Encryption is Futile: Reconstructing 3D-Printed Models Using the Power Side-Channel

Jacob Gatlin
Auburn University
USA

Sofia Belikovetsky
Johns Hopkins University
USA

Yuval Elovici
Ben-Gurion University of the Negev
Israel

Anthony Skjellum
University of Tennessee at Chattanooga
USA

Joshua Lubell
Paul Witherell
National Institute of Standards and Technology
USA

Mark Yampolskiy
Auburn University
USA

ABSTRACT

Outsourced Additive Manufacturing (AM) exposes sensitive design data to external malicious actors. Even with end-to-end encryption between the design owner and 3D-printer, side-channel attacks can be used to bypass cyber-security measures and obtain the underlying design. In this paper, we develop a method based on the power side-channel that enables accurate design reconstruction in the face of full encryption measures without any prior knowledge of the design. Our evaluation on a Fused Deposition Modeling (FDM) 3D Printer has shown 99 % accuracy in reconstruction, a significant improvement on the state of the art. This approach demonstrates the futility of pure cyber-security measures applied to Additive Manufacturing.

CCS CONCEPTS

• **Applied computing** → **Computer-aided manufacturing**; • **Security and privacy** → **Side-channel analysis and countermeasures**; • **Social and professional topics** → **Intellectual property**; *Digital rights management*.

KEYWORDS

Additive Manufacturing, 3D Printing, Side-Channel Attack, Intellectual Property Theft, IP Theft.

ACM Reference Format:
Jacob Gatlin, Sofia Belikovetsky, Yuval Elovici, Anthony Skjellum, Joshua Lubell, Paul Witherell, and Mark Yampolskiy. 2021. Encryption is Futile: Reconstructing 3D-Printed Models Using the Power Side-Channel. In *24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '21), October 6–8, 2021, San Sebastian, Spain.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3471621.3471850

1 INTRODUCTION

Additive Manufacturing (AM) has proven vulnerable to a variety of attacks. Given that this is a high-growth and valuable industry [44], cyber-security defenses are regularly considered, especially in the private sector. However, in the case of outsourced manufacturing, cross-domain (physical-to-cyber [47]) attacks can circumvent pure cyber-security measures. In this paper, we develop such a side-channel attack to steal the design data of a manufactured object.

Additive Manufacturing provides significant advantages over traditional manufacturing methods. For instance, General Electric[1] has claimed it provides production flexibility, reduced weight, complex geometries, and graded materials [15]. Smaller manufacturers can also benefit, but are often prohibited by the high capital expenditure and specialized operational knowledge. Outsourcing business models have flourished to address this need; world leaders include ThyssenKrupp AG TechCenter Additive Manufacturing (Germany) [41], Akhani3D (South Africa) [3], RapidDirect (China) [32], and Treatstock (United States) [42]. While the above manufacturers are reputable, not all manufacturers will be trustworthy or safe from insider threats. A malicious AM service provider is in a unique position to steal technical data.

To address this threat, systems have been proposed offering Digital Rights Management (DRM) and adapting traditional cyber-security for AM environments. In a recent patent, Oligshclaeger et al. [30] propose to use cryptographic keys unique to a 3D printer for end-to-end encryption of designs. In a similar proposal by GE [19], technical data is decrypted by a Trusted Platform Module (TPM) (a dedicated security chip) integrated in the 3D printer, limiting access to *plaintext* (i.e., decrypted) data and cryptographic keys. Several companies, such as Identify3D [24], offer commercial options that implement DRM and encryption.

Identify3D in particular is addressing the distributed and outsourced manufacturing market: their technology suite is claimed to provide "intellectual property protection, manufacturing repeatability, and traceability" to "unlock the potential of distributed manufacturing" [25]. Taken together, we believe these individual developments constitute a trend towards securing outsourced manufacturing via DRM, which will further boost the already growing

[1]Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the authors or their organisations, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

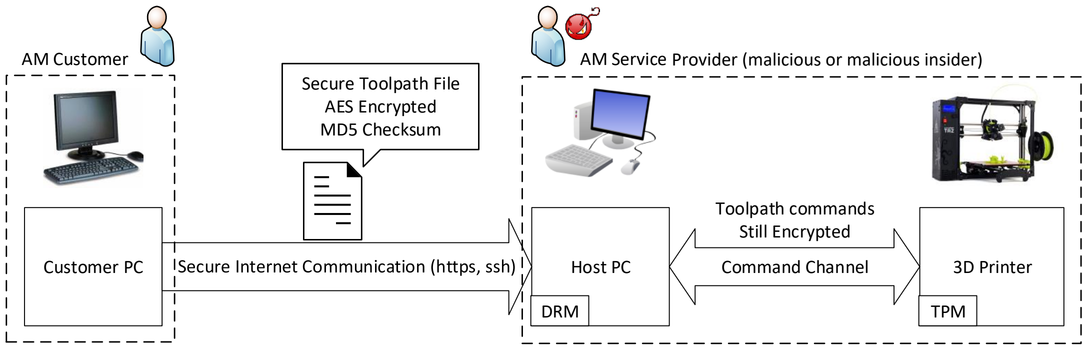


Figure 1: DRM-Protected Outsourcing to a Malicious AM Service Provider.

sector. We wish to examine a case where this technology is not only available, but has achieved its apogee; we assume the cyber-security implementation is comprehensive and faultless. Even in this case, we seek demonstrate that pure cyber-security measures are not sufficient to protect IP against a malicious contract manufacturer.

A number of publications have explored offensive or defensive reconstructions of AM IP using side-channels [11, 17, 22, 37]. The side-channels utilized in these works include the acoustic, electromagnetic (EM), visual, and inertial readings; these data streams have been used individually and in combination, through reconstructive approaches based on data mining, machine learning with or without human assistance, and control theory. We develop a novel approach to reconstruction utilizing a side-channel not yet used for this purpose, the actuator current draw. This side channel has numerous advantages compared to those previously explored: the potential for perfect or near-perfect reconstruction, excellent portability to all classes of actuators, and a level of stealth that allows for multiple viable threat models.

Threat Model: In this paper, we assume a Man-at-the-End (MATE) threat model applied to outsourced AM. Comprehensive digital security measures are assumed to be in place to protect the customer's data, such as end-to-end encryption of the model and a TPM module on the 3D printer (see Figure 1). We assume that either the service provider itself is malicious, or has a malicious insider employee with some access to the AM equipment. The intent of the malicious party is to reconstruct the customer's 3D printed model. We assume that they are unable to compromise the cyber-security measures, but are able to instrument the 3D printer at will. Potential implementations of the sensor suite are stealthy enough that a single insider could instrument a 3D printer and capture the side-channel data. This permits the attack against any printer an insider has even temporary physical access to. Under this threat model, we develop an approach to capture and reconstruct the 3D printed model that bypasses cyber-security measures entirely.

Overall, we contribute a significant evolution of the state of the art in reconstruction, a novel approach to collecting and interpreting AM actuator signals, and a demonstration of MATE attacks in AM circumventing defensive systems.

This paper is structured as follows: Section 2 reviews the literature. Section 3 outlines our approach and the challenges involved. The approach is presented in pseudocode in Section 4. Section 5 explains our experimental evaluation and presents our results. We discuss the limitations of the approach in Section 6, and further topics in Section 7. Section 8 concludes the paper.

2 STATE OF THE ART

Multiple authors have surveyed the field of AM security [27, 31, 48]. The theft of technical data emerged as a topic early in the field's development. It was mentioned in 2014 by Yampolskiy et al. [45, 46], with practical attacks published in 2016 by Do et al. [14] and Belikovetsky et al. [9] using cyber means to access design files. The majority of AM sabotage attacks, such as Sturm et al. [40], Belikovetsky et al. [9], and Zeltmann et al. [50], acquire and modify the design files, implying a data theft attack as a prerequisite [20].

A variety of cyber-security solutions have been proposed to respond to the threat of data theft in AM. Adkins et al.'s patent [2] for an approach to securely delivering design files describes a system applying encryption on a trusted device, and practices for maintaining trust through the print process. Safford et al. [35] propose a similar system with hardware based on TPM technology. Such systems, robust as they might be to cyber attacks, can often be circumvented with side-channel attacks.

Side-channels are a common vulnerability of embedded systems. They are the physical emanations of Cyber-Physical Systems (CPS), such as electromagnetics or sound, which can be correlated to the original processes. Classical approaches include Van Eck's remote spying on Cathode Ray Tube displays by their emanations [43], and Koscher's Simple and Differential Power Analysis for identifying encryption algorithms and cryptographic keys [26].

