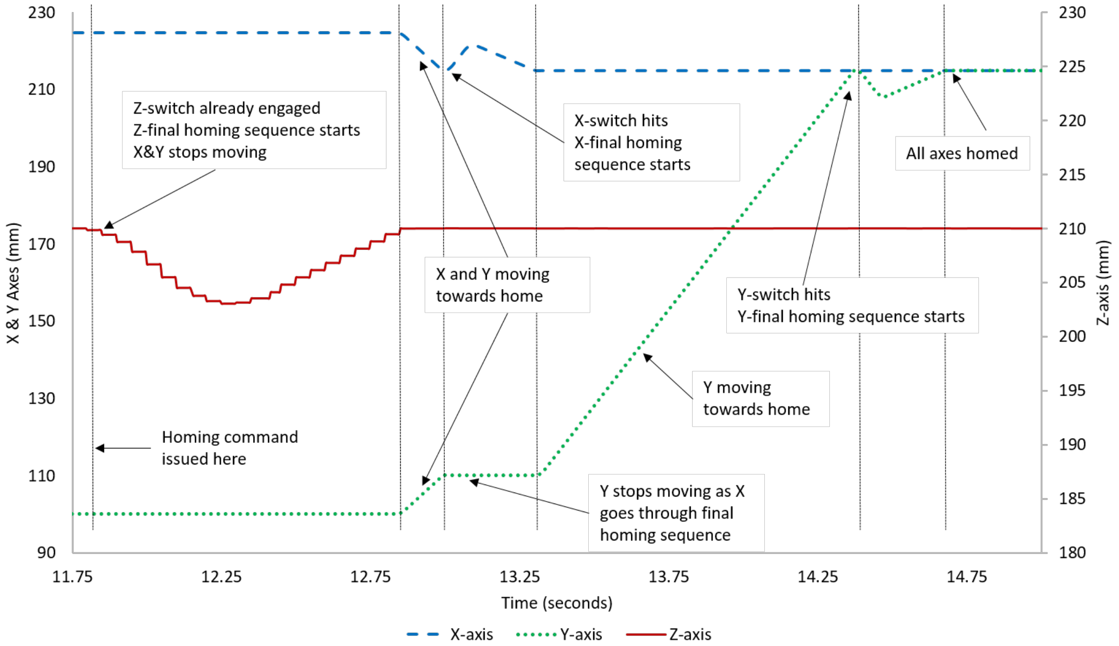


Fig. 55: Sensors data depiction for homing function profile

data of X,Y and Z axes. When the homing command was issued, the Z-axis was already homed, hence 'Z-homing-final' move is immediately triggered whereas X and Y motors wait. Once Z-axis is homed, X and Y motors simultaneously move towards their homing positions before X-axis hits the limit switch and transitions to 'X-final-homing' sequence while the Y motor waits. Once X-axis is homed, Y motor moves towards the homing position and transitions to 'Y-homing-final' state before hitting the limit switch again to complete the homing function.

5.4.4.2 Fingerprinting the automatic bed leveling function

Automatic bed leveling is a crucial firmware function that influences the printing profile and impacts most attack detection schemes. It involves several lower-tier states, including all-axes homing, printhead homing, move-to-the-next-point, slow-leveling-sequence, fast-leveling-sequence, and nozzles swapping. To generate test cases, we can follow the example of axes-homing, extract common and fixed patterns and their correlations, and create a fingerprint.

The case study printer uses three-point bed leveling in a strict sequence of subtasks. The automatic leveling function begins with an all-axes-homing task, followed by moving the extruder to a specific location and waiting for the printing bed and nozzle to reach the desired temperature values. The printer then initiates a sequence of leveling operations using both the nozzles and different points on the printing bed, followed by the all-axes-homing sequence to complete the function. The nozzle and the printing bed may take several minutes to reach the desired temperature.

Figure 56 shows one snapshot of the five sensor data after the printing bed and nozzle reaches the desired temperature. Although the sequence is long, multiple attempts for the printing bed leveling initiated from different states result in the same sequence of actions.

5.4.5 Incorporating Firmware Functions into Sensor Data

This module takes the fingerprint of firmware functions and uses them to adjust the sensor data. In the case of automatic bed leveling, `PrintSafe` matches the fingerprint of the leveling function and then measures and updates the z-axis values corresponding to the three non-collinear points on the printing bed. These points, denoted by $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$, and $P_3 = (x_3, y_3, z_3)$, are learned during the latest auto-leveling exercise.

To determine the compensatory z-value for any point $P = (x, y, z)$ on the printing bed, we use Equation 5.8. This equation utilizes the three points on the bed to calculate the equation of the plane that they define. We can then use this equation to determine the z-value for any point on the bed, allowing us to nullify the effect

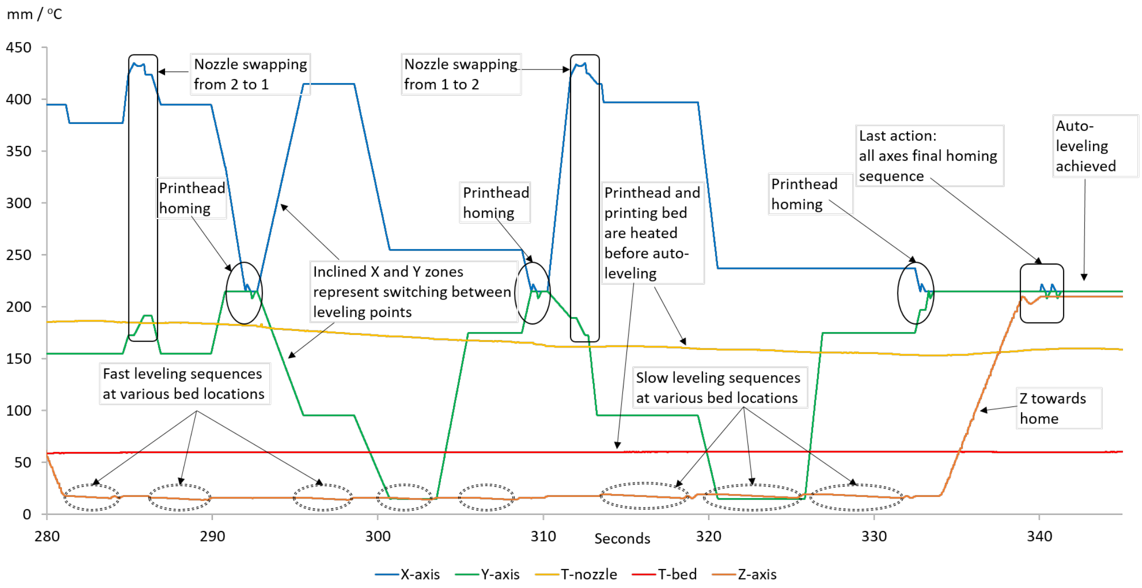


Fig. 56: Kinetic and thermodynamic sensors data captured during an automatic printing bed leveling exercise

of continuous z-axis movement due to automatic bed leveling. During printing, the z-axis value is updated for every incoming sample based on the sample's location on the printing bed. This process ensures that all the process evaluation algorithms pertaining to G-code versus sensor data set can be utilized without modification.

$$\begin{aligned}
&((y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1))(x - x_1) \\
&+ ((z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z3 - z_1))(y - y_1) \qquad (5.8) \\
&+ ((x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1))(z - z_1) = 0
\end{aligned}$$

5.4.6 Process Analysis

The case study utilizes all the process analysis algorithms that were presented in Section 5.3.4. These algorithms are employed to examine all the aspects that were mentioned in the framework illustrated in Figure 46. The implementation covers all the sabotage attacks available in the existing literature covered in Section 5.2. Most of the algorithms used in the case study are implemented in Java. However, some of the work related to fingerprinting is performed using Python. Separate threads are used for data acquisition and analysis. The performance of the system is discussed in detail in the evaluation section.

5.4.7 PrintSafe Calibration

To ensure accurate and reliable results, `PrintSafe` is calibrated at the beginning of the experiment to compensate for various errors in sensing resolution, measurement, quantization, and printing process. The overall effect of calibration is incorporated in the minimum detection threshold values, which are set to ensure zero false positives and zero false negatives.

5.4.7.1 Finding Optimal Threshold Settings

This section outlines the process of finding optimal threshold values for `PrintSafe` to accurately detect integrity violations. To obtain the optimal threshold values, we collected traces of 20 objects with varying shapes, numbers of layers, infill patterns, and densities. For each parameter, we looked for a value that would result in zero false positives for the training prints. The results are presented in Table 24, and a brief account of the selection process is discussed in the following subsections.

5.4.7.2 Geometry Parameters

`PrintSafe` considers not only the outer geometry of an object but also the infill pattern as part of the geometry and matches each infill line for its location and dimension. We used seven parameters for layer geometry, including the biggest and cumulative area mismatch, path and vertex deviation for G-code commands, filament consumption difference per command and per layer, and max z-axis difference. We individually tuned each parameter to a level where PrintSafe passes all benign prints and highlights the reason for failure.

As part of the layer geometry integrity check, we verified the filament consumption on a per-move and per-layer basis. For short move commands, typically 1 to 5 mm, the filament consumption may differ up to 3.5% due to sampling error. However, this error is not accumulated and is compensated in the next move. For longer moves, the error is less than 1% in all benign cases. We selected a 1% alert threshold per layer and a 5% alert threshold for a single move. If the layer's mass changes by 1%, or if any move command uses 5% extra or less filament, it is considered an integrity compromise.

5.4.7.3 Thermodynamic Profile

Heating Elements In the investigation of PrintSafeś effectiveness in detecting anomalies during the printing process, we discovered that the temperature reading from PrintSafeś sensor for the nozzle was 2-3^{o}C less than the reading from the printer's internal sensor. This difference arose because PrintSafeś sensor is located beside the nozzle tip, whereas the internal sensor is positioned slightly farther from the extrusion point and closer to the heating element, resulting in a marginally higher temperature reading. We factored in this consistent difference in our algorithm. We also noted that the actual nozzle temperature could fluctuate by 2-3^{o}C along a hysteresis curve. We performed mechanical testing during the attack study and discovered that temperature changes of less than 5^{o}C did not have a significant effect on the object's properties. Therefore, to account for these benign fluctuations and approximations, we set the alert threshold to 4^{o}C.

Similarly, we observed a temperature difference of 1-2^{o}C for the printing bed. Many research studies that examined the impact of changes in bed temperature on object quality have taken ΔT to be 10^{o}C, 20^{o}C, or higher [80, 81]. We found it reasonable to trigger an alert at 3^{o}C deviation.

Cooling Elements The hotend cooling fan is responsible for controlling the temperature of the heated nozzle. The fan remains ON after the nozzle reaches a specific temperature, and its speed cannot be controlled through G-code instruction. `PrintSafe` considers the hotend cooling fan to be in either the ON or OFF state and can detect a reliable change in state using frequency power spectrogram analysis. The part cooling fans are installed in pairs, and their speed can be manipulated through a G-code instruction $M106Sxyz$, where the value of xyz ranges from 0 to 255 (maximum speed). Researchers have shown that reducing the fan speed to 50% can have an impact on the printed part. In our study, we introduced six zones ranging from zero to maximum speed. `PrintSafe` can reliably detect variations of more than 5% in speed, implying no false negatives for any inter-zone speed fluctuations.

5.4.7.4 Timing Profile

The printing process involves three sources of timing: the layer-end time of the G-code file, the result of `PrintSafe` modeling, and the actual printing time. After estimating the time-profile, `PrintSafe` compensates for the difference between the layer-printing time provided by the G-code and the actual printing time by distributing the error uniformly. Since we are working on a millisecond scale, this approximation seems appropriate. However, we noticed that the G-code layer-printing time does not always match the actual printing time. To account for this inaccuracy, we relaxed the time window to 2 seconds. We set a minimum duration of 0.5 seconds for a mismatch to persist, and a minimum of 2

These timing parameters, along with the other parameters described in the previous subsections, were optimized to achieve zero false positives, which defines the claimed detection performance of PrintSafe. Table 24 specifies the parameters, related attack types, and their final values that we use as the alert thresholds.

5.5 Evaluating PrintSafe for attack detection

PrintSafeś performance is scrutinized against a corpus of 33 sophisticated attacks. A majority of the attacks are executed on rectangular prisms of dimensions 50 mm x 50 mm x 4 mm, and rectangular bars of dimensions 60 mm x 6 mm x 4 mm. The objects are sliced using a layer height of 0.2 mm, a printing speed of 50 mm/sec, 205^{o}C nozzle temperature, and 2 top and bottom layers. With the exception of infill pattern and density-related attacks, the 'LINE' type infill pattern at a 45^o raster angle is employed. For each attack instance, three samples are printed.

5.5.1 Attacks selection

To assess the efficacy of PrintSafe in detecting anomalies during the printing process, we have devised a comprehensive suite of sabotage attacks that target all aspects of the process that can be directly manipulated, such as nozzle kinetics, z-kinetics, filament kinetics, nozzle thermodynamics (including heating and cooling functions), and bed thermodynamics. This suite encompasses all known sabotage attacks documented in the literature, and we have taken into consideration the possibility that an attacker could compromise any stage of the printing process, including design, slicing, and printing, using any existing techniques [77, 24]. While some of the attacks included in the study can only be launched from a specific stage of the process (for example, an object scaling attack can only be launched if the design file

S.N	Performance Parameters	Category	Alert Thresholds
1	Single mismatched area	Layer geometry	If the mismatched area is greater than 1 mm^2
2	Detection of cumulative mismatched area	Layer geometry	If cumulative mismatch exceeds 2% per layer, and min dimension is ¿ 0.2 mm
3	Detection of nozzle temperature deviation	Nozzle thermodynamics	If the nozzle temperature deviates by more than 4^{o}C
4	Detection of bed temperature difference	Bed thermodynamics	If the bed temperature difference exceeds 3^{o}C for 500 ms
5	Detection of hotend cooling fan state mismatch	Nozzle thermodynamics	If axial fan binary state mismatches with the design
6	Detection of speed manipulation of part cooling fan	Nozzle thermodynamics	Speed categorized in 6 zones; alert if cooling fan speed zone mismatches
7	Mismatched samples count	Timing Profile	If samples mismatch per layer exceeds 2%
8	Mismatch period	Timing Profile	Continuous mismatch period ≥ 500 ms
9	Detection of maximum layer thickness difference	Geometry	If the maximum layer thickness difference is greater than 0.05 mm for 500 ms
10	Sample search window	Timing Profile	If sample search time exceeds 2 seconds
11	Detection of filament consumption change per move	Density, Geometry	If the filament consumption change per move is greater than 5%
12	Detection of filament consumption change per layer	Density, Geometry	If the filament consumption deviation per layer is greater than 1%
13	Detection of maximum nozzle deviation	Layer geometry	If the maximum nozzle deviation is greater than 0.75 mm perpendicular to the move path
14	Detection of maximum vertex deviation	Layer geometry	If the maximum vertex deviation is greater than 0.75 mm
15	Detection of single mismatched volume	Object geometry	If the single mismatched volume (any shape) is greater than 0.05 mm^3
16	Detection of cumulative mismatched volume	Object geometry	If the cumulative mismatched volume is greater than 1%

Table 24.: Thresholds for detecting performance PrintSafe

is obtained, while a hotend cooling fan state attack can only be launched through printer firmware), most of the attacks can be launched from multiple stages. The attack magnitudes have been chosen to exceed the detection thresholds outlined in Table 24."

5.5.2 Attack results

Scaling attacks Two variants of scaling attacks are launched by increasing the size of the object by factors of 1.2 and 1.1. Although PrintSafe does not directly measure the scaling factor, the overall attack footprint for the selected objects is large enough to be detected. If the layer dimensions deviate by 0.3mm or more due to scaling, PrintSafe can reliably detect it on the same layer. Scaling also causes the infill pattern to relocate and alters the number of printing instructions and timing profiles. Thus, these attacks are relatively easy to detect.

Fitment attacks In this category of attacks, a feature such as a slot or a shaft is intentionally misaligned to prevent it from coupling with its intended counterpart. In one variant, a through slot in the center of a rectangular prism is shifted by 0.5 mm per layer along the x-axis. In another variant, the shift is reduced to 0.1 mm per layer. While the first variant is detected on the first attacked layer, the deviation of 0.1 mm is currently undetectable in the implemented system. However, as the deviation accumulates over the layers, it can be detected after the third layer.

Internal geometry attacks This category of attacks encompasses actions such as cavity insertion and deletion, as well as manipulation of infill patterns and densities. For the purpose of single layer detection in PrintSafe, a cavity should have a cross-sectional area of at least 1mm^2. However, PrintSafe's comprehensive object-map analysis allows for the detection of smaller cavities if the cumulative deviation volume exceeds 0.05 mm^3. By examining the mismatched volumes, both the insertion and deletion of cavities are considered violations of integrity by `PrintSafe`.

