Spatiotemporal G-code Modeling for Secure FDM-based 3D Printing

Muhammad Haris Rais
Virginia Commonwealth University
raismh@vcu.edu

Ye Li
Bradley University
yli@bradley.edu

Irfan Ahmed
Virginia Commonwealth University
iahmed3@vcu.edu

ABSTRACT

3D printing constructs physical objects by building and stacking layers according to the CAD (Computer-aided Design) information. Attackers target a printing object by manipulating the printing parameters such as nozzle movement and temperature. The existing research on secure 3D printing mostly focuses on nozzle-kinetics, while attacks on filament-kinetics and thermodynamics can also damage the printed object. The detection of these attacks mainly relies on creating master-profile and machine learning by printing every unique object in a protected environment. In the fourth industrial revolution, such an approach is not suitable due to mass-customization rather than bulk production. This paper presents Sophos, a framework to detect nozzle-kinetic, filament-kinetic and thermodynamic attacks on the fused deposition modeling (FDM)-based 3D printing process. Sophos design does not require any prior learning for every unique object. It can detect the attacks on the first print using spatiotemporal G-code modeling, aligning it with the Industry 4.0 vision. Sophos is scalable and supports modular upgrades to suit different printing requirements. Its design allows the detection threshold to be reduced conveniently to as low as the 3D printer's resolution, shifting the game to a more interesting study of attack patterns than attack magnitudes.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics.

KEYWORDS

3D printing, Filament-kinetic Attacks, Thermodynamic Attacks

ACM Reference Format:
Muhammad Haris Rais, Ye Li, and Irfan Ahmed. 2021. Spatiotemporal G-code Modeling for Secure FDM-based 3D Printing. In *ACM/IEEE 12th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2021) (ICCPS '21), May 19–21, 2021, Nashville, TN, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3450267.3450545

1 INTRODUCTION

3D printing is adopted as an essential element of Industry 4.0 [1] and is increasingly utilized to print critical functional components of physical processes such as automobiles and airplanes. Thus, the cybersecurity of 3D printing process is a growing concern. Among the seven categories of additive manufacturing processes defined by American Society for Testing and Materials (ASTM) 52900:2015, Material Extrusion based technique, Fused Deposition Modeling (FDM) captures around 50% of 3D printing market share [2]. In FDM printing, a molten filament is extruded from a hot nozzle that moves in 2-dimensions to create a thin layer geometry, stacked with multiple layers to print a complete 3D object.

FDM is a complex process [3] involving multiple printing parameters that influence the mechanical properties of the final object. Researchers have demonstrated that the object properties can be altered by manipulating the manufacturing parameters, such as object orientation [4], printer's fan speed [5], nozzle temperature [6], printing bed temperature [7], fusing material patterns [8], and combination of properties [3].

The 3D printing process can be attacked for different objectives; the most common ones are intellectual property (IP) theft and sabotage attacks [9]. As IP theft attacks can be completely passive, the research to protect against IP theft attack focuses around protecting the cyber-domain information and obfuscating the unavoidable side-channels (such as acoustic signals from the printer motors).

Sabotage attack is a different category where the attacker targets a 3D printing environment to weaken, damage, or destroy a 3D printing object by causing geometrical nonconformity and workpiece deformation. The focus of this paper is to detect low profile sabotage attacks where small malicious modifications or deviations in the manufacturing parameters compromise the object properties without causing any visual deformation [5, 10, 11]. The modifications may be achieved by targeting the nozzle-kinetics, filament-kinetics, or thermodynamics of the printing process. Researchers have proposed multiple techniques to monitor nozzle-kinetics (covered in related work), but as per our knowledge, no work is available in sabotage-attacks detection that utilizes filament-kinetics, nozzle-kinetics, and thermodynamics in a consolidated framework.

To find (malicious) deviations in a 3D printing object, Sophos utilizes the printing instructions file (G-code) as a reference to expected behavior. G-code is a series of instructions that defines how an object should be printed but does not provide an instantaneous state of the process. Using the G-code, Sophos employs spatiotemporal modeling and interpolation functions to derive a comprehensive set of desired instantaneous printing state.

Sophos utilizes the layer change as a logical marker to split the evaluation process on a per-layer basis. This split transforms a 3D object into a series of 2D shapes for simplified analysis and helps in early attack detection (as soon as the first attacked layer is printed). The printing process is presented in space and time

ICCPS '21, May 19–21, 2021, Nashville, TN, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8353-0/21/05…$15.00
https://doi.org/10.1145/3450267.3450545

domains. In the space-domain, each layer is represented by a bitmap where each pixel is an instance of the printing state. The bitmap representation provides scalability property to the framework. By merely adjusting the pixel dimensions and the acquisition sensors' resolution, the system's resolution can be improved. For the time-domain analysis, Sophos generates the timing profile from the reference design file (G-code file) and compares it with the acquired data to identify timing integrity issues. From the third standpoint, Sophos tests the integrity of the toolpath sequence (the complete path that nozzle takes during the printing) by verifying the move instruction vertices and the nozzle deviation from the desired path. Sophos also evaluates the filament-kinetic and thermodynamic profiles to reliably detect the attacks that do not use nozzle-kinetic deviations.

We implement and evaluate Sophos on forty different attack instances of sixteen types of attacks on a 3D printer. The results show that Sophos can detect all attack instances successfully with zero false negatives (FN) and false positives (FP).

The rest of the paper is organized as follows. Section 2 describes the background and related work. Sophos framework for attack detection is presented in section 3. Sections 4 describes the implementation of the Sophos. Section 5 presents the results and evaluation, followed by limitations, future work and conclusion.

2 RELATED WORK

The part of 3D printing security that relates to this work is the detection of sabotage attacks. Various techniques to address the print integrity attacks on the FDM process have been proposed by researchers. Almost all the efforts are focused on acquiring the printer state through variety of independently deployed sensors, and not relying on the system's feedback. Chhetri *et al.* [12] was the first to use audio sensors for 3D printer's security by picking the acoustic signals generated by the stepper motors. The approach resulted in overall 77% accuracy in detecting path and speed modifications. For small deviations, 3 mm or less, the true positive rate was 71% and false positive rate was over 30%. Belikovetsky *et al.* [13] used the acoustic sensor of a cell phone with a different algorithm to achieve higher accuracy and improved resolution. The solution was able to detect modifications that sustain for at least 1 second. If modification pertains to reordering of commands, the minimum detectable threshold changes to 2.6 seconds.

Gao *et al.* [5] utilized IMU sensors and cameras (for voice and video signal), and applied sensor fusion techniques and random forest to detect the attack on kinetic properties. The selected test cases were significantly distant from the original print; around 20% change in infill, change of fan speed from 100% to 25%, and printing speed change from 30 mm/s to 120 mm/sec. The accuracy of the solution under small deviations was not ascertained. Wu *et al.* [14] used static and moving camera images to detect infill pattern attacks. The injected attack patterns were big enough to cover around 10-20% of the infill area. Bayens *et al.* [15] used a mic, IMU sensors and camera to verify the infill pattern. Specially doped filament was used to verify material integrity via CT Scan. The infill pattern selected for testing the technique were Honeycomb and Rectilinear with 20%, 40% and 60% density, which is a big difference to detect.

Gatlin *et al.* [16] conducted an interesting study of creating power signature profile of an object by measuring the electric current drawn by the stepper motors during printing. The approach required multiple prints to create a master profile, and could not detect individual layer thickness variations and filament extrusion attacks. Yu *et al.* [17] extended the previous work [12] and added more side-channels to improve the results. By studying acoustic, magnetic and visual information, the model successfully detected a deviation of 4 mm long, and authors mentioned the limitation of identifying smaller changes.