The first practical side-channel exploitation in AM was conducted by Al Faruque et al. [4] using acoustic emanations from the motors of a Fused Deposition Modeling (FDM) 3D Printer. The authors identified that motor noise could be correlated with operating speed, direction, and the axis of movement. Using a machine-learning approach, they reconstructed motor movements for the printing of a basic polygon. They achieved 78.35 % accuracy of axis prediction, on average, and had a 17.82 % error in movement distance estimation.

Shortly afterwards, Hojjati et al. [22] and Song et al. [37] presented machine-learning-based reconstructions using the acoustic side-channel, relying on smartphone audio sensors. Hojjati et al. used a library of recorded printhead motions in addition to magnetic side-channel info and a human-managed error correction process. They report accuracy in terms of end gap lengths for segments (ranging 0.5-4.2mm) and angle deviations (ranging 0-3 degrees). Song et al. report an average Mean Tendency Error of 5.87 % in a similar model to that of Al-Faruque et al. [4], but raise concern over whether their metrics were meaningful. Both papers test only on perimeters with no infill.

Gao et al. [17] made significant strides in reconstruction accuracy and metrics in their 2018 work on process monitoring. Using a multi-sensor suite of accelerometer, magnetometer, and camera, the authors reconstruct the physical properties of toolpath, extrusion speed, layer thickness, and fan speed. Each of the physical properties required its own reconstruction approach, and the authors deploy different metrics based on their own judgement of appropriateness. The most noteworthy is the use of the Hausdorff Distance, a distance measure that reports the deviation between two sampled curves, which the authors applied on a per-layer basis. For their Kalmann-filter-based reconstruction of the toolpath, the authors achieved error ranging from 0.735 mm to 0.869 mm. The experiment was conducted on full printed models, with infill percentages ranging from 5 to 20 %; lower infill percentages corresponded to lower Hausdorff Distances.

In addition and in response to published side-channel attacks, some authors have proposed methods either to defend AM systems with side-channel techniques, or else to defend against them. A number of side-channel papers in the field utilize them to detect sabotage attacks or print errors [6–8, 11, 18, 39], using much the same approach as a data-theft attack. Recently, a collaboration of Yu, Chhetri, and Al-Faruque [49] made significant improvements to attack detection accuracy employing a multi-modal approach, examining many identified side-channels at once and correlating their information. Work by Chhetri et al. discusses modifications to toolpath generation either to increase [10, 33] or decrease [12, 33] the side-channel information that could be used in reconstruction. Al-Faruque et al. acquired a 2019 patent [5] on defending side-channel attacks with their mutual information reduction strategy, implemented in the toolpath-generation stages of the AM process. Gatlin et al. [18] utilized the power side-channel for detecting sabotage attacks, but did not attempt reconstruction, instead using a Principal Component Analysis–based signature.

3 CHALLENGES OF RECONSTRUCTION

In this work, we focus on 3D printers employing the FDM process, which is widely used in both desktop and high-end polymer 3D printers. The process is based on the extrusion of heated polymer filament through a nozzle, built into a print head which can be moved along the X-, Y-, and Z-axes, and extrude or retract filament along the E-axis. Both printhead movement and filament extrusion are driven by motors, controlled by an on-board motor controller.

The bipolar hybrid stepper motor is a commonly used positional actuator in FDM printers; our test printer, the Lulzbot Taz 6 (See Figure 2), uses these motors exclusively. Each motor has two phases and each phase is connected by two wires, forming an electrical loop delivering current from the controller. The current is delivered in a sinusoidal pattern, making it Alternating Current (AC). Both the positive and negative peaks of the current on either phase cause the motor to advance or reverse depending on the firing sequence.

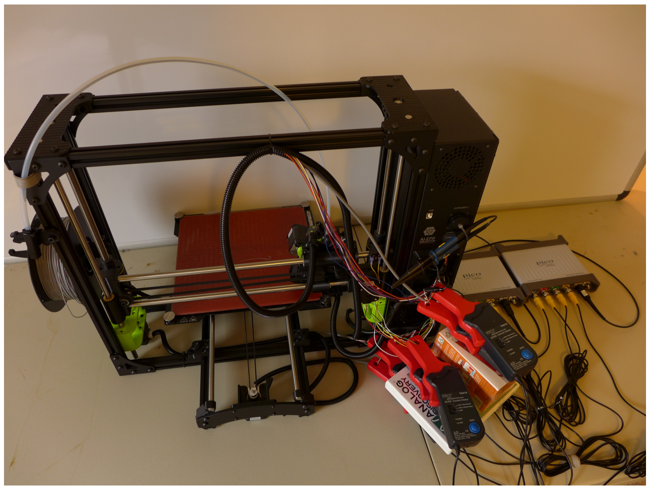


Figure 2: A Lulzbot Taz 6 printer, instrumented by Picoscope 5444D oscilloscopes. The probes are Picoscope's 60A Inductive Current probes. Each motor has two clamps attached, one for each phase. The fan controller is also instrumented by a standard voltage probe. The data captured here is transmitted to a host PC running the PicoScope application.

The peaks of the current directly cause the motor to move, and the mechanical construction of the motor ensures that positional error does not accumulate [29]. A single peak on one phase will predictably move the motor from one step position to another. The direction of movement is determined by the sequence in which the two phases are fired. The speed of movement is determined by the speed of the firing sequence.

The *toolpath*, a sequence of movement and printing commands, is issued in the *G-code* format, a legacy format commonly used in desktop 3D printers. The toolpath is interpreted by the printer and translated into a series of motor movements. The motor controller generates current in the correct pattern to achieve these movements. The particular properties of this current and recognizing the patterns necessary for executing G-code commands present several challenges to the reconstruction process, described in the remainder of this section.

We adapt our instrumentation strategy from the work of Gatlin et al. on power side-channel signatures [18], using inductive current clamps to monitor motor phases[2]. While that work instrumented a single phase of each motor, we instrument both to capture the full motor behavior. Our implementation of the sensor suite is obvious and constructed from laboratory-grade equipment, but the technical requirements (sampling in the 2 KHz range on 8 channels) can be easily achieved with a more surreptitious and smaller setup based around a microcontroller and less precise current probes. Such an

[2]While in-line ammeters or pullup resistors can also capture this data, they have notable downsides compared to inductive clamps: they can interfere with actuation signals, are difficult to remove for in-person inspections, and might be detectable by anti-tamper impedance testing.

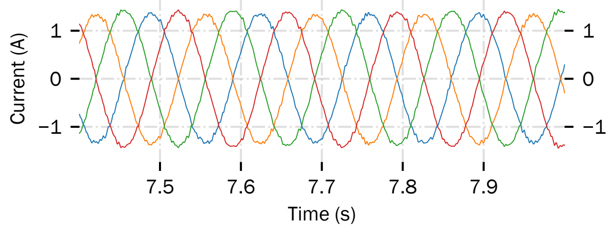


Figure 3: Constant-speed motor trace, after applying a low-pass filter. This plot is generated in code from our processed data.

implementation could be hidden within the enclosure of even a desktop 3D printer.

3.1 Unidirectional Movement

The simplest section of oscilloscope trace to interpret is unidirectional movement at a constant speed. This will occur in the middle of any linear movement command. Observing the current at this point will produce a trace as shown in Figure 3. In this figure, and in all other trace figures, we plot a positive and negative version of the current on each phase, producing a total of four. The firing sequence of the phases can then be seen just by looking at positive peaks; this representation is also convenient when processing the data in Section 4.

Peaks have two independent characteristics by which they can be recognized: *height* and *prominence*. *Height* is the absolute value of the peak at its highest point. In Figure 3, the peaks have a consistent height of approximately 1.5 A; in other situations, the height can vary dramatically. The *prominence* of the peak can be simply described as the height of the peak over the adjacent minima. This corresponds to the amplitude of a signal in electrical terms. The prominence of the peaks in Figure 3 is consistently around 3 A. In other situations, the prominence of a given peak can be significantly less. Interpreting each peak as a single step, we can precisely track position. From there we can use the known property of step size, or distance per motor step, to determine both printhead and filament position at any time, and the linear speed of movements.

The phases are all operating at the same constant frequency (corresponding to a constant movement speed) and firing in the same order. It must be noted that, at the beginning and end of a linear movement, the motor controller ramps up and ramps down the frequency, respectively. This produces acceleration and deceleration in the movement, and the artifacts from it in the trace should be accounted for in reconstruction.

3.2 Start, Stop, and Dwell

The trace exhibits specific behavior at the beginning and end of movements. This behavior includes apparent changes in frequency and absolute value. This occurs at the beginning and end of every print, as shown in Figure 4. This also occurs during Dwell commands, which halt movement on every axis for a set period, as shown in Figure 5.