Z-profile attacks The efficacy of `PrintSafe` in mitigating Layer Thickness Attacks on the z-axis was assessed by conducting four distinct attacks. The first attack involved modifying the layer thickness parameter from 0.2 mm to 0.1 mm at the slicing stage, which effectively doubled the number of layers. In the subsequent two attacks, the layer thickness parameter was altered for selective layers. When the attack intensity was 0.1 mm, the `PrintSafe` system was able to detect it promptly on the first attacked layer. However, when the attack intensity was 0.04 mm, `PrintSafe` failed to identify the attack on the first layer but was able to detect it reliably on the second layer. Another variation of the attack involved altering the layer-thickness for a specific feature by 0.06 mm, which resulted in the layer-change event and triggered the alert system.

Nozzle thermodynamics attacks The nozzle's thermal profile can be influenced by modifications to the heating or cooling profiles, or both, in order to introduce residual thermal stresses in the printed part. Modifying the heating profile involves utilizing the "M104" command to reduce the temperature by 5^{o}C while the extruder is printing the central portion of the rectangular bars. Once the extruder crosses the target zone, the temperature starts returning to normal. Figure 57 presents a heatmap capturing one instance of this attack detected by `PrintSafe`.

Two attacks were conducted on the cooling profile. The first attack involved using the "M106" command to adjust the part cooling fan speed by 33%, achieving four different speeds (0%, 33%, 66%, and 100%). Each speed was maintained for

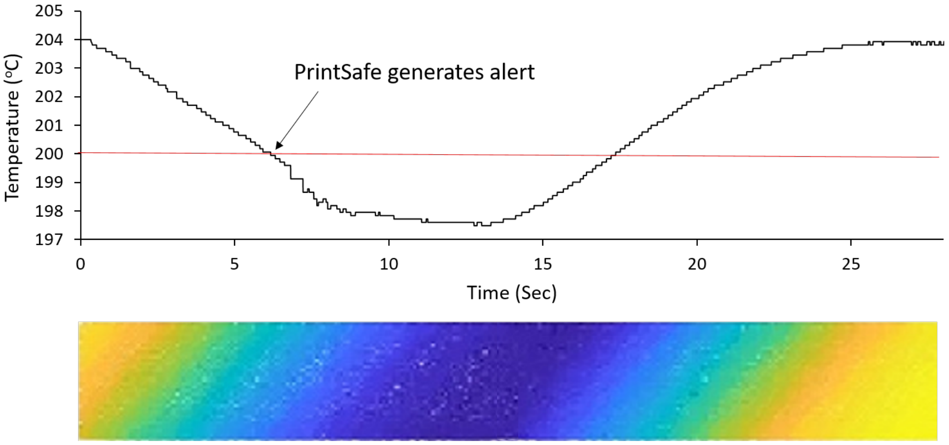
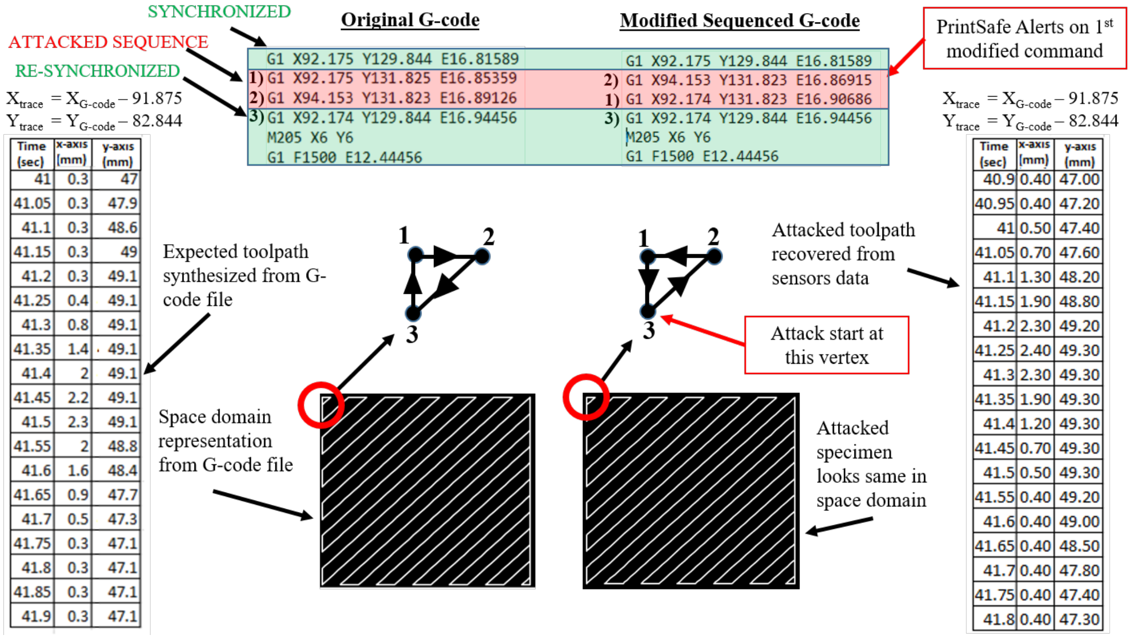


Fig. 57: Nozzle thermal profile heatmap for the attacked sample

30 seconds before being reverted back to the reference value. The second attack was carried out through Marlin firmware by modifying the value of 'EXTRUDER_AUTO_FAN_TEMPERATURE' from 40 to 240, preventing the operation of the hotend fan once the nozzle temperature reached 40oC. `PrintSafe` reliably detected all instances of these attacks.

Printing bed thermodynamics attacks In this attack, the attacker reduces the printing bed temperature 10^{o}C below the value specified in the original G-code file to induce warping and poor adhesion. We implemented two instances of these attacks. In the first instance, the initial printing bed temperature value is modified. As achieving the glass transition temperature mentioned in the G-code file is a prerequisite to start printing, `PrintSafe` alerts on the first layer about this anomaly. In the other instance, the attacker modifies the printing bed temperature during the third layer. As the bed temperature is a slow varying process, it took around 35 seconds to cross `PrintSafe` alert detection threshold. Hence this attack was detected on the next layer.


SYNCHRONIZED
ATTACKED SEQUENCE
RE-SYNCHRONIZED
PrintSafe Alerts on 1st modified command
Expected toolpath synthesized from G-code file
Space domain representation from G-code file
Attacked toolpath recovered from sensors data
Attack start at this vertex
Attacked specimen looks same in space domain
1
2
3
1
2
3

Original G-code

```
   G1 X92.175 Y129.844 E16.81589
1) G1 X92.175 Y131.825 E16.85359
2) G1 X94.153 Y131.823 E16.89126
3) G1 X92.174 Y129.844 E16.94456
   M205 X6 Y6
   G1 F1500 E12.44456
```

Modified Sequenced G-code

```
   G1 X92.175 Y129.844 E16.81589
2) G1 X94.153 Y131.823 E16.86915
1) G1 X92.174 Y131.823 E16.90686
3) G1 X92.174 Y129.844 E16.94456
   M205 X6 Y6
   G1 F1500 E12.44456
```

$X_{trace} = X_{G\text{-}code} - 91.875$
$Y_{trace} = Y_{G\text{-}code} - 82.844$

Time (sec)	x-axis (mm)	y-axis (mm)
41	0.3	47
41.05	0.3	47.9
41.1	0.3	48.6
41.15	0.3	49
41.2	0.3	49.1
41.25	0.4	49.1
41.3	0.8	49.1
41.35	1.4	49.1
41.4	2	49.1
41.45	2.2	49.1
41.5	2.3	49.1
41.55	2	48.8
41.6	1.6	48.4
41.65	0.9	47.7
41.7	0.5	47.3
41.75	0.3	47.1
41.8	0.3	47.1
41.85	0.3	47.1
41.9	0.3	47.1

$X_{trace} = X_{G\text{-}code} - 91.875$
$Y_{trace} = Y_{G\text{-}code} - 82.844$

Time (sec)	x-axis (mm)	y-axis (mm)
40.9	0.40	47.00
40.95	0.40	47.20
41	0.50	47.40
41.05	0.70	47.60
41.1	1.30	48.20
41.15	1.90	48.80
41.2	2.30	49.20
41.25	2.40	49.30
41.3	2.30	49.30
41.35	1.90	49.30
41.4	1.20	49.30
41.45	0.70	49.30
41.5	0.50	49.30
41.55	0.40	49.20
41.6	0.40	49.00
41.65	0.40	48.50
41.7	0.40	47.80
41.75	0.40	47.40
41.8	0.40	47.30

Fig. 58: Timing profile attack: A localized re-sequencing of three G-code move commands

Timing profile attacks The timing profile is an intriguing category of attacks that includes anisotropy, manipulation of printing speed, insertion or removal of commands, and re-sequencing attacks. Figure 58 shows an example of a low-footprint command re-sequencing attack conducted on an internal layer of a rectangular prism. The original toolpath follows the sequence '3-1-2-3' in a clockwise direction to print the top-left vertex. The attacker modifies the sequence to a counterclockwise direction. Although both the attacked and benign specimens appear the same in the space domain and the timing profile deviation is less than one second (below the detection threshold), the G-code command verification algorithm 10 detects this re-sequencing on the first modified command.

Filament kinetics attacks This category of attacks exploits the filament-kinetics process to induce density variations in the printed object. We conducted two variants

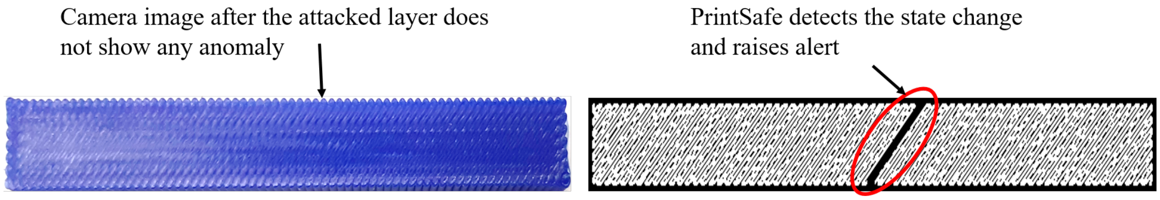

Camera image after the attacked layer does not show any anomaly
PrintSafe detects the state change and raises alert

Fig. 59: Filament-kinetic state manipulation attack

of these attacks. In the first attack, the filament motor state is toggled to reduce the filament density at the target zone. As presented in Figure 59, the attack leaves no conspicuous visual effect due to the residual filament's presence on the tip of the nozzle. Since PrintSafe monitors the filament motion, the attack is successfully detected by all three per-layer detection algorithms. In the second variant, the attacker only modifies the speed of the filament motor to reduce the overall deposited material by 3% for 20% of G-code commands pertaining to the target area of the printed object. PrintSafe successfully detects the attack through cumulative tests of 1% deviation in the filament consumption per layer.

5.5.3 Analysis and discussion

The results show that PrintSafe is capable of detecting sabotage attacks on all direct-manipulable processes. The attacks cover the known attacks in the literature. PrintSafe considerably improves the detection horizon. For thermodynamics (heating and cooling processes) and toolpath resequencing attacks, PrintSafe is capable to detect far more accurately than the feasible sabotage attacks. Kinetic manipulations below 1 mm^2 contiguous mismatched area per layer fall within the confusion zone and are ignored by PrintSafe to avoid false positives. The low-magnitude sabotage attacks in the literature try to magnify the attack impact by replicating the attack over multiple layers. PrintSafe accumulative analysis module keeps track of

S/No	Attack action	Printing aspect	Attack magnitude	PrintSafe performance	Other methods
1	Object scaling	Outer geometry	1.2x, 1.1x	Detected if Δ x/y ≥ 0.3 mm	Not specified
2	Feature scaling	Outer geometry	1.2x , 1.1x	Detected if Δ x/y ≥ 0.3 mm	Not specified
3	Feature misalignment	Fitment / geometry	A slot is drifted by 0.5 mm per layer	Detected on the 1^{st} attacked layer	Not detected
4	Feature misalignment	Fitment / geometry	A slot is drifted by 0.1 mm per layer	Detected on the 3^{rd} attacked layer	Not detected
5	Inserting cavity at designing stage	Internal geometry	0.1 mm x 0.1mm x 5 layers	Detected on 1^{st} layer due to toolpath impact	4 mm over one axis [75]
6	Removing cavity at designing stage	Internal geometry	Filled up existing 0.1mm x 0.1mm x 5 layers	Detected on 1^{st} layer due to toolpath impact	4 mm over one axis [75]
7	Inserting cavity at post-slicing stage	Internal geometry	1 mm x 1 mm x 5 layers	Detected on 1^{st} layer	4 mm over one axis [75]
8	Inserting cross-layer cavity at post-slicing stage	Internal geometry	0.3 mm x 0.3 mm x 10 layers	Detected on 1^{st} layer if Δ e per cmd ≥ 5%, OR when mismatched vol ≥ 0.05 mm^3	Not detected
9	Removing cavity at post-slicing stage	Internal geometry	0.3 mm x 0.3 mm x 10 layers	Detected on 1^{st} layer only if Δ e per cmd ≥ 5%, Else detected on 3rd layer as mismatched vol ≥ 0.05 mm^3	Not detected
10	Inserting cross-layer irregular cavity	Internal geometry	Irregular shape, but each layer footprint ≤ $1mm^2$	Detected if delta e per cmd ≥ 5; OR when the mismatched vol ≥ 0.05 mm^3	Not detected
11	Changing 20% infill density to 21%	Internal geometry	Used Grid type pattern	Detected on the 1^{st} infill layer	±10% change [18], if Δt ≥ 1sec [20]
12	Changing 40% infill density to 41%	Internal geometry	Used Line type pattern	Detected on the 1^{st} infill layer	±10% change [18], if Δt > 1sec [20]
13	Changing infill lines angle 45 to 44	Internal geometry	Used Line type pattern	Detected on the 1^{st} infill layer	Not specified
14	Changing infill pattern	Internal geometry	Replaced Line pattern with Triangular pattern	Detected on the 1^{st} infill layer	Detected [14]
15	Changing infill pattern	Internal geometry	Replaced Triangular pattern with Grid pattern	Detected on the 1^{st} infill layer	Detected [14]
16	Changing layer height at slicing stage	Z-profile	Modified layer height from 0.2 mm to 0.1 mm	Detected on the 2^{nd} layer	Detected
17	Changing layer height for single layer	Z-profile	Modified height for 3 internal layers by 0.1mm	Detected on the 1^{st} attacked layer	0.1 mm [75]; if over multiple layers [19]
18	Changing layer height for continuous layers	Z-profile	Modified consecutive layers height by 0.04 mm	Detected on the 2^{nd} layer as ΔZ ≥ 0.05 mm	0.1 mm [75]; if over multiple layers [19]
19	Changing layer height for partial layer	Z-profile	Modified layer height at a critical location by 0.06 mm	Detected on the 1^{st} layer if ΔZ ≥ 0.05 mm	Not specified
20	Manipulating nozzle thermal profile	Nozzle thermodynamics	Modified nozzle temperature by 5^oC at object center	Detected ifΔT ≥ 4^oC on the 1^{st} layer	Not detected
21	Toggling hotend fan state	Nozzle thermodynamics	Switched OFF the hotend fan through firmware attack	Detected on the first complete audio sample after change	Not detected
22	Changing part cooling fan state	Nozzle thermodynamics	Swtiching OFF the fan for 30 seconds	Detected on the first complete audio sample after change	Not detected
23	Manipulating part cooling fan speed	Nozzle thermodynamics	Modified Fan speed from 100% to 66% and 33%	Detected on the first complete audio sample after change	Not detected
24	Manipulating printing bed thermal profile	Bed thermodynamics	Reduced bed temperature by 10^oC for 1^{st} 2 layers	Detected if ΔT ≥ 3^oC (alert after 35 sec)	Not detected
25	Object warping via bed thermal profile	Bed thermodynamics	Reduced bed temperature by 5^oC for 3rd layer	Detected if Delta T ≥4^oC	Not detected
26	Anisotropy attack	Timing profile	Re-sliced after changing object orientation	Detected on the first layer	Detected
27	Modifying printing speed	Timing profile	Increased the printing speed for 10 G-codes	Detected if and when cum. Δ t ≥ 2 seconds	Δv ≥ ±25mm/s [75], 0.8 sec [20]
28	Modifying commands sequence	Timing profile	Switched a small printing seq. from clockwise to CCW	Detected if length ≥ 1mm	Δt ≥ 2.26 sec [75]
29	G-code moves insertion or deletion	Timing profile	Added 1 G-code 2mm long without extrusion	Detected if the length ≥1mm OR Δt ≥ 2 secs	1 sec duration (translated to ¿10 mm) [20]
30	G-code moves insertion or deletion	Timing profile	Added 2 G-codes 2mm each without extrusion	Detected till the length ≥1mm OR Δt ≥ 2 secs	Not specified; assuming as above
31	G-code moves insertion or deletion	Timing profile	Added multiple move cmds < 1mm each without extrusion	Detected if cumulative Δt ≥ 2 secs	Not specified
32	Regional density variation via filament state	Filament-kinetics	Switched OF the filament motor for 1 G-code	Detected if command length ≥ 1 mm	Not detected
33	Regional density variation via filament speed	Filament-kinetics	Reduced filament length by 3% for 20% Gcode cmds	Detected if ΔE per cmd ≥5% & length ≥1mm OR ΔE per layer ≥1%	Not detected

Table 25.: PrintSafe sabotage attack detection performance viz-a-viz other methods

cross-layer attacks and can detect these low-magnitude attacks on the coming layers if they cross the volume mismatch threshold.