These approaches do not solve the problem of filament extrusion rate variations. If an attacker stops the filament extrusion motor or changes its speed, the object specifications will be changed. Camera-based inspection technique proposed by Wu *et al.* [14] may work (though not tested) in one case when the attacker halts the extrusion motor completely; however if only the density is changed by reducing the filament motor's speed, it seems unlikely that a visual camera will detect the change. Another challenge with camera based approach is that the printhead on top of the object obstructs the camera's view, and it is not feasible to pause the printing to capture a clean image after each layer.

The minimum detectable change in the existing techniques is not small enough to restrict 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. Researchers have already shown the impact of some of those attacks and their practicality [10, 18].

Another important limitation in most of the existing work is the learning phase, that 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 Industry 4.0 vision of mass customization.

3 PROPOSED ATTACK DETECTION FRAMEWORK - SOPHOS

To ascertain the process integrity in 3D printing, we propose Sophos framework.

3.1 Overview of Sophos Framework

Figure 1 provides an overview of Sophos, our proposed framework for checking integrity of FDM-based 3D printing process. Sophos deploys independent sensors to obtain the reliable state of printing process, and uses G-code file as a 'source of truth' to verify the acquired sensors' data. We prefer G-code over stereolithography (STL) file because it contains more design parameters to define the final object characteristics. Sophos transforms both the G-code file and the sensors data into comparable space and time-domain vectors. In addition to analyzing the printing process in space and time domains, Sophos checks the integrity of the G-code's execution on a per-instruction basis.

3.2 Acquiring and Transforming Sensor-data

Sophos measures the critical parameters of filament-kinetics, nozzle-kinetics and thermodynamics of the printing process, including location of the printhead in x,y and z axes, length of the filament extruded (e), an temperature of the nozzle and the printing bed. For acquisition of the state of these distinct physical processes, the

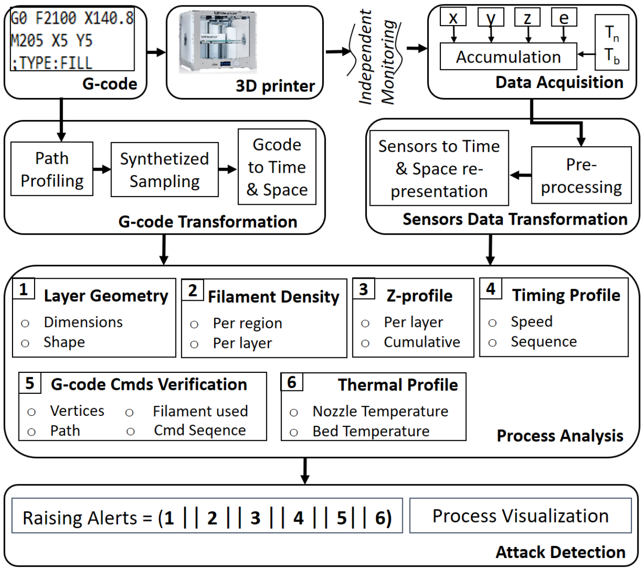


Figure 1: Sophos framework for secure 3D printing process

sampling rate is worked out inline with the suitability for the parameter while avoiding over-burdening the acquisition system. For example, nozzle location changes much faster than its temperature; thus the former is sampled at a higher rate.

3.2.1 Location-data Acquisition. The printer firmware converts a G-code move instruction to electrical signals, and applies them to the relevant stepper motors. The motors draw electric current to convert electrical energy to motor rotation. The rotor is mechanically coupled (via belts/shafts/gears) to the target object (printhead or filament) to accomplish the desired movement. There are multiple options to measure this movement. Acoustic sensors are susceptible to common environmental noises. It is also complicated to accurately distinguish between the noises of different motors, and the direction of move. We evaluated accelerometers, but found them to be too sensitive to the usual vibrations on the printing table. Electrical current has direct relation to the motor rotation, but with changing load of the driving shafts/belts, the profile may change. Sophos utilizes optical encoders for the movement tracking. The rotary optical encoders have a pair of LED and photo-detector with a disc, marked with opaque and transparent patterns, rotating through a slot between them generating electrical pulses. Sophos uses rotary encoders to detect x,y and e axes movement. A linear optical encoder, that replaces the disc with a marked strip, is used for detecting the printing bed movement. Sophos data acquisition performance depends upon the sensors' resolution, and the deployment quality.

3.2.2 Temperature Measurement. Printing bed and the extrusion nozzle are the two important thermodynamic components of a FDM printer. Sophos uses a thermistor with simple voltage divider circuit to measure printing bed temperature. For measuring nozzle temperature, Sophos utilizes thermocouple with driver circuit to amplify the weak signal. These are commonly available sensors, and used by researchers for studying thermal stresses in FDM [19].

3.2.3 Layer-by-layer Analysis. We consider FDM printer as a system with G-code file as the input, and printed object as the system's output. We represent a printed object as the set of instantaneous states of the printer; the states can be represented in time or space-domain.

2D Multi-attribute Matrices (LM). Sophos uses the layer-change event as a logical marker to transform 3D object into a series of multi-attribute 2D arrays. We denote the array as the layer-map (LM$_k$) where 'k' represents the kth layer. Each pixel $LM_{k_{i,j}}$ of the kth layer is defined in Equation 1

$$LM_{k_{i,j}} = [t, T, e, c] \tag{1}$$

where 't', 'T', 'e' and 'c' are the time of print of the pixel, nozzle temperature, length of the filament consumed, and the pixel's color (extrusion status) respectively. LM provides the space-domain view for each layer, but does not cater for layer transition event. Sophos represents the layer-change (LC) as a uni-dimensional array as shown in Equation 2

$$LC = \{lc_1, lc_2, \ldots, lc_n\} \tag{2}$$

where lc$_i$ represents the z-axis relative displacement between the nozzle and the printing bed after the completion of ith layer. By combining 1 and 2, we can represent the complete 3D printing process as shown in Equation 3.

$$3D\_object = \{\{LM_1, lc_1\}, \{LM_2, lc_2\} \ldots \{LM_n, lc_n\}\} \tag{3}$$

Multiattribute Timed Samples Sequence (TP). Due to the importance of timing integrity in 3D printing process, Sophos examines the process in time-domain as well. As in space-domain, we use layer-change event as boundary, and calculate the set of time-indexed samples named as 'Time Profile' (TP) for each layer. A single sample is expressed as a set of attributes as shown in Equation 4.

$$TP_{k_t} = \{x, y, e, c, T_n, T_b\} \tag{4}$$

where x,y represent the printhead location, T$_n$ and T$_b$ are nozzle and bed temperatures. A 3D printed object can be completely represented in time-domain by a set of TPs and LC for each layer.

$$3D\_object = \{\{TP_1, lc_1\}, \{TP_2, lc_2\} \ldots \{TP_n, lc_n\}\} \tag{5}$$

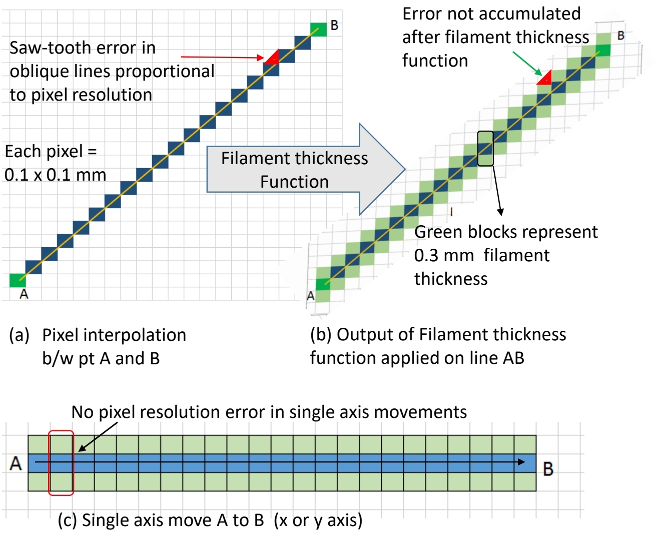
Equation 5 provides time-domain standpoint to view the process integrity. Equations 3 and 5 transformation helps in identifying the anomalies after each layer, instead of waiting for the object to be completely printed. Per-layer analysis also helps in examining internal layers infill issues that may be concealed in the final object.