Both figures illustrate a challenge associated with these transitional periods: changing DC offsets. The height of the first peak

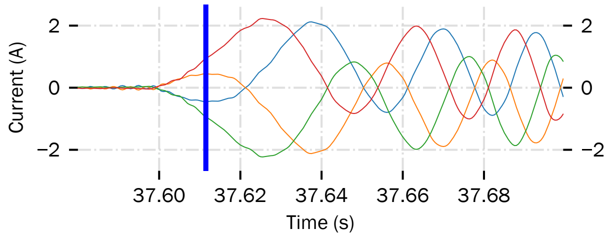


Figure 4: Beginning of motor movement after inactivity. The DC offset of the phases start at nearly 2A, then converges over time.

after a period of inactivity is visibly lower, and the second peak is visibly higher, than their normal range. Over time, they gradually stabilize at the same level, as in the linear movement of Figure 3. In electrical terms, this difference in the absolute height of an AC signal is referred to as its Direct Current (DC) offset. Notably, the prominence of the peaks does not vary dramatically. A change in DC offset over time means that height cannot be relied upon to distinguish peaks.

For reconstruction, it is helpful to identify periods of inactivity, as opposed to normal inter-peak periods. On the Z- and E-axis motors, there are long periods of inactivity while the print head repositions along X and Y. Identifying these inactive periods helps filter out false positive peaks.

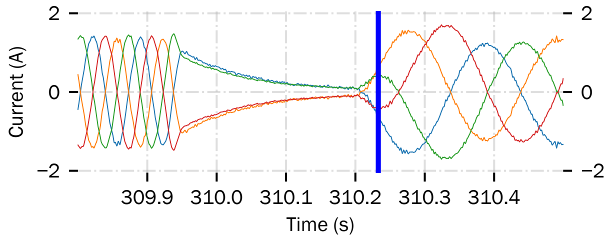


Figure 5: Dwell shown in the trace. Note the much lower height of the first peak at 310.25s. A change in speed is visible from the first section of activity to the second.

To aid in this filtering, we can observe the change of inter-peak intervals while transitioning into and out of inactivity. Before a period of inactivity, there is a stable inter-peak interval corresponding to constant motion. There will then be a single larger interval between the last peak before and the first peak after the inactive section. After that, the next movement command begins and, after a short period of time, shows stable inter-peak intervals once more.

3.3 Reversal of Direction

When a sequence of G-code commands reverses the direction of movement along an axis, we observe the behavior seen in Figure 6. The characteristic feature of a reversal is that it changes the firing order of the phases. They can be detected by testing for this change, but several artifacts of the trace can complicate the process. In the figures showing reversals, we mark the first peak in the new direction with a vertical blue line.

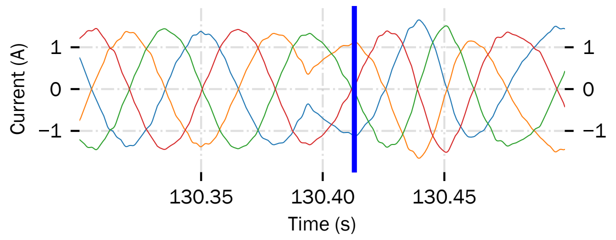


Figure 6: Valid reversal shown in the trace. After the central green peak, 130.4s, the direction of travel is reversed. The blue line marks the first peak after the reversal. This reversal presents no additional problems for reconstruction.

The DC offset of the trace can change during reversals to the point that it impacts peak recognition. In Figure 7, peaks in the vicinity of the reversals have both a low absolute value and a low prominence, but are still valid peaks (i.e., shift the motor position). This can occur not only to the reversal peak, but also to any peak before or after. Each potential missing peak represents a uniquely malformed firing order, when in reality only well-behaved reversals actually occur.

3.4 High Frequency Noise

A consistent property of the raw trace is the presence of high-frequency noise. In most cases, the signal-to-noise ratio is high, and trace properties measured across longer periods of time are unaffected. More precise properties, such as locating a peak, are affected. To locate peaks better and more consistently, we preprocess the raw trace with a low-pass filter[3], a common technique in signal processing. All trace figures are presented after filtering.

Applying a low-pass filter to these current traces creates a problem of its own. The frequency of the current and the speed of the motor are directly proportional, and the valid range of speeds overlaps in part with the frequencies of noise. Thus, we must restrict the cutoff frequency of our low-pass filter to above this range, and compensate for the remaining noise in other ways.

3.5 Trace Synchronization

The full printer system is composed of multiple motors acting together, and it is necessary to maintain synchronization when capturing their current traces. Done poorly, this can cause issues with misordered motor movements across different axes. These misalignments can distort the reconstructed shape significantly.

While it is possible to synchronize the beginning of multiple traces captured individually, the duration of several commands is not always fixed, which leads to desynchronization. These include motion commands such as Home (G28), and thermal commands that use a feedback loop to reach a specific target temperature.

To provide reliable synchronization throughout the duration of a print, we capture all motor signals simultaneously in a single print run. With our selected equipment, this requires multiple oscilloscopes. We chose to synchronize to a consistent movement pattern

[3] A low-pass filter is a signal processing filtering technique that dampens the power of a signal above a cutoff frequency. There are multiple ways to apply low-pass filters; in this work, we use an 8th-order Butterworth filter.

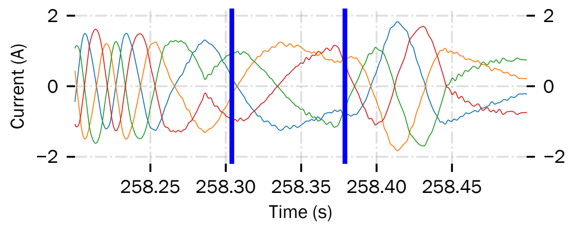


Figure 7: Reversal with multiple issues complicating reconstruction. Peaks from 258.25s to 258.4s are low-prominence, of varying width, and represent two reversals in rapid succession.

in the preamble, a printer-specific block of G-code that occurs before each print, to begin oscilloscope capture. The captures could also be synchronized with a manual signal to each oscilloscope, by chaining the oscilloscope triggers, or using a single oscilloscope with more channels.

4 RECONSTRUCTION APPROACH

At a high level, the reconstruction process consists of several stages. First, we preprocess the traces to filter noise. Then, we identify basic features of the trace such as peaks and periods of inactivity. Next, we attempt to map the features onto motor behavior, such as reversals. Because of the difficulties explained in Section 3.3, this stage generates a number of errors. The next stage applies a heuristic approach to correct these errors. Once a corrected sequence of features is finalized, they are used to track the position of the print head over time. During this process we distinguish between extruding and non-extruding moves to produce a point cloud corresponding to the printed figure.

We implemented our reconstruction approach in Python 3.2; we summarize the essential operation of the algorithm in pseudocode. Where our code relies on an external library for non-trivial functionality, we provide specific reference to the library and call.

4.1 Loading Oscilloscope Data

Our reconstruction algorithm operates on the synchronized traces of the current delivered to each phase of each motor on the printer. We refer to the individual motor traces as the X, Y, Z, and E traces. Our oscilloscopes capture the full waveforms corresponding to each axis; upon print completion, these readings are exported as a CSV file.

Our algorithm receives CSV files of measured current values and corresponding timestamps, with the structure in Figure 8. The negative timestamps correspond to the values captured by the oscilloscopes before the trigger signal, at time 0s.

```
Time(s)|Channel A(A)|Channel B|Channel C|Channel D

-2.00050006,0.01537894,0.03644808,0.02383735,0.02752830
-2.00000006,0.01968504,0.03875492,0.02552903,0.02921998
-1.99950006,0.02399114,0.03875492,0.02614419,0.02552903
...
```

Figure 8: Trace Capture in CSV Format (Excerpt).

Entries from the CSV files are read into TraceEntry data structures. The timestamp from the CSV file is read directly into the *time* field. The sample values of each motor's two phases are saved in *phase0* and *phase2*; the inverted values (negated) are saved in *phase1* and *phase3*, respectively. This duplication simplifies the peak recognition process in a later stage.

In addition to the timestamp and values, each entry is associated with an *axis* marker and a set of flags corresponding to motor behaviors. These flags are initially cleared; later stages of the algorithm will set them as appropriate. Descriptions of the flags are provided in the relevant sections.

The individual *traceEntry* elements are stored in sequential data structures, one for each axis. As the sizes of the datasets range from hundreds of thousands to hundreds of millions of these entries, we use the NumPy library's Array structure rather than standard Python datatypes.

4.2 Peak Detection

Later stages of our algorithm operate primarily on peaks and their timestamps, so we must first recognize the peaks from the trace.

The raw data has significant high-frequency noise. We first apply a low-pass filter to remove this noise, using the SciPy *butter* filter. The filter is applied with cutoff frequencies of 300 Hertz for the X-, Y-, and Z-axes, and 275Hz for the E-axis. Our process for determining these values is discussed in greater detail in Section 7.