5.6 Conclusion

This study presents PrintSafe, a modular framework designed to detect attacks on the integrity of fused filament fabrication (FFF) printing. PrintSafe is capable of detecting cyberattacks that target any sub-processes to sabotage the printed part. It achieves this using readily available sensors and multi-domain analysis in time, frequency, and space domains, resulting in much higher accuracy than existing methods. Importantly, PrintSafe does not require prior learning for each object, making it suitable for Industry 4.0, which emphasizes the production of customized products over bulk production. The PrintSafe framework utilizes a function profiling engine to incorporate intelligent actions (such as automatic printing bed leveling) performed by the printer's firmware, which may not be present in the original design file used as the source of truth. One of the key advantages of PrintSafe is its ability to detect attacks in real-time, as they happen. This saves valuable time and printing resources. PrintSafe generates a synthesized dataset from the G-code file and analyzes each layer as it is printed. Additionally, it performs an accumulative analysis of previous layers to detect cross-layer attacks. The framework provides detailed information about the process status and any detected attacks, giving users a comprehensive view of the printing process. Overall, PrintSafe represents a significant improvement in FFF integrity checking and has the potential to enhance the security and quality of 3D printing processes.

CHAPTER 6

FROMEPP - FORENSIC READINESS FRAMEWORK FOR FUSED FILAMENT FABRICATION BASED 3D PRINTING

Additive manufacturing (a.k.a., 3D printing) materializes an object by stacking thin layers of material from ground zero. It is increasingly utilized in industry to print critical components of automobiles, airplanes, etc. Failure of a 3D-printed part (such as a turbine blade) during operation may incur immense damage to the system and the surroundings, incentivizing cyberattacks on the printed object. A forensically ready printing setup facilitates a post-incident investigation. Currently, no forensic readiness model exists for an additive manufacturing (AM) process in the literature, whereas conventional cyber-domain-specific models do not consider AM processes and may be ineffective in investigating 3D-printed parts at a crime scene. This chapter presents a forensic readiness framework, `FRoMEPP` *for the material extrusion-based 3D printing process to acquire and preserve forensic data after identifying important information sources in the printing process chain.* `FRoMEPP` *framework provides practical technical guidance to organizations striving for a forensically ready printing environment. It also benefits the regulatory bodies in formalizing compliance criteria for critical 3D printing setups. We implement* `FRoMEPP` *framework on a typical material-extrusion printer, Ultimaker-3, and evaluate it through a case study by implementing three sabotage attacks involving thermal profile manipulation, internal voids, and printing timing integrity compromise. The evaluation results show that* `FRoMEPP` *can effectively investigate and present traces of the attacks against 3D-printed parts.* [1]

[1] *This chapter is based on my paper presented at DFRWS-EU Conference 2023 [69]*

6.1 Introduction

Digital forensic readiness (DFR) [82] prepares an organization for a potential forensic investigation through a well-planned timely acquisition of information that may not remain available after an attack or incident [83]. Forensic readiness of different computing infrastructures comprises different sets of evidence, information sources, and evidence retrieval methods. Any major leap in technology mandates a reassessment of the applicability and effectiveness of existing forensic methods. For example, the National Institute of Standards and Technology (NIST) identified 65 challenges in applying conventional techniques to cloud forensics [7], resulting in the development of dedicated forensic readiness models for cloud environment [84, 85]. For the same reason, dedicated research is required to explore suitable forensic models for additive manufacturing (AM) processes.

With the emergence of Industry 4.0, the rising AM market has gained further significance [5]. More functional components are now 3D-printed, raising the incentive for attackers to attack the printed objects. The research community is actively working on exploring new vulnerabilities and defense techniques, but there is no published work about making an AM process forensically ready. This project is an attempt to fill this gap.

AM or 3D printing is a manufacturing method that materializes an object from ground zero by adding material, typically by stacking thin layers. Figure 60 presents a 3D-printed object life cycle. To print an object, its computer-aided design (CAD) file is converted into an outer surface geometry representation, typically in stereolithography (STL) format. Utilizing the STL file and the user-defined design parameters, a slicer software produces a printer-specific sequence of instructions (G-code). G-code commands are sent to the printer where the firmware sequentially executes them to print the object. Around this core process, there are supporting processes like procurement and job provisioning.

A 3D-printed object can be attacked at various stages of its lifecycle (refer to Figure 60), including attacks on provisioning service (e.g., the computer network of a printing service provider) or during procurement and logistics [86, 87, 38]. This chapter targets the forensics of *manufacturing process chain* consisting of designing, slicing, printing, and quality control. Note that the manufacturing process covers the cyber-physical boundary and is more vulnerable to cyberattacks on printed objects than the warehousing or operation phase. A sabotage attack on a 3D-printed part aims at modifying its physical properties, such as the fit & form, and strength [59]. Recently, researchers have demonstrated that planned tiny deviations within a printer's specification tolerances can still degrade the mechanical strength of the printed parts [33]. Low-magnitude attacks are more likely to pass the non-destructive quality checks. If an attacked part fails during operation, the investigator will have to identify the intruder and prove that the intruder's actions have caused the defect leading to the accident. Therefore, a DFR solution for AM is not complete without acquiring physical-domain information to correlate between a cyberattack and

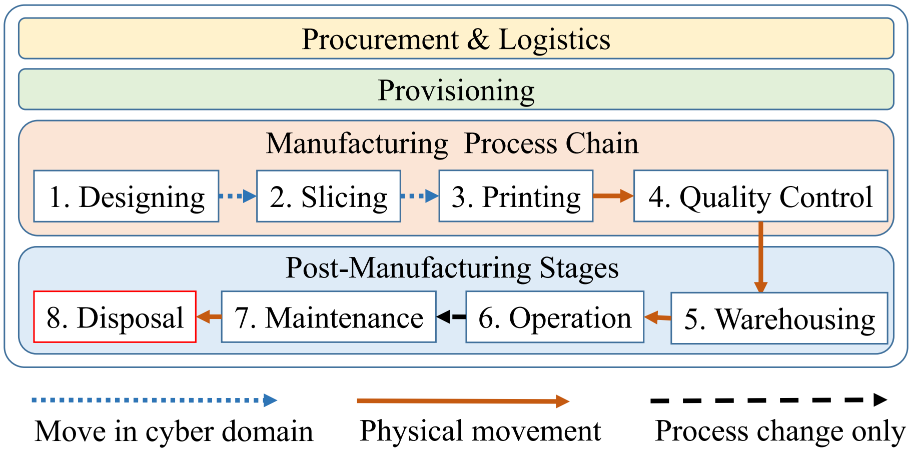


Fig. 60: 3D-printed object life cycle

manipulations in a printing environment and 3D object.

This work also explains an implementation and evaluation of FRoMEPP on a real-world 3D printer against sabotage attacks, and further presents post-incident log analysis to demonstrate FRoMEPP effectiveness; the analysis involves extracting forensically useful artifacts from FRoMEPP dataset and presenting them in comprehensible format for forensic investigation.

There are three main contributions of this work, which are as follows.

- A novel DFR framework, FRoMEPP for material extrusion-based 3D printing
- FRoMEPP's implementation on Ultimaker-3, a real-world material extrusion printer
- Case studies of forensic investigation of sabotage attacks using FRoMEPP

The rest of the chapter is organized as follows. Section 6.2 discusses the related work, followed by the proposed framework in Section 6.3. FRoMEPP implementation on Ultimaker 3 is presented in Section 6.4. Section 6.5 demonstrates FRoMEPP capabilities in conducting post-incident forensic analysis. Section 6.6 presents a case study to evaluate FRoMEPP against the known attacks, followed by future work and the conclusion.

6.2 Related Work

To the best of our knowledge, there is no work in the literature discussing the DFR model for an AM process. This section briefly mentions the relevant work on IT forensic readiness, AM forensics, and cyber-physical systems (CPS) forensics.

Rowlington proposes a ten-step process for organizations to achieve DFR, and presents its advantages for organizations and law enforcement agencies [82]. Grobler *et al.* split DF into proactive, live, and post-incident phases and discuss various dimensions of proactive forensics to present an overall picture of a forensic readiness framework [88]. Valjarevic *et al.* organize DFI into four process groups, where the first group comprises forensic readiness processes. Moving down from high-level abstraction, they split the readiness group into three chronological sub-groups: planning, implementation, and assessment processes [89]. Using a different approach, Elyas *et al.* propose a DFR framework to produce evidence that fulfills regulatory compliance, legal proceedings, and the organization's internal investigation goals [90].

Although researchers have proposed the use of 3D printing to facilitate forensic investigations in various domains [91, 92, 93], the forensics of AM process itself is minimally explored. Forensically sound data acquisition for embedded devices is a challenging task due to the lack of standardized methods and tools [94, 95]. Garcia *et al.* present an experiment to extract forensic information from the PC running the printer control software. Using standard forensic software, they analyze the changes in files and registry entries after executing a printing task [96].

As a 3D printer is a CPS, we briefly cover the forensics of CPS relevant to our work. Within different types of CPS, industrial control systems (ICS) [97] have been actively researched with a focus on forensically analyzing network traffic and device data [98, 99, 100, 101, 102, 103, 104, 105]. These approaches rely on extracting information from cyber domain and do not involve independent measurement of physical process parameters. Ab Rahman *et al.* present a framework based on *forensics by design* approach for the cyber-physical cloud system, emphasizing the importance of incorporating forensic requirements in the design phase [106]. Extracting artifacts from a CPS has been done in the past (but not for 3D printers). Rais *et al.* propose a hardware-based approach to reliably extract the memory contents of ICS devices [49] .

Most of the above-discussed papers only explore information sources in the cyber domain. Interestingly, AM security researchers have used physical domain knowledge in exploring new attacks and defense techniques. Examples of monitoring physical-domain parameters include capturing filament heat signatures through thermal cameras, nozzle temperature monitoring, extruder movement tracking through stepper motors' acoustic signals, electric current, and accelerometers [107, 25, 12, 19, 21]. Although these studies do not discuss AM forensics, they confirm the feasibility of gathering the physical-domain information.

6.3 FRoMEPP Forensic Readiness Framework

6.3.1 Material-extrusion Based 3D Printing Process

Material-extrusion, commonly known as fused filament fabrication, is the most widely used additive manufacturing method [34]. In a typical material-extrusion printer, the continuous filament is pushed through a heated nozzle onto the printing bed. As the nozzle extrudes the filament, it follows a planned path to deposit a thin layer of material on the printing bed. Once one layer is printed, the relative distance between the nozzle and the printing bed is increased to create space for the next layer.

Material extrusion is a complex process to mathematically model due to its dependency on a number of factors such as printing sequence, layer thickness, printing orientation, infill pattern, solidification of the extruded material, etc. For instance, as the hot molten filament is extruded, its heat energy creates a bonding with the existing material on the bed before getting solidified. The bonding process depends on a host of factors, including the extrusion rate, the shape and size of the nozzle, the extrusion pattern, the nozzle, and the printing bed heating profile, and the cooling fans speed. The process complexity offers opportunities to the attackers who are finding higher incentives in attacking 3D printing process.

6.3.2 Attack Model

We assume an advanced attacker that has access to expert-level knowledge of material-extrusion-based AM process to design inconspicuous and sophisticated attacks. The attacker can compromise the 3D printing process by either exploiting the cyber domain components of the process chain or by installing malicious firmware through a USB drive or an SD card. The purpose of these tough yet realistic assumptions is to design a forensic framework that caters for sophisticated attacks.

This work focuses on active attacks and assumes that the attacker will sabotage the primary printing process at any stage. Passive attacks such as intellectual property theft via side-channel monitoring [108] are not in scope.

6.3.3 DFR requirements for material extrusion-based AM environments

Before creating the DFR framework, it is imperative to understand its objectives. We analyze the complete AM process chain to identify the potential indicators of compromise and formulate the below-mentioned set of requirements that an effective AM-specific DFR solution should address. The literature review confirms that these requirements engulf all the existing attacks.

1. Monitor the printing process in both the cyber and the physical domains

2. Acquire operating system and application-level forensic information from all involved cyber-domain actors.

3. Acquire the object-specific cyber-domain artifacts (such as CAD / STL / G-codes)

4. Acquire the inter-stage network traffic captures

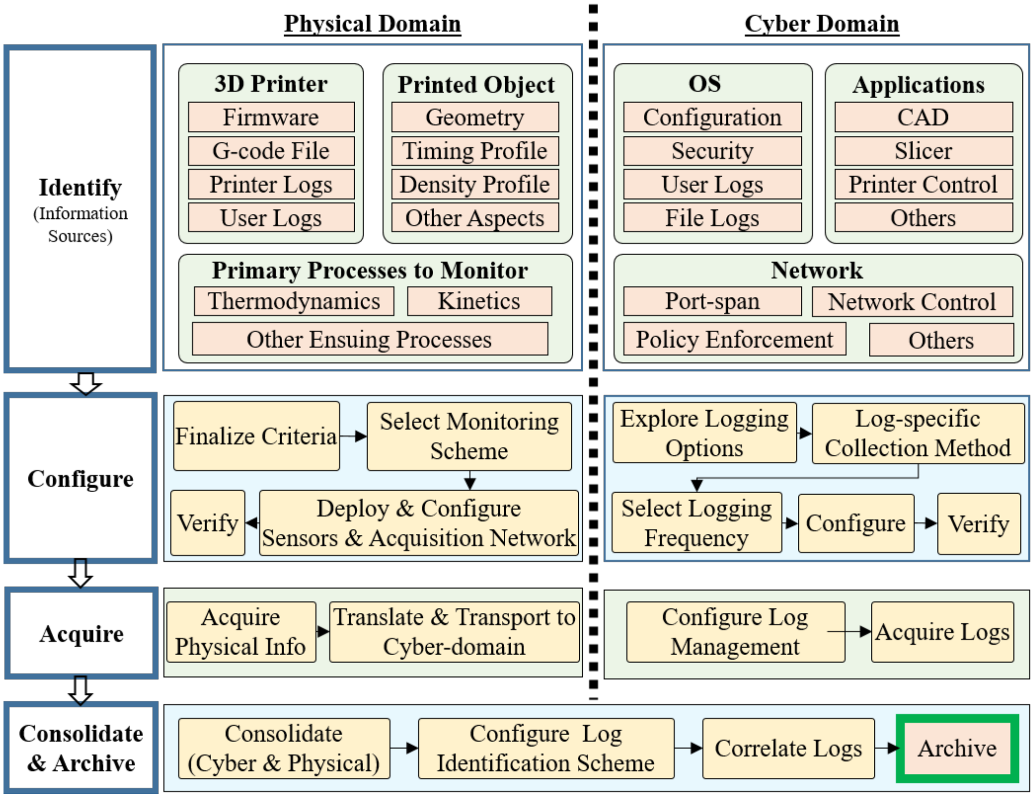


Fig. 61: Forensic readiness framework, FRoMEPP for material extrusion based additive manufacturing (3D printing) environment

5. Capture the printer's view of the process by extracting forensically important logs from the printer

6. Independently monitor the printing operation with no or minimal (within operational tolerances) intrusion

7. Correlate a printed object to its corresponding logs

8. Offer capability to analyze the process on intuitive boundaries, such as per layer or per instruction basis

9. Facilitate an interruption-free printing operation during any post-incident forensic investigation

10. Preserve the dataset in accordance with standard forensic soundness guidelines

6.3.4 FRoMEPP - proposed DFR framework for AM process

This section presents an overview of the proposed framework outlined in Figure 61. The left column represents generic DFR tasks at a higher abstraction level, and the right side details the information and activities to accomplish the task for AM DFR.

The first step in creating a forensically ready printing setup is to identify the useful data sources in the process chain. Cyber-domain data sources are classified as OS, Network, and Applications. As cyber-domain logs alone do not provide the required details to authentically answer all the forensic questions, critical AM-specific physical processes are identified and monitored. From the forensic perspective, we categorize the physical processes as the primary (directly influenced through cyber manipulations) and the secondary processes. FRoMEPP framework suggests monitoring all independent primary processes. The second step is to establish monitoring criteria and a compliant acquisition scheme for data retrieval. To ensure forensic soundness, out-of-band sensors are deployed in the physical domain. For the cyber domain, network data acts as an independent source to scrutinize the OS and the applications logs.