3.3 G-code Transformation

The G-code file is a series of instructions provided to the printer. It contains move instructions in x,y,z, and e axes, and specifies parameters like temperature values, max-speed, max-acceleration etc. The slicer software that generates G-code file provides an estimated time taken to print each layer, with no breakdown of this duration (typically 10s to 100s of seconds). On the contrary, the Sophos sensors are providing state information at the sampling interval (in milliseconds). For instantaneous comparison of the G-code and the sensors' time-samples sequence, we convert the G-code instruction-set into a synthetic time-sampled sequence, and

Algorithm 1 : Path profiling and attribute assignment

```
Output: Updated Layer Map (LM_k)
Input: Points A, B , Layer Map (LM_k)
Calculate major Axis x or y, and slope of AB
∀ i ∈ majorAxis
    Find j in minorAxis ⟵ 10 * (Round to 0.1 * (slope * i))
∀ pixel (i,j) in LM_k, assign Attributes:
    Time_of_print(i,j)⟵ (i / majorAxis) * Δt
    Nozzle_temperature (i,j) ⟵(T_A + T_B ) / 2
    Filament_length (i,j) ⟵ (i / majorAxis) * Δe
Repeat attribute assignment for filament thickness
```


B
Saw-tooth error in oblique lines proportional to pixel resolution
Each pixel = 0.1 x 0.1 mm
A
Filament thickness Function
Error not accumulated after filament thickness function
B
Green blocks represent 0.3 mm filament thickness
A
No pixel resolution error in single axis movements
A
B

(a) Pixel interpolation b/w pt A and B

(b) Output of Filament thickness function applied on line AB

(c) Single axis move A to B (x or y axis)

Figure 2: Path profile for move command with pixel approximation error

space-domain representation. We estimate LM, TP, and bitmap (bmp) image for each layer.

Path Profiling. The G-code move commands only provide the end points of the move instruction. We use path profiling function that incorporates 1^{st} and 2^{nd} order time derivatives to estimate the move profile complying to the max speed and acceleration constraints defined in G-code. For pixel dimensions of 0.1 mm x 0.1 mm (used later in case study), a 20 cm x 20 cm printing bed corresponds to 2000 x 2000 pixels matrix. The move instruction is linear whereas the extruded filament is a 3D paste. Through filament thickness function, we convert linear path to 2D rectangular bar, where the width of the bar is dependent on the nozzle dimensions and the printing profile. For moves involving single axis (x or y axis), this conversion is lossless as seen in Figure 2 (c). For oblique lines, the error is proportional to the angle of the line with the nearest axis, and the pixel dimensions. However, the error does not accumulate after applying filament thickness function, as visible in Figure 2 (a) & (b). Third dimension (height of the extruded filament) is examined through filament consumption and z-axis movement (LC). For sensor-data conversion, Sophos uses a compatible set of profiling functions based on interpolation functions to complete any missing information between two consecutive data samples. Path profiling and attribute-assignment algorithm is presented in Algorithm 1. The algorithm updates the LM as per the latest steady sample's values.

3.4 Process Analysis for Integrity Checking

Sophos uses space-domain and time-domain representations LM, TP and LC for six verification tests to detect attacks on the printing process. These tests examine the process for different types of attacks after each layer is printed.

3.4.1 Layer Geometry. In this test, each layer is examined for the overall dimensions and shape using LM_{Gcode} and LM_{Sensor}. We consider the infill lines as geometry features. The color or extrusion status attribute of the layer-map is used in this test. Cumulative test compares the overall dimensions of the object during the printing. For the instance-based test, Sophos calculates a differential layer-map (Equation 6) by evaluating each pixel (x_i,y_j) in the G-code bitmap for a corresponding matching pixel in the synchronized LM_{Sensor}. The vicinity threshold in synchronization function compensates for benign printing deviations, sensor measurement and filament thickness estimation functions errors.

$$\Delta LM_k = Diff(LM_{Gcode}\ ,\ sync(LM_{Sensor})) \tag{6}$$

With ΔLM_k, we calculate the single biggest mismatched area, and total mismatched area per layer. For biggest area calculation, the criteria is to check all contiguous mismatched areas, may they be of any irregular shape. Among other details, this algorithm generates accurate bmp images of the desired and actual print for each layer.

3.4.2 Z-Axis Profile. The z-axis motor is engaged once the layer-change event occurs. Z-axis movement dictates the layer thickness, and is very small and short-lived, demanding higher resolution sensing. To examine layer thickness, Sophos continuously monitors the z-location through a small sliding window to attain a stable reading across the ends of the window. To ascertain the validity of the layer-thickness, Sophos uses LC vectors to compare individual layer's thickness, and object's thickness after each layer.

3.4.3 Timing Profile. Sophos uses TP_{Gcode} and TP_{Sensor} vectors to evaluate the timing integrity of the printing process. This analysis is helpful in detecting printing speed attacks, in which the geometry remains the same but the object is printed at a different speed. Another use-case of timing profile verification is to detect printing sequence modification attack. If the profiles deviate by more than a specified threshold, an alert is raised. The detection threshold depends upon the accuracy of synthesized sampling.

3.4.4 G-Code Commands Verification. The TP_{Gcode} is generated through motion equations (accommodating up to 2^{nd} order derivatives). The higher-order terms (such as max-jerk setting) are ignored, and may result in minor inaccuracies. The G-code move instructions provide another standpoint to analyze the process in discrete steps. Sophos sequentially picks the G-code move commands (bigger than a minimum threshold), and verifies the sensors' acquired data. The algorithm starts by synchronizing the acquired samples with the first selected move command. Synchronization

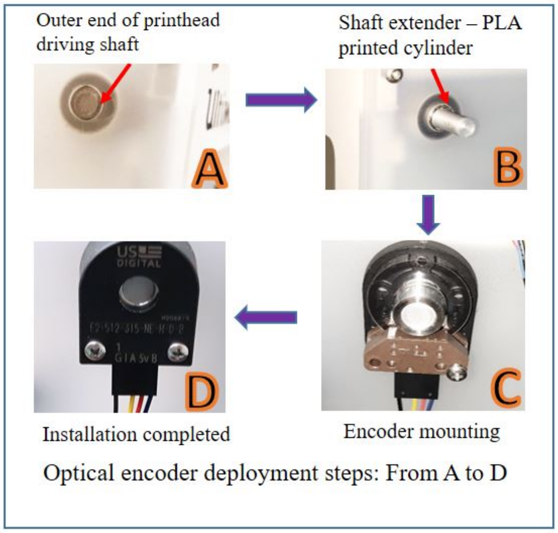

Outer end of printhead driving shaft
Shaft extender – PLA printed cylinder
A
B
C
D
Installation completed
Encoder mounting
Optical encoder deployment steps: From A to D