After applying the filter we use the *find_peaks* function from the SciPy Signals package [36] to identify the peaks, marking them by setting the *isPeak* flag in the corresponding *traceEntry*. The parameters of this function, *height* and *prominence*, are applied differently per axis:

```
X:  height=0.4  prominence=0.3
Y:  height=0.4  prominence=0.3
Z:  height=0.1  prominence=0.125
E:  height=0.1  prominence=0.2
```

The selection criteria for these are also discussed in Section 7.

There are several errors that we observed in flagging peaks that must be handled before further processing. While peaks should not occur simultaneously, noise and level-shifting behavior means that there are sometimes simultaneous peaks on both phases of an axis. Even peaks that are only in very close proximity can cause issues. The first case is handled while applying peak flags from *find_peaks* to the trace; if two phases attempt to flag the same timestamp as a peak, the phase with a higher peak is preserved. The proximity case is handled after all peaks have been flagged. If a peak occurs on a phase that is lower than another phase at the time, the algorithm checks if it is within a time threshold of the nearest peak- either within half of the average period between nearby peaks or a fixed threshold of 0.025s. If both hold true, then the low peak is removed.

4.3 Reversals and Error Correction

The largest issue to overcome in this algorithm is handling malformed segments of the trace. Because of the motors' internal structure, there are only two valid firing orders that they can perform, and we can only reconstruct motion that exhibits this behavior (see Section 3 for details). The firing orders are associated with forward and backward movement, and the motor controller switches between them to execute a reversal. Figure 9 illustrates this behavior.

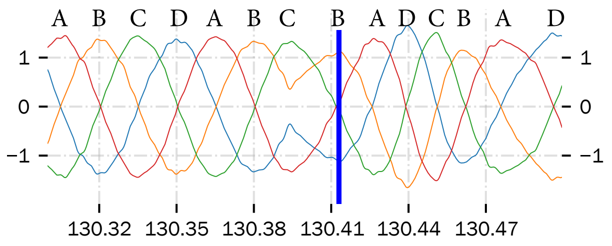


Figure 9: A reversal captured in the trace. Peaks are annotated A, B, C, or D to indicate the firing order. Note how the order reverses at 130.41s.

As a result of the difficulties discussed in Section 3, the trace contains both false-negative and false-positive peaks. Our observation shows that many of these occur around reversals or dwells, or during periods of inactivity. Either type of invalid peak is incredibly likely to create an invalid firing order[4]. Until these are corrected, it is not possible to accurately reconstruct motion. Further, if a section is recognized as correct but not accurate to the actual motor behavior, the reconstructed motion will drift: our estimate of the printhead position will accumulate error based on the difference between reconstruction and reality.

We attempt to correct these errors by first segmenting the trace into sections with valid firing orders and sections with invalid firing orders. The valid sections are processed according to the normal logic of motor operation. For invalid sections, contextual information is gathered and saved alongside the section in a list. This list is processed by a heuristic solver, which attempts to find the best weighted reconstruction for that sections by adding and deleting peaks. The best solutions are applied to the trace.

4.4 Segmentation into Good and Bad Sections

With all peaks recognized, our approach next identifies the firing order on each axis. The initial synchronization of our traces occurs on a long, linear, multiaxial move immediately before the print. This ensures there is a long enough section of consistent movement on every axis to recognize the firing order dynamically.

The pseudocode in Figure 10 details the process of identifying the firing order. Identifying the firing order from the data, rather than using a hard-coded order makes the algorithmic approach independent of the physical instrumentation, such as which wire is clamped by which probe. The firing order for an axis is "locked in" after the first 16 consistent peaks have been detected. This firing order is maintained for the remainder of the print, although it may be inverted when reversals occur.

Next, the algorithm scans across the list of peaks, comparing each in sequence to the firing order. It recognizes three cases: valid forward motion, valid reverse motion, and invalid firing orders. The pseudocode in Figure 11 outlines the process. As in the trace diagram of Figure 9, if the last peak to fire was, for example, B, then the next peak can be valid if it is on phases A or C. One phase represents forward motion, the other represents a reversal,

[4]Peak errors, much like parity error checking, always produce firing order errors in sections with only one bad peak. However, sections with two consecutive bad peaks can produce valid firing orders that are nonetheless not accurate to motor movements.

```
axisPeaks = getPeaks(axis)

// Initialize firing order buffer
int FOBuffer[4] = axisPeaks[0:3]

predictionCount = 0
while predictionCount < 16:
  predictionIndex = nextPeakIndex % 4

  if axisPeaks[nextPeakIndex] == FOBuffer[predictionIndex]:
    predictionCount++
  else: // capture new firing order buffer
    FOBuffer = axisPeaks[nextPeakIndex-3:nextPeakIndex]
    predictionCount = 0

  nextPeakIndex++
```

Figure 10: Identifying Firing Order.

```
badOrderSections = []

while peakIndex < axisLength:

  nextPeak = getNextPeak(peakIndex)

  if nextPeak == predictForward(FOBuffer, peakIndex):

    if badSection != NULL:
      goodPeaksCounter++
      if goodPeaksCount > goodPeaksNeeded:
        // Close out bad section.
        endBound = FindEndOfBadSection()
        SaveBadSectionInfo(originalFOBuffer, mustReverse,
                           startBound, endBound)
        badOrderSections.append(badSection)
        badSection = NULL

  elif nextPeak == predictReversal(FOBuffer, peakIndex):
    reverseBuffer(FOBuffer)

    if badSection == NULL:
      flagReversal(peakIndex)
    else:
      // Toggle the mustReverse flag
      mustReverse = !mustReverse
      // Good reversals also increment the counter
      goodPeaksCounter++
      // Accumulated enough peaks to end bad section?
      if goodPeaksCount > goodPeaksNeeded:
      // Close out bad section.
        endBound = FindEndOfBadSection()
        SaveBadSectionInfo(originalFOBuffer, mustReverse,
                           startBound, endBound)
        badOrderSections.append(badSection)
        badSection = NULL

  else:
  // Found a bad peak. Begin looking for boundaries.
    if badSection != NULL:
      badSection = createNewBadSection(peakIndex)

    // Any bad peak resets the counter and FOBuffer.
    goodPeaksCounter = 0
    shiftFOBufferToCurrentPeak(peakIndex)

  peakIndex += 1

return badOrderSections
```

Figure 11: Identifying Good and Bad Sections.

depending on the history of the firing order going into that peak. Firing on a D peak immediately after a B peak would be an error, since these peaks are nonadjacent.

When two nonadjacent peaks are processed in series, the algorithm treats this as the beginning of an invalid section. Next, it searches for the end of the section, which occurs when any valid sequence is detected for three peaks in a row. Three peaks are necessary because the correction process (handled in the next stage) can require changes as far as three peaks after the last invalid section. Since there is no way of knowing how many valid peaks have occurred since the beginning of the invalid section, we must build up the detected firing sequence one peak at a time, determining for each peak if it is part of a potentially valid firing order. We achieve this by circularly shifting the firing order so that the current peak's phase is in the starting position, then compare the following peaks to the forward and reverse phases of the new firing order. Because there might be reversals in this sequence of three peaks, it may be necessary to invert the firing order again on the second valid peak.

When the end of a bad section is detected, the algorithm saves the endpoints and contextual information of the section to a list. The contextual information contains whether the firing order reversed during the section, the timestamps at the start and end of the section, as well as on which axis the section occurred. The algorithm may then resume normal behavior and continue detecting reversals. This continues until all peaks in the axis are processed. The result is a partially annotated list of peaks and a list of invalid sections. This process is repeated for each axis.

Table 1: Solutions considered for badly ordered pairs and their effects on firing order. The beginning *ABCD* exemplifies a possible firing order prior to the badly ordered pair (indicated in red and underlined).

Detected	Corrected	Solution
ABCDAC	ABCDABC	Insert Forward Peak
	ABCDADC	Insert Reverse Peak
	ABCDC	Delete First Peak
	ABCDA	Delete Second Peak
ABCDAA	ABCDABA	Insert Forward Peak
	ABCDADA	Insert Reverse Peak
	ABCDA	Delete Peak

4.5 Heuristic Correction of Bad Sections

The invalid sections produced by the previous stage are processed individually in this stage. We recognize two forms of error: missing peaks and duplicate peaks. Missing peaks are flagged whenever two nonadjacent phases are detected in sequence; duplicate peaks are flagged when the same phase is detected twice in a row. Badly ordered sections can contain any number of badly ordered pairs, and we have observed up to 20 in a single section in our test set. Each badly ordered pair has multiple potentially valid corrections: 4 for missing peaks and 3 for duplicate peaks. This produces a large problem space, and the available data does not clearly identify the correct solution. Therefore, we have developed a heuristic approach to search the problem space and identify the solution that best fits a defined set of metrics.