FRoMEPP assigns a unique identifier to every printed object. The physical and the cyber logs are also assigned identification tags to correlate with the printed object logs. The information collected from various cyber and physical sources is standardized and converted to comprehensible formats for analysis. The consolidated and correlated data sets from both domains along with the metadata are preserved and archived.

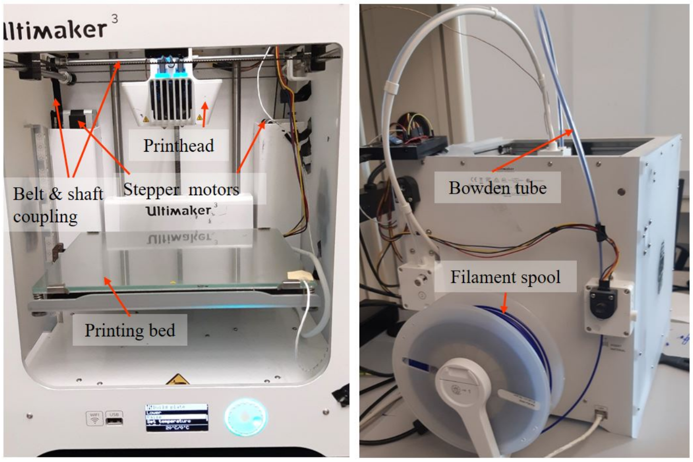


Fig. 62: Ultimaker-3 material extrusion-based 3D printer

6.4 Framework Implementation and Illustration on a Real-world 3D Printer

This section elaborates the `FRoMEPP` framework through an implementation study on a material extrusion-based 3D printer - Ultimaker-3. Presented in Figure 62, Ultimaker-3 is a dual-nozzle cartesian coordinates 3D printer comprising one stepper motor for each x/y/z/filament axes. The printer is controlled through an open-source software - Cura, that receives an STL file and converts it into Ultimaker-compatible G-code commands. The control PC hosting Cura is connected to the printer over LAN. We use Comsol Multiphysics 5.4 and AutoCAD 2019 tools to create design files.

6.4.1 Identify the information of interest

We examine AM elements in both cyber and physical domains as candidates for information sources.

6.4.1.1 Cyber-domain information

Information sources in the cyber domain are classified under three main categories; operating systems, computer networks, and relevant applications.

Operating system logs An attack may use the OS of a cyber-domain device as a launching pad, leaving important traces of unlawful activities. Information about user sessions, file activities, jump lists, and OS-level security event logs help understand the attack mechanism [109, 110]. Interested readers may refer to [96] for the OS level forensic traces in AM environment.

Network logs For OS and application layer attacks, the network logs present independent and forensically sound evidence. Network traffic between the printer and the external world includes Cura communication over HTTP and user connections over SSH protocol or HTTP. This implementation uses Wireshark software to capture Control PC Ethernet traffic.

Application logs Generally, OS logs offer details about the attacker and the attack path but less information about the attack. If the traffic between two nodes is encrypted, information extraction from their network traffic becomes more challenging [111]. On the other hand, application logs often offer detailed and comprehensible information about the attacker's manipulations. Main applications in the 3D printing process include CAD, slicer, and printer control applications. Printing service providers have other supporting software such as provisioning and billing application.

In this implementation, we utilize the auto-save, backup files, and log file generation options in AutoCAD software to attain helpful traces of anomalous or unauthorized activity. For slicing and printer-control functions, we use Cura version 4.10.0

software. Cura generates informational and error log files saved in temporary folders. These files reveal important information, including configuration changes, error logs, and the recent printed files list.

6.4.1.2 Physical-domain information

Although the intrusion traces may be discovered in cyber logs, they cannot offer conclusive evidence to pin the responsibility of the component's failure on the attacker. Physical-domain data extraction fills the missing link. Being later in the process chain and not in control of the attacker in most cyberattacks, physical-domain data about the printing state is more reliable than cyber-domain logs.

Printer logs: A 3D printer is an advanced embedded system available in various hardware architectures and firmware. Basic printers use a single controller for printing and user interface. Ultimaker-3 uses A20-OLinuXino-LIME2 as the printer's mainboard, running a custom Linux OS based on the Debian Jessie release. The real-time kinetic and thermodynamic functions are offloaded to a separate controller with dedicated firmware. An essential aspect of the printer's security is firmware integrity. This implementation periodically extracts the running firmware, list of recently printed files, event messages, and user login details from the printer.

Printed object logs: The outcome of the 3D printing process is the printed object. Being the most common target of attack [59], the state of the printed object is measured throughout the printing. Monitoring the object's state is more than mere visual inspection. Some attacks (such as thermodynamic attacks) cause no visual deformation but still damage the printed object [8]. Material extrusion is a complex process involving kinetics, thermodynamics, crystallization [3], glass transition [4],

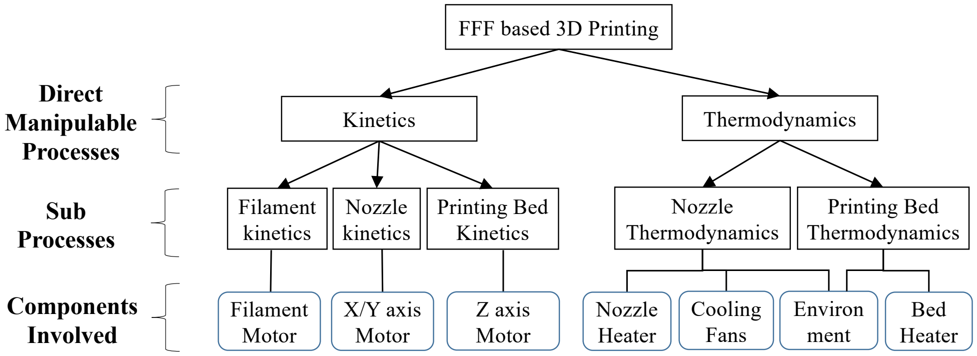


Fig. 63: Direct-manipulable processes and controlling components

microstructure-related and other properties. `FRoMEPP` recommends monitoring the primary or direct-manipulable processes.

Direct-manipulable processes in material extrusion-based printing: In material extrusion, kinetics and thermodynamics are the processes that a cyber-attacker can exploit. As all the remaining processes are their subsequent effects, monitoring kinetics and thermodynamics cover cyber-manipulations targeted towards other properties. Figure 63 represents the direct-manipulable processes of material extrusion-based printing, along with the components controlling them. The three sub-processes of kinetics, i.e., filament kinetics, nozzle kinetics, and printing-bed kinetics, are controlled through filament motor, x,y, or ρ, θ axes motors, and z-axis motor, respectively. The thermodynamic process is influenced by the nozzle heater, printing bed heater, and cooling fans. Environmental temperature and airflow may also slightly impact the thermodynamic process. Monitoring the instantaneous state of the components mentioned in Figure 63 during printing is analogous to monitoring the object being printed, and thus adequate in investigating attacks on the object.

6.4.2 Configure the information sources for logging

After identifying the information sources, we configure them for log retrieval. Being extensively researched, the cyber-domain configuration is briefly discussed here. We only focus on AM-specific applications and network logs. We collect the network traffic directly from the Ethernet interface card of the control PC through Wireshark version 3.6.2. Traffic capturing is configured as a permanent process, and a new file is created on an hourly basis.

Cura logs exhibit multiple retention patterns; such as cyclic (most recent retained or periodic erasure). We configure the logging on a *pull* basis at a frequency in accordance with the retention period and the log's buffer size. By default, Cura saves an error log file at '*C:/users/username/AppData/Roaming/Cura/*' with the name '*stderr.log*'. Another important file named '*cura.log*' placed at '*C:/users/username/ AppData/Roaming/Cura/version/*' contains Cura configuration parameters values, ten most recent files, and other active configuration details.

From a logging perspective, the printer and the printed object are constituents of the physical domain. Logging configuration for the printer depends upon the provisioning options provided by the vendor. Ultimaker-3 offers Secure Shell (SSH) access to the printer for configuration and code changes. A user can add a suitable code to push the desired logs, such as important event details.

Measuring the printed object state for forensics is a less researched topic. Therefore, we discuss the four-step 'configure' task presented in Figure 61 in detail.

6.4.2.1 Finalize the monitoring criteria

Numerous methods are available to observe the printed object's state during printing. To evaluate the performance and suitability of a monitoring scheme for a printing setup, FRoMEPP recommends the following five points criteria.

Sensing system resolution and feasible parameters The choice of the printing state sensing scheme depends on the required resolution for each monitored component. We set an expected resolution of 0.1 mm for the printhead, 0.05 mm for the printing bed, and $1^o C$ resolution for the thermal components. These values are derived in consideration of the printer specifications.

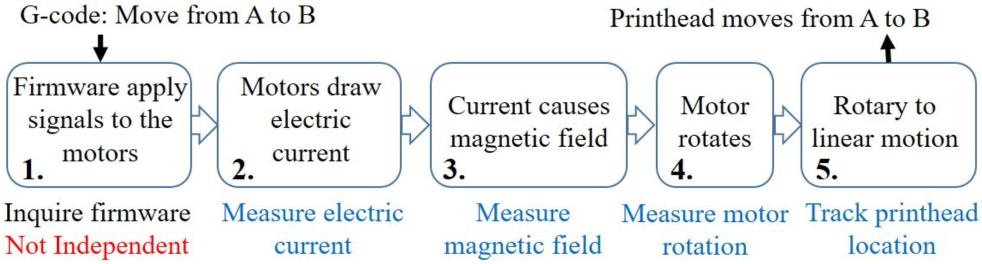


Fig. 64: Kinetics process stages with kinetic measurement options

Observing the end effect AM process transitions through multiple stages before accomplishing a printed object. If the actual printing is at Stage$_n$ and the observation point is at Stage$_i$ where $i < n$, the manipulations at Stage$_{i+j}$ where $0 < j < n - i$, will go unobserved. Thus, it is natural to move the observation point as close to the end effect as feasible.

Figure 64 elaborates this concept using Ultimaker-3 kinetic process example. When the printer executes a move instruction, the firmware calculates the per-axis distance, converts it to electrical signals, and applies them to the stepper motors. The motors transform electrical energy into magnetic energy and rotate the driving shaft. Mechanical coupling translates shaft rotation to the printhead movement. Stage 2 to Stage 5 of Figure 64 offer independent provisions to measure the printhead state. Stage 5 is the most appropriate choice, if feasible, as it rules out intermediate-stages errors.

Noise resilience Noise negatively impacts the accuracy and the resolution of a monitoring scheme and may increase its algorithmic complexity and processing overhead. Some monitoring approaches are more resilient to specific environmental noises than others. For instance, unlike the electric current, magnetic field measurement suffers from the interference of other motors' magnetic fields or external sources. For this implementation, we set a criterion that the selected scheme should be resilient to magnetic interference, routine sounds in the lab, and minor vibrations in the platform (printer table).

Non-intrusiveness and simplicity of deployment Intrusiveness is evaluated from deployment and operational perspectives. Measuring the electric current does not interfere with its operation, but the deployment involves partial disassembly and re-wiring. On the other hand, measuring the printed object state through a camera (by iteratively pausing the printing) offers non-intrusive deployment. However, the 'pause' operation modifies the timing and thermodynamic profile. `FRoMEPP` recommends process monitoring to be operationally non-intrusive.

Independent monitoring Forensic soundness of the acquired information is vital for any DFR solution. Information acquired from an actor under attack loses its evidentiary weight as a sophisticated attacker can modify its generated logs [112]. `FRoMEPP` recommends out-of-band sensors to measure the process state.

6.4.2.2 Select a suitable monitoring scheme

After formalizing the criteria, we evaluate seven schemes for monitoring the kinetic processes, including in-band firmware query, accelerometers, magnetometers, optical encoders, acoustic sensors, camera imaging, and measuring the electric current drawn by kinetic components. Each approach has its merits and limitations. Features are split into mandatory and non-mandatory categories. The overall score of a scheme is calculated using Equation 6.1, where S_i is the score of i^{th} monitoring scheme, r_{ik} is the binary result of k^{th} mandatory feature, w_{ij} and s_{ij} are the respective weight and the score of a j^{th} feature for the i^{th} scheme. To ensure forensic soundness, we elevate two features as mandatory; monitoring should be (1) independent and (2) operationally non-intrusive. The remaining features are assigned an equal weight (for simplicity).

$$S_i = (\prod_{k=1}^{m} r_{ik} \quad * \quad \sum_{j=1}^{n} w_{ij} * s_{ij}) \tag{6.1}$$

Two approaches are rejected for not fulfilling mandatory requirements; (1) 'Inquiring the firmware' for being in-band and thus not forensically sound, and (2) 'camera imaging (as available to us)' for being operationally intrusive. 'Resolution' and 'Noise resilience' features of the schemes are assessed in view of the results in the existing literature. Optical encoders and electric current measurement schemes get the highest points. Camera imaging tops the 'Observing the end effect' feature, as it rules out all possible machine issues. Overall, the optical encoder-based sensing scheme gets the highest points in our scenario and is selected for the kinetic processes monitoring, followed by the electric current sensing, accelerometers, acoustic sensors, and magnetometers.

Although the scoring is applicable to generic material-extrusion-based setups, we recommend a re-scoring for each unique criterion and printing environment. For the thermodynamics process monitoring as per Figure 63, we restrict the scope to the nozzle and printing platform thermal profile measurement using a thermocouple and a thermistor, respectively.

6.4.2.3 Deploy and configure acquisition system

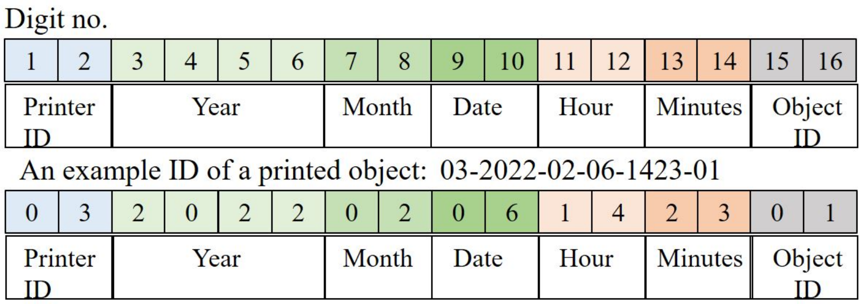
Rotary optical encoders are deployed on the printhead connecting shafts for the printhead kinetics. A linear encoder is installed to track the printing bed. A k-type thermocouple is deployed at the tip of the heated nozzle, and a surface-mount thermistor is annexed to a corner of the heated platform. An Arduino board energizes the sensors and collects the data. Interrupt routines are used for the high-velocity kinetic data, and the slow-varying thermodynamic data is polled periodically. The data is further sent to the project PC over a USB interface. Sensors specifications and installation procedure is detailed here [31].

6.4.2.4 Evaluate the forensic soundness of the system

Unlike the established practices in the cyber domain, the forensic soundness of the proposed physical-domain monitoring methods needs to be ascertained. Reference data, physical measurements, and in-system readings (where possible) are used to verify monitoring scheme data. Data acquired from the printer uses standard SSH connection, or REST APIs over HTTP. The biggest artifact in size is the firmware that matches exactly with the copy securely attained from the vendor. The printed object's logs are verified through test cases covering the operational spectrum of the unit under test. For instance, rotary encoders data is verified in small steps over one complete rotation to rule out deployment errors such as axial play and runout.

6.4.3 Acquire the logging data

After configuring the physical and the cyber-domain data extraction schemes, data collection is started. For a large-scale setup, a standard logging management suite may be used to handle the variety of logs. For this demonstration, we develop a small piece of software to manage and preserve the logs. While the cyber-domain and the printer logs follow a standard or a proprietary format, no data format exists for the external sensors network. We develop two structures for the printed object data. To avoid overloading the Arduino board, we use a small data structure for the high-velocity kinetic data, and a bigger structure for the consolidated data from the entire sensors network.



Digit no.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Printer ID		Year				Month		Date		Hour		Minutes		Object ID	

An example ID of a printed object: 03-2022-02-06-1423-01

0	3	2	0	2	2	0	2	0	6	1	4	2	3	0	1
Printer ID		Year				Month		Date		Hour		Minutes		Object ID	

Fig. 65: 16-digit unique identifier for printed object and its logs

6.4.4 Consolidate and archive

DFR software receives the physical-domain data, pre-processes it, and utilizes interpolation functions to fill in the missing data fields to standardize the data set. At this point, the physical and the cyber domain data is available as a cyber domain resource. Consolidation of logs from both domains offers operational ease in the post-incident investigation. Each logging category has a different frequency, ranging from 5 ms for the fast-moving kinetic data to a day for retrieving firmware copy. To correlate among various logs, an identification mechanism is required.