Figure 3: Optical encoder deployment process

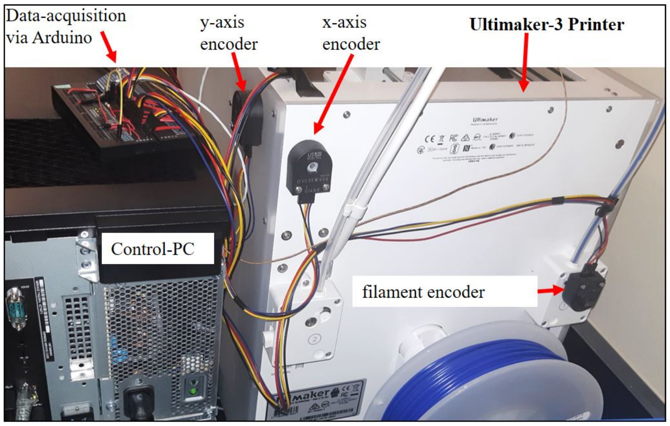


Figure 4: Sophos testbed setup

refers to the ability to find samples sub-sequence from S$_i$ (first sample) to S$_j$ (last sample of the sub-sequence), such that,

$$|d_{i,j} - |\overline{AB}|| < th_{dis} \quad AND \quad |\theta_{i,j} - \theta_{\overline{AB}}| < th_{\theta} \tag{7}$$

where d$_{i,j}$ is the distance between S$_i$ and S$_j$, θ_{AB} is the slope of line $\overline{AB}$ with x-axis, $\theta_{i,j}$ is the slope of line $\overline{S_i S_j}$ and x-axis, th$_{dis}$ and th$_\theta$ are the displacement and slope thresholds respectively.

Synchronization helps in translating absolute coordinates to relative ones by calculating δx and δy (difference between point A and sample S$_i$). This mechanism rules out the error accumulation problem that could occur due to the inaccuracies in the sensor data measurement and transformation stages of the framework. After synchronization, Sophos sequentially picks the move commands, and verifies if every next sample is getting monotonically closer to B, while not deviating from the line $\overline{AB}$. Another check performed in every command is the filament consumption check. In case, the filament consumption changes above the allowed threshold, the command fails the integrity check, and alert is raised. Algorithm 2 explains the verification process.

Algorithm 2 : G-code commands verification

```
Output: Integrity Status AND Process Logs
Input: G-code file , SensorData: FastSampleSequence
Synchronize sample series "S" with 1st command>1mm: move
from A to B
∀ i ∈ SampleSequence
    Assume s$_i$ = A , if ∃ s$_j$ s.t.
        |d$_{i,j}$ - |AB|| ≤ th$_{vertex}$ // ie. B in vicinity of (s$_j$)
        AND slope$_{AB}$ ≡ slope$_{(s_i,s_j)}$
    if s$_j$ NotFound ⟶ return:   Sync Failed
    else    Sync Achieved, Verify Individual Cmds
    for k in list_of_Move commands)
∀ k ∈ Gcode_Mov_Cmds
    Find sample s$_i$ corresponding to vertex A
    if A found:
        refineVertex(A)⟶ s$_i$
    while B found: Chronologically test fol
        1.   d(s$_{i+1}$ - $_B$) ≤ d(s$_i$ - $_B$)
        2.   dist [s$_i$-AB] ≤ th$_{point2line}$
        if B found:
            refineVertex(B)⟶ s$_j$
        else: return Cmd k failed due to {Reason}
        Filament consumption test
        if ((e$_{sj}$ − e$_{si}$) − (e$_B$ - e$_A$ ) ) ≤ th$_{filament}$
            Command k Verified
            else return Cmd k failed due to {Reason}
    return Cmd verification passed
```

3.4.5 Thermodynamic Profile. For thermal profile verification, Sophos uses the sensors reading against the G-code temperature value. G-code can have legitimate temperature change instructions. Such change may take few seconds to be completed. Sophos approximates the temperate modification phenomenon as a linear equation. For a G-code instruction of increasing temperature by 5^{o}C, Sophos approximation may result in temporary mismatch in expected and actual readings. Owing to the small magnitude and duration of error, we stick to the approximation function, and compensated for the error using allowed threshold parameter for the number of mismatched samples. Sophos also requires a deviation threshold that depends upon the benign temperature changes due to hysteresis and the profile settings.

4 IMPLEMENTATION AND TESTBED DETAILS

4.1 Testbed Setup

The generic Sophos framework can be implemented on any FDM printer that allow sensors mounting for measuring the displacement of the monitored axes; whether from the stepper motors, or directly from the targeted moving part. After reviewing the product literature of few printers on internet, we find printers like Ultimaker3, Prusa i3 MK3, Lulzbot TAZ6, Creality 3D Ender-3 pro to be feasible for Sophos deployment. For this experiment, we use Ultimaker-3

Purpose	Sensor Type	Vendor	Model Number	Specs	Resolution as per System Deployment
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
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

Table 1: Sophos sensors specifications

printer connected via Ethernet to control-PC (core i7-8700 / 16GB RAM / Windows OS), and managed through open source Cura 4.0 software. The nozzle and the filament diameters were 0.4 mm and 2.85 mm respectively.

4.1.1 Sensors' Selection and Deployment. Table 1 provides the list of sensors deployed on the printer. The sensors are selected to achieve sub-mm detection capability for the object geometry, and 1^{o}C temperature accuracy. Encoder with 4 times more pulses per revolution is used for filament motor to partly compensate for its slow speed. Figure 3 illustrates our sensors' deployment method on the printer, which is non-intrusive, simple, robust and quick to replicate. We printed 8 mm diameter, 15 mm length PLA cylindrical rods, and affixed them to the outer exposed end of the rotating shaft of the measured axes. During this manual extension of shaft, we suspect to exceed the acceptable tolerances of the axial and radial play for the optical encoder (± 0.01 in and ±0.04 in respectively). Exceeding the tolerance does not void the experiment, but it does affect the accuracy slightly. Instead of investing time in removing small mechanical coupling errors, we measured the results through stage wise testing, and calculated the reported accuracy. If the resultant accuracy is below the required value, a compensation profile can be calculated on a per-axis basis.

The thermocouple is a shielded flexible wire that is conveniently routed besides the tip of the nozzle, and secured with the help of a heat resistant polyimide adhesive strip based platform supported by the nozzle cover. The thermistor is secured to the printing bed through the stick-on patch. The print quality before and after the sensors installation is perfectly same as the sensors are very lightweight, and do not interfere with the printing. Figure 4 presents the implementation setup prepared for the case-study.

4.1.2 Data Accumulation. The implementation uses a Stemtera Arduino compatible breadboard. Interrupt routines are used for fast changing location data, while temperatures are polled periodically. The data is sampled at two different rates.

For the sampling rate, 'the higher the better' philosophy does not work well. Sophos uses sampled data to ascertain printer's state. To track slow movement of the filament, the sampling rate is lowered to ensure at least one pulse between two consecutive samples. On the other side, a too slow sampling rate may skip short critical moves. This conflict is resolved through dual sampling rates based on sensors' resolution, printing speed, and data acquisition module - as shown in Equation 8

$$S_{fast_i} = \{t_i, x_i, y_i\} \; ; \; S_{steady_i} = \{t_i, x_i, y_i, z_i, e_i, T_{n_i}, T_{b_i}\} \qquad (8)$$

Through experiments, we found 50 ms suitable for steady samples for our case study. The fast sampling rate pertaining to the nozzle's instantaneous location is dependent on Arduino's time to serve single interrupt, and the rate of total interrupts generated by the sensors at the peak printing speed. In our experiment, we found 5 ms to be optimal value for fast-samples, below which the increased noisy data outweighed the benefit of faster sampling.