Correcting a badly ordered section involves either deleting the first or second peak of a badly ordered pair, or inserting a new peak in between them. The inserted peak can be on either the forward phase or the reverse phase. These four changes represent all of the corrections the heuristic solver considers. All four apply to missing

```
// Container Function
def badSectionSolverContainer(badSections)
  for section in badSections:
    // each solution documents the bad section and peaks
    // solutions can be applied to correct the section
    solutions = []

    badPeakIndex = 0
    while badPeakIndex < len(section):

      if missingPeakError(badPeakIndex)
        solutions.append(missingSolutions(badPeakIndex))

      elif duplicatePeakError(badPeakIndex)
        solutions.append(duplicateSolutions(badPeakIndex))

      badPeakIndex++

    bestSolution, bestCost = searchSolutions(solutions, 0)
    applySolutionToTrace(bestSolution)
```

Figure 12: Heuristic Solver Container.

peak pairs. For duplicate peak pairs, deleting the first and deleting the second peak are functionally the same choice with respect to the firing order. Therefore, duplicate peak pairs have only the three solutions of deletion, insertion of a forward peak, and insertion of a reverse peak. This is illustrated in Table 1.

The solutions themselves can be compared on the quality of the peaks inserted or deleted. We do this by defining a cost function for each of the solutions that considers the height, prominence, proximity to other peaks, and several other characteristics of the peaks to insert or delete. The insertion cost functions operate on the segment of a trace in between the badly ordered peaks, searching for the best point to insert a new peak on the target phase. The deletion functions evaluate the first and second peaks of the badly ordered pair to determine which is more likely to be a false positive. Both use a relatively complex set of metrics and weightings, which we tuned by gradually identifying error cases in our test set and creating general metrics to correct them.

In addition to the heuristic costs for individual pairs, the solution must consider the entire section and its context to ensure the firing order is still valid. For example: consider the sequence ABCDACD. The badly ordered pair, AC, is identified as a missing peak pair and can be repaired by any of our four solutions. However, in the context of the surrounding sequence, different choices could produce an additional reversal, multiple reversals, or none at all. Based on the surrounding sequences, a badly ordered section should contain either an even or odd number of reversals; this is taken into account by the solver, which discards solutions that do not satisfy this constraint.

Pseudocode summarizing the heuristic process is presented in Figures 12 and 13. The solver consists of two portions, a non-recursive container and a recursive search function. The non-recursive container iterates over all badly ordered sections for each axis. The recursive search explores the solution space for each badly ordered section, eventually returning the solution with the lowest cost. The container then applies this solution to the trace, annotating the appropriate peaks with reversals. After the list of badly ordered sections is exhausted, the trace is fully annotated and can be used to produce a point cloud reconstruction of the 3D printed object.

```
def searchSolutions(solutions, cost)

  if len(solutions) == 0:
  // We've arrived at a leaf node in the search

    tempFixedPeaks = applySolutionToTrace(solution)
    // Applying solution makes a list of peaks
    // We process these with the same loop in Fig.16
    reversed = countReversals(tempFixedPeaks)

    if reversed % 2 == mustReverse
    // The solution is acceptable; return its cost
      return solution, cost
    else:
    // The solution doesn't match reversal behavior
    // Return infinite cost
      return solution, cost=Inf

  for peakPair in solutions:
    if peakPair.errorType == "missing":
      FWCost = insertCost(peakPair, forward)
      tempSolution = takeBranch(solutions, insertForward)
      FWSolution, FWCost =
        searchSolutions(tempSolution, cost + FWCost)

      RVCost = insertCost(peakPair, reverse)
      tempSolution = takeBranch(solutions, insertReverse)
      RVSolution, RVCost =
        searchSolutions(tempSolution, cost + RVCost)

      DCCost = deleteCost(peakPair, current)
      tempSolution = takeBranch(solutions, deleteCurrent)
      DCSolution, DCCost =
        searchSolutions(tempSolution, cost + DCCost)

      DNCost = deleteCost(peakPair, next)
      tempSolution = takeBranch(solutions, deleteNext)
      DNSolution, DNCost =
        searchSolutions(tempSolution, cost + DNCost)

      bestCost = minCost(FWCost, RVCost, DCCost, DNCost)
      if bestCost == FWCost
        bestSolution = FWSolution
      elif bestCost == RVCost
        bestSolution = RVSolution
      elif bestCost == DCCost
        bestSolution = DCSolution
      elif bestCost == DNCost
        bestSolution = DNSolution

        return bestSolution, bestCost

    elif peakPair.errorType == "duplicate"
    // Duplicate peaks are handled as above
```

Figure 13: Heuristic Recursive Solver.

4.6 Point Cloud Generation

Upon completion of the heuristic search and correction process, the trace contains valid timestamped peaks on the X-, Y-, Z-, and E-axes, all properly tagged wherever a reversal occurs. Although the firing order is now fully correct, there may still be discrepancies between the listed peaks and the real behavior of the printer. In the final stage, we implemented a state machine to track the position and behavior of the print head and translate its motions into a point cloud, with each entry an (X, Y, Z) tuple.

The state machine is initialized with a starting position of (0, 0, 0), then begins to iterate through the peaks in timestamp order. Processing a peak updates the position of the printhead based on the axis and movement direction of the peak. The E-axis is tracked to determine whether a new position is a part of an extruding or non-extruding move. Dwell annotations on the E-axis peaks mark the beginning and end for periods of extrusion. If the E-axis is inactive, the position is updated but not saved to the point cloud.

After every peak is consumed, the output from the state machine is the full point cloud representation of the print. By default, it is in units of discrete motor steps; we convert to millimeters to compare the reconstructions to the original models more easily. We used per-axis steps to millimeter ratios of: $6:1$ for X, $6:1$ for Y, and $95.2:1$ for Z. The E-axis is used only to track extrusion and non-extrusion, and does not need to be converted. The point cloud can be exported as a $.XYZ$ file, a common format for point clouds and meshes, and manipulated by external modeling programs.

5 EXPERIMENTAL EVALUATION

For our experiment, we selected a range of models that vary in size and geometric complexity. The simplest is a cube, 10mm on a side, which prints entirely in linear movements (G0 or G1 G-code commands). The remaining figures contain both linear and non-linear movements such as arcs and splines (G2, G3, and G5 G-code commands) and complex surfaces. The models are rendered in Figure 14.

These models are sliced[5] in Cura Lulzbot Edition 3.6.20 for printing on our Lulzbot Taz 6. The slicer settings are consistent across all models. We use an infill parameter of 20 % with 45 degree rotation and the default behavior of thicker base layers and full infill for top and bottom layers. We are therefore reconstructing realistic sliced and printed models, rather than handmade or simplified print patterns.

The reconstruction process runtime varied between 15 minutes to 2 hours, depending on model size. Given the threat model, we do not consider this restrictive.

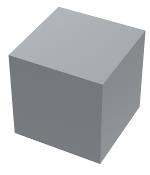
The results of our process are given in Table 2. Simpler and shorter models such as the Cube and Ninja Star are reconstructed perfectly, while some of the more complex models such as the Stanford Bunny exhibit visible misalignments across layers. Most models are only moderately distorted and remain recognizable. A human engineer could correct the errors and achieve the quality of the simple objects.

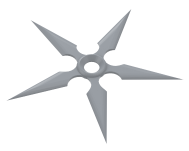
Quantifying these errors and determining their meaningfulness has presented a novel challenge not appropriately addressed in the literature. Chhetri et al. [11], who first demonstrated an acoustic side-channel attack on AM systems, quantified their reconstruction using average percentage of axis recognition and average movement distance error. They achieved 78.35 % accurate axis recognition and 17.82 % movement distance error for their test prints, which were zero-infill perimeter polygons.

Using these distance-oriented metrics, we achieved 100 % axis recognition (because all individual motors were instrumented). Across our tested models, we can calculate total distance error as $bad\_section\_steps/total\_steps$. This produces a maximum error of 1.7 % for the Rook model, and a minimum of 0.07 % for the Wrench. On average, the error was 0.79 %. This figure represents an upper bound of error, because not every step within a bad section is incorrect.

Gao et al. [17] evaluated their reconstruction of the infill path using a metric derived from the Hausdorff Distance. Their approach locates the point in the reconstructed path that is most distant from any point in the original path, and reports that distance as an upper bound on error.

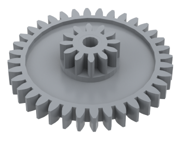
However, we believe all approaches to date for measuring accuracy have limited use in the context of AM reconstruction. This topic is explored further in Section 7.1.