6.4.4.1 Unique identifier for logs correlation

An identification scheme connects each unique printed object to its corresponding logs. Identification schemes can utilize on-the-object or off-the-object marking methodology. Although on-the-object schemes (such as 3D watermarks) are more scalable and error-resistant, their integration with the current process chain is not readily available. From a forensic perspective, an organization may employ any suitable object tagging mechanism. Printing an object (or a batch of objects) is a unique event in the time domain, making timing information a feasible object identifier. A unique ID is assigned to the physical object, and also to all the log categories having a one-to-one relationship with the printed object. ID tag in this study has 6 fields and 16 decimal digits. The first two digits represent the unique printer on the farm. Digits 3 to 14 mark the printing start time. The last two digits are used to disambiguate among multiple objects within a single print job having the same start time. ID 03-2022-02-06-1423-01 presented in Figure 65 refers to the first object printed through printer 3 at 2:23 PM on Feb 6, 2022.

6.4.4.2 Archiving

The data set in our study is archived using a four-tier functional hierarchy: log category, organizational structure, raw logs, and extracted artifacts. Five log categories are defined; printer, printed object, OS, applications, and network logs. The organizational structure of logs is aligned with the identification scheme. To archive a new log, the software traverses the repository tree from the root down to the node hosting that log. Any non-existent node in the path is created during this operation. For example, the software extracts Cura logs on an hourly basis. At 1400 hrs, the software traverses the repository from *Root/Application logs/Cura ID/Date/*. A new directory *1400* is created, and the logs are saved.

6.5 Post-incident Forensic Log Analysis

A malicious intrusion in 3D printing process may be aimed to disrupt the printing service or sabotage the printed object. Inducing obvious defects in an object, such as modifying its shape and size, is a simple sabotage attack. A sophisticated sabotage attack may induce non-obvious defects in the object, so that it may pass the quality assurance check and be installed in a critical system where its premature failure can cause more damage. In this section, we present how FRoMEPP data set helps identify traces of simple and sophisticated attacks including information about the attack mechanism and the attacker. Generally, the raw logs require further analysis to identify useful evidence. Analyzing the physical-domain logs require different tools than conventional IT.

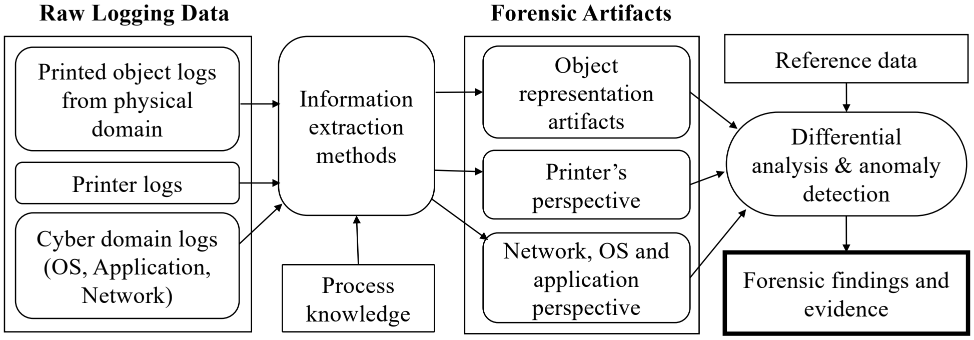
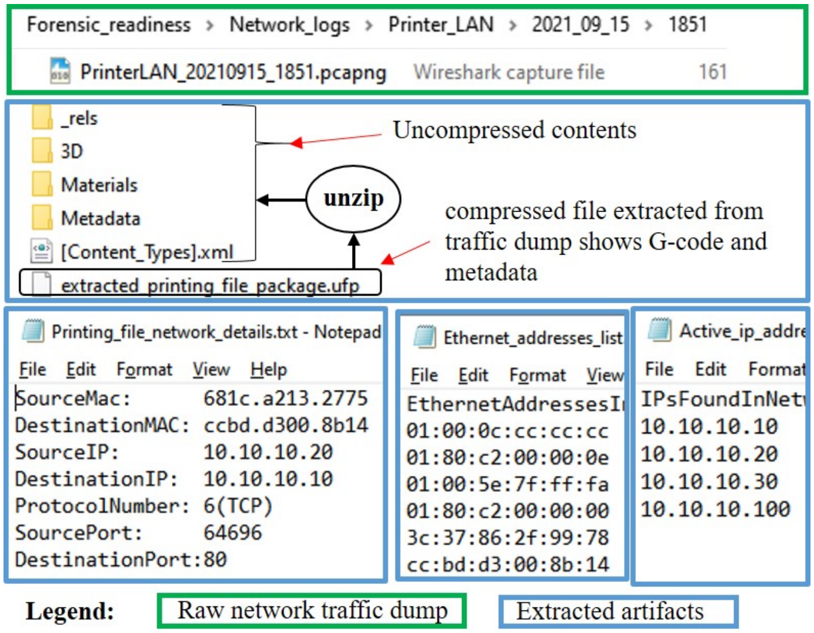

Raw Logging Data
Printed object logs from physical domain
Printer logs
Cyber domain logs (OS, Application, Network)
Information extraction methods
Process knowledge
Forensic Artifacts
Object representation artifacts
Printer's perspective
Network, OS and application perspective
Reference data
Differential analysis & anomaly detection
Forensic findings and evidence

Fig. 66: Forensic information and artifacts extraction process

Figure 66 presents a formalization of the forensic artifacts extraction process used in our implementation. Three categories of logs are applied to the relevant extraction methods that utilize process knowledge and correlation algorithms to extract useful information. These algorithms return important artifacts such as printed object geometry and density profile, thermodynamic profile, timing profile, printer firmware, design files, and session and error logs.



Forensic_readiness › Network_logs › Printer_LAN › 2021_09_15 › 1851
PrinterLAN_20210915_1851.pcapng Wireshark capture file 161
_rels
3D
Materials
Metadata
[Content_Types].xml
extracted printing file package.ufp
unzip
Uncompressed contents
compressed file extracted from traffic dump shows G-code and metadata
Printing_file_network_details.txt - Notepad
File Edit Format View Help
Ethernet_addresses_list
File Edit Format View
Active_ip_addre
File Edit Format
Legend:
Raw network traffic dump
Extracted artifacts

```
SourceMac:       681c.a213.2775
DestinationMAC:  ccbd.d300.8b14
SourceIP:        10.10.10.20
DestinationIP:   10.10.10.10
ProtocolNumber:  6(TCP)
SourcePort:      64696
DestinationPort:80
```

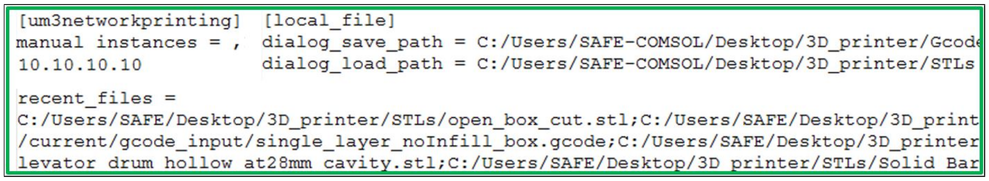
```
EthernetAddressesI
01:00:0c:cc:cc:cc
01:80:c2:00:00:0e
01:00:5e:7f:ff:fa
01:80:c2:00:00:00
3c:37:86:2f:99:78
cc:bd:d3:00:8b:14
```

```
IPsFoundInNet
10.10.10.10
10.10.10.20
10.10.10.30
10.10.10.100
```

Fig. 67: Forensic artifacts extracted from network-traffic dump

6.5.1 Network artifacts

Figure 67 presents a few artifacts extracted from the network traffic. An unencrypted compressed file found in the dump comprises the G-code file and other metadata related to the printing request. We also discover the user machine MAC and IP addresses and TCP port numbers. The active IP and Ethernet addresses shown in the figure can help in identifying suspicious users. Ultimaker-3 also hosts a web server offering unauthenticated view-only access. The network traces will capture all access events.



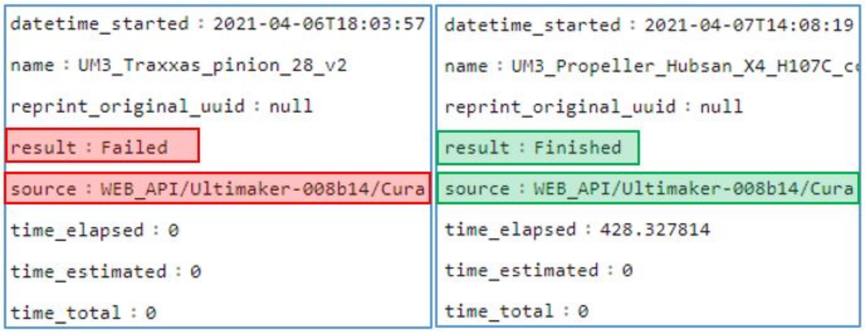
```
[um3networkprinting]  [local_file]
manual instances = ,  dialog_save_path = C:/Users/SAFE-COMSOL/Desktop/3D_printer/Gcode
10.10.10.10           dialog_load_path = C:/Users/SAFE-COMSOL/Desktop/3D_printer/STLs

recent_files =
C:/Users/SAFE/Desktop/3D_printer/STLs/open_box_cut.stl;C:/Users/SAFE/Desktop/3D_print
/current/gcode_input/single_layer_noInfill_box.gcode;C:/Users/SAFE/Desktop/3D_printer
levator drum hollow at28mm cavity.stl;C:/Users/SAFE/Desktop/3D printer/STLs/Solid Bar
```

Fig. 68: cura.cfg excerpt showing IP address, paths, and recent files

6.5.2 Cura artifacts

A few forensically important Cura logs are discussed here. The log file *stderr.log* contains error messages with the timestamp and an assigned severity level. The contents are flushed when Cura restarts. Figure 68 displays a part of the information present in another log file, *cura.cfg*, indicating the printer's IP address, the default file path, and the ten most recent files processed by Cura. A subfolder, *quality_changes*, tracks the changes in the printing configuration profiles.



```
datetime_started : 2021-04-06T18:03:57
name : UM3_Traxxas_pinion_28_v2
reprint_original_uuid : null
result : Failed
source : WEB_API/Ultimaker-008b14/Cura
time_elapsed : 0
time_estimated : 0
time_total : 0
```

```
datetime_started : 2021-04-07T14:08:19
name : UM3_Propeller_Hubsan_X4_H107C_c
reprint_original_uuid : null
result : Finished
source : WEB_API/Ultimaker-008b14/Cura
time_elapsed : 428.327814
time_estimated : 0
time_total : 0
```

Fig. 69: Printer logs showing job name, status, and timestamps

6.5.3 Printer artifacts

Ultimaker-3 offers a set of Representational State Transfer (REST) APIs to control and monitor the printer. We utilize selected APIs to iteratively extract the information of interest. Figure 69 presents a snapshot of the printer logs showing the printing job name, its status, and the important timestamps related to the job. Using SSH connection, we extract the printer firmware and the list of prints located at */usr/share/griffin/griffin/machines/jedi.hex* and */var/spool/cluster/../*, respectively.

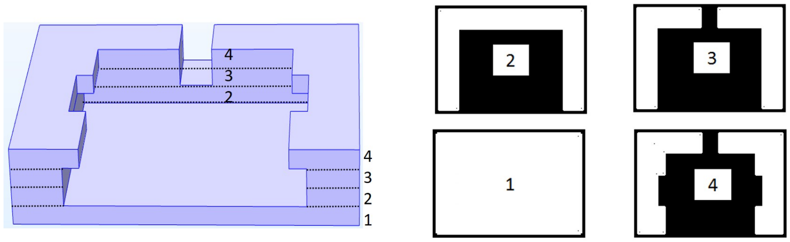


Fig. 70: A four layers model and its 2D slices

6.5.4 Printed object artifacts

A layer-change event naturally splits 3D printing operation, motivating us to analyze the process on a per-layer basis. Through `FRoMEPP` data set, we recreate the geometry of each layer to examine the process in the space domain. As two similar-looking geometries may have been accomplished using different toolpath sequences, we also analyze the printing process in the time domain.

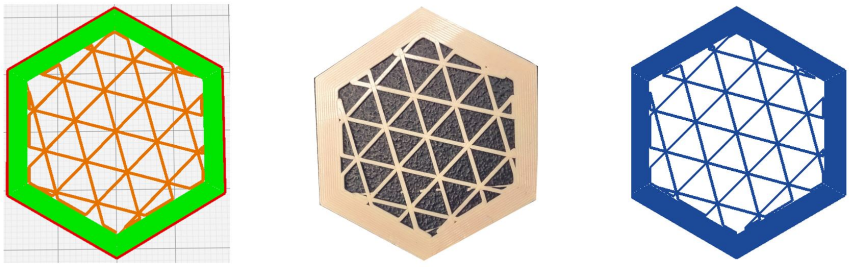


Fig. 71: Slicer representation (left), actual printed object (center), and accurate image recovered from FRoMEPP dataset (right)

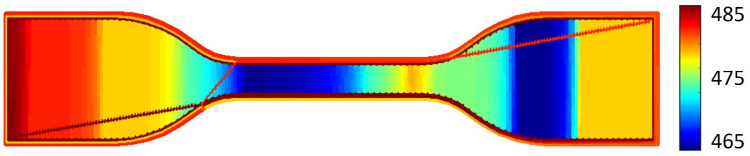


Fig. 72: Thermodynamic profile showing $\pm 10^{o}C$ variation in nozzle temperature at targeted locations

Figure 70 presents the slices of a four layers model as 2D bitmap images. This presentation style is commonly used by slicer software, making the comparison between the actual print and the intended design simplified, detailed, and demonstrable. Although these xy plane slices, shown in Figure 70, align with the printing direction, bitmaps of xz and yz plane cross-sections can also be extracted from the acquired data. Figure 71 represents an accurate recovery of the printed hexagon through `FRoMEPP`.

The thermodynamic profile of the printed object is mainly driven by the filament temperature at the time of extrusion. A cyberattacker can manipulate the nozzle temperature to cause weak bonding or residual thermal stresses. Figure 72 reflects the nozzle temperature at the instant when the pixel received the filament. The heatmap shows a temperature fluctuation of approximately ±10^{o}C.

The timing profile for the entire printing job or a single layer can be presented by plotting individual printing parameter values against time. Users may also employ other intuitive and beneficial techniques to view the process details such as recovering a 3D animation of the printing job from the sensors data set.

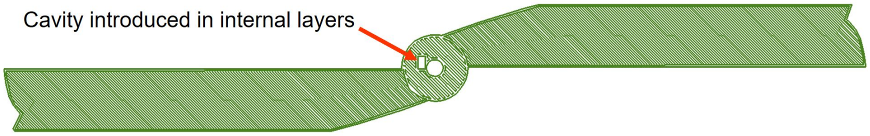


Fig. 73: Space domain representation of an internal layer of drone propeller showing a malicious cavity

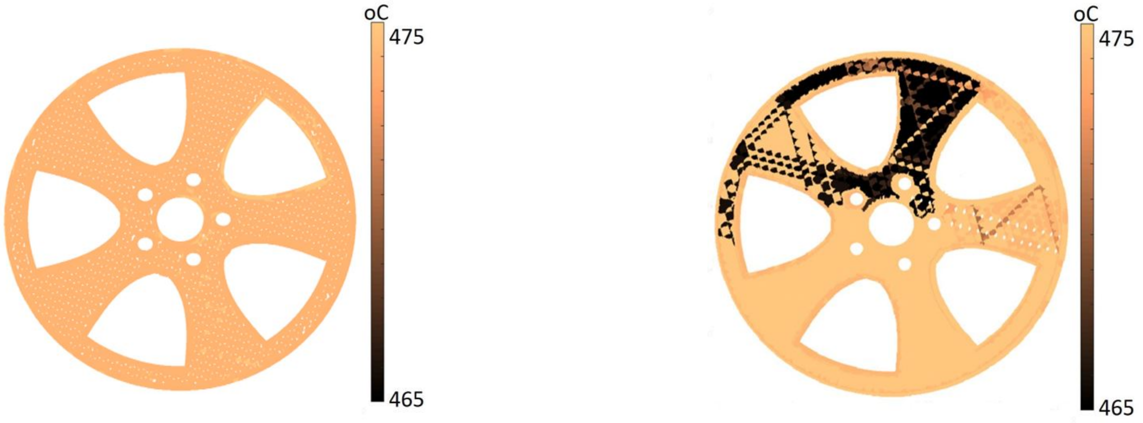
6.6 FRoMEPP Case Study on Sabotage Attacks

Damaging the printed object is a key motive in attacking 3D printing process. This study evaluates `FRoMEPP` on three practical sabotage attacks in the literature [12, 8].

Attack Scenario An important printing facility receives complaints of premature failure of three critical parts (a car wheel, a drone propeller, and a drive shaft) printed a few months ago. The facility owner ordered a forensic investigation. The facility had already implemented `FRoMEPP` framework on Ultimaker 3 as guided in Section 6.4 and 3.6 and provides its access to the forensic investigator. The investigator attains the unique identifiers for the suspected and corresponding known-good prints.