S /No	Performance Parameters	Related Attack	Alert Thresholds
1	Single Mismatched Area	Geometry	1 mm^2
2	Cumulative mismatched area	Geometry	2% per layer, min dimension >0.2 mm
3	Nozzle Temperature Deviation	Thermodynamics	5^{o}C
4	Time window for sample search	Timing Profile	2 seconds
5	Samples Mismatch per Layer	Timing Profile	2%
6	Continuous Mismatch Duration	Timing Profile	500 ms
7	Max Layer Thickness Difference	Geometry	0.05 mm for 500 ms
8	Bed Temperature Difference	Thermodynamics	5^{o}C for 500 ms
9	Filament Consumption Deviation per move	Density, Geometry	5% diff per move
10	Filament Consumption Deviation per Layer	Density, Geometry	1% diff per layer
11	Max Nozzle Deviation	Geometry	0.75 mm ⊥ to move path
12	Max Vertex Deviation	Geometry	0.75 mm

Table 2: Sophos performance thresholds

4.2 Sophos Calibration

At the start of the experiment, Sophos is calibrated once to compensate for the sensing resolution, measurement, quantization and printing process errors. The overall effect is absorbed in the minimum detection values to ensure zero FP.

4.3 Finding Optimal Threshold Settings

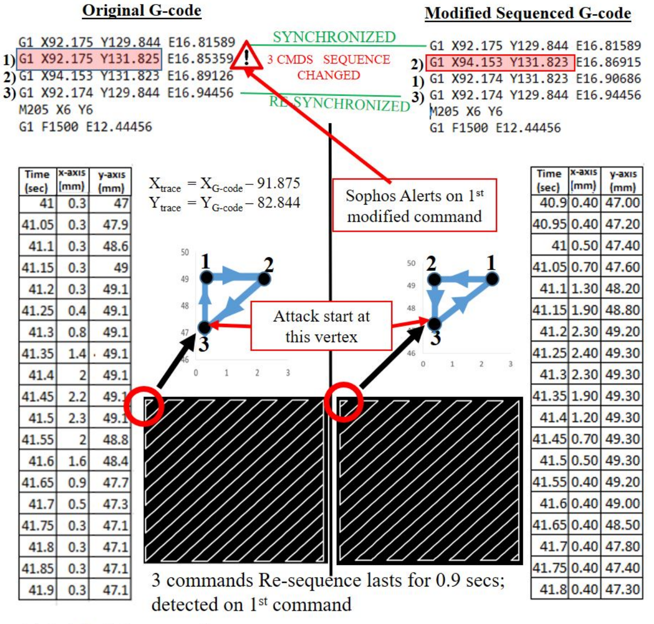
To find the optimal threshold values, we collected traces of 20 objects of different shapes (rectangular prisms, bars, letter E), number of layers, infill patterns and densities; and examined each parameter for a value that would result in zero FP for the training prints. The results are mentioned in Table 2. A brief account of the selection process is discussed in succeeding subsections.

4.3.1 Geometry Parameters. Sophos does not only match the outer geometry, it considers infill pattern as part of geometry, and matches each infill line for its location and dimension. Under this criteria, a simple rectangular prism becomes a complex geometry. We used 7 parameters for layer geometry: biggest & cumulative area mismatch; path & vertex deviation for G-code command, filament consumption difference per command and per layer, and max z-axis difference. As Sophos highlights the failure's reason, each parameter is individually tuned to a level where Sophos passes all benign prints.

Acronym	Attack Name	Attack Mechanism	Affected Domain
FS	Filament status	Switch off the extruder for 1 infill command	FK
FD	Filament density	Reduce the filament speed for 2 infill lines	FK
FC	Filament-kinetic based cavity	Retract the filament for 2 infill lines 2 mm long, Path change:Nil; Time Diff:0.4 sec	FK, NK minimal
TN	Nozzle temperature	Change the nozzle temperature by ±10^oC in central region of rectangular bar	THM
TB	Bed temperature	Change the bed temperature by 6^oC	THM
IFP	Infill pattern	Swap among the Line/Triangle/Grid or other infill patterns	FK, NK
IFD	Infill density	Change infill density from 20% to 21% ; 30% to 31% ; 40% to 41% (1% change at different density levels)	FK, NK
Z1	Z-axis 1 layer	Change layer thickness of a single layer by 0.1 mm	ZK
ZM	Z-axis multiple layers	Change thickness of three layers by 0.04 mm each	ZK
OG	Outer geometry	Change outer dimensions by 0.3 mm for a single axis	FK, NK
DC	Cavity via design file	Create a cavity of 1x1 mm, 2x2 mm, 3x2 mm through design file	FK, NK
PS	Path sequence	Modify 3 mov cmds, Δ t <1sec	FK, NK
TP1	Add/delete single move	Add 2 mm long, <1 sec cmd w/o extrusion	FK, NK
TP2	Add/delete 2 moves	Add 2 cmds of 2 mm and under 1sec duration w/o filament extrusion [A>B, B>A]	FK, NK
TPM	Add/delete multiple small moves	Insert multiple cmds <1 mm each, lasting for 2 secs	NK
V	Printing speed	Change printing speed to cause $\Delta t \geq 2$sec, w/o path change	FK, NK

FK:Filament-kinetics NK:Nozzle-kinteics THM: Thermodynamics

Table 3: Selected attacks for experiment



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

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

SYNCHRONIZED
3 CMDS SEQUENCE CHANGED
RE-SYNCHRONIZED
Sophos Alerts on 1st modified command
Attack start at this vertex
1
2
3
50
49
48
47
46
0
1
2
3
3 commands Re-sequence lasts for 0.9 secs; detected on 1st command

(a) Original Sequence Trace (b) Modified Sequence Trace

Figure 5: Commands re-sequencing attack trace

As part of the layer geometry integrity check, we verified the filament consumption on 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, it is not accumulated, and compensated in the next move. For longer moves, the error is less than 1% in all benign cases. We selected 1% as the alert threshold per layer and 5% for single move.

4.3.2 Thermodynamic Profile. Sophos nozzle temperature reading is found to be 2 - 3^oC less than the printer's internal sensor's reading. Sophos sensor is hosted just besides the nozzle tip and outside the nozzle, while internal sensor is slightly more distant from the extrusion point, and relatively closer to the heating element; thus showing slightly higher temperature. As the difference is consistent over our zone of interest, it is incorporated in the algorithm. We also observed that the actual nozzle temperature, following a hysteresis curve, may occasionally fluctuate by 2 - 3^oC. During the attack study, we conducted mechanical testing for temperature changes of less than 5^oC, and did not notice any significant effect on properties. Therefore, to incorporate this benign fluctuation and the approximations, we raise the alert threshold to 5^oC. Similarly, we observed a temperature difference of 1-2^oC for the printing bed. Many research studies on examining the impact of changing bed temperature on the object quality, have taken ΔT as 10^oC, 20^oC, or higher [20, 21], we find it reasonable to produce an alert at 5^oC deviation.