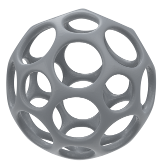

(a) Cube

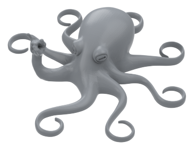

(b) Ninja Star

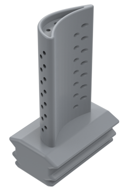
(c) Wrench

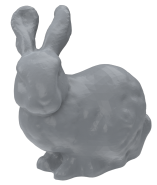
(d) Rook


(e) Gear


(f) Bucky Ball


(g) Octopus


(h) Turbine Blade


(i) Stanford Bunny

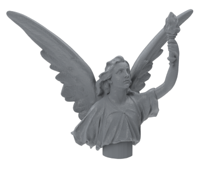
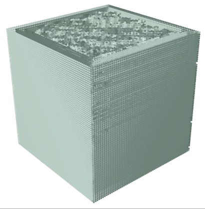
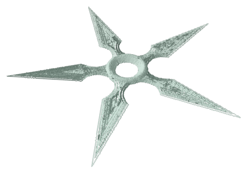
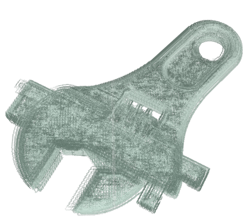
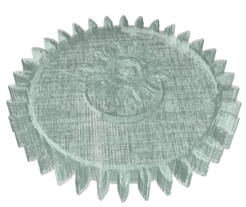
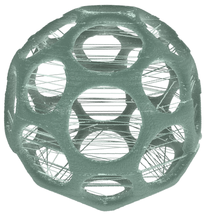
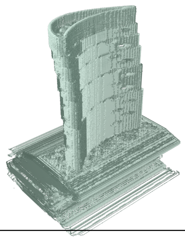
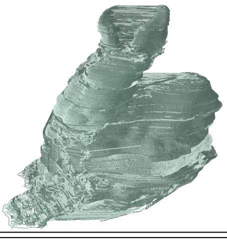

(j) Stanford Lucy

Figure 14: Original STL models used in the experiment.

[5]Slicer is. a common term for the software used to generate a printer-executable toolpath from a model file and printer configuration parameters, controlling elements such as infill patterns and support structures.

Table 2: Point cloud renderings and metrics of the reconstructed models. Any support structure is included in the rendering.

Metrics		Render	Metrics		Render
Name:	Cube		Name:	Ninja Star	
Print Duration:	13.63 min		Print Duration:	4.48 min	
Steps Traveled:	98,098		Steps Traveled:	65,534	
Points in Cloud:	80,196		Points in Cloud:	49,297	
Sections:	2,344		Sections:	1,384	
Bad Sections:	351		Bad Sections:	349	
B.S. Max. Length:	2		B.S. Max. Length:	4	
B.S. Avg. Length:	1.00		B.S. Avg. Length:	1.1	
Name:	Wrench		Name:	Rook	
Print Duration:	44.33 min		Print Duration:	49.98 min	
Steps Traveled:	774,063		Steps Traveled:	429,117	
Points in Cloud:	563,305		Points in Cloud:	340,909	
Sections:	21,445		Sections:	23,735	
Bad Sections:	5,334		Bad Sections:	6,497	
B.S. Max. Length:	7		B.S. Max. Length:	9	
B.S. Avg. Length:	1.05		B.S. Avg. Length:	1.15	
Name:	Gear		Name:	Bucky Ball	
Print Duration:	50 min		Print Duration:	154 min	
Steps Traveled:	728,078		Steps Traveled:	1,731,428	
Points in Cloud:	611,807		Points in Cloud:	1,126,327	
Sections:	25,893		Sections:	68,796	
Bad Sections:	5,572		Bad Sections:	21,269	
B.S. Max. Length:	6		B.S. Max. Length:	7	
B.S. Avg. Length:	1.15		B.S. Avg. Length:	1.09	
Name:	Octopus		Name:	Turbine Blade	
Print Duration:	66.58 min		Print Duration:	84.98 min	
Steps Traveled:	959,332		Steps Traveled:	879,667	
Points in Cloud:	715,662		Points in Cloud:	629,200	
Sections:	19,375		Sections:	25,192	
Bad Sections:	5,684		Bad Sections:	8,386	
B.S. Max. Length:	7		B.S. Max. Length:	5	
B.S. Avg. Length:	1.08		B.S. Avg. Length:	1.07	
Name:	Stan. Bunny		Name:	Stan. Lucy	
Print Duration:	302 min		Print Duration:	242 min	
Steps Traveled:	5,629,158		Steps Traveled:	3,827,019	
Points in Cloud:	4,490,563		Points in Cloud:	2,578,250	
Sections:	82,030		Sections:	98,730	
Bad Sections:	23,167		Bad Sections:	29,193	
B.S. Max. Length:	11		B.S. Max. Length:	7	
B.S. Avg. Length:	1.10		B.S. Avg. Length:	1.13	

6 FUNDAMENTAL LIMITATIONS

While our approach achieved significantly higher accuracy on more complex models than prior reconstruction attempts it still contains uncorrected errors. We suspect that some can be corrected, while others are fundamental to the approach. The most fundamental limitations are discussed below.

The most interesting limitation arises from the disconnect between the toolpath motions and the true shape of the printed part. When the printhead is positioned, the nozzle is placed slightly above the extrusion position to leave room for filament to extrude and attach without collisions. The extruded filament also has a diameter, which changes shape while it solidifies. During printing motions, the elasticity of the molten filament means that it will continue to extrude after the extruder motor stops moving. When planning a toolpath, the slicer attempts to account for these and other physical characteristics of the material and printing system. Many of these compensation techniques are adjusted by the user, vary according to printer and settings, and are largely opaque to our side-channel analysis. Producing a mesh directly from the reconstructed point cloud will produce a smaller mesh than the original because of these offsets, and it will contain small gaps due to flow-rate manipulation.

Another key limitation is that the approach cannot distinguish between body and supporting printed material. Many objects must be printed with support material to allow for large overhangs and prevent print defects such as sagging or non-adhesion. Given variation in support structure design and use, recognizing it automatically based on geometric characteristics would appear to be a difficult proposition.

These key limitations are based on the characteristics of the 3D printer, filament, and options available in the slicer. Undoubtedly, further study will uncover more limitations but also the means to overcome them. The techniques necessary to evaluate and compensate for these are beyond the scope of the present work.

7 FURTHER DISCUSSION

7.1 Measuring Error in Reconstruction

From a security perspective, the key benefit of disclosing this attack is to enable realistic risk analysis. For infringement, an attacker needs an accurate enough reconstruction method. As defenders, we must therefore assess the quality of the attack from this perspective. However, there are several fundamental issues in assessing reconstructions of printed models.

The point cloud generated by our method not only describes the exterior surface of the object, but also describes the interior points corresponding to infill, internal voids, support structures, and other features. Established methods for comparing similar paths, such as the Hausdorff Distance [21], can be adapted by restricting their application to individual layers (as in Gao et al. [17]), but even then questions remain.

Most importantly, it is difficult to ascertain if any existing metrics correspond meaningfully to the printability of an object. For example, one step of drift in our system corresponds to 0.16mm. This may seem like an inconsequential amount of drift, but it could be enough to completely collapse a support structure, which is often only one step wide. Drift on a support structure, even in this smallest possible increment, could lead to a failed print. If it occurs elsewhere, it may not even be visible to the naked eye. Figure 16 demonstrates this difference: the degree of drift in the Octopus print is consistently 1-4 steps, which is barely noticeable in the main body but would cause a print failure in the fine details. The impact of error on printability, therefore, is not necessarily proportional to simple measures like distance.

New metrics must be developed for assessing the quality of reconstructions in AM. The preceding discussion implies that these metrics may only be meaningful for specific AM technologies. Solving these and eventually further to-be-identified challenges requires dedicated investigations that are outside the scope of this paper.

7.2 Implications for Risk Management

The presented reconstruction attack poses a threat to outsourced FDM printing processes that circumvents encryption-based DRM technology. But what risk does this pose for a business relying on an AM service provider using FDM printing? A business facing such a risk must decide on a response: in-house their printing operations, deploy technical and policy strategies, or accept the risk and proceed without a mitigation strategy.

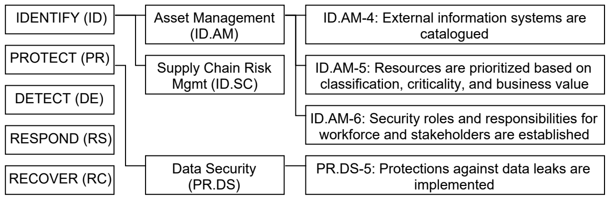

IDENTIFY (ID)
PROTECT (PR)
DETECT (DE)
RESPOND (RS)
RECOVER (RC)
Asset Management (ID.AM)
Supply Chain Risk Mgmt (ID.SC)
Data Security (PR.DS)
ID.AM-4: External information systems are catalogued
ID.AM-5: Resources are prioritized based on classification, criticality, and business value
ID.AM-6: Security roles and responsibilities for workforce and stakeholders are established
PR.DS-5: Protections against data leaks are implemented

Figure 15: Cybersecurity Framework Core subcategories [1].