6.6.1 Investigating printed object logs

The suspicion of sabotage of the printed parts encouraged the forensic expert to initiate the investigation with the object logs. The expert analyzes them from three distinct standpoints: space-domain or geometrical analysis, thermodynamic analysis, and time-domain analysis.



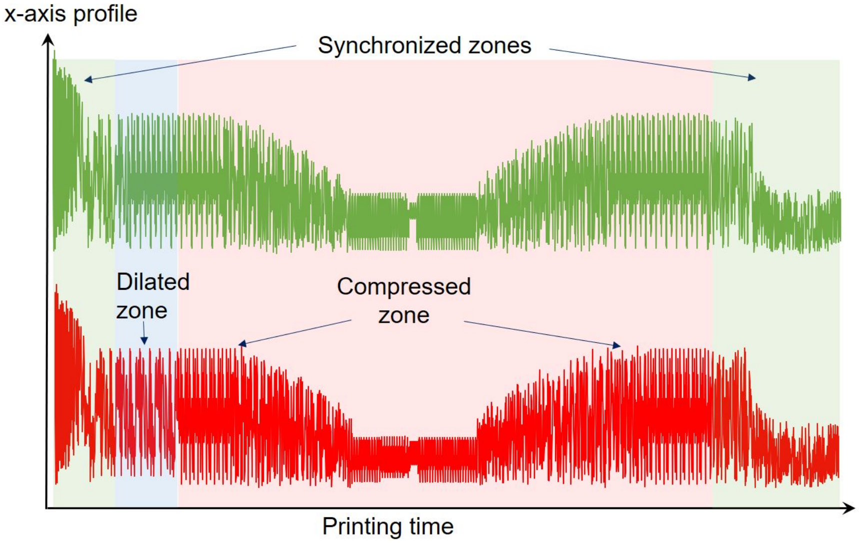
(a) Expected profile with minimal temperature variations

(b) Attacked object shows 10^{o}C reduction at one part

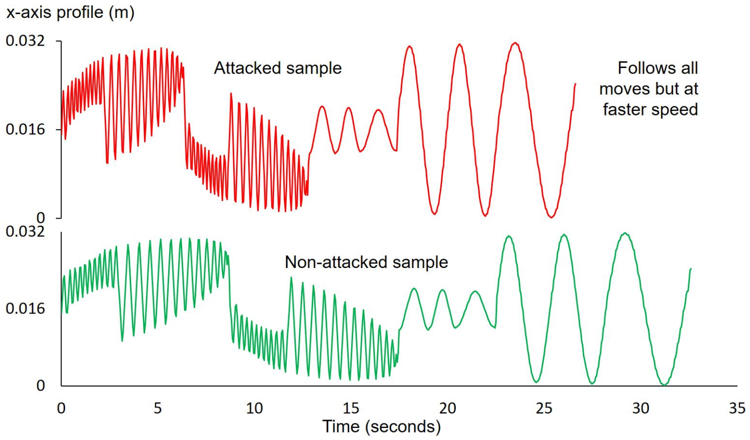
Fig. 74: Recovered heatmaps of attacked and non-attacked objects

Space-domain analysis As presented in Section 3.6, the space-domain or the geometrical information for every layer is archived as a bitmap file, where each pixel of the bitmap represents 0.01 mm^2 area of the layer's geometry. The size and shape of the car wheel and the drive shaft match their non-attacked counterparts, but the drone propeller shows signs of a malicious cavity in the internal layers. Figure 73 presents the bitmap image of the sixteenth layer showing a malicious cavity near the center. The cavity size is measurable by counting the pixels (as bitmaps are accurate and to the scale), and found to be 1 mm x 2 mm in 80% of the internal affected layers.

Thermodynamic profile analysis The thermodynamic profile of the car wheel presented in Figure 74b highlights a suspicious pattern. One of the spokes is printed at a different temperature than the rest of the wheel. Although a few degrees of thermal variation is expected during printing, a 10^oC reduction in temperature at a specific location is not a random error. The pattern is reinforced in all internal layers. Repeated reduction and reversion at a specific location in selected layers rule out hardware issues with the heating system. Manipulating the thermal profile may induce residual thermal stress causing strength reduction and warping [8]. The original G-code file did not contain temperature modification instructions. The thermodynamic analysis of the other two objects does not reveal any anomaly.



(a) Timing profile of attacked and non-attacked driving shafts



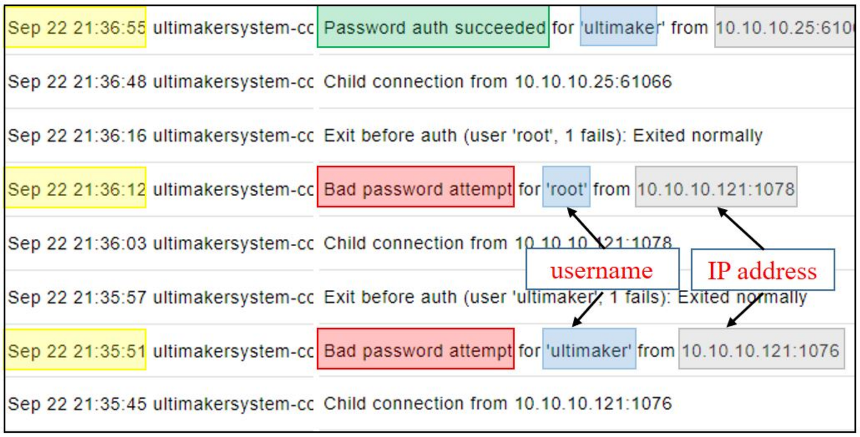
(b) Single layer profile of an attacked and non-attacked object

Fig. 75: Timing profiles of the attacked and non-attacked shafts

Timing profile analysis The space domain and the thermodynamic profile analysis of the driving shaft do not reveal any abnormality. The timing profile also does not offer any hint of malicious action. However, some synchronization issues are observed on examining the timing profile against known-good logs. The x-axis kinetic profile presented in Figure 75a shows that the overall printing time is unchanged but the two profiles are not locally synchronized. To further drill down, the investigator compares per-layer timing profiles and finds that the time taken to print an internal layer is less than the default time. On the contrary, the initial layers were printed slowly. Other axes' kinetic profiles manifest similar patterns. Figure 75b presents a comparison of a single-layer timing profile of suspected and non-attacked objects. This behavior shows that the critical zones were printed at a higher speed to influence the part's strength [113], and the non-critical ones were slowed down to compensate for the time gain. The printed objects logs confirm malicious modifications in all three objects.

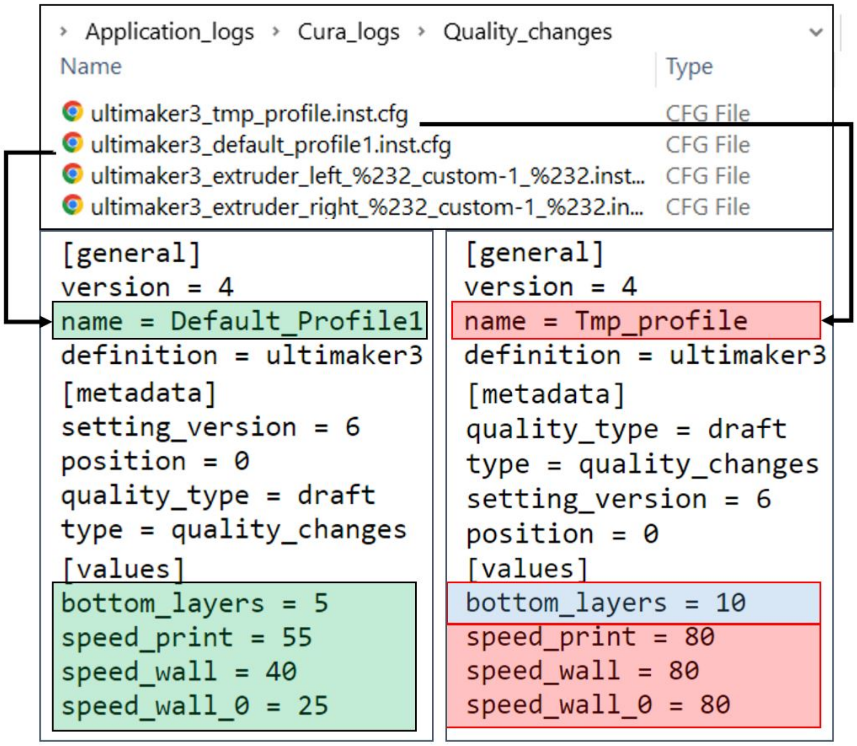
6.6.2 Investigating the printer logs

To identify the attack mechanism and the attacker's details, the expert examines the logs from the printer. A few suspicious login attempts are discovered in the logs. The attempts were made from two source IP addresses using default usernames. As the user updated the default root password, no attempt to the root was successful. However, access to a non-root default user was successful. Figure 76 captures a few of the login attempts. As a non-root user does not provide adequate process modification privileges, the suspected attacker did not pursue the attack through the printer. One of the IP addresses is linked to a printer control computing machine. The other IP address is assigned to the computer of an employee who is not authorized to connect to the printing system.



```
Sep 22 21:36:55 ultimakersystem-cc Password auth succeeded for 'ultimaker' from 10.10.10.25:610
Sep 22 21:36:48 ultimakersystem-cc Child connection from 10.10.10.25:61066
Sep 22 21:36:16 ultimakersystem-cc Exit before auth (user 'root', 1 fails): Exited normally
Sep 22 21:36:12 ultimakersystem-cc Bad password attempt for 'root' from 10.10.10.121:1078
Sep 22 21:36:03 ultimakersystem-cc Child connection from 10.10.10.121:1078
Sep 22 21:35:57 ultimakersystem-cc Exit before auth (user 'ultimaker', 1 fails): Exited normally
Sep 22 21:35:51 ultimakersystem-cc Bad password attempt for 'ultimaker' from 10.10.10.121:1076
Sep 22 21:35:45 ultimakersystem-cc Child connection from 10.10.10.121:1076
```

Fig. 76: Login attempts status logs extracted through REST API



```
ultimaker3_tmp_profile.inst.cfg                          CFG File
ultimaker3_default_profile1.inst.cfg                     CFG File
ultimaker3_extruder_left_%232_custom-1_%232.inst...      CFG File
ultimaker3_extruder_right_%232_custom-1_%232.in...       CFG File
```

```
[general]
version = 4
name = Default_Profile1
definition = ultimaker3
[metadata]
setting_version = 6
position = 0
quality_type = draft
type = quality_changes
[values]
bottom_layers = 5
speed_print = 55
speed_wall = 40
speed_wall_0 = 25
```

```
[general]
version = 4
name = Tmp_profile
definition = ultimaker3
[metadata]
quality_type = draft
type = quality_changes
setting_version = 6
position = 0
[values]
bottom_layers = 10
speed_print = 80
speed_wall = 80
speed_wall_0 = 80
```

Fig. 77: Cura logs shows new or modified profiles and parameters

6.6.3 Investigating Cura and the network logs

Examination of the network dump provides the G-code files corresponding to the three objects under investigation. These files were sent from the IP address of the control machine. The control machine also hosts the Cura slicer software. All three recovered files were modified implying that either the control machine IP address was spoofed or the machine was hijacked. Ethernet (MAC) address associated with the IP address in the network dump confirms that the printing instructions were issued from the correct machine. Cura 'Recent files' logs show that two input files were fed to Cura software from a different path than the actual working directory. The third object file (driving_shaft.stl) was launched from the correct path. 'Cura profile logs' reveals that the printing profile in Cura software was modified and then reverted back. A suspicious printing profile, named 'Tmp_profile' increased the printing speed and the number of bottom layers from 5 to 10 as presented in Figure 77. The modified parameter does not apply to the top and bottom layers printing speed. Slicing a design file with the modified profile results in faster printing of the intermediate layers, whereas increasing the number of low-speed bottom layers compensates for the time difference.

6.7 Future Work

Preserving the evidence is a standardized task in the conventional IT domain and the entire ecosystem, including researchers, vendors, operators, and regulators, is well acquainted. With the increasing use of AM in critical manufacturing, we expect to see more interest in all facets of AM forensics. Instead of relying on conventional IT forensics, it is helpful to research the methodologies and processes best suited for AM. Our proposed framework, `FRoMEPP` and its illustration focus on material extrusion-based printing. In the future, we intend to conduct studies on other AM techniques. An essential aspect of our proposed approach is the inclusion of physical processes in the monitoring system. A future direction is to utilize this approach to create a generic forensic readiness framework applicable to all CPS.

As commercial decision-makers often overlook security and forensics, a practical challenge is to suggest compliance criteria for the AM equipment, service providers, and customer setups.

6.8 Conclusion

This chapter presented a forensic readiness framework, FRoMEPP, for material extrusion-based AM process incorporating the cyber and the physical domain information sources. FRoMEPP is explained through an implementation on a common printer - Ultimaker-3. The study formalized the physical-domain data acquisition process by identifying direct-manipulable sub-processes and ranking the available acquisition options based on a set of mandatory and discretionary features. The implementation also discussed the forensic artifacts extraction process from the acquired data. Some of the extracted forensic artifacts include per-layer geometry, timing and thermodynamic profiles of the printed object, copy of the running firmware, design files, recent user-activity lists, and configuration changelogs.

Through a case study of three sophisticated sabotage attacks, we demonstrated the effectiveness of FRoMEPP in identifying information about the attack, the attacker, and the attack mechanism. The presented artifacts can help find answers to the forensic questions related to the printing deviations and the failure causes, making this implementation a strong candidate to be replicated in important 3D printing setups. The study also serves as a foundational work to facilitate the standardization and regulatory organizations in creating compliance criteria and forensic readiness standards for AM echo cycle.

CHAPTER 7

CONCLUSIONS

Additive Manufacturing represents a paradigm shift from traditional manufacturing technologies, such as Subtractive Manufacturing, and poses a unique set of vulnerabilities and cyberattack vectors. As Additive Manufacturing emerges as a critical component of Industry 4.0, this thesis endeavors to elevate the state of the art in fused filament fabrication-based 3D printing cybersecurity through an exhaustive analysis of attack opportunities and corresponding countermeasures.

In Chapter 2, we outline filament-kinetic and dynamic-thermal attacks that fulfill inconspicuousness criteria, remain within known attack detection horizons, and yet compromise the mechanical strength of printed components.

Chapter 3 proposes attacks that manipulate the extrudate bonding process via inconspicuous kinetic deviations that fall within printer trueness specifications. Our study showcases the efficacy of intelligently designed attack patterns within these constraints to subvert the integrity of printed parts.

The impact of firmware attacks has been grossly underrepresented in the existing literature. In Chapter 4, we introduce an attack taxonomy and categorization tree that formalize offensive research in this domain. We also present ten intriguing attacks that span various attack surfaces, including surveillance, denial of printing service, printer damage, facility contamination, and printed part sabotage. Our study also presents a novel assessment of the feasibility of attaining attack objectives at different stages of the printing process.

These attacks motivate us to explore cyber-physical frameworks to secure the printing process. Even if traditional cybersecurity measures fail to detect a breach, the cyber-physical security solution may still identify it. Chapter 5 proposes a modular attack detection framework that employs spatiotemporal modeling of G-code files, semi-supervised fingerprinting of firmware-introduced functionalities, and independent sensor-based monitoring of physical printing states. `PrintSafe` leverages a series of algorithms in the time, space, and frequency domains to scrutinize the printing process for anomalies that may indicate malicious intervention. The effectiveness of `PrintSafe` in identifying sophisticated attacks is demonstrated in a case study, where it successfully detected 33 attacks of smaller magnitudes than those detectable by the existing state of the art. These results affirm the efficacy of `PrintSafe` in providing advanced and robust security measures for the FFF printing process.

Chapter 6 proposes a novel forensic readiness framework for FFF printing setups. The framework evaluates the process chain and identifies forensically valuable information sources, outlining a forensic methodology for successfully investigating cyberattacks on the FFF printing process. This study will also aid regulatory bodies in establishing compliance criteria for security-sensitive printing setups.

In summary, this thesis comprehensively analyzes attack opportunities across all stages of the printing process and demonstrates the efficacy of consolidating cyber and physical knowledge to secure the FFF process and conduct forensic analysis of the cyberattacks.