4.3.3 Timing Profile. There are 3 timing sources involved: 1^{st} is the G-code file's layer-end time, 2^{nd} is the result of Sophos modeling, and 3^{rd} one is the actual time taken for the printing. After estimating the time-profile, Sophos compensates for the difference with G-code provided layer-printing time by distributing the error uniformly. As we are operating on milliseconds scale, this approximation seems appropriate. However, we find out that the G-code layer-printing time does not match with the actual printing time. To compensate for this inaccuracy, we relaxed the time window to 2 seconds. Minimum duration for a mismatch to persist is 0.5 seconds, and minimum 2% mismatched samples per layer are considered as timing integrity breach.

The finalized set of parameters with zero FPR define the claimed detection performance of Sophos. Table 2 specifies the parameters, related attack types, and their final values that we use as the alert thresholds.

4.3.4 Alert Generation. The implementation ensures zero FP for each individual test by setting parameter alert thresholds beyond the confusion zone. Thus, Sophos considers a single parameter's breach as a violation, and uses logical OR operation among the six test categories as shown in Figure 1.

4.4 Process Visualization

Since Sophos works on bitmap files, it inherently provides a strong visualization feature precisely showing each layer's outlook at pixel's resolution. Sophos also points out the layer where the attack is detected alongwith the print-time. It also identifies which command in the G-code failed the integrity check. This information helps the user to examine the changes made by the attacker, and take remedial steps.

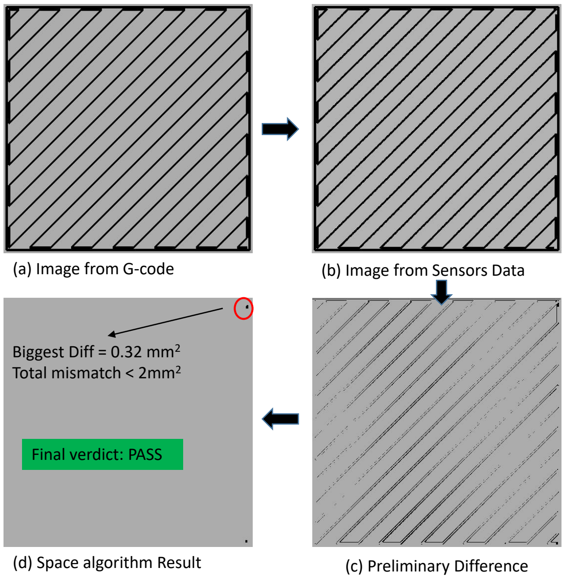

(a) Image from G-code
(b) Image from Sensors Data
Biggest Diff = 0.32 mm^2
Total mismatch < 2mm^2
Final verdict: PASS
(d) Space algorithm Result
(c) Preliminary Difference

Figure 6: Detection of benign deviations

5 SOPHOS EVALUATION FOR ATTACK DETECTION

5.1 Attacks Selection for Evaluation

To evaluate Sophos performance, we utilize the testbed and formulate a set of sixteen types of filament-kinetic, nozzle-kinetic, and dynamic-thermal attacks. The attacks are launched through modifying the G-code file (executing by the target printer), by assuming a reuse of any of the existing attack techniques [22, 23]. The nozzle-kinetics based attacks are the low-intensity variants of the existing attacks used by researchers. Less explored filament-kinetic attacks can create similar impact on the printed object without involving nozzle-kinetics. Dynamic-thermal attacks do not disturb filament or nozzle-kinetics. The attacks are performed on rectangular prisms (50 mm x 50 mm x 4 mm) and rectangular bars (60 mm x 40 mm x 4 mm). The layer height is set at 0.2 mm, printing speed at 50 mm/sec, and the infill-pattern used for attacks is "LINE" at 45^o raster angle, except for infill pattern attack. As every infill line is considered as a part of the geometry, the selected objects are not simple shapes. The way Sophos is designed, 45^o raster-angle caters for the highest error case for the pixel-approximation error (refer to Figure 2). Table 3 summarizes the attacks and the affected domains. To evaluate Sophos calibration for FP, we included the benign versions of these attacks as part of the experiment.

5.2 Attack Detection Results

During threshold-selection exercise for the calibration of Sophos, we crossed the confusion zone, and expect zero FP and FN for the test cases used in the evaluation. Sophos successfully detects all the attacks with zero FN, and all the benign counterparts with zero FP. Table 4 shows the consolidated attack detection performance, and the existing state of the art.

5.2.1 *Filament-kinetic Attacks (FS, FD, FC).*

Figures 6 and 7 presents Sophos performance in distinguishing similar looking benign deviations from geometry based attack. The total mismatched pixel in Figure 6 (c) are around 25% more than in Figure 7 (c). However, after passing through synchronization, contiguousness criteria, and parameter thresholds, Sophos correctly distinguishes between an attack and a small benign deviation. Figure 8 compares the actual images and the sophos images for the filament-kinetic cavity (FC) and filament-kinetic state (FS) attacks. For FS attack, there is no visible change. However, for FC attack, the cavity is visible when the attacked layer is being printed. Figure 8 (c) and (d) shows that Sophos detected the deviation in both cases.

5.2.2 *Dynamic-thermal Attacks (TN, TB).*

High and low temperature variations are conveniently detected at the first attacked layer. The heatmaps in Figure 9 generated from Sophos results show obvious deviations in the attacked samples temperature profile.

5.2.3 *Infill Pattern and Density Attacks (IFP, IFD).*

Change of infill pattern is invisible when the printing is completed. However, this attack is fairly obvious for Sophos, and raises multiple alerts on the first attacked layer. The affected parameters include the layer geometry, filament distribution, timing profile, and G-code commands.

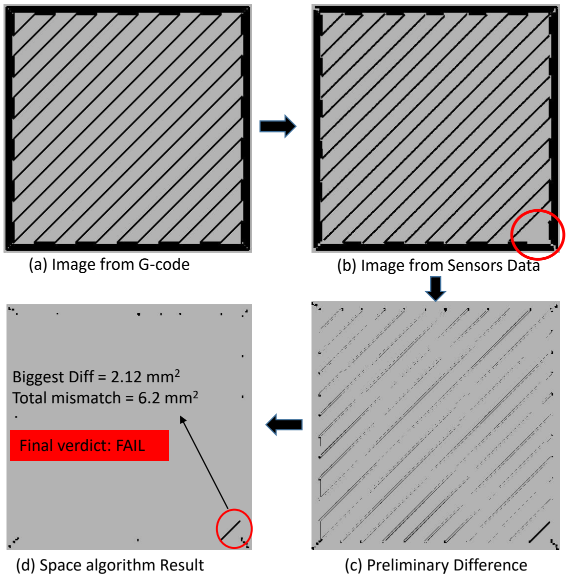

(a) Image from G-code
(b) Image from Sensors Data
Biggest Diff = 2.12 mm^2
Total mismatch = 6.2 mm^2
Final verdict: FAIL
(d) Space algorithm Result
(c) Preliminary Difference

Figure 7: Kinetic attack detection

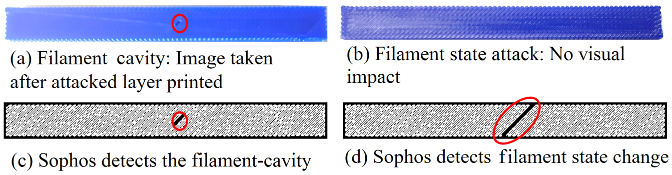

(a) Filament cavity: Image taken after attacked layer printed
(b) Filament state attack: No visual impact
(c) Sophos detects the filament-cavity
(d) Sophos detects filament state change