To support this decision process, organizations might turn to NIST's guidance in the Cybersecurity Framework [1]. While this framework is voluntary in U.S. industry, its use is mandated within the U.S. government itself [23]. The Framework is hierarchical with five high-level risk management functions at the top, categories of security outcomes in the middle, and over 100 distinct outcome subcategories at the bottom. Each subcategory includes informative references to existing standards and best practices. Figure 15 provides a top-level view of the Framework's structure, with relevant categories and subcategories. Below, we provide specific examples where the threat of attack developed in this paper could be accounted for from within the cyber-security Framework.

A straightforward subcategory to apply is Data Security (PR.DS-5), stating that protections against data leaks should be implemented. Side-channel theft of design data would represent a substantial data leak. We have supported that no technical, cyber-security means exists to mitigate this risk, and therefore it should be addressed with other security measures. Within the framework this could include ID.AM-4, for cataloging external information systems (which must include the manufacturer's equipment). It could also include ID.AM-5, for prioritizing resources based on criticality (the existence of a practical attack increases criticality), and ID.AM-6 requires establishing cyber-security roles for the workforce and for third-party stakeholders (such as outsourced manufacturers). Further subcategories can also be applied to particular organizations and manufacturing arrangements. For example, subcategories of Supply Chain Risk Management specify routine assessment of suppliers for compliance with contractual obligations.

The Framework can provide a roadmap for addressing newly emerged threats such as this side-channel attack within the constraints of business objectives and organizational risk tolerance. The selected examples we provide are not comprehensive, and future work will be required to address this threat within this framework. Encryption may be futile in some scenarios, but business operations should not be. Security must reach beyond encryption when it isn't sufficient to the task.

7.3 Implications for Standards

The attack presented in this paper has shown that AM is susceptible to vulnerabilities beyond those addressed in well-known cyber-security and even cyber-physical security standards, such as those referenced by NIST in [38]. The processes for creating AM-specific digital representations, the subsequent transformations they undergo, and their eventual realization as a physical object create vulnerabilities not present in other manufacturing systems [20].

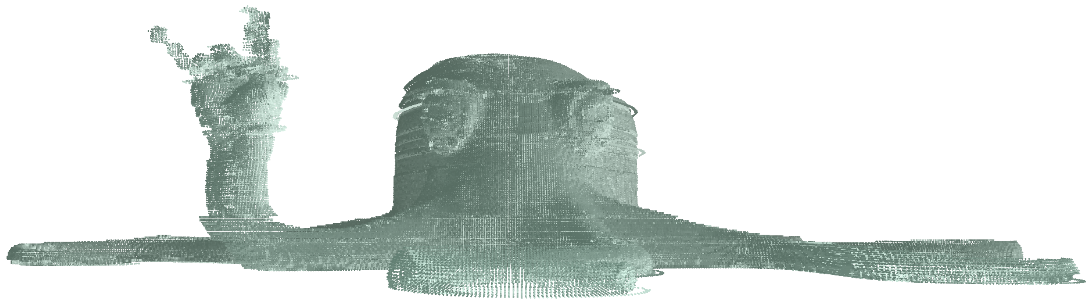


Figure 16: The reconstructed Octopus model, visualized as a point cloud. While the degree of drift present here would not affect the main body of the print, it would likely cause print failure in the fine detail of the raised hand.

These vulnerabilities are not restricted to a single location, and may exist across a supply chain. A standards-based approach to identifying and addressing the vulnerabilities would help organizations better secure their AM processes.

A publicly available standard for AM should provide a schematic reference for the design to product transformation, with detailed information models of each step of the process and associated security threats. These information models must break down the larger activity of "AM part production" into sub-activities, similar to [16]. A standards-based approach to modeling the information flow of AM processes will bring together the industry's knowledge of when and where security threats can exist and how they can be exploited. This could lead to documentation of threat models, attack surfaces, and vulnerabilities in AM, providing similar benefits to the Common Vulnerabilities and Exposures (CVE) List, a widely used list of cyber-security vulnerabilities and their identifiers [13].

In the scenario described in this paper, the sub-activity of a part build is exploited to compromise an AM part while bypassing cyber-security measures. A baseline reference of the part build sub-activity would include the exposure of design data via this side-channel. Standards can inform organizations in this way by enumerating the attack surfaces associated with AM activities, thereby helping to identify vulnerabilities and supporting risk analyses. Such standards would enable organizations to develop informed policies to better-secure their own systems.

8 CONCLUSION

To support distributed manufacturing with AM, multiple cyber-security models have emerged that protect digital representations of the design and other technical data. Under a Man-at-the-End threat model, which applies in the case of a malicious service provider or insider, the adversary's possession of the target hardware makes side-channel attacks possible. In this paper, we have demonstrated the effectiveness of a side-channel approach to technical data theft.

The instrumentation system and approach developed in this paper were able to reconstruct complete printed parts entirely from the electrical side-channel, without any information about the digital design files. Many of our tested models were reproduced with no noticeable aberrations, resulting in functionally identical counterfeit versions. Others demonstrated the limitations of a fully automated reconstruction, but still contained enough information that a dedicated manual process could recover the object in full. All observed errors were in the form of fixed-size shifts in the 1-10 step range corresponding to 0.16-1.6mm; these can be identified and repaired manually. These results constitute a significant step forward in the art of AM reconstruction.

Current defensive technologies would not be able to detect or prevent this attack. Cyber-security measures would be circumvented. The mutual-information reduction of Al-Faruque [5] or Chhetri [12] will be difficult to apply when the sampled channel is in direct control of the actuator. Future countermeasures research might investigate impedance monitoring to detect probing, creating signal artifacts that impede reconstruction, or identifying reconstructed products. Interesting work has been done in the industrial controls space on control signal watermarks to detect spoofed sensor readings [28, 34], which might be adapted to use here. It must be noted, however, that improvements to the approach could make such solutions unworkable. The most effective countermeasures available today appear to be the non-technical, procedural controls discussed in Section 7.2.

In a direct followup, we intend to test the applicability of this approach across multiple FDM printers, of the same and differing models. If the attack is portable between different models of printer and tolerant of drift in equipment behavior over time, it significantly changes the economics of the attack. This would enable a business model in which an adversary develops this technique and sells to less-skilled malicious insiders or manufacturers, much as is common in cyber-attacks today.

While this paper demonstrates the attack for an FDM 3D printer, the same approach should be viable for other AM processes and actuators. We believe that further refinement can push the accuracy of this approach to 100 %, producing reconstructions functionally identical to the original part. Consequently, this attack has direct and serious implications to the AM outsourcing model. No purely digital system of ensuring intellectual property protection is sufficient against a Man-at-the-End attack. Our work demonstrates that when side-channels are exploited, encryption is futile.

ACKNOWLEDGMENTS

This work was funded in part by the U.S. Department of Commerce, National Institute of Standards and Technology under Grant NIST-70NANB20H193.

REFERENCES

[1] 2018. National Institute of Standards and Technology Framework for Improving Critical Infrastructure Cybersecurity (Cybersecurity Framework), Version 1.1. (April 2018).

[2] Chris Adkins, Stephan Thomas, et al. 2019. System and method for data management and security for digital manufacturing. US Patent App. 16/128,988.

[3] akhani3D. 2020. akhani3D Production Additive Manufacturing Service Bureau. https://akhani3d.com.

[4] Mohammad Abdullah Al Faruque, Sujit Rokka Chhetri, Arquimedes Canedo, and Jiang Wan. 2016. Acoustic side-channel attacks on additive manufacturing systems. In *2016 ACM/IEEE 7th international conference on Cyber-Physical Systems (ICCPS)*. IEEE, 1–10.

[5] Mohammad Abdullah Al Faruque, Jiang Wan, and Sujit Rokka Chhetri. 2019. Defending side channel attacks in additive manufacturing systems. US Patent 10,212,185.

[6] Mohammed Albakri, Logan Sturm, Christopher B Williams, and Pablo Tarazaga. 2015. Non-destructive evaluation of additively manufactured parts via impedance-based monitoring. In *Solid Freeform Fabrication Symposium*. 1475–1490.

[7] Christian Bayens, Tuan Le, Luis Garcia, Raheem Beyah, Mehdi Javanmard, and Saman Zonouz. 2017. See No Evil, Hear No Evil, Feel No Evil, Print No Evil? Malicious Fill Patterns Detection in Additive Manufacturing. (2017).