Appendix A

ABBREVIATIONS

AFI	Attack feasibility index
DeTSA	Denial of target system availability
DoPS	Denial of printing service
FFF	Fused filament fabrication
FDM	Fused deposition modeling
FRoMEPP	Forensic readiness framework for material extrusion printing process
LC	Layer change profile
LTP	Layer timing profile
LM	Layer map
MiTM	Man in the middle
OM	Object map
OT	Operational technology
OTP	Object timing profile
PYOG	Print your own grave
SaTS	Sabotage of target system
SD	Secure digital
SPEC	Socio-political-economic-cultural
SuTS	Surveillance of target system
STL	Stereolithography
VOC	Volatile organic compound

Appendix B

LIST OF PUBLICATIONS

1. **Muhammad Haris Rais**, Ye Li, and Irfan Ahmed. 2021. Spatiotemporal G-code modeling for secure FDM-based 3D printing. In Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems (ICCPS '21). Association for Computing Machinery, New York, NY, USA, 177–186. DOI:https:// doi.org/10.1145/3450267.3450545

2. **Muhammad Haris Rais**, Ye Li, Irfan Ahmed. Dynamic-thermal and localized filament-kinetic attacks on fused filament fabrication based 3D printing process, Additive Manufacturing, Volume 46, 2021, 102200, ISSN 2214-8604, https://doi.org/10.1016/j.addma.2021.102200.

3. **Muhammad Haris Rais**, Rima Asmar Awad, Juan Lopez, Irfan Ahmed, JTAG-based PLC Memory Acquisition Framework for Industrial Control Systems, Forensic Science International: Digital Investigation, Volume 37, Supplement, 2021, 301196, ISSN 2666-2817, https://doi.org/10.1016/j.fsidi.2021.301196.

4. **Muhammad Haris Rais**, Rima Asmar Awad, Juan Lopez, Irfan Ahmed, Memory forensic analysis of a programmable logic controller in industrial control systems, Forensic Science International: Digital Investigation, Volume 40, Supplement, 2022, 301339, ISSN 2666-2817, https://doi.org/10.1016/j.fsidi.2022.301339 **Best student paper award**

5. **Muhammad Haris Rais**, Muhammad Ahsan, Irfan Ahmed, FRoMEPP: Dig-

ital forensic readiness framework for material extrusion based 3D printing process, Forensic Science International: Digital Investigation, Volume 44, Supplement,2023,301510,ISSN 2666-2817,https://doi.org/10.1016/j.fsidi.2023.301510. (https://www.sciencedirect.com/science/article/pii/S2666281723000112)

6. **Muhammad Haris Rais**, Ahsan, M., Sharma, V., Barua, R., Prins, R., Ahmed, I. (2022). LOW-MAGNITUDE INFILL STRUCTURE MANIPULATION ATTACKS ON FUSED FILAMENT FABRICATION 3D PRINTERS. In: Staggs, J., Shenoi, S. (eds) Critical Infrastructure Protection XVI. ICCIP 2022. IFIP Advances in Information and Communication Technology, vol 666. Springer, Cham. https://doi.org/10.1007/978-3-031-20137-0_8

7. Muhammad Ahsan, **Muhammad Haris Rais**, Irfan Ahmed, SOK: Side Channel Monitoring for Additive Manufacturing - Bridging Cybersecurity and Quality Assurance Communities, Euro S&P July 2023, Status: **Accepted**

8. Rima Asmar Awad, **Muhammad Haris Rais**, Michael Rogers, Irfan Ahmed, Vincent Paquit, Towards generic memory forensic framework for programmable logic controllers, Forensic Science International: Digital Investigation, Volume 44, Supplement, 2023, 301513, ISSN 2666-2817,

   https://doi.org/10.1016/j.fsidi.2023.301513.

9. **Muhammad Haris Rais**, Muhammad Asad Arfeen, Evolving Internet Traffic Trend in Pakistan, 2016. Presented at 1st International Electrical Engineering Congress, May. 13-14, 2016 in IEP Centre, Karachi, Pakistan

10. **Muhammad Haris Rais**, Muhammad Ahsan, Irfan Ahmed, Weaponizing 3D printers: How firmware attacks enable adversarial objectives, NDSS 2024, Status:**Submitted**

REFERENCES

[1] Irene Buj-Corral, Aitor Tejo-Otero, and Felip Fenollosa-Artés. "Use of FDM Technology in Healthcare Applications: Recent Advances". In: *Fused Deposition Modeling Based 3D Printing*. Ed. by Harshit K. Dave and J. Paulo Davim. Cham: Springer International Publishing, 2021, pp. 277–297. ISBN: 978-3-030-68024-4. DOI: 10.1007/978-3-030-68024-4_15. URL: https://doi.org/10.1007/978-3-030-68024-4_15.

[2] Elizabeth Matias and Bharat Rao. "3D printing: On its historical evolution and the implications for business". In: *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*. 2015, pp. 551–558. DOI: 10.1109/PICMET.2015.7273052.

[3] Wangwang Yu et al. "Melt crystallization of PLA/Talc in fused filament fabrication". In: *Materials & Design* 182 (2019), p. 108013. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2019.108013. URL: https://www.sciencedirect.com/science/article/pii/S0264127519304514.

[4] Martin Spoerk et al. "Effect of the printing bed temperature on the adhesion of parts produced by fused filament fabrication". In: *Plastics, Rubber and Composites* 47.1 (2018), pp. 17–24. DOI: 10.1080/14658011.2017.1399531. eprint: https://doi.org/10.1080/14658011.2017.1399531. URL: https://doi.org/10.1080/14658011.2017.1399531.

[5] Ugur M. Dilberoglu et al. "The Role of Additive Manufacturing in the Era of Industry 4.0". In: *Procedia Manufacturing* 11 (2017). 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017,

27-30 June 2017, Modena, Italy, pp. 545–554. ISSN: 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2017.07.148.

[6] Inc Global Industry Analysts. *Additive Manufacturing & Material - Global Market Trajectory & Analytics*. 2021. (Visited on 11/19/2021).

[7] NIST Cloud Computing Forensic Science Working Group. *NIST Cloud Computing Forensic Science Challenges*. 2014. URL: https://csrc.nist.gov/csrc/media/publications/nistir/8006/draft/documents/draft_nistir_8006.pdf.

[8] Muhammad Haris Rais, Ye Li, and Irfan Ahmed. "Dynamic-thermal and Localized Filament-kinetic Attacks on Fused Filament Fabrication based 3D Printing Process". In: *Additive Manufacturing* (2021), p. 102200. ISSN: 2214-8604. DOI: https://doi.org/10.1016/j.addma.2021.102200. URL: https://www.sciencedirect.com/science/article/pii/S2214860421003614.

[9] Emad Abouel Nasr. *Rapid Prototyping: Theory and Practice*. Jan. 2006.

[10] Mark Yampolskiy et al. "Security of additive manufacturing: Attack taxonomy and survey". In: *Additive Manufacturing* 21 (2018), pp. 431–457. ISSN: 2214-8604. DOI: https://doi.org/10.1016/j.addma.2018.03.015.

[11] M. A. Al Faruque et al. "Acoustic Side-Channel Attacks on Additive Manufacturing Systems". In: *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*. 2016, pp. 1–10.

[12] Sofia Belikovetsky et al. "dr0wned – Cyber-Physical Attack with Additive Manufacturing". In: (Aug. 2017). URL: https://www.usenix.org/conference/woot17/workshop-program/presentation/belikovetsky.

[13] Steven Eric Zeltmann et al. "Manufacturing and Security Challenges in 3D Printing". In: *JOM* 68.7 (July 2016), pp. 1872–1881. ISSN: 1543-1851. DOI: 10.1007/s11837-016-1937-7. URL: https://doi.org/10.1007/s11837-016-1937-7.

[14] Yang Gao et al. "Watching and Safeguarding Your 3D Printer: Online Process Monitoring Against Cyber-Physical Attacks". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.3 (Sept. 2018). DOI: 10.1145/3264918. URL: https://doi.org/10.1145/3264918.

[15] Shouling Ding et al. "Effects of nozzle temperature and building orientation on mechanical properties and microstructure of PEEK and PEI printed by 3D-FDM". In: *Polymer Testing* 78 (2019), p. 105948. ISSN: 0142-9418. DOI: https://doi.org/10.1016/j.polymertesting.2019.105948.

[16] Martin Spoerk et al. "Effect of the printing bed temperature on the adhesion of parts produced by fused filament fabrication". In: *Plastics, Rubber and Composites* 47.1 (2018), pp. 17–24. DOI: 10.1080/14658011.2017.1399531.

[17] Mark Yampolskiy et al. "Security Challenges of Additive Manufacturing with Metals and Alloys". In: *Critical Infrastructure Protection IX*. Ed. by Mason Rice and Sujeet Shenoi. Cham: Springer International Publishing, 2015, pp. 169–183. ISBN: 978-3-319-26567-4.

[18] Christian Bayens et al. "See No Evil, Hear No Evil, Feel No Evil, Print No Evil- Malicious Fill Patterns Detection in Additive Manufacturing". In: *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1181–1198. ISBN: 978-1-931971-40-9.

[19] J. Gatlin et al. "Detecting Sabotage Attacks in Additive Manufacturing Using Actuator Power Signatures". In: *IEEE Access* 7 (2019), pp. 133421–133432.

[20] S. Belikovetsky et al. “Digital Audio Signature for 3D Printing Integrity”. In: *IEEE Transactions on Information Forensics and Security* 14.5 (2019), pp. 1127–1141.

[21] 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)*. 2016, pp. 1–8.

[22] Samuel Bennett Moore, William Bradley Glisson, and Mark Yampolskiy. “Implications of Malicious 3D Printer Firmware”. In: *Proceedings of Hawaii Int. Conf.Syst.Sci,2017*. 2017, pp. 1–10. DOI: `10.24251/HICSS.2017.735`. URL: `http://hdl.handle.net/10125/41899`.

[23] Xiao Zi Hang. *Three demos of attacking arduino and reprap 3d printers, code to Keynote at XCon2013 (2013).* `https://github.com/secmobi/attack-arduino-and-reprap`. 2016.

[24] Logan D. Sturm et al. “Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the .STL file with human subjects”. In: *Journal of Manufacturing Systems* 44 (2017), pp. 154–164. ISSN: 0278-6125. DOI: `https://doi.org/10.1016/j.jmsy.2017.05.007`. URL: `http://www.sciencedirect.com/science/article/pii/S0278612517300961`.

[25] Guanxiong Miao et al. “Cyber-physical system for thermal stress prevention in 3D printing process”. In: *The International Journal of Advanced Manufacturing Technology* 100 (Jan. 2019). DOI: `10.1007/s00170-018-2667-5`.

[26] Luis Garcia et al. “Hey My Malware Knows Physics Attacking PLCs with Physical Model Aware Rootkit”. In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. NDSS 2017. San Diego,

CA, USA: Internet Society, 2017. ISBN: 1-891562-46-0. URL: http://dx.doi.org/10.14722/ndss.2017.23313.

[27] Vosynek, Petr et al. “Influence of Process Parameters of Printing on Mechanical Properties of Plastic Parts Produced by FDM 3D Printing Technology”. In: *MATEC Web Conf.* 237 (2018), p. 02014. DOI: 10.1051/matecconf/201823702014.

[28] Matthew McCormack et al. “Security Analysis of Networked 3D Printers”. In: *2020 IEEE Security and Privacy Workshops (SPW)*. 2020, pp. 118–125. DOI: 10.1109/SPW50608.2020.00035.

[29] Boohyung Lee et al. “Firmware Verification of Embedded Devices Based on a Blockchain”. In: *Quality, Reliability, Security and Robustness in Heterogeneous Networks*. Ed. by Jong-Hyouk Lee and Sangheon Pack. Cham: Springer International Publishing, 2017, pp. 52–61. ISBN: 978-3-319-60717-7.

[30] Bo-Yuan Huang et al. “Formal Security Verification of Concurrent Firmware in SoCs using Instruction-Level Abstraction for Hardware*”. In: *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. 2018, pp. 1–6. DOI: 10.1109/DAC.2018.8465794.

[31] Muhammad Haris Rais, Ye Li, and Irfan Ahmed. “Spatiotemporal G-code modeling for secure FDM-based 3D printing”. In: *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*. 2021, pp. 177–186.

[32] Mohammad Abdullah Al Faruque et al. “Forensics of thermal side-channel in additive manufacturing systems”. In: *University of California, Irvine* (2016).

[33] Muhammad Haris Rais et al. “LOW-MAGNITUDE INFILL STRUCTURE MANIPULATION ATTACKS ON FUSED FILAMENT FABRICATION 3D

PRINTERS". In: *Critical Infrastructure Protection XVI*. Ed. by Jason Staggs and Sujeet Shenoi. Cham: Springer Nature Switzerland, 2022, pp. 205–232.

[34] Ian Gibson et al. "Material Extrusion". In: *Additive Manufacturing Technologies*. Cham: Springer International Publishing, 2021, pp. 171–201. ISBN: 978-3-030-56127-7. DOI: `10.1007/978-3-030-56127-7_6`. URL: `https://doi.org/10.1007/978-3-030-56127-7_6`.

[35] Javaid Butt and Raghunath Bhaskar. "Investigating the Effects of Annealing on the Mechanical Properties of FFF-Printed Thermoplastics". In: *Journal of Manufacturing and Materials Processing* 4.2 (2020). ISSN: 2504-4494. DOI: `10.3390/jmmp4020038`. URL: `https://www.mdpi.com/2504-4494/4/2/38`.

[36] Mark Yampolskiy et al. "Security of additive manufacturing: Attack taxonomy and survey". In: *Additive Manufacturing* 21 (2018), pp. 431–457. ISSN: 2214-8604.

[37] Yanzhou Fu et al. "In situ monitoring for fused filament fabrication process: A review". In: *Additive Manufacturing* 38 (2021), p. 101749. ISSN: 2214-8604.

[38] L Sturm et al. "Cyber-physical vunerabilities in additive manufacturing systems". In: *Context* 7.8 (2014), pp. 951–963.

[39] Elizabeth Kurkowski et al. "MANIPULATION OF G-CODE TOOLPATH FILES IN 3D PRINTERS: ATTACKS AND MITIGATIONS". In: *Critical Infrastructure Protection XVI*. Ed. by Jason Staggs and Sujeet Shenoi. Cham: Springer Nature Switzerland, 2022, pp. 155–174. ISBN: 978-3-031-20137-0.

[40] Hammond Pearce et al. "FLAW3D: A Trojan-Based Cyber Attack on the Physical Outcomes of Additive Manufacturing". In: *IEEE/ASME Transactions on Mechatronics* 27.6 (2022), pp. 5361–5370.

[41] Mingtao Wu et al. "Detecting Attacks in CyberManufacturing Systems: Additive Manufacturing Example". In: *MATEC Web of Conferences* 108 (Jan. 2017), p. 06005. DOI: 10.1051/matecconf/201710806005.

[42] Mingtao Wu, Zhengyi Song, and Young B Moon. "Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods". In: *Journal of intelligent manufacturing* 30.3 (2019), pp. 1111–1123.

[43] J. Hanssen. *Fortus 360mc/400mc Accuracy Study*. Tech. rep. Stratasys, Eden Prairie, Minnesota, 2013.

[44] Soo-Yeon Kim et al. "Precision and trueness of dental models manufactured with different 3-dimensional printing techniques". In: *American Journal of Orthodontics and Dentofacial Orthopedics* 153.1 (2018), pp. 144–153. ISSN: 0889-5406. DOI: https://doi.org/10.1016/j.ajodo.2017.05.025. URL: https://www.sciencedirect.com/science/article/pii/S0889540617306479.

[45] Bilal Msallem et al. "Evaluation of the Dimensional Accuracy of 3D-Printed Anatomical Mandibular Models Using FFF, SLA, SLS, MJ, and BJ Printing Technology". In: *Journal of Clinical Medicine* 9.3 (2020). ISSN: 2077-0383. DOI: 10.3390/jcm9030817. URL: https://www.mdpi.com/2077-0383/9/3/817.

[46] S.F. Khan et al. "Effect of infill on tensile and flexural strength of 3D printed PLA parts". In: *IOP Conference Series: Materials Science and Engineering* 429 (Nov. 2018), p. 012101. DOI: 10.1088/1757-899x/429/1/012101. URL: https://doi.org/10.1088/1757-899x/429/1/012101.

[47] Xia Gao et al. "Fused filament fabrication of polymer materials: A review of interlayer bond". In: *Additive Manufacturing* 37 (2021), p. 101658. ISSN: 2214-

8604. DOI: https://doi.org/10.1016/j.addma.2020.101658. URL: https://www.sciencedirect.com/science/article/pii/S2214860420310307.

[48] Assaf Morag and Itamar Maouda. “Understanding the evolving threat landscape – APT techniques in a container environment”. In: *Network Security* 2021.12 (2021), pp. 13–17. ISSN: 1353-4858. DOI: https://doi.org/10.1016/S1353-4858(21)00145-8. URL: https://www.sciencedirect.com/science/article/pii/S1353485821001458.

[49] Muhammad Haris Rais et al. “JTAG-based PLC memory acquisition framework for industrial control systems”. In: *Forensic Science International: Digital Investigation* 37 (2021), p. 301196. ISSN: 2666-2817. DOI: https://doi.org/10.1016/j.fsidi.2021.301196. URL: https://www.sciencedirect.com/science/article/pii/S2666281721001049.