Figure 8: Filament-kinetic attack detection by Sophos

Attack	Sophos Performance	Existing State of the Art
FS	Detected, Min Duration ≥ 100ms; 1% Δe per layer OR 5% Δe per cmd	Not Detected
FD	Detected, 1% Δe per layer OR 5% Δe per cmd	Not Detected
FC	1% Δe per layer OR 5% Δe per cmd OR 1mm^2 cavity size	Not Detected
TN	Detected, attacks over 5^{o}C change	Not Detected
TB	Detected, attacks over 5^{o}C change	Not Detected
IFP	All pattern changes detected	Detected [5]
IFD	Detected; 1% infill density change	Demonstrated ±10% change [15], Deviation > 1sec [13]
Z1	Detected; Net deviation ≥ 0.05 mm	0.1 mm [17]; if over multiple layers [16]
ZM	Detected; Net deviation ≥ 0.05mm	0.1 mm [17]; if over multiple layers [16]
OG	Detected; 1mm^2 area OR 2% total mismatch OR 0.3 mm single axis change	Not addressed as a benchmark
DC	Detected; ≥ 1x1 mm	4 mm over one axis [17]
PS	Detected, if path deviation > 1 mm even if Δt < 1 sec	Δt ⩾ 2.26 sec [17]
TP1	Detected, if distance > 1 mm	1 sec duration (translated to >10 mm) [13]
TP2	Detected; if distance > 1mm	Not presented; we assume 1 sec (as above)
TPM	Detected, if cumulative time diff > 2sec OR path deviation > 1 mm	Not presented; we assume 1 sec (as above)
V	Detected, as and if Δt > 2 sec OR path deviation > 1 mm	Δv ≥ ±25mm/s [17], 0.8 sec [13]

Table 4: Sophos attack detection performance - viz-a-viz existing state of the art

5.2.4 *Layer Thickness Attacks (Z1, ZM).*

Sophos performance is tested over 2 different z-axis attacks. In first attack, a single layer's thickness is changed by 0.1 mm. The attack is immediately detected. In the second attack, the layer thickness is modified by 0.04 mm for 3 layers. This attack is detected at the 2nd attacked layer when the net height difference crosses the set threshold value. If the attack only continues for one layer, Sophos would miss it. In such case, the net difference in height of the object would remain under the detection threshold, and the effectiveness of such an attack may be questioned.

5.2.5 *Outer Geometry (OG).*

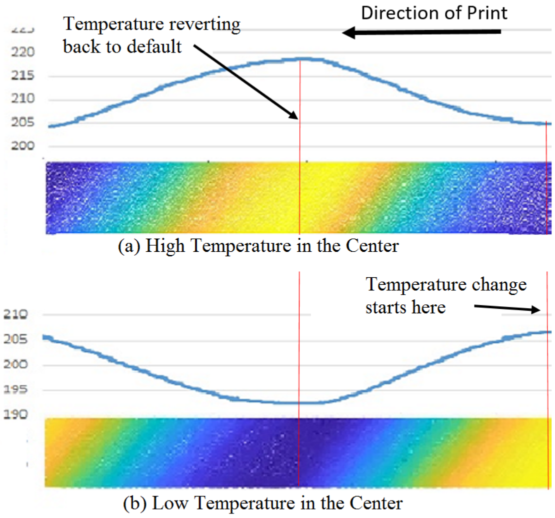
Unlike infill lines whose thickness is around the nozzle's diameter (0.4 mm for our case study), the outer geometry has bigger features. For the test samples, the outer geometry deviation of 0.3 mm over one axis is always detected. From area's perspective, 1mm^2 contiguous area deviation is always detectable by Sophos.

5.2.6 *Cavity Attack through Modified Design (DC).*

Some low profile attacks are found relatively easier to detect than expected. Modifying a cavity by 1 mm x 1 mm is supposedly a minimal change in design file, it actually causes changes in multiple G-code commands, and is easily and always detected by Sophos.

5.2.7 *Toolpath Attacks through G-code Commands (PS, TP1, TP2, TPM).*

These attacks manipulate G-code instructions to modify the kinetic properties of the object. The attacks are conducted by inserting / deleting / modifying existing G-code commands. The state of the art for modified sequence detectable duration is 2.6 seconds [13]. We propose a low profile version of re-sequencing attack that completes in less than 1 second. To re-order the command sequence without changing the geometry for a linear traversing printer, we require a minimum of 3 commands creating a triangle. The duration of the targeted commands before and after re-sequencing is around 0.9 seconds. The total time difference experienced between the 2 sequences is less than 100 ms. The attack is picked on the first modified command as shown in Figure 5. The original trace expected only y-axis movement, but the actual trace shows move in both x and y axes. Sophos detects the attack within 150 ms of its launch.



(a) High Temperature in the Center

(b) Low Temperature in the Center

Figure 9: Dynamic-thermal attacks trace

5.2.8 *Printing Speed Attack (V).*

During this attack, correct sequence of command is followed at a different speed. After the attack is initiated, the timing profile continuously deviates from the desired profile. If the attack sustains for 2 seconds, it is always detected by the timing profile check.

5.3 Process Visualization Information

Sophos precisely points out the location, time, and attack parameters in a 3D object. Recall that Figure 8 (c) and (d) identify the missing infill lines during a filament-kinetic attack. Figure 5 highlights the G-code instruction at which the re-sequencing attack is launched. For timing integrity attack, Sophos highlights the exact time and the sample where the integrity violation starts. In addition to attack visualization, Sophos helps in identifying unintended parameters settings errors that may cause the actual print to deviate from the actual design.

5.4 Comparison with Existing Approaches

Table 4 compares Sophos results with the existing techniques [5, 13, 15–17] which also rely on independent monitoring through external sensors. The detection thresholds of Sophos are much improved from the existing techniques. Sophos also detects thermodynamic and filament-kinetic attacks that are not covered by the existing approaches.

In case of timing profile attacks where path is not changed, Sophos reliably detects if the timing deviation is greater than 2 seconds. The reasons for the higher detection threshold for timing integrity are the inaccuracies in the "source of truth" and the "G-code transformation" modules. The estimated time mentioned by the slicer software in the G-code file is not accurate, and can have up to 5% error. Some users of the same product have reported higher errors. However, if an attacker tries to exploit the 2 seconds window and attempts to change the geometry by 1 mm or higher, the attack is immediately detected.

6 LIMITATIONS AND FUTURE WORK

In our implementation, we did not cover fan speed attack that influences thermodynamic properties. Existing work [5] does cover this attack vector, and can be accommodated. Detecting 1 mm deviation is a substantial improvement in the state of the art, but even sub-mm changes can be damaging for the final object. Ideally, the monitoring system's resolution should be higher or matching the monitored system's resolution. With Sophos, this task is feasible by merely upgrading the 'Data Acquisition' module with better sensors. Another improvement relates to the resilience of the 3D printing process. If the system detects a small deviation, should the printing be stopped? A minimal change can be harmless or damaging depending upon multiple factors, such as the magnitude, the location in the object, and the type of deviation. Real-time impact evaluation of low profile attacks will help in taking a rationale-based decision for aborting or continuing the printing job.

7 CONCLUSION

In this paper, we proposed Sophos, a fine-grained and modular integrity checking framework for FDM. Sophos utilizes ubiquitous and inexpensive sensors, and multi-domain analysis to detect inconspicuous attacks with much improved resolution than the state of the art. Sophos successfully detects the filament-kinetic, and thermodynamic attacks. Unlike many other approaches, Sophos does not require any prior learning on a per-object basis, making it suitable for Industry 4.0 for customized products instead of bulk production. Sophos detects the attacks at the same layer where they occur, thus saving precious time and printing resource. It also provides detailed information about the process status and the attacks.