[8] Sofia Belikovetsky, Yosef Solewicz, Mark Yampolskiy, Jinghui Toh, and Yuval Elovici. 2018. Digital Audio Signature for 3D Printing Integrity. *IEEE Transactions on Information Forensics and Security* (2018).

[9] Sofia Belikovetsky, Mark Yampolskiy, Jinghui Toh, and Yuval Elovici. 2016. dr0wned-Cyber-Physical Attack with Additive Manufacturing.

[10] Sujit Rokka Chhetri, Anomadarshi Barua, Sina Faezi, Francesco Regazzoni, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. 2019. Tool of spies: Leaking your ip by altering the 3d printer compiler. *IEEE Transactions on Dependable and Secure Computing* (2019).

[11] Sujit Rokka Chhetri, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. 2016. KCAD: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems. In *Proceedings of the 35th International Conference on Computer-Aided Design*. ACM, 74.

[12] Sujit Rokka Chhetri, Sina Faezi, and Mohammad Abdullah Al Faruque. 2018. Information leakage-aware computer-aided cyber-physical manufacturing. *IEEE Transactions on Information Forensics and Security* 13, 9 (2018), 2333–2344.

[13] The MITRE Corporation. 2021. Common Vulnerabilities and Exposures Database. https://cve.mitre.org.

[14] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. 2016. A data exfiltration and remote exploitation attack on consumer 3D printers. *IEEE Transactions on Information Forensics and Security* 11, 10 (2016), 2174–2186.

[15] General Electric. 2020. What is Additive Manufacturing? https://www.ge.com/additive/additive-manufacturing.

[16] Shaw C Feng, Paul Witherell, Gaurav Ameta, and Duck Bong Kim. 2017. Activity model for homogenization of data sets in laser-based powder bed fusion. *Rapid Prototyping Journal* (2017).

[17] Yang Gao, Borui Li, Wei Wang, Wenyao Xu, Chi Zhou, and Zhanpeng Jin. 2018. Watching and safeguarding your 3D printer: Online process monitoring against cyber-physical attacks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–27.

[18] Jacob Gatlin, Sofia Belikovetsky, Samuel B Moore, Yosef Solewicz, Yuval Elovici, and Mark Yampolskiy. 2019. Detecting sabotage attacks in additive manufacturing using actuator power signatures. *IEEE Access* 7 (2019), 133421–133432.

[19] GE Reports. 2015. *The FAA Cleared The First 3D Printed Part To Fly In A Commercial Jet Engine From GE.* Technical Report.

[20] Lynne MG Graves, Joshua Lubell, Wayne King, and Mark Yampolskiy. 2019. Characteristic Aspects of Additive Manufacturing Security From Security Awareness Perspectives. *IEEE Access* 7 (2019), 103833–103853.

[21] Felix Hausdorff. 1914. *Grundzüge der mengenlehre*. Vol. 7. von Veit.

[22] Avesta Hojjati, Anku Adhikari, Katarina Struckmann, Edward Chou, Thi Ngoc Tho Nguyen, Kushagra Madan, Marianne S Winslett, Carl A Gunter, and William P King. 2016. Leave your phone at the door: Side channels that reveal factory floor secrets. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 883–894.

[23] White House. 16 May, 2017. Exec. Order No. 13,800, 82 Fed. Reg. 22391. *Office of the Press Secretary* (16 May, 2017).

[24] Identify3D. 2020. Identify3D Info Sheet. https://identify3d.com/wp-content/uploads/2019/06/Identify3DInfosheet-1.pdf.

[25] Identify3D. 2021. Identify3D Home Webpage. https://identify3d.com/.

[26] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Annual international cryptology conference*. Springer, 388–397.

[27] Priyanka Mahesh, Akash Tiwari, Chenglu Jin, Panganamala R Kumar, AL Narasimha Reddy, Satish TS Bukkapatanam, Nikhil Gupta, and Ramesh Karri. 2020. A Survey of Cybersecurity of Digital Manufacturing. *Proc. IEEE* (2020).

[28] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. 2015. Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems Magazine* 35, 1 (2015), 93–109.

[29] Ltd. New Japan Radio Co. 2020. *Stepper Motor Basics*.

[30] Zachary Oligschlaeger, Benjamin Baltes, and Jennifer Chin. 2020. Secure 3D Printing. https://www.freepatentsonline.com/y2020/0326683.html

[31] Jaco Prinsloo, Saurabh Sinha, and Basie von Solms. 2019. A Review of Industry 4.0 Manufacturing Process Security Risks. *Applied Sciences* 9, 23 (2019), 5105.

[32] RapidDirect. 2020. RapidDirect Online CNC Machining and Prototype Manufacturing Service. https://www.rapiddirect.com.

[33] Sujit Rokka Chhetri and Mohammad Abdullah Al Faruque. 2020. *Data-Driven Defense Through Leakage Minimization*. Springer International Publishing, Cham, 67–90. https://doi.org/10.1007/978-3-030-37962-9_4

[34] Jose Rubio-Hernan, Luca De Cicco, and Joaquin Garcia-Alfaro. 2017. On the use of watermark-based schemes to detect cyber-physical attacks. *EURASIP Journal on Information Security* 2017, 1 (2017), 1–25.

[35] David R Safford and Monty Wiseman. 2019. Hardware Rooted Trust for Additive Manufacturing. *IEEE Access* 7 (2019), 79211–79215.

[36] SciPy.org. 2020. Documentation of scipy.signal.find_peaks. https://docs.scipy.org/.

[37] Chen Song, Feng Lin, Zhongjie Ba, Kui Ren, Chi Zhou, and Wenyao Xu. 2016. My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3d printers. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 895–907.

[38] Keith Stouffer, Timothy Zimmerman, CheeYee Tang, Michael Pease, Joshua Lubell, Jeffrey Cichonski, and John McCarthy. 2020. *Cybersecurity Framework Version 1.1 Manufacturing Profile*. Technical Report. National Institute of Standards and Technology.

[39] Logan Sturm, Mohammed Albakri, Christopher B Williams, and Pablo Tarazaga. 2016. In-situ Detection of Build Defects in Additive Manufacturing via Impedance-Based Monitoring. (2016), 1458–1478.

[40] L Sturm, CB Williams, JA Camelio, J White, and R Parker. 2014. Cyber-physical vunerabilities in additive manufacturing systems. *Context* 7 (2014), 8.

[41] thyssenkrupp. 2020. thyssenkrupp 3D-Druck TechCenter: Additive Manufacturing. https://www.thyssenkrupp-additive-manufacturing.com.

[42] Treatstock. 2020. Treatstock Smart Manufacturing Platform. https://www.treatstock.com.

[43] Wim Van Eck. 1985. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security* 4, 4 (1985), 269–286.

[44] Terry Wohlers. 2017. *Wohlers Report 2017 3D Printing and Additive Manufacturing State of the Industry Annual Worldwide Progress Report*. Wohlers Associates, Inc., Fort Collins, Colorado, USA. www.wohlersassociates.com.

[45] M Yampolskiy, TR Andel, JT McDonald, WB Glisson, and A Yasinsac. 2014. Towards Security of Additive Layer Manufacturing. WiP presented at The 30th Annual Computer Security Applications Conference (ACSAC) 2014.

[46] Mark Yampolskiy, Todd R Andel, J Todd McDonald, William B Glisson, and Alec Yasinsac. 2014. Intellectual property protection in additive layer manufacturing: Requirements for secure outsourcing. In *Proceedings of the 4th Program Protection and Reverse Engineering Workshop*. 1–9.

[47] Mark Yampolskiy, Peter Horvath, Xenofon D Koutsoukos, Yuan Xue, and Janos Sztipanovits. 2013. Taxonomy for description of cross-domain attacks on CPS. In *Proceedings of the 2nd ACM international conference on High confidence networked systems*. ACM, 135–142.

[48] Mark Yampolskiy, Wayne E King, Jacob Gatlin, Sofia Belikovetsky, Adam Brown, Anthony Skjellum, and Yuval Elovici. 2018. Security of Additive Manufacturing: Attack Taxonomy and Survey. *Additive Manufacturing* (2018).

[49] Shih-Yuan Yu, Arnav Vaibhav Malawade, Sujit Rokka Chhetri, and Mohammad Abdullah Al Faruque. 2020. Sabotage attack detection for additive manufacturing systems. *IEEE Access* 8 (2020), 27218–27231.

[50] Steven Eric Zeltmann, Nikhil Gupta, Nektarios Georgios Tsoutsos, Michail Maniatakos, Jeyavijayan Rajendran, and Ramesh Karri. 2016. Manufacturing and Security Challenges in 3D Printing. *JOM* (2016), 1–10.