[50] Srikanth Yoginath et al. “Stealthy Cyber Anomaly Detection On Large Noisy Multi-Material 3D Printer Datasets Using Probabilistic Models”. In: *Proceedings of the 2022 ACM CCS Workshop on Additive Manufacturing (3D Printing) Security*. AMSec’22. Los Angeles, CA, USA: Association for Computing Machinery, 2022, pp. 25–38. ISBN: 9781450398831. DOI: 10.1145/3560833.3563564. URL: https://doi.org/10.1145/3560833.3563564.

[51] KBV Research. “Share & Industry Trends Analysis Report by Type, by Technology, by Sales Channel, by End-Use, by Regional Outlook and Forecast, 2021–2027”. In: *ReportLinker: Lyon, France* (2022).

[52] GE Aviation. *New manufacturing milestone: 30,000 additive fuel nozzles.* 2018.

[53] Nayanee Gupta, Christopher Weber, and Sherrica Newsome. "Additive manufacturing: status and opportunities". In: *Science and Technology Policy Institute, Washington* (2012).

[54] Sujit Rokka Chhetri et al. "Manufacturing supply chain and product lifecycle security in the era of industry 4.0". In: *Journal of Hardware and Systems Security* 2.1 (2018), pp. 51–68.

[55] Jacob Gatlin et al. "Encryption is Futile: Reconstructing 3D-Printed Models Using the Power Side-Channel". In: RAID '21. San Sebastian, Spain: Association for Computing Machinery, 2021, pp. 135–147. ISBN: 9781450390583. DOI: 10.1145/3471621.3471850. URL: https://doi.org/10.1145/3471621.3471850.

[56] Mohammad Abdullah Al Faruque et al. "Acoustic Side-Channel Attacks on Additive Manufacturing Systems". In: *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*. 2016, pp. 1–10. DOI: 10.1109/ICCPS.2016.7479068.

[57] Robert C Nickerson, Upkar Varshney, and Jan Muntermann. "A method for taxonomy development and its application in information systems". In: *European Journal of Information Systems* 22.3 (2013), pp. 336–359. DOI: 10.1057/ejis.2012.26. eprint: https://doi.org/10.1057/ejis.2012.26. URL: https://doi.org/10.1057/ejis.2012.26.

[58] Joseph Flynt. *A guide to choosing firmware for your 3D printer*. https://3dinsider.com/choosing-firmware-3d-printer/. [Online; accessed 02-Feb-2023]. 2022.

[59] Mark Yampolskiy et al. “Security of additive manufacturing: Attack taxonomy and survey”. In: *Additive Manufacturing* 21 (2018), pp. 431–457. ISSN: 2214-8604. DOI: `https://doi.org/10.1016/j.addma.2018.03.015`.

[60] Yao Pan et al. “Taxonomies for reasoning about cyber-physical attacks in IoT-based manufacturing systems”. In: (2017).

[61] Stephan Berger, Olga Bürger, and Maximilian Röglinger. “Attacks on the Industrial Internet of Things – Development of a multi-layer Taxonomy”. In: *Computers & Security* 93 (2020), p. 101790. ISSN: 0167-4048. DOI: `https://doi.org/10.1016/j.cose.2020.101790`. URL: `https://www.sciencedirect.com/science/article/pii/S0167404820300754`.

[62] Priyanka Mahesh et al. “A survey of cybersecurity of digital manufacturing”. In: *Proceedings of the IEEE* 109.4 (2020), pp. 495–516.

[63] Mingtao Wu and Young B Moon. “Taxonomy of cross-domain attacks on cybermanufacturing system”. In: *Procedia Computer Science* 114 (2017), pp. 367–374.

[64] Samip Dhakal, Fehmi Jaafar, and Pavol Zavarsky. “Private Blockchain Network for IoT Device Firmware Integrity Verification and Update”. In: *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*. 2019, pp. 164–170. DOI: `10.1109/HASE.2019.00033`.

[65] Yanlin Li, Jonathan M. McCune, and Adrian Perrig. “VIPER: Verifying the Integrity of PERipherals’ Firmware”. In: *Proceedings of the 18th ACM Conference on Computer and Communications Security*. CCS ’11. Chicago, Illinois, USA: Association for Computing Machinery, 2011, pp. 3–16. ISBN: 9781450309486. DOI: `10.1145/2046707.2046711`. URL: `https://doi.org/10.1145/2046707.2046711`.

[66] John D Howard and Thomas A Longstaff. *A common language for computer security incidents.* Tech. rep. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia ..., 1998.

[67] Robin Gandhi et al. "Dimensions of Cyber-Attacks: Cultural, Social, Economic, and Political". In: *IEEE Technology and Society Magazine* 30.1 (2011), pp. 28–38. DOI: `10.1109/MTS.2011.940293`.

[68] Riccardo Colella, Francesco Paolo Chietera, and Luca Catarinucci. "Analysis of FDM and DLP 3D-Printing Technologies to Prototype Electromagnetic Devices for RFID Applications". In: *Sensors* 21.3 (2021). ISSN: 1424-8220. DOI: `10.3390/s21030897`. URL: `https://www.mdpi.com/1424-8220/21/3/897`.

[69] Muhammad Haris Rais, Muhammad Ahsan, and Irfan Ahmed. "FRoMEPP: Digital Forensic Readiness Framework for Material Extrusion based 3D Printing Process". In: ().

[70] Karthik et. al Babu. "Fire behavior of 3D-printed polymeric composites". In: *Journal of Materials Engineering and Performance*. Springer. 2021, 30:4745–4755. URL: `https://link.springer.com/article/10.1007/s11665-021-05627-1`.

[71] Qian Zhang et al. "Chemical Composition and Toxicity of Particles Emitted from a Consumer-Level 3D Printer Using Various Materials". In: *Environmental Science & Technology* 53.20 (2019). PMID: 31513393, pp. 12054–12061. DOI: `10.1021/acs.est.9b04168`. eprint: `https://doi.org/10.1021/acs.est.9b04168`. URL: `https://doi.org/10.1021/acs.est.9b04168`.

[72] Sandra Pirela et al. "Nanoparticle exposures from nano-enabled toner-based printing equipment and human health: state of science and future research

needs". In: *Critical reviews in toxicology* 47 (May 2017), pp. 1–27. DOI: 10.1080/10408444.2017.1318354.

[73] Atmel. *AVR068: STK500 Communication Protocol.* URL: https://www.diericx.net/downloads/STK500v2.pdf.

[74] Tobias Hullette Gabriel Boyd. *3D printing bridging: 6 tips for perfect bridges.* Last Updated: Nov 12, 2022.

[75] S. Yu et al. "Sabotage Attack Detection for Additive Manufacturing Systems". In: *IEEE Access* 8 (2020), pp. 27218–27231.

[76] Melissa Gaskill. *EPA Proposes to Strengthen Air Quality Standards to Protect the Public from Harmful Effects of Soot.* 2023.

[77] Q. Do, B. Martini, and K. R. Choo. "A Data Exfiltration and Remote Exploitation Attack on Consumer 3D Printers". In: *IEEE Transactions on Information Forensics and Security* 11.10 (2016), pp. 2174–2186.

[78] Sujit Rokka Chhetri, Arquimedes Canedo, and Mohammad Abdullah 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.

[79] Statistica. *Statistica, Worldwide most used 3D printing technologies, as of July 2018.* URL: https://www.statista.com/statistics/756690/worldwide-most-used-3d-printing-technologies.

[80] Young Choi et al. "Influence of Bed Temperature on Heat Shrinkage Shape Error in FDM Additive Manufacturing of the ABS-Engineering Plastic". In: *World Journal of Engineering and Technology* 04 (Jan. 2016), pp. 186–192. DOI: 10.4236/wjet.2016.43D022.

[81] Nahal Aliheidari et al. “The impact of nozzle and bed temperatures on the fracture resistance of FDM printed materials”. In: *Behavior and Mechanics of Multifunctional Materials and Composites 2017*. Ed. by Nakhiah C. Goulbourne. Vol. 10165. International Society for Optics and Photonics. SPIE, 2017, pp. 222–230. DOI: 10.1117/12.2260105. URL: https://doi.org/10.1117/12.2260105.

[82] Robert Rowlingson et al. “A ten step process for forensic readiness”. In: *International Journal of Digital Evidence* 2.3 (2004), pp. 1–28.

[83] Avinash Singh, Adeyemi R. Ikuesan, and Hein S. Venter. “Digital Forensic Readiness Framework for Ransomware Investigation”. In: *Digital Forensics and Cyber Crime*. Ed. by Frank Breitinger and Ibrahim Baggili. Cham: Springer International Publishing, 2019, pp. 91–105. ISBN: 978-3-030-05487-8.

[84] B K S P Kumar Raju and G Geethakumari. “An advanced forensic readiness model for the cloud environment”. In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*. 2016, pp. 765–771. DOI: 10.1109/CCAA.2016.7813819.

[85] Ameer Pichan, Mihai Lazarescu, and Sie Teng Soh. “Cloud forensics: Technical challenges, solutions and comparative analysis”. In: *Digital investigation* 13 (2015), pp. 38–57.

[86] Nikhil Gupta et al. “Additive Manufacturing Cyber-Physical System: Supply Chain Cybersecurity and Risks”. In: *IEEE Access* 8 (2020), pp. 47322–47333. DOI: 10.1109/ACCESS.2020.2978815.

[87] Munira Mohd Ali et al. “A product life cycle ontology for additive manufacturing”. In: *Computers in Industry* 105 (2019), pp. 191–203. ISSN: 0166-3615.

DOI: https://doi.org/10.1016/j.compind.2018.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0166361518301647.

[88] C.P. Grobler, C.P. Louwrens, and S.H. von Solms. “A Framework to Guide the Implementation of Proactive Digital Forensics in Organisations”. In: *2010 International Conference on Availability, Reliability and Security*. 2010, pp. 677–682. DOI: 10.1109/ARES.2010.62.

[89] Aleksandar Valjarevic and H.S. Venter. “Implementation guidelines for a harmonised digital forensic investigation readiness process model”. In: *2013 Information Security for South Africa*. 2013, pp. 1–9. DOI: 10.1109/ISSA.2013.6641041.

[90] Mohamed Elyas et al. “Towards A Systemic Framework for Digital Forensic Readiness”. In: *Journal of Computer Information Systems* 54.3 (2014), pp. 97–105. DOI: 10.1080/08874417.2014.11645708. eprint: https://doi.org/10.1080/08874417.2014.11645708. URL: https://doi.org/10.1080/08874417.2014.11645708.

[91] Roshan K Chaudhary et al. “Current and evolving applications of three-dimensional printing in forensic odontology: A review”. In: *International Journal of Forensic Odontology* 3.2 (2018), p. 59.

[92] Rachael M Carew, Ruth M Morgan, and Carolyn Rando. “A preliminary investigation into the accuracy of 3D modeling and 3D printing in forensic anthropology evidence reconstruction”. In: *Journal of forensic sciences* 64.2 (2019), pp. 342–352.

[93] Katie Foster et al. “The use of additive manufacturing for the presentation of ballistic toolmark evidence in court”. In: (2015).

[94] Mauro Conti et al. “Internet of Things security and forensics: Challenges and opportunities”. In: *Future Generation Computer Systems* 78 (2018), pp. 544–546. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.07.060. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17316667.

[95] Maria Stoyanova et al. “A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues”. In: *IEEE Communications Surveys Tutorials* 22.2 (2020), pp. 1191–1221. DOI: 10.1109/COMST.2019.2962586.

[96] Victor Garcia and Cihan Varol. “Digital forensics of 3D printers”. In: *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE. 2018, pp. 1–8.

[97] Irfan Ahmed et al. “A SCADA System Testbed for Cybersecurity and Forensic Research and Pedagogy”. In: *Proceedings of the 2nd Annual Industrial Control System Security Workshop (ICSS)*. Los Angeles, CA, USA, 2016. ISBN: 978-1-4503-4788-4.

[98] I. Ahmed et al. “Programmable Logic Controller Forensics”. In: *IEEE Security Privacy* 15.6 (Nov. 2017), pp. 18–24. ISSN: 1540-7993.

[99] Syed Ali Qasim, Juan Lopez, and Irfan Ahmed. “Automated Reconstruction of Control Logic for Programmable Logic Controller Forensics”. In: *Information Security*. Cham: Springer International Publishing, 2019, pp. 402–422. ISBN: 978-3-030-30215-3.

[100] I. Ahmed et al. “SCADA Systems: Challenges for Forensic Investigators”. In: *Computer* 45.12 (Dec. 2012), pp. 44–51. ISSN: 0018-9162.

[101] Syed Ali Qasim, Jared M Smith, and Irfan Ahmed. “Control logic forensics framework using built-in decompiler of engineering software in industrial control systems”. In: *Forensic Science International: Digital Investigation* 33 (2020), p. 301013.

[102] Rima Asmar Awad et al. “Tools, techniques, and methodologies: A survey of digital forensics for scada systems”. In: *Proceedings of the 4th Annual Industrial Control System Security Workshop*. 2018, pp. 1–8.

[103] Muhammad Haris Rais et al. “Memory forensic analysis of a programmable logic controller in industrial control systems”. In: *Forensic Science International: Digital Investigation* 40 (2022). Selected Papers of the Ninth Annual DFRWS Europe Conference, p. 301339. ISSN: 2666-2817. DOI: https://doi.org/10.1016/j.fsidi.2022.301339. URL: https://www.sciencedirect.com/science/article/pii/S2666281722000087.

[104] Saranyan Senthivel et al. “Denial of Engineering Operations Attacks in Industrial Control Systems”. In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. CODASPY ’18. Tempe, AZ, USA: ACM, 2018, pp. 319–329. ISBN: 978-1-4503-5632-9.

[105] Adeen Ayub, Hyunguk Yoo, and Irfan Ahmed. “Empirical Study of PLC Authentication Protocols in Industrial Control Systems”. In: *2021 IEEE Security and Privacy Workshops (SPW)*. 2021, pp. 383–397. DOI: 10.1109/SPW53761.2021.00058.

[106] Nurul Hidayah Ab Rahman et al. “Forensic-by-Design Framework for Cyber-Physical Cloud Systems”. In: *IEEE Cloud Computing* 3.1 (2016), pp. 50–59. DOI: 10.1109/MCC.2016.5.

[107] M. A. Faruque. "Forensics of Thermal Side-Channel in Additive Manufacturing Systems". In: 2016.

[108] Mark Yampolskiy et al. "Taxonomy for Description of Cross-Domain Attacks on CPS". In: *Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*. HiCoNS '13. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 2013, pp. 135–142. ISBN: 9781450319614. DOI: `10.1145/2461446.2461465`. URL: `https://doi.org/10.1145/2461446.2461465`.

[109] Raihana Md Saidi et al. "Windows registry analysis for forensic investigation". In: *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*. IEEE. 2013, pp. 132–136.

[110] Bhupendra Singh and Upasna Singh. "A forensic insight into Windows 10 Jump Lists". In: *Digital Investigation* 17 (2016), pp. 1–13. ISSN: 1742-2876. DOI: `https://doi.org/10.1016/j.diin.2016.02.001`. URL: `https://www.sciencedirect.com/science/article/pii/S1742287616300202`.

[111] Erwin van de Wiel, Mark Scanlon, and Nhien-An Le-Khac. "Enabling Non-Expert Analysis OF Large Volumes OF Intercepted Network Traffic". In: *Advances in Digital Forensics XIV*. Ed. by Gilbert Peterson and Sujeet Shenoi. Cham: Springer International Publishing, 2018, pp. 183–197. ISBN: 978-3-319-99277-8.

[112] Liam O Murchu Nicolas Falliere and Eric Chien. "W32.Stuxnet Dossier". In: (2011). URL: `https://www.symantec.com/content/en/us/enterprise/media/%20security_response/whitepapers/w32_stuxnet_dossier.pdf`.

[113] Lukasz Miazio. “Impact of Print Speed on Strength of Samples Printed in FDM Technology”. In: *Agricultural Engineering* 23 (June 2019), pp. 33–38. DOI: 10.1515/agriceng-2019-0014.

VITA

Muhammad Haris Rais graduated with a B.E degree in Electrical Engineering from the National University of Sciences and Technology in Pakistan. For over a decade, he has worked in the ICT industry, designing and implementing nationwide network and security solutions for various corporate and telecom service providers. After completing his M.E. degree in Computer Engineering, he joined the Ph.D. program at VCU's Computer Science department in 2019. Currently, he is a Ph.D. candidate and research assistant at the Security and Forensics Engineering (SAFE) lab under the guidance of Professor Dr. Irfan Ahmed. His research primarily focuses on the cybersecurity and forensics of Additive Manufacturing and Industrial Control Systems, demonstrating his passion for the application of technology in safeguarding modern industrial processes.