ACKNOWLEDGEMENT

This work was, in part, supported by the Virginia Commonwealth Cyber Initiative (CCI).

REFERENCES

[1] Ugur M. Dilberoglu, Bahar Gharehpapagh, Ulas Yaman, and Melik Dolen. The role of additive manufacturing in the era of industry 4.0. *Procedia Manufacturing*, 11: 545 – 554, 2017. ISSN 2351-9789. doi: https://doi.org/10.1016/j.promfg.2017.07.148. URL http://www.sciencedirect.com/science/article/pii/S2351978917303529. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy.

[2] Statistica. Statistica, worldwide most used 3d printing technologies, as of july 2018, 2020. URL https://www.statista.com/statistics/756690/worldwide-most-used-3d-printing-technologies.

[3] Shahrain Mahmood, AJ Qureshi, Kheng Lim Goh, and Didier Talamona. Tensile strength of partially filled fff printed parts: experimental results. *Rapid prototyping journal*, 23(1):122–128, 2017. ISSN 1355-2546.

[4] Steven Eric Zeltmann, Nikhil Gupta, Nektarios Georgios Tsoutsos, Michail Maniatakos, Jeyavijayan Rajendran, and Ramesh Karri. Manufacturing and security challenges in 3d printing. *JOM*, 68(7):1872–1881, Jul 2016. ISSN 1543-1851. doi: 10.1007/s11837-016-1937-7. URL https://doi.org/10.1007/s11837-016-1937-7.

[5] Yang Gao, Borui Li, Wei Wang, Wenyao Xu, Chi Zhou, and Zhanpeng Jin. Watching and safeguarding your 3d printer: Online process monitoring against cyber-physical attacks. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3), September 2018. doi: 10.1145/3264918. URL https://doi.org/10.1145/3264918.

[6] Shouling Ding, Bin Zou, Peng Wang, and Hongjian Ding. Effects of nozzle temperature and building orientation on mechanical properties and microstructure of peek and pei printed by 3d-fdm. *Polymer Testing*, 78:105948, 2019. ISSN 0142-9418. doi: https://doi.org/10.1016/j.polymertesting.2019.105948. URL http://www.sciencedirect.com/science/article/pii/S0142941819304994.

[7] Martin Spoerk, Joamin Gonzalez-Gutierrez, Janak Sapkota, Stephan Schuschnigg, and Clemens Holzer. Effect of the printing bed temperature on the adhesion of parts produced by fused filament fabrication. *Plastics, Rubber and Composites*, 47(1):17–24, 2018. doi: 10.1080/14658011.2017.1399531. URL https://doi.org/10.1080/14658011.2017.1399531.

[8] Mark Yampolskiy, Lena Schutzle, Uday Vaidya, and Alec Yasinsac. Security challenges of additive manufacturing with metals and alloys. In Mason Rice and Sujeet Shenoi, editors, *Critical Infrastructure Protection IX*, pages 169–183, Cham, 2015. Springer International Publishing. ISBN 978-3-319-26567-4.

[9] Mark Yampolskiy, Wayne E. King, Jacob Gatlin, Sofia Belikovetsky, Adam Brown, Anthony Skjellum, and Yuval Elovici. Security of additive manufacturing: Attack taxonomy and survey. *Additive Manufacturing*, 21:431 – 457, 2018. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2018.03.015. URL http://www.sciencedirect.com/science/article/pii/S221486041730502X.

[10] Samuel Bennett Moore, William Bradley Glisson, and Mark Yampolskiy. Implications of malicious 3d printer firmware. In *Proceedings of Hawaii Int. Conf.Syst.Sci,2017*, pages 1–10, 2017. doi: 10.24251/HICSS.2017.735. URL http://hdl.handle.net/10125/41899.

[11] Tian-Ming Wang, Jun-Tong Xi, and Ye Jin. A model research for prototype warp deformation in the fdm process. *International Journal of Advanced Manufacturing Technology*, 33:1087–1096, 08 2007. doi: 10.1007/s00170-006-0556-9.

[12] 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)*, pages 1–8, 2016.

[13] S. Belikovetsky, Y. A. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici. Digital audio signature for 3d printing integrity. *IEEE Transactions on Information Forensics and Security*, 14(5):1127–1141, 2019.

[14] Mingtao Wu, Heguang Zhou, Longwang Lin, Bruno Silva, Zhengyi Song, Jackie Cheung, and Young Moon. Detecting attacks in cybermanufacturing systems: Additive manufacturing example. *MATEC Web of Conferences*, 108:06005, 01 2017. doi: 10.1051/matecconf/201710806005.

[15] Christian Bayens, Tuan Le, Luis Garcia, Raheem Beyah, Mehdi Javanmard, and Saman Zonouz. 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)*, pages 1181–1198, Vancouver, BC, August 2017. USENIX Association. ISBN 978-1-931971-40-9. URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/bayens.

[16] J. Gatlin, S. Belikovetsky, S. B. Moore, Y. Solewicz, Y. Elovici, and M. Yampolskiy. Detecting sabotage attacks in additive manufacturing using actuator power signatures. *IEEE Access*, 7:133421–133432, 2019.

[17] S. Yu, A. V. Malawade, S. R. Chhetri, and M. A. Al Faruque. Sabotage attack detection for additive manufacturing systems. *IEEE Access*, 8:27218–27231, 2020.

[18] Sofia Belikovetsky, Mark Yampolskiy, Jinghui Toh, Jacob Gatlin, and Yuval Elovici. dr0wned – cyber-physical attack with additive manufacturing. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC, August 2017. USENIX Association. URL https://www.usenix.org/conference/woot17/workshop-program/presentation/belikovetsky.

[19] Guanxiong Miao, Sheng-Jen Hsieh, J. Segura, and Jia-Chang Wang. Cyber-physical system for thermal stress prevention in 3d printing process. *The International Journal of Advanced Manufacturing Technology*, 100, 01 2019. doi: 10.1007/s00170-018-2667-5.

[20] Young Choi, Cheol-Min Kim, Hwan-Seock Jeong, and Jeong-Ho Youn. Influence of bed temperature on heat shrinkage shape error in fdm additive manufacturing of the abs-engineering plastic. *World Journal of Engineering and Technology*, 04: 186–192, 01 2016. doi: 10.4236/wjet.2016.43D022.

[21] Nahal Aliheidari, Rajasekhar Tripuraneni, Cameron Hohimer, Josef Christ, Amir Ameli, and Siva Nadimpalli. The impact of nozzle and bed temperatures on the fracture resistance of FDM printed materials. In Nakhiah C. Goulbourne, editor, *Behavior and Mechanics of Multifunctional Materials and Composites 2017*, volume 10165, pages 222 – 230. International Society for Optics and Photonics, SPIE, 2017. doi: 10.1117/12.2260105. URL https://doi.org/10.1117/12.2260105.

[22] Q. Do, B. Martini, and K. R. Choo. A data exfiltration and remote exploitation attack on consumer 3d printers. *IEEE Transactions on Information Forensics and Security*, 11(10):2174–2186, 2016.

[23] Logan D. Sturm, Christopher B. Williams, Jamie A. Camelio, Jules White, and Robert Parker. Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the .stl file with human subjects. *Journal of Manufacturing Systems*, 44:154 – 164, 2017. ISSN 0278-6125. doi: https://doi.org/10.1016/j.jmsy.2017.05.007. URL http://www.sciencedirect.com/science/article/pii/S0278612517300961.